

THE RING PROGRAMMING LANGUAGE

```
changingoperator + plus  
changingkeyword SEE PRINT  
  
Print 5 plus 5  
  
changingoperator plus +  
changingkeyword PRINT SEE
```

Syntax
Flexibility

```
Load "gameengine.ring"  
  
func main  
{  
    oGame = New Game  
    {  
        title = "My First Game"  
        sprite  
        {  
            type = GE_TYPE_PLAYER  
            x=400 y=400 width=100 height=100  
            file = "images/player.png"  
            transparent = true  
            Animate=false  
            Move=true  
            Scaled=true  
        }  
    }  
}
```

Declarative
Programming

```
# Natural Code  
new program {  
    Accept 2 numbers then print the sum  
}  
  
# Natural Code Implementation  
class program  
    # Keywords  
    Accept=0 numbers=0 then=0 print=0 the=0 sum=0  
  
    # Execution  
    func braceexprval x  
        value = x  
    func getnumbers  
        for x=1 to value  
            see "Enter Number (" + x + ") : " give nNumber  
            aNumbers + nNumber  
        next  
    func getsum  
        nSum = 0  
        for x in aNumbers nSum += x next  
        see "The Sum : " + nSum  
    private  
        value=0 aNumbers=[]
```

Natural
Language
Programming

The Complete Reference

Ring Documentation

Release 1.6

Eng. Mahmoud Fayed and Dr. Atif Alamri

November 30, 2017

1	Applications developed in little hours	1
1.1	Quotes about Ring	1
1.2	FetchStockData Application	3
1.3	Fifteen Puzzle Game 2	4
1.4	Google API Shortener Application	5
1.5	Analog Clock	6
1.6	TicTacToe Game	7
1.7	Squares Puzzle Game	8
1.8	Video-Music-Player Application	9
1.9	Calculator Application	10
1.10	Windows StartUp Manager Application	11
1.11	Werdy Application	12
1.12	Samples in this book	13
1.13	Innovative	19
1.14	Practical	20
2	Introduction	22
2.1	Motivation	22
2.2	Ring and other languages	23
2.3	History	23
2.4	Features	24
2.5	License	26
3	Language Design	28
3.1	Why Ring?	28
3.2	Designed for a Clear Goal	28
3.3	Simple	28
3.4	Trying to be natural	29
3.5	Encourage Organization	30
3.6	Compact Syntax	30
3.7	Define Natural Statements	31
3.8	Define Declarative Languages	33
3.9	Transparent Implementation	34
3.10	Visual Implementation	35
3.11	Smart Garbage Collector	36
4	What is new in Ring 1.6?	37
4.1	List of changes and new features	37
4.2	Employee Application	37
4.3	New Tool: Ring2EXE	38
4.4	Better Ring For Android	38

4.5	New Tool: Folder2qrc	40
4.6	Better Scripts for building Ring	40
4.7	RingConsoleColors Extension	40
4.8	RingMurmurHash Extension	41
4.9	Better Ring Notepad	41
4.10	Better RingQt	43
4.11	Better StdLib	44
4.12	Better RingVM	44
4.13	Better RingREPL	44
4.14	Using Tab instead of char(9)	45
4.15	Using CR as Carriage return	45
4.16	Using the ! operator as not	45
4.17	Using && and operators	45
4.18	Using ? to print expression then new line	46
5	What is new in Ring 1.5?	48
5.1	List of changes and new features	48
5.2	Video-Music-Player Application	48
5.3	Windows StartUp Manager Application	49
5.4	Calculator Application	50
5.5	Better Ring Notepad	50
5.6	Better StdLib	53
5.7	Better WebLib	54
5.8	Better RingQt	57
5.9	Better Objects Library	59
5.10	RingFreeGLUT Extension	59
5.11	RingOpenGL Extension	69
5.12	Better Code Generator for Extensions	72
5.13	Better Documentation Generator for Extensions	73
5.14	Ring VM - Tracing Functions	73
5.15	Trace Library and Interactive Debugger	77
5.16	More Syntax Flexibility	79
5.17	Type Hints Library	80
5.18	Better Quality	81
5.19	What is new in Ring 1.5.1?	81
5.20	What is new in Ring 1.5.2?	86
5.21	What is new in Ring 1.5.3?	87
5.22	What is new in Ring 1.5.4?	90
6	What is new in Ring 1.4?	92
6.1	List of changes and new features	92
6.2	Change: Basic Extensions are separated from RingVM	92
6.3	The Natural Library	93
6.4	New Style is added to Ring Notepad	95
6.5	RingREPL	96
6.6	Convert between Numbers and Bytes	96
6.7	Better StdLib	97
6.8	Better WebLib	97
6.9	Better RingQt	99
6.10	Qt Class Convertor	99
6.11	What is new in Ring 1.4.1?	100
7	What is new in Ring 1.3?	104
7.1	List of changes and new features	104

7.2	Better RingQt	104
7.3	Better Ring Notepad	108
7.4	Ring mode for Emacs Editor	111
7.5	Better StdLib	112
7.6	Better LoopExit Command	112
7.7	New Functions	113
7.8	Return Self by Reference	113
7.9	Using '<' and ':' operators as 'from' keyword	114
7.10	Embedding Ring in Ring without sharing the State	114
7.11	RingZip Library	115
7.12	Form Designer	115
8	What is new in Ring 1.2?	117
8.1	List of changes and new features	117
8.2	New Functions	117
8.3	Better Functions	118
8.4	Better Ring Notepad	118
8.5	Better RingQt	119
8.6	Objects Library for RingQt	119
8.7	RingLibCurl	120
8.8	Better Call Command	121
8.9	Using NULL instead of NULLPointer()	121
8.10	Display Warnings Option	121
8.11	Better Quality	122
9	What is new in Ring 1.1?	123
9.1	List of changes and new features	123
9.2	Better Natural Language Programming Support	123
9.3	Generate/Execute Ring Object Files (*.ringo)	124
9.4	Syntax Flexibility and different styles for I/O and Control Structures	124
9.5	New Functions and Changes	126
9.6	StdLib functions and classes written in Ring	127
9.7	RingLibSDL	129
9.8	Demo Project - Game Engine for 2D Games	129
9.9	RingSQLite	130
9.10	Better Code Generator for Extensions	131
9.11	Using Self.Attribute in the Class Region to define new attributes	131
9.12	Using This.Attribute in nested Braces inside the Class Methods	131
9.13	Better Documentation	132
10	Building From Source Code	133
10.1	Building using Microsoft Windows	133
10.2	Building using Ubuntu Linux	135
10.3	Building using Fedora Linux	137
10.4	Building using MacOS X	138
10.5	Building using CMake	140
11	How to contribute?	141
11.1	Special thanks to contributors	141
11.2	Documentation	141
11.3	Testing	141
11.4	Samples	142
11.5	Applications	142
11.6	Editors Support	142
11.7	Libraries in Ring	142

11.8	Extensions in C/C++	142
11.9	Compiler and Virtual Machine (VM)	142
11.10	Ideas and suggestions	142
12	Getting Started - First Style	143
12.1	Hello World	143
12.2	Run the program	143
12.3	Create Executable File	143
12.4	Not Case-Sensitive	143
12.5	Multi-Line literals	143
12.6	Getting Input	144
12.7	No Explicit End For Statements	144
12.8	Using ? to print expression then new line	144
12.9	Writing Comments	145
13	Getting Started - Second Style	146
13.1	Hello World	146
13.2	Run the program	146
13.3	Create Executable File	146
13.4	Not Case-Sensitive	146
13.5	Multi-Line literals	146
13.6	Getting Input	147
13.7	No Explicit End For Statements	147
13.8	Writing Comments	147
14	Getting Started - Third Style	148
14.1	Hello World	148
14.2	Run the program	148
14.3	Create Executable File	148
14.4	Not Case-Sensitive	149
14.5	Multi-Line literals	149
14.6	Getting Input	149
14.7	No Explicit End For Statements	149
14.8	Writing Comments	149
15	Using Ring Notepad	151
15.1	Ring Notepad - Main Window	151
15.2	Creating and running your first Console Application	152
15.3	Creating and running your first GUI/Mobile Application	154
15.4	Creating and running your first Web Application	155
15.5	Creating and running your first Desktop/Mobile Game	157
15.6	The Main File in the Project	158
15.7	The File Menu	159
15.8	The Edit Menu	159
15.9	The View Menu	160
15.10	The Program Menu	162
15.11	The Browser Menu	162
15.12	The Tools Menu	163
15.13	The Distribute Menu	163
15.14	The Help Menu	164
16	Using Other Code Editors	165
16.1	Using Notepad++	165
16.2	Using Geany	166
16.3	Using Atom	167

16.4	Using Sublime Text 2	168
16.5	Using Visual Studio IDE	169
16.6	Using Emacs Editor	170
17	Variables	171
17.1	Dynamic Typing	171
17.2	Deep Copy	171
17.3	Weakly Typed	172
18	Operators	173
18.1	Arithmetic Operators	173
18.2	Relational Operators	173
18.3	Logical Operators	173
18.4	Bitwise Operators	174
18.5	Assignment Operators	174
18.6	Misc Operators	174
18.7	Operators Precedence	175
19	Control Structures - First Style	176
19.1	Branching	176
19.2	Looping	177
19.3	Using The Step option with For in	178
19.4	Using For in to modify lists	179
19.5	Do Again Loop	179
19.6	Exit Command	179
19.7	Exit from two loops	180
19.8	Loop Command	180
19.9	Exit/Loop inside sub functions	180
19.10	Short-circuit evaluation	181
19.11	Comments about evaluation	181
20	Control Structures - Second Style	183
20.1	Branching	183
20.2	Looping	184
20.3	Exceptions	185
21	Control Structures - Third Style	186
21.1	Branching	186
21.2	Looping	187
21.3	Exceptions	189
22	Getting Input	191
22.1	Give Command	191
22.2	GetChar() Function	191
22.3	Input() Function	192
23	Functions - First Style	193
23.1	Define Functions	193
23.2	Call Functions	193
23.3	Declare parameters	194
23.4	Send Parameters	194
23.5	Main Function	194
23.6	Variables Scope	195
23.7	Return Value	195
23.8	Recursion	196

24 Functions - Second Style	197
24.1 Define Functions	197
24.2 Call Functions	197
24.3 Declare parameters	198
24.4 Send Parameters	198
24.5 Main Function	198
24.6 Variables Scope	199
24.7 Return Value	199
24.8 Recursion	200
25 Functions - Third Style	201
25.1 Define Functions	201
25.2 Call Functions	201
25.3 Declare parameters	202
25.4 Send Parameters	202
25.5 Main Function	203
25.6 Variables Scope	203
25.7 Return Value	203
25.8 Recursion	204
26 Program Structure	205
26.1 Source Code File Sections	205
26.2 Using Many Source Code Files	205
27 Lists	206
27.1 Create Lists	206
27.2 Add Items	206
27.3 Get List Size	207
27.4 Delete Item From List	207
27.5 Get List Item	207
27.6 Set List Item	208
27.7 Search	208
27.8 Sort	209
27.9 Reverse	209
27.10 Insert Items	210
27.11 Nested Lists	210
27.12 Copy Lists	210
27.13 First-class lists	211
27.14 Using Lists during definition	211
27.15 Passing Lists to Functions	211
27.16 Access List Items by String Index	212
27.17 Passing Parameters Using List	212
27.18 Swap Items	213
28 Strings	214
28.1 String Literals	214
28.2 Get String Length	214
28.3 Convert Letters Case	214
28.4 Access String Letters	215
28.5 Left() Function	215
28.6 Right() Function	215
28.7 Trim() Function	216
28.8 Copy() Function	216
28.9 Lines() Function	216
28.10 Substr() Function	217

28.11 Find substring	217
28.12 Get substring from position to end	217
28.13 Get Number of Characters From Position	217
28.14 Transform Substring To Another Substring	217
28.15 strcmp() Function	218
28.16 str2list() and list2str() Functions	218
29 Date and Time	220
29.1 Clock() Function	220
29.2 ClocksPerSecond() Function	220
29.3 Time() Function	220
29.4 Date() Function	221
29.5 TimeList() Function	221
29.6 AddDays() Function	222
29.7 DiffDays() Function	222
29.8 EpochTime() Function	223
30 Check Data Type and Conversion	224
30.1 Check Data Type	224
30.2 IsString() Function	224
30.3 IsNumber() Function	224
30.4 IsList() Function	225
30.5 Type() Function	225
30.6 IsNULL() Function	225
30.7 Check Character	226
30.8 IsAlNum() Function	226
30.9 IsAlpha() Function	226
30.10 IsCntrl() Function	227
30.11 IsDigit() Function	227
30.12 IsGraph() Function	227
30.13 IsLower() Function	227
30.14 IsPrint() Function	228
30.15 IsPunct() Function	228
30.16 IsSpace() Function	228
30.17 IsUpper() Function	228
30.18 IsXdigit() Function	229
30.19 Conversion	229
30.20 Number() Function	229
30.21 String() Function	229
30.22 Ascii() Function	230
30.23 Char() Function	230
30.24 Hex() Function	230
30.25 Dec() Function	230
30.26 Str2hex() Function	231
30.27 Hex2str() Function	231
31 Mathematical Functions	232
31.1 List of functions	232
31.2 Example	232
31.3 Random() Function	234
31.4 Unsigned() Function	235
31.5 Decimals() Functions	235
31.6 Using _ in numbers	236
31.7 Using f after numbers	236

32 Files	237
32.1 Read() Function	238
32.2 Write() Function	238
32.3 Dir() Function	238
32.4 Rename() Function	239
32.5 Remove() Function	239
32.6 Fopen() Function	239
32.7 Fclose() Function	240
32.8 Fflush() Function	240
32.9 Freopen() Function	240
32.10 Tempfile() Function	241
32.11 Tempname() Function	241
32.12 Fseek() Function	241
32.13 Ftell() Function	241
32.14 Rewind() Function	242
32.15 Fgetpos() Function	242
32.16 Fsetpos() Function	242
32.17 Clearerr() Function	242
32.18 Feof() Function	242
32.19 Ferror() Function	242
32.20 Perror() Function	243
32.21 Fgetc() Function	243
32.22 Fgets() Function	243
32.23 Fputc() Function	243
32.24 Fputs() Function	243
32.25 Ungetc() Function	243
32.26 Fread() Function	244
32.27 Fwrite() Function	244
32.28 Fexists() Function	244
32.29 Example	244
32.30 Numbers and Bytes	246
33 System Functions	247
33.1 System() Function	247
33.2 SysGet() Function	248
33.3 IsMSDOS() Function	248
33.4 IsWindows() Function	248
33.5 IsWindows64() Function	248
33.6 IsUnix() Function	248
33.7 IsMacOSX() Function	249
33.8 IsLinux() Function	249
33.9 IsFreeBSD() Function	249
33.10 IsAndroid() Function	249
33.11 Example	249
33.12 Windowsnl() Function	250
33.13 Get Command Line Arguments	250
33.14 Get Active Source File Name	251
33.15 PrevFileName() Function	251
33.16 CurrentDir() Function	252
33.17 ExeFileName() Function	252
33.18 ChDir() Function	252
33.19 ExeFolder() Function	252
33.20 Version() Function	252
33.21 Shutdown() Function	253

34 Eval() and Debugging	254
34.1 Try/Catch/Done	254
34.2 Eval() Function	254
34.3 Raise() Function	255
34.4 Assert() Function	256
35 Demo Programs	257
35.1 Language Shell	257
35.2 Main Menu	258
36 ODBC Functions	261
36.1 odbc_init() Function	261
36.2 odbc_drivers() Function	262
36.3 odbc_datasources() Function	262
36.4 odbc_close() Function	262
36.5 Print List of ODBC Drivers	262
36.6 Print List of ODBC Data Sources	263
36.7 odbc_connect() Function	264
36.8 odbc_disconnect() Function	264
36.9 Open and Close Connection	264
36.10 odbc_execute() Function	264
36.11 odbc_colcount() Function	264
36.12 odbc_fetch() Function	265
36.13 odbc_getdata() Function	265
36.14 Execute Query and Print Result	265
36.15 odbc_tables() Function	265
36.16 odbc_columns() Function	266
36.17 odbc_autocommit() Function	267
36.18 odbc_commit() Function	267
36.19 odbc_rollback() Function	267
36.20 Transactions and Using Commit and Rollback	267
36.21 Save and Restore images	268
37 MySQL Functions	270
37.1 MySQL_Info() Function	270
37.2 MySQL_Init() Function	271
37.3 MySQL_Error() Function	271
37.4 MySQL_Connect() Function	271
37.5 MySQL_Close() Function	271
37.6 MySQL_Query() Function	271
37.7 Create Database	272
37.8 Create Table and Insert Data	272
37.9 MySQL_Insert_ID() Function	273
37.10 MySQL_Result() Function	274
37.11 MySQL_Next_Result() Function	274
37.12 Print Query Result	274
37.13 MySQL_Columns() Function	274
37.14 MySQL_Result2() Function	275
37.15 MySQL_Escape_String() Function	276
37.16 Save Image inside the database	276
37.17 Restore Image From The Database	277
37.18 MySQL_AutoCommit() Function	277
37.19 MySQL_Commit() Function	277
37.20 MySQL_Rollback() Function	277

37.21 Transaction Example	277
38 SQLite Functions	279
38.1 sqlite_init() function	279
38.2 sqlite_open() function	279
38.3 sqlite_execute() function	279
38.4 sqlite_close() function	279
38.5 Example	280
39 Security and Internet Functions	282
39.1 MD5() Function	282
39.2 SHA1() Function	283
39.3 SHA256() Function	283
39.4 SHA512() Function	283
39.5 SHA384() Function	284
39.6 SHA224() Function	284
39.7 Encrypt() Function	284
39.8 Decrypt() Function	285
39.9 Encryption and Decryption Example	285
39.10 File Hash	285
39.11 Randbytes() Function	286
39.12 Download() Function	286
39.13 SendEmail() Function	286
40 Object Oriented Programming (OOP)	287
40.1 Classes and Objects	287
40.2 Access Objects Using Braces	289
40.3 Composition	290
40.4 Setter and Getter	291
40.5 Private Attributes and Methods	291
40.6 Operator Overloading	292
40.7 Inheritance	293
40.8 Dynamic Attributes	294
40.9 Packages	294
40.10 Printing Objects	295
40.11 Find() and List of Objects	295
40.12 Sort() and list of objects	296
40.13 Using Self.Attribute and Self.Method()	298
40.14 Using This.Attribute and This.Method()	299
41 Functional Programming	300
41.1 Pure Functions	300
41.2 First-class Functions	301
41.3 Higher-order Functions	301
41.4 Anonymous and Nested Functions	302
41.5 Equality of functions	303
42 Reflection and Meta-programming	305
42.1 locals() Function	306
42.2 globals() Function	306
42.3 functions() Function	307
42.4 cfunctions() Function	307
42.5 islocal() Function	308
42.6 isglobal() Function	308
42.7 isfunction() Function	309

42.8	isfunction() Function	309
42.9	packages() Function	309
42.10	ispackage() Function	310
42.11	classes() Function	311
42.12	isclass() Function	311
42.13	packageclasses() Function	312
42.14	ispackageclass() Function	312
42.15	classname() Function	313
42.16	objectid() Function	313
42.17	isobject() Function	314
42.18	attributes() Function	314
42.19	methods() Function	314
42.20	isattribute() Function	315
42.21	isprivateattribute() Function	315
42.22	ismethod() Function	316
42.23	isprivatemethod() Function	316
42.24	addattribute() Function	317
42.25	addmethod() Function	317
42.26	getattribute() function	319
42.27	setattribute() function	320
42.28	mergemethods() Function	321
42.29	packagename() Function	322
43	Stdlib Functions	323
43.1	Puts() function	323
43.2	Print() function	323
43.3	Print2Str() Function	324
43.4	GetString() function	324
43.5	GetNumber() function	324
43.6	AppPath() function	324
43.7	JustFilePath() function	324
43.8	JustFileName() function	325
43.9	Value() function	325
43.10	Times() function	325
43.11	Map() function	326
43.12	Filter() function	326
43.13	Split() function	326
43.14	SplitMany() function	327
43.15	NewList() function	327
43.16	Capitalized() function	327
43.17	IsSpecial() function	328
43.18	IsVowel() function	328
43.19	LineCount() function	328
43.20	Factorial() function	328
43.21	Fibonacci() function	329
43.22	IsPrime() function	329
43.23	Sign() function	329
43.24	List2File() function	330
43.25	File2List() function	330
43.26	StartsWith() function	330
43.27	EndsWith() function	330
43.28	GCD() function	331
43.29	LCM() function	331
43.30	SumList() function	331

43.31	ProdList() function	332
43.32	EvenOrOdd() function	332
43.33	Factors() function	332
43.34	Palindrome() function	333
43.35	IsLeapYear() function	333
43.36	BinaryDigits() function	333
43.37	MatrixMulti() function	334
43.38	MatrixTrans() function	334
43.39	DayOfWeek() function	334
43.40	Permutation() function	335
43.41	ReadLine() function	335
43.42	SubString() function	335
43.43	ChangeString() function	336
43.44	Sleep() function	336
43.45	IsMainSourceFile() function	336
43.46	DirExists() function	337
43.47	MakeDir() function	337
43.48	Fsize() function	337
43.49	TrimAll() function	337
43.50	TrimLeft() function	338
43.51	TrimRight() function	338
43.52	EpochTime() function	338
43.53	SystemCmd() Function	338
43.54	ListAllFiles() Function	338
43.55	SystemSilent() Function	339
43.56	OSCreateOpenFolder() Function	339
43.57	OSCopyFolder() Function	339
43.58	OSDeleteFolder() Function	339
43.59	OSCopyFile() Function	340
43.60	OSDeleteFile() Function	340
43.61	OSRenameFile() Function	340
44	Stdlib Classes	341
44.1	StdBase Class	341
44.2	String Class	342
44.3	List Class	344
44.4	Stack Class	346
44.5	Queue Class	346
44.6	HashTable Class	347
44.7	Tree Class	348
44.8	Math Class	349
44.9	DateTime Class	353
44.10	File Class	354
44.11	System Class	355
44.12	Debug Class	356
44.13	DataType Class	357
44.14	Conversion Class	358
44.15	ODBC Class	359
44.16	MySQL Class	360
44.17	SQLite Class	361
44.18	Security Class	362
44.19	Internet Class	363
45	Declarative Programming using Nested Structures	364

45.1	Creating Objects inside Lists	364
45.2	Composition and Returning Objects and Lists by Reference	365
45.3	Executing code after the end of object access	367
45.4	Declarative Programming on the top of Object-Oriented	367
45.5	More beautiful Code	368
46	Natural Language Programming	370
46.1	History	370
46.2	Example	370
46.3	Change the Ring Keyword 'And'	371
46.4	Change the Ring Operator '+'	372
46.5	Change the '=' operator to 'is'	373
46.6	Using Eval() with our Natural Code	374
46.7	BraceStart and BraceEnd Methods	375
46.8	BraceExprEval Method	376
46.9	Real Natural Code	376
46.10	BraceError() Method	377
46.11	Clean Natural Code	378
47	Using the Natural Library	380
47.1	Natural Library - Demo Program	380
47.2	Defining Commands	382
47.3	Natural Library - Operators	385
47.4	Defining commands using classes	385
48	Web Development (CGI Library)	387
48.1	Configure the Apache web server	387
48.2	Ring CGI Hello World Program	388
48.3	Hello World Program using the Web Library	388
48.4	Web Library Features	389
48.5	HTTP Get Example	389
48.6	HTTP POST Example	394
48.7	Upload Files	396
48.8	Cookies	399
48.9	URL Encode	401
48.10	Templates	402
48.11	HTML Special Characters	404
48.12	Hash Functions	405
48.13	Random Image	407
48.14	HTML Lists	408
48.15	HTML Tables	410
48.16	Gradient	411
48.17	Generating Pages using Objects	412
48.18	HtmlPage Class	416
48.19	Using Bootstrap Library using Functions	417
48.20	Using Bootstrap Library using Objects	418
48.21	CRUD Example using MVC	421
48.22	Users registration and Login	423
48.23	Database, ModelBase & ControllerBase classes	429
48.24	WebLib API	434
48.25	Application Class	436
48.26	Page Class	436
48.27	ScriptFunctions Class	438
48.28	StyleFunctions Class	439

48.29	WebPage Class	439
48.30	HtmlPage Class	440
49	Using RingLibCurl	441
49.1	Get Request	441
49.2	Post Request	441
49.3	Facebook Login	441
49.4	Save Output to String	442
49.5	Get Stock Data From Yahoo	443
50	Using RingZip	445
50.1	Create Zip File	445
50.2	Extract Zip File	445
50.3	Print Files in Zip file	445
50.4	Using RingZip Classes	445
50.5	Zip Class Reference	447
50.6	ZipEntry Class Reference	447
51	Graphics and 2D Games programming using RingAllegro	448
51.1	Drawing, Animation and Input	448
51.2	Using TrueType Fonts	455
51.3	Playing Sound Files	456
51.4	Scaling and Rotating Images	457
51.5	Display Transparent Image	458
51.6	Using Threads	459
52	Using RingLibSDL	462
52.1	Create Window	462
52.2	Display Image	462
52.3	Switch between two images	463
52.4	Draw Rectangle	463
52.5	Display PNG Images	464
52.6	Use TTF Fonts	464
52.7	Display Transparent Images	464
52.8	Close Window Event	465
52.9	Mouse Events	466
52.10	Play Sound	467
53	Demo Project - Game Engine for 2D Games	469
53.1	Project Layers	469
53.2	Graphics Library bindings	469
53.3	Interface to graphics library	470
53.4	Game Engine Classes	470
53.5	Games Layer	470
53.6	Game Class	471
53.7	GameObject Class	471
53.8	Sprite Class	472
53.9	Text Class	472
53.10	Animate Class	473
53.11	Sound Class	473
53.12	Map Class	473
53.13	Using the Game Engine - Creating the Game Window	474
53.14	Using the Game Engine - Drawing Text	474
53.15	Using the Game Engine - Moving Text	475
53.16	Using the Game Engine - Playing Sound	477

53.17	Using the Game Engine - Animation	478
53.18	Using the Game Engine - Animation and Functions	479
53.19	Using the Game Engine - Sprite - Automatic Movement using Keyboard	481
53.20	Using the Game Engine - Sprite - Keypress event	482
53.21	Using the Game Engine - Sprite - Mouse event	482
53.22	Using the Game Engine - Sprite - State event	483
53.23	Using the Game Engine - Animate - Events	484
53.24	Using the Game Engine - Map	486
53.25	Using the Game Engine - Map Events	488
53.26	Using the Game Engine - Object and Drawing	489
53.27	Stars Fighter Game	492
53.28	Flappy Bird 3000 Game	500
53.29	Super Man 2016 Game	507
54	Building Games For Android	519
54.1	Download Requirements and Update the Android SDK	519
54.2	Project Folder	519
54.3	Building the project	520
55	Using RingOpenGL and RingFreeGLUT for 3D Graphics	522
55.1	Samples Source (Authors)	522
55.2	What is RingOpenGL?	522
55.3	What is RingFreeGLUT?	523
55.4	The First Window using RingFreeGLUT	523
55.5	Drawing using RingOpenGL	524
55.6	The First Triangle	526
55.7	Window Resize Event	527
55.8	Triangle Rotation	528
55.9	Keyboard Events and Colors	529
55.10	The Camera	532
55.11	Mouse Events	539
55.12	Menu Events	542
55.13	Using Fonts	549
55.14	Frames Per Second	557
55.15	Make a Cube using RingOpenGL and RingFreeGLUT	566
56	Using RingOpenGL and RingAllegro for 3D Graphics	570
56.1	3D Cube and Texture	570
56.2	Many Cubes	574
56.3	TicTacToe 3D Game	580
57	Desktop and Mobile development using RingQt	590
57.1	The First GUI Application	590
57.2	Using Layout	592
57.3	Using the QTextEdit Class	593
57.4	Using the QListWidget Class	594
57.5	Using QTreeView and QFileSystemModel	597
57.6	Using QTreeWidget and QTreeWidgetItem	598
57.7	Using QComboBox Class	599
57.8	Creating Menubar	600
57.9	Context Menu	603
57.10	Creating Toolbar	604
57.11	Creating StatusBar	605
57.12	Using QDockWidget	606
57.13	Using QTabWidget	607

57.14 Using QTableWidget	609
57.15 Using QProgressBar	610
57.16 Using QSpinBox	611
57.17 Using QSlider	612
57.18 Using QDateEdit	613
57.19 Using QDial	614
57.20 Using QWebView	617
57.21 Using QCheckBox	618
57.22 Using QRadioButton and QButtonGroup	619
57.23 Adding Hyperlink to QLabel	621
57.24 QVideoWidget and QMediaPlayer	622
57.25 Using QFrame	624
57.26 Display Image using QLabel	625
57.27 Menubar and StyleSheet Example	626
57.28 QLineEdit Events and QMessageBox	628
57.29 Other Widgets Events	630
57.30 Using the QTimer Class	632
57.31 Using QProgressBar and Timer	633
57.32 Display Scaled Image using QLabel	634
57.33 Using the QFileDialog Class	635
57.34 Drawing using QPainter	636
57.35 Printing using QPrinter	638
57.36 Creating More than one Window	639
57.37 Playing Sound	640
57.38 Using the QColorDialog Class	640
57.39 Using qLCDNumber Class	642
57.40 Movable Label Example	642
57.41 QMessageBox Example	643
57.42 Using QInputDialog Class	644
57.43 Dialog Functions	646
57.44 KeyPress and Mouse Move Events	647
57.45 Moving Objects using the Mouse	648
57.46 Inheritance from GUI Classes	652
57.47 Using QDesktopWidget Class	653
57.48 Rotate Text	654
57.49 Change Focus	656
57.50 Regular Expressions	657
57.51 Simple Client and Server Example	658
57.52 Dynamic Objects	660
57.53 Weight History Application	661
57.54 Notepad Application	665
57.55 The Cards Game	680
57.56 Classes and their Methods to use the default events	686
57.57 Methods to use Events with Events Filter	689
57.58 The Difference between Qt and RingQt	690
57.59 RingQt Classes and their Qt Documentation	691
57.60 New Classes names - Index Start from 1	691
57.61 Creating Reports using the WebLib and the GUILib	692
58 Building RingQt Applications for Mobile	695
58.1 Download Requirements	695
58.2 Update the Android SDK	695
58.3 Install Qt for Android	696
58.4 Comments about developing for Android using RingQt	697

58.5	Using Ring2EXE	698
59	Objects Library for RingQt Application	699
59.1	Library Usage	699
59.2	Example	699
59.3	Open_WindowAndLink() Function	702
59.4	Open_WindowInPackages() Function	703
59.5	Objects Library Source Code	703
60	Using the Form Designer	704
60.1	The Designer Windows	705
60.2	The Toolbox	705
60.3	The Properties	705
60.4	Running Forms	706
60.5	Events Code	706
60.6	Keyboard Shortcuts	709
60.7	Menubar Designer	709
60.8	Window Flags	710
60.9	Entering Items	711
60.10	Using Layouts	711
60.11	More Samples and Tests	711
61	Scope Rules for Variables and Attributes	712
61.1	Three Scopes	712
61.2	Defining Variables and Variables Access	712
61.3	How Ring find the variable?	713
61.4	Using Object.Attribute	713
61.5	The Self Object	713
61.6	How Ring Define Variables and Attributes	714
61.7	Conflict between Global Variables and Class Attributes	714
61.8	Conflict between Class Attributes and Local Variables	715
61.9	Using Braces to access objects inside Class Methods	716
61.10	Accessing the class attributes from braces inside class methods	719
61.11	Creating a Class for each Window in GUI applications	720
61.12	Conflict between self inside braces and self in the class region	721
61.13	Using braces to escape from the current object scope	724
61.14	Summary of Scope Rules	725
62	Scope Rules for Functions and Methods	727
62.1	How Ring find the Functions and Methods?	727
62.2	Example about Sharing Names between Functions and Methods	728
62.3	Calling a function sharing the name with a method in the current class	729
63	Syntax Flexibility	731
63.1	Change Language Keywords	731
63.2	Change Language Operators	732
63.3	Load Syntax Files	733
63.4	Using “()” around the function parameters	734
63.5	Using Semi-colon after and between statements	734
63.6	Using \$ and @ in the start of the variable name	735
63.7	Using the ‘elseif’ keyword as ‘but’ in if statement	735
63.8	Using the ‘else’ keyword as ‘other’ in switch statement	736
63.9	Using the ‘end’ keyword in different control structures	736
63.10	Using braces to start and end different control structures	737
63.11	Using ‘put’ and ‘get’ as ‘see’ and ‘give’	738

63.12	Using ‘case’ as ‘on’ in switch statements	738
63.13	Using ‘def’ as ‘func’ in functions/methods definition	739
63.14	Using braces { } in Packages/Classes/Functions	739
63.15	Using ‘end’ keyword after Packages/Classes/Functions	739
63.16	Using ‘endpackage’/‘endclass’/‘endfunc’ keywords after Packages/Classes/Functions	740
64	Introduction to the Type Hints Library	741
64.1	Why Type Hints?	741
64.2	Example	741
64.3	User Types	741
64.4	Using Types inside Code	742
64.5	Rules	742
65	Command Line Options	744
65.1	Printing Tokens	744
65.2	Printing Rules	746
65.3	Printing Intermediate Code	750
65.4	Printing Final Intermediate Code	753
65.5	CGI Support	757
65.6	No Run	757
65.7	Printing Instruction Operation Code	757
65.8	Performance	757
65.9	Generate Object File	758
66	Distributing Ring Applications	759
66.1	Distributing Applications for Microsoft Windows	759
66.2	Protecting the Source Code	759
66.3	Creating Windows Installer	760
66.4	Using C/C++ Compiler and Linker	760
66.5	Distributing Applications and Games for Mobile	760
67	Distributing Ring Applications using Ring2EXE	761
67.1	Using Ring2EXE	761
67.2	How Ring2EXE works?	761
67.3	Example	762
67.4	Options	762
67.5	Building standalone console application	763
67.6	Distributing RingAllegro Applications	763
67.7	Distributing RingQt Applications	764
67.8	Distributing Applications for Mobile using RingQt	764
67.9	Building the Cards Game for Mobile using RingQt	765
67.10	Building the Weight History Application for Mobile using RingQt	765
67.11	Building the Form Designer for Mobile using RingQt	766
67.12	Creating the Qt resource file using Folder2qrc	768
67.13	Important Information about Ring2EXE	768
68	Low Level Functions	770
68.1	callgc() function	771
68.2	varptr() function	771
68.3	space() function	771
68.4	nullpointer() function	772
68.5	object2pointer() function	772
68.6	pointer2object() function	772
68.7	ptrcmp() function	773
68.8	ringvm_cfunctionslist() function	774

68.9	ringym_functionslist() function	774
68.10	ringym_classeslist() function	775
68.11	ringym_packageslist() function	775
68.12	ringym_memorylist() function	776
68.13	ringym_calllist() function	778
68.14	ringym_fileslist() function	779
68.15	ringym_settrace()	780
68.16	ringym_tracedata()	780
68.17	ringym_traceevent()	780
68.18	ringym_tracefunc()	781
68.19	ringym_scopescount()	781
68.20	ringym_evalinscope()	781
68.21	ringym_passerror()	781
68.22	ringym_hideerrormsg()	781
68.23	ringym_callfunc()	782
68.24	Example - Using the Trace Functions	782
68.25	Example - The Trace Library	786
69	The Trace Library and the Interactive Debugger	790
69.1	Loading the Trace library	790
69.2	Trace All Events	790
69.3	Trace control flow between functions	790
69.4	Pass Error	791
69.5	Interactive Debugger	791
69.6	Execute Program Line by Line	791
69.7	BreakPoint	792
69.8	Disable BreakPoints	792
69.9	Using the Interactive Debugger	792
70	Embedding Ring in Ring	795
70.1	Embedding Ring in Ring without sharing the State	795
70.2	Serial Execution of Programs	796
70.3	ring_state_setvar()	796
71	Extension using the C/C++ languages	798
71.1	ring_ext.h	798
71.2	ring_ext.c	798
71.3	Module Organization	799
71.4	Function Structure	800
71.5	Check Parameters Count	800
71.6	Display Error Message	801
71.7	Check Parameters Type	801
71.8	Get Parameters Values	801
71.9	Return Value	802
71.10	Function Prototype	802
71.11	Sin() Function Implementation	802
71.12	Fopen() and Fclose() Functions Implementation	802
71.13	Ring API - List Functions	803
71.14	Ring API - String Functions	805
71.15	MySQL_Columns() Function Implementation	805
71.16	Dynamic/Shared Libraries (DLL/So/Dylib) and LoadLib() function	806
72	Embedding Ring Language in C/C++ Programs	808
72.1	Ring State	808
72.2	Ring State Functions	808

72.3	Ring State Variables	809
73	Code Generator for wrapping C/C++ Libraries	811
73.1	Using the tool	811
73.2	Configuration file	811
73.3	Using the function prototype	811
73.4	Adding code to the generated code	813
73.5	Prefix for Functions Names	813
73.6	Generate function to wrap structures	813
73.7	Determine Structure Members Types	814
73.8	Defining Constants	814
73.9	Register New Functions	814
73.10	Writing comments in the configuration file	815
73.11	Executing code during code generation	815
73.12	Enum and Numbers	815
73.13	Filtering using Expressions	816
73.14	Constants Type	816
73.15	Configuration file for the Allegro Library	816
73.16	Threads Support	818
73.17	Code Generator Rules for Wrapping C++ Classes	819
73.18	Using configuration file that wrap C++ Library	820
73.19	Configuration file for the Qt Framework	820
73.20	Configuration Files Examples	830
74	RingLibCurl Functions Reference	831
75	RingLibZip Functions Reference	840
76	RingConsoleColors Functions Reference	841
77	RingMurmurHash Functions Reference	843
77.1	MurmurHash1 functions	843
77.2	MurmurHash2 functions	843
77.3	MurmurHash3 functions	843
77.4	Example	844
78	RingAllegro Functions Reference	845
79	RingLibSDL Functions Reference	864
80	RingFreeGLUT Functions Reference	880
81	RingOpenGL (OpenGL 1.1) Functions Reference	889
82	RingOpenGL (OpenGL 1.2) Functions Reference	914
83	RingOpenGL (OpenGL 1.3) Functions Reference	940
84	RingOpenGL (OpenGL 1.4) Functions Reference	969
85	RingOpenGL (OpenGL 1.5) Functions Reference	999
86	RingOpenGL (OpenGL 2.0) Functions Reference	1030
87	RingOpenGL (OpenGL 2.1) Functions Reference	1064
88	RingOpenGL (OpenGL 3.0) Functions Reference	1106

89 RingOpenGL (OpenGL 3.1) Functions Reference	1151
90 RingOpenGL (OpenGL 3.2) Functions Reference	1197
91 RingOpenGL (OpenGL 3.3) Functions Reference	1244
92 RingOpenGL (OpenGL 4.0) Functions Reference	1291
93 RingOpenGL (OpenGL 4.1) Functions Reference	1338
94 RingOpenGL (OpenGL 4.2) Functions Reference	1385
95 RingOpenGL (OpenGL 4.3) Functions Reference	1432
96 RingOpenGL (OpenGL 4.4) Functions Reference	1479
97 RingOpenGL (OpenGL 4.5) Functions Reference	1526
98 RingOpenGL (OpenGL 4.6) Functions Reference	1573
99 RingQt Classes Reference	1621
99.1 QApp Class	1621
99.2 QDesktopServices Class	1621
99.3 QTest Class	1621
99.4 QObject Class	1622
99.5 QWidget Class	1622
99.6 QLabel Class	1628
99.7 QPushButton Class	1629
99.8 QPixmap Class	1629
99.9 QPixmap Class	1630
99.10 QPixmap2 Class	1631
99.11 QIcon Class	1631
99.12 QSize Class	1631
99.13 QLineEdit Class	1631
99.14 QVBoxLayout Class	1633
99.15 QHBoxLayout Class	1634
99.16 QTextEdit Class	1634
99.17 QListWidget Class	1637
99.18 QTreeView Class	1638
99.19 QDir Class	1640
99.20 QFileSystemModel Class	1640
99.21 QTreeWidget Class	1642
99.22 QTreeWidgetItem Class	1644
99.23 QComboBox Class	1645
99.24 QMenuBar Class	1647
99.25 QMenu Class	1648
99.26 QToolBar Class	1649
99.27 QMainWindow Class	1649
99.28 QStatusBar Class	1651
99.29 QDockWidget Class	1651
99.30 QTabWidget Class	1652
99.31 QTableWidgetItem Class	1653
99.32 QFrame Class	1655
99.33 QFrame2 Class	1655
99.34 QFrame3 Class	1655
99.35 QAbstractScrollArea Class	1655

99.36 QAbstractItemView Class	1656
99.37 QTableView Class	1658
99.38 QTableWidget Class	1659
99.39 QProgressBar Class	1661
99.40 QSpinBox Class	1662
99.41 QAbstractSlider Class	1663
99.42 QSlider Class	1664
99.43 QDateEdit Class	1664
99.44 QDateTimeEdit Class	1664
99.45 QAbstractSpinBox Class	1666
99.46 QDial Class	1667
99.47 QWebView Class	1667
99.48 QUrl Class	1668
99.49 QCheckBox Class	1669
99.50 QAbstractButton Class	1670
99.51 QRadioButton Class	1671
99.52 QButtonGroup Class	1671
99.53 QMediaPlayer Class	1672
99.54 QMediaPlaylist Class	1673
99.55 QVideoWidget Class	1673
99.56 QAction Class	1674
99.57 QEvent Class	1676
99.58 QMessageBox Class	1676
99.59 QTimer Class	1677
99.60 QFileDialog Class	1677
99.61 QPainter Class	1679
99.62 QPainter2 Class	1681
99.63 QPicture Class	1681
99.64 QPen Class	1682
99.65 QColor Class	1682
99.66 QPrinter Class	1685
99.67 QFont Class	1686
99.68 QBrush Class	1688
99.69 QByteArray Class	1689
99.70 QIODevice Class	1690
99.71 QAbstractSocket Class	1691
99.72 QNetworkProxy Class	1693
99.73 QTcpSocket Class	1694
99.74 QTcpServer Class	1694
99.75 QHostAddress Class	1695
99.76 QHostInfo Class	1695
99.77 QFileInfo Class	1696
99.78 QDirModel Class	1697
99.79 QFontDialog Class	1698
99.80 QDialog Class	1698
99.81 QTextCursor Class	1699
99.82 QColorDialog Class	1701
99.83 QStringList Class	1701
99.84 QKeySequence Class	1702
99.85 QLCDNumber Class	1703
99.86 QInputDialog Class	1703
99.87 QAllEvents Class	1705
99.88 QDesktopWidget Class	1708
99.89 QRect Class	1709

99.90 QTextDocument Class	1710
99.91 QTextBlock Class	1712
99.92 QTime Class	1713
99.93 QListWidgetItem Class	1714
99.94 QSystemTrayIcon Class	1715
99.95 QDate Class	1715
99.96 QTextCodec Class	1716
99.97 QSqlDatabase Class	1716
99.98 QSqlDriver Class	1717
99.99 QSqlQuery Class	1718
99.100QSqlError Class	1719
99.101QSqlIndex Class	1719
99.102QSqlRecord Class	1720
99.103QSqlField Class	1720
99.104QSqlDriverCreatorBase Class	1721
99.105QVariant Class	1721
99.106QNetworkRequest Class	1722
99.107QNetworkAccessManager Class	1723
99.108QNetworkReply Class	1724
99.109QPainterPath Class	1724
99.110QImage Class	1725
99.111IRingCodeHighlighter Class	1727
99.112QXmlStreamReader Class	1727
99.113QXmlStreamWriter Class	1729
99.114QXmlStreamNotationDeclaration Class	1730
99.115QXmlStreamNamespaceDeclaration Class	1730
99.116QXmlStreamEntityResolver Class	1730
99.117QXmlStreamEntityDeclaration Class	1730
99.118QXmlStreamAttributes Class	1731
99.119QXmlStreamAttribute Class	1731
99.120QThread Class	1731
99.121QThreadPool Class	1732
99.122QRegularExpression Class	1733
99.123QRegularExpressionMatch Class	1733
99.124QRegularExpressionMatchIterator Class	1734
99.125QJsonArray Class	1734
99.126QJsonDocument Class	1735
99.127QJsonObject Class	1736
99.128QJsonParseError Class	1736
99.129QJsonValue Class	1736
99.130QPlainTextEdit Class	1737
99.131CodeEditor Class	1739
99.132QGridLayout Class	1740
99.133QTextCharFormat Class	1741
99.134QCameraViewfinder Class	1742
99.135QGraphicsVideoItem Class	1742
99.136QVideoWidgetControl Class	1743
99.137QCamera Class	1743
99.138QCameraImageCapture Class	1744
99.139QMediaObject Class	1744
99.140QHeaderView Class	1745
99.141QFontMetrics Class	1747
99.142QSplashScreen Class	1748
99.143QBoxLayout Class	1748

99.144QLayout Class	1749
99.145QLinearGradient Class	1750
99.146QGradient Class	1750
99.147QPointF Class	1750
99.148QPoint Class	1751
99.149QScrollArea Class	1751
99.150QSplitter Class	1752
99.151QCompleter Class	1752
99.152QCompleter2 Class	1753
99.153QCompleter3 Class	1754
99.154QString2 Class	1754
99.155QProcess Class	1754
99.156QMdiArea Class	1755
99.157QMdiSubWindow Class	1756
99.158QCursor Class	1757
99.159QListView Class	1757
99.160QAxObject Class	1758
99.161QAxBase Class	1758
99.162QUuid Class	1759
99.163QToolButton Class	1759
99.164QSerialPort Class	1760
99.165QSerialPortInfo Class	1761
99.166QStringRef Class	1761
99.167QMutex Class	1763
99.168QMutexLocker Class	1763
99.169QBuffer Class	1763
99.170QBluetoothAddress Class	1764
99.171QBluetoothDeviceDiscoveryAgent Class	1764
99.172QBluetoothDeviceInfo Class	1764
99.173QBluetoothHostInfo Class	1765
99.174QBluetoothLocalDevice Class	1765
99.175QDateTime Class	1765
99.176QScreen Class	1766
99.177QWindow Class	1767
99.178QGuiApplication Class	1771
99.179QCoreApplication Class	1773
99.180QTextBrowser Class	1774
99.181QRegion Class	1775
100Frequently Asked Questions (FAQ)	1777
100.1 Why do we need Yet Another Programming Language (YAPL)?	1777
100.2 Why Ring is weakly typed?	1778
100.3 What are the advantages to using Ring over Lisp or Smalltalk?	1778
100.4 Why Ring is largely focussed on UI creation?	1779
100.5 Is Ring some sort of improvement over PHP?	1779
100.6 What are the advantages to using Ring over native C or C++?	1779
100.7 What is the difference between Ring and Python? And is Ring Open Source?	1779
100.8 What are the advantages to using Ring over Perl, PHP, Python or Ruby?	1780
100.9 What are the advantages to using Ring over Tcl or Lua?	1781
100.10What are the advantages to using Ring over C# or Java?	1781
100.11The documentation says functional programming is supported, but then this happens?	1781
100.12Why the ability to define your own languages Instead of just handing over the syntax so you can parse it using whatever code you like?	1782
100.13Why you can specify the number of loops you want to break out of?	1782

100.14	Why Ring uses ‘See’, ‘Give’, ‘But’ and ‘Ok’ Keywords?	1782
100.15	What is the philosophy behind data types in Ring?	1782
100.16	What about the Boolean values in Ring?	1783
100.17	What is the goal of including the “Main” function in Ring?	1784
100.18	Why the list index start from 1 in Ring?	1785
100.19	Why Ring is not case-sensitive?	1785
100.20	Why the Assignment operator uses Deep Copy?	1786
100.21	Is there constructor methods in Ring?	1787
100.22	What happens when we create a new object?	1787
100.23	Can we use the attributes by accessing the Getter and Setter methods?	1788
100.24	Why should a search of global names be made while defining the class attributes?	1789
100.25	Why Ring doesn’t avoid the conflict between Global Variables and Class Attributes Names?	1790
100.26	Where can I write a program and execute it?	1791
100.27	How to get the file size using ftell() and fseek() functions?	1791
100.28	How to get the current source file path?	1791
100.29	What about predefined parameters or optional parameters in functions?	1791
100.30	How to print keys or values only in List/Dictionary?	1792
100.31	Why I get a strange result when printing nl with lists?	1792
100.32	Could you explain the output of the StrCmp() function?	1793
100.33	How to use many source code files in the project?	1793
100.34	Why this example use the GetChar() twice?	1794
100.35	How to use NULL and ISNULL() function?	1795
100.36	How to print lists that contains objects?	1796
100.37	How to insert an item to the first position in the list?	1797
100.38	How to print new lines and other characters?	1797
100.39	Why we don’t use () after the qApp class name?	1797
100.40	Why the window title bar is going outside the screen?	1798
100.41	How to create an array of buttons in GUI applications?	1798
100.42	How to Close a window then displaying another one?	1799
100.43	How to create a Modal Window?	1799
100.44	How can I disable maximize button and resize window?	1800
100.45	How to use SQLite using ODBC?	1801
100.46	Can I connect to dbase/harbour database?	1802
100.47	Why setClickEvent() doesn’t see the object methods directly?	1803
100.48	Why I get Calling Function without definition Error?	1803
100.49	Can Ring work on Windows XP?	1804
100.50	How to extend RingQt and add more classes?	1804
100.51	How to add Combobox and other elements to the cells of a QTableWidgetItem?	1807
100.52	How to perform some manipulations on selected cells in QTableWidgetItem?	1807
100.53	Which of 3 coding styles are commonly used or recommended by the community?	1808
101	Language Reference	1809
101.1	Language Keywords	1809
101.2	Language Functions	1811
101.3	Compiler Errors	1811
101.4	Runtime Errors	1812
101.5	Environment Errors	1813
101.6	Language Grammar	1814
101.7	Virtual Machine (VM) Instructions	1815
102	Resources	1822
102.1	Ring Language Website	1822
102.2	Ring Group	1822
102.3	Contact the Authors	1822

APPLICATIONS DEVELOPED IN LITTLE HOURS

Ring is a new programming language that focuses on the Natural Language Programming and Declarative Programming paradigms and will let you think different about programming and how to solve your problems in a better way. It's just released in 2016.01.25! In little days we got thousands of downloads and many developers started learning and using the language. Their feedback are the secret behind the language progress and success. They said that Ring is powerful, beautiful and easy to learn, Some of them provided good examples about what can be done using Ring in little hours. They are very happy with the language productivity.

1.1 Quotes about Ring

"I like Smalltalk very much but now I like Ring better!" , Gal Zsolt (Hungary)

"I find the language and its syntax very natural and easy to follow." , Bhudda (United States)

"Very nice approach for a new language." , Matth Moestl (Austria)

"Very interesting! I will keep an eye on it!" , Eslipak (Argentina)

"I'd like to see some benchmarks. Otherwise, at first glance, it looks really promising." , Alex Deva (Sweden)

"Excellent" , Liju Sankar (United States)

"I wish you the best with this project." , David O'Neil (United States)

"Just fantastic." , Jose Antonio (Mexico)

"This looks like it was developed by some very competent people." , Jim Clack (United States)

"The Ring programming language seems pretty interesting so far." , Eric Johnson (United States)

"Thank you for this awesome language and wonderful ready to use Qt binding." , Martial FAESSEL

"I think it's great what he does for the community of developers and novice programming." , Marino Esteban

"Ring is just awesome. The language is so cool and fluent. I am sure it's going to be BIG." , Ahmed Zain

"What a proud, really wish you Eng. Mahmoud Samir and Your Team moving forward ^_^"

and from now , considered me a big fan of the Ring programming language." , Zainab Mahmoud

"Well guys I love this language and it appears that you have created perfect language." , Moemen Ghulmi

"Good work Mahmoud, I've installed ring pl, and it's very perfect language." , Ahmed Omar (Egypt)

"Thanks for this great startup programming language. I wish you best of luck." , Elkhoulaja Khalid

"Congratulations! I am very happy and I wish you Success and good luck." , Abobasmla Hassan

"Good Features of multi-use language on the Web, Mobile and Desktop." , Abdelrhman Haider

“Very interesting effort.” , Giannakakis Kostas (Greece)

“I am too lazy to open comment window and write message. But in your case I must say “Perfect” Really, create new remarkable language like your Ring is really perfect job. Even create editor for your language in your language with only few rows... Even noticed in rosettacode.org !

I read your previous articles and I tried these examples a few days ago and I will continue. I love Ring.

P.S.: Anders Hejlsberg, Niklaus Wirth, Bjarne Stroustrup, Ada Lovelace Hall of fame is waiting...”

, Martin Nedopil (Czech Republic)

“Ring seems very attractive to me through its very easy design and the Qt bindings. I like its declarative approach and the generous documentation.”

, Shalok Shalom (Austria)

“Ring (and plenty of extension library + Qt) is wonderful.” , Kovacs Attila (Hungary)

“Since two days I’m trying Ring and I’m really impressed, in add to power commands and easy use, it’s really very efficient and very fast.

Each day I hope to find the couple of the year PWCT+RING ... Maybe for my Christmas gift!!!! HO HO HO HO

Continue your fantastic job and congratulations.” , Jose Le Roux (France)

“There are 3 different styles, it looks like Python and C” , 64remy

“I was taking a tour around Rosettacode and have found Ring. I like the syntax a lot. It’s clean and easy to understand. It looks like a very clean BASIC dialect without sigils. I can say that this is the easiest and the most BASIC-like language I’ve ever tried.”

, Tomaaz

“Thanks for your effort. I took a quick look and found it interesting.

You are trying to follow more or less like Clipper with simple command and no rigid declaration rules. Good.”

, Anand Gupta

“Thanks for this wonderful language” , Vinod kc (India)

“Very enlightening. good job!” , Southmountain (United States)

“The thing I liked was the loop exiting” , Leon de boer (Australia)

“An outstanding and easy language to program with.” , Kenneth Burgo (Philippines)

“I chose your language as I feel I can understand it better than other languages” , Harry Singh

“I like the totality of the language, far more features than expected and the freedom of expressiveness is unique.”

, Evikone

“Thank you very much Mahmoud! I am using ring for many experiments and so far I love it.

I really want to continue using ring and contribute what I can.” , John (SienSystem)

“Sir, Very Good” , Kamlesh Patel

“That’s more than a “cool” syntax, the example of writing free-form text between curly-brackets such that each word calls a function. Which could be interesting (A syntax like that would be nice for declaring text styles)”

, LaurieCheers

“If you browse around you see they have listed 160 contributors. This year they have entered Top 100 in the TIOBE index. Lot of effort seem to have been made to make this language pop out and catch the attention of masses.”

, Htuhola

“I like the idea of The Ring being in ANSI C

it’s an impressive creation, and a lot of skill went into it” , Garry Taylor

“Very innovative language! Syntactically clean” , CodeProject Member

“The author must be commended for the readily-obvious hard work and effort that has gone into creating a rich ecosystem for his language. It seems that the language is quite extensive as well. I would find it useful to see a BNF grammar and concise coverage of its semantics.”

, Xx-Leninist-1917-Xx (Reddit)

“I can see the AI of the future using this technology to solve computational problems for..... the humans.” ,
Cryptonite (United States)

“I like your programming language, I like you are going to develop mobile app using RingQt and also I appreciate your web library.”

, Domenico D’Oria (Italy)

“Congratulations for the great work with this new programming language.” , Kenny Silva (Venezuela)

“Ring is an amazingly full-featured language and so well documented (the bane of most newer languages out there!)”
, Alex McCullie

” I found the language yesterday, and liked the Qt bindings, as they give a declarative way to create a QtWidgets GUI.”
, Cochise Cesar

1.2 FetchStockData Application

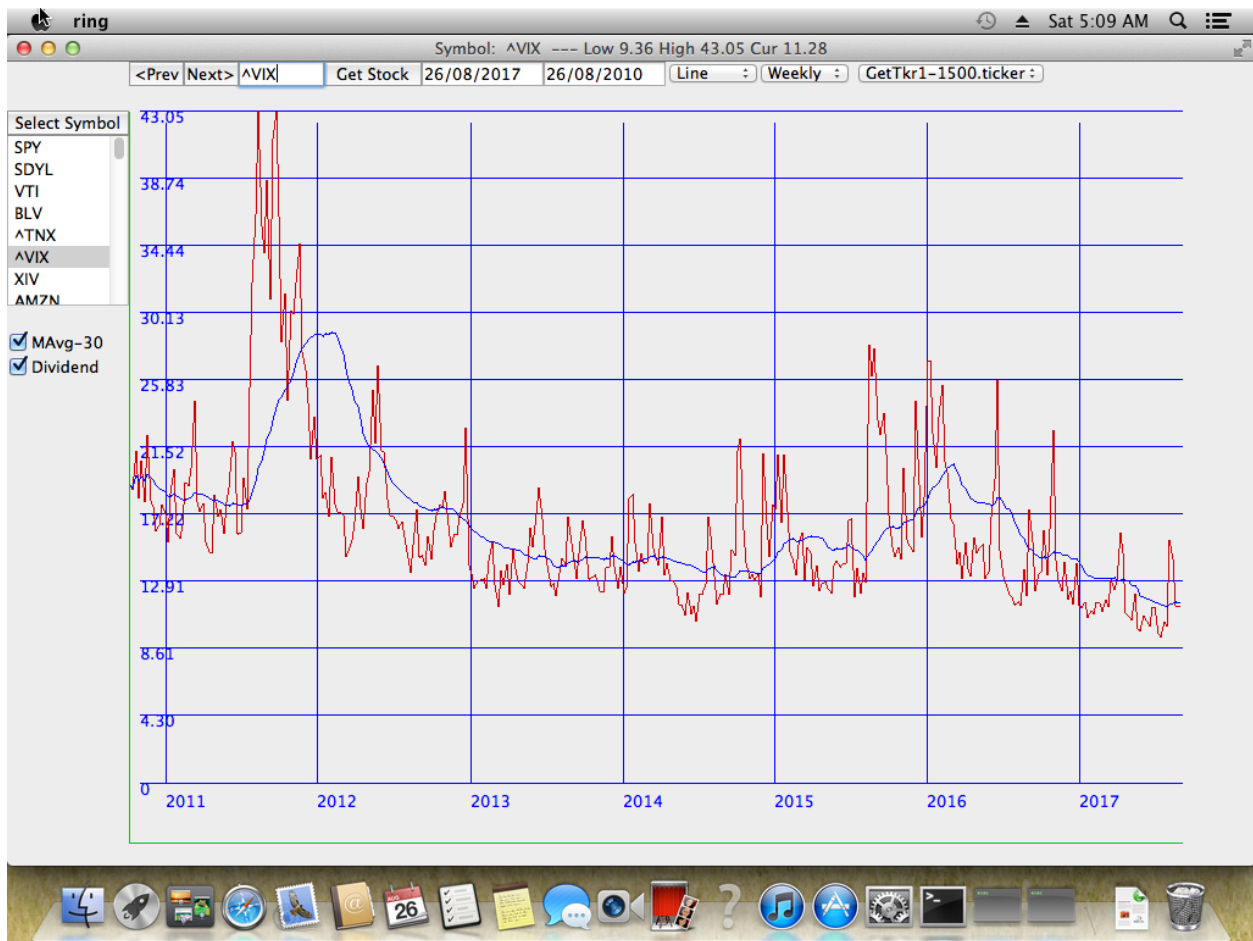
URL : https://groups.google.com/forum/#!topic/ring-lang/-fa1U_SXSjo

Author : Bert Mariani

This App is written in Ring.

It will fetch stock data from Yahoo and draw various types of charts. Any valid stock ticker can be entered, or selected from the drop down list - Select Symbol.

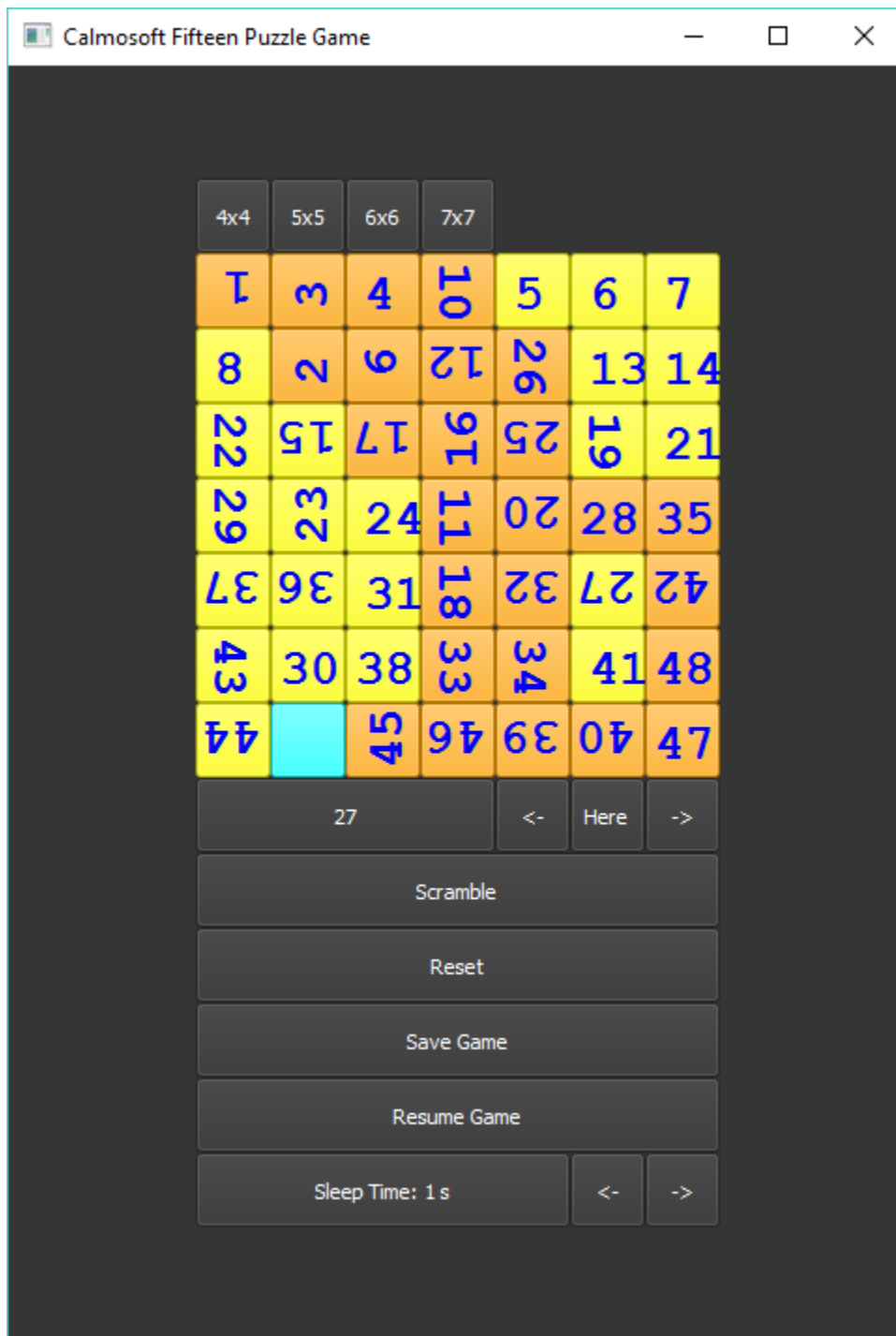
It will not guarantee that you make money in the stock market. But it will visualize the history of the stock.



1.3 Fifteen Puzzle Game 2

URL : <https://github.com/ring-lang/ring/blob/master/applications/fifteenpuzzle/CalmoSoftFifteenPuzzleGame.ring>

Author : Gal Zsolt (CalmoSoft)



1.4 Google API Shortener Application

Author : John Storm (SienSystem)

Hi everyone,

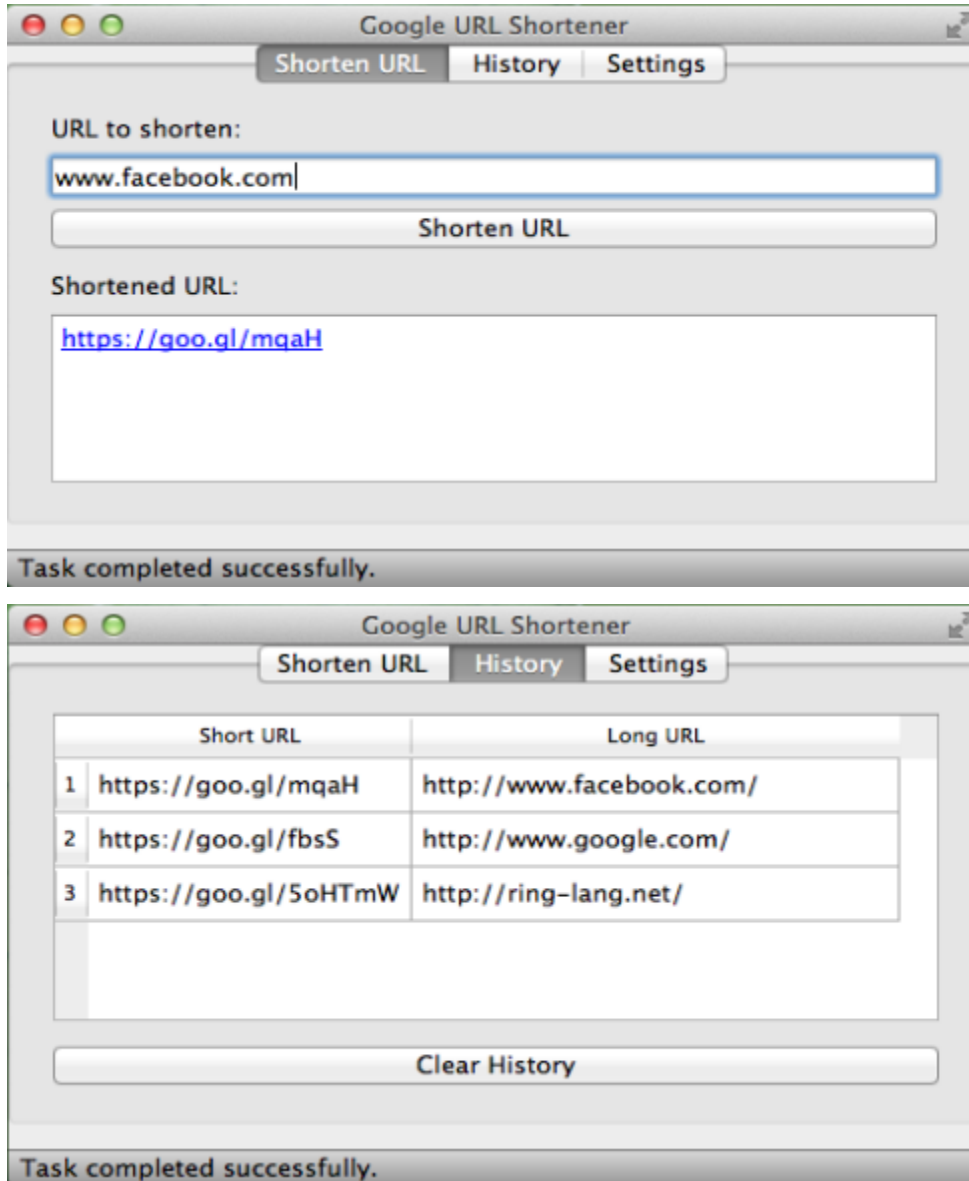
Wanted to (finally) share a first release version of a small but very nice tool to shorten urls using the Google shortener

API. The code is fully documented, maybe too documented, but, provides a good reference as to what is going on.

You can obtain your own API key, or you can use my key to test and use the tool.

Hope you enjoy the application. Please feel free to test at your convenience.

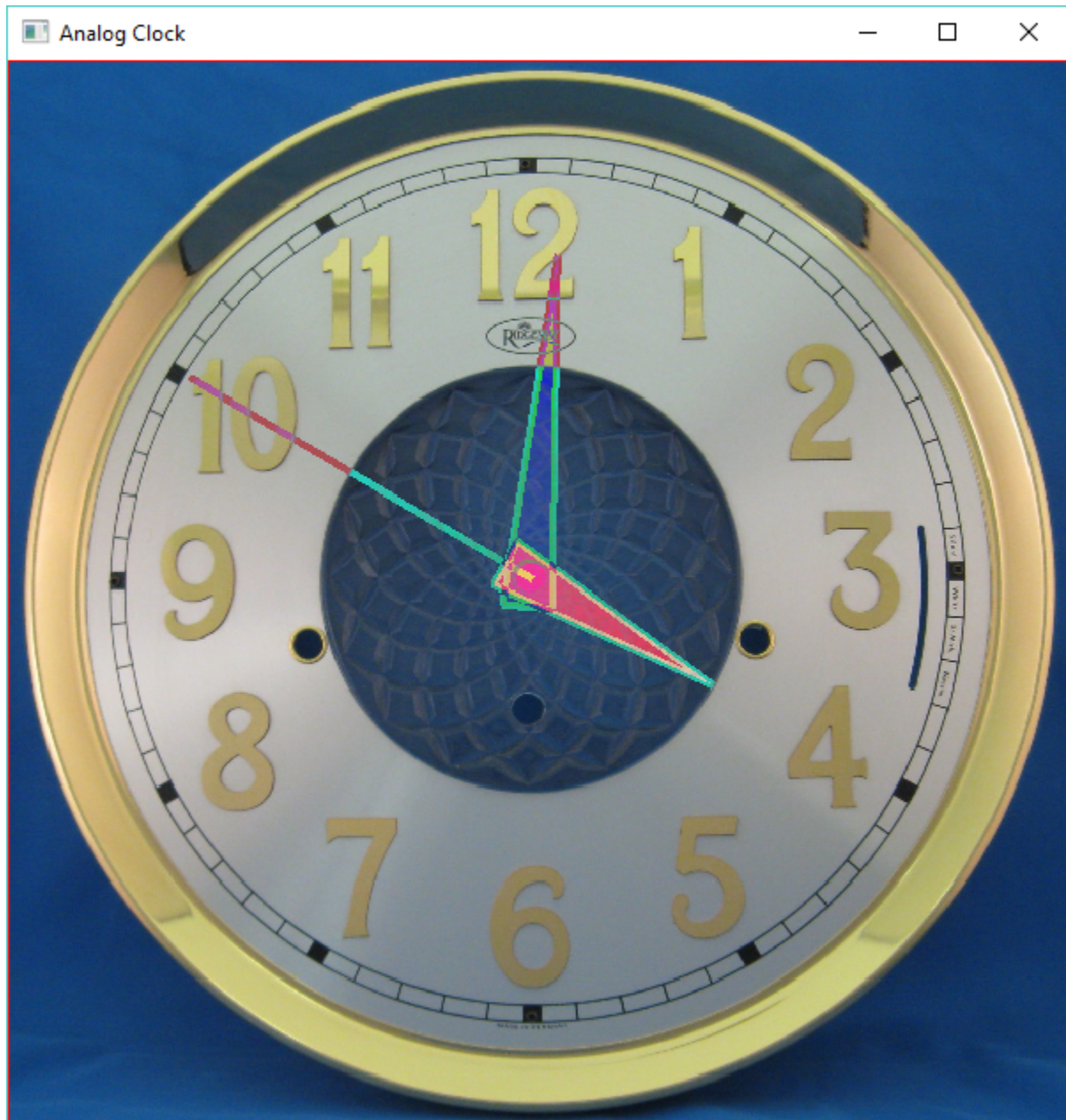
Ring is FUN!!



1.5 Analog Clock

URL : <https://github.com/ring-lang/ring/blob/master/applications/analogclock/AnalogClock-Image.ring>

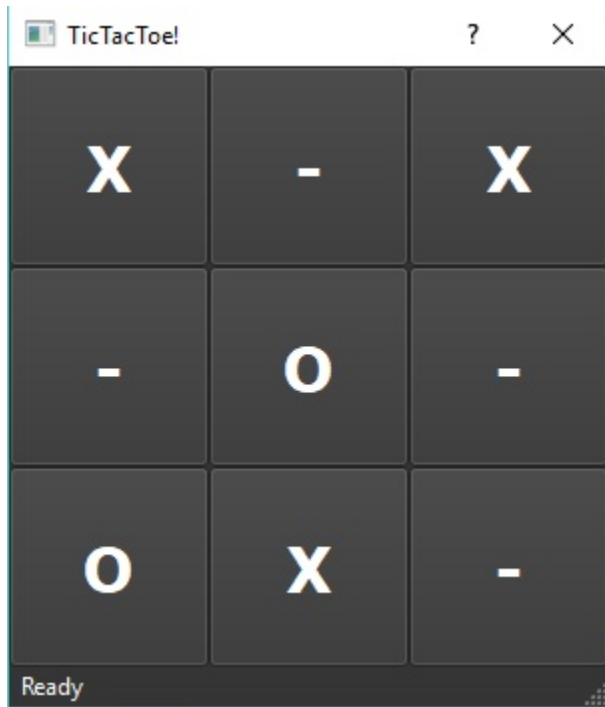
Author : Bert Mariani



1.6 TicTacToe Game

URL : <https://github.com/AbdelrahmanGIT/RingSamples/blob/master/src/TecTacToe.ring>

Author : Abdelrahman Mohammed



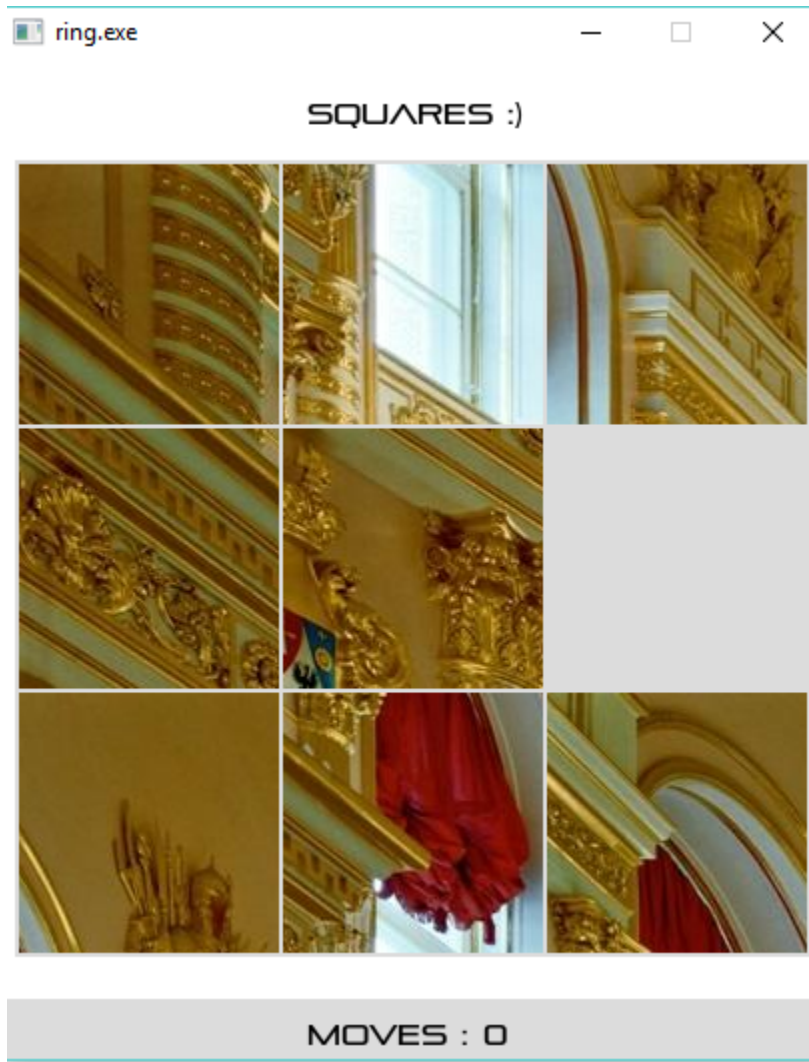
1.7 Squares Puzzle Game

URL : https://github.com/MajdiSobain/RingAllegro_SquaresPuzzle

Author : Majdi Sobain

This project is about (Squares Puzzle) popular game that I have programmed using ring language with its RingAllegro Library. The principle of this game is very known to all of us, which is moving squares to get the real full shape of the original picture.

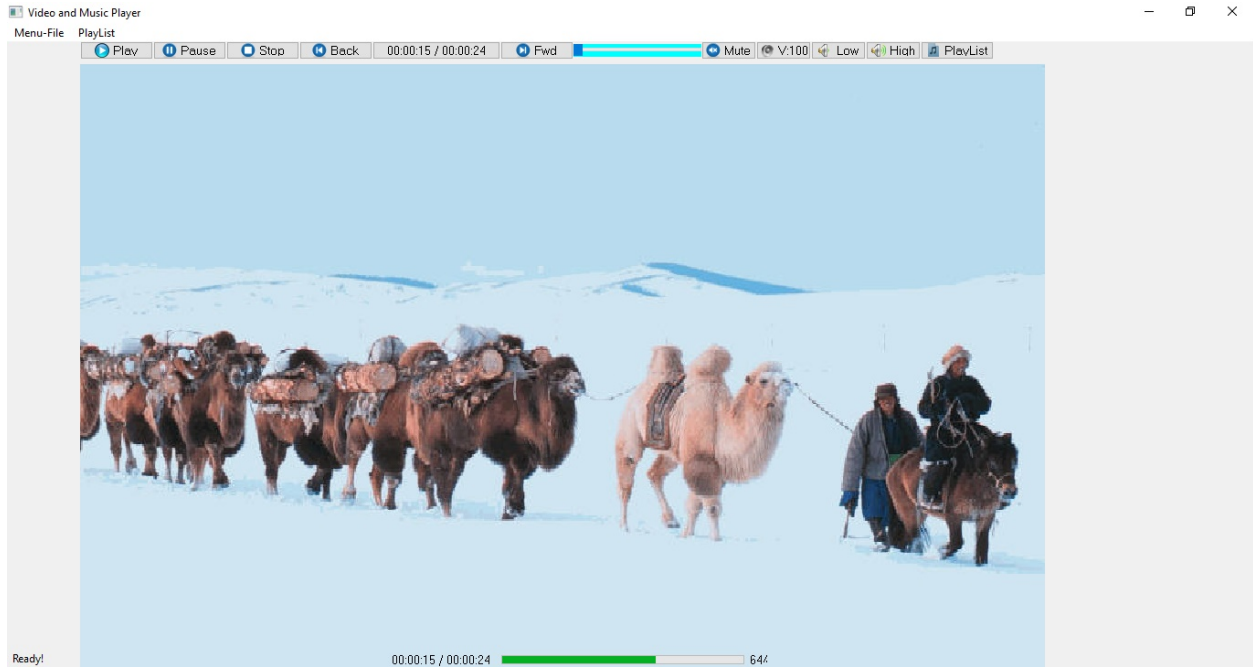
This game could be played using mouse and keyboard as well, showing a message of congratulations at the successful solving.



1.8 Video-Music-Player Application

Author : Bert Mariani

Screen Shot:

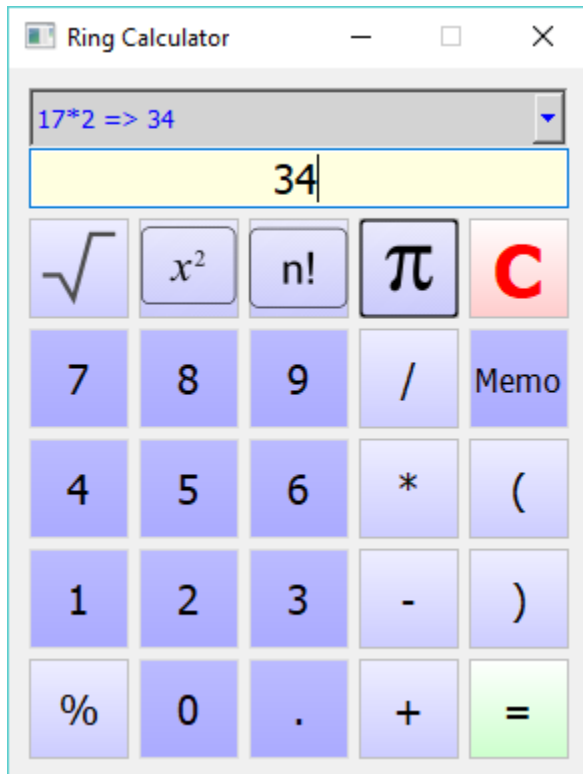


1.9 Calculator Application

Author : Magdy Ragab

Updated Version by Gal Zsolt and Bert Mariani

Screen Shot:



1.10 Windows StartUp Manager Application

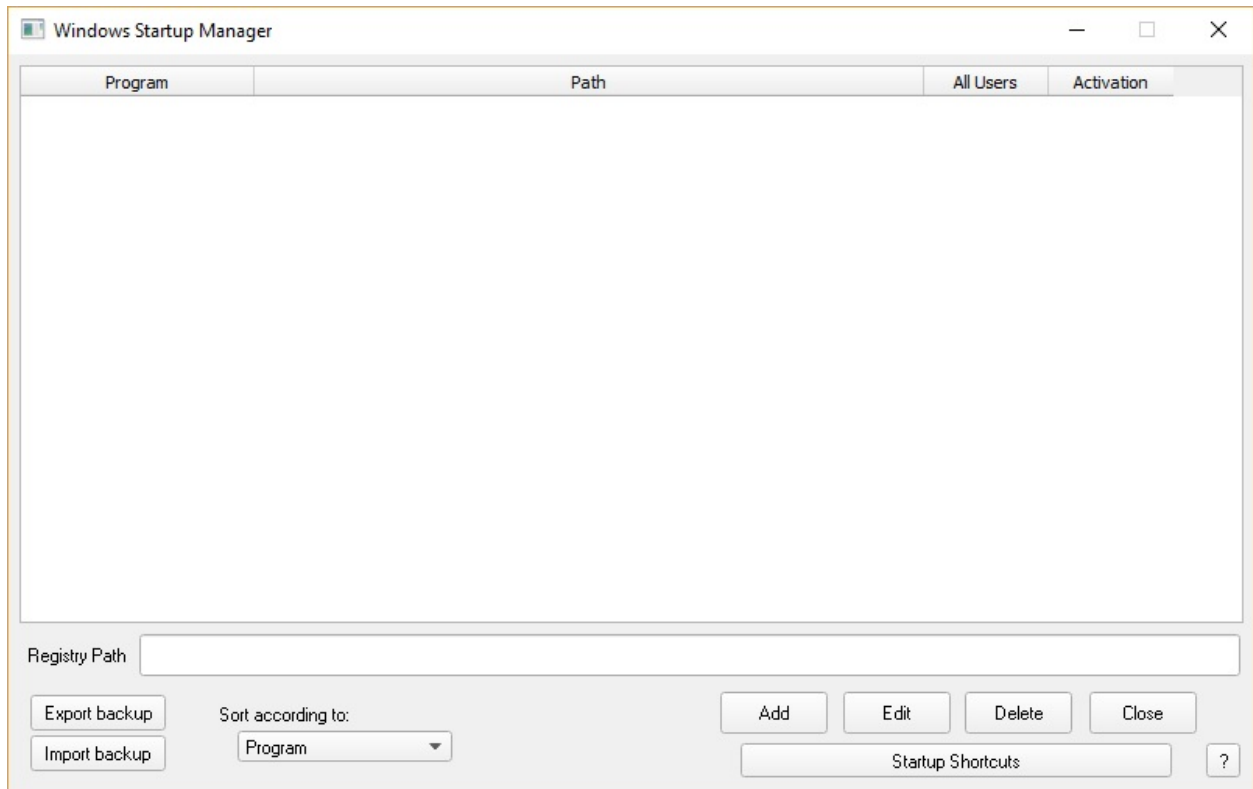
URL : <https://github.com/ring-lang/WinStartupManager>

Author : Majdi Sobain

Windows Startup Manager is an application that has the ability to let the user control what programs should start at Windows booting time. It gives you the ability to show, edit, delete, or even add new programs entries to be launched at Windows booting time.

It is specialized at managing Windows startup entries that are stored in Windows Registry only, but provides a quick option to edit programs shortcuts type entries.

Screen Shot:



1.11 Werdy Application

URL : <https://github.com/ring-lang/werdy>

Author : Magdy Ragab

Quran application includes reading suras, searching and bookmarking.

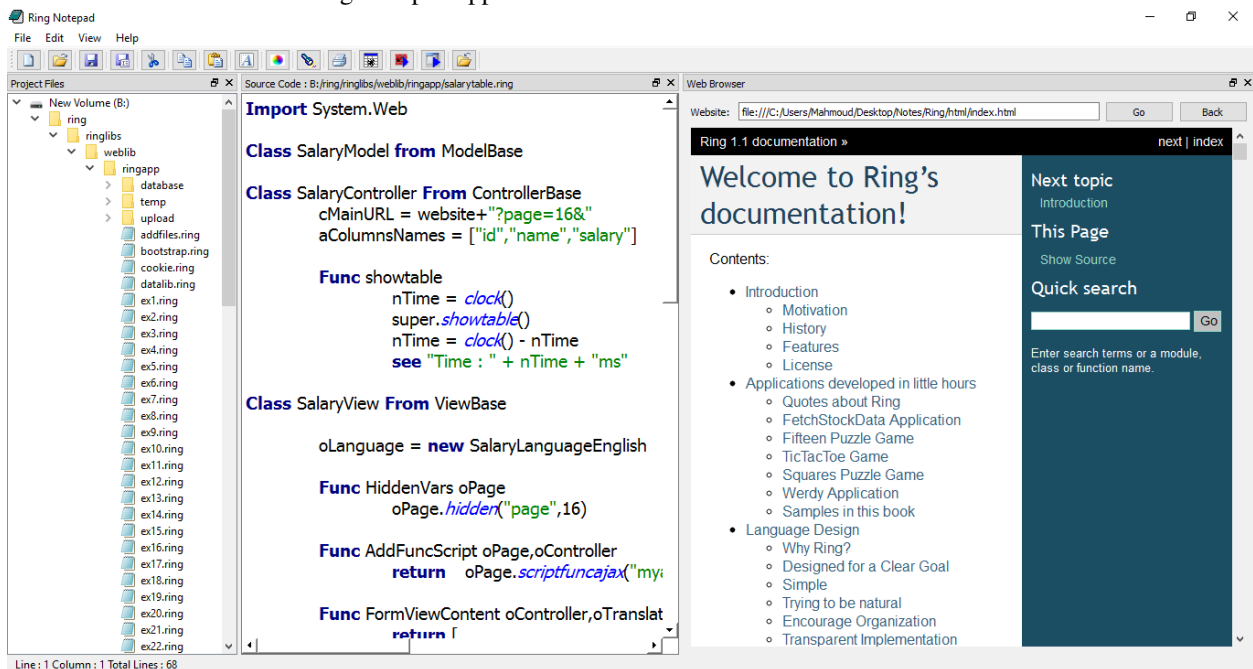
The application is provided for Windows, Linux and Android.



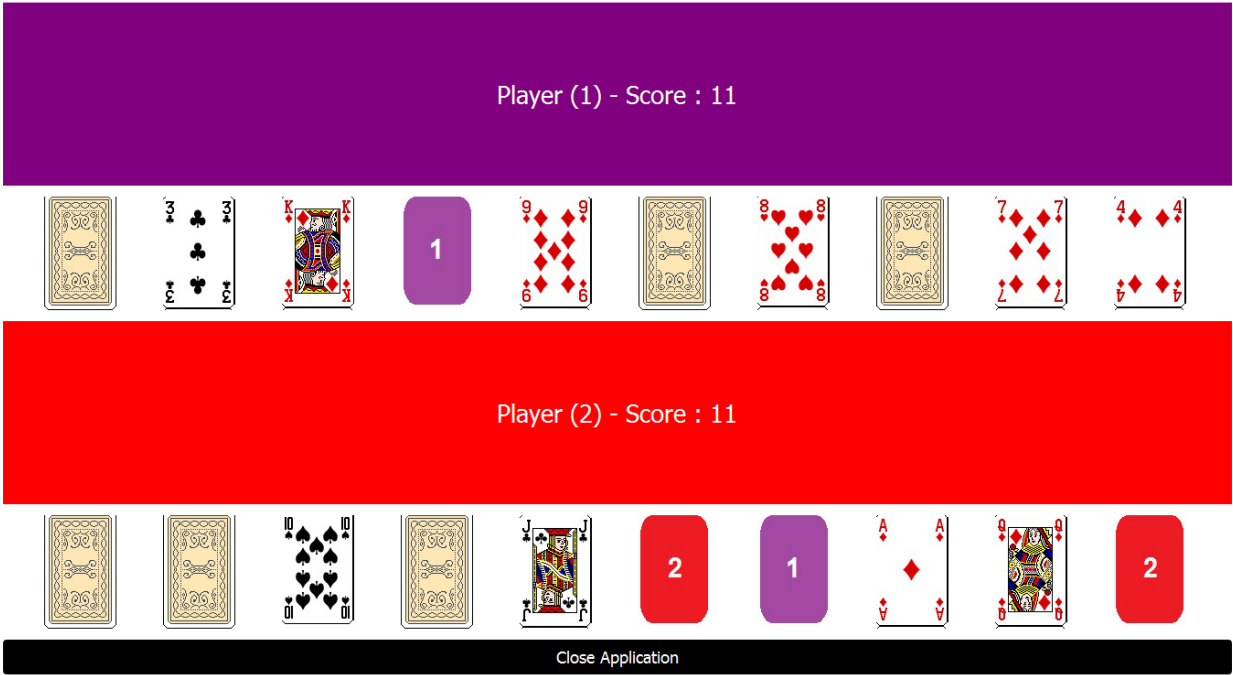
1.12 Samples in this book

The next samples are developed in little hours and we will introduce them through this book.

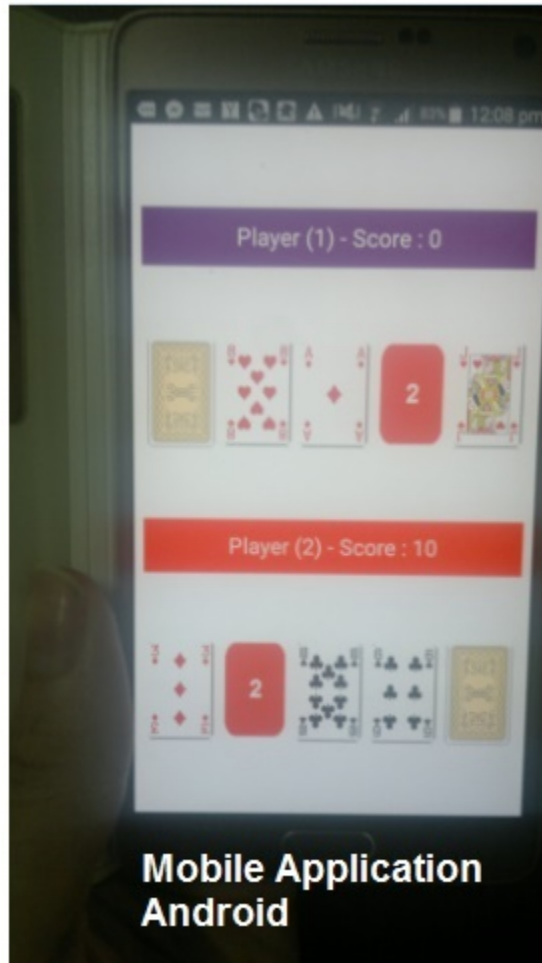
The next screen shot for the Ring Notepad application



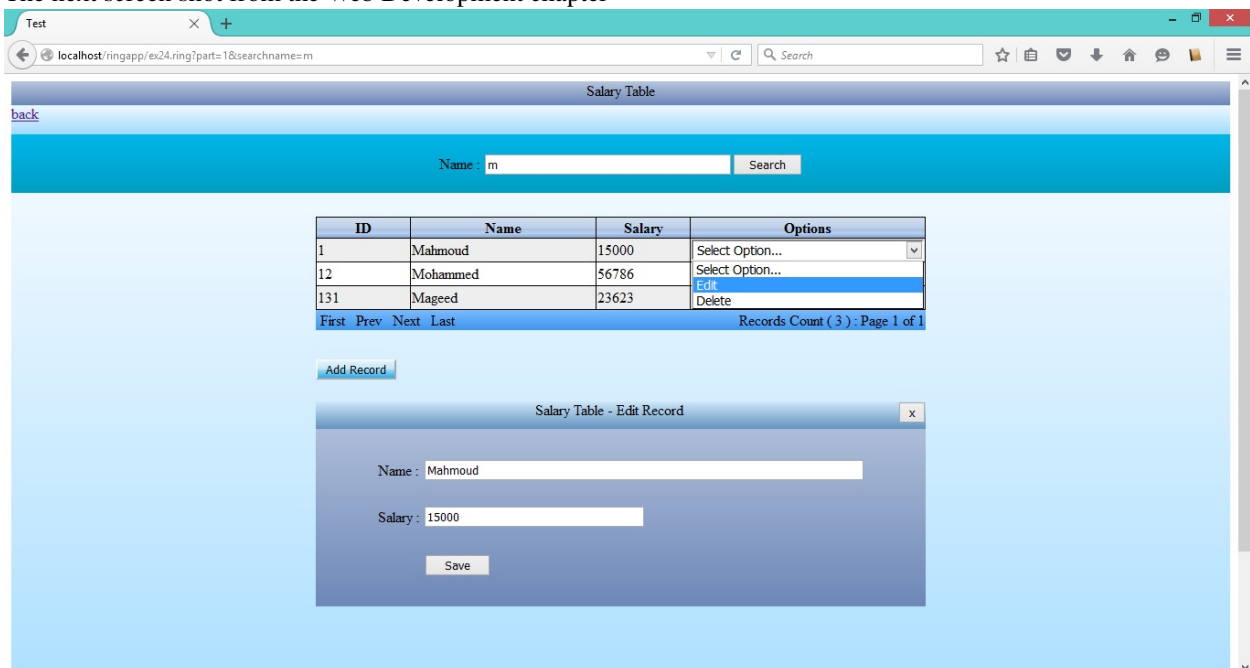
The next screen shot for the Cards Game



The next screen shot for the Cards Game (Android)



The next screen shot from the Web Development chapter



The next screen shots for simple 2D Games that we will present in the Game Engine Chapter.

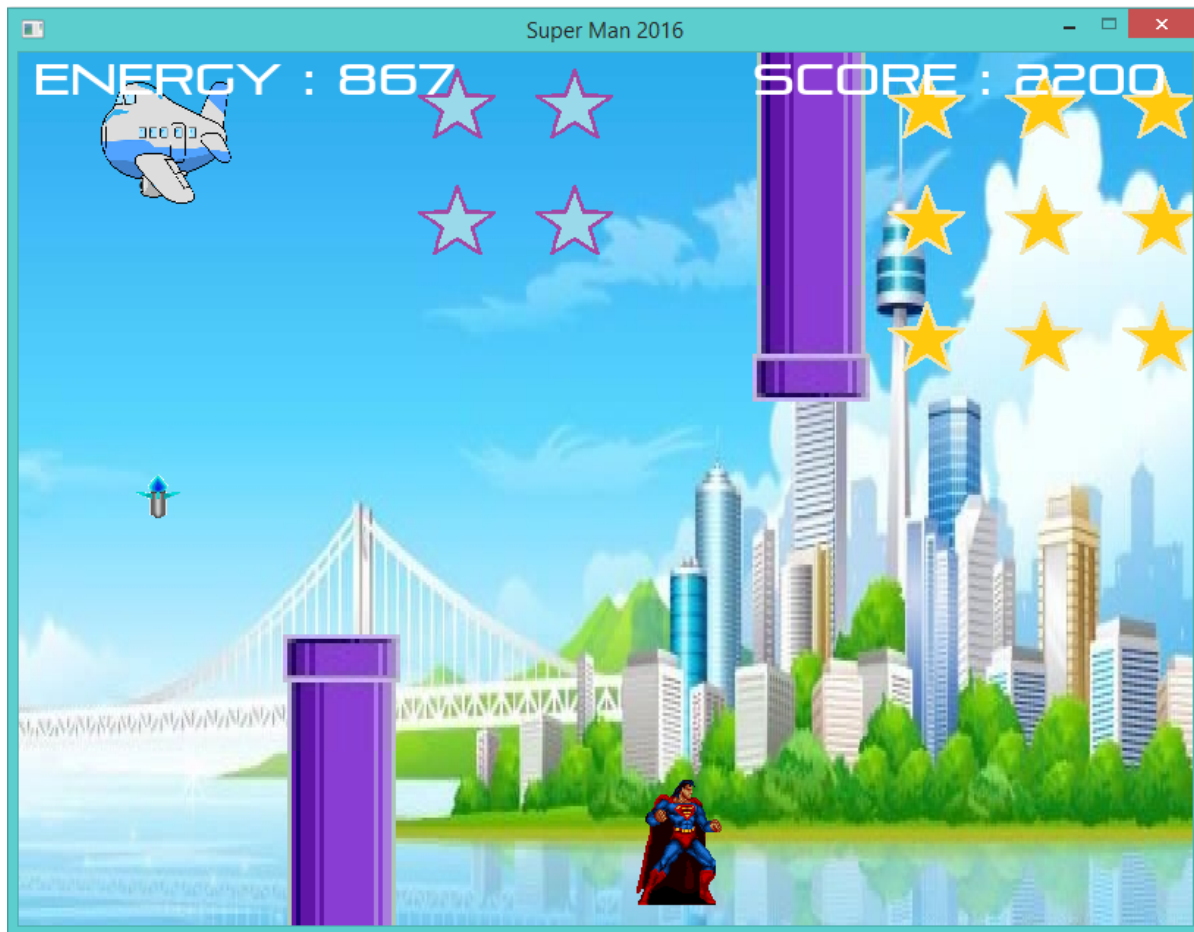
Stars Fighter Game



Flappy Bird 3000 Game

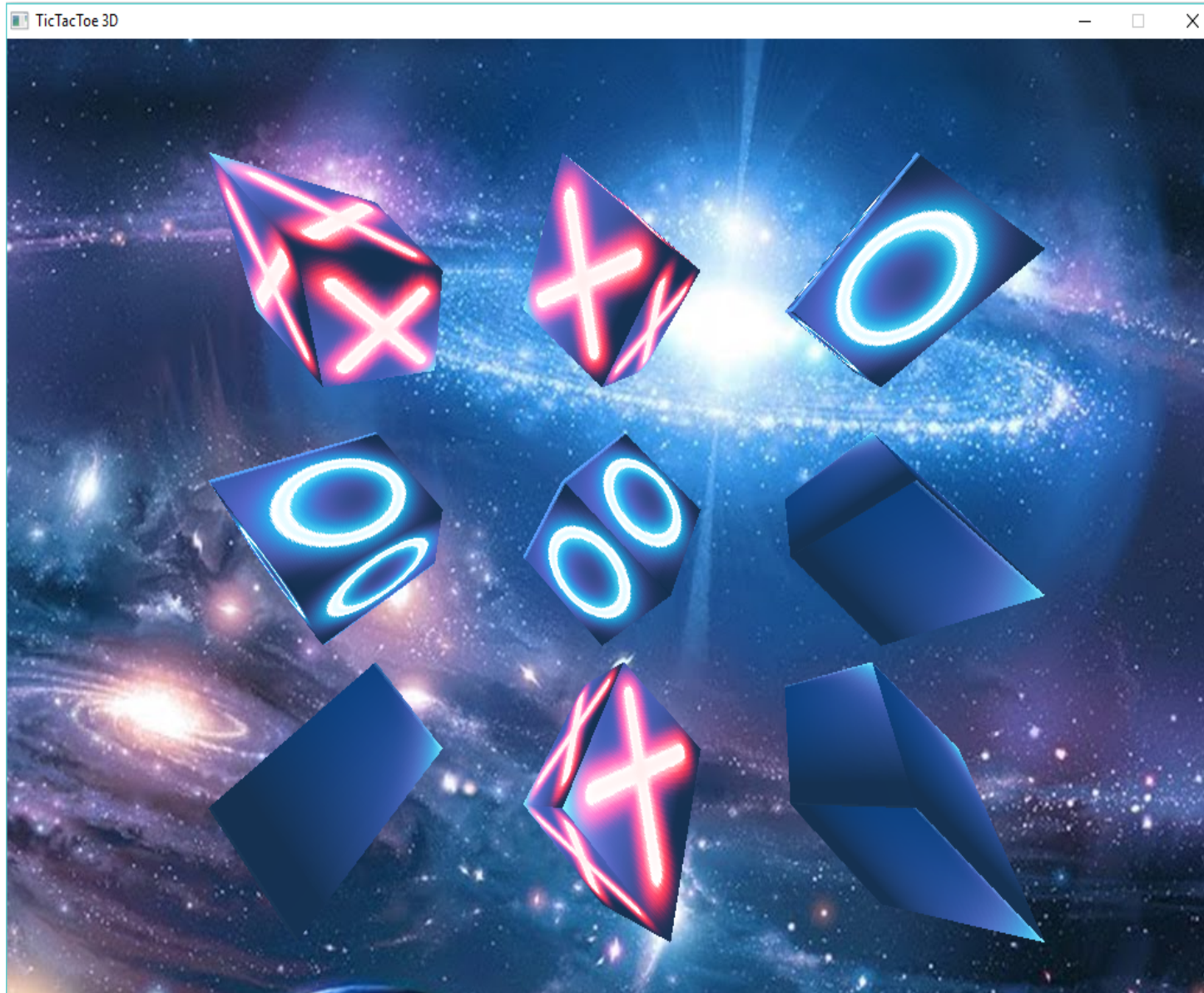


Super Man 2016 Game



The next screen shot for the TicTacToe 3D Game

Screen Shot:



1.13 Innovative

The language comes with better support for Natural Language Programming and Declarative Programming. The innovation comes in supporting these paradigms with new practical techniques on the top of Object-Oriented Programming and Functional Programming. No need to know anything about (Compilers and Parsing). You get the language constructs ready for use to create domain-specific languages in a fraction of time.

Articles:-

- Natural Language Programming Library :

<https://www.codeproject.com/Articles/1200766/Using-the-Natural-Language-Programming-Library-NLP>

- Natural Language Programming :

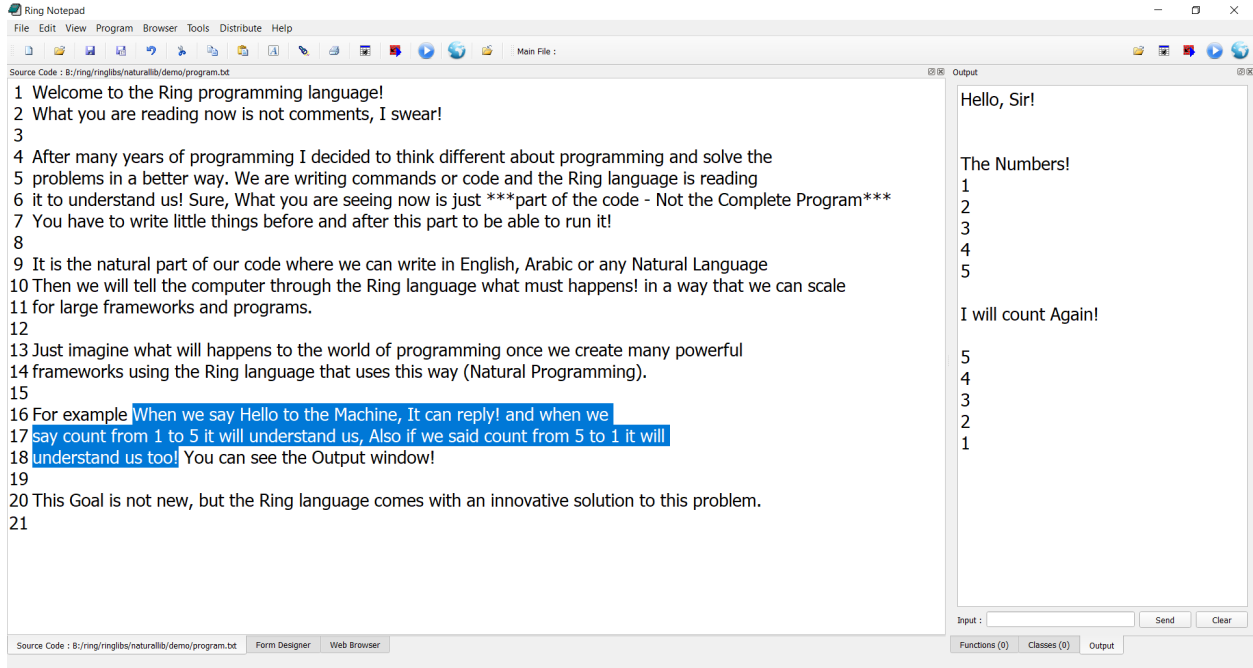
<https://www.codeproject.com/Articles/1138605/Natural-Language-Programming-in-the-Ring-Programmi>

- Syntax Flexibility :

<https://www.codeproject.com/Articles/1137388/Syntax-Flexibility-in-the-Ring-Programming-Language>

- The Ring Programming Language :

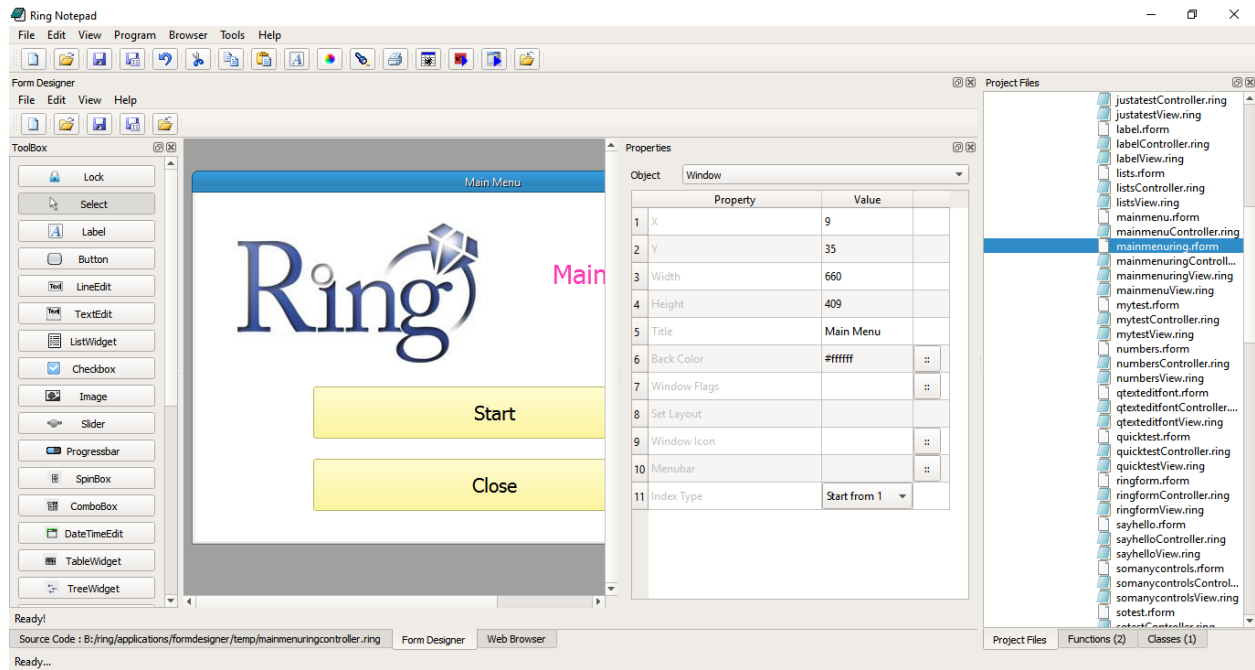
<https://www.codeproject.com/Articles/1089887/The-Ring-Programming-Language>



1.14 Practical

Many of the Ring libraries (StdLib, WebLib, Natural Library, Games Engine, etc.) and the Ring IDE (Ring Notepad, Form Designer, etc.) are written in the Ring language itself. Ring is ready for use in production and increase the developers productivity.

Check the Form Designer source code : <https://github.com/ring-lang/ring/tree/master/applications/formdesigner>



INTRODUCTION



Welcome to the Ring programming language!

Ring is an Innovative and practical general-purpose multi-paradigm language that can be embedded in C/C++ projects, extended using C/C++ code and/or used as standalone language. The supported programming paradigms are Imperative, Procedural, Object-Oriented, Functional, Meta programming, Declarative programming using nested structures, and Natural programming. The language is portable (Windows, Linux, macOS, Android, etc.) and can be used to create Console, GUI, Web, Games and Mobile applications. The language is designed to be Simple, Small, Flexible and Fast. Its Dynamic Language (Dynamic Typing and Weakly Typed) that compile the source code to byte code then execute it by the Ring Virtual Machine, which is integrated with the Ring Compiler in one program.

In this chapter we are going to discuss the goals behind the language design and implementation.

2.1 Motivation

In Nov. 2011, I started to think about creating a new version of the [Programming Without Coding Technology \(PWCT\)](#) software from scratch.

I was interested in creating multi-platform edition of the software beside adding support for Web & Mobile development. Most of the PWCT source code was written in VFP and the software comes with a simple scripting language for creating the components called (RPWI). The software contains components that support code generation in programming languages like Harbour, C, Supernova & Python.

What I was looking for is a programming language that can be used to build the development environment, provides multi-platform support, more productivity, better performance, can be used for components scripting & can be used for developing different kinds of applications.

Instead of using a mix of programming languages, I decided to use one programming language for creating the development environment, for components scripting & for creating the applications.

I looked at many programming languages like C, C++, Java, C#, Lua, PHP, Python & Ruby. I avoided using C or C++ directly because I want high-level of productivity more than the level provided by these languages, also a language behind visual programming environment for novice programmers or professionals must be easy to use & productive.

Java & C# are avoided for some reason too! I wanted to use a dynamic programming language and these languages are static typing, Java is multi-platform, also C# through Mono, but the use of huge number of classes and forcing the use of Object-Orientation, using a verbose language is not right for me. I need a small language, but fast and productive, also I need better control on the Garbage Collector (GC), I need a better one that is designed for fast applications.

Lua is small and fast, but it's avoided because I need more powerful language for large applications.

PHP is a Web programming language and it's syntax is very similar to C, this leads to a language not general as I want and not simple as I need to have.

Python & Ruby are more like what I need, but I need something more simple, smaller, faster & productive.

Python and Ruby are Case-Sensitive, the list index start counting from 0, you have to define the function before calling it, Ruby usage of Object-Orientation and message passing is more than what I need and decrease performance, Python syntax (indentation, using self, :, pass & _) is not good for my goals.

All of these languages are successful languages, and very good for their domains, but what I need is a different language that comes with new ideas and intelligent implementation (Innovative, Ready, Simple, Small, Flexible and Fast).

2.2 Ring and other languages

Ring is an innovative programming language that comes with better support for Natural Language Programming and Declarative Programming. The innovation comes in supporting these paradigms with new practical techniques on the top of Object-Oriented Programming and Functional Programming.

Also Ring is influenced by the next programming languages

- Lua
- Python
- Ruby
- C
- C#
- BASIC
- QML
- xBase
- Supernova

2.3 History

In Sept. 2013 I started the design and the implementation of the Ring programming language. After 21 months of development, In May 2015 the language Compiler & Virtual Machine were ready for use!

After that I spent three months testing the language again, trying to discover any bug to fix, writing better tests, by the end of August 2015, all know bugs were fixed, Writing many tests and testing automation helped a lot in getting a stable product.

In September 12, 2015, most of the documentation was written. Before releasing the language I started the marketing by writing a post in Arabic language about it to my facebook profile page asking for contributors interested in the language idea after reading a short description, In the same day I got a lot of emails from developers and friends interested to contribute!

Ring 1.0 is released on January 25, 2016

Ring 1.1 is released on October 6, 2016

Ring 1.2 is released on January 25, 2017

Ring 1.3 is released on May 15, 2017

Ring 1.4 is released on June 29, 2017

Ring 1.5 is released on August 21, 2017

Ring 1.6 is released on November 30, 2017

2.4 Features

The Ring language comes with the next features

Tip: The language is ready for production!

- Free Open Source (MIT License)
- Hybrid Implementation (Compiler + Virtual Machine)
- Declarative programming on the top of Object-Oriented programming
- Natural Language programming on the top of Object-Oriented programming
- Syntax Flexibility (You can change the language keywords and operators)
- Compact Syntax, No explicit end for statements (No ; or ENTER is required)
- Using braces { } we can access objects and use attributes/methods as variables/functions
- Transparent Implementation
- Visual Implementation - Developed using Visual Programming (PWCT)
- **A small language**
 - The compiler + The Virtual Machine are 15,000 lines of C code
 - The other 500,000 lines of code are related to libraries!
- Written in ANSI C (The code is generated)
- Optional Printing for Tokens/Grammar/Byte-Code during execution
- Portable (Windows, Linux & Mac OS X)
- Comments (One line & Multi-lines)
- Not Case-Sensitive
- Dynamic Typing
- Weakly typed
- Lexical Scoping (Global, Local & Object State)
- Default scope for variables inside functions (Local)
- Default scope for variables outside functions (global)
- Garbage Collector - Automatic Memory Management (Escape Analysis and Reference Counting)
- Structure Programming

- Rich control structures & Operators
- For in get item by reference not value, you can read/edit the item
- Use exit to go outside from more than one loop
- Procedures/Functions
- Main Function (optional)
- Call Function before the definition
- Recursion
- Multi-line literals
- Access (read/write) string letter by index
- The list index start by 1
- No keyword to end Functions/Classes/Packages
- Range operator ex: 1:10 and “a”:”z”
- First Class Variables, Lists, Objects and Functions
- Store/Copy Lists/Objects by value (Deep Copy)
- Pass Lists/Objects by reference
- Native Object-Oriented Support
 - Encapsulation
 - Setter/Getter (optional)
 - private state (optional)
 - Instantiation
 - Polymorphism
 - Composition
 - Inheritance (Single Inheritance)
 - Operator Overloading
 - Packages
- Reflection and Meta-programming
- Clear program structure (Statements then functions then packages & classes)
- Exception Handling
- Eval() to execute code during run-time
- 8-bit clean, work on binary data directly
- I/O commands
- Math functions
- String functions
- List functions
- File processing functions
- Database support (ODBC, SQLite & MySQL)

- Security Functions (OpenSSL)
- Internet Functions (LibCurl)
- Zip Functions
- CGI Library (Written in Ring)
 - HTTP Get
 - HTTP Post
 - File upload
 - Cookies
 - URL Encode
 - HTML Templates
 - HTML Special Characters
 - HTML Generation using Functions
 - HTML Generation using Classes
 - CRUD Example (using MVC)
 - Users Example (Register, Login and Check)
- Extension using C/C++ (Simple API)
- Embedding the language in C/C++ programs
- **Comes with code generator (Written in Ring) to quickly wrap C/C++ Libraries**
 - Used to Support Allegro by creating RingAllegro
 - Used to Support LibSDL by creating RingLibSDL
 - Used to Support Qt by creating RingQt
- Create 2D Games for Desktop and Mobile (Using the Allegro Library or the LibSDL Library)
- Comes with simple Game Engine for 2D Games
- Support FreeGLUT and OpenGL for 3D Graphics
- Create GUI Applications for Desktop and Mobile (Using the Qt Framework)
- Comes with IDE contains the Code Editor (Ring Notepad) and the Form Designer
- Comes with Ring2EXE to distribute applications

2.5 License

The Ring Programming Language

<http://ring-lang.net/>

Version 1.6

The MIT License (MIT)

Copyright (c) Mahmoud Fayed

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use,

copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LANGUAGE DESIGN

In this chapter we will learn about the basic concepts behind the language design.

3.1 Why Ring?

The language is simple, trying to be natural, encourage organization and comes with transparent and visual implementation. It comes with compact syntax and a group of features that enable the programmer to create natural interfaces and declarative domain-specific languages in a fraction of time. It is very small, fast and comes with smart garbage collector that puts the memory under the programmer control. It supports many programming paradigms, comes with useful and practical libraries. The language is designed for productivity and developing high quality solutions that can scale.

3.2 Designed for a Clear Goal

- Applications programming language.
- Productivity and developing high quality solutions that can scale.
- Small and fast language that can be embedded in C/C++ projects.
- Simple language that can be used in education and introducing Compiler/VM concepts.
- General-Purpose language that can be used for creating domain-specific libraries, frameworks and tools.
- Practical language designed for creating the next version of the Programming Without Coding Technology software.

3.3 Simple

Ring is a very simple language, and has a very straightforward syntax. It encourages programmers to program without boilerplate code

`See "Hello, World!"`

The Main function is optional and will be executed after the statements, and is useful for using the local scope.

`Func Main
 See "Hello, World!"`

Uses Dynamic Typing and Lexical scoping. No \$ is required before the variable name! You can use the '+' operator for string concatenation and the language is weakly typed and will convert automatically between numbers and strings based on the context.

```
nCount = 10      # Global variable
Func Main
    nID = 1 # Local variable
    See "Count = " + nCount + nl + " ID = " + nID
```

3.4 Trying to be natural

Ring is not case-sensitive

```
See "Enter your name ? "
Give name
See "Hello " + Name      # Name is the same as name
```

The list index starts from 1

```
aList = ["one", "two", "three"]
See aList[1]      # print one
```

Call functions before definition

```
one()
two()
three()
Func one
    See "One" + nl
Func two
    See "two" + nl
Func three
    See "three" + nl
```

The assignment operator uses Deep copy (no references in this operation)

```
aList = ["one", "two", "three"]
aList2 = aList
aList[1] = 1
see alist[1]      # print 1
see aList2[1]     # print one
```

Pass numbers and strings by value, but pass lists and objects by reference. The for in loop can update the list items.

```
Func Main
    aList = [1,2,3]
    update(aList)
    see aList      # print one two three

Func update aList
    for x in aList
        switch x
        on 1 x = "one"
        on 2 x = "two"
        on 3 x = "three"
        off
    next
```

Using Lists during definition

```
aList = [ [1,2,3,4,5] , aList[1] , aList[1] ]
see aList      # print 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
```

Exit from more than one loop

```
for x = 1 to 10
    for y = 1 to 10
        see "x=" + x + " y=" + y + nl
        if x = 3 and y = 5
            exit 2      # exit from 2 loops
        ok
    next
next
```

3.5 Encourage Organization

The language encourage organization, Forget bad days using languages where the programmer start with function then class then function and a strange mix between things!

Each source file follow the next structure

- Load Files
- Statements and Global Variables
- Functions
- Packages and Classes

This enable us to use Packages, Classes and Functions without the need to use a keyword to end these components.

We can write one line comments and multi-line comments The comment starts with # or // Multi-line comments are written between /* and */

```
/*
    Program Name : My first program using Ring
    Date        : 2015.05.08
*/

See "What is your name? "      # print message on screen
give cName                    # get input from the user
see "Hello " + cName           # say hello!

// See "Bye!"
```

3.6 Compact Syntax

The language is not line sensitive, you don't need to write ; after statements, also you don't need to press ENTER or TAB, so we can write the next code

```
See "The First Message" See " Another message in the same line! " + nl
See "Enter your name?" Give Name See "Hello " + Name
```

The next code create a class called Point contains three attributes X,Y and Z. No keywords is used to end the package/class/function definition. Also, we can write the attributes names directly below the class name.

```
Class Point X Y Z
```

We can use classes and functions before their definition, In this example we will create new object, set the object attributes then print the object values.

```
o1 = New point o1.x=10 o1.y=20 o1.z=30 See O1 Class Point X Y Z
```

Instead of using the dot '.' operator to access the object attributes and methods we can use braces { } to access the object, then we can use the object attributes and methods.

```
o1 = New point { x=10 y=20 z=30 } See O1 Class Point X Y Z
```

Now we will call a method after accessing the object using { }

```
oPerson = new Person
{
    Name = "Somebody"
    Address = "Somewhere"
    Phone = "0000000"
    Print() # here we call the Print() method
}
Class Person Name Address Phone
    Func Print
        See "Name :" + name + nl +
            "Address :" + Address + nl +
            "Phone : " + phone + nl
```

When we use { } to access the object then write any attribute name, the language will check the class for any setter/getter methods that will be called automatically.

```
New Number {
    See one # Execute GetOne()
    See two # Execute GetTwo()
    See three # Execute GetThree()
}
Class Number one two three
    Func GetOne
        See "Number : One" + nl
        return 1
    Func GetTwo
        See "Number : Two" + nl
        return 2
    Func GetThree
        See "Number : Three" + nl
        return 3
```

3.7 Define Natural Statements

After the object access using { } if the class contains a method called BraceEnd() it will be executed!

```
TimeForFun = new journey
# The first surprise!
TimeForFun {
    Hello it is me # What a beautiful programming world!
}
# Our Class
Class journey
```

```

hello=0 it=0 is=0 me=0
func GetHello
    See "Hello" + nl
func braceEnd
    See "Goodbye!" + nl

```

We can execute code written in strings using the Eval() function

```

cCode = "See 'Code that will be executed later!' "
Eval(cCode)      # execute the code to print the message

```

We can create a list then execute code generated from that list

```

aWords = ["hello","it","is","me"]
for word in aWords cCode=word+"=0" eval(cCode) next

```

We can read text files using the Read(cFileName) function and we can write files using the Write(cFileName,cString) function.

```

See "Enter File Name:" Give cFileName See Read(cFileName) # Print the file content

```

The next example presents how to create a class that defines two instructions The first instruction is : I want window The second instruction is : Window title = Expression Also keywords that can be ignored like the 'the' keyword

```

New App
{
    I want window
    The window title = "hello world"
}

Class App

    # Attributes for the instruction I want window
    i want window
    nIwantwindow = 0
    # Attributes for the instruction Window title
    # Here we don't define the window attribute again
    title
    nWindowTitle = 0
    # Keywords to ignore, just give them any value
    the=0

    func geti
        if nIwantwindow = 0
            nIwantwindow++
        ok

    func getwant
        if nIwantwindow = 1
            nIwantwindow++
        ok

    func getwindow
        if nIwantwindow = 2
            nIwantwindow= 0
            see "Instruction : I want window" + nl
        ok
        if nWindowTitle = 0
            nWindowTitle++

```



```

                                ok

func setttitle cValue
    if nWindowTitle = 1
        nWindowTitle=0
        see "Instruction : Window Title = " + cValue + nl

                                ok

```

To complete the previous example, use `read()` to get the content of a file that contains

```

I want window
The window title = "hello world"

```

Then use `eval()` to execute the content of that file!. Also, you can update the methods `GetWindow()` and `SetTitle()` to create Real windows using the GUI Library

3.8 Define Declarative Languages

We learned how to use Natural statements to execute our code and using the same features we can use nested structures to execute our code.

The next example from the Web library, generate HTML document using the Bootstrap library. No HTML code is written directly in this example, we created a similar language (just as example) Then using this declarative language that uses nested structures, we generated the HTML Document.. The idea in this example is that the `GetDiv()` and `GetH1()` methods return an object that we can access using `{ }` and after each object access the method `BraceEnd()` will be executed to send the generated HTML to the parent object until we reach to the root where `BraceEnd()` will print the output.

```

Load "weblib.ring"
Import System.Web

Func Main

    BootstrapWebPage ()
    {
        div
        {
            classname = :container
            div
            {
                classname = :jumbotron
                H1 {    text("Bootstrap Page")    }
            }
            div
            {
                classname = :row
                for x = 1 to 3
                div
                {
                    classname = "col-sm-4"
                    H3 { html("Welcome to the Ring programming language") }
                    P  { html("Using a scripting language is very fun!") }
                }
                next
            }
        }
    }
}

```

The classes that power the declarative interface looks like this

```

Class Link from ObjBase
    title link
    Func braceend
        cOutput = nl+GetTabs() + "<a href='" +
            Link + "'> " + Title + " </a> " + nl

Class Div from ObjBase
    Func braceend
        cOutput += nl+'<div'
        addattributes()
        AddStyle()
        getobjdata()
        cOutput += nl+"</div>" + nl
        cOutput = TabMLString(cOutput)

```

3.9 Transparent Implementation

Ring comes with transparent implementation. We can know what is happening in each compiler stage and what is going on during the run-time by the Virtual Machine Example : `ring helloworld.ring -tokens -rules -ic`

```
See "Hello, World!"
```

Output

```

=====
Tokens - Generated by the Scanner
=====

Keyword : SEE
Literal : Hello, World!
EndLine

=====

Grammar Rules Used by The Parser
=====

Rule : Program --> {Statement}

Line 1
Rule : Factor --> Literal
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR --> BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot --> EqualOrNot
Rule : Expr --> LogicNot
Rule : Statement --> 'See' Expr

=====

```

```
=====
Byte Code - Before Execution by the VM
=====
```

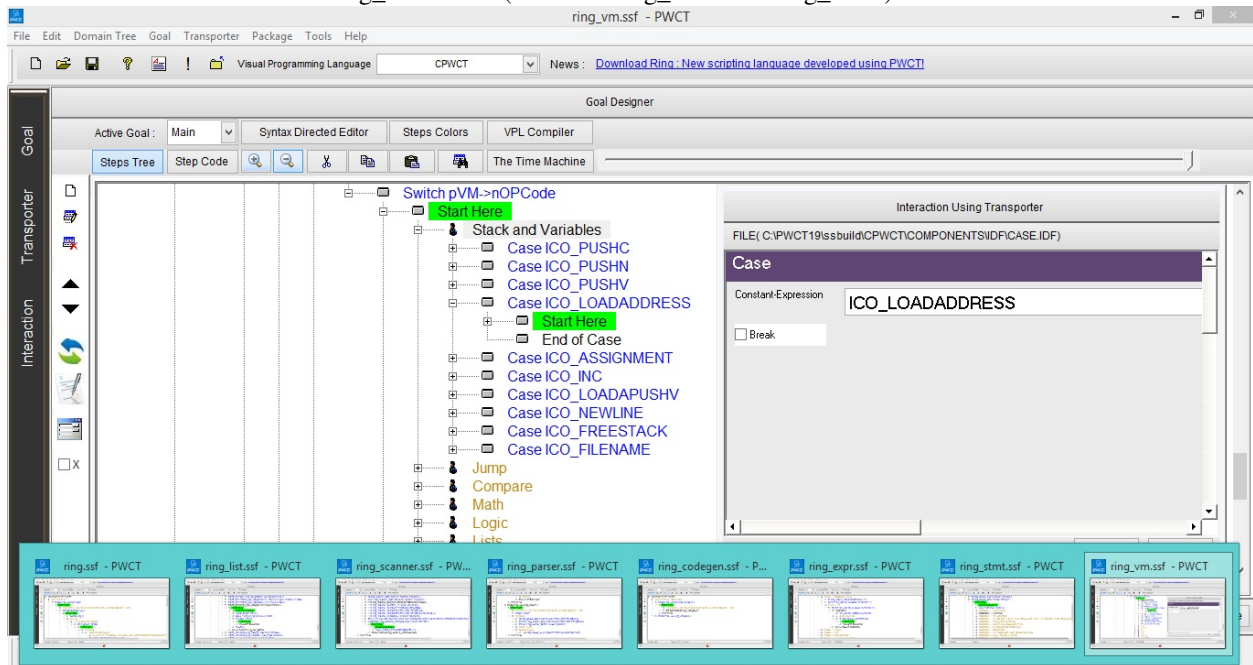
PC	OPCode	Data
1	FuncExE	
2	PushC	Hello, World!
3	Print	
4	ReturnNull	

```
=====
Hello, World!
```

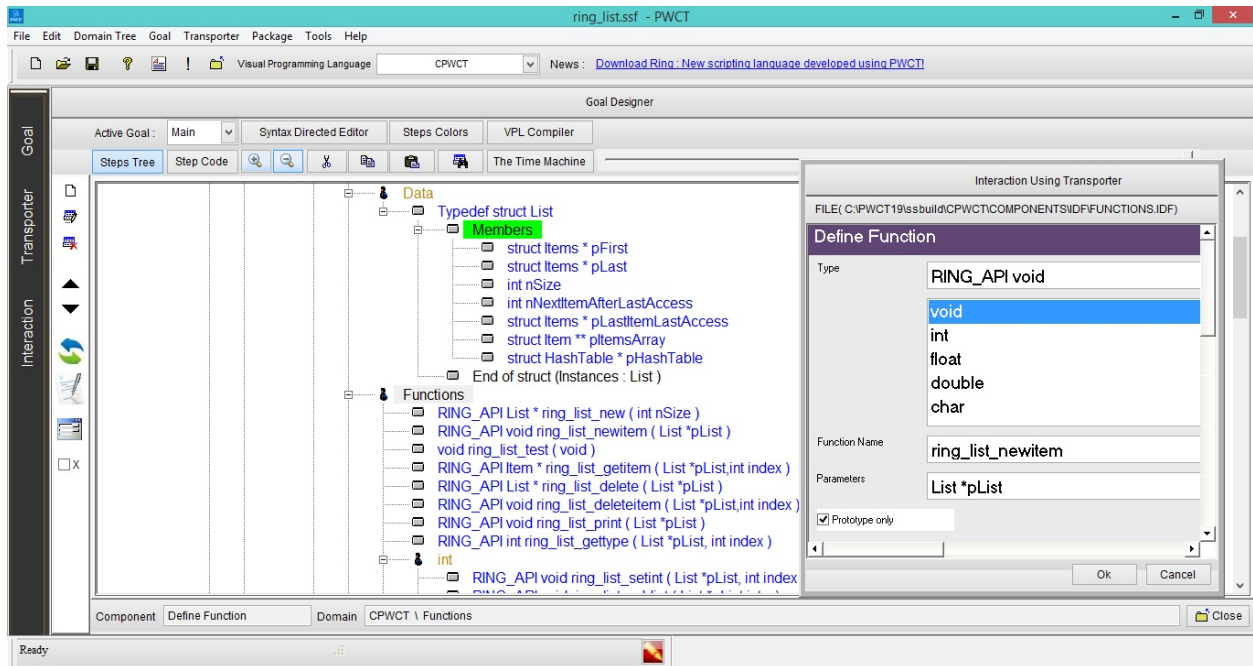
3.10 Visual Implementation

The Ring programming language is designed using the PWCT visual programming tool and you will find the visual source of the language in the folder “visualsrc” - *.ssf files and the generated source code (In the C Language) in the src folder and the include folder.

The next screen shot from the ring_vm.ssf file (Generate ring_vm.c and ring_vm.h)



The next screen shot from the ring_list.ssf file (Generate ring_list.c and ring_list.h)



3.11 Smart Garbage Collector

Avoid memory problems :-

- Invalid Memory Access
- Memory leaks
- Uninitialized Memory Access
- Dangling pointer

Rules :-

- Global variables always stay in the memory, until you delete these variables using the assignment statement.
- Local variables always deleted after the end of the function.
- The programmer have full control on when to delete the variable from the memory using the Assignment statement.

Example:

```
aList = [1,2,3,4,5]
aList = "nice"
```

After the second line directly, The list [1,2,3,4,5] will be deleted from the memory and we will have a string “nice”

- The programmer can call the function `callgc()` to force running the garbage collector.
- If we have a reference to a variable (when we pass objects and lists to functions), then deleting variables will be based on reference counting, if no references everything will be deleted, but if we have a reference, the data will stay in memory.

WHAT IS NEW IN RING 1.6?

In this chapter we will learn about the changes and new features in Ring 1.6 release.

4.1 List of changes and new features

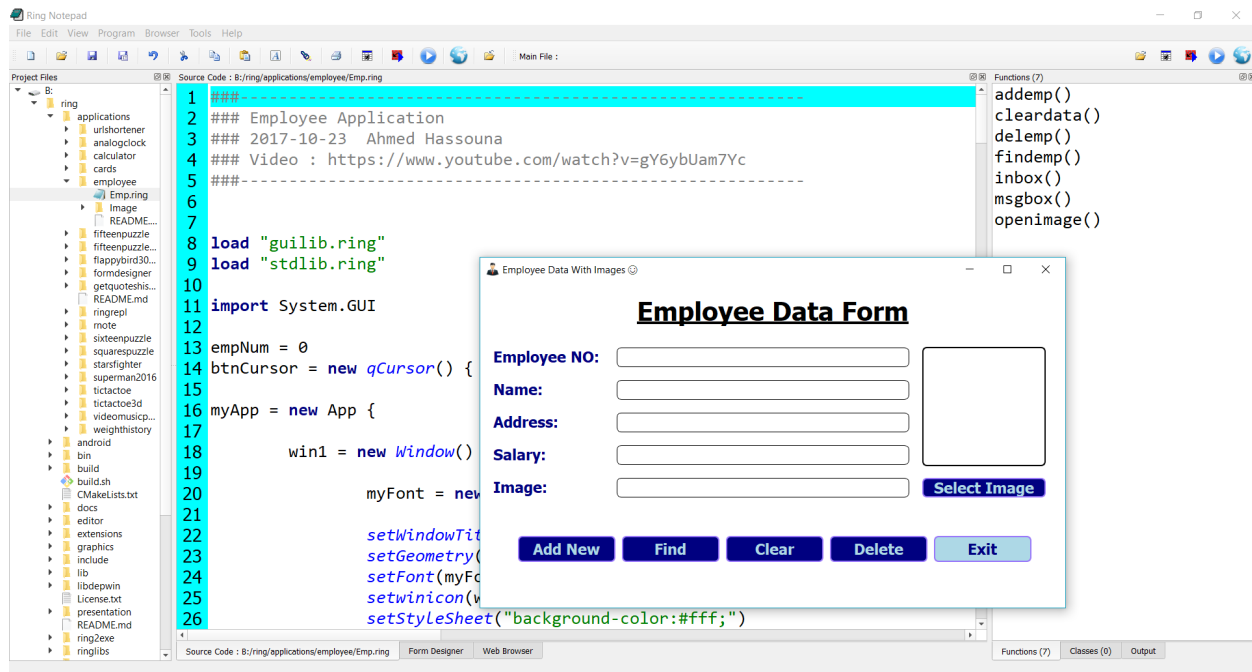
Ring 1.6 comes with many new features!

- Employee Application
- New Tool: Ring2EXE
- Better Ring For Android
- New Tool : Folder2qrc
- Better Scripts for building Ring
- RingConsoleColors Extension
- RingMurmurHash Extension
- Better Ring Notepad
- Better RingQt
- Better StdLib
- Better RingVM
- Better RingREPL
- Using Tab instead of char(9)
- Using CR as Carriage return
- Using the ! operator as not
- Using && and || operators
- Using ? to print expression then new line

4.2 Employee Application

The Employee application is added to ring/applications

Developer: Ahmed Hassouna



4.3 New Tool: Ring2EXE

In Ring 1.6 we have a nice tool called Ring2EXE (Written in Ring itself)

Using Ring2EXE we can distribute applications quickly for Windows, Linux, macOS and Mobile devices

Read the chapter “Distributing Ring Applications using Ring2EXE” for more information!

4.4 Better Ring For Android

Ring For Android (using RingQt) is updated to use the Ring Object File (*.ringo) instead of using many source code files (*.ring)

The next screen shot is an example of building the cards game for Android

We are using cards.ringo instead of cards.ring

If you have large project (many *.ring files) it will use only one *.ringo file.



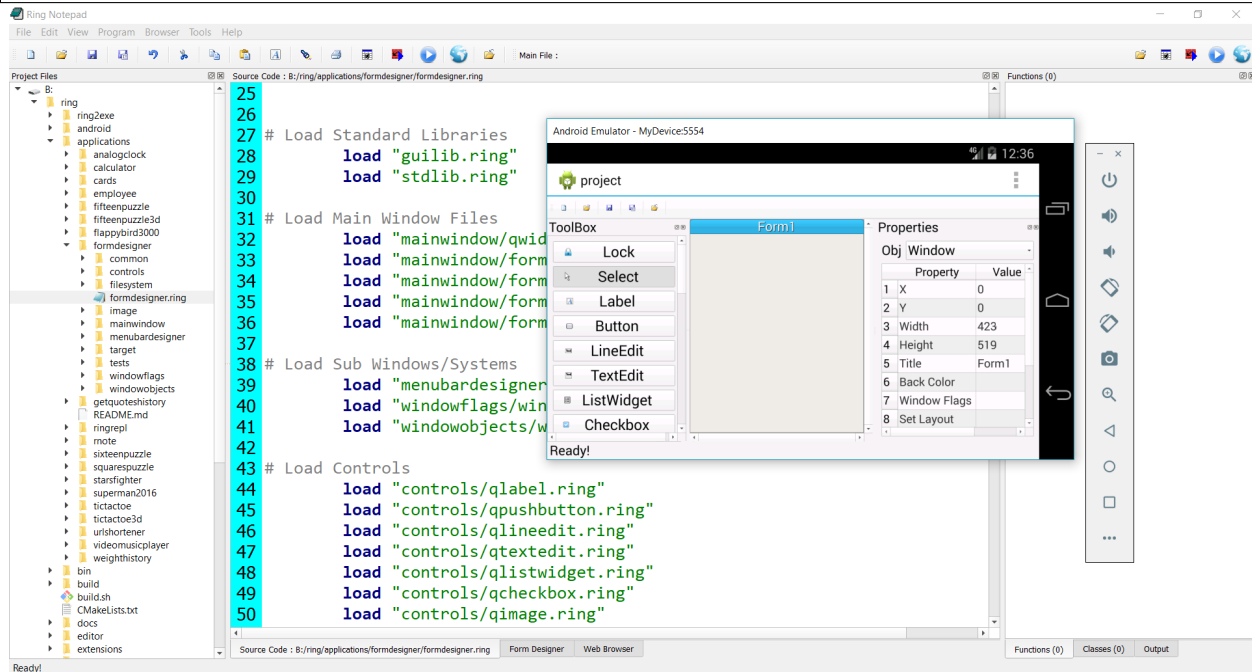
To prepare Qt project to distribute RingQt application for Mobile use Ring2EXE

Example

```
ring2exe cards.ring -dist -mobileqt
```

Example (2)

```
ring2exe formdesigner.ring -dist -mobileqt
```



4.5 New Tool: Folder2qrc

When we have large RingQt project that contains a lot of images and files, We need to add these files to the resource file (*.qrc) when distributing applications for Mobile devices.

Instead of adding these files one by one, Ring 1.6 comes with a simple tool that save our time, It's called Folder2qrc.

Example:

```
folder2qrc formdesigner.ring
```

We determine the main source file while we are in the application folder, and Folder2qrc will check all of the files in the current folder and sub folders, Then add them to the resource file after the mainfile.ringo (In our example this will be formdesigner.ringo)

The output file will be : project.qrc

You can open it and remove the files that you don't need in the resources!

4.6 Better Scripts for building Ring

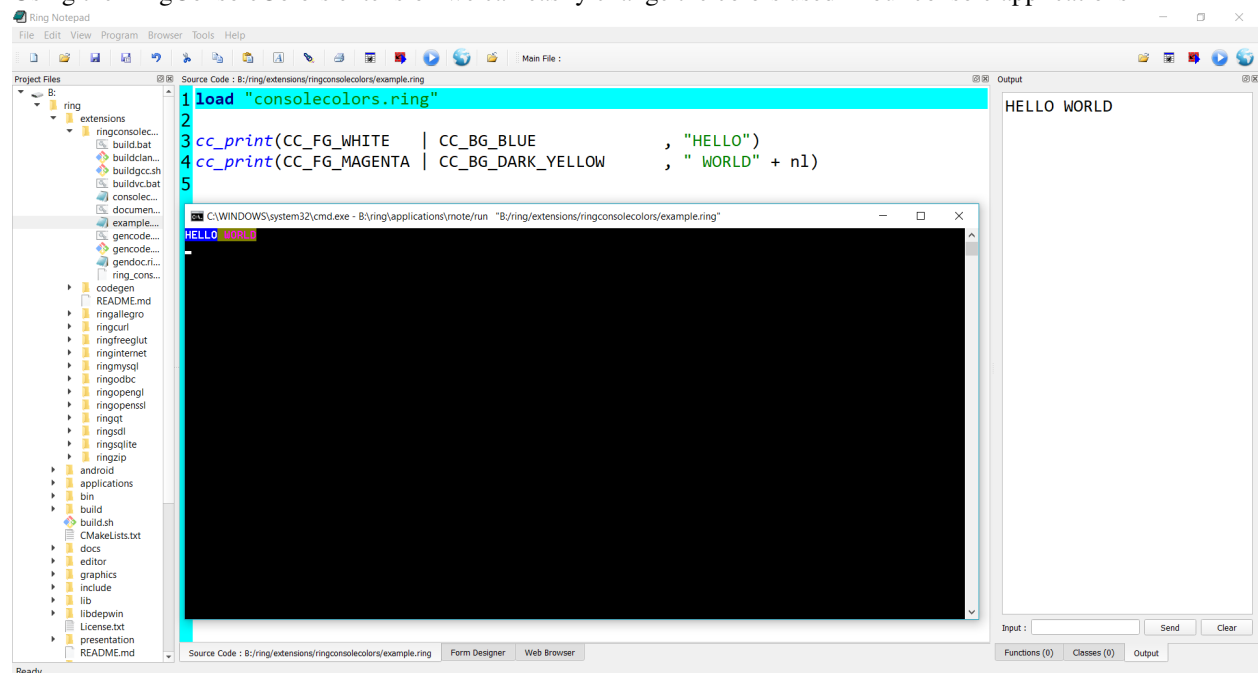
Ring 1.6 comes with better scripts for building Ring from source code.

The updates are tested on 32bit and 64bit systems on Windows, Linux (Ubuntu,Fedora) and macOS.

The scripts for Windows are updated to find the Visual C/C++ compiler based on your Visual Studio version.

4.7 RingConsoleColors Extension

Using the RingConsoleColors extension we can easily change the colors used in our console applications



For more information check the RingConsoleColors chapter in the documentation.

4.8 RingMurmurHash Extension

Ring 1.6 comes with the RingMurmurHash extension!

Developer: Hassan Ahmed

Example:

```
load "murmurhashlib.ring"

key = "Ring Language"

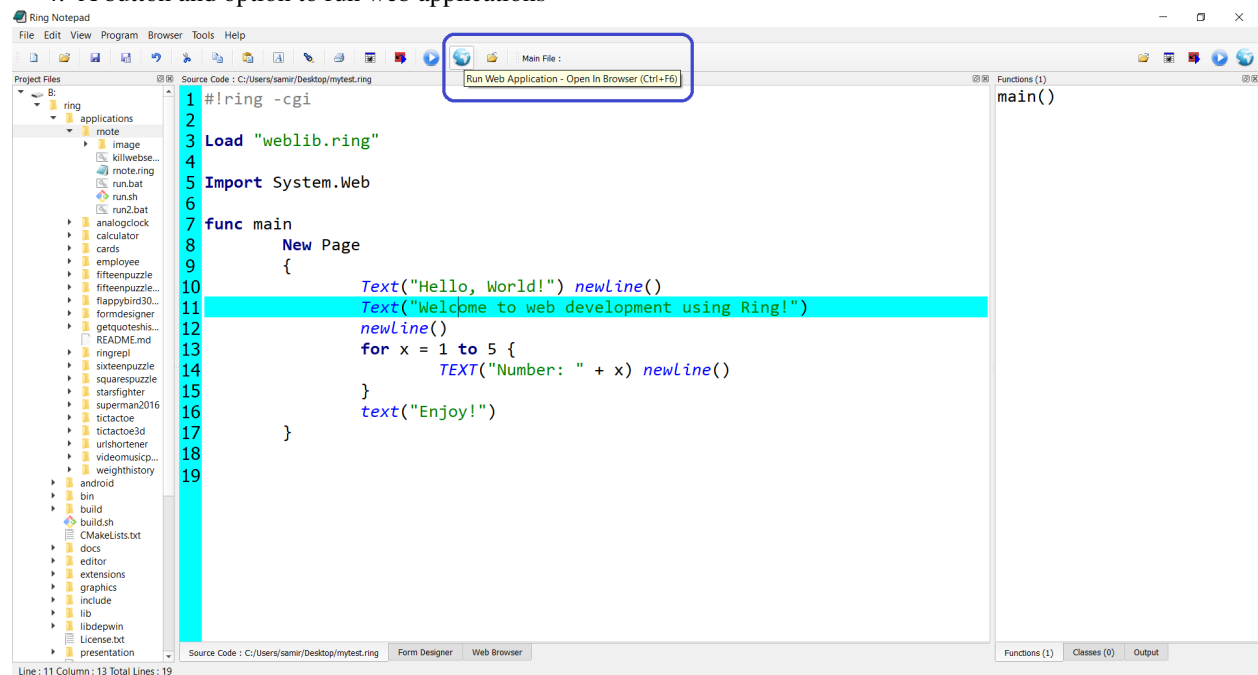
see murmurhash3_x86_32(key, 0, 0) + nl // Output: 1894444853
see murmurhash3_x86_32(key, 0, 1) + nl // Output: 70eae35
```

For more information check the RingMurmurHash chapter in the documentation.

4.9 Better Ring Notepad

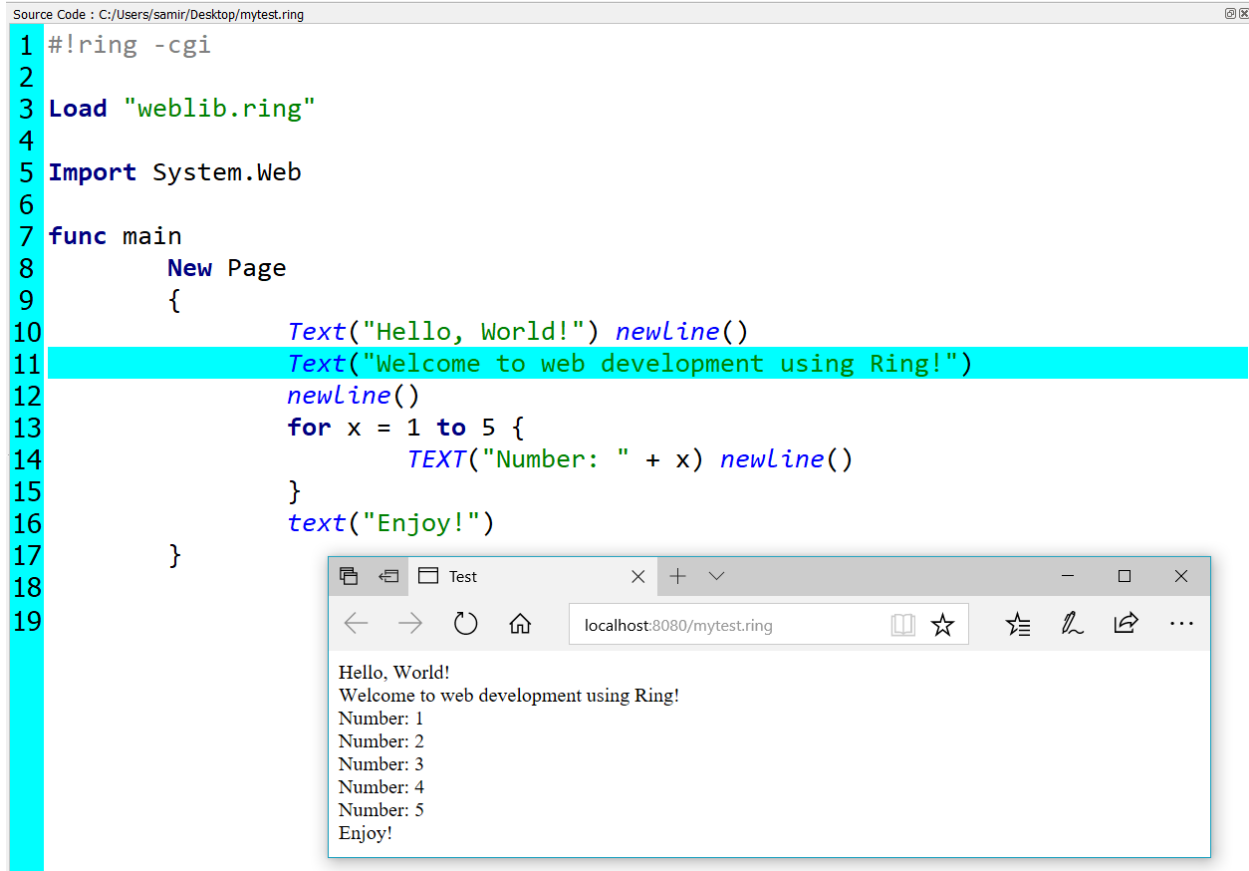
Ring Notepad comes with the next updates

1. Automatic setting for the Main File when we Run the application (using the Main file buttons).
2. Main File - Automatic save before running.
3. When we run GUI application - don't change the focus to the text box used for the input in the Output Window.
4. A button and option to run web applications

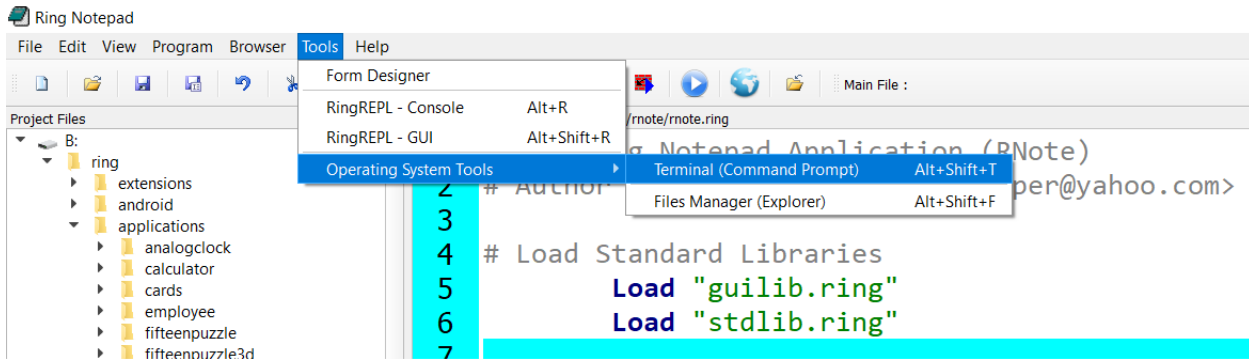


For Windows users, Ring 1.6 comes with Apache Web server!

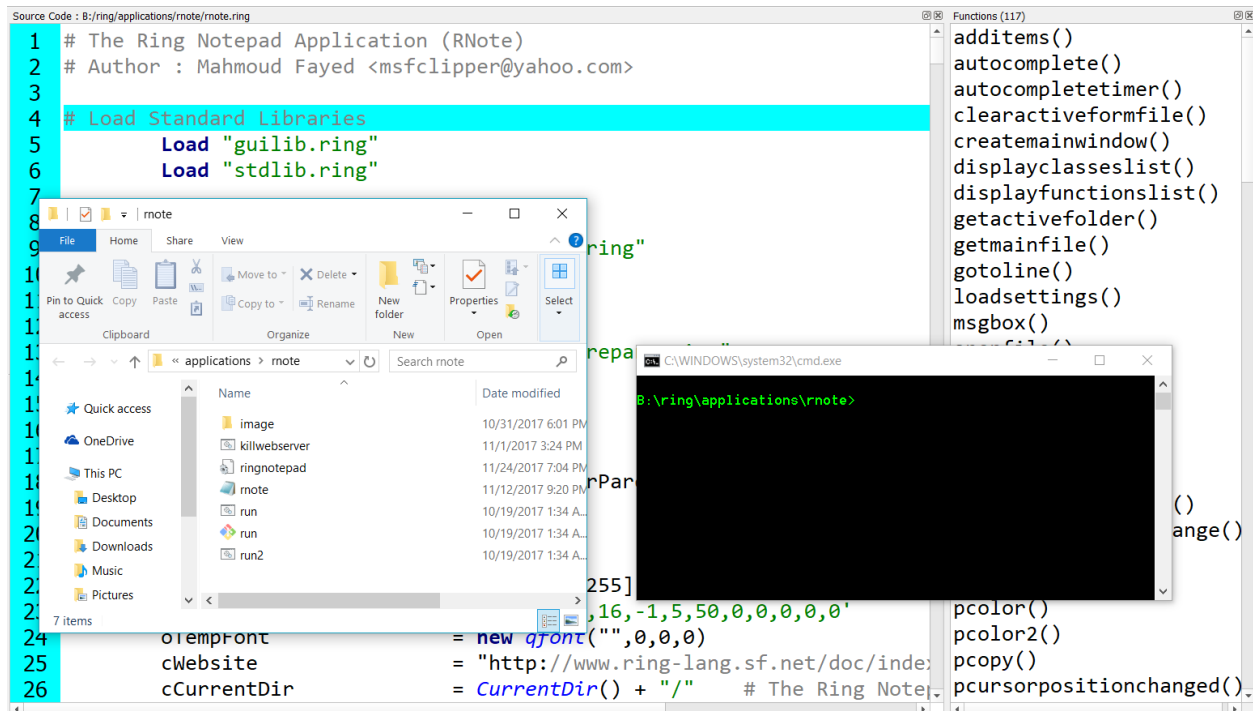
We can run any web application from any folder directly without doing any configuration.



5. Tools - Operating System - Terminal (Command Prompt) & Files Manager (Explorer).

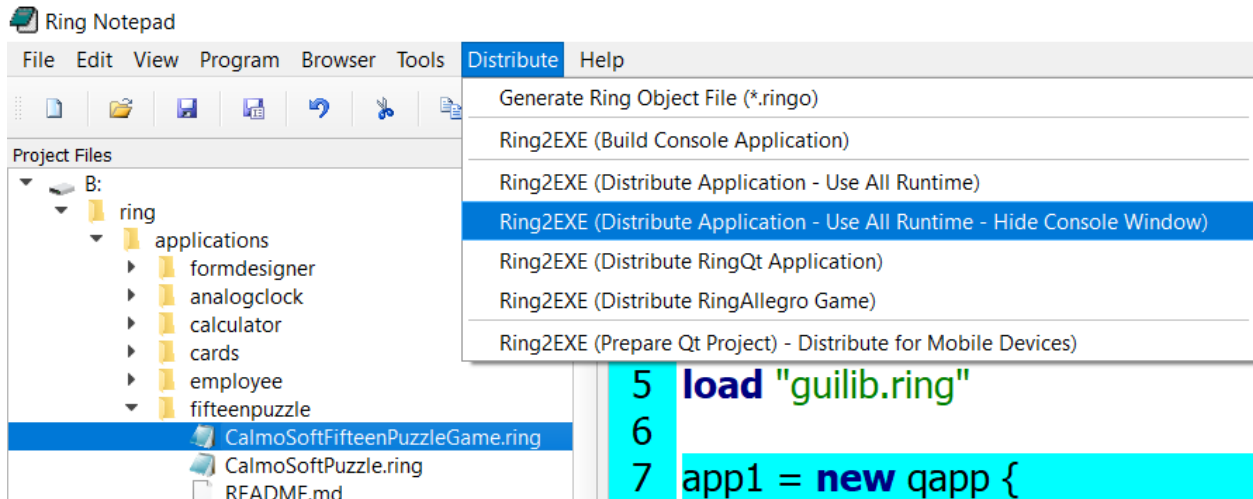


So we can quickly open the Command Prompt or the Explorer at the application folder.



6. Support *.sh & *.bat extensions.

7. New Menu: Distribute



4.10 Better RingQt

RingQt comes with the next updates

1. QAllEvents - getkeytext() Method
2. QSqlQuery - exec_2() Method
3. QDockWidget Events
4. AppFile() Function
5. IsMobile() Function

6. QRegion Class
7. QCoreApplication class

4.11 Better StdLib

StdLib comes with the next updates

1. Factors() function is updated (Return the output instead of printing it)
2. Palindrome() function is updated (Return the output instead of printing it)
3. Using stdlibcore.ring we can use the StdLib functions (Without Classes)

Also this is useful when developing standalone console applications

Because using stdlib.ring (functions & classes) will load libraries like RingLibCurl, RingOpenSSL, etc.

4. New Functions

- SystemSilent(cCommand) Function : Execute system commands without displaying the output.
- OSCreateOpenFolder(cFolder) : Create folder then change the current folder to this new folder
- OSCopyFolder(cParentFolder,cFolderNameToCopy) : Copy folder to the current directory
- OSDeleteFolder(cFolder) : Delete Folder
- OSCopyFile(cFileName) : Copy File to the current directory
- OSDeleteFile(cFileName) : Delete File
- OSRenameFile(cOldFileName,cNewFileName) : Rename file

4.12 Better RingVM

RingVM comes with the next updates

1. Support using many getter methods in Expressions
2. Support using this & self in setter/getter/normal methods without calling setter/getter methods.
3. TempName() function is updated (Better Code)
4. ExeFileName() function is updated (Better Code)
5. Private Attributes - Support re-usage in the class region (After the keyword private)
6. Ring API : ring_scanner_runobjstring()
7. ring_state_setvar() function

4.13 Better RingREPL

RingREPL comes with the next updates

1. RingREPL will get command line parameters to determine the style.
2. Setting RingREPL Style based on Ring Notepad Style.

4.14 Using Tab instead of char(9)

The variable Tab is defined to be used instead of char(9)

Example (1):

```
see :one + nl + tab + :two + nl + tab + tab + :three
```

Output:

```
one
    two
        three
```

You can change the variable to another value

Example (2):

```
tab = " "
see :one + nl + tab + :two + nl + tab + tab + :three
```

Output:

```
one
  two
   three
```

4.15 Using CR as Carriage return

The next example count from 1 to 10 in the same line during 10 seconds

```
load "stdlibcore.ring"
for x = 1 to 10 see x sleep(1) see cr next
```

4.16 Using the ! operator as not

We have = and != in the Ring language

But for the logical operators we have and, or & not

Now we can use the ! operator as not

Example:

```
if ! false
    see "True!" + nl
ok
```

Output

```
True!
```

4.17 Using && and || operators

In Ring we have the next keywords for the logical operations

- and
- or
- not

Now we have also the next operators

- &&
- ||
- !

Example:

```
if one() and two()
    see "Test1 - Fail" + nl
else
    see "Test1 - Pass" + nl
ok

if two() or one()
    see "Test2 - Pass" + nl
else
    see "Test2 - Fail" + nl
ok

if one() && two()
    see "Test3 - Fail" + nl
else
    see "Test3 - Pass" + nl
ok

if two() || one()
    see "Test4 - Pass" + nl
else
    see "Test4 - Fail" + nl
ok

func one return True

func two return False
```

Output:

```
Test1 - Pass
Test2 - Pass
Test3 - Pass
Test4 - Pass
```

4.18 Using ? to print expression then new line

It's common to print new line after printing an expression, We can use the ? operator to do that!

Example:

```
? "Hello, World!"
for x = 1 to 10
```

```
? x  
next
```

Output:

```
Hello, World!  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

WHAT IS NEW IN RING 1.5?

In this chapter we will learn about the changes and new features in Ring 1.5 release.

5.1 List of changes and new features

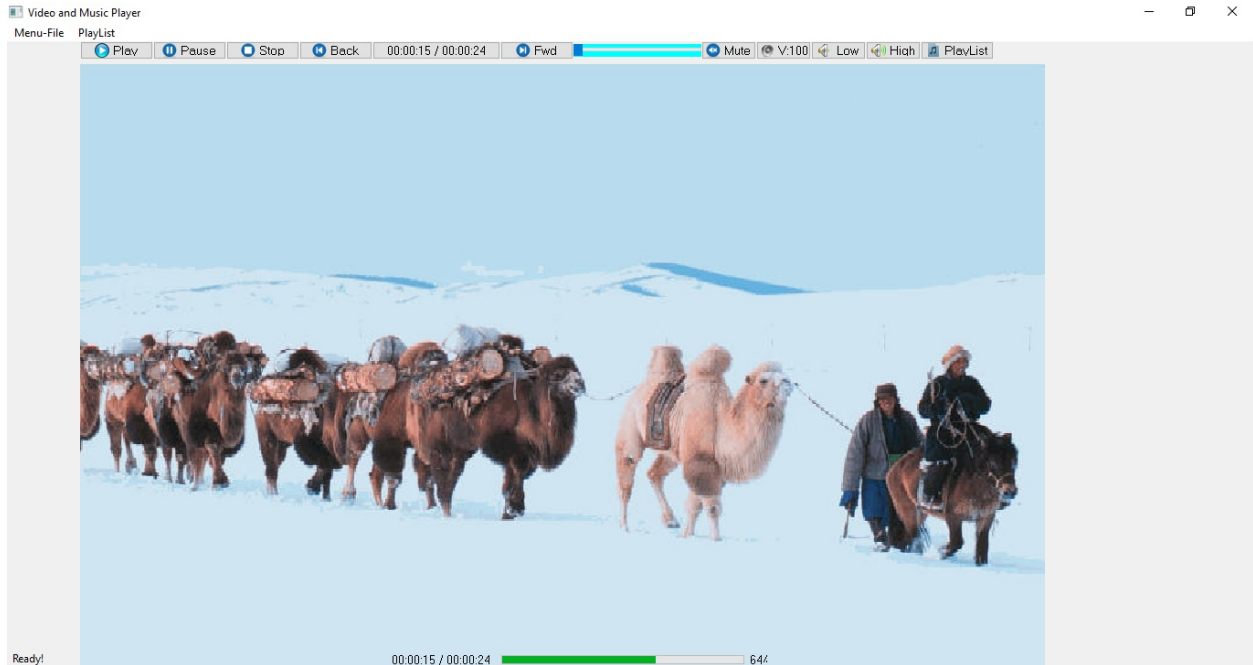
Ring 1.5 comes with many new features!

- Video-Music-Player Application
- Windows StartUp Manager Application
- Calculator Application
- Better Ring Notepad
- Better StdLib
- Better WebLib
- Better RingQt
- Better Objects Library
- RingFreeGLUT Extension
- RingOpenGL Extension
- Better Code Generator for Extensions
- Better Documentation Generator for Extensions
- Ring VM - Tracing Functions
- Trace Library and Interactive Debugger
- More Syntax Flexibility
- Type Hints Library
- Better Quality

5.2 Video-Music-Player Application

The Video-Music-Player application is added to the Applications folder.

Screen Shot:

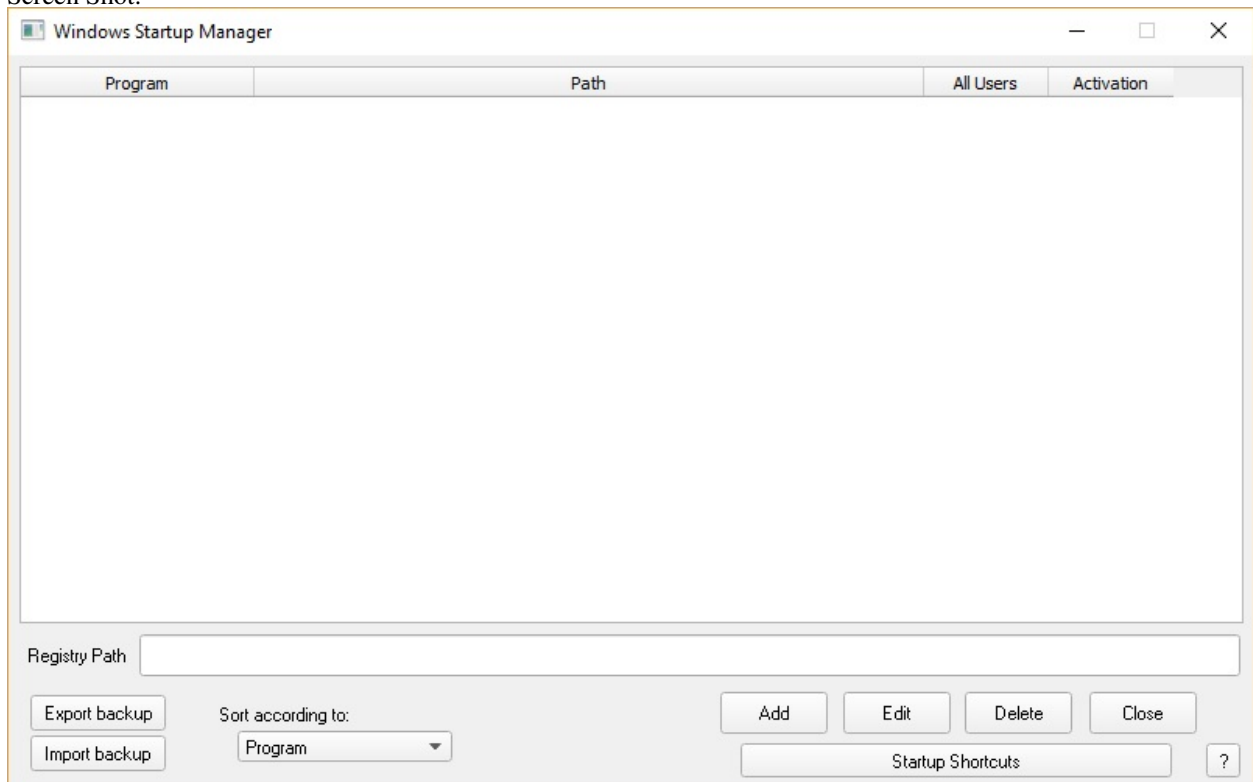


5.3 Windows StartUp Manager Application

The Windows StartUp Manager

URL : <https://github.com/ring-lang/WinStartupManager>

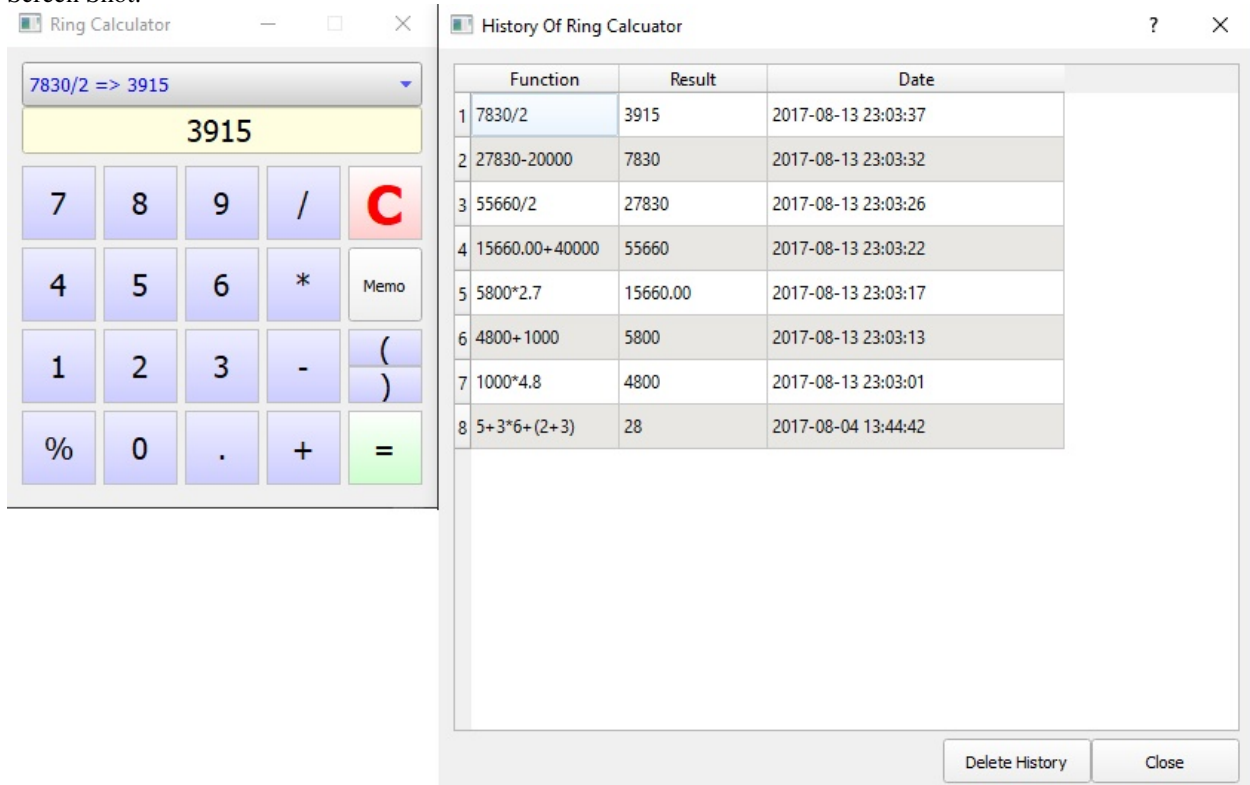
Screen Shot:



5.4 Calculator Application

The Calculator application is added to the Applications folder.

Screen Shot:



5.5 Better Ring Notepad

1. Ring Notepad is updated to include some new styles and the Main File ToolBar

The idea of the Main File ToolBar is to determine the main file in the project. When the project contains many source code files

This way you can run the project (Main File) at any time while opening other files in the project without the need to switch to the Main File to run the project.

To quickly use this feature

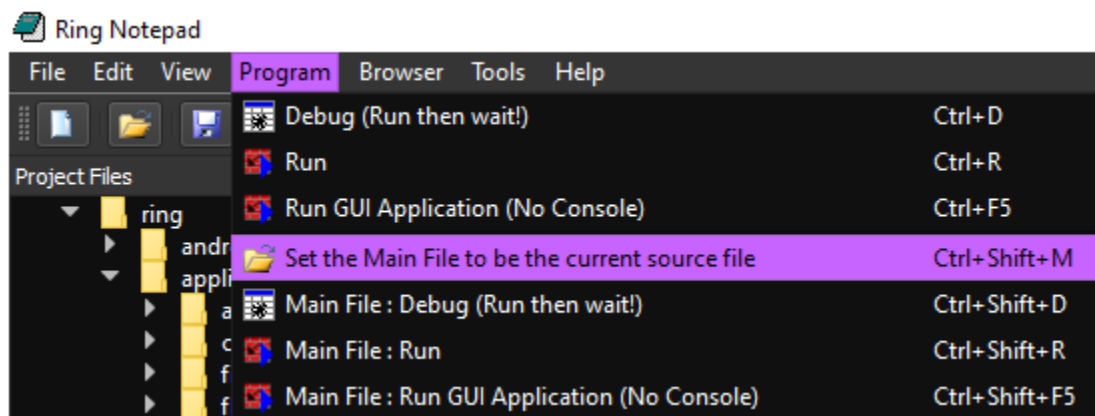
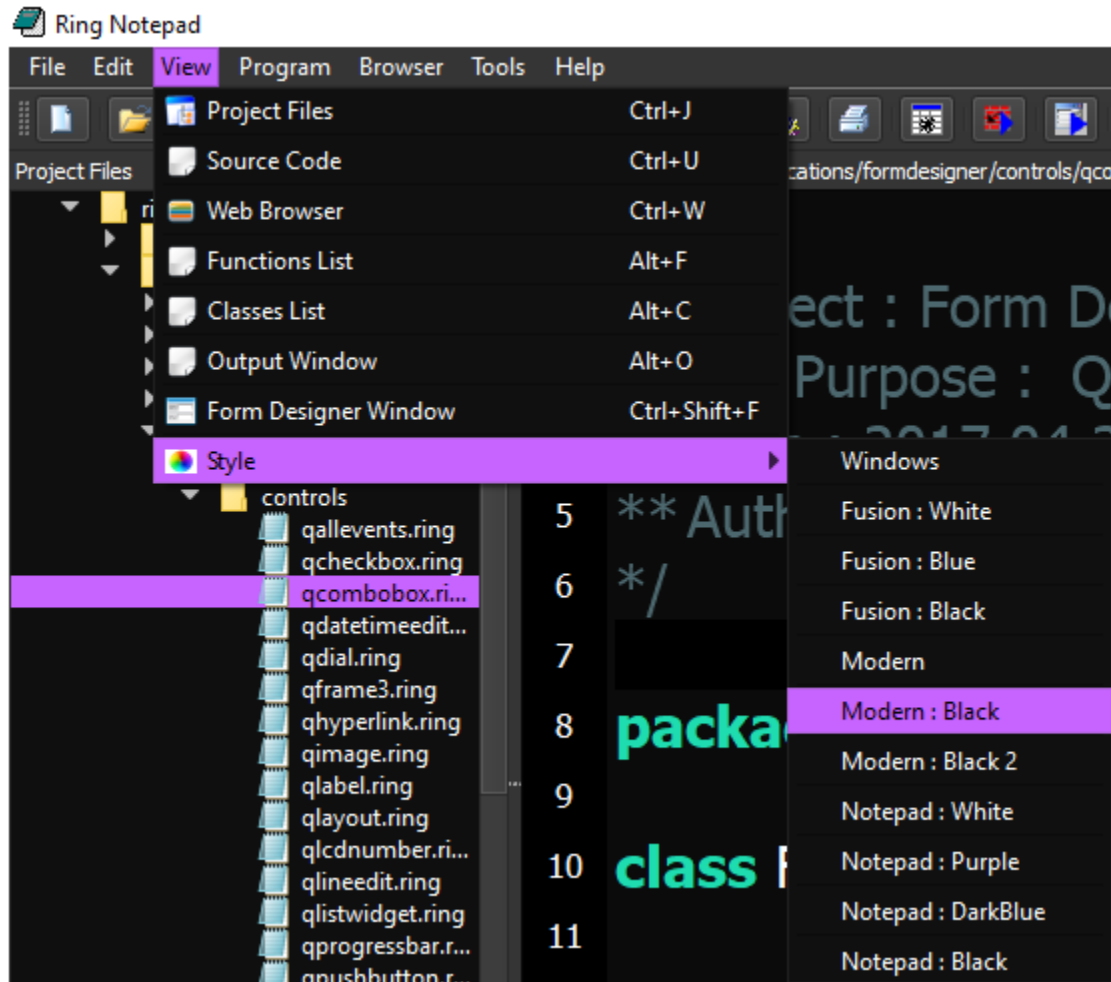
(Open the project main file)

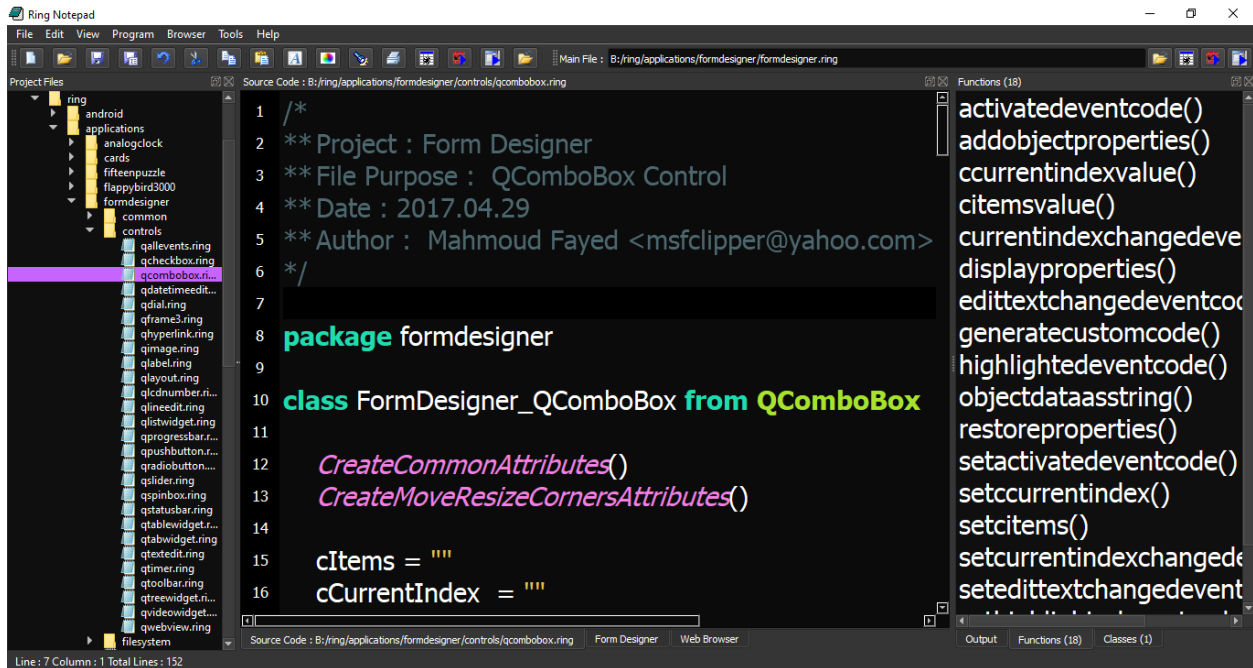
Press **Ctrl+Shift+M** to set the current source code file as the main file

Open and modify other source code files in the project

To run the project (Main File) at any time press **Ctrl+Shift+F5** (GUI) or **Ctrl+Shift+D** (Console)

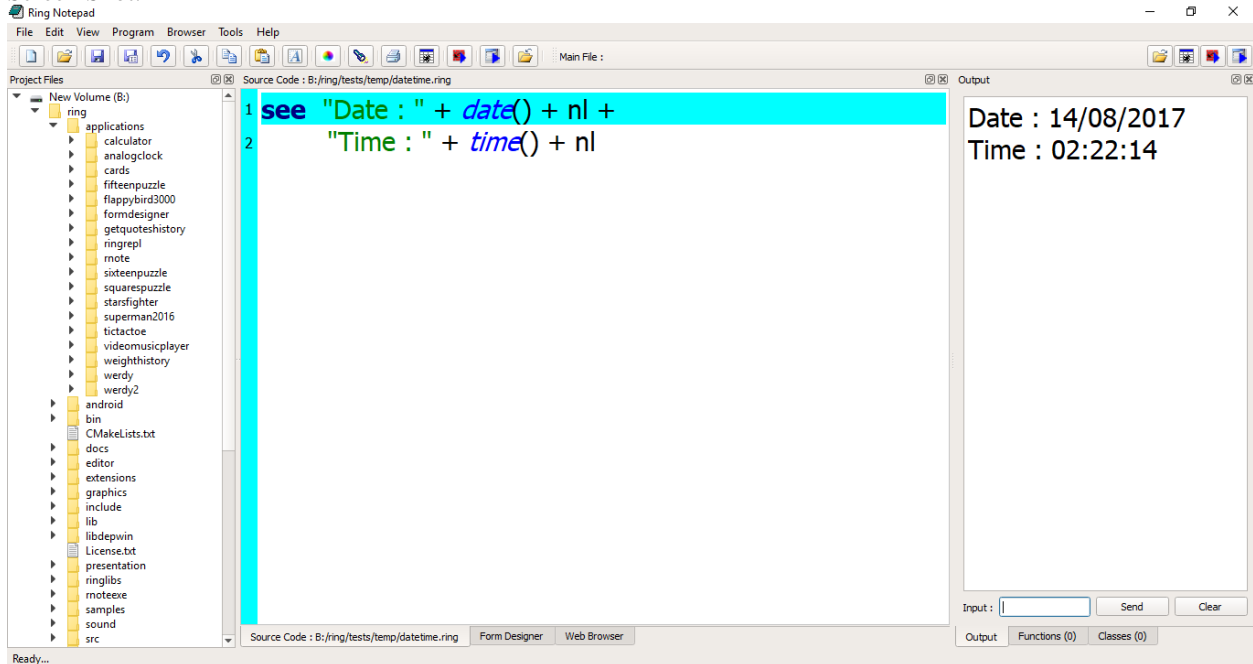
Screen Shots:





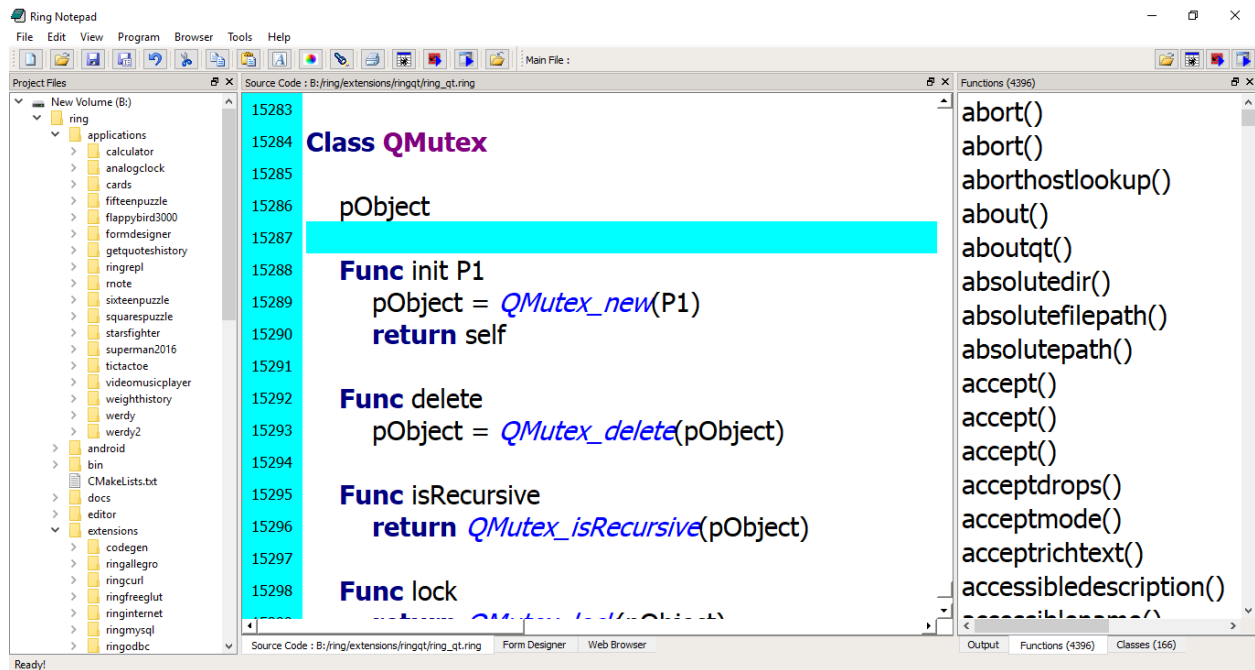
- The output window is updated to display the new lines correctly and contains the “Clear” button.

Screen Shot:



- The Ring Notepad is updated to quickly open and switch between large files while preparing the functions/classes lists in the background.

Screen Shot:



5.6 Better StdLib

New Functions

- Print2Str()
- ListAllFiles()
- SystemCmd()

1. The Print2Str() is a new function added to the StdLib

Example:

```
load "stdlib.ring"

world = "World!"
mystring = print2str("Hello, #{world} \nIn Year \n#{2000+17} \n")

see mystring + nl
```

Output:

```
Hello, World!
In Year
2017
```

2. The ListAllFiles() is a new function added to the StdLib

Using this function we can quickly do a process on a group of files in a folder and it's sub folders.

Example:

```
load "stdlib.ring"
aList = ListAllFiles("c:/ring/ringlibs", "ring") # *.ring only
```

```
aList = sort(aList)
see aList
```

Example:

```
load "stdlib.ring"
see listallfiles("b:/ring/ringlibs/weblib","") # All Files
```

3. The SystemCmd() is a new function added to the StdLib

The function will execute a system command like the System() function but will return the output in a string.

Example:

```
cYou = SystemCmd("whoami")
See "SystemCmd: whoami =====" + nl + cYou + nl
```

Output:

```
SystemCmd: whoami =====
desktop-umberto\umberto
```

5.7 Better WebLib

The WebLib is updated to include the HTMLPage class

Using this class we can create HTML documents without printing the output to the standard output

So instead of using the WebLib in Web Applications only

We can use it in Console/GUI/Mobile Applications too

Example:

```
load "stdlib.ring"
load "weblib.ring"

import System.Web

func main

    mypage = new HtmlPage {
        h1 { text("Customers Report") }
        Table
        {
            style = stylewidth("100%") + stylegradient(4)
            TR
            {
                TD { WIDTH="10%" text("Customers Count : " ) }
                TD { text (100) }
            }
        }
        Table
        {
            style = stylewidth("100%") + stylegradient(26)
            TR
            {
                style = stylewidth("100%") + stylegradient(24)
```

```

        TD { text("Name " ) }
        TD { text("Age" ) }
        TD { text("Country" ) }
        TD { text("Job" ) }
        TD { text("Company" ) }
    }
    for x = 1 to 100
    TR
    {
        TD { text("Test" ) }
        TD { text("30" ) }
        TD { text("Egypt" ) }
        TD { text("Sales" ) }
        TD { text("Future" ) }
    }
    next
}

write("report.html",mypage.output())

```

Using this feature we can create reports quickly using WebLib & GUILib together

Example:

```

load "stdlib.ring"
load "weblib.ring"
load "guilib.ring"

import System.Web
import System.GUI

new qApp {
    open_window(:CustomersReportController)
    exec()
}

class CustomersReportController

    oView = new CustomersReportView

    func Start
        CreateReport()

    func CreateReport
        mypage = new HtmlPage {
            h1 { text("Customers Report") }
            Table
            {
                style = stylewidth("100%") + stylegradient(4)
                TR
                {
                    TD { WIDTH="10%"
                        text("Customers Count : " ) }
                    TD { text (100) }
                }
            }
        }
        Table

```

```

        {
            style = stylewidth("100%") + stylegradient(26)
            TR
            {
                style = stylewidth("100%") +
                    stylegradient(24)
                TD { text("Name " ) }
                TD { text("Age" ) }
                TD { text("Country" ) }
                TD { text("Job" ) }
                TD { text("Company" ) }
            }
            for x = 1 to 100
                TR
                {
                    TD { text("Test" ) }
                    TD { text("30" ) }
                    TD { text("Egypt" ) }
                    TD { text("Sales" ) }
                    TD { text("Future" ) }
                }
            next
        }
    }
    write("report.html", mypage.output())

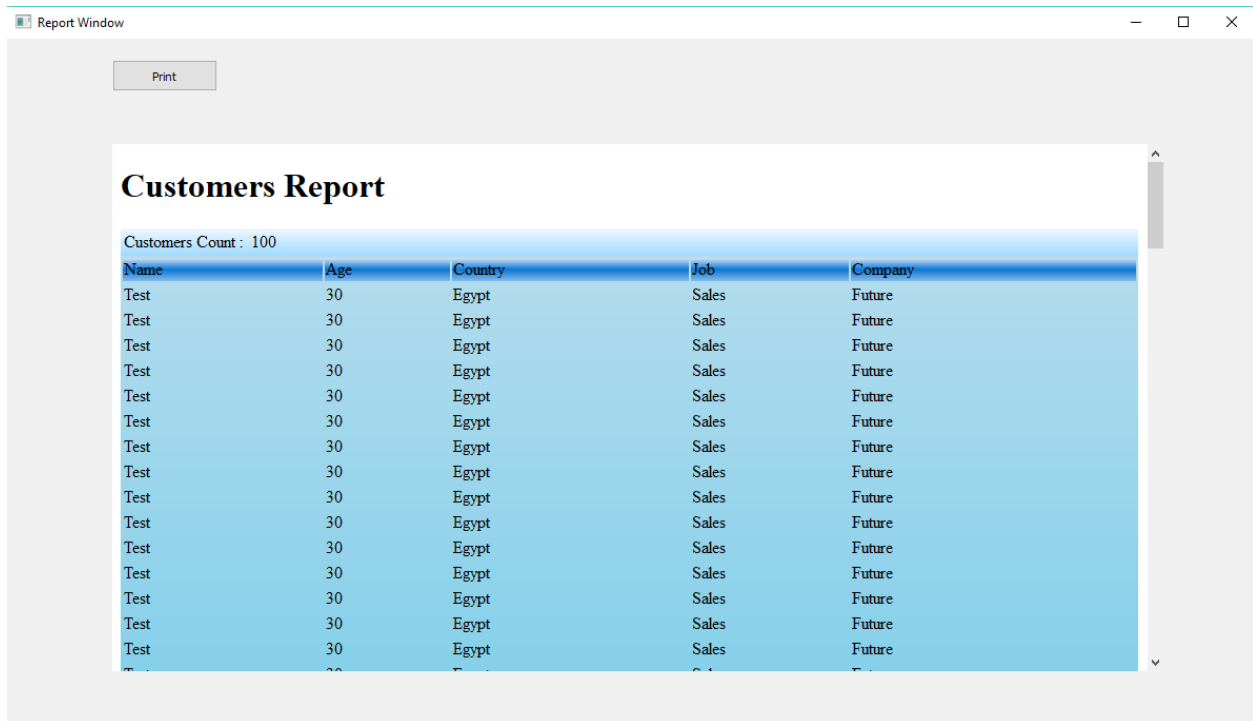
    func PrintEvent
        printer1 = new qPrinter(0) {
            setoutputformat(1)
            setoutputfilename("report.pdf")
        }
        oView {
            web.print(printer1)
            web.show()
        }
        system ("report.pdf")

class CustomersReportView

    win = new window() {
        setwindowtitle("Report Window")
        setgeometry(100,100,500,500)
        web = new webview(win) {
            setgeometry(100,100,1000,500)
            loadpage(new qurl("file:///"+
                currentdir()+"/report.html"))
        }
        new pushbutton(win) {
            setGeometry(100,20,100,30)
            setText("Print")
            setclickEvent(Method(:PrintEvent))
        }
        showMaximized()
    }
}

```

Screen Shot:



5.8 Better RingQt

New classes added to RingQt :

- QStringRef
- QMutex
- QMutexLocker
- QBuffer
- QBluetoothAddress
- QBluetoothDeviceDiscoveryAgent
- QBluetoothDeviceInfo
- QBluetoothHostInfo
- QBluetoothLocalDevice
- QBluetoothServer
- QBluetoothServiceDiscoveryAgent
- QBluetoothServiceInfo
- QBluetoothSocket
- QBluetoothTransferManager
- QBluetoothTransferReply
- QBluetoothTransferRequest
- QBluetoothUuid

Example:

```

### Submits your car VIN - Vehicle Id Number - to the Web Site - vpic.nhtsa.dot.gov -
### Parses XML data returned
### Prints out the car info result

load "libcurl.ring"
load "guilib.ring"
load "stdlib.ring"

curl = curl_easy_init()

# request = "3G1JC5248YS251015?format=xml"  ### VIN - Chevrolet
request = "3GYFK62847G247323?format=xml"  ### VIN - Cadillac

call_type = "decodevinvalues/"
url = "https://vpic.nhtsa.dot.gov/api/vehicles/"
url_request = url + call_type + request

    See "URL Request: " + url_request +nl

curl_easy_setopt(curl, CURLOPT_URL, url_request)
response = curl_easy_perform_silent(curl);

    See nl + "Response Raw: " + response +nl +nl

curl_easy_cleanup(curl)

xml = new qxmlstreamreader()
xml.adddata_2(response)

x = new qstringref()
while not xml.atend()
    if xml.error()
        see xml.errorstring() see nl
        exit loop

    ok

    x = xml.text()
    if not x.length() = 0
        see "Length: " see x.length() + " --- "
        see "Value: " see x.tostring() see nl

    ok

    xml.readnext()

end

get x

###-----
### Results
#
# ==>Value: 115
# ==>Value: Results returned successfully
# ==>Value: VIN(s): 3G1JC5248YS251015
# ==>Value: 3G1JC5248YS251015
# ==>Value: Sedan/Saloon
# ==>Value: 4
# ==>Value: 2200.0

```

```
# ==>Value: 134.25223700841
# ==>Value: 2.2
# ==>Value: 4
# ==>Value: LN2
# ==>Value: CHEVROLET
# ==>Value: GENERAL MOTORS LLC
# ==>Value: Cavalier
# ==>Value: 2000
# ==>Value: Ramos Arzipe
# ==>Value: PASSENGER CAR
# ==>Value: 4
# ==>Value: In-Line
# ==>Value: 1st Row (Driver & Passenger)
# ==>Value: Sequential Fuel Injection (SFI)
# ==>Value: Mexico
# ==>Value: NA
# ==>Value: Manual
# ==>Value: Body Type: Sedan, 4-6 Window, Notchback (GM codes: 19, 69)
# ==>Value: Name Plate: Chevrolet, Pontiac
# ==>Value: 0 - VIN decoded clean. Check Digit (9th position) is correct
# ==>Value: LAN
# ==>Value: 984
#
###-----
```

5.9 Better Objects Library

The function `Open_WindowInPackages()` is added to the Objects library.

The `Open_WindowInPackages()` function is the same as `Open_Window()` but takes an extra list that determine the packages to import before opening the window.

Syntax:

```
Open_WindowInPackages(cClassName,aPackagesList)
```

Example:

The next example from the Form Designer source code, Open the Window Flags window using the `open_windowInPackages()` function.

We determine the class name “WindowFlagsController” and the packages name.

The Window Flags window uses the FormDesigner and System.GUI packages.

```
open_windowInPackages(:WindowFlagsController,[
    "formdesigner",
    "System.GUI"
])
```

5.10 RingFreeGLUT Extension

Ring 1.5 comes with RingFreeGLUT extension to support the FreeGLUT library

Example:

```

/*
    This sample is based on C Tutorials
    from : http://www.lighthouse3d.com/tutorials/glut-tutorial/
*/

load "freeglut.ring"
load "opengl2llib.ring"

// angle of rotation for the camera direction
angle = 0.0

// actual vector representing the camera's direction
lx=0.0 lz=-1.0

// XZ position of the camera
x=0.0 z=5.0

// the key states. These variables will be zero
//when no key is being presses
deltaAngle = 0.0
deltaMove = 0
xOrigin = -1

// Constant definitions for Menus
C_RED = 1
C_GREEN = 2
C_BLUE = 3
C_ORANGE = 4

C_FILL = 5
C_LINE = 6

// Pop up menu identifiers
fillMenu=NULL
fontMenu=NULL
mainMenu=NULL
colorMenu=NULL

// color for the nose
red = 1.0
blue=0.5
green=0.5

// scale of snowman
scale = 1.0

// menu status
menuFlag = 0

// default font
font = GLUT_BITMAP_TIMES_ROMAN_24

C_INT_GLUT_BITMAP_8_BY_13 = 7
C_INT_GLUT_BITMAP_9_BY_15 = 8
C_INT_GLUT_BITMAP_TIMES_ROMAN_10 = 9
C_INT_GLUT_BITMAP_TIMES_ROMAN_24 = 10
C_INT_GLUT_BITMAP_HELVETICA_10 = 11

```

```

C_INT_GLUT_BITMAP_HELVETICA_12  = 12
C_INT_GLUT_BITMAP_HELVETICA_18  = 13

// width and height of the window
h = 0
w = 0

// variables to compute frames per second
frame=0
time=0
timebase=0
s = ""

func changeSize
    w = glutEventWidth()
    h = glutEventHeight()

    // Prevent a divide by zero, when window is too short
    // (you cant make a window of zero width).
    if h = 0
        h = 1

    ok

    ratio = w * 1.0 / h

    // Use the Projection Matrix
    glMatrixMode(GL_PROJECTION)

    // Reset Matrix
    glLoadIdentity()

    // Set the viewport to be the entire window
    glViewport(0, 0, w, h)

    // Set the correct perspective.
    gluPerspective(45.0, ratio, 0.1, 100.0)

    // Get Back to the Modelview
    glMatrixMode(GL_MODELVIEW)

func drawSnowMan

    glScalef(scale, scale, scale)
    glColor3f(1.0, 1.0, 1.0)

// Draw Body
    glTranslatef(0.0 ,0.75, 0.0)
    glutSolidSphere(0.75,20,20)

// Draw Head
    glTranslatef(0.0, 1.0, 0.0)
    glutSolidSphere(0.25,20,20)

// Draw Eyes
    glPushMatrix()
    glColor3f(0.0,0.0,0.0)
    glTranslatef(0.05, 0.10, 0.18)
    glutSolidSphere(0.05,10,10)

```

```

    glTranslatef(-0.1, 0.0, 0.0)
    glutSolidSphere(0.05,10,10)
    glPopMatrix()

// Draw Nose
    glColor3f(red, green, blue)
    glRotatef(0.0,1.0, 0.0, 0.0)
    glutSolidCone(0.08,0.5,10,2)

    glColor3f(1.0, 1.0, 1.0)

func renderBitmapString x,y,z,font,string
    glRasterPos3f(x, y,z)
    for c in string
        glutBitmapCharacter(font,ascii(c))
    next

func renderStrokeFontString x,y,z,font,string
    glPushMatrix()
    glTranslatef(x, y,z)
    glScalef(0.002, 0.002, 0.002)
    for c in string
        glutStrokeCharacter(font, Ascii(c));
    next
    glPopMatrix()

func restorePerspectiveProjection

    glMatrixMode(GL_PROJECTION)
    // restore previous projection matrix
    glPopMatrix()

    // get back to modelview mode
    glMatrixMode(GL_MODELVIEW)

func setOrthographicProjection

    // switch to projection mode
    glMatrixMode(GL_PROJECTION)

    // save previous matrix which contains the
    // settings for the perspective projection
    glPushMatrix()

    // reset matrix
    glLoadIdentity()

    // set a 2D orthographic projection
    gluOrtho2D(0, w, h, 0)

    // switch back to modelview mode
    glMatrixMode(GL_MODELVIEW)

func computePos deltaMove

```

```

    x += deltaMove * lx * 0.1
    z += deltaMove * lz * 0.1

func renderScene

    if deltaMove
        computePos(deltaMove)
    ok

    // Clear Color and Depth Buffers
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

    // Reset transformations
    glLoadIdentity()

    // Set the camera
    gluLookAt(
        x, 1.0, z,
        x+lx, 1.0, z+lz,
        0.0, 1.0, 0.0)

    // Draw ground

    glColor3f(0.9, 0.9, 0.9)
    glBegin(GL_QUADS)
        glVertex3f(-100.0, 0.0, -100.0)
        glVertex3f(-100.0, 0.0, 100.0)
        glVertex3f(100.0, 0.0, 100.0)
        glVertex3f(100.0, 0.0, -100.0)
    glEnd()

    // Draw 9 SnowMen
    for i = -3 to -1
        for j = -3 to -1
            glPushMatrix()
            glTranslatef(i*10.0, 0.0, j * 10.0)
            drawSnowMan()
            number = (i+3)*3+(j+3)
            renderBitmapString(0.0, 0.5, 0.0,font ,""+number)
            glPopMatrix()
        next
    next

    // Code to compute frames per second
    frame++

    time=glutGet(GLUT_ELAPSED_TIME)
    if time - timebase > 1000
        s = "RingFreeGLUT - FPS: " + (frame*1000.0/(time-timebase))
        timebase = time
        frame = 0
    ok

    // Code to display a string (fps) with bitmap fonts
    setOrthographicProjection()

    glPushMatrix()
    glLoadIdentity()

```

```

renderBitmapString(5,30,0,GLUT_BITMAP_HELVETICA_18,s)
glPopMatrix()

restorePerspectiveProjection()

glutSwapBuffers()

// -----
//          KEYBOARD
// -----

func processNormalKeys
    key = glutEventKey()
    xx = glutEventX()
    yy = glutEventY()

    switch key
        on 27
            glutDestroyMenu(mainMenu)
            glutDestroyMenu(fillMenu)
            glutDestroyMenu(colorMenu)
            glutDestroyMenu(fontMenu)
            Shutdown()
        off

func pressKey

    key = glutEventKey()
    xx = glutEventX()
    yy = glutEventY()

    switch key
        on GLUT_KEY_UP
            deltaMove = 0.5
        on GLUT_KEY_DOWN
            deltaMove = -0.5
        off

func releaseKey

    key = glutEventKey()

    switch key
        on GLUT_KEY_UP
            deltaMove = 0
        on GLUT_KEY_DOWN
            deltaMove = 0
        off

// -----
//          MOUSE
// -----

func mouseMove

```



```

xx = glutEventX()
yy = glutEventY()

// this will only be true when the left button is down
if xOrigin >= 0

    // update deltaAngle
    deltaAngle = (xx - xOrigin) * 0.001

    // update camera's direction
    lx = sin(angle + deltaAngle)
    lz = -cos(angle + deltaAngle)

ok

func mouseButton

    button = glutEventButton()
    state = glutEventState()
    xx = glutEventX()
    yy = glutEventY()

    // only start motion if the left button is pressed
    if button = GLUT_LEFT_BUTTON
        // when the button is released
        if state = GLUT_UP
            angle += deltaAngle
            xOrigin = -1
        else
            // state = GLUT_DOWN
            xOrigin = xx
        ok
    ok

// -----
//             MENUS
// -----

func processMenuStatus

    status = glutEventStatus()

    if status = GLUT_MENU_IN_USE
        menuFlag = 1
    else
        menuFlag = 0
    ok

func processMainMenu

    // nothing to do in here
    // all actions are for submenus

func processFillMenu

```

```

    option = glutEventValue()

    switch option
        on C_FILL
            glPolygonMode(GL_FRONT, GL_FILL)
        on C_LINE
            glPolygonMode(GL_FRONT, GL_LINE)
    off

func processFontMenu

    option = glutEventValue()

    switch (option) {
        on C_INT_GLUT_BITMAP_8_BY_13
            font = GLUT_BITMAP_8_BY_13
        on C_INT_GLUT_BITMAP_9_BY_15
            font = GLUT_BITMAP_9_BY_15
        on C_INT_GLUT_BITMAP_TIMES_ROMAN_10
            font = GLUT_BITMAP_TIMES_ROMAN_10
        on C_INT_GLUT_BITMAP_TIMES_ROMAN_24
            font = GLUT_BITMAP_TIMES_ROMAN_24
        on C_INT_GLUT_BITMAP_HELVETICA_10
            font = GLUT_BITMAP_HELVETICA_10
        on C_INT_GLUT_BITMAP_HELVETICA_12
            font = GLUT_BITMAP_HELVETICA_12
        on C_INT_GLUT_BITMAP_HELVETICA_18
            font = GLUT_BITMAP_HELVETICA_18
    off

func processColorMenu

    option = glutEventValue()

    switch option
        on C_RED
            red = 1.0
            green = 0.0
            blue = 0.0
        on C_GREEN
            red = 0.0
            green = 1.0
            blue = 0.0
        on C_BLUE
            red = 0.0
            green = 0.0
            blue = 1.0
        on C_ORANGE
            red = 1.0
            green = 0.5
            blue = 0.5
    off

func createPopupMenus

```

```

fontMenu = glutCreateMenu(:processFontMenu)

glutAddMenuEntry("BITMAP_8_BY_13 ",C_INT_GLUT_BITMAP_8_BY_13 )
glutAddMenuEntry("BITMAP_9_BY_15",C_INT_GLUT_BITMAP_9_BY_15 )
glutAddMenuEntry("BITMAP_TIMES_ROMAN_10 ",C_INT_GLUT_BITMAP_TIMES_ROMAN_10 )
glutAddMenuEntry("BITMAP_TIMES_ROMAN_24",C_INT_GLUT_BITMAP_TIMES_ROMAN_24 )
glutAddMenuEntry("BITMAP_HELVETICA_10 ",C_INT_GLUT_BITMAP_HELVETICA_10 )
glutAddMenuEntry("BITMAP_HELVETICA_12",C_INT_GLUT_BITMAP_HELVETICA_12 )
glutAddMenuEntry("BITMAP_HELVETICA_18",C_INT_GLUT_BITMAP_HELVETICA_18 )

fillMenu = glutCreateMenu(:processFillMenu)

glutAddMenuEntry("Fill",C_FILL)
glutAddMenuEntry("Line",C_LINE)

colorMenu = glutCreateMenu(:processColorMenu)
glutAddMenuEntry("Red",C_RED);
glutAddMenuEntry("Blue",C_BLUE);
glutAddMenuEntry("Green",C_GREEN);
glutAddMenuEntry("Orange",C_ORANGE);

mainMenu = glutCreateMenu(:processMainMenu)

glutAddSubMenu("Polygon Mode", fillMenu)
glutAddSubMenu("Color", colorMenu)
glutAddSubMenu("Font",fontMenu)
// attach the menu to the right button
glutAttachMenu(GLUT_RIGHT_BUTTON)

// this will allow us to know if the menu is active
glutMenuStatusFunc(:processMenuStatus)

// -----
//          MAIN
// -----

func main

    // init GLUT and create window
    glutInit()
    glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA)
    glutInitWindowPosition(100,100)
    glutInitWindowSize(320,320)
    glutCreateWindow("RingFreeGLUT - Test - 9 SnowMan")

    // register callbacks
    glutDisplayFunc(:renderScene)
    glutReshapeFunc(:changeSize)
    glutIdleFunc(:renderScene)

    glutIgnoreKeyRepeat(1)
    glutKeyboardFunc(:processNormalKeys)
    glutSpecialFunc(:pressKey)
    glutSpecialUpFunc(:releaseKey)

    // here are the two new functions
    glutMouseFunc(:mouseButton)

```

```

glutMotionFunc (:mouseMove)

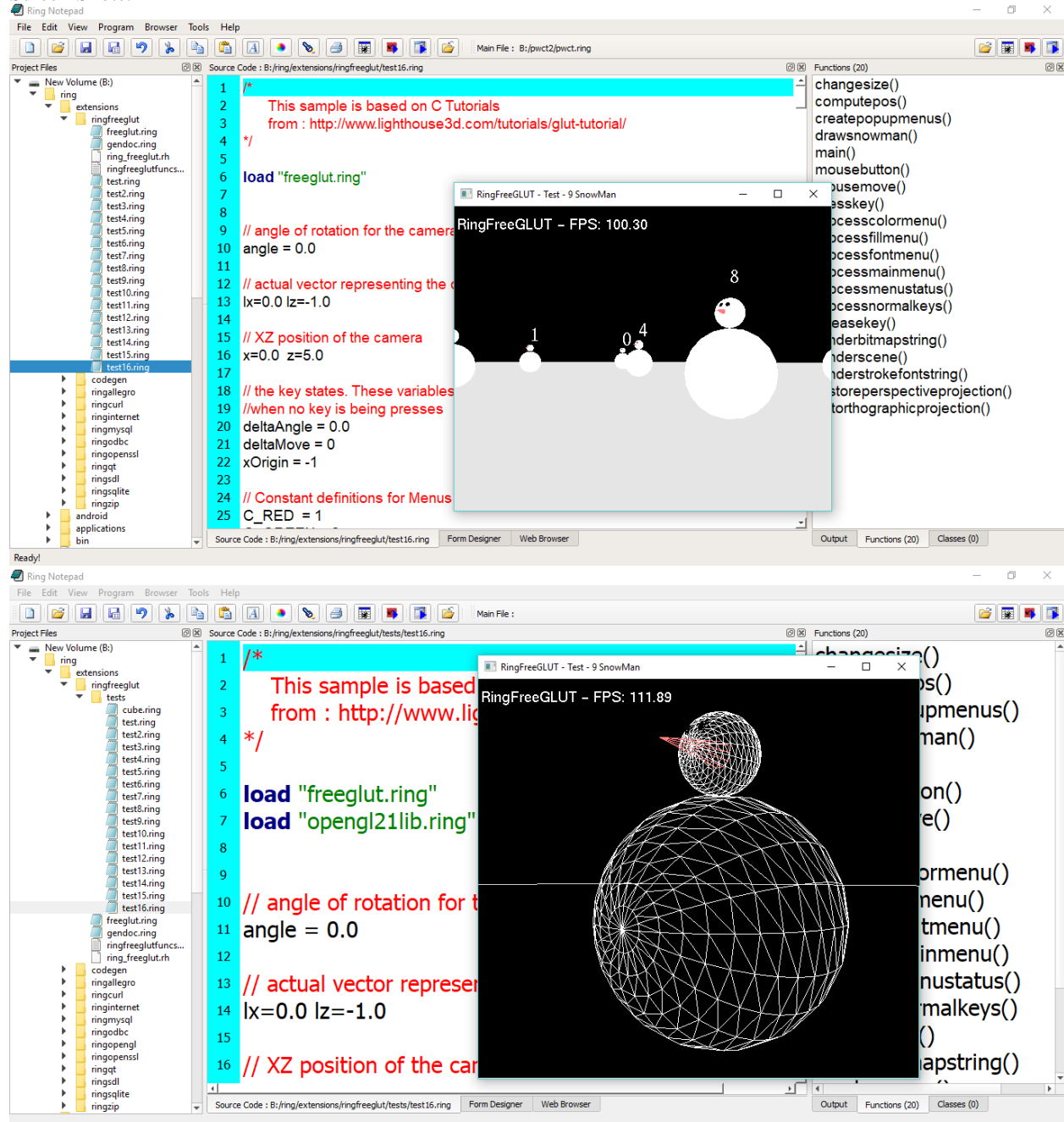
// OpenGL init
glEnable(GL_DEPTH_TEST)
glEnable(GL_CULL_FACE)

// init Menus
createPopupMenu()

// enter GLUT event processing cycle
glutMainLoop()

```

Screen Shots:



5.11 RingOpenGL Extension

Ring 1.5 comes with RingOpenGL and support for the next versions

- OpenGL 1.1
- OpenGL 1.2
- OpenGL 1.3
- OpenGL 1.4
- OpenGL 1.5
- OpenGL 2.0
- OpenGL 2.1
- OpenGL 3.0
- OpenGL 3.2
- OpenGL 3.3
- OpenGL 4.0
- OpenGL 4.1
- OpenGL 4.2
- OpenGL 4.3
- OpenGL 4.4
- OpenGL 4.5
- OpenGL 4.6

Example:

```
/*
    This sample is based on C Tutorials
    from :
           http://www.wikihow.com/Make-a-Cube-in-OpenGL
*/

load "freeglut.ring"
load "opengl2llib.ring"

// -----
// Global Variables
// -----
rotate_y=0
rotate_x=0

// -----
// display() Callback function
// -----
func display

    // Clear screen and Z-buffer
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)

    // Reset transformations
```

```

glLoadIdentity()

// Rotate when user changes rotate_x and rotate_y
glRotatef( rotate_x, 1.0, 0.0, 0.0 )
glRotatef( rotate_y, 0.0, 1.0, 0.0 )

//Multi-colored side - FRONT
glBegin(GL_POLYGON)

glColor3f( 1.0, 0.0, 0.0 )    glVertex3f( 0.5, -0.5, -0.5 )    # P1 is red
glColor3f( 0.0, 1.0, 0.0 )    glVertex3f( 0.5, 0.5, -0.5 )    # P2 is green
glColor3f( 0.0, 0.0, 1.0 )    glVertex3f( -0.5, 0.5, -0.5 )    # P3 is blue
glColor3f( 1.0, 0.0, 1.0 )    glVertex3f( -0.5, -0.5, -0.5 )    # P4 is purple

glEnd()

// White side - BACK
glBegin(GL_POLYGON)
glColor3f( 1.0, 1.0, 1.0 )
glVertex3f( 0.5, -0.5, 0.5 )
glVertex3f( 0.5, 0.5, 0.5 )
glVertex3f( -0.5, 0.5, 0.5 )
glVertex3f( -0.5, -0.5, 0.5 )
glEnd()

// Purple side - RIGHT
glBegin(GL_POLYGON)
glColor3f( 1.0, 0.0, 1.0 )
glVertex3f( 0.5, -0.5, -0.5 )
glVertex3f( 0.5, 0.5, -0.5 )
glVertex3f( 0.5, 0.5, 0.5 )
glVertex3f( 0.5, -0.5, 0.5 )
glEnd()

// Green side - LEFT
glBegin(GL_POLYGON)
glColor3f( 0.0, 1.0, 0.0 )
glVertex3f( -0.5, -0.5, 0.5 )
glVertex3f( -0.5, 0.5, 0.5 )
glVertex3f( -0.5, 0.5, -0.5 )
glVertex3f( -0.5, -0.5, -0.5 )
glEnd()

// Blue side - TOP
glBegin(GL_POLYGON)
glColor3f( 0.0, 0.0, 1.0 )
glVertex3f( 0.5, 0.5, 0.5 )
glVertex3f( 0.5, 0.5, -0.5 )
glVertex3f( -0.5, 0.5, -0.5 )
glVertex3f( -0.5, 0.5, 0.5 )
glEnd()

// Red side - BOTTOM
glBegin(GL_POLYGON)
glColor3f( 1.0, 0.0, 0.0 )
glVertex3f( 0.5, -0.5, -0.5 )
glVertex3f( 0.5, -0.5, 0.5 )
glVertex3f( -0.5, -0.5, 0.5 )

```

```

glVertex3f( -0.5, -0.5, -0.5 )
glEnd()

glFlush()
glutSwapBuffers()

// -----
// specialKeys() Callback Function
// -----
func specialKeys

    key = glutEventKey()

    // Right arrow - increase rotation by 5 degree
    switch Key

    on GLUT_KEY_RIGHT
        rotate_y += 5

    // Left arrow - decrease rotation by 5 degree
    on GLUT_KEY_LEFT
        rotate_y -= 5

    on GLUT_KEY_UP
        rotate_x += 5

    on GLUT_KEY_DOWN
        rotate_x -= 5

    off

    // Request display update
    glutPostRedisplay()

// -----
// main() function
// -----
func main

    // Initialize GLUT and process user parameters
    glutInit()

    // Request double buffered true color window with Z-buffer
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH)

    // Create window
    glutCreateWindow("Awesome Cube")

    // Enable Z-buffer depth test
    glEnable(GL_DEPTH_TEST)

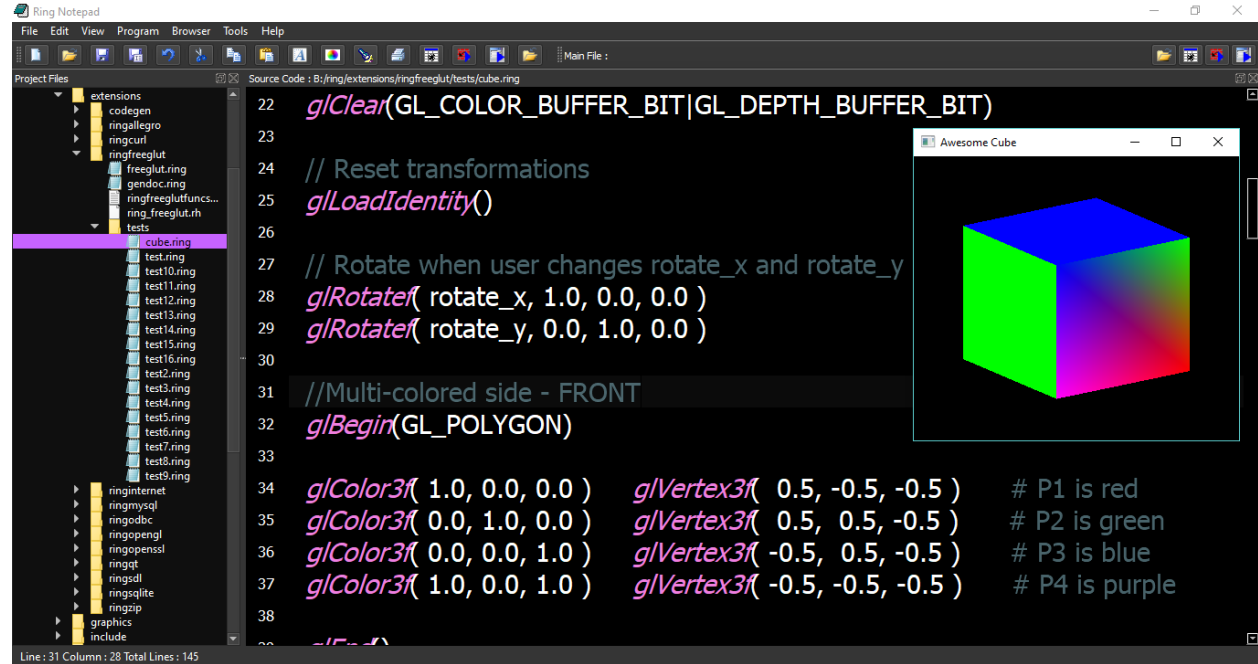
    // Callback functions
    glutDisplayFunc(:display)
    glutSpecialFunc(:specialKeys)

```

```
// Pass control to GLUT for events
glutMainLoop()

// Return to OS
```

Screen Shot:



5.12 Better Code Generator for Extensions

The Code Generator is updated to support <constant> type, So we can have constants other than numbers, for example : Strings and Pointers.

When we have pointers we can determine the pointer type. To use this feature, before <constant> and </constant> we can use

```
$nDefaultConstantType = C_CONSTANT_TYPE_POINTER
$cDefaultConstantPointerType = "void *"
```

The next example from the RingFreeGLUT extension

```
<runcode>
    $nDefaultConstantType = C_CONSTANT_TYPE_POINTER
    $cDefaultConstantPointerType = "void"
</runcode>
<constant>
    GLUT_STROKE_ROMAN
    GLUT_STROKE_MONO_ROMAN
    GLUT_BITMAP_9_BY_15
    GLUT_BITMAP_8_BY_13
    GLUT_BITMAP_TIMES_ROMAN_10
    GLUT_BITMAP_TIMES_ROMAN_24
    GLUT_BITMAP_HELVETICA_10
    GLUT_BITMAP_HELVETICA_12
```



```

    GLUT_BITMAP_HELVETICA_18
</constant>

```

5.13 Better Documentation Generator for Extensions

The documentation generator for extensions is updated to generate a list of constants in the generated documentation. The previous versions provides the functions prototype only, Now we have the list of constants too.

5.14 Ring VM - Tracing Functions

In Ring 1.5 the next functions are added to Ring VM

- RingVM_SetTrace(cCode)
- RingVM_TraceData() -> aDataList
- RingVM_TraceEvent() -> nTraceEvent
- RingVM_TraceFunc() -> cCode
- RingVM_ScopesCount() -> nScopes
- RingVM_EvalInScope(nScope,cCode)
- RingVM_PassError()
- RingVM_HideErrorMsg(lStatus)
- RingVM_CallFunc(cFuncName)

Example:

```

load "tracelib.ring"

ringvm_settrace("mytrace() ")

see "Hello, world!" + nl
see "Welcome" + nl
see "How are you?" +nl
mytest()
new myclass { mymethod() }

func mytest
    see "Message from mytest" + nl

func mytrace
    see "==== The Trace function is Active =====" + nl +
        "Trace Function Name : " + ringvm_TraceFunc() + nl +
        "Trace Event : "
    switch ringvm_TraceEvent()
        on TRACEEVENT_NEWLINE          see "New Line"
        on TRACEEVENT_NEWFUNC          see "New Function"
        on TRACEEVENT_RETURN            see "Return"
        on TRACEEVENT_ERROR             see "Error"
        on TRACEEVENT_BEFORECFUNC       see "Before C Function"
        on TRACEEVENT_AFTERCFUNC        see "After C Function"
    off

```

```

    see nl +
        "Line Number : " + ringvm_tracedata() [TRACEDATA_LINENUMBER] + nl +
        "File Name    : " + ringvm_tracedata() [TRACEDATA_FILENAME] + nl +
        "Function Name : " + ringvm_tracedata() [TRACEDATA_FUNCNAME] + nl +
        "Method or Function : "
        if ringvm_tracedata() [TRACEDATA_METHODORFUNC] =
            TRACEDATA_METHODORFUNC_METHOD
            see "Method"
        else
            if ringvm_tracedata() [TRACEDATA_FUNCNAME] = NULL
                see "Command"
            else
                see "Function"
        ok
    ok
    see nl + Copy("=",42) + nl

class myclass
    func mymethod
        see "Message from mymethod" + nl

```

Output:

```

===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : After C Function
Line Number : 3
File Name    : test1.ring
Function Name : ringvm_settrace
Method or Function : Function
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 5
File Name    : test1.ring
Function Name :
Method or Function : Command
=====
Hello, world!
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 6
File Name    : test1.ring
Function Name :
Method or Function : Command
=====
Welcome
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 7
File Name    : test1.ring
Function Name :
Method or Function : Command
=====
How are you?

```

```

===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 8
File Name : test1.ring
Function Name :
Method or Function : Command
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : New Function
Line Number : 8
File Name : test1.ring
Function Name : mytest
Method or Function : Function
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 12
File Name : test1.ring
Function Name : mytest
Method or Function : Function
=====
Message from mytest
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 14
File Name : test1.ring
Function Name : mytest
Method or Function : Function
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : Return
Line Number : 8
File Name : test1.ring
Function Name :
Method or Function : Command
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 9
File Name : test1.ring
Function Name :
Method or Function : Command
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 43
File Name : test1.ring
Function Name :
Method or Function : Command
=====
===== The Trace function is Active =====

```

```

Trace Function Name : mytrace()
Trace Event : Before C Function
Line Number : 9
File Name : test1.ring
Function Name : ismethod
Method or Function : Function
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : After C Function
Line Number : 9
File Name : test1.ring
Function Name : ismethod
Method or Function : Function
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : New Function
Line Number : 9
File Name : test1.ring
Function Name : mymethod
Method or Function : Method
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 44
File Name : test1.ring
Function Name : mymethod
Method or Function : Method
=====
Message from mymethod
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : Return
Line Number : 9
File Name : test1.ring
Function Name :
Method or Function : Command
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : Before C Function
Line Number : 9
File Name : test1.ring
Function Name : ismethod
Method or Function : Function
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : After C Function
Line Number : 9
File Name : test1.ring
Function Name : ismethod
Method or Function : Function
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()

```

```

Trace Event : Before C Function
Line Number : 9
File Name   : test1.ring
Function Name : ismethod
Method or Function : Function
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : After C Function
Line Number : 9
File Name   : test1.ring
Function Name : ismethod
Method or Function : Function
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 11
File Name   : test1.ring
Function Name :
Method or Function : Command
=====

```

5.15 Trace Library and Interactive Debugger

Ring 1.5 comes with the Trace Library and the Interactive Debugger

Using this library we can trace events, execute programs line by line, open the Interactive Debugger when an error happens or at breakpoints.

Example:

The next example uses a Breakpoint to open the Interactive Debugger!

```

load "tracelib.ring"

test1()

func test1
    x = 10
    see :test1 + nl
    t = 12
    BreakPoint()
    see "After breakpoint!" +nl
    see "t = " + t + nl
    see "End of program!" + nl

```

Screen Shots:

We have the Interactive Debugger at the Breakpoint!

The screenshot shows the Ring IDE with a source code window on the left and an interactive debugger window on the right. The source code window displays the following code:

```

5 # BreakPoint
6
7 load "tracelib"
8
9 test1()
10
11 func test1
12     x = 1
13     see :Command (Exit)
14     t = 1
15     Break
16     see "Command (LocalsData)"
17     see "Command (Globals)"
18     see "We can execute Ring code"
19
20

```

The interactive debugger window shows the following commands and their descriptions:

```

=====
Interactive Debugger
=====
Command (Exit)      : End Program
Command (Cont)      : Continue Execution
Command (Locals)     : Print local variables names
Command (LocalsData) : Print local variables data
Command (Globals)    : Print global variables names
We can execute Ring code
=====
code:>

```

We can print the variables values

The screenshot shows the Ring IDE with a source code window on the left and an interactive debugger window on the right. The source code window displays the following code:

```

6
7 load "tracelib.ring"
8
9 test1()
10
11 func test1
12     x = 10
13     see :test1 + nl
14     t = 12
15     BreakPoint()
16     see "After breakpoint!" +nl
17     see "t = " + t + nl
18     see "End of program!" + nl
19
20

```

The interactive debugger window shows the following commands and their descriptions:

```

=====
Command (Exit)      : End Program
Command (Cont)      : Continue Execution
Command (Locals)     : Print local variables names
Command (LocalsData) : Print local variables data
Command (Globals)    : Print global variables names
We can execute Ring code
=====
code> localsdata

```

Variable	Type	Value
x	NUMBER	10
t	NUMBER	12

code> _

We can change the variables values then continue execution

The screenshot shows the Ring IDE with a source code window on the left and an interactive debugger window on the right. The source code window displays the following code:

```

5 # BreakPoint
6
7 load "tracelib.ring"
8
9 test1()
10
11 func test1
12     x = 10
13     see :test1 + nl
14     t = 12
15     BreakPoint()
16     see "After breakpoint!" +nl
17     see "t = " + t + nl
18     see "End of program!" + nl
19
20

```

The interactive debugger window shows the following commands and their descriptions:

```

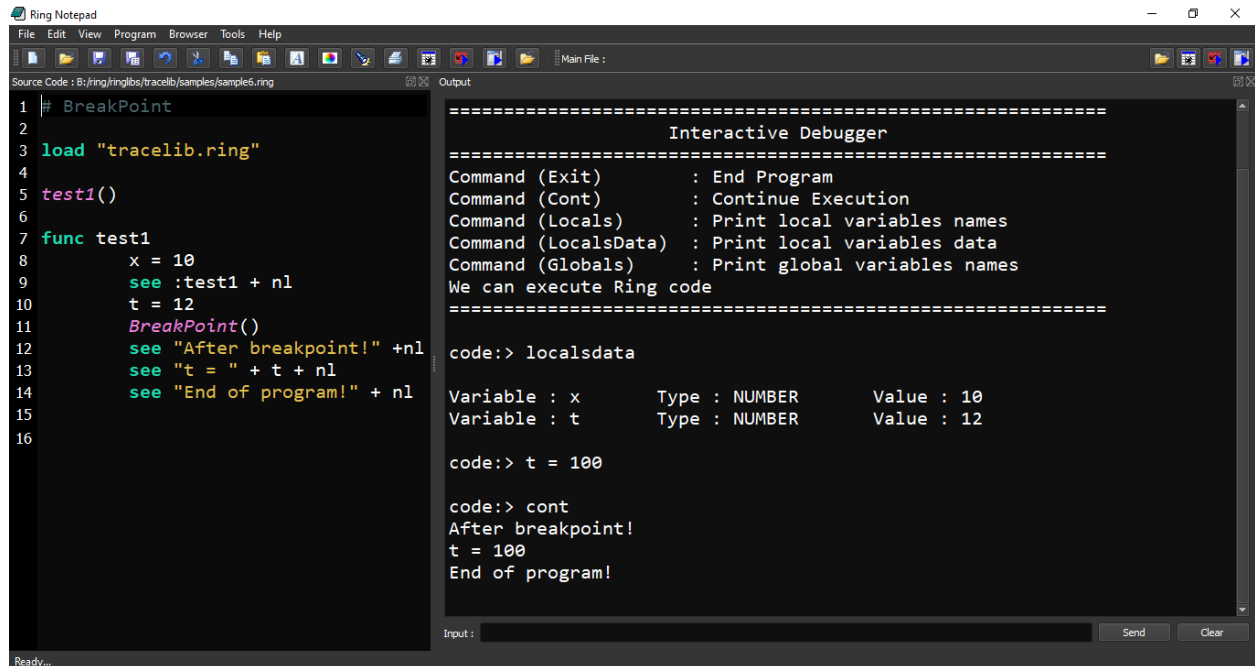
code> localsdata

```

Variable	Type	Value
x	NUMBER	10
t	NUMBER	12

code> x = 100
code> t = 200
code> cont
After breakpoint!
t = 200
End of program!

We can run the Interactive Debugger in the Output Window



5.16 More Syntax Flexibility

- Using braces { } in Packages/Classes/Functions

Example:

```

load "stdlib.ring"

import mypackage

new myclass {
    myfunc()
}

package mypackage
{
    class myclass
    {
        func myfunc
        {
            print("Hello, World!\n")
        }
    }
}

```

- Using 'end' keyword after Packages/Classes/Functions

Example:

```

import mypackage

new myclass {
    myfunc()
}

```

```

package mypackage
  class myclass
    def myfunc
      put "Hello, World!"
    end
  end
end

```

- Using ‘endpackage’/‘endclass’/‘endfunc’ keywords after Packages/Classes/Functions

Example:

```

import mypackage

new myclass { myfunc() }

package mypackage
  class myclass
    func myfunc
      see "welcome" + nl
    endfunc
  endclass
endpackage

```

5.17 Type Hints Library

Ring 1.5 comes with the Type Hints library

Using this library we can add the type information to the source code which will be very useful for tools like

- Code Editors
- Static-Analysis

Example:

```

load "typehints.ring"

see sum(3,4) + nl ;
see sayHello("Mahmoud");

int func sum(int x,int y) {
  return x+y ;
}

string func sayHello(string name) {
  return "Hello " + name ;
}

```

The library is very powerful and support the User types (Classes) automatically!

Example:

```

load "typehints.ring"

import mypackage

test() { main([:one,:two,:three]) }

```



```
myclass func test() {
    see "Testing User Types!" + nl
    return new myclass
}

package mypackage {
    public class myclass {
        public static void func main(list args) {
            see "welcome" + nl
            see args
        }
    }
}
```

Also You can use the types inside the code (not only the function prototype)

Example:

```
load "typehints.ring"

int    sum = sum(3,4)
string msg = sayHello("Mahmoud")

see "Sum = " + sum + nl + msg + nl

int func sum(int x,int y) {
    return x+y ;
}

string func sayHello(string name) {
    return "Hello " + name ;
}
```

Rules:

- To use the types in the function prototype, You must use ‘(‘ and ‘)’ around parameters
- To use the types in the function code, You must set the variable value (Assignment).

Note: Ring is a dynamic language, No type checking will be done by the compiler.

5.18 Better Quality

Based on Ring usage every day in practical projects

Ring 1.5 is more stable and more productive!

We are adding features based on clear vision and practical needs.

Also the documentation is better.

5.19 What is new in Ring 1.5.1?

- Better Documentation

- StdLib - Factorial() function update
- RingVM - Better code for clearing the stack in the Class Region.
- Sample : 3D Cube (OpenGL) + Texture Image using GameLib (RingAllegro)

Source Code:

```
load "gamelib.ring"
load "opengl21lib.ring"

func main

    new GraphicsApp {
        start()
    }

class GraphicsApp from GraphicsAppBase

    TITLE = "Ring Cube"

    bitmap texture

    xrot = 0.0
    yrot = 0.0
    zrot = 0.0

    func loadresources

        bitmap = al_load_bitmap("ring.bmp")
        texture = al_get_opengl_texture(bitmap)

    func destroyResources

        al_destroy_bitmap(bitmap)

    func drawScene

        w = 800 h = 600
        ratio = w / h

        glViewport(0, 0, w, h)
        glMatrixMode(GL_PROJECTION)
        glLoadIdentity()

        gluPerspective(45, ratio, 1, 100)
        glMatrixMode(GL_MODELVIEW)
        glLoadIdentity()

        glEnable(GL_TEXTURE_2D)
        glShadeModel(GL_SMOOTH)
        glClearColor(0.0, 0.0, 0.0, 0.5)
        glClearDepth(1.0)
        glEnable(GL_DEPTH_TEST)
        glEnable(GL_CULL_FACE)
        glDepthFunc(GL_LEQUAL)
        glHint(GL_PERSPECTIVE_CORRECTION_HINT, GL_NICEST)

        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
```

```

glLoadIdentity();
glTranslatef(0.0,0.0,-5.0);

glRotatef(xrot,1.0,0.0,0.0);
glRotatef(yrot,0.0,1.0,0.0);
glRotatef(zrot,0.0,0.0,1.0);

glBindTexture(GL_TEXTURE_2D, texture)

glBegin(GL_QUADS)
    // Front Face
    glTexCoord2f(0.0, 0.0) glVertex3f(-1.0, -1.0, 1.0)
    glTexCoord2f(1.0, 0.0) glVertex3f( 1.0, -1.0, 1.0)
    glTexCoord2f(1.0, 1.0) glVertex3f( 1.0,  1.0, 1.0)
    glTexCoord2f(0.0, 1.0) glVertex3f(-1.0,  1.0, 1.0)
    // Back Face
    glTexCoord2f(1.0, 0.0) glVertex3f(-1.0, -1.0, -1.0)
    glTexCoord2f(1.0, 1.0) glVertex3f(-1.0,  1.0, -1.0)
    glTexCoord2f(0.0, 1.0) glVertex3f( 1.0,  1.0, -1.0)
    glTexCoord2f(0.0, 0.0) glVertex3f( 1.0, -1.0, -1.0)
    // Top Face
    glTexCoord2f(0.0, 1.0) glVertex3f(-1.0,  1.0, -1.0)
    glTexCoord2f(0.0, 0.0) glVertex3f(-1.0,  1.0,  1.0)
    glTexCoord2f(1.0, 0.0) glVertex3f( 1.0,  1.0,  1.0)
    glTexCoord2f(1.0, 1.0) glVertex3f( 1.0,  1.0, -1.0)
    // Bottom Face
    glTexCoord2f(1.0, 1.0) glVertex3f(-1.0, -1.0, -1.0)
    glTexCoord2f(0.0, 1.0) glVertex3f( 1.0, -1.0, -1.0)
    glTexCoord2f(0.0, 0.0) glVertex3f( 1.0, -1.0,  1.0)
    glTexCoord2f(1.0, 0.0) glVertex3f(-1.0, -1.0,  1.0)
    // Right face
    glTexCoord2f(1.0, 0.0) glVertex3f( 1.0, -1.0, -1.0)
    glTexCoord2f(1.0, 1.0) glVertex3f( 1.0,  1.0, -1.0)
    glTexCoord2f(0.0, 1.0) glVertex3f( 1.0,  1.0,  1.0)
    glTexCoord2f(0.0, 0.0) glVertex3f( 1.0, -1.0,  1.0)
    // Left Face
    glTexCoord2f(0.0, 0.0) glVertex3f(-1.0, -1.0, -1.0)
    glTexCoord2f(1.0, 0.0) glVertex3f(-1.0, -1.0,  1.0)
    glTexCoord2f(1.0, 1.0) glVertex3f(-1.0,  1.0,  1.0)
    glTexCoord2f(0.0, 1.0) glVertex3f(-1.0,  1.0, -1.0)
glEnd()

xrot += 0.3
yrot += 0.2
zrot += 0.4

```

```
class GraphicsAppBase
```

```

    display event_queue ev timeout
    timer  redraw      = true

    FPS                      = 60

    SCREEN_W                = 800
    SCREEN_H                = 600

    KEY_UP                  = 1

```

```

KEY_DOWN      = 2
KEY_LEFT      = 3
KEY_RIGHT     = 4

Key = [false, false, false, false]

TITLE = "Graphics Application"

func start
    SetUp()
    loadResources()
    eventsLoop()
    destroy()

func setup
    al_init()
    al_init_image_addon()
    al_set_new_display_flags(ALLEGRO_OPENGL)
    display = al_create_display(SCREEN_W, SCREEN_H)
    al_set_window_title(display, TITLE)
    al_clear_to_color(al_map_rgb(0, 0, 0))
    event_queue = al_create_event_queue()
    al_register_event_source(event_queue,
        al_get_display_event_source(display))
    ev = al_new_allegro_event()
    timeout = al_new_allegro_timeout()
    al_init_timeout(timeout, 0.06)
    timer = al_create_timer(1.0 / FPS)
    al_register_event_source(event_queue,
        al_get_timer_event_source(timer))
    al_start_timer(timer)
    al_install_mouse()
    al_register_event_source(event_queue,
        al_get_mouse_event_source())
    al_install_keyboard()
    al_register_event_source(event_queue,
        al_get_keyboard_event_source())

func eventsLoop
    while true
        al_wait_for_event_until(event_queue, ev, timeout)
        switch al_get_allegro_event_type(ev)
        on ALLEGRO_EVENT_DISPLAY_CLOSE
            exit
        on ALLEGRO_EVENT_TIMER
            redraw = true
        on ALLEGRO_EVENT_MOUSE_AXES
            mouse_x = al_get_allegro_event_mouse_x(ev)
            mouse_y = al_get_allegro_event_mouse_y(ev)
        on ALLEGRO_EVENT_MOUSE_ENTER_DISPLAY
            mouse_x = al_get_allegro_event_mouse_x(ev)
            mouse_y = al_get_allegro_event_mouse_y(ev)
        on ALLEGRO_EVENT_MOUSE_BUTTON_UP
            exit
        on ALLEGRO_EVENT_KEY_DOWN

```

```

        switch al_get_allegro_event_keyboard_keycode(ev)
            on ALLEGRO_KEY_UP
                key[KEY_UP] = true
            on ALLEGRO_KEY_DOWN
                key[KEY_DOWN] = true
            on ALLEGRO_KEY_LEFT
                key[KEY_LEFT] = true
            on ALLEGRO_KEY_RIGHT
                key[KEY_RIGHT] = true

        off
    on ALLEGRO_EVENT_KEY_UP
        switch al_get_allegro_event_keyboard_keycode(ev)
            on ALLEGRO_KEY_UP
                key[KEY_UP] = false
            on ALLEGRO_KEY_DOWN
                key[KEY_DOWN] = false
            on ALLEGRO_KEY_LEFT
                key[KEY_LEFT] = false
            on ALLEGRO_KEY_RIGHT
                key[KEY_RIGHT] = false
            on ALLEGRO_KEY_ESCAPE
                exit

        off
    off
    if redraw and al_is_event_queue_empty(event_queue)
        redraw = false
        drawScene()
        al_flip_display()
    ok
    callgc()

end

func destroy

    destroyResources()
    al_destroy_timer(timer)
    al_destroy_allegro_event(ev)
    al_destroy_allegro_timeout(timeout)
    al_destroy_event_queue(event_queue)
    al_destroy_display(display)

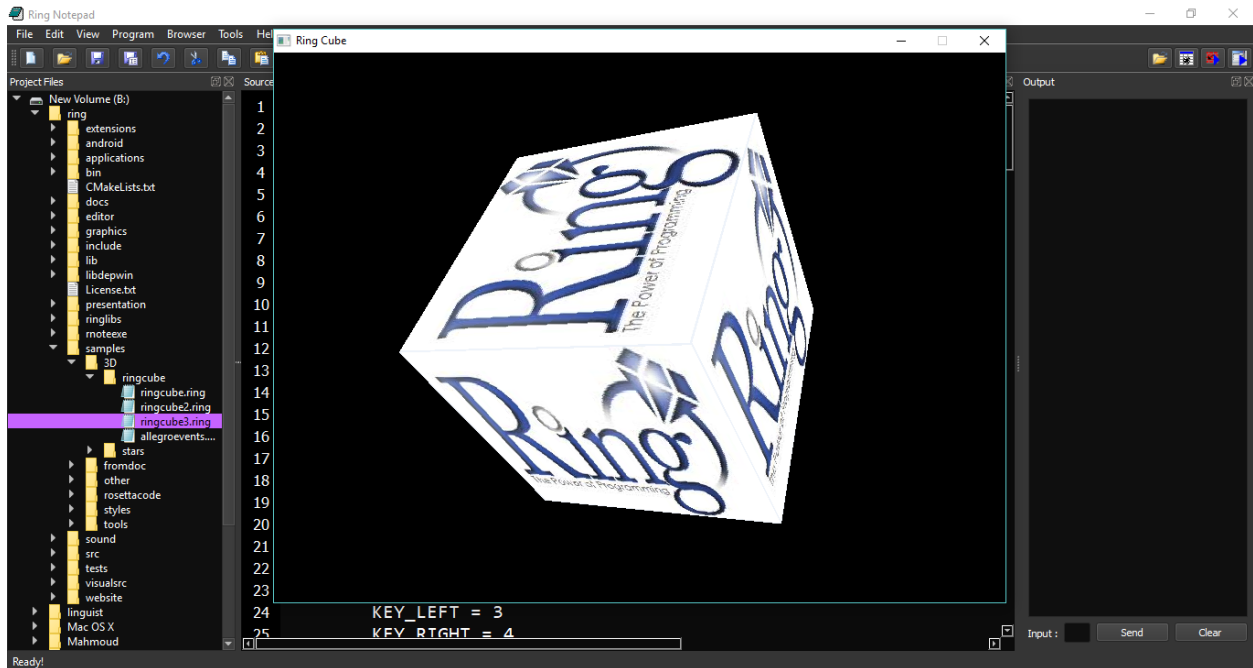
func loadresources

func drawScene

func destroyResources

```

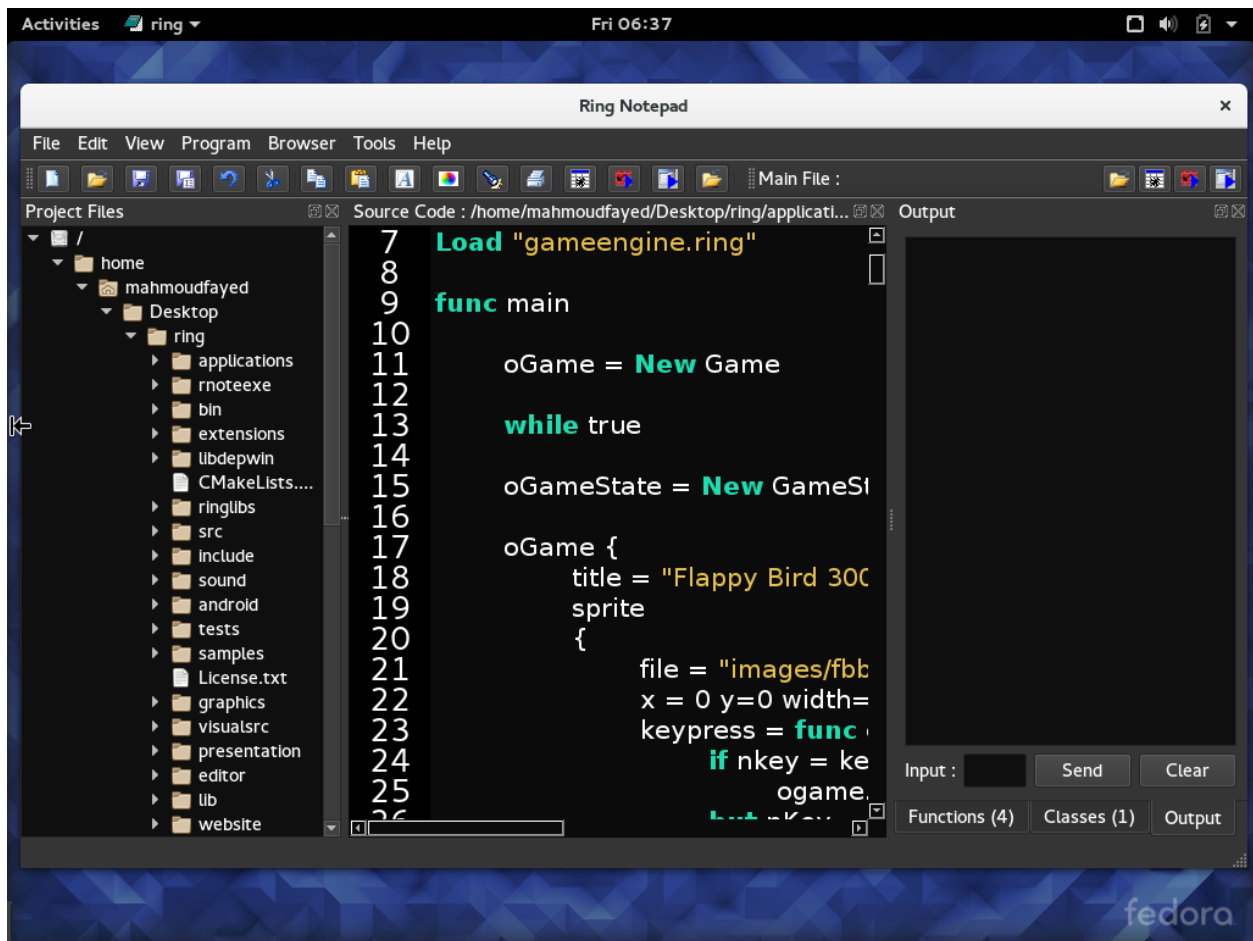
Screen Shot:



5.20 What is new in Ring 1.5.2?

- Documentation - Chapter “Applications developed in little hours” is updated
- Ring Notepad - Display programs output in the output window on all platforms
- Form Designer - Help Menu - Open CHM/PDF files without displaying the console window
- Form Designer - Better response to Resize/Move Events when moving the Mouse quickly
- Form Designer - New/Open/Save As, will open the Controller class in Ring Notepad
- Form Designer - Added “Close Form” option to the file menu
- Ring Notepad - Run, will save the current file (Also the opened Form) automatically
- GetQuotesHistory Application - Updated to work on MacOS X and Qt 5.2
- Calculator Application - Updated to include more features!
- RingVM - Classification for Environment Errors (Check Chapter : Language Reference)
- RingQt - New methods added to QAllEvents for faster Events execution
- RingQt - Fusion Black Style - Better colors for disabled controls
- Scripts - For building Ring on Fedora Linux (Check Chapter : Building From Source Code)

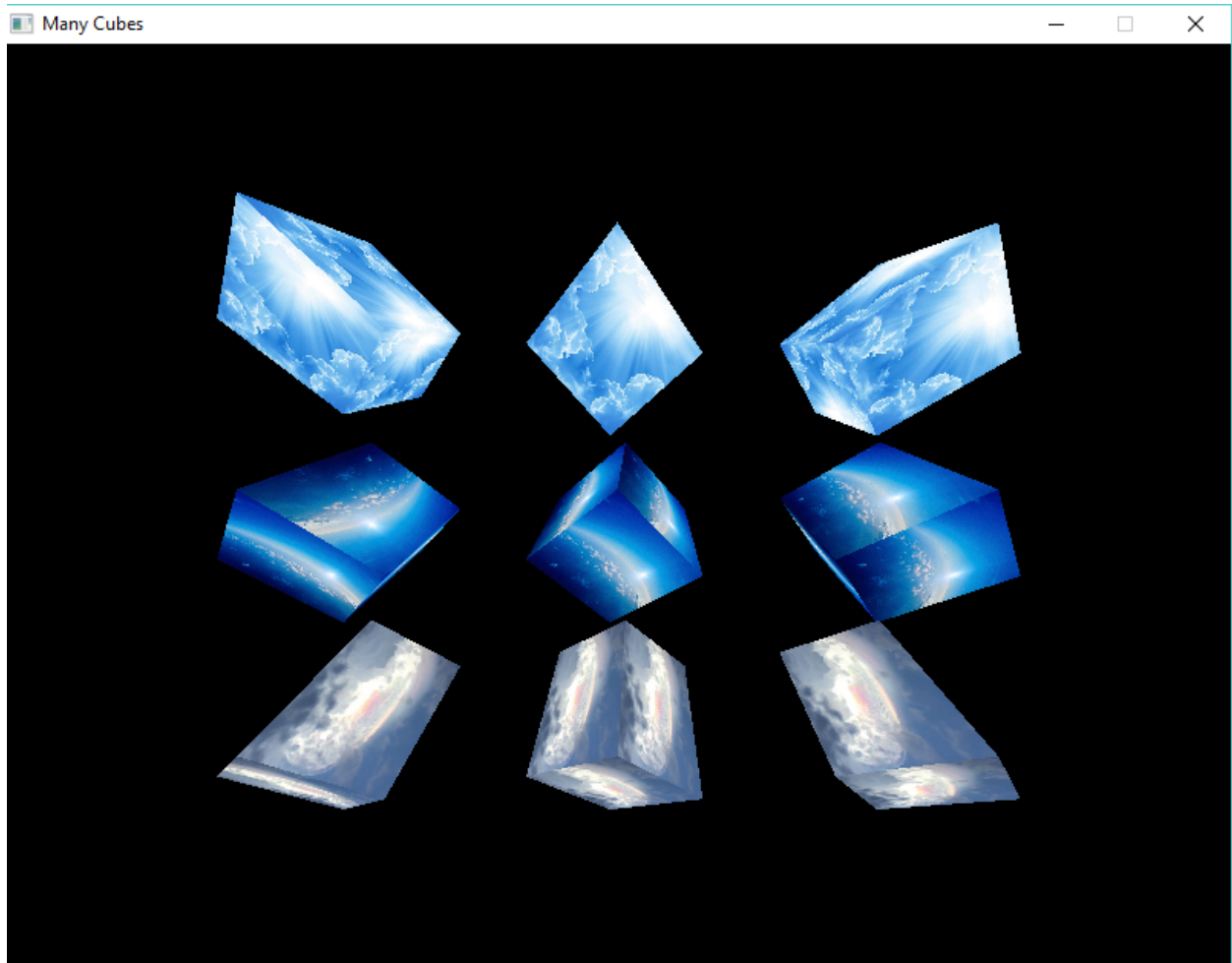
Screen Shot:



5.21 What is new in Ring 1.5.3?

- Form Designer : Close Action will notify Ring Notepad to be able to open the Form again
- Form Designer : Save Action will open the controller class in Ring Notepad
- Form Designer : Keep current control selected when selecting many controls using CTRL Key
- Form Designer : Nice form back color when used in Ring Notepad (Style: Modern Black)
- RingOpenSSL : Updated to support newer versions like OpenSSL 1.1
- Building Scripts : Updated to work on Fedora 26 (64bit)
- OpenGL : New Sample - Many Cubes (samples/3D/manycubes)

Screen Shot:



- RingQt : Add QDateTime Class
- RingQt : New methods added to QMenu and QCursor Classes

Example:

```
load "guilib.ring"

new QApplication {
    win = new QWidget() {
        setWindowTitle("Context Menu")
        resize(400,400)
        myfilter = new QAllEvents(win) {
            setContextMenuEvent("myMenu()")
        }
        installEventFilter(myfilter)
        show()
    }
    exec()
}

func myMenu

    new QMenu(win) {
```



```

        oAction = new QAction(win) {
            settext("new")
            SetCLickevent("See :New")
        }
        addaction(oAction)
        oAction = new QAction(win) {
            settext("open")
            SetCLickevent("See :Open")
        }
        addaction(oAction)
        oAction = new QAction(win) {
            settext("save")
            SetCLickevent("See :Save")
        }
        addaction(oAction)
        oAction = new QAction(win) {
            settext("close")
            SetCLickevent("See :Close")
        }
        addaction(oAction)
        oCursor = new qCursor()
        exec(oCursor.pos())
    }

```

- Compiler : Support using _ in numbers

Example:

```

x = 1_000_000
see type(x)+nl
see x+1+nl

```

Output:

```

NUMBER
1000000001

```

- Compiler : Support using f after numbers

Example:

```

x = 19.99f
see type(x) + nl

```

Output:

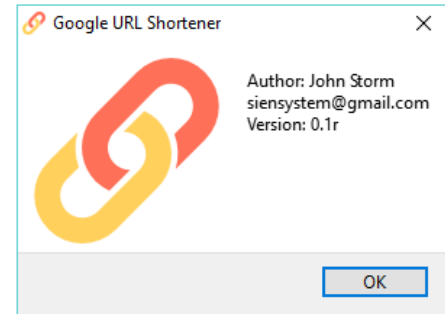
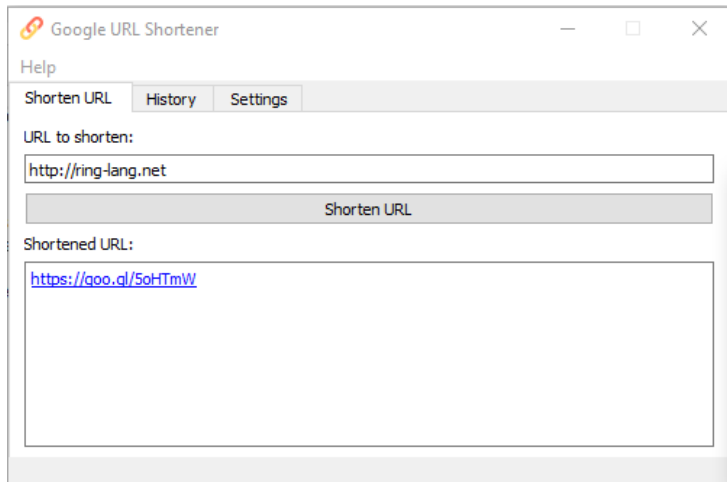
```

NUMBER

```

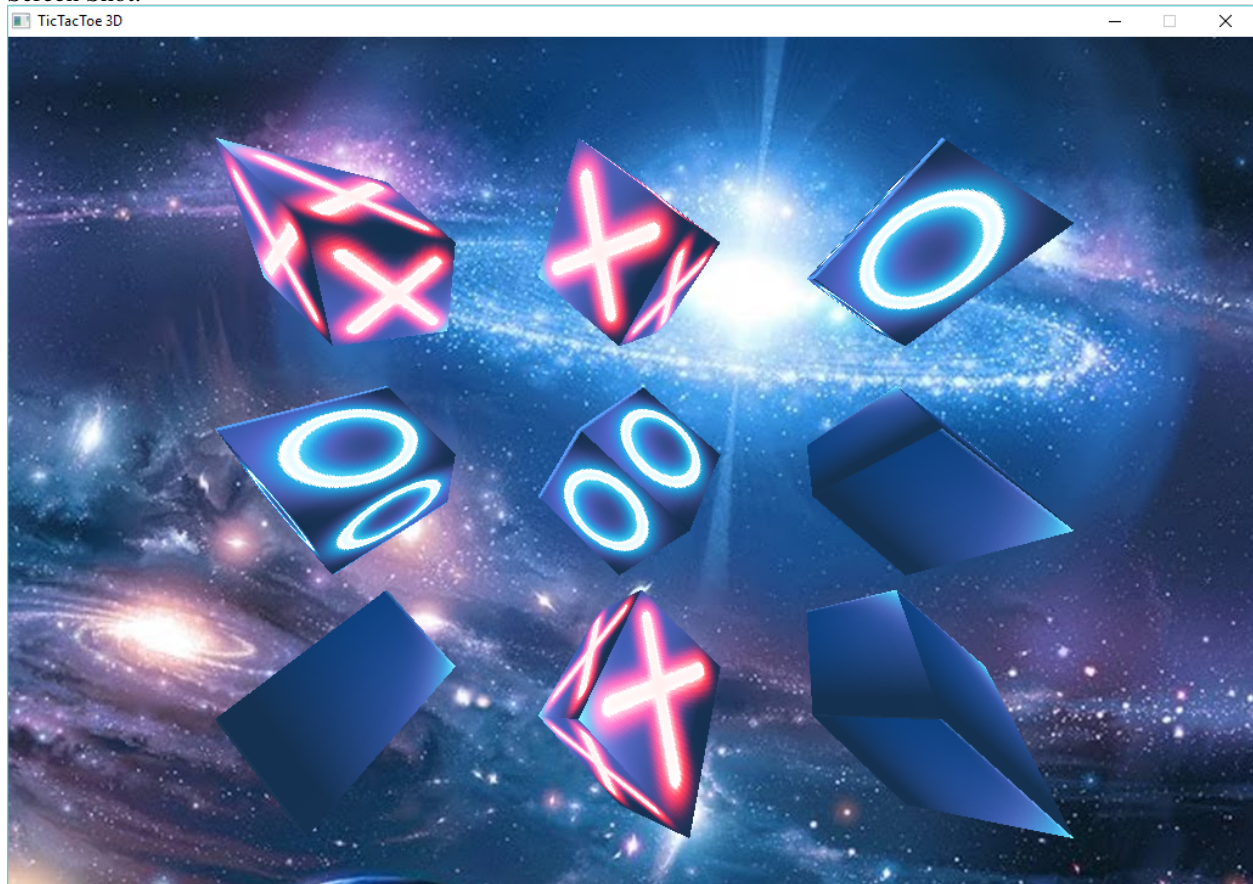
- Google API Shortener Application

Screen Shots:



- TicTacToe 3D Game

Screen Shot:



5.22 What is new in Ring 1.5.4?

- CalmoSoft Fifteen Puzzle Game 3D
- Ring Notepad - New Styles

- Ring Notepad - Better Toolbar Style
- Ring Notepad - View Modes
- Ring Notepad - QTextEdit - don't set back color for the scroll bars
- Ring Notepad - Style Fusion (White) - use Silver color for comments
- Ring Notepad - Tab and Shift-Tab - Indent multiple lines
- Form Designer - Better Toolbar Style
- Form Designer - Nice bgcolor for Window Flags and Menubar Designer
- Form Designer - Default back color for controls
- RingQt - Added grab() and windowHandle() methods to QWidget class
- RingQt - Added new methods to QPixmap Class
- **RingQt - Added Classes :-**
 - QScreen
 - QWindow
 - QGuiApplication
 - QTextBrowser
- Code Generator for Extensions - Nonew Option - Support Parent Class
- Ring VM - Internal Implementation - Pass state to Strings and Lists objects
- Ring VM - Garbage Collector - Memory Pool for Small Objects
- Ring VM - Better code for Saving/Restoring the State

WHAT IS NEW IN RING 1.4?

In this chapter we will learn about the changes and new features in Ring 1.4 release.

6.1 List of changes and new features

Ring 1.4 comes with many new features

- Change: Basic Extensions are separated from RingVM
- The Natural Library
- New Style is added to Ring Notepad
- RingREPL
- Convert between Numbers and Bytes
- Better StdLib
- Better WebLib
- Better RingQt
- Qt Class Convertor

6.2 Change: Basic Extensions are separated from RingVM

In Ring 1.4 the next libraries are separated from RingVM

- RingODBC
- RingMySQL
- RingSQLite
- RingOpenSSL
- RingInternet

To use these libraries, Use the Load command.

```
load "odbc.lib.ring"  
# use ODBC Functions
```

```
load "mysql.lib.ring"  
# use MySQL Functions
```

```
load "sqlitelib.ring"
# use SQLite Functions
```

```
load "openssllib.ring"
# use OpenSSL Functions ( Hash and Security functions)
```

```
load "internetlib.ring"
# use Internet Functions ( Download() and SendEmail() )
```

If you will use all of these libraries, You can just use stdlib.ring And the stdlib.ring will load odbclib.ring, mysqllob.ring, sqlitelib.ring, openssllib.ring and internetlib.ring files.

```
load "stdlib.ring"
```

6.3 The Natural Library

Ring 1.4 comes with the Natural Library to quickly define a language that contains a group of commands.

We will write the natural code in a Text file, for example program.txt

File: program.txt

```
Welcome to the Ring programming language!
What you are reading now is not comments, I swear!

After many years of programming I decided to think different about
programming and solve the problems in a better way.

We are writing commands or code and the Ring language is reading
it to understand us! Sure, What you are seeing now is
just ***part of the code - Not the Complete Program***
You have to write little things before and after this
part to be able to run it!

It is the natural part of our code where we can write in English,
Arabic or any Natural Language Then we will tell the computer
through the Ring language what must happens! in a way that we can scale
for large frameworks and programs.

Just imagine what will happens to the world of programming once
we create many powerful frameworks using the Ring language that
uses this way (Natural Programming).

For example When we say Hello to the Machine, It can reply! and when we
say count from 1 to 5 it will understand us, Also if
we said count from 5 to 1 it will
understand us too! You can see the Output window!

This Goal is not new, but the Ring language comes
with an innovative solution to this problem.
```

Output:

```
Hello, Sir!
```

```
The Numbers!
```

```

1
2
3
4
5
I will count Again!
5
4
3
2
1

```

To execute the natural code, We have start.ring

In start.ring we define the language and the commands.

File: start.ring

```

load "stdlib.ring"
load "naturallib.ring"

New NaturalLanguage {
    SetLanguageName(:MyLanguage)
    SetCommandsPath(CurrentDir()+"/../command")
    SetPackageName("MyLanguage.Natural")
    UseCommand(:Hello)
    UseCommand(:Count)
    RunFile("program.txt")
}

```

We defined a language called MyLanguage, We have folder for the language commands.

Each command will define a class that belong to the MyLanguage.Natural package.

We will define two commands, Hello and Count.

So we must have two files for defining the commands in the CurrentDir()+"/../command" folder

File: hello.ring

```

DefineNaturalCommand.SyntaxIsKeyword([
    :Package = "MyLanguage.Natural",
    :Keyword = :hello,
    :Function = func {
        See "Hello, Sir!" + nl + nl
    }
])

```

File: count.ring

```

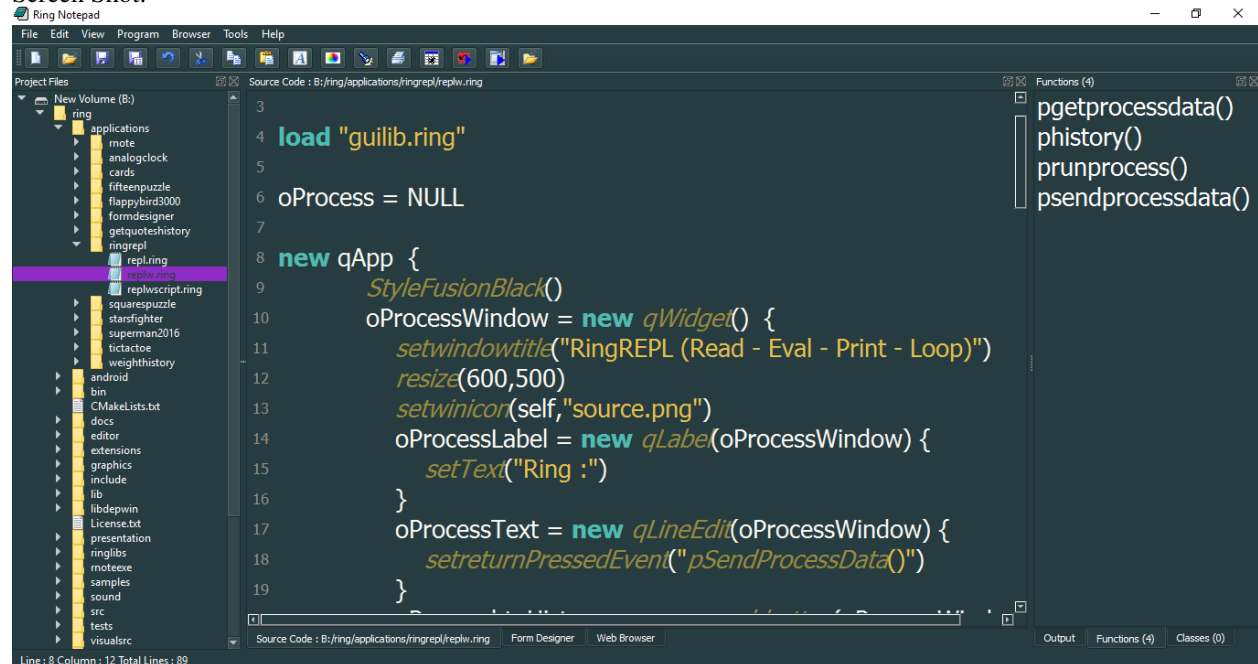
DefineNaturalCommand.SyntaxIsKeywordNumberNumber([
    :Package = "MyLanguage.Natural",
    :Keyword = :count,
    :Function = func {
        if not isattribute(self, :count_times) {
            AddAttribute(self, :count_times)
            Count_Times = 0
        }
        if Expr(1) > Expr(2) {
            nStep = -1
        }
        else
            nStep = 1
        }
        if Count_Times = 0 {
            see nl+"The Numbers!" + nl
            Count_Times++
        }
        else
            see nl + "I will count Again!" +nl
        }
        for x = Expr(1) to Expr(2) step nStep {
            see nl+x+nl
        }
        CommandReturn(fabs(Expr(1)-Expr(2))+1)
    }
])

```

6.4 New Style is added to Ring Notepad

In Ring Notepad - From View - Styles - Select the (Modern) Style

Screen Shot:

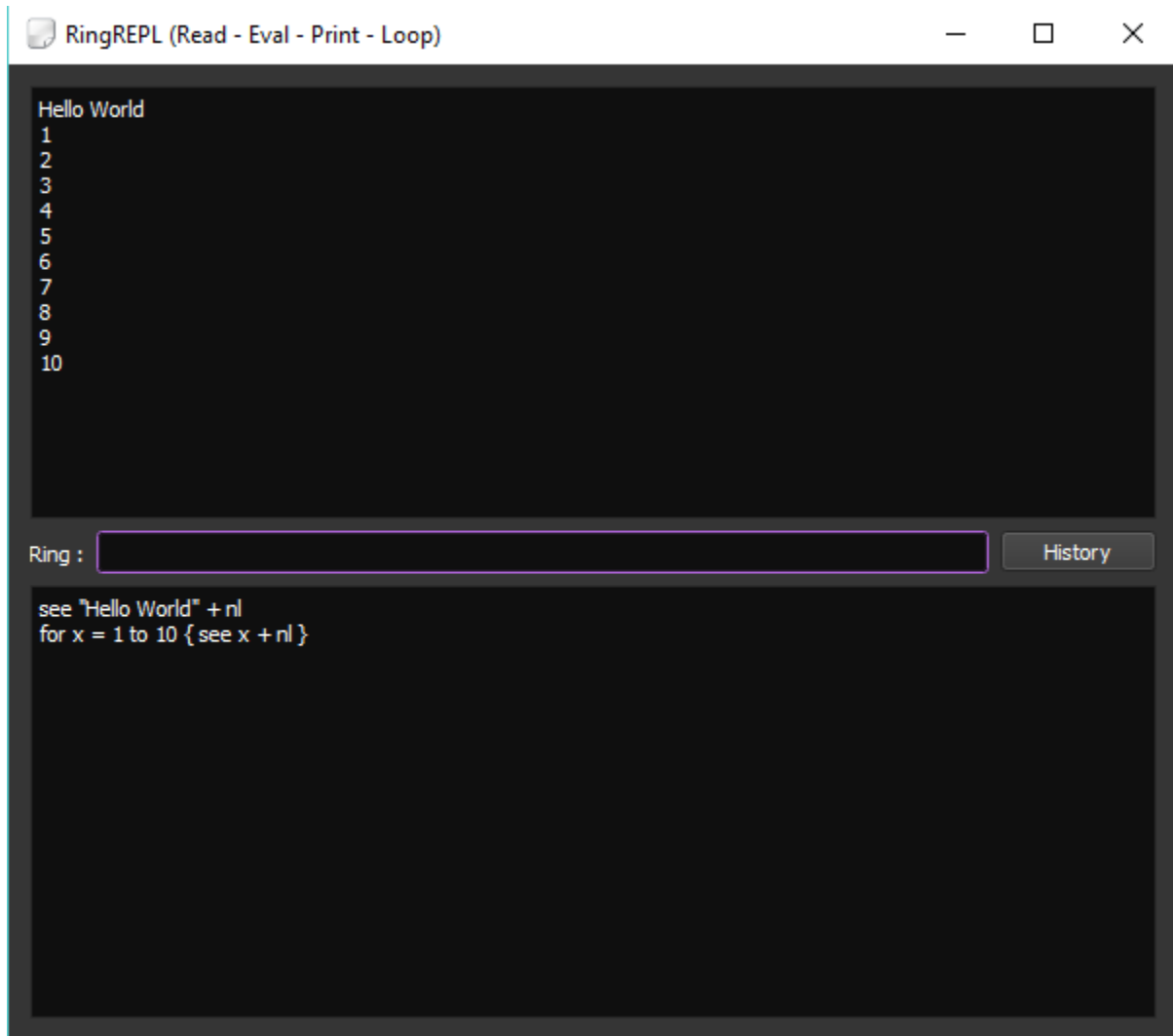


6.5 RingREPL

In the application folder, You will find RingREPL (Read-Eval-Print-Loop)

Also you can run it from Ring Notepad (Menubar - Tools)

Screen Shot:



6.6 Convert between Numbers and Bytes

Ring 1.4 comes with the next functions to convert between Numbers and Bytes.

- Int2Bytes()
- Float2Bytes()
- Double2Bytes()
- Bytes2Int()
- Bytes2Float()

- Bytes2Double()

Example:

```
see "Test Int2Bytes() and Bytes2Int() - Value : 77" + nl
r = Int2Bytes(77)
see "Int Size : " + len(r) + nl
see r + nl
see Bytes2Int(r) + nl
see "Test Float2Bytes() and Bytes2Float() - Value 77.12" + nl
r = Float2Bytes(77.12)
see "Float Size : " + len(r) + nl
see r + nl
see Bytes2Float(r) + nl
see "Test Double2Bytes() and Bytes2Double() - Value 9999977.12345" + nl
r = Double2Bytes(9999977.12345)
see "Double Size : " + len(r) + nl
see r + nl
decimals(5)
see Bytes2Double(r) + nl
```

6.7 Better StdLib

The StdLib is updated to include the next functions

- FSize()

The print() function is updated to accept local variables.

```
load "stdlib.ring"

func main
    print("Enter your name : ")      ;
    Name = getString()              ;
    print( "Hello : #{Name} " )      ;
    return                          ;
```

6.8 Better WebLib

The web library is updated

- Provide better error message
 1. Error (WebLib-1) : REQUEST_METHOD is empty ! - Run this script from the browser
 2. Error (DataLib-1) : Can't connect to the database server!
- Better Template() function - can accept NULL instead of object as the second paramter.

```
html(template("main.rhtml", NULL))
```

- The Form Class is updated to support the “target” attribute.

```
BootstrapWebPage()
{
    Title = "The Ring Programming Language"
    html(template("main.rhtml", NULL))
}
```

```

div {
    classname = :container
    div
    {
        id = "div3"
        color = "black"
        backgroundcolor = "white"
        width = "100%"
        form
        {
            method = "POST"
            Action = website
            Target = "codeoutput"
            input { type="hidden" name="page" value=1 }
            Table
            {
                style = stylewidth("100%") +
                    stylegradient(3)
                TR
                {
                    TD { align="center"
                        WIDTH="10%"
                        text("Code :")
                    }
                    TD {
                        html(`
                        <textarea name = "cCode"
                        rows="5"
                        style="width : 100%; ">
                        See "Hello, World!" + nl
                        </textarea>`)
                    }
                }
            }
            Input { type = "submit"
                classname="btn btn-primary btn-block"
                value = "Execute" }
            Table
            {
                style = stylewidth("100%") +
                    stylegradient(34)
                TR
                {
                    TD { align="center"
                        WIDTH="10%"
                        text("Output :")
                    }
                    TD {
                        html(`
                        <iframe name="codeoutput"
                        width="100%"
                        style="background-color:white;">
                        </iframe>`)
                    }
                }
            }
        }
    }
}

```

```
        }
    }

    }
    html(template("footer.rhtml", NULL))
}
```

6.9 Better RingQt

The next functions are added to RingQt

- SetDialogIcon(cIconFile)
- MsgInfo(cTitle,cMessage)
- ConfirmMsg(cTitle,cMessage)
- InputBox(cTitle,cMessage)
- InputBoxInt(cTitle,cMessage)
- InputBoxNum(cTitle,cMessage)
- InputBoxPass(cTitle,cMessage)

The next classes are added to RingQt

- QToolButton
- QSerialPort
- QSerialPortInfo

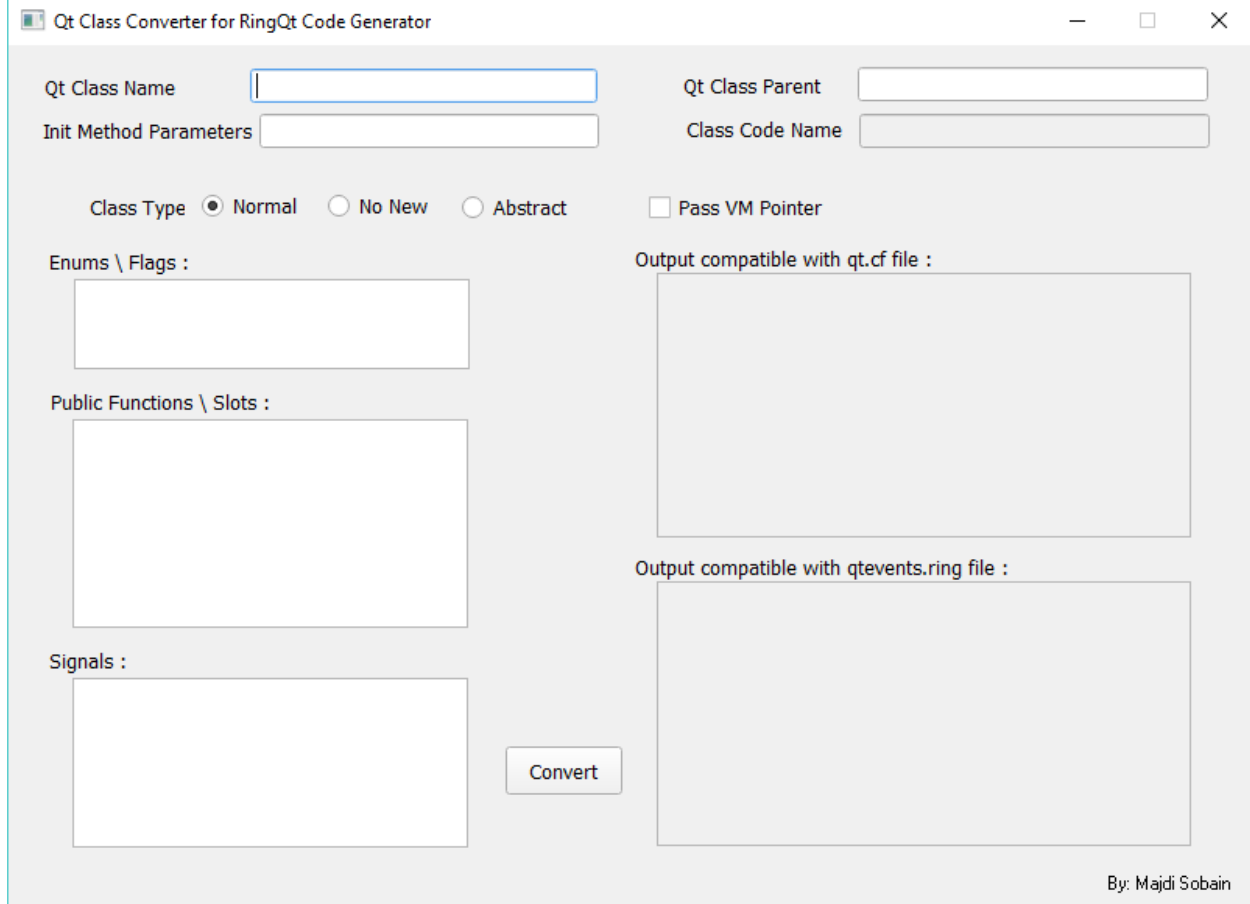
6.10 Qt Class Convertor

Ring 1.4 comes with a simple tool that help in porting Qt classes to RingQt.

You will find it in ring/samples/tools/QtClassConverter

Online : <https://github.com/ring-lang/ring/tree/master/samples/tools/QtClassConverter>

Screen Shot:



6.11 What is new in Ring 1.4.1?

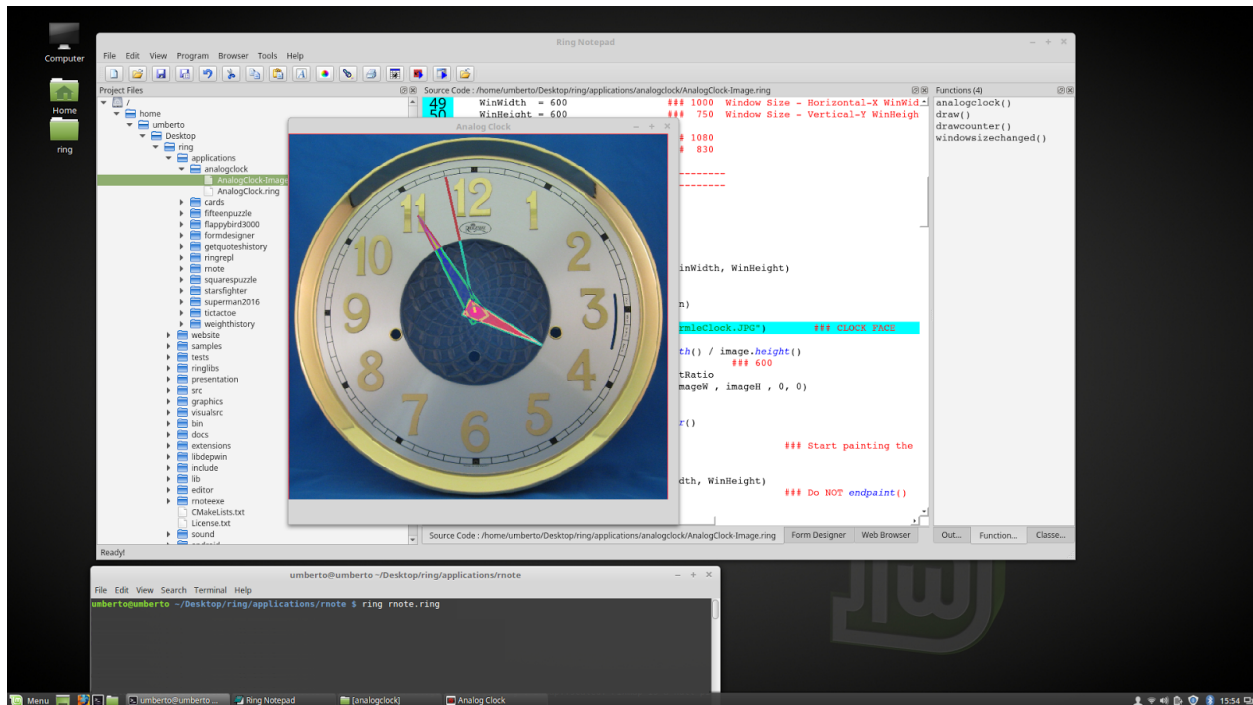
Ring 1.4.1 comes with the next changes

- Better Scripts for Building from Source Code
- Better Colors for the Modern Style in Ring Notepad
- Better StdLib
- Better RingQt
- New Sample : Sixteen Puzzle

The scripts are updated for building from source code.

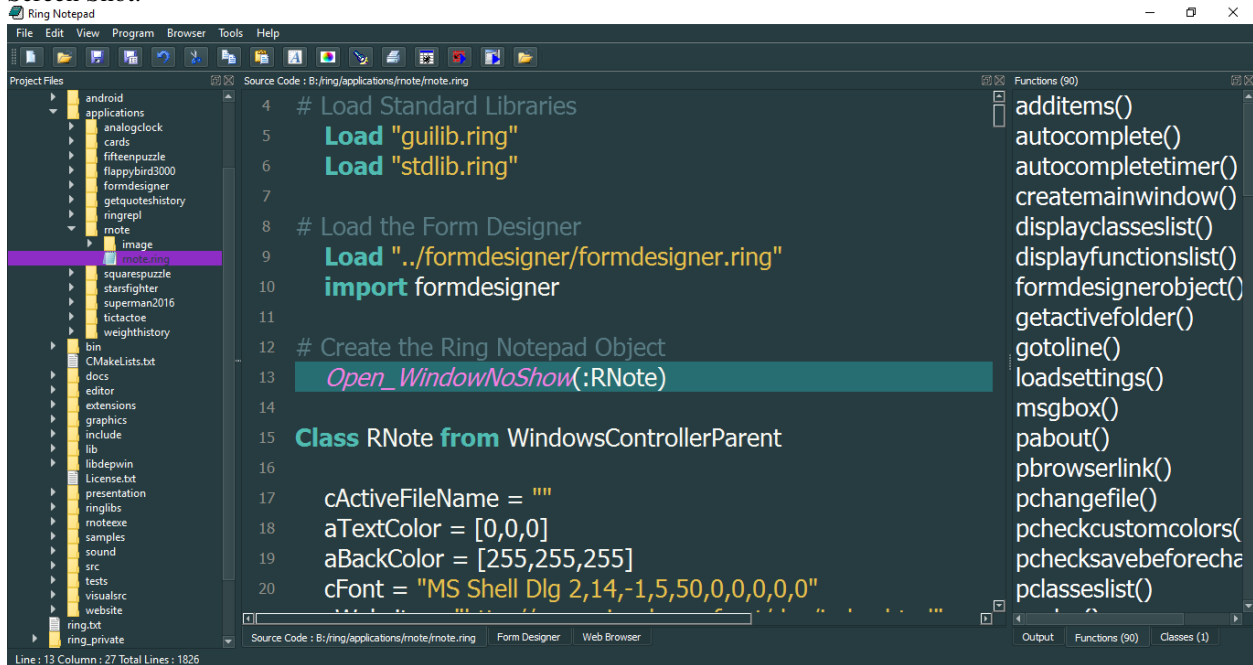
Tested using Windows, Ubuntu Linux, Linux Mint and MacOS X.

Screen Shot:



In Ring Notepad - the (Modern) Style colors are updated

Screen Shot:



The StdLib is updated to include the next functions

- TrimLeft()
- TrimRight()
- TrimAll()
- EpochTime()

The next functions are updated to display the dialogs on the top of other windows.

- `SetDialogIcon(cIconFile)`
- `MsgInfo(cTitle,cMessage)`
- `ConfirmMsg(cTitle,cMessage)`
- `InputBox(cTitle,cMessage)`
- `InputBoxInt(cTitle,cMessage)`
- `InputBoxNum(cTitle,cMessage)`
- `InputBoxPass(cTitle,cMessage)`

The Sixteen Puzzle is added to the Applications folder.

Screen Shot:



WHAT IS NEW IN RING 1.3?

In this chapter we will learn about the changes and new features in Ring 1.3 release.

7.1 List of changes and new features

Ring 1.3 comes with many new features

- Better RingQt
- Better Ring Notepad
- Ring mode for Emacs Editor
- Better StdLib
- Better Loop/Exit Command
- New Functions
- Return Self by Reference
- Using ‘<’ and ‘.’ operators as ‘from’ keyword
- Embedding Ring in Ring without sharing the State
- RingZip Library
- Form Designer

7.2 Better RingQt

(1) Another version of QPixmap class is added (QPixmap2) which takes (int width,int height) during object init.

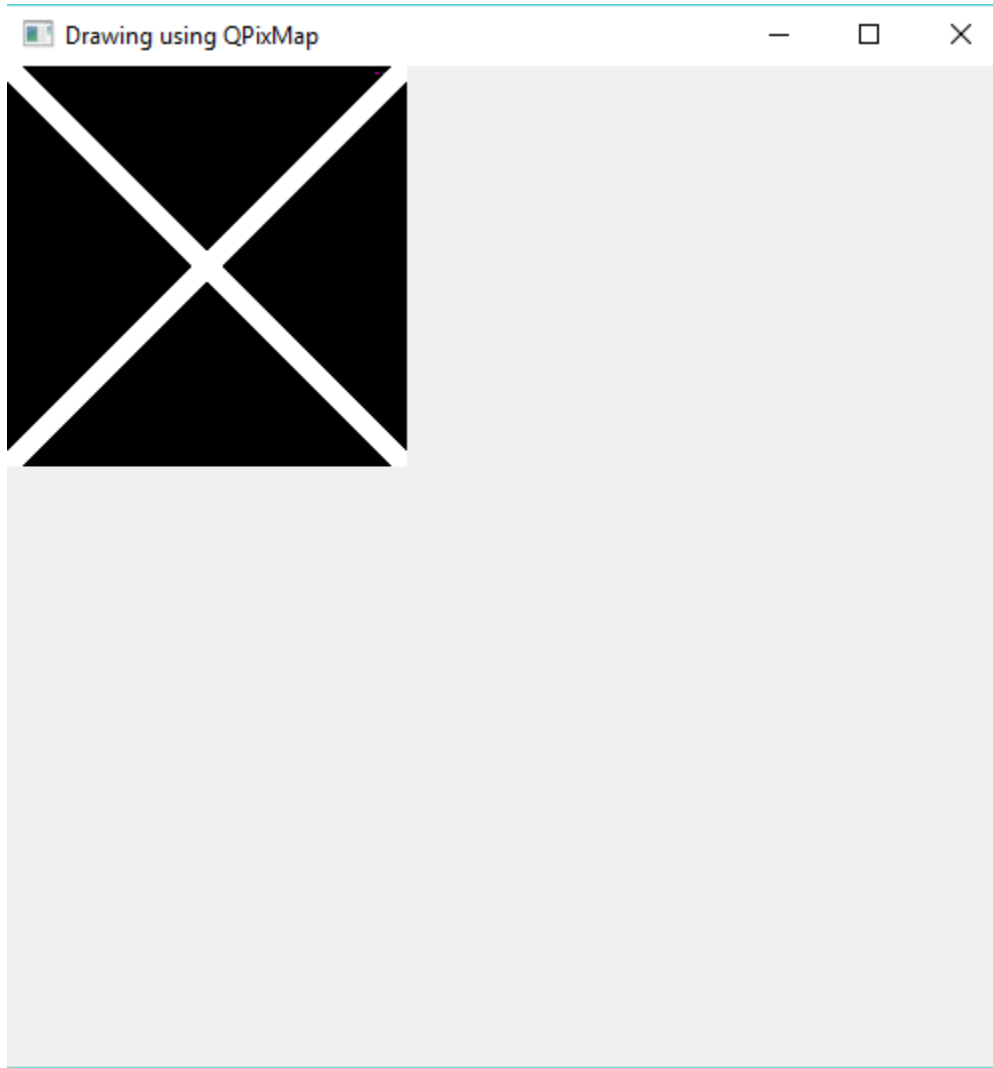
Example:

```
Load "guilib.ring"
New qapp
{
    win1 = new qwidget()
    {
        setwindowtitle("Drawing using QPixmap")
        setgeometry(100,100,500,500)
        labell = new qlabel(win1)
        {
            setgeometry(10,10,400,400)
            settext("")
        }
    }
}
```



```
    }
    imageStock = new QLabel(win1)
    {
        image = new QPixmap(200,200)
        color = new QColor() {
            setRgb(255,255,255,255)
        }
        pen = new QPen() {
            setColor(color)
            setWidth(10)
        }
        new QPainter() {
            begin(image)
                setPen(pen)
                drawLine(0,0,200,200)
                drawLine(200,0,0,200)
            endPaint()
        }
        setPixmap(image)
    }
    show()
}
exec()
```

Screen Shot:



2. The Objects Library is updated to include the next functions

- Last_WindowID()
- Open_WindowNoShow()
- Open_WindowAndLink()

Also the class name (WindowViewBase) is changed to (WindowsViewParent).

In The next code for example the Open_WindowAndLink() will create an object from the SecondWindowController Class Then will add the Method SecondWindow() to the FirstWindowController Class Also will add the Method FirstWindow() to the SecondWindowController Class

So the SendMessage() method in FirstWindowController class can use the SecondWindow() method to access the object.

```
class firstwindowController from windowsControllerParent

    oView = new firstwindowView

    func OpenSecondWindow
        Open_WindowAndLink (:SecondWindowController, self)
```

```

func SendMessage
    if IsSecondWindow()
        SecondWindow().setMessage("Message from the first window")
    ok

func setMessage cMessage
    oView.Label1.setText(cMessage)

```

3. The next classes are added to RingQt

- QPixmap2
- QScrollArea
- QSplitter
- QCompleter
- QCompleter2
- QCompleter3
- QProcess
- QMdiArea
- QMdiSubWindow
- QCursor
- QListView
- QDesktopServices

4. Many constants are defined in qt.rh (loaded by guilib.ring)

5. New Classes names - Index Start from 1

We added new classes to RingQt - another version of classes where the class names doesn't start with the "q" letter
Also updated methods so the index start from 1 when we deal with the GUI controls like

- ComboBox
- ListWidget
- TableWidget
- TreeWidget

These classes are inside guilib.ring under the package name : System.GUI

To use it

```

load "guilib.ring"

import System.GUI

```

This doesn't have any effect on our previous code, It's just another choice for better code that is consistent with Ring rules.

Also the form designer is updated to provide us the choice between using classes where (index start from 0) or (index start from 1)

Example (Uses the Form Designer)

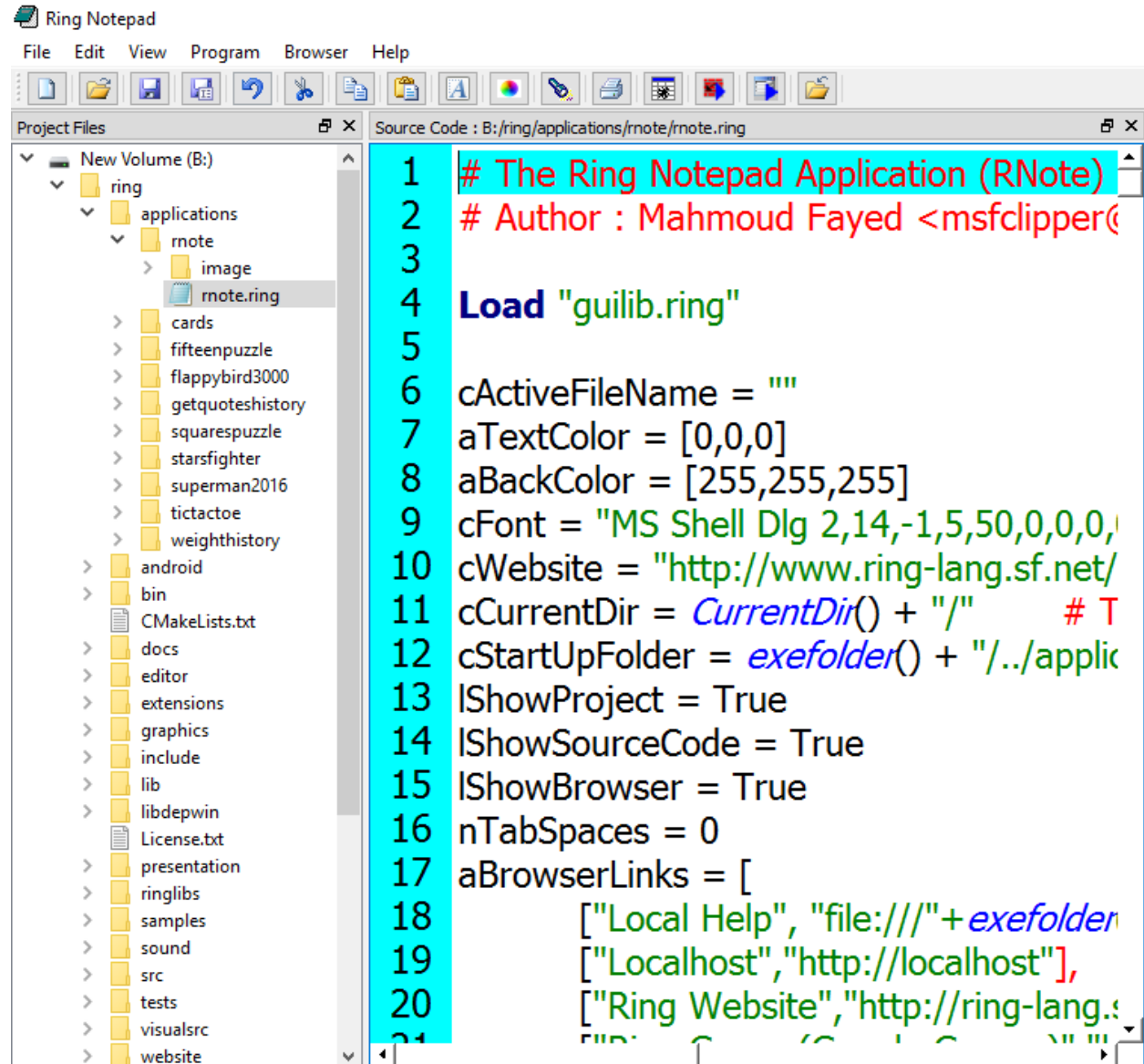
1. <https://github.com/ring-lang/ring/blob/master/applications/formdesigner/tests/indexstart/indexstartView.ring>

2. <https://github.com/ring-lang/ring/blob/master/applications/formdesigner/tests/indexstart/indexstartController.ring>

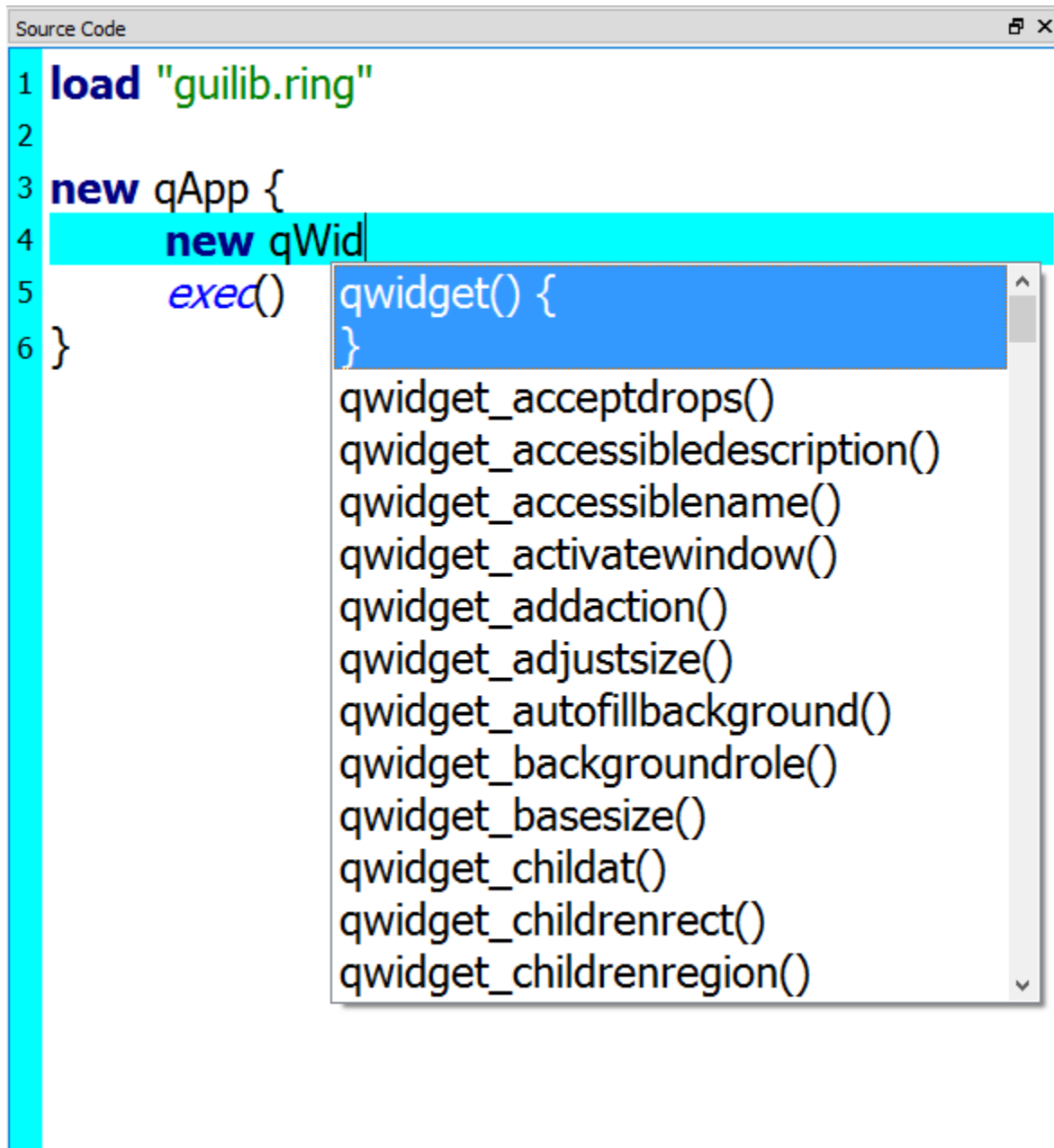
7.3 Better Ring Notepad

1. Using QPlainTextEdit instead of QTextEdit
2. Displaying the line number for each line in the source code file.

Screen Shot:



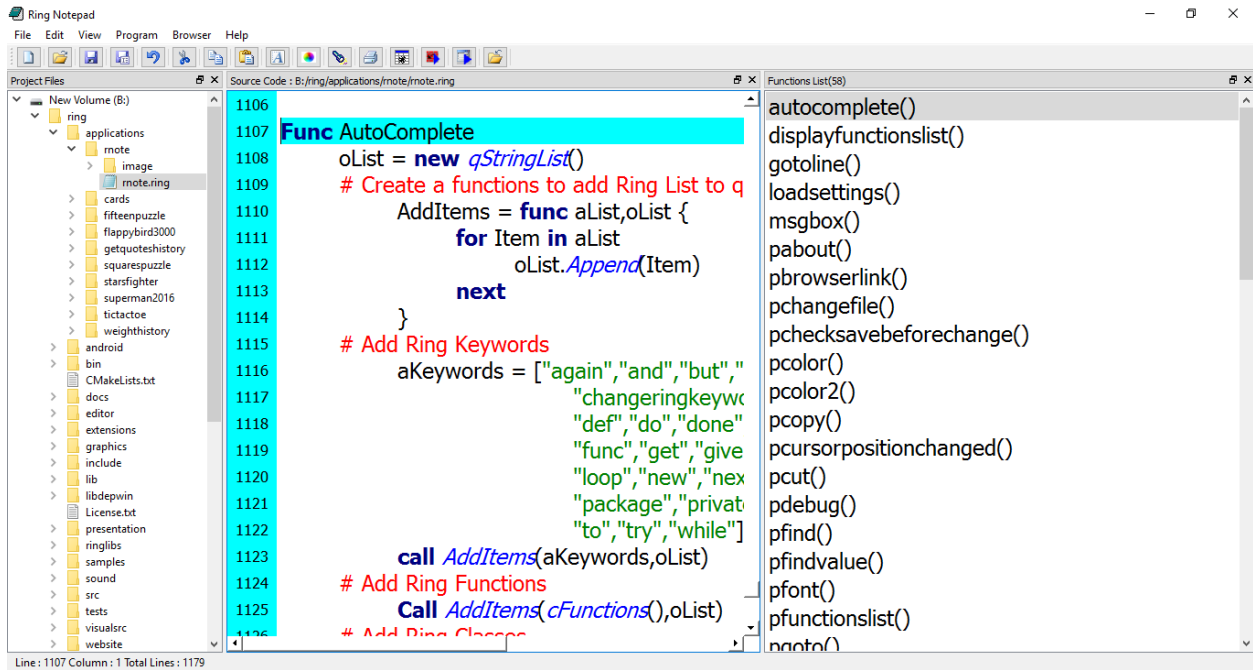
3. Auto-Complete for Ring functions names, classes and words in the opened file.



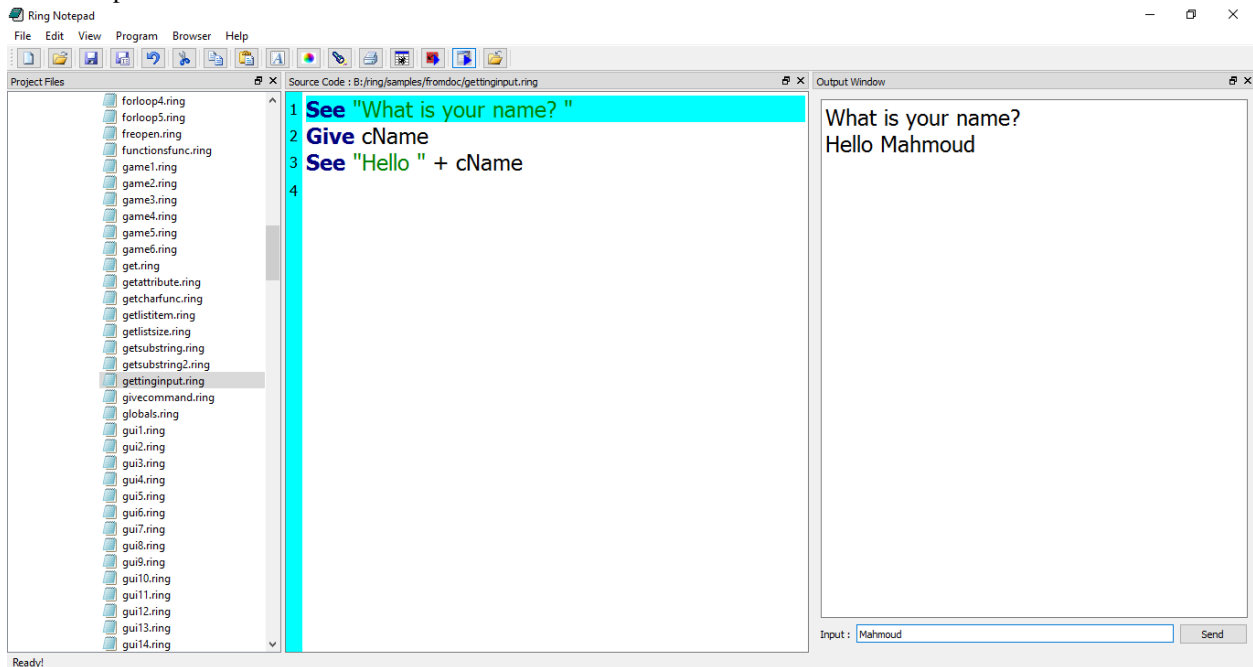
```
Source Code
1 load "guilib.ring"
2
3 new QApplication {
4   new QWidget
5   exec()
6 }
```

QWidget() {
}
QWidget_accepttdrops()
QWidget_accessibledescription()
QWidget_accessiblename()
QWidget_activatewindow()
QWidget_addaction()
QWidget_adjustsize()
QWidget_autofillbackground()
QWidget_backgroundrole()
QWidget_basesize()
QWidget_childat()
QWidget_childrenrect()
QWidget_childrenregion()

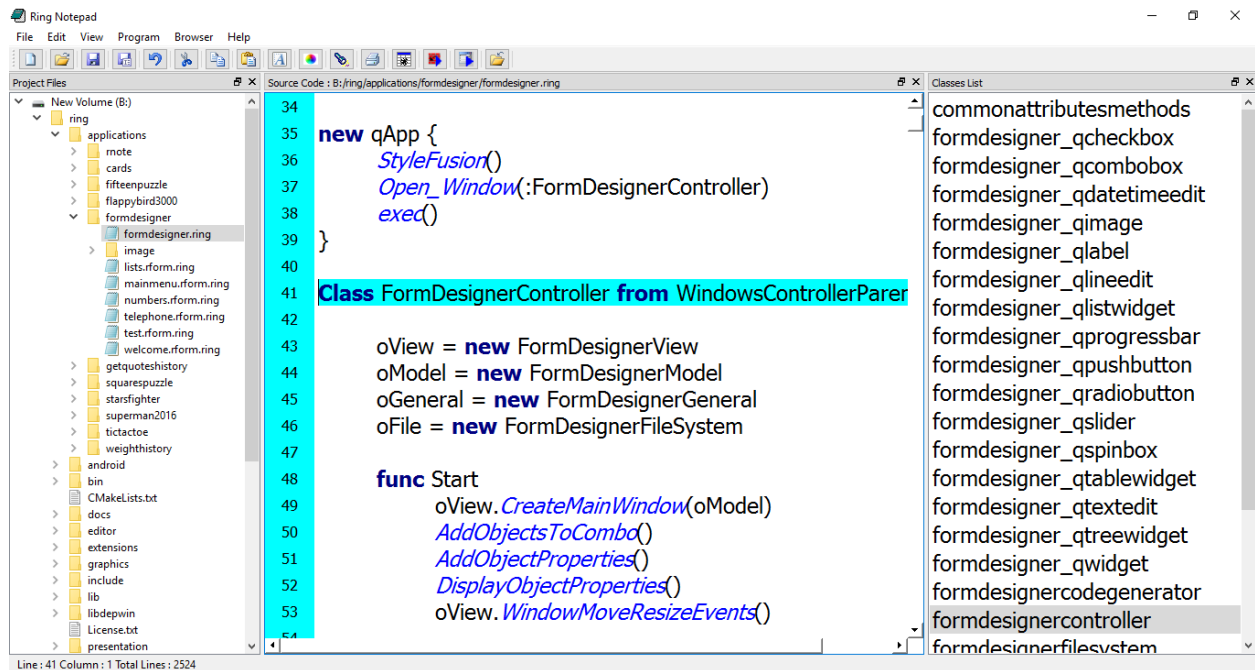
4. Functions and Methods List



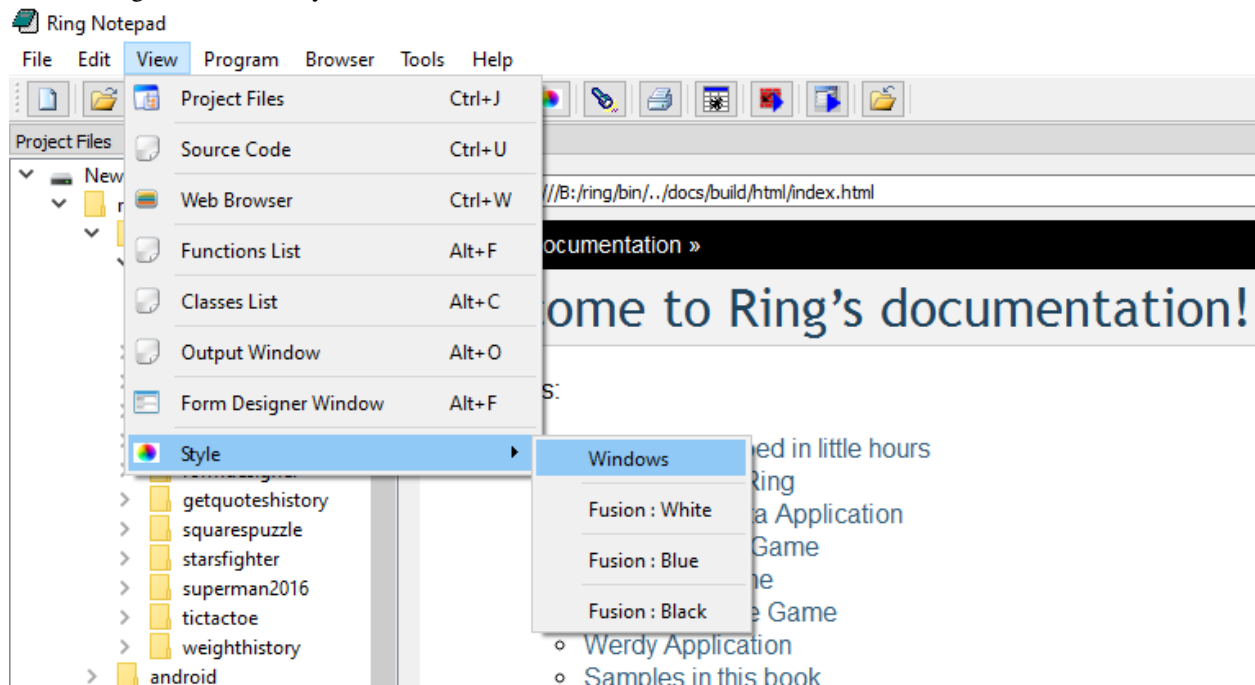
5. Output Window



6. Classes List



7. Change the Current Style



7.4 Ring mode for Emacs Editor

Ring 1.3 comes with Ring mode for Emacs Editor

Screen Shot:

```

func checkwin ogame
  if oGameState.gameresult return ok
  if oGameState.enemies = 0
    oGameState.gameresult = true
    oGame {
      if oGameState.level < 30
        text {
          point = 400
          size = 30
          file = "fonts/pirulen.ttf"
          text = "Level Completed!"
          nStep = 3
          x = 500 y=10
          state = func ogame,oself {
            if oself.y >= 400
              ogame.shutdown = true
              oGameState.level++
              oGameState.enemies = oGameState.level
              oGameState.gameresult = false
              ok
            }
          }
        }
      else
        text {
          point = 400
          size = 30
          nStep = 3
          file = "fonts/pirulen.ttf"
          text = "You Win !!!"
          x = 500 y=10
        }
    }
  }
}

- \--- game.ring 77% L304 Git-master (ring)

```

7.5 Better StdLib

The StdLib is updated to include the next functions

- SplitMany()
- JustFilePath()
- JustFileName()

7.6 Better Loop|Exit Command

The Loop|Exit command is updated to accept Expressions after the command (not only numbers).

The syntax:

```
Loop|Exit [Number]
```

Changed to

```
Loop|Exit [Expression]
```

Example

```

XLoop = 2      # The outer loop
YLoop = 1      # The first inner loop
for x = 1 to 10
  for y = 1 to 10
    see "x=" + x + " y=" + y + nl
    if x = 3 and y = 5
      exit XLoop
  ok

```



```
next
next
```

7.7 New Functions

- `PackageName()` function
- `Swap()` function

Example:

```
aList = [:one,:two,:four,:three]
see aList
see copy(" ",50) + nl
swap(aList,3,4)
see aList
```

Output

```
one
two
four
three
*****
one
two
three
four
```

7.8 Return Self by Reference

In this release, using Return Self in class methods will return the object by reference.

Example:

```
mylist = [new mytest() {
    see self
    x = 20
    see self
}]

see mylist

class mytest
    x = 15
    func init
        return self    # Return by reference
```

Output

```
x: 15.000000
x: 20.000000
x: 20.000000
```

7.9 Using ‘<’ and ‘:’ operators as ‘from’ keyword

In this release of the Ring language we can use the ‘<’ and ‘:’ operators as the ‘from’ keyword

Syntax (1):

```
class Cat from Animal
```

Syntax (2):

```
class Cat < Animal
```

Syntax (3):

```
class Cat : Animal
```

7.10 Embedding Ring in Ring without sharing the State

From Ring 1.0 we already have functions for embedding Ring in the C language. Also we can execute Ring code inside Ring programs using the eval() function. In this release we provide functions for embedding Ring in Ring programs without sharing the state.

Advantages:

1. Quick integration for Ring programs and applications together without conflicts.
2. Execute and run Ring code in safe environments that we can trace.

Example:

```
pState = ring_state_init()
ring_state_runcode(pState, "See 'Hello, World!'+\n")
ring_state_runcode(pState, "x = 10")

pState2 = ring_state_init()
ring_state_runcode(pState2, "See 'Hello, World!'+\n")
ring_state_runcode(pState2, "x = 20")

ring_state_runcode(pState, "see x +\n")
ring_state_runcode(pState2, "see x +\n")

v1 = ring_state_findvar(pState, "x")
v2 = ring_state_findvar(pState2, "x")

see v1[3] + \n
see v2[3] + \n

ring_state_delete(pState)
ring_state_delete(pState2)
```

Output:

```
Hello, World!
Hello, World!
10
20
10
20
```

7.11 RingZip Library

Ring 1.3 comes with the RingZip library for creating, modifying and extracting *.zip files.

Example (1): Create myfile.zip contains 4 files

```
load "ziplib.ring"
oZip = zip_openfile("myfile.zip", 'w')
zip_addfile(oZip, "test.c")
zip_addfile(oZip, "zip.c")
zip_addfile(oZip, "zip.h")
zip_addfile(oZip, "miniz.h")
zip_close(oZip)
```

Example (2): Extract myfile.zip to myfolder folder.

```
load "ziplib.ring"
zip_extract_allfiles("myfile.zip", "myfolder")
```

Example (3): Print file names in the myfile.zip

```
load "ziplib.ring"
oZip = zip_openfile("myfile.zip", 'r')
for x=1 to zip_filescount(oZip)
    see zip_getfilenamebyindex(oZip,x) + nl
next
zip_close(oZip)
```

Example (4) : Using Classes instead of Functions

```
load "ziplib.ring"

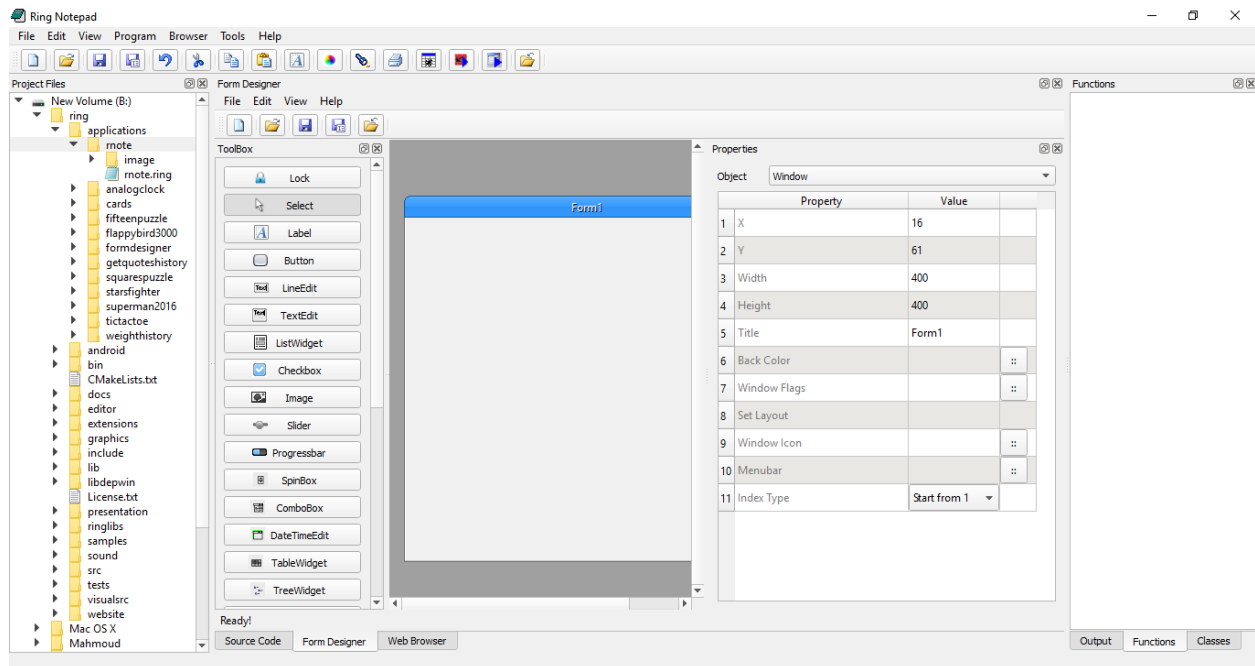
new Zip {
    SetFileName("myfile.zip")
    Open("w")
    AddFile("test.c")
    AddFile("zip.c")
    AddFile("zip.h")
    AddFile("miniz.h")
    Close()
}
```

7.12 Form Designer

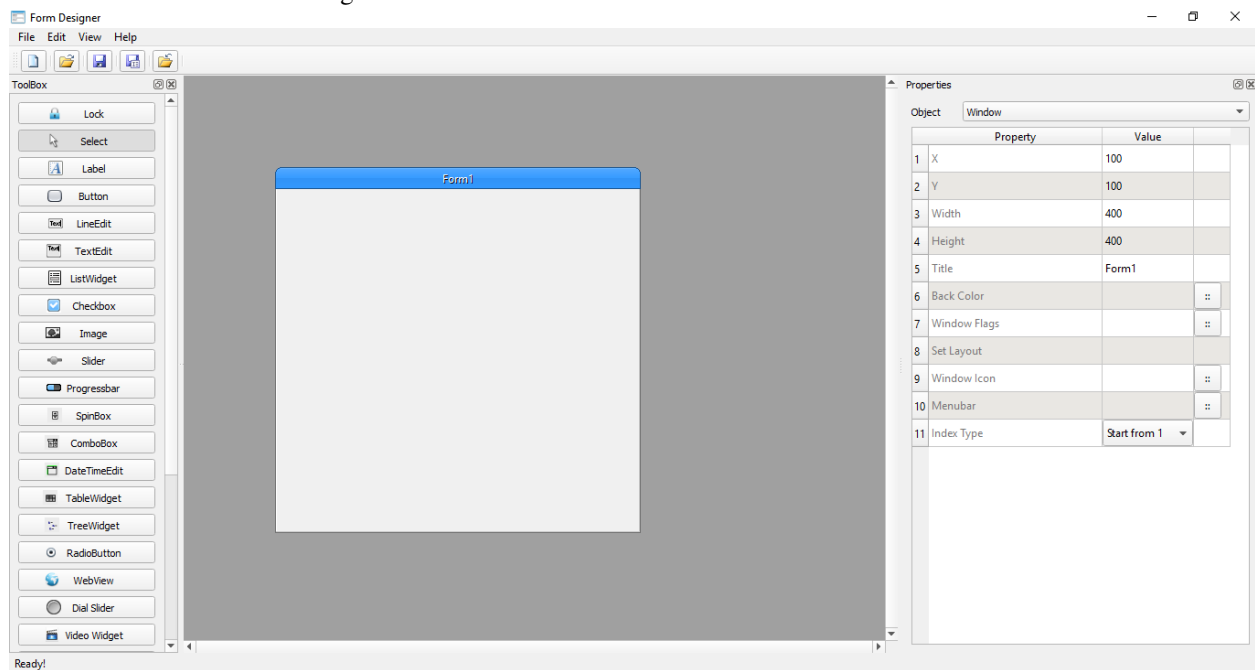
Ring 1.3 comes with the Form Designer to quickly design your GUI application windows/forms and generate the Ring source code.

It's written in Ring (Around 8000 Lines of code) using Object-Oriented Programming and Meta-Programming.

We can run the From Designer from Ring Notepad



Also we can run the Form Designer in another window.



WHAT IS NEW IN RING 1.2?

In this chapter we will learn about the changes and new features in Ring 1.2 release.

8.1 List of changes and new features

Ring 1.2 comes with many new features

- New Functions
- Better Functions
- Better Ring Notepad
- Better RingQt
- Objects Library for RingQt
- RingLibCurl
- Better Call Command
- Using NULL instead of NULLPointer()
- Display Warnings Option
- Better Quality

8.2 New Functions

- PtrCmp() Function is a new function that compare between C pointers like the GUI objects.
- PrevFileName() Function is added to return the previous active source file name.
- RingVM_CFunctionsList() Function is added to return a list of functions written in C.
- RingVM_FunctionsList() Function is added to return a list of functions written in Ring.
- RingVM_ClassesList() Function is added to return a list of Classes.
- RingVM_PackagesList() Function is added to return a list of Packages.
- RingVM_MemoryList() Function is added to return a list of Memory Scopes and Variables.
- RingVM_CallList() Function is added to return a list of the functions call list.
- RingVM_FilesList() Function is added to return a list of the Ring Files.

Example:

```
fp = fopen("ptrcmp.ring", "r")
fp2 = fp
fp3 = fopen("ptrcmp.ring", "r")

see ptrcmp(fp, fp2) + nl
see ptrcmp(fp, fp3) + nl

fclose(fp)
fclose(fp3)
```

Output:

```
1
0
```

Also we can compare between them using the '=' operator

Example:

```
fp = fopen("ptrcmp2.ring", "r")
fp2 = fopen("ptrcmp2.ring", "r")
fp3 = fp
see fp = fp2
see nl
see fp = fp3
fclose(fp)
fclose(fp2)
```

Output:

```
0
1
```

Example:

The next function in stdlib.ring uses the PrevFileName() to know if the file of the caller function is the main source file of the program or not.

```
Func IsMainSourceFile
    if PrevFileName() = sysargv[2]
        return true
    ok
    return false
```

8.3 Better Functions

The find() function is updated to support searching in lists using C pointers like GUI Objects.

The type() function is updated to display the C pointers types (like the GUI Object Class Name).

8.4 Better Ring Notepad

The Ring Notepad will save the current line number of opened files to be restored when we switch between files.

Also Ring Notepad will ask the user to save the file if the file content is changed when the user switch between files.

8.5 Better RingQt

RingQt classes are updated to include methods to get events (The code that will be executed when an event is fired). This is necessary to enable/disable events for some time or to get the events information.

For example the next code disable an event then call a method then enable the event again.

```
cEvent = oView.oListResult.getCurrentItemChangedEvent()
oView.oListResult.setCurrentItemChangedEvent("")
FindValueAction()      # Call Method while an event is disabled
oView.oListResult.setCurrentItemChangedEvent(cEvent)
```

Also the QAllEvents class is updated where we can set the output from the event function to be true or false using a new method added to the class called setEventOutput.

```
Load "guilib.ring"

MyApp = New qApp {
    win = new QWidget() {
        setWindowTitle("Hello World")
        setGeometry(100,100,370,250)
        linedit1 = new QLineEdit(win) {
            setGeometry(10,100,350,30)
            setInputMask("9999;_")
            oFilter = new QAllEvents(linedit1)
            oFilter.setFocusOutEvent("pMove()")
            installEventFilter(oFilter)
        }
        linedit2 = new QLineEdit(win) {
            setGeometry(10,150,350,30)
        }
        show()
    }
    exec()
}

func pMove
    win.setWindowTitle("xxxx")
    oFilter.setEventOutput(False)
```

8.6 Objects Library for RingQt

Ring 1.2 comes with the Objects library for RingQt applications. Instead of using global variables for windows objects and connecting events to objects using the object name, the Objects Library will manage the GUI objects and will provide a more natural API to quickly create one or many windows from the same class and the library provide a way to quickly set methods to be executed when an event is fired. Also the library provide a natural interface to quickly use the parent or the caller windows from the child or sub windows.

The Objects Library is designed to be used with the MVC Design Pattern.

The Objects Library is merged in RingQt so you can use it directly when you use RingQt

Example :

```
load "guilib.ring"

new qApp {
```

```

        open_window( :MainWindowController )
        exec()
    }

    class MainWindowController from WindowsControllerParent
        oView = new MainWindowView
        func SubWindowAction
            Open_window( :SubWindowController )
            Last_Window().SetParentObject(self)

    class MainWindowView from WindowsViewParent
        win = new QWidget() {
            SetWindowTitle("Main Window")
            btnSub = new QPushButton(win) {
                setText("Sub Window")
                setClickEvent( Method( :SubWindowAction ) )
            }
            resize(400,400)
        }

    class SubWindowController from WindowsControllerParent
        oView = new SubWindowView
        func SetMainWindowTitleAction
            Parent().oView.win.SetWindowTitle("Message from the Sub Window")
            oView.win.SetWindowTitle("Click Event Done!")

    class SubWindowView from WindowsViewParent
        win = new QWidget() {
            SetWindowTitle("Sub Window")
            btnMsg = new QPushButton(win) {
                setText("Set Main Window Title")
                setClickEvent( Method( :SetMainWindowTitleAction ) )
            }
            btnClose = new QPushButton(win) {
                Move(200,0)
                setText("Close")
                setClickEvent( Method( :CloseAction ) )
            }
            resize(400,400)
        }
    }

```

8.7 RingLibCurl

The LibCurl library is used starting from Ring 1.0 for the Download() and SendEmail() functions implementation. In Ring 1.2 more functions are added to provide a powerful library (RingLibCurl) around LibCurl.

Example:

```

load "libcurl.ring"

curl = curl_easy_init()

cPostThis = "page=4&Number1=4&Number2=5"
curl_easy_setopt(curl, CURLOPT_URL, "http://localhost/ringapp/index.ring?page=3")
curl_easy_setopt(curl, CURLOPT_POSTFIELDS, cPostThis)

```



```
curl_easy_perform(curl)

curl_easy_cleanup(curl)
```

8.8 Better Call Command

The Call command is updated to support calling functions from object attributes also (not only variables).

For example the next code from the Stars Fighter Game

```
cFunc = oself.keypress
call cFunc(oGame,oSelf,Key_Space)
```

Can be written in one line

```
call oself.keypress(oGame,oSelf,Key_Space)
```

8.9 Using NULL instead of NULLPointer()

We can pass NULL to functions instead of using NULLPointer()

For example the next code from RingLibSDL

```
SDL_RenderCopy(SDL_ren,tex,NULLPointer(),rect)
```

Can be written as in the next line

```
SDL_RenderCopy(SDL_ren,tex,NULL,rect)
```

8.10 Display Warnings Option

In Ring 1.2 the Ring compiler is updated to include the Display Warnings option (-w)

Example:

```
load "stdlib.ring"
load "stdlib.ring"
```

compiling the program using the Display Warnings option will display the file duplication warnings, While without that option the error will pass silent.

This is a warning (not an error) because in large projects you may use the same file more than one time. For example it's common to start each file with the next code. where the function IsMainSourceFile() is part from the stdlib.ring

```
load "stdlib.ring"
if IsMainSourceFile()
    // Testing
ok
```

8.11 Better Quality

Ring 1.2 is more stable, We discovered and fixed more bugs during Ring usage everyday in practical projects. Some functions are optimized to be faster like the `SubStr()` function. Also the documentation is more better.

WHAT IS NEW IN RING 1.1?

In this chapter we will learn about the changes and new features in Ring 1.1 release.

9.1 List of changes and new features

Ring 1.1 comes with many new features

- Better Natural Language Programming Support
- Generate/Execute Ring Object Files (*.ringo)
- Syntax Flexibility and different styles for I/O and Control Structures
- New Functions and Changes
- StdLib functions and classes written in Ring
- RingLibSDL
- Demo Project - Game Engine for 2D Games
- RingSQLite
- Better Code Generator for Extensions
- Using Self.Attribute in the Class Region to define new attributes
- Using This.Attribute in nested Braces inside the Class Methods
- Better Documentation

9.2 Better Natural Language Programming Support

Ring is an innovative language because of it's compact syntax, smart implementation (small, transparent & visual) and it's ability to create declarative and natural domain specific languages in a fraction of time.

This release add support for calling methods when an expression is evaluated

check this example:

```
# Natural Code
new program {
    Accept 2 numbers then print the sum
}

# Natural Code Implementation
```

```

class program
  # Keywords
  Accept=0 numbers=0 then=0 print=0 the=0 sum=0

  # Execution
  func braceexprval x
    value = x
  func getnumbers
    for x=1 to value
      see "Enter Number (" + x + ") :" give nNumber
      aNumbers + nNumber
    next
  func getsum
    nSum = 0
    for x in aNumbers nSum+= x next
    see "The Sum : " + nSum
private
  value=0 aNumbers=[]

```

Output:

```

Enter Number (1) :3
Enter Number (2) :4
The Sum : 7

```

for more information see the “Natural Language Programming” chapter.

9.3 Generate/Execute Ring Object Files (*.ringo)

This feature enable you to distribute your applications without distributing the source code. Also it makes application distribution a simple process where you get one Ring object file for the complete project (many source code files). Also using Ring object file remove the loading time required for compiling the application.

Check the “command line options” chapter to know more about this feature.

9.4 Syntax Flexibility and different styles for I/O and Control Structures

Programmers are sensitive to the programming language syntax. Great programmers know how to work using many different styles but each programmer may have his/her favorite style.

Each programming language comes with a style that you may like or not. Ring is just one of these languages, but as a response to many programmers asking for a better syntax we decided to provide more options.

Also some of these features are very necessary for Natural Language Programming.

Example :

We have two commands to change language keywords and operators.

```

ChangeRingOperator + plus
ChangeRingKeyword see print

Print 5 plus 5

```

```
ChangeRingOperator plus +
ChangeRingKeyword print see
```

We have new styles (Optional) for Input/Output.

Example :

```
Put "What is your name? "
Get cName
Put "Hello " + cName
```

Example :

```
Load "stdlib.ring"

Print("What is your name? ")      # print message on screen
cName=GetString()                 # get input from the user
print("Hello #{cName}")           # say hello!
```

We have new styles (optional) for control structures.

Example :

```
While True

    Put "
        Main Menu
        -----
        (1) Say Hello
        (2) About
        (3) Exit

    " Get nOption

    Switch nOption
    Case 1
        Put "Enter your name : "
        Get name
        Put "Hello " + name + nl
    Case 2
        Put "Sample : using while loop" + nl
    Case 3
        Bye
    Else
        Put "bad option..." + nl
    End
End
```

Example :

```
Load "stdlib.ring"

While True {

    print("
        Main Menu
        -----
        (1) Say Hello
        (2) About
        (3) Exit
```

```
        ")

    nOption = GetString()

    switch nOption {
    case 1
        print("Enter your name : ")
        name = getstring()
        print("Hello #{name}\n")
    case 2
        print("Sample : using switch statement\n")
    case 3
        Bye
    else
        print("bad option...\n")
    }
}
```

Check the next chapters:-

- Getting Started - Second Style
- Getting Started - Third Style
- Control Structures - Second Style - May looks like Lua and Ruby
- Control Structures - Third Style - May looks like C (uses braces)
- Syntax Flexibility

Note: All of these styles are provided automatically by the compiler at the same time, It's better to select one style for the same project (you can create your style as a mix from these styles) for example you can use Put/Get and Braces.

9.5 New Functions and Changes

Changed:

- get() function : changed to sysget()
- sort() function : can now work on list of objects
- find() function : can now work on list of objects

Added:

- clocksperssecond()
- CurrentDir()
- ExeFileName()
- ChDir()
- ExeFolder()
- varptr()
- space()
- nullpointer()

- object2pointer()
- pointer2object()

Check the next chapters

- System Functions
- Object Oriented Programming (OOP)
- Low Level Functions

9.6 StdLib functions and classes written in Ring

Ring 1.1 comes with a library called StdLib, it's written in Ring by the help of Ring Team

The library provide a useful group of new functions and classes

Example:

```
Load "stdlib.ring"

Puts("Test Times() ")
Times ( 3 , func { see "Hello, World!" + nl } )
```

Example:

```
Load "stdlib.ring"

Puts("Test Map() ")
See Map( 1:10, func x { return x*x } )
```

Example:

```
Load "stdlib.ring"

Puts("Test Filter() ")
See Filter( 1:10 , func x { if x <= 5 return true else return false ok } )
```

Example:

```
Load "stdlib.ring"

See "Testing the String Class" + nl
oString = new string("Hello, World!")
oString.println()
oString.upper().println()
oString.lower().println()
oString.left(5).println()
oString.right(6).println()
```

Example:

```
Load "stdlib.ring"

oList = new list ( [1,2,3] )
oList.Add(4)
oList.print()
```

Example:

```
Load "stdlib.ring"

oStack = new Stack
oStack.push(1)
oStack.push(2)
oStack.push(3)
see oStack.pop() + nl
```

Example:

```
Load "stdlib.ring"

oQueue = new Queue
oQueue.add(1)
oQueue.add(2)
oQueue.add(3)
see oQueue.remove() + nl
```

Example:

```
Load "stdlib.ring"

ohashtable = new hashtable
See "Test the hashtable Class Methods" + nl
ohashtable {
    Add("Egypt", "Cairo")
    Add("KSA", "Riyadh")
    see self["Egypt"] + nl
    see self["KSA"] + nl
    see contains("Egypt") + nl
    see contains("USA") + nl
    see index("KSA") + NL
    print()
    delete(index("KSA"))
    see copy(" ", 60) + nl
    print()
}
```

Example:

```
Load "stdlib.ring"

otree = new tree
See "Test the tree Class Methods" + nl
otree {
    set("The first step") # set the root node value
    see value() + nl
    Add("one")
    Add("two")
    Add("three") {
        Add("3.1")
        Add("3.2")
        Add("3.3")
        see children
    }
    see children
    oTree.children[2] {
        Add("2.1") Add("2.2") Add("2.3") {
            Add("2.3.1") Add("2.3.2") Add("test")
        }
    }
}
```



```

        }
    }
    oTree.children[2].children[3].children[3].set("2.3.3")
}
see copy(" ", 60) + nl
oTree.print()

```

Check the next chapters:

- StdLib Functions
- StdLib Classes

9.7 RingLibSDL

Ring 1.0 provided RingAllegro to be able to create games using the Allegro game programming library

Now Ring 1.1 provide RingLibSDL also so we can have the choice between Allegro or LibSDL

Example:

```

Load "libsdl.ring"

SDL_Init(SDL_INIT_EVERYTHING)
win = SDL_CreateWindow("Hello World!", 100, 100, 640, 480, SDL_WINDOW_SHOWN)
SDL_Delay(2000)
SDL_DestroyWindow(win)
SDL_Quit()

```

See the RingLibSDL Chapter.

9.8 Demo Project - Game Engine for 2D Games

In practice we would create a game engine in a language like C/C++ to get the best performance then provide Ring classes to use the engine.

But many 2D Games are simple and creating a game engine in Ring will be fast enough in many cases

Also this would be a good demo project to learn about the language concepts where we build things using Object Oriented Programming (OOP) then access the power that we have using declarative programming using nested structures or using natural programming.

In this project we selected the first way (declarative programming using nested structures)

Example:

```

Load "gameengine.ring" # Give Control to the Game Engine

func main              # Called by the Game Engine

    oGame = New Game    # Create the Game Object
    {
        title = "My First Game"
        text {
            x = 10  y=50
            animate = false
            size = 20
        }
    }

```

```

        file = "fonts/pirulen.ttf"
        text = "game development using ring is very fun!"
        color = rgb(0,0,0)          # Color = black
    }
    text {
        x = 10  y=150
        # Animation Part =====
        animate = true              # Use Animation
        direction = GE_DIRECTION_INCVERTICAL  # Increase y
        point = 400                 # Continue until y=400
        nStep = 3                   # Each time y+= 3
        #=====
        size = 20
        file = "fonts/pirulen.ttf"
        text = "welcome to the real world!"
        color = rgb(0,0,255)        # Color = Blue
    }
    Sound {                          # Play Sound
        file = "sound/music1.wav"    # Sound File Name
    }
}                                     # Start the Events Loop

```

See the “Demo Project - Game Engine for 2D Games” chapter.

9.9 RingSQLite

Ring 1.0 provided support for ODBC to use any database and provided native support for MySQL.

Now Ring 1.1 provide native support for SQLite database too.

Example:

```

oSQLite = sqlite_init()

sqlite_open(oSQLite,"mytest.db")

sql = "CREATE TABLE COMPANY(" +
        "ID INT PRIMARY KEY     NOT NULL," +
        "NAME           TEXT     NOT NULL," +
        "AGE             INT      NOT NULL," +
        "ADDRESS          CHAR(50)," +
        "SALARY           REAL );"

sqlite_execute(oSQLite,sql)

sql = "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) " +
        "VALUES (1, 'Mahmoud', 29, 'Jeddah', 20000.00 ); " +
        "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) " +
        "VALUES (2, 'Ahmed', 27, 'Jeddah', 15000.00 ); " +
        "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) " +
        "VALUES (3, 'Mohammed', 31, 'Egypt', 20000.00 );" +
        "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) " +
        "VALUES (4, 'Ibrahim', 24, 'Egypt ', 65000.00 );"

sqlite_execute(oSQLite,sql)

aResult =  sqlite_execute(oSQLite,"select * from COMPANY")

```

```

for x in aResult
    for t in x
        see t[2] + nl
    next
next
see copy(" ", 50) + nl
for x in aResult
    see x["name"] + nl
next
sqlite_close(oSQLite)

```

9.10 Better Code Generator for Extensions

We are using the code generator (written in Ring) every day to add new libraries to Ring.

The generator is used to create RingQt and RingAllegro

Also in Ring 1.1 it's used to create RingLibSDL.

more features are added like

- Set/Get structure members (numbers & pointers)
- Using constants
- Better Generated Code

See the Code Generator chapter.

9.11 Using Self.Attribute in the Class Region to define new attributes

We can use Self.Attribute in the Class Region (after the class name and before any methods) to define new attributes.

```

class Person
    name          # Define name as attribute if it's not a global variable
    address
    phone

class person2
    self.name      # Must Define the attribute
    self.address
    self.phone

```

9.12 Using This.Attribute in nested Braces inside the Class Methods

We can use nested braces {} while we are inside methods to access another objects, In this case the current object scope will be changed while we are inside the brace and Self will point to the object that we access using braces {}. In this case we can use This.Attribute and This.Method() to access the object that will be created from the current class.

Check the Object Oriented Programming chapter for more information.

Also Check the Weight History Application in GUI Development using RingQt chapter.

9.13 Better Documentation

Ring 1.1 documentation (800 pages) is better than Ring 1.0 documentation (340 pages)

Many chapters are added for providing better information about the language like

- Language Reference
- Scope Rules
- FAQ

And more!

BUILDING FROM SOURCE CODE

The Ring programming language is a free open source product (MIT License).

You can build Ring using CMake or using Scripts (Batch Files or Shell Scripts).

The next steps explains building using scripts.

10.1 Building using Microsoft Windows

Get the source code

```
git clone http://github.com/ring-lang/ring.git
```

Build Ring (Compiler/VM)

```
cd ring/src  
buildvc.bat  
buildvcw.bat
```

Build Ring2EXE

```
cd ../ring2exe  
buildring2exe.bat
```

Build RingODBC

```
cd ../extensions/ringodbc  
buildvc.bat
```

Build RingMySQL

```
cd ../extensions/ringmysql  
buildvc.bat
```

Build RingSQLite

```
cd ../extensions/ringsqlite  
buildvc.bat
```

Build RingOpenSSL

```
cd ../extensions/ringopenssl  
buildvc.bat
```

Build RingInternet

```
cd ../extensions/ringinternet
buildvc.bat
```

Build RingMurmurHash

```
cd ../extensions/ringmurmurhash
buildvc.bat
```

Generate RingConsoleColors Source Code and Build

```
cd ../extensions/ringconsolecolors
gencode.bat
buildvc.bat
```

Generate RingAllegro Source Code and Build

```
cd ../extensions/ringallegro
gencode.bat
buildvc.bat
```

Generate RingLibCurl Source Code and Build

```
cd ../extensions/ringcurl
gencode.bat
buildvc.bat
```

Generate RingZip Source Code and Build

```
cd ../extensions/ringzip
gencode.bat
buildvc.bat
```

Generate RingFreeGLUT Source Code and Build

```
cd ../extensions/ringfreeglut
gencode.bat
buildvc.bat
```

Generate RingOpenGL Source Code and Build

The ringopengl folder contains many sub folders for different OpenGL versions

Starting from OpenGL 1.1 to OpenGL 4.6

```
cd ../extensions/ringopengl/opengl21
gencode.bat
buildvc.bat
```

Install Qt 5.5 : <https://download.qt.io/archive/qt/5.5/5.5.1/>

Generate RingQt Source Code and Build

```
cd ../extensions/ringqt
gencode.bat
buildmingw32.bat
```

To be able to call ring from any folder

```
cd ../../bin
install.bat
```

Add Ring/bin to System path

```
Hit "windows key".
Type "Edit the System environment variables"
Select "Advanced" tab.
Click on "Enviroment Variables..."
Double click on "Path"
Add at the end the new path separated by semicolon.
;C:\Ring\Bin
```

Run Ring Notepad

```
cd applications/rnote
ring rnote.ring
```

10.2 Building using Ubuntu Linux

Get the source code

```
git clone http://github.com/ring-lang/ring.git
```

Install Libraries

```
cd ring/src
./installdep.sh
```

Build Ring (Compiler/VM)

```
sudo ./buildgcc.sh
```

Build Ring2EXE

```
cd ../ring2exe
sudo ./buildring2exe.sh
```

Build RingODBC

```
cd ../extensions/ringodbc
./buildgcc.sh
```

Build RingMySQL

```
cd ../extensions/ringmysql
./buildgcc.sh
```

Build RingSQLite

```
cd ../extensions/ringsqlite
./buildgcc.sh
```

Build RingOpenSSL

```
cd ../extensions/ringopenssl
./buildgcc.sh
```

Build RingInternet

```
cd ../extensions/ringinternet
./buildgcc.sh
```

Build RingMurmurHash

```
cd ../extensions/ringmurmurhash
./buildgcc.sh
```

Generate RingConsoleColors Source Code and Build

```
cd ../extensions/ringconsolecolors
./gencode.sh
./buildgcc.sh
```

Generate RingAllegro Source Code and Build

```
cd ../extensions/ringallegro
./gencode.sh
./buildgcc.sh
```

Generate RingLibCurl Source Code and Build

```
cd ../extensions/ringcurl
./gencode.sh
./buildgcc.sh
```

Generate RingZip Source Code and Build

```
cd ../extensions/ringzip
./gencode.sh
./buildgcc.sh
```

Generate RingFreeGLUT Source Code and Build

```
cd ../extensions/ringfreeglut
./gencode.sh
./buildgcc.sh
```

Generate RingOpenGL Source Code and Build

The ringopengl folder contains many sub folders for different OpenGL versions

Starting from OpenGL 1.1 to OpenGL 4.6

```
cd ../extensions/ringopengl/opengl21
gencode.sh
buildgcc.sh
```

Generate RingQt Source Code and Build

```
cd ../extensions/ringqt
./gencode.sh
./buildgcc.sh
```

To be able to call ring from any folder

```
cd ../../bin
sudo ./install.sh
```

Run Ring Notepad

```
cd applications/rnote
ring rnote.ring
```


10.3 Building using Fedora Linux

Get the source code

```
git clone http://github.com/ring-lang/ring.git
```

Install Libraries

```
cd ring/src
./installdepfedora.sh
```

Build Ring (Compiler/VM)

```
sudo ./buildgcc.sh
```

Build Ring2EXE

```
cd ../ring2exe
sudo ./buildring2exe.sh
```

Build RingODBC

```
cd ../extensions/ringodbc
./buildgcc.sh
```

Build RingMySQL

```
cd ../extensions/ringmysql
./buildgccfedora.sh
```

Build RingSQLite

```
cd ../extensions/ringsqlite
./buildgcc.sh
```

Build RingOpenSSL

```
cd ../extensions/ringopenssl
./buildgcc.sh
```

Build RingInternet

```
cd ../extensions/ringinternet
./buildgcc.sh
```

Build RingMurmurHash

```
cd ../extensions/ringmurmurhash
./buildgcc.sh
```

Generate RingConsoleColors Source Code and Build

```
cd ../extensions/ringconsolecolors
./gencode.sh
./buildgcc.sh
```

Generate RingAllegro Source Code and Build

```
cd ../extensions/ringallegro
./gencode.sh
./buildgcc.sh
```

Generate RingLibCurl Source Code and Build

```
cd ../extensions/ringcurl
./gencode.sh
./buildgcc.sh
```

Generate RingZip Source Code and Build

```
cd ../extensions/ringzip
./gencode.sh
./buildgcc.sh
```

Generate RingFreeGLUT Source Code and Build

```
cd ../extensions/ringfreeglut
./gencode.sh
./buildgcc.sh
```

Generate RingOpenGL Source Code and Build

The ringopengl folder contains many sub folders for different OpenGL versions

Starting from OpenGL 1.1 to OpenGL 4.6

```
cd ../extensions/ringopengl/opengl21
gencode.sh
buildgcc.sh
```

Generate RingQt Source Code and Build

```
cd ../extensions/ringqt
./gencode.sh
./buildgccfedora.sh
```

To be able to call ring from any folder

```
cd ../../bin
sudo ./install.sh
```

Run Ring Notepad

```
cd applications/rnote
ring rnote.ring
```

10.4 Building using MacOS X

Get the source code

```
git clone http://github.com/ring-lang/ring.git
```

Install homebrew (follow the directions on homebrew's homepage). Install Libraries

```
cd ring/src
./installdepmac.sh
```

Build Ring (Compiler/VM)

```
./buildclang.sh
```

Build Ring2EXE

```
cd ../ring2exe
sudo ./buildring2exe.sh
```

Build RingODBC

```
cd ../extensions/ringodbc
./buildclang.sh
```

Build RingMySQL

```
cd ../extensions/ringmysql
./buildclang.sh
```

Build RingSQLite

```
cd ../extensions/ringsqlite
./buildclang.sh
```

Build RingOpenSSL

```
cd ../extensions/ringopenssl
./buildclang.sh
```

Build RingInternet

```
cd ../extensions/ringinternet
./buildclang.sh
```

Build RingMurmurHash

```
cd ../extensions/ringmurmurhash
./buildclang.sh
```

Generate RingConsoleColors Source Code and Build

```
cd ../extensions/ringconsolecolors
./gencode.sh
./buildclang.sh
```

Generate RingAllegro Source Code and Build

```
cd ../extensions/ringallegro
./gencode.sh
./buildclang.sh
```

Generate RingLibCurl Source Code and Build

```
cd ../extensions/ringcurl
./gencode.sh
./buildclang.sh
```

Generate RingZip Source Code and Build

```
cd ../extensions/ringzip
./gencode.sh
./buildclang.sh
```

Generate RingFreeGLUT Source Code and Build

```
cd ../extensions/ringfreeglut
./gencode.sh
./buildclang.sh
```

Generate RingOpenGL Source Code and Build

The ringopengl folder contains many sub folders for different OpenGL versions Starting from OpenGL 1.1 to OpenGL 4.6

```
cd ../extensions/ringopengl/opengl21
./gencode.sh
./buildclang.sh
```

Generate RingQt Source Code and Build

```
cd ../extensions/ringqt
./gencode.sh
./buildclang.sh
```

To be able to call ring from any folder

```
cd ../../bin
sudo ./install.sh
```

Run Ring Notepad

```
cd applications/rnote
sudo ring rnote.ring
```

10.5 Building using CMake

Install libraries (MySQL Client, OpenSSL, LibCurl, Allegro 5 and Qt 5.5)

```
cmake .
make
```

HOW TO CONTRIBUTE?

Ring is a free-open source project, Everyone is welcome to contribute to Ring.

Project Home : <https://github.com/ring-lang/ring>

You can help in many parts in the project

- Documentation
- Testing
- Samples
- Applications
- Editors Support
- Libraries in Ring
- Extensions in C/C++
- Compiler and Virtual Machine (VM)
- Ideas and suggestions

11.1 Special thanks to contributors

Throughout the creation of this project, Ring relied heavily on contributions from experts along with college students. Their input was invaluable, and we want to take a moment to thank them and recognize them for all of their hard work.

Ring Team: <http://ring-lang.sf.net/team.html>

11.2 Documentation

You can modify anything in the documentation, by updating the text files (*.txt) in this folder : <https://github.com/ring-lang/ring/tree/master/docs/source>

The documentation is created using Sphinx : <http://www.sphinx-doc.org/en/stable/>

11.3 Testing

You can write new tests in this folder

<https://github.com/ring-lang/ring/tree/master/tests/scripts>

11.4 Samples

You can add new samples to this folder

<https://github.com/ring-lang/ring/tree/master/samples/other>

11.5 Applications

You can add new applications to this folder

<https://github.com/ring-lang/ring/tree/master/applications>

11.6 Editors Support

You can help in supporting Ring in different code editors

Check the next folder

<https://github.com/ring-lang/ring/tree/master/editor>

11.7 Libraries in Ring

You can update and add libraries to this folder

<https://github.com/ring-lang/ring/tree/master/ringlibs>

11.8 Extensions in C/C++

You can add and update extensions in this folder

<https://github.com/ring-lang/ring/tree/master/extensions>

11.9 Compiler and Virtual Machine (VM)

- Source Code (C Language) : <https://github.com/ring-lang/ring/tree/master/src>
- Visual Source (PWCT) : <https://github.com/ring-lang/ring/tree/master/visualsrc>

11.10 Ideas and suggestions

You can share your ideas, suggestions and questions in this group

<https://groups.google.com/forum/#!forum/ring-lang>

GETTING STARTED - FIRST STYLE

12.1 Hello World

The next program prints the Hello World message on the screen (std-out).

```
see "Hello World"
```

12.2 Run the program

to run the program, save the code in a file, for example : hello.ring then from the command line or terminal, run it using Ring

```
ring hello.ring
```

12.3 Create Executable File

Using Ring2EXE we can create executable file for our application

```
ring2exe hello.ring -static
```

12.4 Not Case-Sensitive

Since the Ring language is not case-sensitive, the same program can be written in different styles

Tip: It's better to select one style and use it in all of the program source code

```
SEE "Hello World"
```

```
See "Hello World"
```

12.5 Multi-Line literals

Using Ring we can write multi-line literal, see the next example

```
See "
    Hello
    Welcome to the Ring programming language
    How are you?

"
```

Also you can use the `nl` variable to insert new line and you can use the `+` operator to concatenate strings

As we have `NL` for new lines, we have `Tab` and `CR` (Carriage return) too!

Note: `nl` value means a new line and the actual codes that represent a newline is different between operating systems

```
See "Hello" + nl + "Welcome to the Ring programming language" +
    nl + "How are you?"
```

12.6 Getting Input

You can get the input from the user using the `give` command

```
See "What is your name? "
Give cName
See "Hello " + cName
```

12.7 No Explicit End For Statements

You don't need to use `;` or press `ENTER` to separate statements. The previous program can be written in one line.

```
See "What is your name? " give cName see "Hello " + cName
```

12.8 Using `?` to print expression then new line

It's common to print new line after printing an expression, We can use the `?` operator to do that!

Example:

```
? "Hello, World!"
for x = 1 to 10
    ? x
next
```

Output:

```
Hello, World!
1
2
3
4
5
6
7
8
```


9
10

12.9 Writing Comments

We can write one line comments and multi-line comments

The comment starts with # or //

Multi-lines comments are written between /* and */

```
/*
    Program Name : My first program using Ring
    Date        : 2016.09.09
    Author       : Mahmoud Fayed
*/

See "What is your name? "      # print message on screen
give cName                    # get input from the user
see "Hello " + cName           # say hello!

// See "Bye!"
```

Note: Using // to comment a lines of code is just a code style.

GETTING STARTED - SECOND STYLE

13.1 Hello World

The next program prints the Hello World message on the screen (std-out).

```
put "Hello World"
```

13.2 Run the program

to run the program, save the code in a file, for example : hello.ring then from the command line or terminal, run it using Ring

```
ring hello.ring
```

13.3 Create Executable File

Using Ring2EXE we can create executable file for our application

```
ring2exe hello.ring -static
```

13.4 Not Case-Sensitive

Since the Ring language is not case-sensitive, the same program can be written in different styles

Tip: It's better to select one style and use it in all of the program source code

```
PUT "Hello World"
```

```
Put "Hello World"
```

13.5 Multi-Line literals

Using Ring we can write multi-line literal, see the next example

```
Put "
    Hello
    Welcome to the Ring programming language
    How are you?

"
```

Also you can use the `nl` variable to insert new line and you can use the `+` operator to concatenate strings

As we have `NL` for new lines, we have `Tab` and `CR` (Carriage return) too!

Note: `nl` value means a new line and the actual codes that represent a newline is different between operating systems

```
Put "Hello" + nl + "Welcome to the Ring programming language" +
    nl + "How are you?"
```

13.6 Getting Input

You can get the input from the user using the `get` command

```
Put "What is your name? "
Get cName
Put "Hello " + cName
```

13.7 No Explicit End For Statements

You don't need to use `;` or press `ENTER` to separate statements. The previous program can be written in one line.

```
Put "What is your name? " get cName put "Hello " + cName
```

13.8 Writing Comments

We can write one line comments and multi-line comments

The comment starts with `#` or `//`

Multi-lines comments are written between `/*` and `*/`

```
/*
    Program Name : My first program using Ring
    Date        : 2016.09.09
    Author       : Mahmoud Fayed
*/

Put "What is your name? "      # print message on screen
get cName                     # get input from the user
put "Hello " + cName           # say hello!

// Put "Bye!"
```

Note: Using `//` to comment a lines of code is just a code style.

GETTING STARTED - THIRD STYLE

14.1 Hello World

The next program prints the Hello World message on the screen (std-out).

```
load "stdlib.ring"
print("Hello World")
```

14.2 Run the program

to run the program, save the code in a file, for example : hello.ring then from the command line or terminal, run it using Ring

```
ring hello.ring
```

14.3 Create Executable File

Using Ring2EXE we can create executable file for our application

```
ring2exe hello.ring -static
```

The -static option will avoid the need to ring.dllring.solring.dylib

But since the stdlib.ring load libraries like (LibCurl, OpenSSL, MySQL, etc)

You will need these libraries!

To avoid the need to these libraries (If you don't need stdlib classes)

Use stdlibcore.ring instead of stdlib.ring as in the next example

```
load "stdlibcore.ring"
print("Hello World")
```

Using stdlibcore.ring You can access the stdlib functions but not the stdlib classes.

if you want to use stdlib.ring and distribute your application

```
ring2exe hello.ring -dist -allruntime -noqt -noallegro
```

14.4 Not Case-Sensitive

Since the Ring language is not case-sensitive, the same program can be written in different styles

Tip: It's better to select one style and use it in all of the program source code

```
LOAD "stdlib.ring"
PRINT("Hello World")
```

```
Load "stdlib.ring"
Print("Hello World")
```

14.5 Multi-Line literals

Using Ring we can write multi-line literal, see the next example

```
Load "stdlib.ring"
Print("
    Hello
    Welcome to the Ring programming language
    How are you?

")
```

Also you can use the `\n` to insert new line and you can use `#{variable_name}` to insert variables values.

```
Load "stdlib.ring"
Print("Hello\nWelcome to the Ring programming language\nHow are you?")
```

14.6 Getting Input

You can get the input from the user using the `getstring()` function

```
Load "stdlib.ring"
Print("What is your name? ")
cName = GetString()
Print("Hello #{cName}")
```

14.7 No Explicit End For Statements

You don't need to use `;` or press ENTER to separate statements. The previous program can be written in one line.

```
Load "stdlib.ring"
Print("What is your name? ") cName=getstring() print("Hello #{cName}")
```

14.8 Writing Comments

We can write one line comments and multi-line comments

The comment starts with # or //

Multi-lines comments are written between /* and */

```
/*
    Program Name : My first program using Ring
    Date        : 2016.09.09
    Author       : Mahmoud Fayed
*/

Load "stdlib.ring"

Print("What is your name? ")    # print message on screen
cName=GetString()              # get input from the user
print("Hello #{cName}")         # say hello!

// print("Bye!")
```

Note: Using // to comment a lines of code is just a code style.

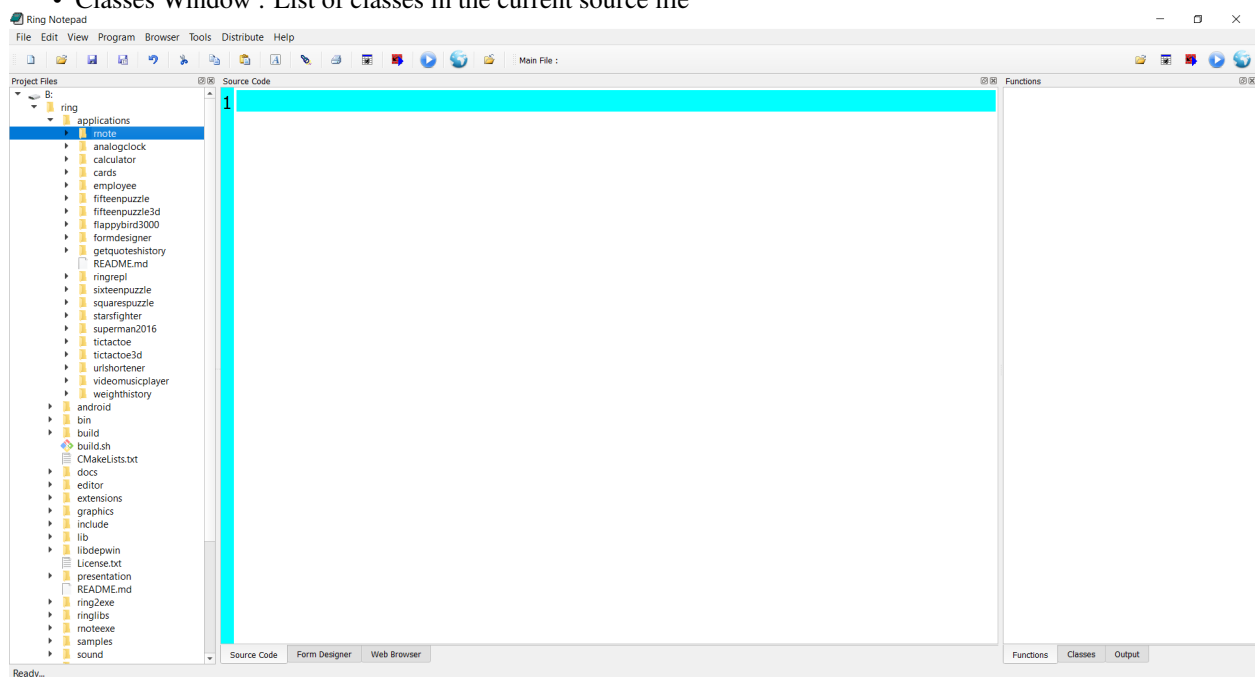
USING RING NOTEPAD

In this chapter we will learn about using Ring Notepad to write and execute Ring programs quickly
Ring Notepad is just a simple application developed using the Ring language.

15.1 Ring Notepad - Main Window

When we run the Ring Notepad we get the next dockable windows

- Project Files Window : where we can select and open any ring file (*.ring) quickly.
- Source Code Window : Where we write the source code.
- Form Designer Window : The Form Designer to create GUI application forms.
- Web Browser Window : Where we read the documentation or quickly open any website.
- Output Window : Output when we run programs that print to the standard output
- Function Window : List of functions in the current source file
- Classes Window : List of classes in the current source file

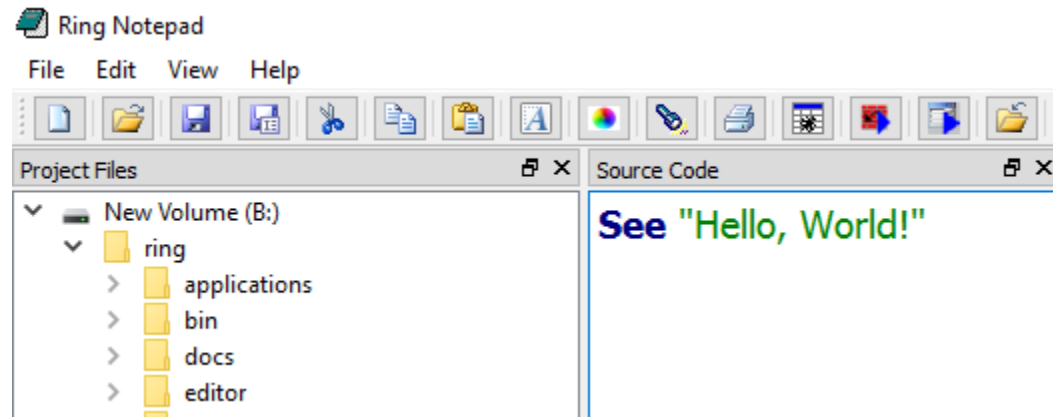


15.2 Creating and running your first Console Application

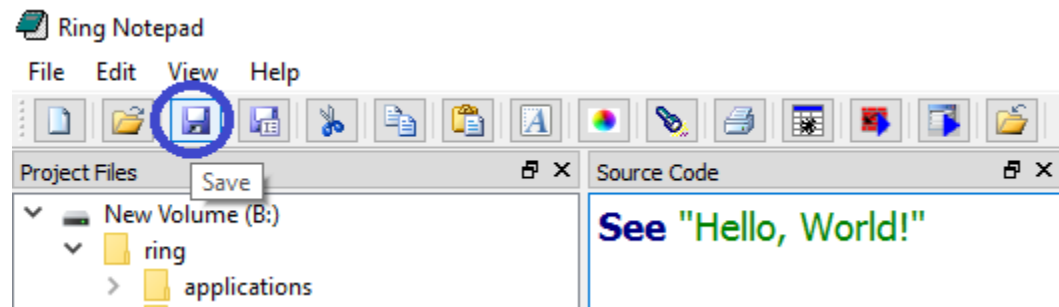
At first we will type the source code

```
See "Hello, World!"
```

As in the next image



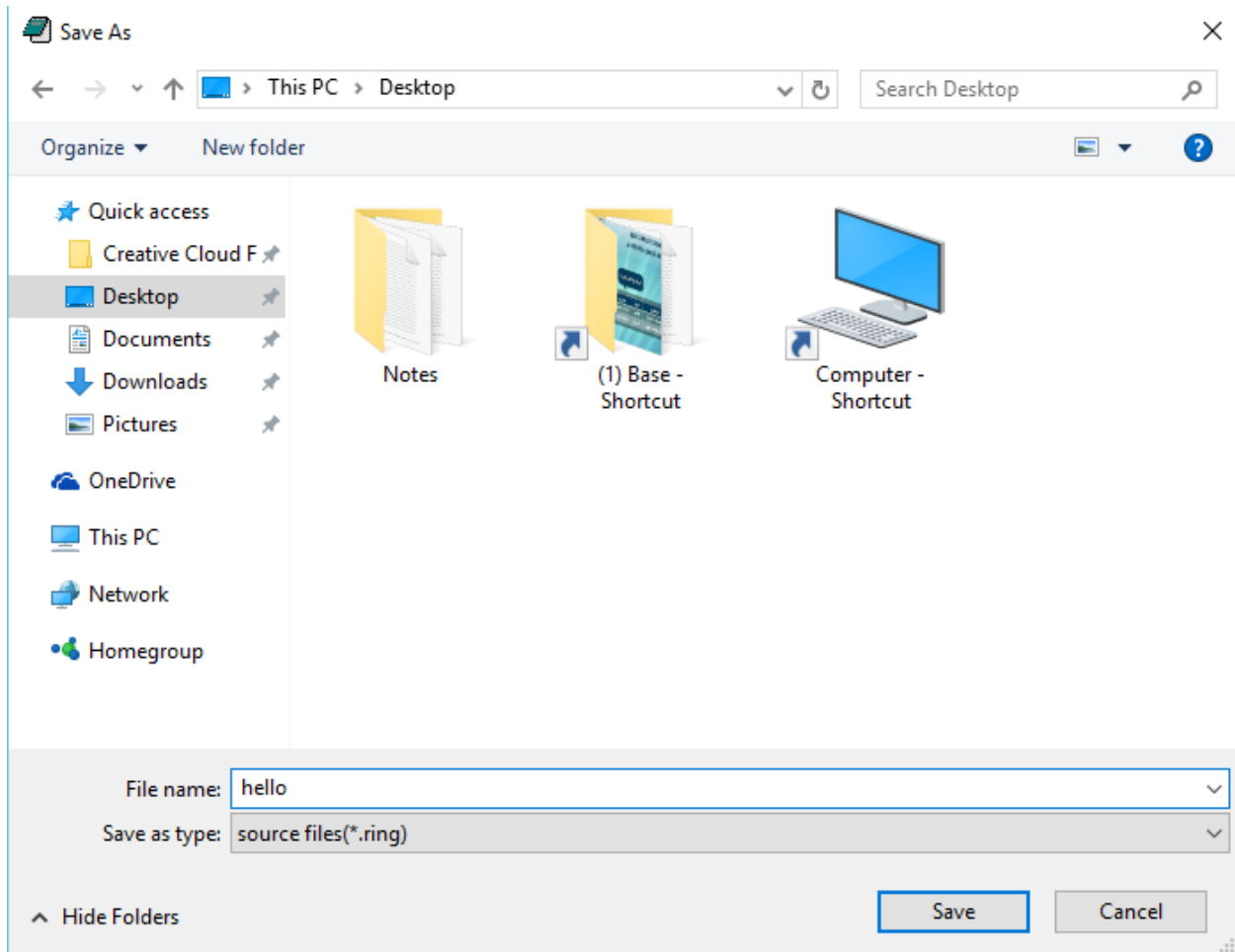
Then we will click on the “Save” button from the toolbar (or press CTRL+S)



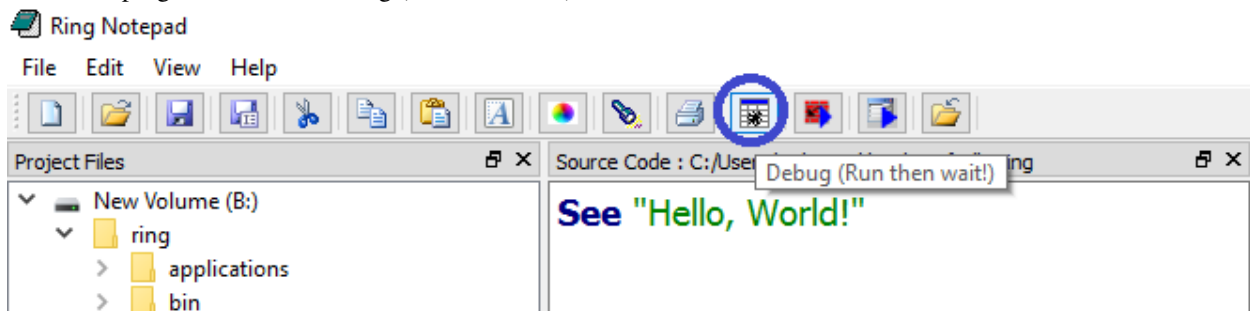
Determine the source code file name and location.

For example type : hello

This will create a new source code file called : hello.ring

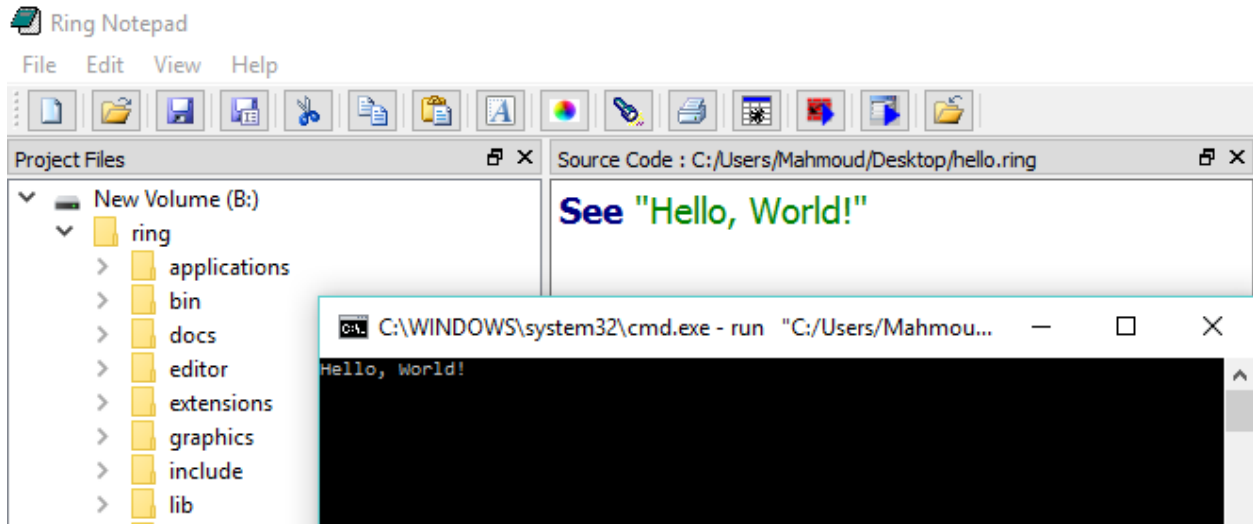


To run the program click on “Debug (Run then wait!)” button from the toolbar



The next screen shot present the application during the runtime

Press Enter to continue and return to the Ring Notepad.



15.3 Creating and running your first GUI/Mobile Application

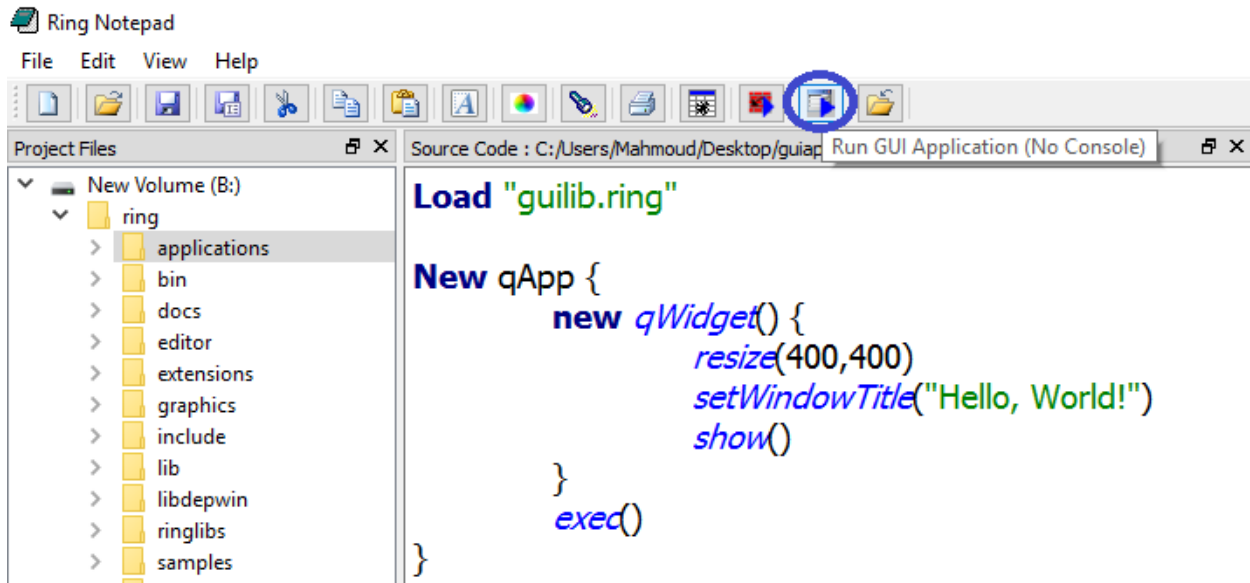
To learn how to create GUI applications using Ring check the “Desktop and Mobile development using RingQt” chapter.

Source Code:

```
Load "guilib.ring"

New qApp {
    new QWidget() {
        resize(400,400)
        setWindowTitle("Hello, World!")
        show()
    }
    exec()
}
```

In Ring notepad we have a special button to run GUI applications without displaying the console window.



The next screen shot present the application during the runtime



15.4 Creating and running your first Web Application

To learn how support Ring in your web server and how to create web applications using Ring check the “Web Development (CGI Library)” chapter.

Note: You need to support the Ring language in your web server to be able to run the next example.

Source Code:

```
#!b:\ring\bin\ring.exe -cgi

load "weblib.ring"

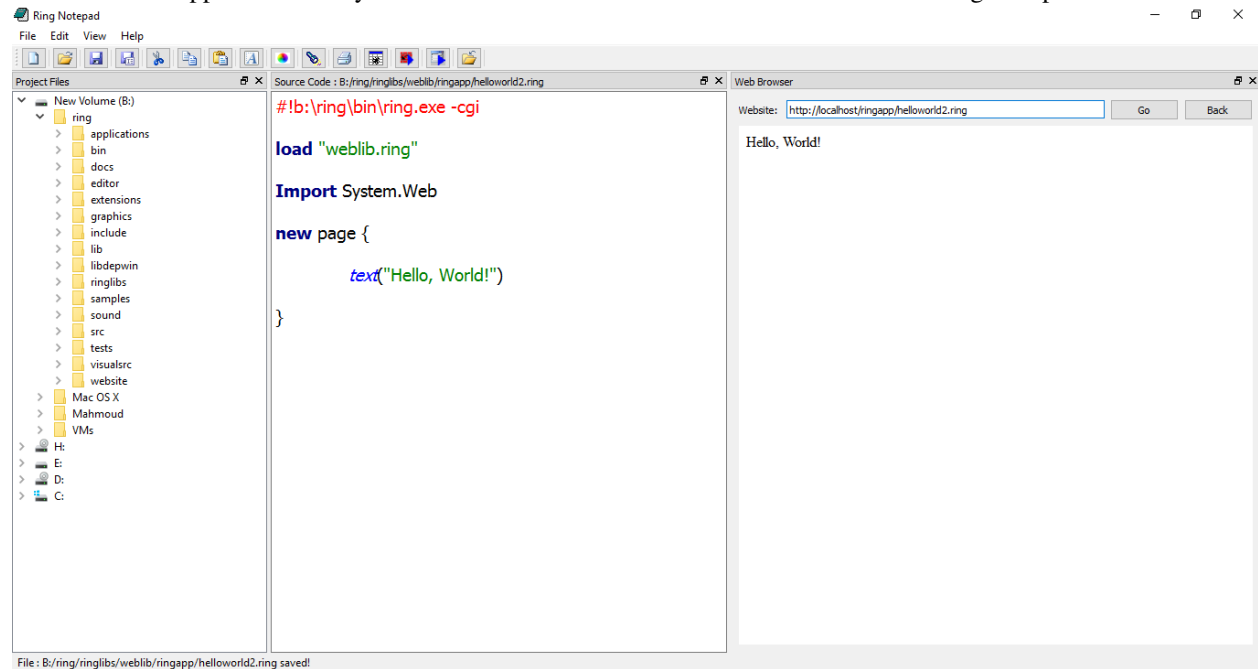
Import System.Web

new page {

    text("Hello, World!")

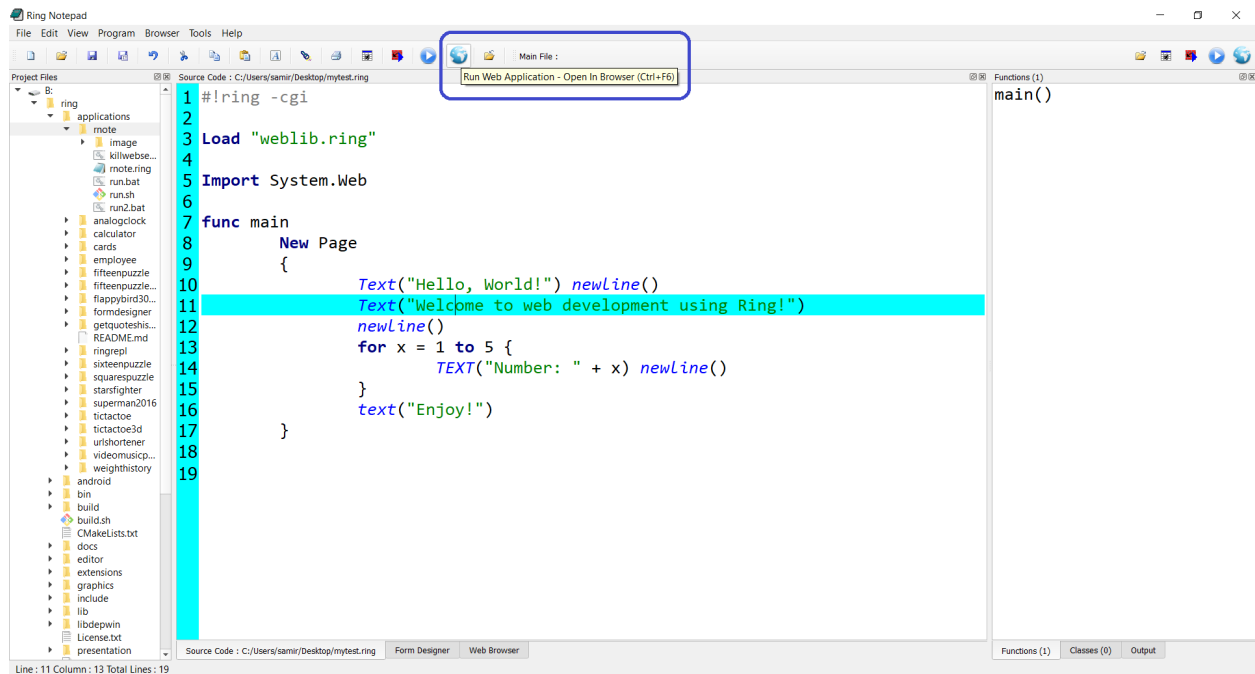
}
```

We can run the application in any web browser or in the browser that are embedded in Ring Notepad.



For Windows users, Ring 1.6 comes with Apache Web server!

We can run any web application from any folder directly without doing any configuration.



15.5 Creating and running your first Desktop/Mobile Game

To learn about creating 2D Games using Ring check the “Demo Project - Game Engine for 2D Games” chapter.

Source Code:

```

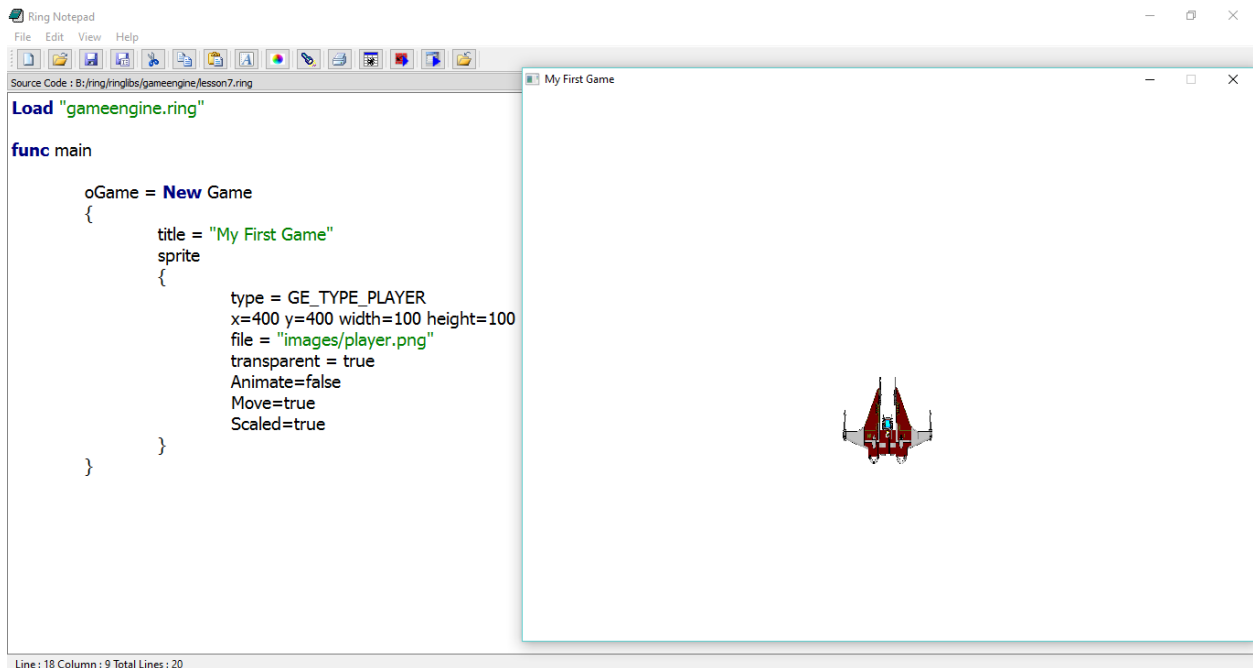
Load "gameengine.ring"

func main

    oGame = New Game
    {
        title = "My First Game"
        sprite
        {
            type = GE_TYPE_PLAYER
            x=400 y=400 width=100 height=100
            file = "images/player.png"
            transparent = true
            Animate=false
            Move=true
            Scaled=true
        }
    }

```

We can run the application as any GUI application.



15.6 The Main File in the Project

The idea of the Main File ToolBar is to determine the main file in the project. When the project contains many source code files.

Using this feature we can run the project (Main File) at any time while opening other files in the project without the need to switch to the Main File to run the project.

To quickly use this feature

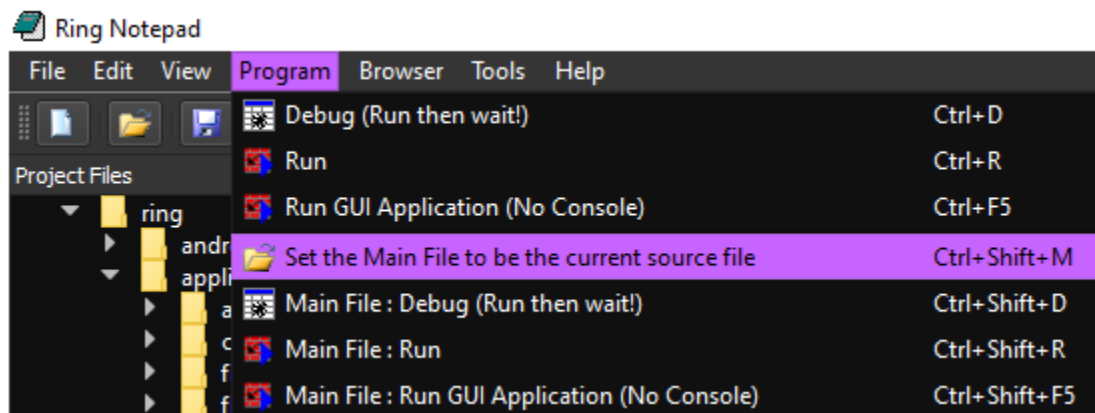
(Open the project main file)

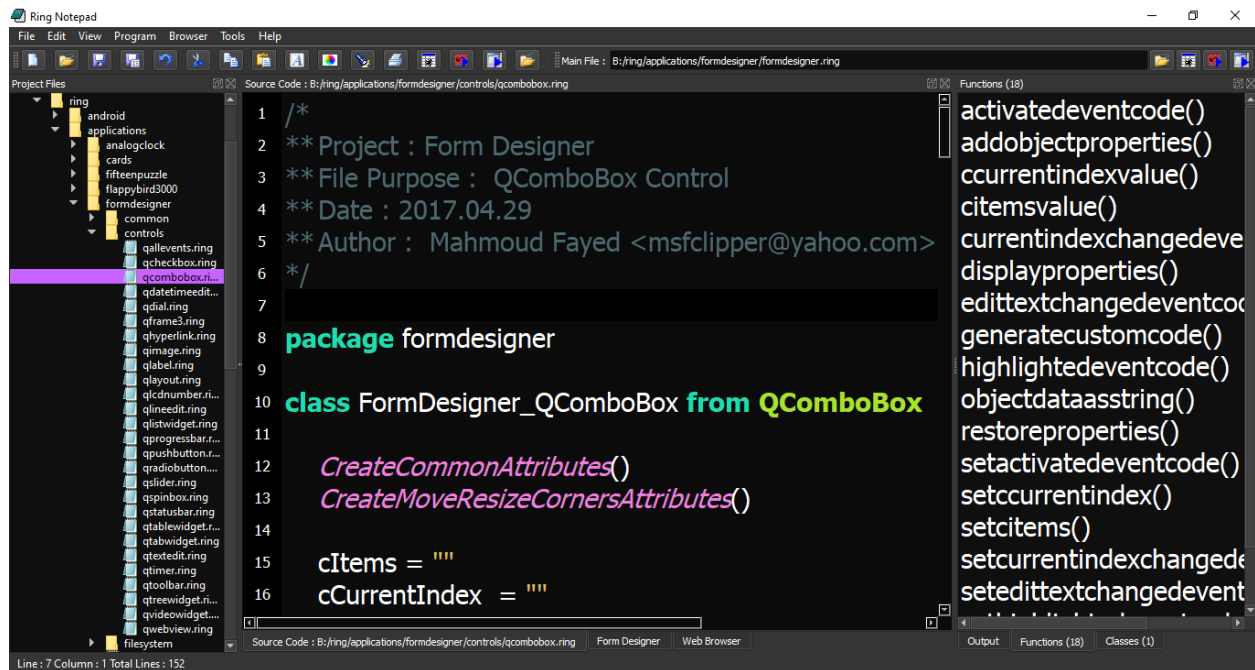
Press **Ctrl+Shift+M** to set the current source code file as the main file.

Open and modify other source code files in the project.

To run the project (Main File) at any time press **Ctrl+Shift+F5** (GUI) or **Ctrl+Shift+D** (Console).

Screen Shots:

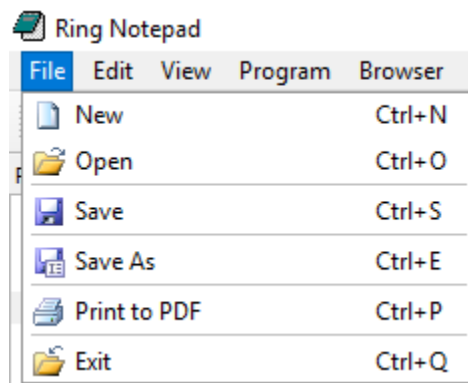




15.7 The File Menu

From this menu we can create, open and save the source code files.

Another feature in this menu is “Print to PDF”



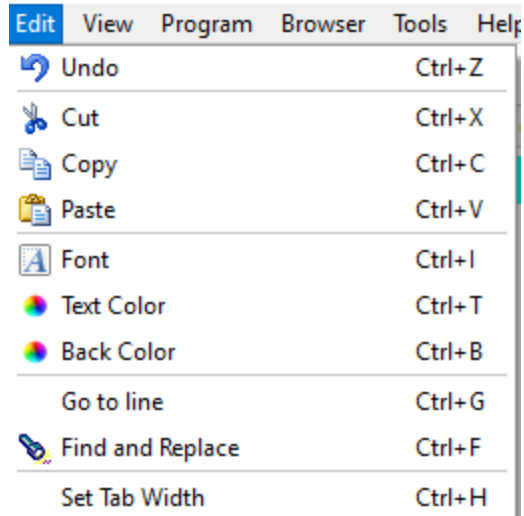
15.8 The Edit Menu

From the Edit menu we can Cut, Copy and Paste text.

Also we can change the font and the colors.

We can Go to a specific line or use the Find and Replace window to find and replace text.

Also We can set the Tab Width (Number of Spaces)

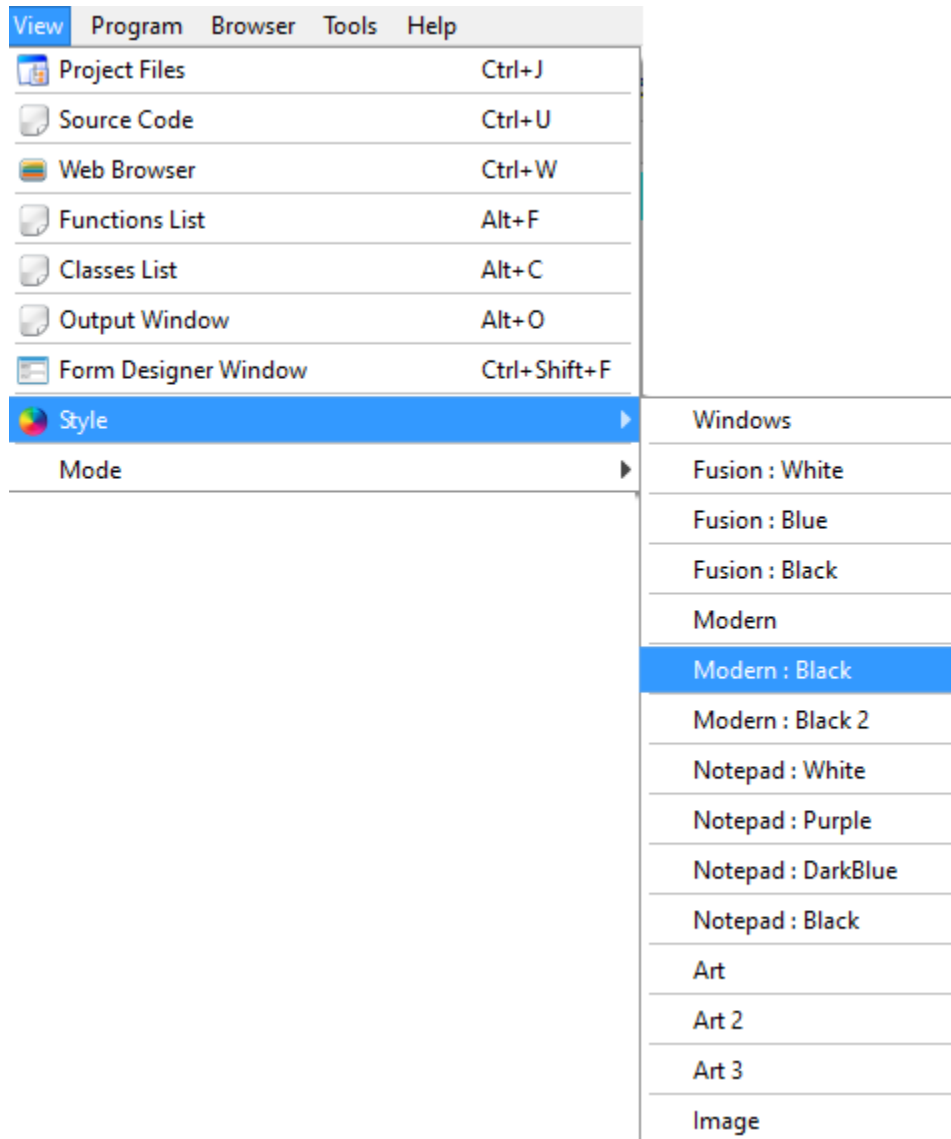


15.9 The View Menu

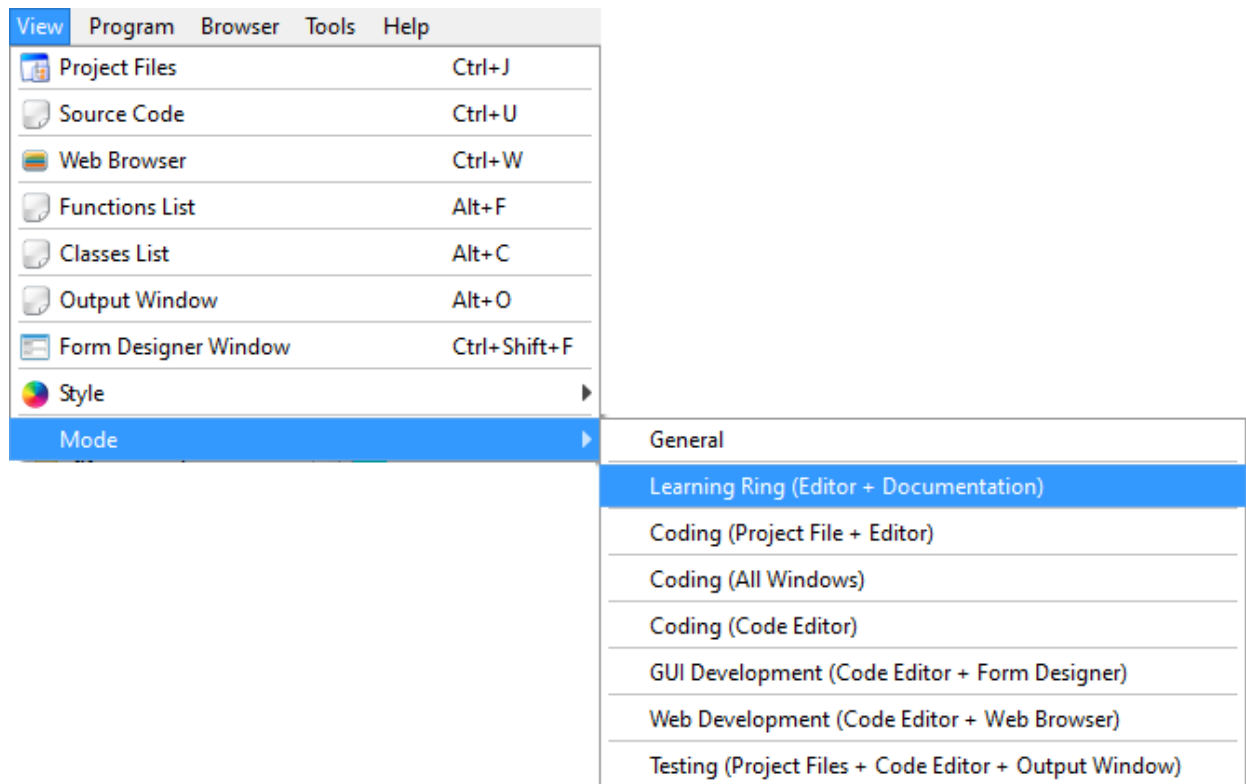
From this menu we can show/hide the dockable windows

Also we can change the Style of the Ring Notepad

Common Styles are (Fusion White and Modern Black)



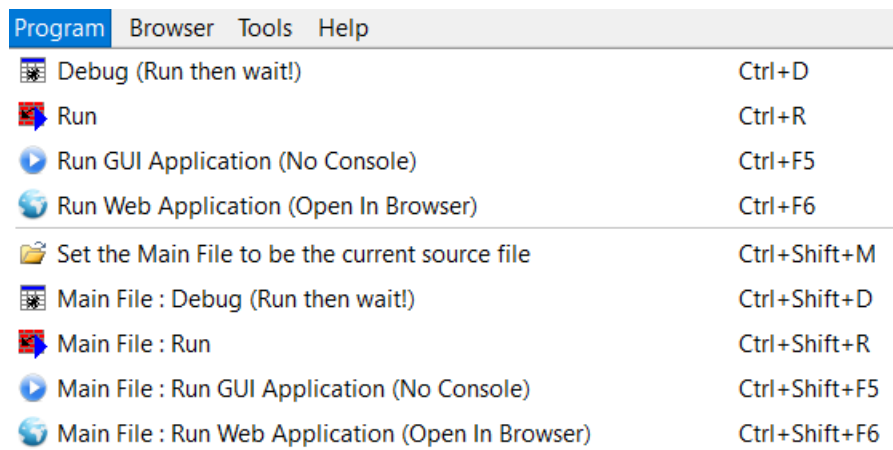
Also we can quickly show/hide group of dockable windows based on the context



15.10 The Program Menu

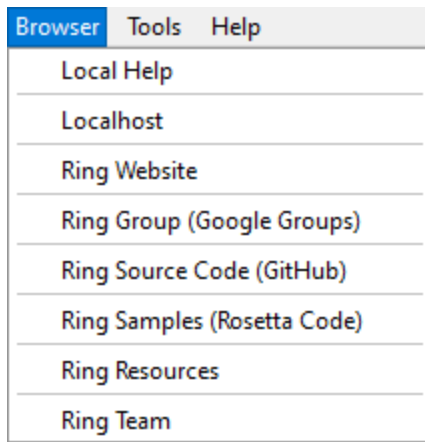
From this menu we can run the programs

Also we can set the Main file in the project



15.11 The Browser Menu

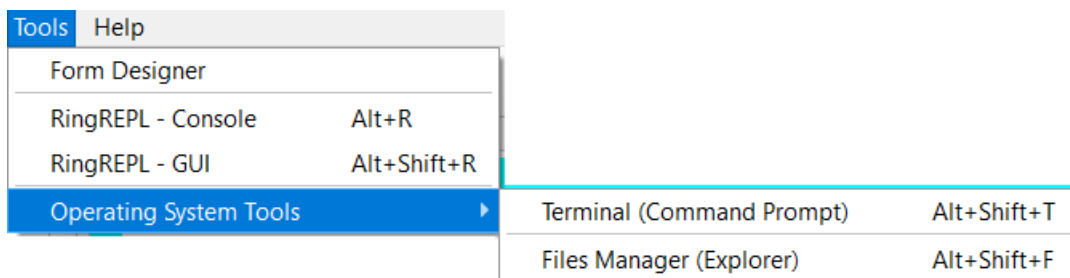
From this menu we can quickly open common links in the browser



15.12 The Tools Menu

From this menu we can run the Form Designer in separate window

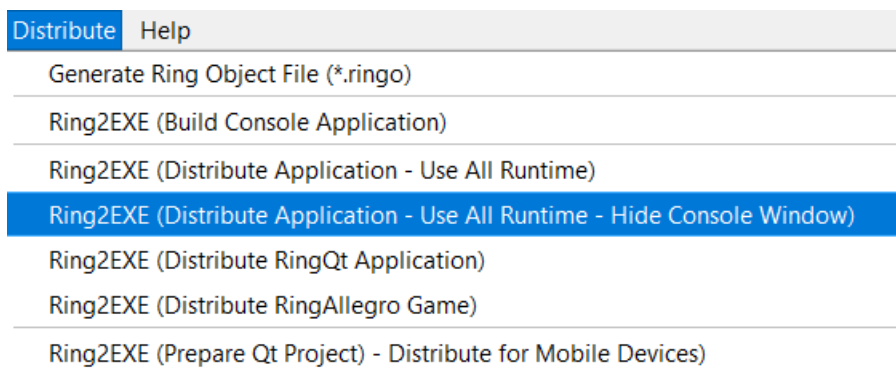
Also we can run the REPL (Read-Eval-Print-Loop) application



15.13 The Distribute Menu

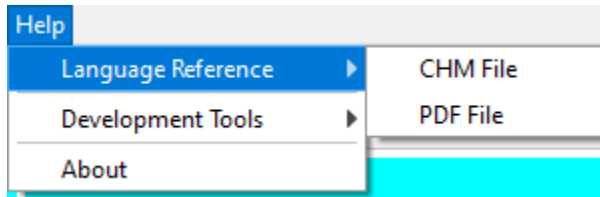
From this menu we can build an executable file for the application

Also we can prepare the application for distribution



15.14 The Help Menu

From this menu we can get the help files (CHM & PDF)



USING OTHER CODE EDITORS

In the Ring/Editor folder you will find extensions for the next editors

- Notepad++
- Geany
- Atom
- Sublime Text 2
- Visual Studio IDE
- Emacs

16.1 Using Notepad++

Folder : ring/editor/notepad_plus_plus

- Open Notepad++
- Open the “Language” menu
- Select “Define your language...”
- Click “Import...”
- select *Ring.xml*
- Select “OK” on the “Import successful” dialog and close the “User Defined Language” dialog/panel
- You may need to restart notepad++

```

1 Load "guilib.ring"
2
3 cActiveFileName = ""
4 aTextColor = [0,0,0]
5 aBackColor = [255,255,255]
6 cFont = "MS Shell Dlg 2,14,-1,5,50,0,0,0,0,0"
7 oDir = new QDir() #comment
8 cWebsite = "file://" + oDir.CurrentPath() + "/../html/index.html"
9 #cWebsite = "http://www.google.com"
10 //this is a comment
11 oSearch = NULL
12 oSearchValue = NULL
13 oSearchCase = NULL
14 oSearchFilter = NULL
15 oReplaceValue = NULL
16
17 lAskToSave = false
18
19 MyApp = New QApplication {
20     win1 = new QMainWindow() {
21
22         setWindowTitle("Ring Notepad")
23         setGeometry(100,100,400,400)
24         aBtns = {
25             new QPushButton(win1) {
26                 setBtnImage(self,"image/new.png")
27                 setClickedEvent("pNew()")
28                 setToolTip("New File")
29             },
30             new QPushButton(win1) {
31                 setBtnImage(self,"image/open.png")
32                 setClickedEvent("pOpen()")
33                 setToolTip("Open File")
34             },
35             new QPushButton(win1) {
36                 setBtnImage(self,"image/save.png")
37                 setClickedEvent("pSave()")
38                 setToolTip("Save")
39             }
40         }
41     }
42 }

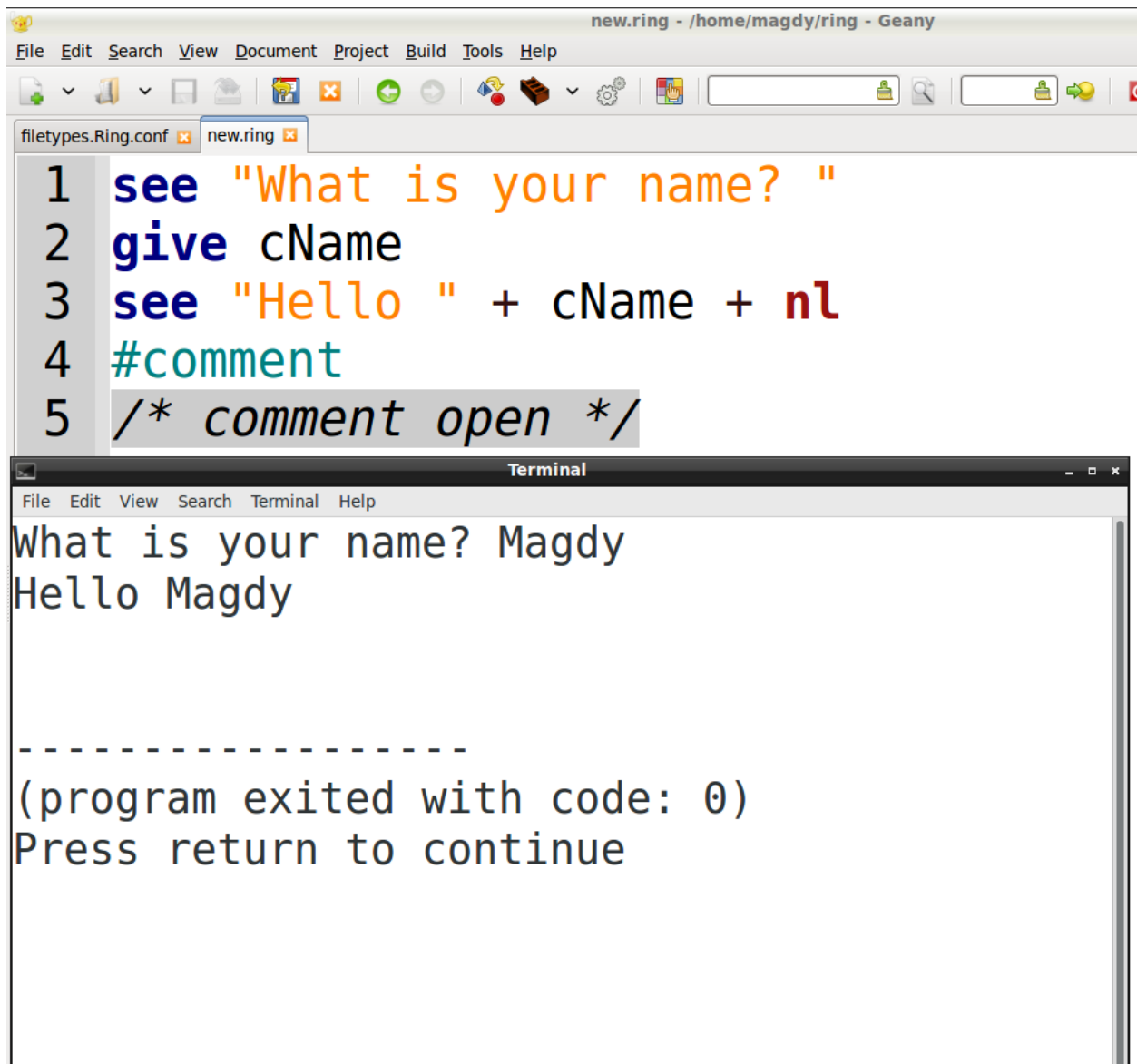
```

User Define File - Ring length: 19238 lines: 773 Ln: 17 Col: 19 Sel: 0 | 0 Dos/Windows UTF-8 INS

16.2 Using Geany

Folder : ring/editor/geany

- Run Geany editor
- Click on “Tools -> configuration files -> filetypes_extensions.conf” menu
- Add this line “Ring=*.ring;” without quotes after [Extensions]
- In unbuntu copy file “filetypes.Ring.conf” to folder “/home/USERNAME/filetypes.Ring.conf”
- You can run your files by pressing F5 button



The screenshot shows two windows. The top window is a Geany IDE titled 'new.ring - /home/magdy/ring - Geany'. It contains a Ring script in 'new.ring' with the following code:

```
1 see "What is your name? "  
2 give cName  
3 see "Hello " + cName + nl  
4 #comment  
5 /* comment open */
```

The bottom window is a terminal titled 'Terminal'. It shows the output of the Ring script:

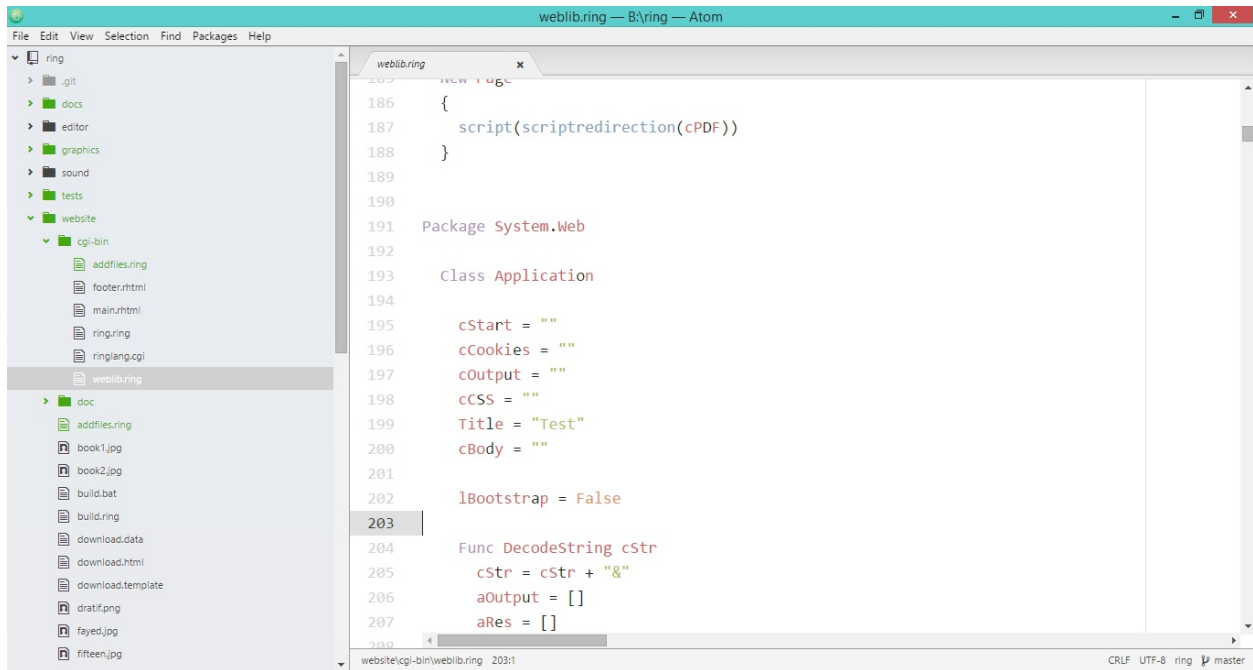
```
What is your name? Magdy  
Hello Magdy  
  
-----  
(program exited with code: 0)  
Press return to continue
```

16.3 Using Atom

Folder : ring/editor/atom

Just Copy the folder atom-language-ring to the next path

```
"C:\Users\{UserName}\.atom\Packages"
```



16.4 Using Sublime Text 2

Folder : ring/editor/sublime text 2

In the folder Sublime_Text_2 you will find the next three files

- 1 - ring.json-tmlanguage
- 2 - ring.sublime-build
- 3 - ring.tmlanguage

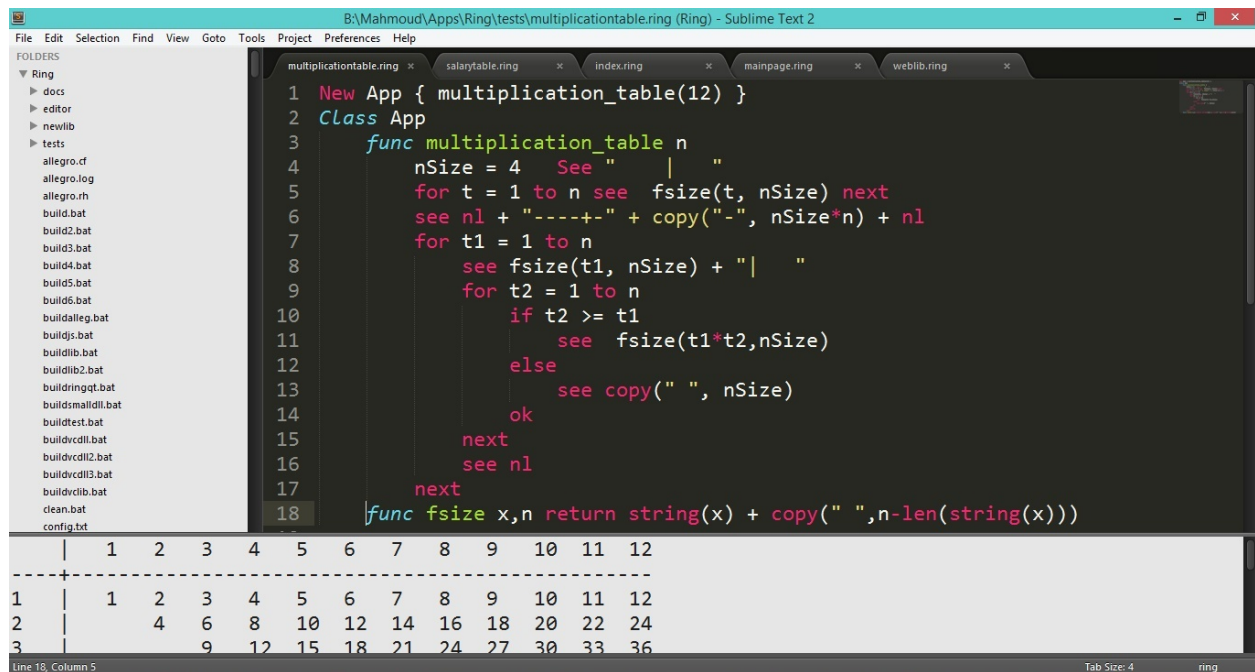
Just Copy the files to the next path

```
"C:\Users\{UserName}\AppData\Roaming\Sublime Text 2\Packages\User\"
```

The file ring.sublime-build includes the next line

```
"cmd": ["B:\\ring\\bin\\ring.exe", "$file"],
```

You can modify it according to the ring.exe path in your machine



```

1 New App { multiplication_table(12) }
2 Class App
3   func multiplication_table n
4     nSize = 4   See " | "
5     for t = 1 to n see fsize(t, nSize) next
6     see n1 + "----+-" + copy("-", nSize*n) + n1
7     for t1 = 1 to n
8       see fsize(t1, nSize) + " | "
9       for t2 = 1 to n
10        if t2 >= t1
11          see fsize(t1*t2,nSize)
12        else
13          see copy(" ", nSize)
14        ok
15      next
16    next
17  next
18  func fsize x,n return string(x) + copy(" ",n-len(string(x)))

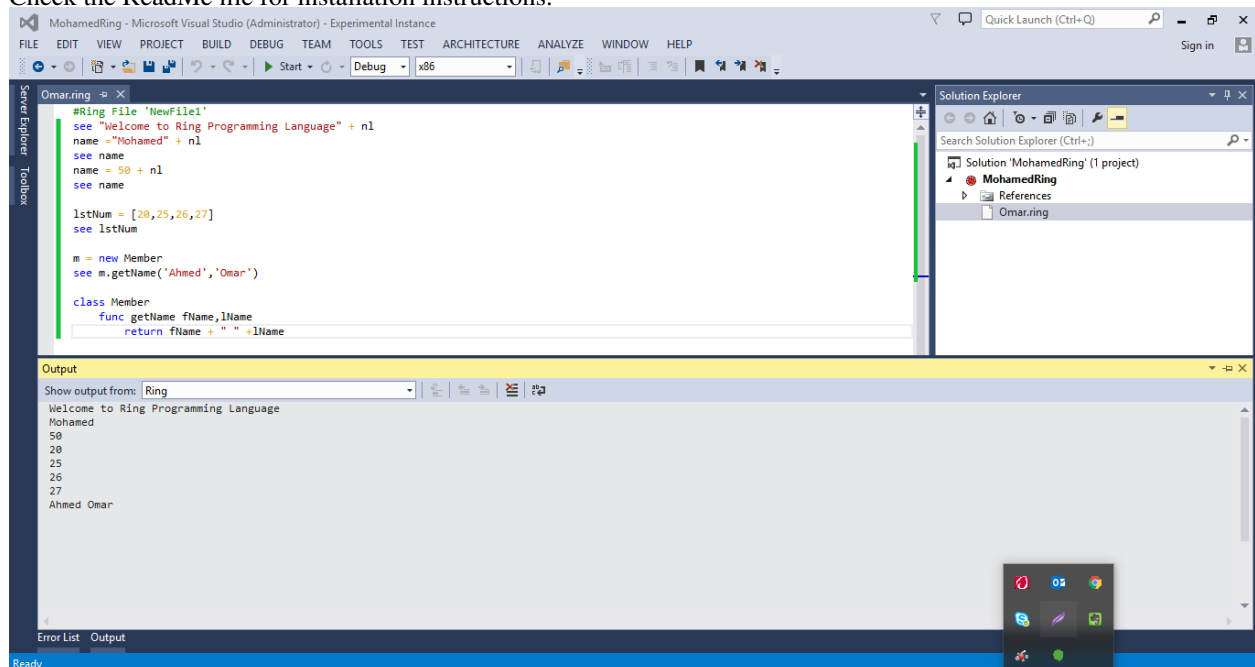
```

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	2	3	4	5	6	7	8	9	10	11	12
2		4	6	8	10	12	14	16	18	20	22	24
3			9	12	15	18	21	24	27	30	33	36

16.5 Using Visual Studio IDE

Folder : ring/editor/visualstudio

Check the ReadMe file for installation instructions.



```

#Ring File 'NewFile1'
see "Welcome to Ring Programming Language" + n1
name = "Mohamed" + n1
see name
name = 50 + n1
see name

lstNum = [20,25,26,27]
see lstNum

m = new Member
see m.getName('Ahmed','Omar')

class Member
  func getName fName,lName
  return fName + " " + lName

```

Output

```

Welcome to Ring Programming Language
Mohamed
50
20
25
26
27
Ahmed Omar

```

16.6 Using Emacs Editor

Folder : ring/editor/emacs

Check the ReadMe file for installation instructions.

Screen Shot:

A screenshot of the Emacs editor window. The title bar shows 'emacs@MAHMOUD-PC'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', and 'Help'. The main text area displays a Ring script for a game. The script defines a 'checkwin' function for 'ogame'. It checks if the game state is 'ok' or if enemies are zero. If enemies are zero, it sets the game result to true and enters a loop where it checks if the level is less than 30. If so, it displays 'Level Completed!' and calls 'ogame' again. If not, it displays 'You Win !!!'. The status bar at the bottom shows '-\--- game.ring 77% L304 Git-master (ring)'.

```
func checkwin ogame
  if oGameState.gameresult return ok
  if oGameState.enemies = 0
    oGameState.gameresult = true
    oGame {
      if oGameState.level < 30
        text {
          point = 400
          size = 30
          file = "fonts/pirulen.ttf"
          text = "Level Completed!"
          nStep = 3
          x = 500 y=10
          state = func ogame,oself {
            if oself.y >= 400
              ogame.shutdown = true
              oGameState.level++
              oGameState.enemies = oGameState.level
              oGameState.gameresult = false
            ok
          }
        }
      else
        text {
          point = 400
          size = 30
          nStep = 3
          file = "fonts/pirulen.ttf"
          text = "You Win !!!"
          x = 500 y=10
        }
    }
  }
```

VARIABLES

To create a new variable, you just need to determine the variable name & value. The value will determine the variable type and you can change the value to switch between the types using the same variable name.

Syntax:

```
<Variable Name> = <Value>
```

Tip: The operator '=' is used here as an Assignment operator and the same operator can be used in conditions, but for testing equality of expressions.

Note: The Variable will contains the real value (not a reference). This means that once you change the variable value, the old value will be removed from memory (even if the variable contains a list or object).

17.1 Dynamic Typing

Ring is a dynamic programming language that uses [Dynamic Typing](#).

```
x = "Hello"           # x is a string
see x + nl
x = 5                 # x is a number (int)
see x + nl
x = 1.2               # x is a number (double)
see x + nl
x = [1,2,3,4]         # x is a list
see x                 # print list items
x = date()            # x is a string contains date
see x + nl
x = time()            # x is a string contains time
see x + nl
x = true              # x is a number (logical value = 1)
see x + nl
x = false             # x is a number (logical value = 0)
see x + nl
```

17.2 Deep Copy

We can use the assignment operator '=' to copy variables. We can do that to copy values like strings & numbers. Also, we can copy complete lists & objects. The assignment operator will do a complete duplication for us. This operation called [Deep Copy](#)

```
list = [1,2,3,"four","five"]
list2 = list
list = []
See list          # print the first list - no items to print
See "*****" + nl
See list2         # print the second list - contains 5 items
```

17.3 Weakly Typed

Ring is a [weakly typed language](#), this means that the language can automatically convert between data types (like string & numbers) when that conversion make sense.

Rules:

```
<NUMBER> + <STRING> --> <NUMBER>
<STRING> + <NUMBER> --> <STRING>
```

Note: The same operator '+' can be used as an arithmetic operator or for string concatenation.

Example:

```
x = 10          # x is a number
y = "20"        # y is a string
sum = x + y     # sum is a number (y will be converted to a number)
Msg = "Sum = " + sum # Msg is a string (sum will be converted to a string)
see Msg + nl
```

OPERATORS

In this chapter we will introduce the operators provided by the Ring programming language.

18.1 Arithmetic Operators

The next table presents all of the arithmetic operators provided by the Ring language. Assume variable X=50 and variable Y=10 then:

Operator	Description	Example	Result
+	Add	x+y	60
-	Subtract	x-y	40
*	Multiplies	x*y	500
/	Divide	x/y	5
%	Modulus	x%y	0
++	Increment	x++	51
--	Decrement	x--	49

18.2 Relational Operators

The next table presents all of the relational operators provided by the Ring language. Assume variable X=50 and variable Y=10 then:

Operator	Description	Example	Result
=	Equal	x = y	False
!=	Not Equal	x != y	True
>	Greater than	x > y	True
<	Less than	x < y	False
>=	Greater or Equal	x >= y	True
<=	Less than or Equal	x <= y	False

18.3 Logical Operators

The next table presents all of the logical operators provided by the Ring language. Assume variable X=True and variable Y=False then:

Operator	Description	Example	Result
and	Logical AND	x and y	False
or	Logical OR	x or y	True
not	Logical Not	not x	False

Another style

Operator	Description	Example	Result
&&	Logical AND	x && y	False
	Logical OR	x y	True
!	Logical Not	! x	False

18.4 Bitwise Operators

The next table presents all of the bitwise operators provided by the Ring language. Assume variable X=8 and variable Y=2 then:

Operator	Description	Example	Result
&	Binary AND	x & y	0
	Binary OR	x y	10
^	Binary XOR	x ^ y	10
~	Binary Ones Complement	~x	-9
<<	Binary Left Shift	x << y	32
>>	Binary Right Shift	x >> y	2

18.5 Assignment Operators

The next table presents all of the assignment operators provided by the Ring language.

Assume variable X=8 then:

Operator	Description	Example	Result
=	Assignment	x = 10	x=10
+=	Add AND assignment	x += 5	x=13
-=	Subtract AND assignment	x -= 3	x=5
*=	Multiply AND assignment	x *= 2	x=16
/=	Divide AND assignment	x /= 3	x=2.67
%=	Modulus AND assignment	x %= 2	x=0
<<=	Left shift AND assignment	x <<= 2	x=32
>>=	Right shift AND assignment	x >>= 2	x=2
&=	Bitwise AND assignment	x &= 4	x=0
=	Bitwise OR and assignment	x = 3	x=11
^=	Bitwise XOR and assignment	x ^= 4	x=12

18.6 Misc Operators

Operator	Description
:literal	using : before identifier mean literal
Start:End	create list contains items from start to end
[list items]	define list items
list[index]	access list item
obj.name	using the dot operator to access object members (attributes/methods).
obj { stmts }	execute statements with direct access to object attributes & methods
func(para,...)	call function using parameters separated by comma
? <expr>	Print expression then new line

18.7 Operators Precedence

The next table present operators from higher precedence (Evaluated first) to lower precedence.

Operator
. [] () {}
- ~ :Literal [list items]
++ --
Start:End
* / %
+ -
<< >>
&
^
< > <= >=
= !=
not !
and or &&
Assignment = += -= *= /= %= >>= <<= &= ^= =
?

Example:

See 3+5*4 *# prints 23*

CONTROL STRUCTURES - FIRST STYLE

In this chapter we are going to learn about the control structures provided by the Ring programming language.

19.1 Branching

- If Statement

Syntax:

```
if Expression
    Block of statements
but Expression
    Block of statements
else
    Block of statements
ok
```

Example:

```
see "
    Main Menu
    -----
    (1) Say Hello
    (2) About
    (3) Exit

    " give nOption

if nOption = 1 see "Enter your name : " give name see "Hello " + name + nl
but nOption = 2 see "Sample : using if statement" + nl
but nOption = 3 bye
else see "bad option..." + nl
ok
```

- Switch Statement

Syntax:

```
switch Expression
on Expression
    Block of statements
other
    Block of statements
off
```


Example:

```
See "
    Main Menu
    -----
    (1) Say Hello
    (2) About
    (3) Exit

    " Give nOption

Switch nOption
On 1 See "Enter your name : " Give name See "Hello " + name + nl
On 2 See "Sample : using switch statement" + nl
On 3 Bye
Other See "bad option..." + nl
Off
```

19.2 Looping

- While Loop

Syntax:

```
while Expression
    Block of statements
end
```

Example:

```
While True

    See "
        Main Menu
        -----
        (1) Say Hello
        (2) About
        (3) Exit

        " Give nOption

    Switch nOption
    On 1
        See "Enter your name : "
        Give name
        See "Hello " + name + nl
    On 2
        See "Sample : using while loop" + nl
    On 3
        Bye
    Other
        See "bad option..." + nl
    Off

End
```

- For Loop

Syntax:

```
for identifier=expression to expression [step expression]
    Block of statements
next
```

Example:

```
# print numbers from 1 to 10
for x = 1 to 10 see x + nl next
```

Example:

```
# Dynamic loop
See "Start : " give nStart
See "End   : " give nEnd
See "Step  : " give nStep
For x = nStart to nEnd Step nStep
    see x + nl
Next
```

Example:

```
# print even numbers from 0 to 10
for x = 0 to 10 step 2
    see x + nl
next
```

Example:

```
# print even numbers from 10 to 0
for x = 10 to 0 step -2
    see x + nl
next
```

- For in Loop

Syntax:

```
for identifier in List/String [step expression]
    Block of statements
next
```

Example:

```
aList = 1:10 # create list contains numbers from 1 to 10
for x in aList see x + nl next # print numbers from 1 to 10
```

19.3 Using The Step option with For in

We can use the Step option with For in to skip number of items in each iteration

Example:

```
aList = 1:10 # create list contains numbers from 1 to 10
# print odd items inside the list
for x in aList step 2
    see x + nl
next
```

19.4 Using For in to modify lists

When we use (For in) we get items by reference.

This means that we can read/edit items inside the loop.

Example:

```
aList = 1:5      # create list contains numbers from 1 to 5
# replace list numbers with strings
for x in aList
    switch x
    on 1 x = "one"
    on 2 x = "two"
    on 3 x = "three"
    on 4 x = "four"
    on 5 x = "five"
    off
next
see aList      # print the list items
```

19.5 Do Again Loop

Syntax:

```
do
    Block of statements
again expression
```

Example:

```
x = 1
do
    see x + nl
    x++
again x <= 10
```

19.6 Exit Command

Used to go outside one or more of loops.

Syntax:

```
exit [expression]      # inside loop
```

Example:

```
for x = 1 to 10
    see x + nl
    if x = 5 exit ok
next
```

19.7 Exit from two loops

The next example presents how to use the exit command to exit from two loops in one jump.

Example:

```
for x = 1 to 10
  for y = 1 to 10
    see "x=" + x + " y=" + y + nl
    if x = 3 and y = 5
      exit 2      # exit from 2 loops
    ok
  next
next
```

19.8 Loop Command

Used to jump to the next iteration in the loop.

Syntax:

```
loop [expression]      # inside loop
```

Example:

```
for x = 1 to 10
  if x = 3
    see "Number Three" + nl
    loop
  ok
  see x + nl
next
```

19.9 Exit/Loop inside sub functions

While we are inside a loop, we can call a function then use the exit and/or loop command inside that function and the command will work on the outer loop.

Example:

```
# print numbers from 1 to 10 except number 5.

for x = 1 to 10
  ignore(x,5)
  see x + nl
next

func ignore x,y
  if x = y
    loop
  ok
```

19.10 Short-circuit evaluation

The logical operators and/or follow the [short-circuit evaluation](#).

If the first argument of the AND operator is zero, then there is no need to evaluate the second argument and the result will be zero.

If the first argument of the OR operator is one, then there is no need to evaluate the second argument and the result will be one.

Example:

```
/* output
** nice
** nice
** great
*/

x = 0 y = 10

if (x = 0 and nice()) and (y = 10 and nice())
    see "great" + nl
ok

func nice see "nice" + nl return 1
```

Example:

```
# No output

x = 0 y = 10

if (x = 1 and nice()) and (y = 10 and nice())
    see "great" + nl
ok

func nice see "nice" + nl return 1
```

Example:

```
/* output
** nice
** great
*/

x = 0 y = 10

if (x = 0 and nice()) or (y = 10 and nice())
    see "great" + nl
ok

func nice see "nice" + nl return 1
```

19.11 Comments about evaluation

- True, False, nl & NULL are variables defined by the language
- True = 1

- False = 0
- nl = new line
- NULL = empty string = ""
- Everything evaluates to true except 0 (False).

Example:

```
# output = message from the if statement

if 5      # 5 evaluates to true because it's not zero (0).
    see "message from the if statement" + nl
ok
```

CONTROL STRUCTURES - SECOND STYLE

In this chapter we are going to learn about the second style of control structures provided by the Ring programming language.

20.1 Branching

- If Statement

Syntax:

```
if Expression
    Block of statements
elseif Expression
    Block of statements
else
    Block of statements
end
```

Example:

```
put "
    Main Menu
    -----
    (1) Say Hello
    (2) About
    (3) Exit

    " get nOption

if nOption = 1 put "Enter your name : " get name put "Hello " + name + nl
elseif nOption = 2 put "Sample : using if statement" + nl
elseif nOption = 3 bye
else put "bad option..." + nl
end
```

- Switch Statement

Syntax:

```
switch Expression
case Expression
    Block of statements
else
    Block of statements
end
```

Example:

```
Put "
    Main Menu
    -----
    (1) Say Hello
    (2) About
    (3) Exit

    " Get nOption

Switch nOption
Case 1 Put "Enter your name : " Get name Put "Hello " + name + nl
Case 2 Put "Sample : using switch statement" + nl
Case 3 Bye
Else Put "bad option..." + nl
End
```

20.2 Looping

- While Loop

Syntax:

```
while Expression
    Block of statements
end
```

Example:

```
While True

    Put "
        Main Menu
        -----
        (1) Say Hello
        (2) About
        (3) Exit

        " Get nOption

    Switch nOption
    Case 1
        Put "Enter your name : "
        Get name
        Put "Hello " + name + nl
    Case 2
        Put "Sample : using while loop" + nl
    Case 3
        Bye
    Else
        Put "bad option..." + nl
    End
End
```

- For Loop

Syntax:


```

for identifier=expression to expression [step expression]
    Block of statements
end

```

Example:

```

# print numbers from 1 to 10
for x = 1 to 10 put x + nl end

```

Example:

```

# Dynamic loop
Put "Start : " get nStart
Put "End   : " get nEnd
Put "Step  : " get nStep
For x = nStart to nEnd Step nStep
    Put x + nl
End

```

Example:

```

# print even numbers from 0 to 10
for x = 0 to 10 step 2
    Put x + nl
end

```

Example:

```

# print even numbers from 10 to 0
for x = 10 to 0 step -2
    put x + nl
end

```

- For in Loop

Syntax:

```

for identifier in List/String [step expression]
    Block of statements
end

```

Example:

```

aList = 1:10    # create list contains numbers from 1 to 10
for x in aList put x + nl end # print numbers from 1 to 10

```

20.3 Exceptions

```

try
    Block of statements
catch
    Block of statements
end

```

CONTROL STRUCTURES - THIRD STYLE

In this chapter we are going to learn about the third style of control structures provided by the Ring programming language.

21.1 Branching

- If Statement

Syntax:

```
if Expression {  
    Block of statements  
elseif Expression  
    Block of statements  
else  
    Block of statements  
}
```

Example:

```
Load "stdlib.ring"  
  
print("  
    Main Menu  
    -----  
    (1) Say Hello  
    (2) About  
    (3) Exit  
    ")  
  
nOption = getnumber()  
  
if nOption = 1 {  
    print("Enter your name : ")  
    name = getstring()  
    print("Hello #{name}\n")  
elseif nOption = 2  
    print("Sample : using if statement\n")  
elseif nOption = 3  
    bye  
else  
    print("bad option...\n")  
}
```

- Switch Statement

Syntax:

```
switch Expression {
case Expression
    Block of statements
else
    Block of statements
}
```

Example:

```
Load "stdlib.ring"

print("
    Main Menu
    -----
    (1) Say Hello
    (2) About
    (3) Exit

")

nOption = GetString()

switch nOption {
case 1
    print("Enter your name : ")
    name = getstring()
    print("Hello #{name}\n")
case 2
    print("Sample : using switch statement\n")
case 3
    Bye
else
    print("bad option...\n")
}
```

21.2 Looping

- While Loop

Syntax:

```
while Expression {
    Block of statements
}
```

Example:

```
Load "stdlib.ring"

While True {

    print("
        Main Menu
        -----
```

```

        (1) Say Hello
        (2) About
        (3) Exit

        ")

    nOption = GetString()

    switch nOption {
    case 1
        print("Enter your name : ")
        name = getstring()
        print("Hello #{name}\n")

    case 2
        print("Sample : using switch statement\n")

    case 3
        Bye

    else
        print("bad option...\n")

    }
}

```

- For Loop

Syntax:

```

for identifier=expression to expression [step expression] {
    Block of statements
}

```

Example:

```

# print numbers from 1 to 10
load "stdlib.ring"
for x = 1 to 10 {
    print("#{x}\n")
}

```

Example:

```

load "stdlib.ring"

# Dynamic loop
print("Start : ") nStart = getnumber()
print("End   : ") nEnd = getnumber()
print("Step  : ") nStep = getnumber()
for x = nStart to nEnd step nStep {
    print("#{x}\n")
}

```

Example:

```

load "stdlib.ring"

# print even numbers from 0 to 10
for x = 0 to 10 step 2 {
    print("#{x}\n")
}

```

Example:

```
load "stdlib.ring"

# print even numbers from 10 to 0
for x = 10 to 0 step -2 {
    print("#{x}\n")
}
```

- For in Loop

Syntax:

```
for identifier in List/String [step expression] {
    Block of statements
}
```

Example:

```
load "stdlib.ring"

aList = 1:10    # create list contains numbers from 1 to 10
for x in aList { print("#{x}\n") } # print numbers from 1 to 10
```

Example:

```
load "stdlib.ring"

aList = 1:10    # create list contains numbers from 1 to 10
# print odd items inside the list
for x in aList step 2 {
    print("#{x}\n")
}
```

When we use (For in) we get items by reference.

This means that we can read/edit items inside the loop.

Example:

```
load "stdlib.ring"

aList = 1:5    # create list contains numbers from 1 to 5
# replace list numbers with strings
for x in aList {
    switch x {
        case 1 x = "one"
        case 2 x = "two"
        case 3 x = "three"
        case 4 x = "four"
        case 5 x = "five"
    }
}
print(aList)    # print the list items
```

21.3 Exceptions

```
try {  
    Block of statements  
catch  
    Block of statements  
}
```

GETTING INPUT

We can get input from the keyboard using

- The Give Command
- The GetChar() Function
- The Input() Function

22.1 Give Command

Syntax:

```
Give VariableName
```

Example:

```
See "Enter the first number : " Give nNum1
See "Enter the second number : " Give nNum2
See "Sum : " + ( 0 + nNum1 + nNum2 )
```

Output:

```
Enter the first number : 3
Enter the second number : 4
Sum : 7
```

22.2 GetChar() Function

We can get one character from the standard input using the GetChar() function

Syntax:

```
GetChar() ---> Character
```

Example:

```
While True
    See "
        Main Menu
        (1) Say Hello
        (2) Exit
    "
    Option = GetChar()
```

```
GetChar() GetChar()  # End of line

# the previous two lines can be replaced with the next line
# Give Option

if Option = 1
    see "Enter your name : " give cName
    see "Hello " + cName
else
    bye
ok
End
```

22.3 Input() Function

We can get input from the keyboard using the Input() function

Syntax:

```
Input(nCount) ----> string
```

The function will wait until nCount characters (at least) are read

Example:

```
See "Enter message (30 characters) : " cMsg = input(30)
See "Message : " + cMsg
```


FUNCTIONS - FIRST STYLE

In this chapter we are going to learn about the next topics :-

- Define functions
- Call functions
- Declare parameters
- Send parameters
- Main Function
- Variables Scope
- Return Value
- Recursion

23.1 Define Functions

To define new function

Syntax:

```
func <function_name> [parameters]
    Block of statements
```

Note: No keyword is required to end the function definition.

Example:

```
func hello
    see "Hello from function" + nl
```

23.2 Call Functions

To call function without parameters, we type the function name then ()

Tip: We can call the function before the function definition and the function code.

Example:

```
hello()

func hello
    see "Hello from function" + nl
```

Example:

```
first()  second()

func first    see "message from the first function" + nl
func second  see "message from the second function" + nl
```

23.3 Declare parameters

To declare the function parameters, after the function name type the list of parameters as a group of identifiers separated by comma.

Example:

```
func sum x,y
    see x+y+nl
```

23.4 Send Parameters

To send parameters to function, type the parameters inside () after the function name

Syntax:

```
funcname(parameters)
```

Example:

```
/* output
** 8
** 3000
*/

sum(3,5)  sum(1000,2000)

func sum x,y see x+y+nl
```

23.5 Main Function

Using the Ring programming language, the Main Function is optional, when it's defined, it will be executed after the end of other statements.

if no other statements comes alone, the main function will be the first [entry point](#)

Example:

```
# this program will print the hello world message first then execute the main function

See "Hello World!" + nl

func main
    see "Message from the main function" + nl
```

23.6 Variables Scope

The Ring programming language uses **lexical scoping** to determine the scope of a variable.

Variables defined inside functions (including function parameters) are local variables. Variables defined outside functions (before any function) are global variables.

Inside any function we can access the variables defined inside this function beside the global variables.

Example:

```
# the program will print numbers from 10 to 1

x = 10                                # x is a global variable.

func main

    for t = 1 to 10                    # t is a local variable
        mycounter()                  # call function
    next

func mycounter

    see x + nl                        # print the global variable value
    x--                               # decrement
```

Note: Using the main function before the for loop declare the t variable as a local variable, It's recommended to use the main functions instead of typing the instructions directly to set the scope of the new variables to local.

23.7 Return Value

The function can return a value using the Return command.

Syntax:

```
Return [Expression]
```

Tip: the Expression after the return command is optional and we can use the return command to end the function execution without returning any value.

Note: if the function doesn't return explicit value, it will return NULL (empty string = "").

Example:

```
if novalue() = NULL
    See "the function doesn't return a value" + nl
ok

func novalue
```

23.8 Recursion

The Ring programming language support [Recursion](#) and the function can call itself using different parameters.

Example:

```
see fact(5)      # output = 120

func fact x if x = 0 return 1 else return x * fact(x-1) ok
```

FUNCTIONS - SECOND STYLE

In this chapter we are going to learn about the next topics :-

- Define functions
- Call functions
- Declare parameters
- Send parameters
- Main Function
- Variables Scope
- Return Value
- Recursion

24.1 Define Functions

To define new function

Syntax:

```
def <function_name> [parameters]
    Block of statements
[end]
```

Note: the keyword 'end' is optional.

Example:

```
def hello
    put "Hello from function" + nl
end
```

24.2 Call Functions

To call function without parameters, we type the function name then ()

Tip: We can call the function before the function definition and the function code.

Example:

```
hello()

def hello
    put "Hello from function" + nl
end
```

Example:

```
first()  second()

def first    put "message from the first function" + nl
def second  put "message from the second function" + nl
```

24.3 Declare parameters

To declare the function parameters, after the function name type the list of parameters as a group of identifiers separated by comma.

Example:

```
def sum x,y
    put x+y+nl
end
```

24.4 Send Parameters

To send parameters to function, type the parameters inside () after the function name

Syntax:

```
funcname(parameters)
```

Example:

```
/* output
** 8
** 3000
*/

sum(3,5) sum(1000,2000)

def sum x,y put x+y+nl
```

24.5 Main Function

Using the Ring programming language, the Main Function is optional, when it's defined, it will be executed after the end of other statements.

if no other statements comes alone, the main function will be the first [entry point](#)

Example:

```
# this program will print the hello world message first then execute the main function
put "Hello World!" + nl

def main
    put "Message from the main function" + nl
end
```

24.6 Variables Scope

The Ring programming language uses **lexical scoping** to determine the scope of a variable.

Variables defined inside functions (including function parameters) are local variables. Variables defined outside functions (before any function) are global variables.

Inside any function we can access the variables defined inside this function beside the global variables.

Example:

```
# the program will print numbers from 10 to 1

x = 10                                # x is a global variable.

def main
    for t = 1 to 10                    # t is a local variable
        mycounter()                  # call function
    end
end

def mycounter
    put x + nl                        # print the global variable value
    x--                               # decrement
end
```

Note: Using the main function before the for loop declare the t variable as a local variable, It's recommended to use the main functions instead of typing the instructions directly to set the scope of the new variables to local.

24.7 Return Value

The function can return a value using the Return command.

Syntax:

```
Return [Expression]
```

Tip: the Expression after the return command is optional and we can use the return command to end the function execution without returning any value.

Note: if the function doesn't return explicit value, it will return NULL (empty string = "").

Example:

```
if novalue() = NULL
    put "the function doesn't return a value" + nl
end

def novalue
```

24.8 Recursion

The Ring programming language support [Recursion](#) and the function can call itself using different parameters.

Example:

```
put fact(5)      # output = 120

def fact x if x = 0 return 1 else return x * fact(x-1) end
```


FUNCTIONS - THIRD STYLE

In this chapter we are going to learn about the next topics :-

- Define functions
- Call functions
- Declare parameters
- Send parameters
- Main Function
- Variables Scope
- Return Value
- Recursion

25.1 Define Functions

To define new function

Syntax:

```
func <function_name> [parameters] ['{'  
    Block of statements  
'}']
```

Example:

```
load "stdlib.ring"  
func hello {  
    print("Hello from function \n")  
}
```

25.2 Call Functions

To call function without parameters, we type the function name then ()

Tip: We can call the function before the function definition and the function code.

Example:

```
load "stdlib.ring"

hello()

func hello {
    print("Hello from function \n")
}
```

Example:

```
load "stdlib.ring"

first() second()

func first { print("message from the first function \n") }

func second { print("message from the second function \n") }
```

25.3 Declare parameters

To declare the function parameters, after the function name type the list of parameters as a group of identifiers separated by comma.

Example:

```
load "stdlib.ring"

func sum(x,y) {
    print(x+y)
}
```

25.4 Send Parameters

To send parameters to function, type the parameters inside () after the function name

Syntax:

```
funcname(parameters)
```

Example:

```
/* output
** 8
** 3000
*/

load "stdlib.ring"

sum(3,5) sum(1000,2000)

func sum(x,y) { print(x+y) }
```

25.5 Main Function

Using the Ring programming language, the Main Function is optional, when it's defined, it will be executed after the end of other statements.

if no other statements comes alone, the main function will be the first entry point

Example:

```
# this program will print the hello world message first then execute the main function

load "stdlib.ring"

print("Hello, World! \n")

func main {
    print("Message from the main function \n")
}
```

25.6 Variables Scope

The Ring programming language uses [lexical scoping](#) to determine the scope of a variable.

Variables defined inside functions (including function parameters) are local variables. Variables defined outside functions (before any function) are global variables.

Inside any function we can access the variables defined inside this function beside the global variables.

Example:

```
# the program will print numbers from 10 to 1

load "stdlib.ring"

x = 10                                # x is a global variable.

func main {
    for t = 1 to 10 {                 # t is a local variable
        mycounter()                  # call function
    }
}

func mycounter {
    print("#{x}\n")                   # print the global variable value
    x--                               # decrement
}
```

Note: Using the main function before the for loop declare the t variable as a local variable, It's recommended to use the main functions instead of typing the instructions directly to set the scope of the new variables to local.

25.7 Return Value

The function can return a value using the Return command.

Syntax:

Return [Expression]

Tip: the Expression after the return command is optional and we can use the return command to end the function execution without returning any value.

Note: if the function doesn't return explicit value, it will return NULL (empty string = "").

Example:

```
load "stdlib.ring"

if novalue() = NULL {
    print("the function doesn't return a value\n")
}

func novalue { }
```

25.8 Recursion

The Ring programming language support [Recursion](#) and the function can call itself using different parameters.

Example:

```
load "stdlib.ring"

print( fact(5) )           # output = 120

func fact(x) { if x = 0 { return 1 else return x * fact(x-1) } }
```

PROGRAM STRUCTURE

In this chapter we will learn about using many source code files in the same project.

26.1 Source Code File Sections

Each source code file may contains the next sections (in the same order).

Source Code File Sections
Load Files
Statements and Global Variables
Functions
Packages and Classes

The application maybe one or more of files.

26.2 Using Many Source Code Files

To include another source file in the project, just use the load command.

Syntax:

```
Load "filename.ring"
```

Note: The Load command is executed directly by the compiler in the parsing stage

Tip: if you don't know the file name until the runtime, or you need to use functions to get the file path, just use eval().

Example:

```
# File : Start.ring

Load "sub.ring"

sayhello("Mahmoud")
```

```
# File : sub.ring

func sayhello cName
    see "Hello " + cName + nl
```

LISTS

In this chapter we are going to learn how to deal with lists.

27.1 Create Lists

We can create new lists by defining the list items inside square brackets.

Example:

```
aList = [1,2,3,4,5]
```

Also we can create new lists using the `:` operator

Example:

```
aList = 1:5  
aList2 = "a":"z"
```

Example:

```
aList = 5:1  
aList2 = "z":"a"
```

Also we can create lists using the `list()` function

Syntax:

```
list = list(size)
```

Example

```
aList = list(10)           # aList contains 10 items
```

Note: the list index starts from 1

27.2 Add Items

To add new items to the list, we can use the `Add()` function.

Syntax:

```
Add(List, Item)
```

Example:

```
aList = ["one", "two"]
add(aList, "three")
see aList
```

Also we can do that using the + operator.

Syntax:

```
List + item
```

Example:

```
aList = 1:10      # create list contains numbers from 1 to 10
aList + 11        # add number 11 to the list
see aList         # print the list
```

27.3 Get List Size

We can get the list size using the len() function

Syntax:

```
Len(List)
```

Example:

```
aList = 1:20  see len(aList)  # print 20
```

27.4 Delete Item From List

To delete an item from the list, we can use the del() function

Syntax:

```
del(list, index)
```

Example:

```
aList = ["one", "two", "other", "three"]
Del(aList, 3)  # delete item number three
see aList      # print one two three
```

27.5 Get List Item

To get an item from the list, we uses the next syntax

```
List[Index]
```

Example:

```
aList = ["Cairo", "Riyadh"]
see "Egypt : " + aList[1] + nl +
    "KSA    : " + aList[2] + nl
```

27.6 Set List Item

To set the value of an item inside the list, we can use the next syntax

```
List[Index] = Expression
```

Example:

```
aList = list(3) # create list contains three items
aList[1] = "one" aList[2] = "two" aList[3] = "three"
see aList
```

27.7 Search

To find an item inside the list we can use the find() function

Syntax:

```
Find(List,ItemValue) ---> Item Index
Find(List,ItemValue,nColumn) ---> Search in nColumn, returns the Item Index
Find(List,ItemValue,nColumn,cAttribute) ---> Item Index
```

Example:

```
aList = ["one", "two", "three", "four", "five"]
see find(aList, "three")           # print 3
```

Example:

```
mylist = [{"one",1},
          ["two",2],
          ["three",3]]

see find(mylist, "two", 1) + nl      # print 2
see find(mylist, 2, 2) + nl        # print 2
```

Also we can use the binarysearch() function to search in sorted list.

Syntax:

```
BinarySearch(List,ItemValue) ---> Item Index
BinarySearch(List,ItemValue,nColumn) ---> Search in nColumn, returns the Item Index
```

Example:

```
aList = ["one", "two", "three", "four", "five"]
aList = sort(aList)
see binarysearch(aList, "three")
```

Output:

```
five
four
one
three
two
4
```


27.8 Sort

We can sort the list using the `sort()` function.

Syntax:

```
Sort(List) ---> Sorted List
Sort(List,nColumn) ---> Sorted List based on nColumn
Sort(List,nColumn,cAttribute) ---> Sorted List based on Object Attribute
```

Example:

```
aList = [10,12,3,5,31,15]
aList = sort(aList) see aList # print 3 5 10 12 15 31
```

We can sort list of strings

Example:

```
mylist = ["mahmoud","samir","ahmed","ibrahim","mohammed"]
see mylist # print list before sorting
mylist = sort(mylist) # sort list
see "list after sort"+nl
see mylist # print ahmed ibrahim mahmoud mohammed samir
```

We can sort a list based on a specific column.

Example:

```
aList = [ ["mahmoud",15000] ,
          ["ahmed", 14000 ] ,
          ["samir", 16000 ] ,
          ["mohammed", 12000 ] ,
          ["ibrahim",11000 ] ]

aList2 = sort(aList,1)
see aList2
```

Output:

```
ahmed
14000
ibrahim
11000
mahmoud
15000
mohammed
12000
samir
16000
```

27.9 Reverse

We can reverse a list using the `reverse()` function.

Syntax:

```
Reverse(List) ----> Reversed List
```

Example:

```
aList = [10,20,30,40,50]
aList = reverse(aList)
see aList           # print 50 40 30 20 10
```

27.10 Insert Items

To insert an item in the list we can use the insert() function.

Syntax:

```
Insert (List, Index, Item)
```

The inserted item will be AFTER the Index

Example:

```
aList = ["A","B","D","E"]
insert(aList,2,"C")      # Inserts AFTER Index 2, "C" into Position 3
see aList                # print A B C D E
```

27.11 Nested Lists

The list may contain other lists

Example:

```
aList = [ 1 , [10,20,30] , 5 , [100,1000,5000] ]
aList2 = [
    "one","two",
    [3,4],
    [20,30], ["three",
               "four",
               "five", [100,200,300]
            ]
]

see aList[2]           # print 10 20 30
see aList[4][3] + nl    # print 5000
see aList2[5][2] + nl   # print four
see aList2[5][4][3]     # print 300
```

27.12 Copy Lists

We can copy lists (including nested lists) using the Assignment operator.

Example:

```

aList = [
  "one", "two",
  [3,4],
  [20,30], ["three",
             "four",
             "five", [100,200,300]
            ]
]

aList2 = aList          # Copy aList to aList2
aList2[5] = "other"     # modify item number five
see aList2[5] + nl      # print other
see aList[5]            # print three four five 100 200 300

```

27.13 First-class lists

Lists are [first-class citizens](#) where we can store lists in variables, pass lists to functions, and return lists from functions.

Example:

```

aList = duplicate( [1,2,3,4,5] )
see aList[10] + nl      # print 5

see mylist()            # print 10 20 30 40 50

func duplicate list
  nMax = len(list)
  for x = 1 to nMax
    list + list[x]
  next
  return list

func mylist return [10,20,30,40,50]

```

27.14 Using Lists during definition

We can use the list items while we are defining the list for the first time.

Example:

```

aList = [ [1,2,3,4,5] , aList[1] , aList[1] ]
see aList              # print 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5

```

27.15 Passing Lists to Functions

Lists are passed to functions by reference, This means that the called function will work on the same list and can modify it.

Example:

```

func main
  aList = [1,2,3,4,5]    # create list, local in function main

```

```

myfunc(aList)           # call function, pass list by reference
see aList               # print 1 2 3 4 5 6 7 8 9 10

func myfunc list
    list + [6,7,8,9,10]

```

27.16 Access List Items by String Index

Instead of using numbers to determine the item index when we get item value or set item value, We can access items using string index if the item is a list contains two items and the first item is a string.

Example:

```

aList = [ ["one",1] , ["two",2] , ["three",3] ]
see aList["one"] + nl +
    aList["two"] + nl +
    aList["three"]      # print 1 2 3

```

This type of lists can be defined in a better syntax using the : and = operators.

Example:

```

aList = [ :one = 1 , :two = 2 , :three = 3 ]
see aList["one"] + nl +
    aList["two"] + nl +
    aList["three"] + nl # print 1 2 3
see aList[1]           # print one 1

```

Tip: using : before identifier (one word) means literal

Note: using = inside list definition create a list of two items where the first item is the left side and the second item is the right side.

We can add new items to the list using the string index

Example:

```

aList = []
aList["Egypt"] = "Cairo"
aList["KSA"] = "Riyadh"
see aList["Egypt"] + nl +      # print Cairo
    aList["KSA"] + nl         # print Riyadh

```

27.17 Passing Parameters Using List

This type of lists is very good for passing parameters to functions Where the order of parameters will not be important (we can change the order).

Also some parameters maybe optional.

Example:

```
myconnect ( [ :server = "myserver.com" , :port = 80 ,
             :username = "mahmoud" , :password = "password" ] )

func myconnect mypara

    # print connection details
    see "User Name : " + mypara[:username] + nl +
        "Password : " + mypara[:password] + nl +
        "Server : " + mypara[:server] + nl +
        "Port : " + mypara[:port]
```

27.18 Swap Items

We can swap the list items using the `Swap()` function.

Example:

```
aList = [:one, :two, :four, :three]
see aList
see copy(" ", 50) + nl
swap(aList, 3, 4)
see aList
```

Output

```
one
two
four
three
*****
one
two
three
four
```

STRINGS

In this chapter we are going to learn about strings creation and manipulation.

28.1 String Literals

Syntax:

```
cStr = "This is a string"
cStr2 = 'Another string'
cStr3 = :JustAnotherString
cStr4 = `Yet "another" 'string' ! `
```

28.2 Get String Length

We can get the string length (letters count inside a string) using the `len()` function

Syntax:

```
len(string) ---> string length
```

Example:

```
cStr = "How are you?"
see cStr + nl
see "String size : " + len(cStr) + nl
```

28.3 Convert Letters Case

Syntax:

```
lower(string) ---> convert string letters to lower case
upper(string) ---> convert string letters to UPPER case
```

Example:

```
cStr = "Welcome To The Ring Programming Language"
see cStr + nl + upper(cStr) + nl + lower(cStr)
```

28.4 Access String Letters

We can access a letter inside a string by the letter index

Syntax:

```
string[index] ---> get string letter
string[index] = letter # set string letter
```

Example:

```
# print user name letter by letter (each letter in new line)

See "Hello, Enter your name : " give cName
for x = 1 to len(cName)
    see nl + cName[x]
next
```

We can use for in to get string letters.

Example:

```
# print user name letter by letter (each letter in new line)

See "Hello, Enter your name : " give cName
for x in cName
    see nl + x
next
```

We can modify the string letters

Example:

```
# convert the first letter to UPPER case

See "Enter your name : " give cName
cName[1] = upper(cName[1])
see "Hello " + cName
```

28.5 Left() Function

We can get a specified number of characters from a string using the Left() function.

The starting position is 1.

Syntax:

```
Left(string, count)
```

Example:

```
see left("Hello World!",5) # print Hello
```

28.6 Right() Function

We can get a specified number of characters from a string using the Right() function.

The starting position is the last character on the right.

Syntax:

```
Right(string, count)
```

Example:

```
see Right("Hello World!", 6) # print World!
```

28.7 Trim() Function

We can remove all leading and trailing spaces from a string using the Trim() function.

Syntax:

```
trim(string)
```

Example:

```
cMsg = "    Welcome    "
see trim(cMsg) # print Welcome
```

28.8 Copy() Function

We can duplicate a string more than one time using the copy() function.

Syntax:

```
copy(string, nCount) ---> string replicated nCount times
```

Example

```
see copy("***hello***", 3) # print ***hello*****hello*****hello***
```

28.9 Lines() Function

We can count the number of lines inside a string using the Lines() function.

Syntax:

```
lines(string) ---> Number of lines inside the string
```

Example:

```
cStr = "Hello
How are you?
are you fine?"
see lines(cStr) # print 3
```


28.10 Substr() Function

We can work on sub strings inside a string using the substr() function. Using Substr() we can

- Find substring
- Get substring from position to end
- Get Number of characters from position
- Transform Substring To Another Substring

28.11 Find substring

Syntax:

```
substr(string, substring) ---> the starting position of substring in string
```

Example:

```
cStr = "Welcome to the Ring programming language"
see substr(cStr, "Ring")           # print 16
```

28.12 Get substring from position to end

Syntax:

```
substr(string, position) ---> Get substring starting from position to end
```

Example:

```
cStr = "Welcome to the Ring programming language"
nPos = substr(cStr, "Ring")         # nPos = 16
see substr(cStr, nPos)              # print Ring programming language
```

28.13 Get Number of Characters From Position

Syntax:

```
substr(string, position, count) ---> Get characters starting from position
```

Example:

```
cStr = "Welcome to the Ring programming language"
nPos = substr(cStr, "Ring")         # nPos = 16
see substr(cStr, nPos, 4)           # print Ring
```

28.14 Transform Substring To Another Substring

Syntax:

```
substr(string, substring, newsubstring) ---> Transformed string (Match case)
substr(string, substring, newsubstring, 1) ---> Transformed string (Ignore case)
```

Example:

```
cStr = "Welcome to the New programming language"
see substr(cStr, "New", "Ring") + nl # print Welcome to the Ring programming language
see substr(cStr, "new", "Ring", 1) + nl # print Welcome to the Ring programming language
```

28.15 strcmp() Function

We can compare between two strings using the strcmp() function.

Syntax:

```
strcmp(cString1, cString2) ---> value = 0 if cString1 = cString2
                                value < 0 if cString1 < cString2
                                value > 0 if cString1 > cString2
```

Example:

```
see strcmp("hello", "hello") + nl +
    strcmp("abc", "bcd") + nl +
    strcmp("bcd", "abc") + nl
```

Output:

```
0
-1
1
```

28.16 str2list() and list2str() Functions

We can convert string lines to list items using the str2list() function. Also we can convert the list to a string using list2str() function.

Syntax:

```
str2list(string) ---> list contains the string lines
list2str(list) ---> string contains the list items
```

Example:

```
/* output:
** Items : 4
** Item : Hello
** Item : How are you ?
** Item : are you fine ?
** Item : ok
** list2Str result = Hello
** How are you ?
** are you fine ?
** ok
** Done
*/
```

```
mystr = "Hello
How are you ?
are you fine ?
ok"

mylist = str2list(mystr)
see "Items : " + len(mylist) + nl

for x in mylist
    see "Item : " + x + nl
next

newstr = list2str(mylist)
see "list2Str result = " + newstr

if mystr == newstr
    see nl + "Done"
else
    see nl + "Error!"
ok
```

DATE AND TIME

In this chapter we are going to learn about the date and time functions.

29.1 Clock() Function

Syntax:

```
Clock() ---> The number of clock ticks from program start
```

Example:

```
See "Calculate performance" + nl  
t1 = clock()  
for x = 1 to 1000000 next  
see clock() - t1
```

29.2 ClocksPerSecond() Function

Return how many clocks in one second

Syntax:

```
clocksperssecond() ---> Number of clocks in one second
```

Example:

```
# Wait 1 second  
t = clock()  
while clock() - t <= clocksperssecond() end
```

29.3 Time() Function

We can get the system time using the Time() function.

Example:

```
See "Time : " + time()
```

29.4 Date() Function

We can get the date using the Date() function.

Syntax:

```
Date() ---> String represent the date "dd/mm/yyyy"
```

Example:

```
See "Date : " + date() # Date : 24/05/2015
```

29.5 TimeList() Function

We can print the date and the time information using the TimeList() function.

Syntax:

```
TimeList() ---> List contains the time and date information.
```

The next table presents the list items

index	value
1	abbreviated weekday name
2	full weekday name
3	abbreviated month name
4	full month name
5	Date & Time
6	Day of the month
7	Hour (24)
8	Hour (12)
9	Day of the year
10	Month of the year
11	Minutes after hour
12	AM or PM
13	Seconds after the hour
14	Week of the year (sun-sat)
15	day of the week
16	date
17	time
18	year of the century
19	year
20	time zone
21	percent sign

Example:

```
/* Output:
** Sun           abbreviated weekday name
** Sunday        full weekday name
** May           abbreviated month name
** May           full month name
** 05/24/15 09:58:38 Date & Time
** 24            Day of the month
** 09            Hour (24)
```

```

** 09          Hour (12)
** 144         Day of the year
** 05          Month of the year
** 58          Minutes after hour
** AM          AM or PM
** 38          Seconds after the hour
** 21          Week of the year (sun-sat)
** 0           day of the week
** 05/24/15    date
** 09:58:38    time
** 15          year of the century
** 2015        year
** Arab Standard Time  time zone
** %           percent sign
*/

See TimeList()

```

Example:

```
See "Day Name : " + TimeList()[2]      # Sunday
```

Example:

```
See "Month Name : " + TimeList()[4]    # May
```

29.6 AddDays() Function

Syntax:

```
AddDays(cDate,nDays) ---> Date from cDate and after nDays
```

Example:

```

cDate = date()
see cDate + nl      # 24/05/2015
cDate = adddays(cDate,10)
see cDate + nl      # 03/06/2015

```

29.7 DiffDays() Function

Syntax:

```
DiffDays(cDate1,cDate2) ---> number of days (Date1 - Date2)
```

Example:

```

cDate1 = date()
see cDate1 + nl      # 24/05/2015
cDate2 = adddays(cDate1,10)
see cDate2 + nl      # 03/06/2015
see "DiffDays = " + diffdays(cDate1,cDate2) + nl      # -10
see "DiffDays = " + diffdays(cDate2,cDate1) + nl      # 10

```

29.8 EpochTime() Function

Syntax:

```
EpochTime( cDate, cTime ) ---> Epoch Seconds
```

Example:

```
###-----
# EpochTime()
# Example --- EpochSec = EpochTime( Date(), Time() )
# Call Format: EpochSec = EpochTime( "15/07/2016", "10:15:30" )
# EpochSec = 1468577730
#-----

Func EpochTime(Date, Time)

    arrayDate = split(Date, "/")
    arrayTime = split(Time, ":")

    Year = arrayDate[3] ; Month = arrayDate[2] ; Day = arrayDate[1]
    Hour = arrayTime[1] ; Minute = arrayTime[2] ; Second = arrayTime[3]

    cDate1 = Day + "/" + Month + "/" + Year
    cDate2 = "01/01/" + Year
    DayOfYear = DiffDays( cDate1, cDate2)

    ### Formula
    tm_sec = Second * 1
    tm_min = Minute * 60
    tm_hour = Hour * 3600
    tm_yday = DayOfYear * 86400
    tm_year = Year - 1900

    tm_year1 = ( tm_year - 70 ) * 31536000
    tm_year2 = ( floor(( tm_year - 69 ) / 4 ) ) * 86400
    tm_year3 = ( floor(( tm_year - 1 ) / 100 ) ) * 86400
    tm_year4 = ( floor(( tm_year + 299 ) / 400 ) ) * 86400

    ### Result
    EpochSec = tm_sec + tm_min + tm_hour + tm_yday + tm_year1 + tm_year2 - tm_year3 + tm_year4

return EpochSec
```

CHECK DATA TYPE AND CONVERSION

In this chapter we are going to learn about the functions that can be used for

- Checking Data Type
- Checking Character
- Conversion

30.1 Check Data Type

The next functions can be used to check the data type

- `isstring()`
- `isnumber()`
- `islist()`
- `type()`
- `isnull()`

30.2 IsString() Function

Using the `IsString()` function we can know if the value is a string or not

Syntax:

```
IsString(value) ----> 1 if the value is a string or 0 if not
```

Example:

```
see isstring(5) + nl +           # print 0
    isstring("hello") + nl      # print 1
```

30.3 IsNumber() Function

Using the `IsNumber()` function we can know if the value is a number or not

Syntax:


```
IsNumber(value) ---> 1 if the value is a number or 0 if not
```

Example:

```
see isnumber(5) + nl +      # print 1
    isnumber("hello") + nl  # print 0
```

30.4 IsList() Function

Using the IsList() function we can know if the value is a list or not

Syntax:

```
IsList(value) ---> 1 if the value is a list or 0 if not
```

Example:

```
see islist(5) + nl +      # print 0
    islist("hello") + nl + # print 0
    islist([1,3,5])       # print 1
```

30.5 Type() Function

We can know the type of a value using the Type() Function.

Syntax:

```
Type(value) ---> The Type as String
```

Example:

```
see Type(5) + nl +      # print NUMBER
    Type("hello") + nl + # print STRING
    Type([1,3,5])       # print LIST
```

30.6 IsNULL() Function

We can check the value to know if it's null or not using the IsNULL() function

Syntax:

```
IsNULL(value) ---> 1 if the value is NULL or 0 if not
```

Example:

```
see isnull(5) + nl +      # print 0
    isnull("hello") + nl + # print 0
    isnull([1,3,5]) + nl + # print 0
    isnull("") + nl +      # print 1
    isnull("NULL") + nl +  # print 1
```

30.7 Check Character

The next functions can be used to check character

- `isalnum()`
- `isalpha()`
- `isctrl()`
- `isdigit()`
- `isgraph()`
- `islower()`
- `isprint()`
- `ispunct()`
- `isspace()`
- `isupper()`
- `isxdigit()`

30.8 IsAlNum() Function

We can test a character or a string using the `IsAlNum()` Function

Syntax:

```
IsAlNum(value) ---> 1 if the value is digit/letter or 0 if not
```

Example:

```
see isalnum("Hello") + nl +      # print 1
    isalnum("123456") + nl +      # print 1
    isalnum("ABCabc123") + nl +    # print 1
    isalnum("How are you")         # print 0 because of spaces
```

30.9 IsAlpha() Function

We can test a character or a string using the `IsAlpha()` Function

Syntax:

```
IsAlpha(value) ---> 1 if the value is a letter or 0 if not
```

Example:

```
see isalpha("Hello") + nl +      # print 1
    isalpha("123456") + nl +      # print 0
    isalpha("ABCabc123") + nl +    # print 0
    isalpha("How are you")         # print 0
```

30.10 IsCntrl() Function

We can test a character or a string using the IsCntrl() Function

Syntax:

```
IsCntrl(value) ---> 1 if the value is a control character (no printing position) or 0 if not
```

Example:

```
See iscntrl("hello") + nl +      # print 0
    iscntrl(nl)           # print 1
```

30.11 IsDigit() Function

We can test a character or a string using the IsDigit() Function

Syntax:

```
IsDigit(value) ---> 1 if the value is a digit or 0 if not
```

Example:

```
see isdigit("0123456789") + nl +      # print 1
    isdigit("0123a")           # print 0
```

30.12 IsGraph() Function

We can test a character or a string using the IsGraph() Function

Syntax:

```
IsGraph(value) ---> 1 if the value can be printed (Except space) or 0 if not
```

Example:

```
see isgraph("abcdef") + nl +      # print 1
    isgraph("abc def")           # print 0
```

30.13 IsLower() Function

We can test a character or a string using the IsLower() Function

Syntax:

```
IsLower(value) ---> 1 if the value is lowercase letter or 0 if not
```

Example:

```
see islower("abcDEF") + nl +      # print 0
    islower("ghi")           # print 1
```

30.14 IsPrint() Function

We can test a character or a string using the IsPrint() Function

Syntax:

```
IsPrint(value) ---> 1 if the value occupies a printing position or 0 if not
```

Example:

```
see isprint("Hello") + nl +      # print 1
    isprint("Nice to see you") + nl +  # print 1
    isprint(nl)                  # print 0
```

30.15 IsPunct() Function

We can test a character or a string using the IsPunct() Function

Syntax:

```
IsPunct(value) ---> 1 if the value is a punctuation character or 0 if not
```

Example:

```
see ispunct("hello") + nl +      # print 0
    ispunct(",")          # print 1
```

30.16 IsSpace() Function

We can test a character or a string using the IsSpace() Function

Syntax:

```
IsSpace(value) ---> 1 if the value is a white-space or 0 if not
```

Example:

```
see isspace(" ") + nl +      # print 1
    isspace("test")          # print 0
```

30.17 IsUpper() Function

We can test a character or a string using the IsUpper() Function

Syntax:

```
IsUpper(value) ---> 1 if the value is an uppercase alphabetic letter or 0 if not
```

Example:

```
see isupper("welcome") + nl +      # print 0
    isupper("WELCOME")          # print 1
```

30.18 IsXdigit() Function

We can test a character or a string using the IsXdigit() Function

Syntax:

```
IsXdigit(value) ---> 1 if the value is a hexadecimal digit character or 0 if not
```

Example:

```
see isxdigit("0123456789abcdef") + nl + # print 1
      isxdigit("123z")                  # print 0
```

30.19 Conversion

The next functions can be used for conversion

- number()
- string()
- ascii()
- char()
- hex()
- dec()
- str2hex()
- hex2str()

30.20 Number() Function

We can convert strings to numbers using the Number() function or the + operator.

Syntax:

```
Number(string) ---> Number
0 + string ---> Number
```

Example:

```
see number("5") + 5 + nl      # print 10
see 0 + "10" + 2             # print 12
```

30.21 String() Function

We can convert numbers to strings using the String() function or the + operator.

Syntax:

```
String(number) ---> String
" " + number ---> String
```

Example:

```
see string(5) + 5 + nl      # print 55
see " " + 10 + 2           # print 102
```

30.22 Ascii() Function

We can get the ASCII code for a letter using the Ascii() function

Syntax:

```
Ascii(character) ---> ASCII Code
```

Example:

```
See ascii("m") + nl +      # print 109
    ascii("M")              # print 77
```

30.23 Char() Function

We can convert the ASCII code to character using the Char() function.

Syntax:

```
Char(ASCII Code) ---> character
```

Example:

```
See char(109) + nl +      # print m
    char(77)              # print M
```

30.24 Hex() Function

We can convert decimal to hexadecimal using the Hex() function.

Syntax:

```
Hex(decimal) ---> hexadecimal
```

Example:

```
See hex(10) + nl +        # print a
    hex(200)              # print c8
```

30.25 Dec() Function

We can convert hexadecimal to decimal using the Dec() function

Syntax:

```
Dec(hexadecimal) ---> decimal
```

Example:

```
See dec("a") + nl +      # print 10
    dec("c8")             # print 200
```

30.26 Str2hex() Function

We can convert string characters to hexadecimal characters using the Str2hex() function.

Syntax:

```
Str2hex(string) ---> hexadecimal string
```

Example:

```
See str2hex("hello")      # print 68656c6c6f
```

30.27 Hex2str() Function

We can convert hexadecimal characters to string using the Hex2str() function

Syntax:

```
Hex2Str(Hexadecimal string) ---> string
```

Example:

```
See hex2str("68656c6c6f") # print hello
```

MATHEMATICAL FUNCTIONS

In this chapter we are going to learn about the mathematical functions

31.1 List of functions

The Ring programming language comes with the next mathematical functions

Function	Description
sin(x)	Returns the sine of an angle of x radians
cos(x)	Returns the cosine of an angle of x radians
tan(x)	Returns the tangent of an angle of x radians
asin(x)	Returns the principal value of the arc sine of x, expressed in radians
acos(x)	Returns the principal value of the arc cosine of x, expressed in radians
atan(x)	Returns the principal value of the arc tangent of x, expressed in radians
atan2(y,x)	Returns the principal arc tangent of y/x, in the interval [-pi,+pi] radians
sinh(x)	Returns the hyperbolic sine of x radians
cosh(x)	Returns the hyperbolic cosine of x radians
tanh(x)	Returns the hyperbolic tangent of x radians
exp(x)	Returns the value of e raised to the xth power
log(x)	Returns the natural logarithm of x
log10(x)	Returns the common logarithm (base-10 logarithm) of x
ceil(x)	Returns the smallest integer value greater than or equal to x
floor(x)	Returns the largest integer value less than or equal to x
fabs(x)	Returns the absolute value of x.
pow(x,y)	Returns x raised to the power of y
sqrt(x)	Returns the square root of x
random(x)	Returns a random number in the range [0,x]
unsigned(n,n,c)	Perform operation using unsigned numbers
decimals(n)	Determine the decimals digits after the point in float/double numbers

31.2 Example

```
See "Mathematical Functions" + nl
See "Sin(0) = " + sin(0) + nl
See "Sin(90) radians = " + sin(90) + nl
See "Sin(90) degree = " + sin(90*3.14/180) + nl

See "Cos(0) = " + cos(0) + nl
See "Cos(90) radians = " + cos(90) + nl
```



```

See "Cos(90) degree = " + cos(90*3.14/180) + nl

See "Tan(0) = " + tan(0) + nl
See "Tan(90) radians = " + tan(90) + nl
See "Tan(90) degree = " + tan(90*3.14/180) + nl

See "asin(0) = " + asin(0) + nl
See "acos(0) = " + acos(0) + nl
See "atan(0) = " + atan(0) + nl
See "atan2(1,1) = " + atan2(1,1) + nl

See "sinh(0) = " + sinh(0) + nl
See "sinh(1) = " + sinh(1) + nl
See "cosh(0) = " + cosh(0) + nl
See "cosh(1) = " + cosh(1) + nl
See "tanh(0) = " + tanh(0) + nl
See "tanh(1) = " + tanh(1) + nl

See "exp(0) = " + exp(0) + nl
See "exp(1) = " + exp(1) + nl
See "log(1) = " + log(1) + nl
See "log(2) = " + log(2) + nl
See "log10(1) = " + log10(1) + nl
See "log10(2) = " + log10(2) + nl
See "log10(10) = " + log10(10) + nl

See "Ceil(1.12) = " + Ceil(1.12) + nl
See "Ceil(1.72) = " + Ceil(1.72) + nl

See "Floor(1.12) = " + floor(1.12) + nl
See "Floor(1.72) = " + floor(1.72) + nl

See "fabs(1.12) = " + fabs(1.12) + nl
See "fabs(1.72) = " + fabs(1.72) + nl

See "pow(2,3) = " + pow(2,3) + nl

see "sqrt(16) = " + sqrt(16) + nl

```

Program Output:

```

Mathematical Functions
Sin(0) = 0
Sin(90) radians = 0.89
Sin(90) degree = 1.00
Cos(0) = 1
Cos(90) radians = -0.45
Cos(90) degree = 0.00
Tan(0) = 0
Tan(90) radians = -2.00
Tan(90) degree = 1255.77
asin(0) = 0
acos(0) = 1.57
atan(0) = 0
atan2(1,1) = 0.79
sinh(0) = 0
sinh(1) = 1.18
cosh(0) = 1

```

```

cosh(1) = 1.54
tanh(0) = 0
tanh(1) = 0.76
exp(0) = 1
exp(1) = 2.72
log(1) = 0
log(2) = 0.69
log10(1) = 0
log10(2) = 0.30
log10(10) = 1
Ceil(1.12) = 2
Ceil(1.72) = 2
Floor(1.12) = 1
Floor(1.72) = 1
fabs(1.12) = 1.12
fabs(1.72) = 1.72
pow(2,3) = 8
sqrt(16) = 4

```

31.3 Random() Function

The Random() function generate a random number and we can set the maximum value (optional).

Syntax:

```
Random(x) ---> Random number in the range [0,x]
```

Example:

```

for x = 1 to 20
    see "Random number : " + random() + nl +
        "Random number Max (100) : " + random(100) + nl
next

```

Program Output:

```

Random number : 31881
Random number Max (100) : 80
Random number : 5573
Random number Max (100) : 63
Random number : 2231
Random number Max (100) : 43
Random number : 12946
Random number Max (100) : 39
Random number : 22934
Random number Max (100) : 48
Random number : 4690
Random number Max (100) : 52
Random number : 13196
Random number Max (100) : 65
Random number : 30390
Random number Max (100) : 87
Random number : 4327
Random number Max (100) : 77
Random number : 12456
Random number Max (100) : 17
Random number : 28438

```

```

Random number Max (100) : 13
Random number : 30503
Random number Max (100) : 6
Random number : 31769
Random number Max (100) : 94
Random number : 8274
Random number Max (100) : 65
Random number : 14390
Random number Max (100) : 90
Random number : 28866
Random number Max (100) : 12
Random number : 24558
Random number Max (100) : 70
Random number : 29981
Random number Max (100) : 77
Random number : 12847
Random number Max (100) : 63
Random number : 6632
Random number Max (100) : 60

```

31.4 Unsigned() Function

We can use unsigned numbers using the Unsigned() function.

Syntax:

```
Unsigned(nNum1,nNum2,cOperator) --> result of cOperator operation on nNum1,nNum2
```

Example:

```

see oat_hash("hello") + nl

# Jenkins hash function - https://en.wikipedia.org/wiki/Jenkins_hash_function
func oat_hash cKey
    h = 0
    for x in cKey
        h = unsigned(h,ascii(x),"+")
        h = unsigned(h,unsigned(h,10,"<<"),"+")
        r = unsigned(h,6,">>")
        h = unsigned(h, r,"^")

    next
    h = unsigned(h,unsigned(h,3,"<<"),"+")
    h = unsigned(h,unsigned(h,11,">>"),"^")
    h = unsigned(h,unsigned(h,15,"<<"),"+")
    return h

```

Output:

```
3372029979.00
```

31.5 Decimals() Functions

We can determine the decimals numbers count after the point in float/double numbers using the decimals() function.

Syntax:

```
Decimals (nDecimalsCount)
```

Example:

```
x = 1.1234567890123
for d = 0 to 14
    decimals(d)
    see x + nl
next
```

Output:

```
1
1.1
1.12
1.123
1.1235
1.12346
1.123457
1.1234568
1.12345679
1.123456789
1.1234567890
1.12345678901
1.123456789012
1.1234567890123
1.12345678901230
```

31.6 Using _ in numbers

We can use _ between numbers digits.

Example:

```
x = 1_000_000
see type(x) + nl
see x + nl
```

Output:

```
NUMBER
100000001
```

31.7 Using f after numbers

We can use the ‘f’ letter after numbers.

Example:

```
x = 19.99f
see type(x) + nl
```

Output:

```
NUMBER
```

FILES

In this chapter we are going to learn about files functions.

- Read()
- Write()
- Dir()
- Rename()
- Remove()
- fopen()
- fclose()
- fflush()
- freopen()
- tempfile()
- tmpname()
- fseek()
- ftell()
- rewind()
- fgetpos()
- fsetpos()
- clearerr()
- feof()
- ferror()
- perror()
- fgetc()
- fgets()
- fputc()
- fputs()
- ungetc()
- fread()

- `fwrite()`
- `fexists()`
- Numbers and Bytes

32.1 Read() Function

We can read the file content using the `Read()` function

Syntax:

```
Read(cFileName) ---> String contains the file content
```

Example:

```
see read("myfile.txt")
```

The read function can read binary files too

Example:

```
see read("myapp.exe")
```

32.2 Write() Function

We can write string to file using the `Write()` function

The write function can write binary data to binary files.

Syntax:

```
Write(cFileName,cString)      # write string cString to file cFileName
```

Example:

```
# copy file
cFile = read("ring.exe")
write("ring2.exe",cFile)
```

32.3 Dir() Function

We can get the folder contents (files & sub folders) using the `Dir()` function.

Syntax:

```
Dir(cFolderPath) ---> List contains files & sub folders.
```

This function returns a list and each list item is a list of two items

- File/sub folder name
- Type (0 = File , 1 = Folder/Directory)

Example:

```

see "Testing DIR() " + nl
mylist = dir("C:\myfolder")
for x in mylist
    if x[2]
        see "Directory : " + x[1] + nl
    else
        see "File : " + x[1] + nl
    ok
next
see "Files count : " + len(mylist)

```

32.4 Rename() Function

We can rename files using the Rename() function

Syntax:

```
Rename(cOldFileName, cNewFileName)
```

Example:

```
rename("file.txt", "help.txt")
```

32.5 Remove() Function

We can delete a file using the Remove() function

Syntax:

```
Remove(cFileName)
```

Example:

```
remove("test.txt")
```

32.6 Fopen() Function

We can open a file using the Fopen() function

Syntax:

```
Fopen(cFileName, cMode) ---> File Handle
```

Mode	Description
"r"	Reading (The file must exist)
"w"	Writing (create empty file / overwrite)
"a"	Appends (create file if it doesn't exist)
"r+"	update (reading/writing)
"w+"	Create empty file (reading/writing)
"a+"	reading & appending

32.7 Fclose() Function

When we open a file using fopen() function, we can close it using the Fclose() function

Syntax:

```
Fclose(file handle)
```

32.8 Fflush() Function

We can flushes the output buffer of a stream using the Fflush() function

Syntax:

```
Fflush(file handle)
```

32.9 Freopen() Function

We can open another file using the same file handle and at the same time close the old file

Syntax:

```
Freopen(cFileName,cMode,file handle) ---> file handle
```

Example:

```
freopen("myprogoutput.txt", "w+", stdout)
see "welcome" + nl
for x = 1 to 10
    see x + nl
next

/*
** Read : https://en.wikipedia.org/wiki/Device\_file#Device\_files
** The next code is not portable, we can use iswindows() before
** using it and we can write special code for each operating system.
*/

freopen("CON", "w", stdout)      # For Microsoft Windows
see "Done" + nl                # print to stdout again
```

Output:

```
# Output to stdout
Done

# Output to file : myprogoutput.txt
welcome
1
2
3
4
5
6
7
```



```
8
9
10
```

32.10 Tempfile() Function

The function Tempfile() creates a temp. file (binary).

The file will be deleted automatically when the stream is closed

Syntax:

```
TempFile() ---> file handle
```

32.11 Tempname() Function

We can generate temp. file name using the Tempname() function

The generated name will be different from the name of any existing file

Syntax:

```
Tempname() ---> generated file name as string
```

32.12 Fseek() Function

We can set the file position of the stream using the Fseek() function

Syntax:

```
Fseek(file handle, nOffset, nWhence) ---> zero if successful
```

The next table presents the nWhence values

Value	Description
0	Beginning of file
1	Current position
2	End of file

32.13 Ftell() Function

We can know the current file position of a stream using the Ftell() function

Syntax:

```
Ftell(file handle) ---> file position as number
```

32.14 Rewind() Function

We can set the file position to the beginning of the file using the Rewind() function

Syntax:

```
Rewind(file handle)
```

32.15 Fgetpos() Function

We can get handle to the current file position using the Fgetpos() function

Syntax:

```
Fgetpos(file handle) ---> position handle
```

32.16 Fsetpos() Function

We can set the current file position using the Fsetpos() function

Syntax:

```
Fsetpos(file handle, position handle)
```

32.17 Clearerr() Function

We can clear the EOF error and the error indicators of a stream using the clearerr() function

Syntax:

```
Clearerr(file handle)
```

32.18 Feof() Function

We can test the end-of-file indicator using the Feof() function

Syntax:

```
Feof(file handle) ---> returns 1 if EOF and 0 if not
```

32.19 Ferror() Function

We can test the error indicator of a given stream using the Ferror() function

Syntax:

```
Ferror(file handle) ---> returns 1 if error and 0 if not
```

32.20 Perror() Function

We can print error message to the stderr using the Perror() function

Syntax:

```
Perror(cErrorMessage)
```

32.21 Fgetc() Function

We can get the next character from the stream using the Fgetc() function

Syntax:

```
Fgetc(file handle) ---> returns character or EOF
```

32.22 Fgets() Function

We can read new line from the stream using the Fgets() function

Syntax:

```
Fgets(file handle, nSize) ---> string
```

The function stop when nSize characters are read, new line character is read or EOF.

32.23 Fputc() Function

We can write a character to the stream using the Fputc() function

Syntax:

```
Fputc(file handle, cChar)
```

32.24 Fputs() Function

We can write a string to the stream using the Fputs() function

Syntax:

```
Fputs(file handle, cString)
```

32.25 Ungetc() Function

We can push a character to the stream using the Ungetc() function

The character will be available for the next read

Syntax:

```
Ungetc(file handle, character)
```

32.26 Fread() Function

We can read data from a stream using the Fread() function

Syntax:

```
Fread(file handle, nSize)
```

32.27 Fwrite() Function

We can write data to a stream using the Fwrite() function

Syntax:

```
Fwrite(file handle, cString)
```

32.28 Fexists() Function

We can check if a file exists using the Fexists() function

Syntax:

```
Fexists(cFileName) ---> returns 1 if the file exists
```

Example:

```
see fexists("b:\mahmoud\apps\ring\ring.exe") + nl +
    fexists("b:\mahmoud\apps\ring\ring2.exe") + nl
```

Output:

```
1
0
```

32.29 Example

The next program test some of the file functions

```
See "testing file functions" + nl

See "open file" + nl
fp = fopen(exefolder() + "../tests/scripts/s65.ring", "r")

See "reopen" + nl
fp = freopen(exefolder() + "../tests/scripts/s78.ring", "r", fp)
See "close file" + nl
fclose(fp)

see "temp file" + nl
```

```

fp = tempfile()
fclose(fp)

see "temp name" + nl
see tempname() + nl

remove(exefolder() + "../tests/scripts/mytest2.txt")
write(exefolder() + "../tests/scripts/testsl.txt","hello")
rename(exefolder() + "../tests/scripts/test1.txt",exefolder() +
        "../tests/scripts/mytests2.txt")

see "print file" + nl
fp = fopen(exefolder() + "../samples/fromdoc/filefuncs.ring","r")
r = fgetc(fp)
while isstring(r)
    see r
    r = fgetc(fp)
end
fclose(fp)

see nl+"print line from the file" + nl
fp = fopen(exefolder() + "../samples/fromdoc/filefuncs.ring","r")
r = fgets(fp,33)
see r + nl
fclose(fp)
fp = fopen(exefolder() + "../tests/scripts/test78.txt","w+")
fseek(fp,0,2) # goto end of file
fputc(fp,"t")
fputc(fp,"e")
fputc(fp,"s")
fputc(fp,"t")
fputs(fp,"tests2")
fclose(fp)

see "print file" + nl
see read(exefolder() + "../tests/scripts/test78.txt")

fp = fopen(exefolder() + "../tests/scripts/test78.txt","r")
see "testing ungetc() " + nl
for x = 1 to 3
    r = fgetc(fp)
    see r + nl
    ungetc(fp,r)
next
fclose(fp)

see "testing fread() " + nl
fp = fopen(exefilename(),"rb")
r = fread(fp,100)
see r + nl
fclose(fp)

see "testing fwrite() " + nl
fp = fopen(exefolder() + "../tests/scripts/test1.txt","wb")
fwrite(fp,r)
fclose(fp)

```

The next example print part of the content of a binary file

```

see "Testing: fread()" + " FileName: " + exefilename() + nl + nl
fp = fopen(exefilename(), "rb")
r = fread(fp, 800)
for n = 1 to len(r)
    if isprint(substr(r, n, 1))
        see substr(r, n, 1)
    else
        see "."
    ok
    ### 80 char per line
    if n % 80 = 0
        see nl
    ok
next
fclose(fp)

```

32.30 Numbers and Bytes

The next functions to convert between Numbers and Bytes.

- Int2Bytes()
- Float2Bytes()
- Double2Bytes()
- Bytes2Int()
- Bytes2Float()
- Bytes2Double()

Example:

```

see "Test Int2Bytes() and Bytes2Int() - Value : 77" + nl
r = Int2Bytes(77)
see "Int Size : " + len(r) + nl
see r + nl
see Bytes2Int(r) + nl
see "Test Float2Bytes() and Bytes2Float() - Value 77.12" + nl
r = Float2Bytes(77.12)
see "Float Size : " + len(r) + nl
see r + nl
see Bytes2Float(r) + nl
see "Test Double2Bytes() and Bytes2Double() - Value 9999977.12345" + nl
r = Double2Bytes(9999977.12345)
see "Double Size : " + len(r) + nl
see r + nl
decimals(5)
see Bytes2Double(r) + nl

```

SYSTEM FUNCTIONS

In this chapter we are going to learn about the system functions

- System()
- SysGet()
- IsMSDOS()
- IsWindows()
- IsWindows64()
- IsUnix()
- IsMacOSX()
- IsLinux()
- IsFreeBSD()
- IsAndroid()
- Windowsnl()
- Get Command Line Arguments
- Get Active Source File Name
- CurrentDir()
- ExeFileName()
- ChDir()
- ExeFolder()
- Version()
- Shutdown()

33.1 System() Function

We can execute system commands using the system() function

Syntax:

<code>System(cCommand)</code>

Example:

```
System("myapp.exe")    # Run myapp.exe
System("ls")           # print list of files
```

33.2 SysGet() Function

We can get environment variables using the Get() function

Syntax:

```
SysGet(cVariable)
```

Example:

```
see sysget("path")    # print system path information
```

33.3 IsMSDOS() Function

We can check if the operating system is MSDOS or not using the IsMSDOS() function

Syntax:

```
IsMSDOS() ---> Returns 1 if the operating system is MS-DOS, Returns 0 if it's not
```

33.4 IsWindows() Function

We can check if the operating system is Windows or not using the IsWindows() function

Syntax:

```
IsWindows() ---> Returns 1 if the operating system is Windows, Returns 0 if it's not
```

33.5 IsWindows64() Function

We can check if the operating system is Windows 64bit or not using the IsWindows64() function

Syntax:

```
IsWindows64() ---> Returns 1 if the operating system is Windows64, Returns 0 if it's not
```

33.6 IsUnix() Function

We can check if the operating system is Unix or not using the IsUnix() function

Syntax:

```
IsUnix() ---> Returns 1 if the operating system is Unix, Returns 0 if it's not
```


33.7 IsMacOSX() Function

We can check if the operating system is macOS or not using the IsMacOSX() function

Syntax:

```
IsMacOSX() ---> Returns 1 if the operating system is Mac OS X, Returns 0 if it's not
```

33.8 IsLinux() Function

We can check if the operating system is Linux or not using the IsLinux() function

Syntax:

```
IsLinux() ---> Returns 1 if the operating system is Linux, Returns 0 if it's not
```

33.9 IsFreeBSD() Function

We can check if the operating system is FreeBSD or not using the IsFreeBSD() function

Syntax:

```
IsFreeBSD() ---> Returns 1 if the operating system is FreeBSD, Returns 0 if it's not
```

33.10 IsAndroid() Function

We can check if the operating system is Android or not using the IsAndroid() function

Syntax:

```
IsAndroid() ---> Returns 1 if the operating system is Android, Returns 0 if it's not
```

33.11 Example

```
see "IsMSDOS()" --> " + ismsdos() + nl
see "IsWindows()" --> " + iswindows() + nl
see "IsWindows64()" --> " + iswindows64() + nl
see "IsUnix()" --> " + isunix() + nl
see "IsMacOSX()" --> " + ismacosx() + nl
see "IsLinux()" --> " + islinux() + nl
see "IsFreeBSD()" --> " + isfreebsd() + nl
see "IsAndroid()" --> " + isandroid() + nl
```

Output:

```
IsMSDOS() --> 0
IsWindows() --> 1
IsWindows64() --> 0
IsUnix() --> 0
IsMacOSX() --> 0
```

```
IsLinux()      --> 0
IsFreeBSD()    --> 0
IsAndroid()    --> 0
```

33.12 Windowsnl() Function

We can get the windows new line string using the Windowsnl() function.

Syntax:

```
WindowsNL() ---> Returns a string contains CR+LF = CHAR(13) + CHAR(10)
```

Example:

```
cStr = read("input.txt")

if iswindows()
    cStr = substr(cStr, windowsnl(), nl)
ok

aList = str2list(cStr)
# to do - list items processing using "for in"
cStr = list2str(aList)

if iswindows()
    cStr = substr(cStr, nl, windowsnl())
ok

write("ouput.txt", cStr)
```

33.13 Get Command Line Arguments

We can get the command line arguments passed to the ring script using the sysargv variable.

The sysargv variable is a list contains the command line parameters.

Example

```
see copy("=", 30) + nl
see "Command Line Parameters" + nl
see "Size : " + len(sysargv) + nl
see sysargv
see copy("=", 30) + nl
if len(sysargv) < 4 return ok
nStart = sysargv[3]
nEnd = sysargv[4]
for x = nStart to nEnd
    see x + nl
next
```

Output

```
b:\mahmoud\apps\ring>ring tests\syspara.ring 1 10
=====
Command Line Parameters
```

```
Size : 4
ring
tests\syspara.ring
1
10
=====
1
2
3
4
5
6
7
8
9
10
```

33.14 Get Active Source File Name

We can get the active source file name (*.ring) using the filename() function

Syntax:

```
filename() ---> String contains the active source file name.
```

Example:

```
see "Active Source File Name : " + filename() + nl
```

Output:

```
Active Source File Name : tests\filename.ring
```

Example:

```
if sysargv[2] = filename()
    see "I'm the main program file!" + nl
    # we can run tests here!
else
    see "I'm a sub file in a program" + nl
ok
```

33.15 PrevFileName() Function

Using the PrevFileName() function we can get the previous active source file name.

The previous file would be the file of the caller function, Or the file of the function that we called before calling PrevFileName().

Syntax:

```
prevfilename() ---> String contains the previous source file name.
```

Example:

The next function in stdlib.ring uses the PrevFileName() to know if the file of the caller function is the main source file of the program or not.

```

Func IsMainSourceFile
    if PrevFileName() = sysargv[2]
        return true
    ok
    return false

```

33.16 CurrentDir() Function

Return the path of the current directory

Syntax:

```
CurrentDir() ---> String contains the path of the current directory
```

33.17 ExeFileName() Function

Return the Ring executable file name

Syntax:

```
exefilename() ---> String contains the Ring executable file name
```

33.18 ChDir() Function

Change the current directory

Syntax:

```
ChDir(cNewPath)
```

33.19 ExeFolder() Function

Return the Ring executable file path

Syntax:

```
exefolder() ---> String contains the Ring executable path
```

33.20 Version() Function

Return the Ring version

Syntax:

```
version() ---> String contains the Ring version
```

Output:

1.6

33.21 Shutdown() Function

Close the application

Syntax:

```
shutdown(nStatus) ---> Close the application
```

EVAL() AND DEBUGGING

In this chapter we are going to learn about

- Error Handling using Try/Catch/Done
- Eval() function
- Raise() function
- Assert() function

34.1 Try/Catch/Done

Syntax:

```
Try      Statements...
Catch    Statements...
Done
```

The statements in the Try block will be executed, if any error happens then the statements in the catch block will be executed.

Inside the catch block we can use the variable `cCatchError` to get the error message

Example:

```
Try      see 5/0
Catch    see "Catch!" + nl + cCatchError
Done
```

Output:

```
Catch!
Error (R1) : Cann't divide by zero !
```

34.2 Eval() Function

We can execute code during the runtime from string using the Eval() function

Syntax:

```
Eval(cCode)
```

Example:

```
Eval("nOutput = 5+2*5 " )
See "5+2*5 = " + nOutput + nl
Eval("for x = 1 to 10 see x + nl next")
Eval("func test see 'message from test!' ")
test()
```

Output:

```
5+2*5 = 15
1
2
3
4
5
6
7
8
9
10
message from test!
```

We can use the Return command to return a value

Example:

```
see Eval("return 5*5")
```

Output:

```
25
```

34.3 Raise() Function

We can raise an exception using the Raise() function

Syntax:

```
Raise(cErrorMessage)
```

The function will display the error message then end the execution of the program.

We can use Try/Catch/Done to avoid exceptions generated by raise() function.

Example:

```
nMode = 10

if nMode < 0 or nMode > 5
    raise("Error : nMode not in the range 1:4")
ok
```

Output:

```
Line 4 Error : nMode not in the range 1:4
In raise in file tests\raise.ring
```

Example:

```
try
    testmode(6)
catch
    see "avoid raise!"
done

testmode(-1)

func testmode nMode

    if nMode < 0 or nMode > 5
        raise("Error : nMode not in the range 1:4")
    ok
```

Output:

```
avoid raise!
Line 12 Error : nMode not in the range 1:4
In raise In function testmode() in file tests\raise2.ring
called from line 7 in file tests\raise2.ring
```

34.4 Assert() Function

We can use the Assert() function to test conditions before executing the code

If the test fail the program will be terminated with an error message contains the assert condition.

Syntax:

```
Assert( condition )
```

Example:

```
x = 10
assert( x = 10)
assert( x = 100 )
```

Output:

```
Line 3 Assertion Failed!
In assert in file tests\assert.ring
```


DEMO PROGRAMS

In this chapter we will see simple demo programs

- Language Shell
- Main Menu

35.1 Language Shell

We can create simple interactive programming environment using the next program

```
while true
  see nl + "code:> "
  give cCode
  try
    eval(cCode)
  catch
    see cCatchError
  done
end
```

Output:

```
code:> see "hello world"
hello world
code:> for x = 1 to 10 see x + nl next
1
2
3
4
5
6
7
8
9
10

code:> func test see "Hello from test" + nl

code:> test()
Hello from test

code:> bye
```

35.2 Main Menu

Example:

```
# Demo Program

while true

    see "

    Main Menu
    =====
    [1] Say Hello
    [2] Sum two numbers
    [3] Stars
    [4] Fact
    [5] Exit

    " give nMenu see nl

    # we can use Switch-ON-Other-OFF instead of IF-BUT-ELSE-OK

    Switch nMenu
    On 1 sayhello()
    On 2 Sum()
    On 3 Stars()
    On 4

        see "Enter Number : " give x
        see "Output : "

        Try

            see Fact(number(x))

        Catch

            see "Error in parameters!" + nl

        Done

    On "5" return
    Other see "bad option" + nl
    Off

end

func sayhello
    see "Enter your name ? " give fname
    see "Hello " + fname + nl

func sum
    see "number 1 : " give num1 see "number 2 : " give num2
    see "Sum : " see 0 + num1 + num2

func stars
    for x = 1 to 10
        see space(8)
        for y = 1 to x see "*" next see nl
    next

func fact x if x = 0 return 1 else return x * fact(x-1) ok
```

```
func space x y = "" for t=1 to x y += " " next return y
```

Output:

```
Main Menu
=====
[1] Say Hello
[2] Sum two numbers
[3] Stars
[4] Fact
[5] Exit

1

Enter your name ? Mahmoud Fayed
Hello Mahmoud Fayed

Main Menu
=====
[1] Say Hello
[2] Sum two numbers
[3] Stars
[4] Fact
[5] Exit

2

number 1 : 3
number 2 : 4
Sum : 7

Main Menu
=====
[1] Say Hello
[2] Sum two numbers
[3] Stars
[4] Fact
[5] Exit

3

*
**
***
****
*****
*****
*****
*****
*****
*****

Main Menu
=====
[1] Say Hello
[2] Sum two numbers
```

```
[3] Stars
[4] Fact
[5] Exit
```

```
4
```

```
Enter Number : 5
```

```
Output : 120
```

```
Main Menu
```

```
=====
```

```
[1] Say Hello
[2] Sum two numbers
[3] Stars
[4] Fact
[5] Exit
```

```
5
```

ODBC FUNCTIONS

This chapter contains the ODBC functions provided by the Ring programming language.

- `odbc_init()`
- `odbc_drivers()`
- `odbc_datasources()`
- `odbc_close()`
- `odbc_connect()`
- `odbc_disconnect()`
- `odbc_execute()`
- `odbc_colcount()`
- `odbc_fetch()`
- `odbc_getdata()`
- `odbc_tables()`
- `odbc_columns()`
- `odbc_autocommit()`
- `odbc_commit()`
- `odbc_rollback()`

Before using the next function load the `odbc.lib` library

```
load "odbc.lib"
# Use ODBC functions
```

36.1 `odbc_init()` Function

We can create ODBC Handle using the `odbc_init()` function

Syntax:

```
odbc_init() ---> ODBC Handle
```

36.2 odbc_drivers() Function

We can get a list of ODBC drivers using the `odbc_drivers()` function

Syntax:

```
odbc_drivers(ODBC Handle) ---> List of Drivers
```

36.3 odbc_datasources() Function

We can get a list of ODBC data sources using the `odbc_datasources()` function

Syntax:

```
odbc_datasources(ODBC Handle) ---> List of Data sources
```

36.4 odbc_close() Function

After the end of using ODBC functions we can free resources using `ODBC_Close()` function

Syntax:

```
odbc_close(ODBC Handle)
```

36.5 Print List of ODBC Drivers

The next example print a list of ODBC drivers.

```
See "ODBC test 1" + nl
oODBC = odbc_init()
See "Drivers " + nl
see odbc_drivers(oODBC)
odbc_close(oODBC)
```

Output:

```
ODBC test 1
Drivers
Microsoft Access-Treiber (*.mdb) - SQLLevel=0
Driver do Microsoft Paradox (*.db ) - SQLLevel=0
Driver do Microsoft Excel(*.xls) - SQLLevel=0
Microsoft Text Driver (*.txt; *.csv) - SQLLevel=0
Driver da Microsoft para arquivos texto (*.txt; *.csv) - SQLLevel=0
Microsoft dBase-Treiber (*.dbf) - SQLLevel=0
SQL Server - CPOut=60
Microsoft Excel Driver (*.xls) - SQLLevel=0
Driver do Microsoft dBase (*.dbf) - SQLLevel=0
Microsoft Paradox-Treiber (*.db ) - SQLLevel=0
Microsoft ODBC for Oracle - CPOut=120
Microsoft Text-Treiber (*.txt; *.csv) - SQLLevel=0
Microsoft Excel-Treiber (*.xls) - SQLLevel=0
Microsoft Access Driver (*.mdb) - SQLLevel=0
Driver do Microsoft Access (*.mdb) - SQLLevel=0
```

```

Microsoft Paradox Driver (*.db) - SQLLevel=0
Microsoft dBase Driver (*.dbf) - SQLLevel=0
Microsoft Access Driver (*.mdb, *.accdb) - UsageCount=3
Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb) - UsageCount=3
Microsoft Access Text Driver (*.txt, *.csv) - UsageCount=3
SQL Server Native Client 10.0 - UsageCount=1
SQL Server Native Client 11.0 - UsageCount=1
Microsoft Access dBASE Driver (*.dbf, *.ndx, *.mdx) - UsageCount=3
Microsoft Access Paradox Driver (*.db) - UsageCount=3
MySQL ODBC 5.3 ANSI Driver - UsageCount=1
MySQL ODBC 5.3 Unicode Driver - UsageCount=1
ODBC Driver 11 for SQL Server - UsageCount=1
Lianja ODBC Driver - CTimeout=60
Microsoft Visual FoxPro Driver - UsageCount=1
Microsoft Visual FoxPro-Treiber - UsageCount=1
Driver para o Microsoft Visual FoxPro - UsageCount=1
Microsoft FoxPro VFP Driver (*.dbf) - UsageCount=1

```

36.6 Print List of ODBC Data Sources

The next example print a list of ODBC data sources.

```

See "ODBC test 2" + nl
pODBC = odbc_init()
See "Data Sources " + nl
see odbc_datasources(pODBC)
odbc_close(pODBC)

```

Output:

```

ODBC test 2
Data Sources
Excel Files - Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb)
MS Access Database - Microsoft Access Driver (*.mdb, *.accdb)
Customer - Microsoft Access Driver (*.mdb)
IdCardData - Microsoft Access Driver (*.mdb)
MyProjectData2 - Microsoft Access Driver (*.mdb)
MyData - Microsoft Access Driver (*.mdb)
MonprojetData - Microsoft Access Driver (*.mdb)
dBASE Files - Microsoft Access dBASE Driver (*.dbf, *.ndx, *.mdx)
myvfpdata - Microsoft Visual FoxPro Driver
FACTORYDATA - Microsoft Access Driver (*.mdb)
TRAININGSYSDATA - Microsoft Access Driver (*.mdb)
RVCSYSDATASQLDB - SQL Server Native Client 11.0
PWCTRVCDATA - Microsoft Access Driver (*.mdb)
MyCompany - Microsoft Access Driver (*.mdb)
HCS - Microsoft Access Driver (*.mdb)
HCS2 - Microsoft Access Driver (*.mdb, *.accdb)
MyProjectData - Microsoft Access Driver (*.mdb)
Xtreme Sample Database 2008 - Microsoft Access Driver (*.mdb)
Lianja_Southwind - Lianja ODBC Driver
Visual FoxPro Database - Microsoft Visual FoxPro Driver
Visual FoxPro Tables - Microsoft Visual FoxPro Driver

```

36.7 odbc_connect() Function

We can connect to the database using the `odbc_connect()` function.

Syntax:

```
odbc_connect(ODBC Handle, cConnectionString)
```

36.8 odbc_disconnect() Function

We can close the connection to the database using the `odbc_disconnect()` function.

Syntax:

```
odbc_disconnect(ODBC Handle)
```

36.9 Open and Close Connection

The next example connect to the database then close the connection

```
See "ODBC test 3" + nl
pODBC = odbc_init()
See "Connect to database" + nl
see odbc_connect(pODBC, "DBQ=test.mdb;Driver={Microsoft Access Driver (*.mdb)}") + nl
See "disconnect" + nl
odbc_disconnect(pODBC)
See "Close database..." + nl
odbc_close(pODBC)
```

Output:

```
ODBC test 3
Connect to database
1
disconnect
Close database...
```

36.10 odbc_execute() Function

We can execute SQL Statements on the database using the `odbc_execute()` function.

Syntax:

```
odbc_execute(ODBC Handle, cSQLStatement)
```

36.11 odbc_colcount() Function

We can get columns count in the query result using the `odbc_colcount()` function.

Syntax:


```
odbc_colcount(ODBC Handle) ---> Columns Count as Number
```

36.12 odbc_fetch() Function

We can fetch a row from the query result using the `odbc_fetch()` function.

Syntax:

```
odbc_fetch(ODBC Handle)
```

36.13 odbc_getdata() Function

We can get column value from the fetched row using the `odbc_getdata()` function.

Syntax:

```
odbc_getdata(ODBC Handle, nColumnNumber) ---> Column Value
```

36.14 Execute Query and Print Result

The next example execute query then print the query result.

```
See "ODBC test 4" + nl
pODBC = odbc_init()
See "Connect to database" + nl
see odbc_connect(pODBC,"DBQ=test.mdb;Driver={Microsoft Access Driver (*.mdb)}") + nl
See "Select data" + nl
see odbc_execute(pODBC,"select * from person") + nl
nMax = odbc_colcount(pODBC)
See "Columns Count : " + nMax + nl
while odbc_fetch(pODBC)
    See "Row data:" + nl
    for x = 1 to nMax
        see odbc_getdata(pODBC,x) + " - "
    next
end
See "Close database..." + nl
odbc_disconnect(pODBC)
odbc_close(pODBC)
```

36.15 odbc_tables() Function

We can get a list of tables inside the database using the `odbc_tables()` function.

We can access the result of this function as we get any query result.

Syntax:

```
odbc_tables(ODBC Handle)
```

Example:

```

See "ODBC test - Get Database Tables" + nl
pODBC = odbc_init()
See "Connect to database" + nl
odbc_connect(pODBC,"DBQ=test.mdb;Driver={Microsoft Access Driver (*.mdb)}") + nl
See "Select data" + nl
odbc_tables(pODBC) + nl
nMax = odbc_colcount(pODBC)
See "Columns Count : " + nMax + nl
while odbc_fetch(pODBC)
    for x = 1 to nMax
        see odbc_getdata(pODBC,x)
        if x != nMax see " - " ok
    next
    See nl
end
See "Close database..."
odbc_disconnect(pODBC)
odbc_close(pODBC)

```

Output:

```

ODBC test - Get Database Tables
Connect to database
Select data
Columns Count : 5
.\test - NULL - Customer - TABLE - NULL
.\test - NULL - employee - TABLE - NULL
.\test - NULL - person - TABLE - NULL
.\test - NULL - tel - TABLE - NULL
Close database...

```

36.16 odbc_columns() Function

We can get a list of columns inside the table using the odbc_columns() function.

Syntax:

```
odbc_columns(ODBC Handle, cTableName)
```

Example:

```

See "ODBC test - Get Table Columns" + nl
pODBC = odbc_init()
See "Connect to database" + nl
odbc_connect(pODBC,"DBQ=test.mdb;Driver={Microsoft Access Driver (*.mdb)}") + nl
See "Get Columns inside the Person Table" + nl
odbc_columns(pODBC,"person") + nl
while odbc_fetch(pODBC)
    see odbc_getdata(pODBC,4) + nl
end
See "Close database..." + nl
odbc_disconnect(pODBC)
odbc_close(pODBC)

```

Output:

```

ODBC test - Get Table Columns
Connect to database
Get Columns inside the Person Table
FIRST
LAST
STREET
CITY
STATE
ZIP
HIREDATE
MARRIED
AGE
SALARY
NOTES
Close database...

```

36.17 `odbc_autocommit()` Function

We can enable or disable the auto commit feature using the `odbc_autocommit()` function.

Syntax:

```
odbc_autocommit(ODBC Handle, lStatus)    # lStatus can be True or False
```

36.18 `odbc_commit()` Function

We can commit updates to the database using the `odbc_commit()` function.

Syntax:

```
odbc_commit(ODBC Handle)
```

36.19 `odbc_rollback()` Function

We can rollback updates to the database using the `odbc_rollback()` function.

Syntax:

```
odbc_rollback(ODBC Handle)
```

36.20 Transactions and Using Commit and Rollback

Example:

```

See "ODBC Test - Transactions and using Commit and Rollback" + nl
pODBC = odbc_init()
See "Connect to database" + nl
see odbc_connect(pODBC, "DBQ=test.mdb;Driver={Microsoft Access Driver (*.mdb)}") + nl
see "insert data..." + nl
odbc_autocommit(pODBC, 0)

```

```

for x = 1 to 10000
    odbc_execute(pODBC,"insert into tel values (" + x + ", 'mahmoud'")
next
for x = 10001 to 15000
    odbc_execute(pODBC,"insert into tel values (" + x + ", 'samir'")
next
odbc_commit(pODBC)

for x = 15001 to 20000
    odbc_execute(pODBC,"insert into tel values (" + x + ", 'fayed'")
next

ODBC_ROLLBACK(pODBC)
odbc_execute(pODBC,"insert into tel values (" + x + ", 'fayed'")
odbc_commit(pODBC)

See "Close database..." + nl
odbc_disconnect(pODBC)
odbc_close(pODBC)

```

Output:

```

ODBC Test - Transactions and using Commit and Rollback
Connect to database
1
insert data...
Close database...

```

36.21 Save and Restore images

The next example save an image inside the database

```

See "ODBC test - Save image in the database" + nl
pODBC = odbc_init()
See "Connect to database" + nl
see odbc_connect(pODBC,"DBQ=test.mdb;Driver={Microsoft Access Driver (*.mdb)}") + nl
see "Read Image File..." + nl
cFile = str2hex(read("tests\mahmoud.jpg"))
see "size " + len(cFile)+nl
see "Save image in the database..." + nl
stmt = "insert into tel values (20000,'mahmoud','" + cFile + "');"
odbc_execute(pODBC,stmt)
See "Close database..." + nl
odbc_disconnect(pODBC)
odbc_close(pODBC)

```

The next example restore the image from the database

```

See "ODBC Test - Restore image from the database" + nl
pODBC = odbc_init()
See "Connect to database" + nl
see odbc_connect(pODBC,"DBQ=test.mdb;Driver={Microsoft Access Driver (*.mdb)}") + nl
See "Select data" + nl
see odbc_execute(pODBC,"select * from tel where id = 20000") + nl
nMax = odbc_colcount(pODBC)
See "Columns Count : " + nMax + nl
if odbc_fetch(pODBC)

```

```
    See "Write image file" + nl
    write("tests\great.jpg",hex2str( odbc_getdata(pODBC,3) ) )
ok
See "Close database..." + nl
odbc_disconnect(pODBC)
odbc_close(pODBC)
```

MYSQL FUNCTIONS

In this chapter we are going to learn about the MySQL functions provided by the Ring programming language.

- MySQL_Info()
- MySQL_Init()
- MySQL_Error()
- MySQL_Connect()
- MySQL_Close()
- MySQL_Query()
- MySQL_Insert_ID()
- MySQL_Result()
- MySQL_Next_Result()
- MySQL_Columns()
- MySQL_Result2()
- MySQL_Escape_String()
- MySQL_AutoCommit()
- MySQL_Commit()
- MySQL_Rollback()

Before using the next function load the mysql.lib ring library

```
load "mysql.lib.ring"  
# Use MySQL functions
```

37.1 MySQL_Info() Function

We can get the MySQL Client version using the MySQL_Info() function.

Syntax:

```
MySQL_Info() ----> string contains the MySQL Client version
```

Example:

```
see "MySQL Client Version : " + mysql_info()
```

Output:

```
MySQL Client Version : 6.1.5
```

37.2 MySQL_Init() Function

We can start using MySQL Client through the MySQL_Init() function.

Syntax:

```
MySQL_Init() ---> MySQL Handle
```

37.3 MySQL_Error() Function

We can get the error message from the MySQL Client using the MySQL_Error() function.

Syntax:

```
MySQL_Error(MySQL Handle) ---> Error message as string
```

37.4 MySQL_Connect() Function

We can connect to the MySQL database server using the MySQL_Connect() function.

Syntax:

```
MySQL_Connect(MySQL Handle, cServer, cUserName, cPassword) ---> lStatus
```

37.5 MySQL_Close() Function

We can close the connection to the MySQL database using the MySQL_Close() function

Syntax:

```
MySQL_Close(MySQL Handle)
```

37.6 MySQL_Query() Function

We can execute SQL queries using the MySQL_Query() function

Syntax:

```
MySQL_Query(MySQL Handle, cSQLQuery)
```

37.7 Create Database

The next example connect to MySQL Server then create new database.

```
See "MySQL Test - Create Database" + nl
con = mysql_init()

See "Connect" + nl
if mysql_connect(con, "localhost", "root", "root") = 0
    see "Cann't connect" + nl
    see "Error : " + mysql_error(con) + nl
    mysql_close(con)
    bye
ok

See "Create Database..." + nl
mysql_query(con, "CREATE DATABASE mahdb")

See "Close Connection" + nl
mysql_close(con)
```

Output:

```
MySQL Test - Create Database
Connect
Create Database...
Close Connection
```

37.8 Create Table and Insert Data

The next example create new table and insert records

```
func main
    see "Create Table and Insert Records" + nl
    con = mysql_init()

    see "Connect" + nl
    if mysql_connect(con, "localhost", "root", "root", "mahdb") = 0
        system_error(con)
    ok

    see "Drop table" + nl
    if mysql_query(con, "DROP TABLE IF EXISTS Employee")
        system_error(con) ok

    see "Create table" + nl
    if mysql_query(con, "CREATE TABLE Employee(Id INT, Name TEXT, Salary INT)")
        system_error(con) ok

    see "Insert data" + nl
    if mysql_query(con, "INSERT INTO Employee VALUES(1, 'Mahmoud', 15000)")
        system_error(con) ok

    if mysql_query(con, "INSERT INTO Employee VALUES(2, 'Samir', 16000)")
        system_error(con) ok

    if mysql_query(con, "INSERT INTO Employee VALUES(3, 'Fayed', 17000)")
```



```

        system_error(con)  ok

    see "Close connection" + nl
    mysql_close(con)

func system_error con
    see mysql_error(con)  mysql_close(con)  bye

```

Output:

```

Create Table and Insert Records
Connect
Drop table
Create table
Insert data
Close connection

```

37.9 MySQL_Insert_ID() Function

We can get the inserted row id using the MySQL_Insert_ID() function

Syntax:

```
MySQL_Insert_ID() ---> Inserted row id as number
```

Example:

```

con = mysql_init()
see "connect to database" + nl
mysql_connect(con,"localhost","root","root","mahdb")
see "drop table" + nl
mysql_query(con, "DROP TABLE IF EXISTS Customers")
see "create table" + nl
mysql_query(con, "CREATE TABLE Customers(Id INT PRIMARY KEY AUTO_INCREMENT, Name TEXT)")
see "insert record" + nl
mysql_query(con, "INSERT INTO Customers(Name) VALUES ('Mahmoud')")
see "insert record" + nl
mysql_query(con, "INSERT INTO Customers(Name) VALUES ('Samir')")
see "insert record" + nl
mysql_query(con, "INSERT INTO Customers(Name) VALUES ('Fayed')")
see "insert record" + nl
mysql_query(con, "INSERT INTO Customers(Name) VALUES ('Test 2015')")

see "inserted row id : " + mysql_insert_id(con) + nl
see "close database" + nl
mysql_close(con)

```

Output:

```

connect to database
drop table
create table
insert record
insert record
insert record
insert record
inserted row id : 4
close database

```

37.10 MySQL_Result() Function

We can get the query result (data without column names) using the MySQL_Result() function.

Syntax:

```
MySQL_Result (MySQL Handle) ---> List contains the query result
```

37.11 MySQL_Next_Result() Function

We can move to the next query result using the MySQL_Next_Result() function. We use this function when we have multiple SQL statements in the same query.

Syntax:

```
MySQL_Next_Result (MySQL Handle)
```

37.12 Print Query Result

The next example execute a query on the database then print the result.

```
con = mysql_init()
see "Connect to database" + nl
mysql_connect(con, "localhost", "root", "root", "mahdb")
see "Execute Query" + nl
mysql_query(con, "SELECT Name FROM Employee WHERE Id=1;" +
                "SELECT Name FROM Employee WHERE Id=3")
see "Print Result" + nl
see mysql_result(con)
mysql_next_result(con)
see mysql_result(con)
see "close database" + nl
mysql_close(con)
```

Output:

```
Connect to database
Execute Query
Print Result
Mahmoud
Fayed
close database
```

37.13 MySQL_Columns() Function

We can get a list of columns names using the MySQL_Columns() function.

Syntax:

```
MySQL_Columns(MySQL Handle) ---> List contains columns information
```

Example:

```
con = mysql_init()
see "Connect to database" + nl
mysql_connect(con, "localhost", "root", "root", "mahdb")
see "Execute Query" + nl
mysql_query(con, "SELECT * FROM Employee")
see "Result" + nl
see mysql_columns(con)
see "Close database" + nl
mysql_close(con)
```

Output:

```
Connect to database
Execute Query
Result
Id
11
3
32768
Name
65535
252
16
Salary
11
3
32768
Close database
```

37.14 MySQL_Result2() Function

Instead of using MySQL_Result() to get the result data without columns names, we can use the MySQL_Result2() to get all of the column names then the query result in one list.

Syntax:

```
MySQL_Result2(MySQL Handle) ---> List (query result starts with columns names)
```

Example:

```
con = mysql_init()
see "Connect to database" + nl
mysql_connect(con, "localhost", "root", "root", "mahdb")
see "Execute Query" + nl
mysql_query(con, "SELECT * FROM Employee")
see "Print Result" + nl
see mysql_result2(con)
see "Close database" + nl
mysql_close(con)
```

Output:

```

Connect to database
Execute Query
Print Result
Id
Name
Salary
1
Mahmoud
15000
2
Samir
16000
3
Fayed
17000
Close database

```

37.15 MySQL_Escape_String() Function

We can store binary data and special characters in the database after processing using MySQL_Escape_String() function

Syntax:

```
MySQL_Escape_String(MySQL Handle, cString) ---> String after processing
```

37.16 Save Image inside the database

Example:

```

See "Read file" + nl
cFile = read("tests\mahmoud.jpg")
con = mysql_init()
See "Connect to database..." + nl
mysql_connect(con, "localhost", "root", "root", "mahdb")
See "Escape string..." + nl
cFile = mysql_escape_string(con, cFile)
stmt = "INSERT INTO photo(id, data) VALUES(1, '" + cFile + "')"
See "Insert data..." + nl
mysql_query(con, stmt)
See "Close database..." + nl
mysql_close(con)

```

Output:

```

Read file
Connect to database...
Escape string...
Insert data...
Close database...

```

37.17 Restore Image From The Database

Example:

```
con = mysql_init()
See "Connect to database..." + nl
mysql_connect(con, "localhost", "root", "root", "mahdb")
See "Read data from database..." + nl
mysql_query(con, "SELECT data FROM photo WHERE id=1")
See "Write new file" + nl
result = mysql_result(con)
write("tests\mahmoud2.jpg", result[1][1])
See "Close database..." + nl
mysql_close(con)
```

Output:

```
Connect to database...
Read data from database...
Write new file
Close database...
```

37.18 MySQL_AutoCommit() Function

We can enable or disable the auto commit feature using the MySQL_AutoCommit() function.

Syntax:

```
MySQL_AutoCommit (MySQL Handle, lStatus)  # lstatus can be True/False
```

37.19 MySQL_Commit() Function

We can commit updates to the database using the MySQL_Commit() function.

Syntax:

```
MySQL_Commit (MySQL Handle)
```

37.20 MySQL_Rollback() Function

We can rollback updates to the database using the MySQL_Rollback() function.

Syntax:

```
MySQL_Rollback (MySQL Handle)
```

37.21 Transaction Example

The next example presents the usage of MySQL_Autocommit(), MySQL_Commit() & MySQL_RollBack() functions.

Example:

```

func main

    con = mysql_init()

    see "Connect" + nl
    if mysql_connect(con, "localhost", "root", "root","mahdb") = 0
        system_error(con) ok

    see "Drop table" + nl
    if mysql_query(con, "DROP TABLE IF EXISTS Employee2")
        system_error(con) ok

    see "Create table" + nl
    if mysql_query(con, "CREATE TABLE Employee2(Id INT, Name TEXT, Salary INT)")
        system_error(con) ok

    see "Insert data" + nl
    if mysql_query(con, "INSERT INTO Employee2 VALUES(1,'Mahmoud',15000)")
        system_error(con) ok

    if mysql_query(con, "INSERT INTO Employee2 VALUES(2,'Samir',16000)")
        system_error(con) ok

    if mysql_query(con, "INSERT INTO Employee2 VALUES(3,'Fayed',17000)")
        system_error(con) ok

    mysql_autocommit(con,False)
    mysql_query(con, "INSERT INTO Employee2 VALUES(4,'Ahmed',5000)")
    mysql_query(con, "INSERT INTO Employee2 VALUES(5,'Ibrahim',50000)")
    mysql_query(con, "INSERT INTO Employee2 VALUES(6,'Mohammed',50000)")
    See "Save transaction (y/n) " give nChoice
    if upper(nChoice) = "Y"
        mysql_commit(con)
    else
        mysql_rollback(con)
    ok

    see "Close connection" + nl
    mysql_close(con)

func system_error con

    see mysql_error(con)
    mysql_close(con)
    bye

```

Output:

```

Connect
Drop table
Create table
Insert data
Save transaction (y/n) y
Close connection

```

SQLITE FUNCTIONS

In this chapter we will learn about using the SQLite database in the Ring programming language.

Before using the next function load the `sqlitelib.ring` library

```
load "sqlitelib.ring"  
# Use SQLite functions
```

38.1 `sqlite_init()` function

Syntax:

```
sqlite_init() ---> SQLite Object
```

38.2 `sqlite_open()` function

Syntax:

```
sqlite_open(SQLite Object, cFileName)
```

38.3 `sqlite_execute()` function

Syntax:

```
sqlite_execute(SQLite Object, cSQLStatement)
```

38.4 `sqlite_close()` function

Syntax:

```
sqlite_close(SQLite Object)
```

38.5 Example

The next code create a SQLite database, add new records then display the data.

```
oSQLite = sqlite_init()

sqlite_open(oSQLite,"mytest.db")

sql = "CREATE TABLE COMPANY(" +
      "ID INT PRIMARY KEY     NOT NULL," +
      "NAME           TEXT     NOT NULL," +
      "AGE            INT       NOT NULL," +
      "ADDRESS        CHAR(50)," +
      "SALARY         REAL );"

sqlite_execute(oSQLite,sql)

sql = "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) " +
      "VALUES (1, 'Mahmoud', 29, 'Jeddah', 20000.00 ); " +
      "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) " +
      "VALUES (2, 'Ahmed', 27, 'Jeddah', 15000.00 ); " +
      "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY)" +
      "VALUES (3, 'Mohammed', 31, 'Egypt', 20000.00 );" +
      "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY)" +
      "VALUES (4, 'Ibrahim', 24, 'Egypt ', 65000.00 );"

sqlite_execute(oSQLite,sql)

aResult =  sqlite_execute(oSQLite,"select * from COMPANY")
for x in aResult
    for t in x
        see t[2] + nl
    next
next
see copy(" ",50) + nl
for x in aResult
    see x["name"] + nl
next
sqlite_close(oSQLite)
```

Output:

```
1
Mahmoud
29
Jeddah
20000.0
2
Ahmed
27
Jeddah
15000.0
3
Mohammed
31
Egypt
20000.0
4
```



```
Ibrahim
24
Egypt
65000.0
*****
Mahmoud
Ahmed
Mohammed
Ibrahim
```

SECURITY AND INTERNET FUNCTIONS

This chapter contains the security and internet functions provided by the Ring programming language for Hashing, Encryption & Decryption.

Before using the next function load the openssllib.ring library

```
load "openssllib.ring"  
# Use OpenSSL functions
```

- MD5()
- SHA1()
- SHA256()
- SHA512()
- SHA384()
- SHA224()
- Encrypt()
- Decrypt()
- Randbytes()

Before using the next function load the internetlib.ring library

```
load "internetlib.ring"  
# Use the Internet functions
```

- Download()
- SendEmail()

39.1 MD5() Function

We can calculate the MD5 hash using the MD5() Function

Syntax:

```
MD5(cString) ---> String contains the MD5 hash of the string cString
```

Example:

```
see "md5('happy') = " + md5("happy") + nl +  
    "md5('Hello') = " + md5("Hello") + nl
```

Output:

```
md5('happy') = 56ab24c15b72a457069c5ea42fcfc640
md5('Hello') = 8b1a9953c4611296a827abf8c47804d7
```

39.2 SHA1() Function

We can calculate the SHA1 hash using the SHA1() Function

Syntax:

```
SHA1(cString) ---> String contains the SHA1 hash of the string cString
```

Example:

```
see "sha1('hello') : " + sha1("hello") + nl +
    "sha1('apple') : " + sha1("apple") + nl
```

Output:

```
sha1('hello') : aaf4c61ddcc5e8a2dabede0f3b482cd9aea9434d
sha1('apple') : d0be2dc421be4fcd0172e5afceea3970e2f3d940
```

39.3 SHA256() Function

We can calculate the SHA256 hash using the SHA256() Function

Syntax:

```
SHA256(cString) ---> String contains the SHA256 hash of the string cString
```

Example:

```
see "sha256('hello') : " + sha256("hello") + nl +
    "sha256('apple') : " + sha256("apple") + nl
```

Output:

```
sha256('hello') : 2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824
sha256('apple') : 3a7bd3e2360a3d29eea436fcfb7e44c735d117c42d1c1835420b6b9942dd4f1b
```

39.4 SHA512() Function

We can calculate the SHA512 hash using the SHA512() Function

Syntax:

```
SHA512(cString) ---> String contains the SHA512 hash of the string cString
```

Example:

```
see "sha512('hello') : " + sha512("hello") + nl +
    "sha512('apple') : " + sha512("apple") + nl +
    "sha512('hello world') : " + sha512("hello world") + nl
```

Output:

```
sha512('hello') : 9b71d224bd62f3785d96d46ad3ea3d73319bfbcb2890caadae2dff72519673c
a72323c3d99ba5c11d7c7acc6e14b8c5da0c4663475c2e5c3adef46f73bcdec043
sha512('apple') : 844d8779103b94c18f4aa4cc0c3b4474058580a991fba85d3ca698a0bc9e52
c5940feb7a65a3a290e17e6b23ee943ecc4f73e7490327245b4fe5d5efb590feb2
sha512('hello world') : 309ecc489c12d6eb4cc40f50c902f2b4d0ed77ee511a7c7a9bcd3ca8
6d4cd86f989dd35bc5ff499670da34255b45b0cfd830e81f605dcf7dc5542e93ae9cd76f
```

39.5 SHA384() Function

We can calculate the SHA384 hash using the SHA384() Function

Syntax:

```
SHA384(cString) ---> String contains the SHA384 hash of the string cString
```

Example:

```
see "sha384('hello') : " + sha384("hello") + nl +
    "sha384('apple') : " + sha384("apple") + nl +
    "sha384('hello world') : " + sha384("hello world") + nl
```

Output:

```
sha384('hello') : 59e1748777448c69de6b800d7a33bbfb9ff1b463e44354c3553bcdb9c666fa
90125a3c79f90397bdf5f6a13de828684f
sha384('apple') : 3d8786fcb588c93348756c6429717dc6c374a14f7029362281a3b21dc10250
ddf0d0578052749822eb08bc0dc1e68b0f
sha384('hello world') : fdbd8e75a67f29f701a4e040385e2e23986303ea10239211af907fcb
b83578b3e417cb71ce646efd0819dd8c088de1bd
```

39.6 SHA224() Function

We can calculate the SHA224 hash using the SHA224() Function

Syntax:

```
SHA224(cString) ---> String contains the SHA224 hash of the string cString
```

Example:

```
see "sha224('hello') : " + sha224("hello") + nl +
    "sha224('apple') : " + sha224("apple") + nl +
    "sha224('hello world') : " + sha224("hello world") + nl
```

Output:

```
sha224('hello') : ea09ae9cc6768c50fcee903ed054556e5bfc8347907f12598aa24193
sha224('apple') : b7bbfdf1a1012999b3c466fdeb906a629caa5e3e022428d1eb702281
sha224('hello world') : 2f05477fc24bb4faefd86517156dafdecec45b8ad3cf2522a563582b
```

39.7 Encrypt() Function

We can use the Encrypt() function to encrypts the data using the Blowfish algorithm.

Syntax:

```
Encrypt(cString, cKey, cIV) ---> Encrypted string
```

39.8 Decrypt() Function

We can use the Decrypt() function to decrypt the data encrypted using the Encrypt() function.

Syntax:

```
Decrypt(cCipher, cKey, cIV) ---> Decrypted string
```

39.9 Encryption and Decryption Example

The next example demonstrates how to use the Encrypt() and Decrypt() functions.

These functions use the Blowfish algorithm.

```
See "Enter a string : " give cStr
list = 0:15 cKey=""      for x in list cKey += char(x) next
list = 1:8  cIV = ""     for x in list cIV += char(x) next
cStr = Encrypt(cStr,cKey,cIV)
See "Cipher Text      : " + cStr + nl +
    "Plain Text       : " + Decrypt(cStr,cKey,cIV) + nl
```

39.10 File Hash

The next example demonstrates how to calculate the hash functions for files

```
cStr = read("myapp.exe")
see "Size : " + len(cStr) + nl +
    "md5 : " + md5(cStr) + nl +
    "sha1 : " + sha1(cStr) + nl +
    "sha256 : " + sha256(cStr) + nl +
    "sha224 : " + sha224(cStr) + nl +
    "sha384 : " + sha384(cStr) + nl +
    "sha512 : " + sha512(cStr) + nl
```

Output:

```
Size : 58079876
md5 : 762eee15d8d2fd73b71ea52538b28667
sha1 : 9212c0c7258bad89a62bd239e1358a9276a9d070
sha256 : 7d6724e69b6c553da749ba31b6185dddc965129b64d9e9bf3de88f67df3b1cdc
sha224 : 5a9c8a7d662bce4f880ba94f90a79362b672528b9efd5abc718c7a3d
sha384 : 18e23f973abedbeb3981c423f12aeadecf96f9c6fb28aeabe3be4c484f8540afcc3861b
b370ce2b59cf3c99c130b856b
sha512 : da3d5e997d06f8b2a7a9964b77f7d82eedb76b245c611082c1639f83f51d83880bcd08f
cd53dcab1167bdca0b82fec5071971ac17c76479d76985ced4ab0d18e
```

39.11 Randbytes() Function

We can generate a string of pseudo-random bytes using the Randbytes() function.

Syntax:

```
Randbytes(nSize) ---> String contains random bytes (bytes count = nSize)
```

Example:

```
salt = randbytes(32)
password = "SecretPassWord@$$%123"
see salt + nl
see sha256("test" + salt) + nl
```

39.12 Download() Function

Syntax:

```
Download(cURL) ---> String contains the server response
```

Example:

```
cStr= download("http://doublesvsoop.sourceforge.net/")
see cStr
write("download.txt",cStr)
```

39.13 SendEmail() Function

Syntax:

```
SendEmail(cSMTPServer,cEmail,cPassword,cSender,cReceiver,cCC,cTitle,cContent)
```

Example:

```
See "Send email..." + nl
sendemail("smtp://smtp.gmail.com:587",
    "email@gmail.com",
    "password",
    "email@gmail.com",
    "somebody@yahoo.com",
    "somebodyelse@yahoo.com",
    "Sending email from Ring",
    "Hello
    How are you?
    Are you fine?
    Thank you!
    Greetings,
    Mahmoud")
see "Done.." + nl
```

OBJECT ORIENTED PROGRAMMING (OOP)

In this chapter we are going to learn how to use the Object-Oriented programming paradigm inside the Ring programming language.

We will learn about

- Classes and Objects
- Access Objects Using Braces
- Composition
- Setter and Getter
- Private Attributes and Methods
- Operator Overloading
- Inheritance
- Dynamic Attributes
- Packages
- Printing Objects
- Find() and List of Objects
- Sort() and List of Objects
- Using Self.Attribute and Self.Method()
- Using This.Attribute and This.Method()

40.1 Classes and Objects

We can define new classes using the next syntax

Syntax:

```
Class <Class Name> [From|<|: <Parent Class Name>]
    [Attributes]
    [Methods]
    [Private
        [Attributes]
        [Methods]
    ]
```

And we can create objects using the next syntax

Syntax:

```
New <Object Name> [ (init method parameters) ] |
                  [ { access object data and methods } ]    ---> Object
```

Example:

```
New point { x=10 y=20 z=30 print() }
Class Point x y z func print see x + nl + y + nl + z + nl
```

Note: We can use { } to access object data and methods.

Tip: we can declare the class attributes directly after the class name.

Output:

```
10
20
30
```

We can rewrite the same program in another style

```
New point                                # create new object using the point class
{                                         # access the new object attributes and methods
    x = 10                               # set the x attribute to 10
    y = 20                               # set the y attribute to 20
    z = 30                               # set the z attribute to 30
    print()                             # call the print method
}                                         # end of object access

Class Point                             # define the Point class
    x y z                               # the class contains three attributes x, y & z
    func print                           # define the print method
        see x + nl +                     # print the x attribute
            y + nl +                     # print the y attribute
            z + nl                       # print the z attribute
```

Also we can write the same program in another way

```
P1 = New Point
P1.x = 10
P1.y = 20
P1.z = 30
P1.Print()
Class Point x y z func print see x + nl + y + nl + z + nl
```

Note: we can use the dot operator after the object name to access object members.

Also we can write the same program in another way

```
new point { print() }
Class Point
    x = 10 y = 20 z = 30
    func print see x + nl + y + nl + z + nl
```


Note: we can set the default values for the class attributes when we declare them.

Also we can write the same program in another way

```
new point(10,20,30)
Class Point
    x y z
    func init p1,p2,p3 x=p1 y=p2 z=p3 print()
    func print see x + nl + y + nl + z + nl
```

Note: we can call the init method directly using () when we create new objects

Also we can write the same program in another way

```
new point( [ :x = 10 , :y = 20 , :z = 30 ] )
Class Point x y z
    func init aPara x = aPara[:x] y = aPara[:y] z = aPara[:z] print()
    func print see x + nl + y + nl + z + nl
```

Tip: using Hash for passing method parameters enable us to create optional parameters and change the order of parameters when adding them to the Hash.

40.2 Access Objects Using Braces

We can access the object at any time using braces { }

Inside the braces we can use the object attributes and methods directly

This can be done when we create the object using the New keyword or at any time using the next syntax

```
ObjectName { access object data and methods }
```

Example:

```
See "Creating the Object" + nl
o1 = new Point
See "Using the Object" + nl
o1 {
    x=5
    y=15
    z=25
    print()
}
Class Point x y z func print see x + nl + y + nl + z
```

We can use braces to access objects when we call functions or methods

Example:

```
o1 = new Point

print( o1 { x=10 y=20 z=30 } )

func print object
    see object.x + nl +
        object.y + nl +
```

```
object.z
```

```
Class Point x y z
```

We can mix between using braces and the dot operator to access the object in the same expression.

Example:

```
o1 = new Point
O1 { x=10 y=20 z=30 }.print()

Class Point x y z
    func print see x + nl + y + nl + z
```

40.3 Composition

The object may contains other objects as attributes.

Using braces to access objects can be nested.

Example:

```
R1 = New Rectangle
{
    Name = "Rectangle 1"

    P1
    {
        X = 10
        Y = 20
    }

    P2
    {
        X = 200
        Y = 300
    }

    Color = "Blue"
}

see "Name : " + R1.Name + nl +
    "Color: " + R1.Color + nl +
    "P1   : (" + R1.P1.X + ", " + R1.P1.Y + ")" + nl +
    "P2   : (" + R1.P2.X + ", " + R1.P2.Y + ")"

Class Rectangle
    name  color
    p1 = new Point
    p2 = new Point

Class Point x y
```

Output:

```
Name : Rectangle 1
Color: Blue
P1   : (10,20)
P2   : (200,300)
```

40.4 Setter and Getter

We can define methods to be used when we set and get object attributes.

Syntax:

```
Class ClassName

    AttributeName
    ...

    Func SetAttributeName
        ...

    Func GetAttributeName
        ...
```

Example:

```
o1 = new person

o1.name = "Mahmoud" see o1.name + nl

o1 { name = "Ahmed" see name }

Class Person

    name family = "Fayed"

    func setname value
        see "Message from SetName() Function!" + nl
        name = value + " " + family

    func getname
        see "Message from GetName() Function!" + nl
        return "Mr. " + name
```

Output:

```
Message from SetName() Function!
Message from GetName() Function!
Mr. Mahmoud Fayed
Message from SetName() Function!
Message from GetName() Function!
Mr. Ahmed Fayed
```

40.5 Private Attributes and Methods

We can define private attributes and methods after the keyword private inside the class body

Example:

```
o1 = new person {
    name = "Test"
    age = 20
    print()
    o1.printsalary()
}

try
    see o1.salary
catch
    see cCatchError + nl
done

try
    o1.increasesalary(1000)
catch
    see cCatchError + nl
done

Class Person

    name age

    func print
        see "Name    : " + name + nl +
            "Age     : " + age + nl

    func printsalary
        see "Salary : " + salary + nl

    private

    salary = 15000

    func increasesalary x
        salary += x
```

Output:

```
Name    : Test
Age     : 20
Salary  : 15000
Error (R27) : Using private attribute from outside the class : salary
Error (R26) : Calling private method from outside the class : increasesalary
```

40.6 Operator Overloading

We can add the operator method to our class to enable using operators with the class objects.

Syntax:

```
Class ClassName

    ...
```

```
Func operator cOperator,Para
```

```
...
```

The function operator takes two paramters, the first represent the operator and the second represent the second parameter after the operator.

Example:

```
o1 = new point { x = 10 y = 10 print("P1   : ") }
o2 = new point { x = 20 y = 40 print("P2   : ") }

o3 = o1 + o2
o3.print("P1+P2 : ")

class point x y

    func operator cOperator,Para
        result = new point
        switch cOperator
        on "+"
            result.x = x + Para.x
            result.y = y + Para.y
        on "-"
            result.x = x - Para.x
            result.y = y - Para.y
        off
        return result

    func print cPoint
        see cPoint + "X : " + x + " Y : " + y + nl
```

Output:

```
P1   : X : 10 Y : 10
P2   : X : 20 Y : 40
P1+P2 : X : 30 Y : 50
```

40.7 Inheritance

We can create class from another class in the class definition using the keyword from.

Syntax:

```
Class <Class Name> [From <Parent Class Name>]
```

We can call a method in the parent class from the child class using the super object.

Syntax:

```
func methodname
    ...
    super.methodname()
    ...
```

Example:

```

Func main
    e1 = new Employee {
        Name = "test"
        age = 20
        job = "programmer"
        salary = 20000000
        print()
    }

Class Human
    Name Age
    func print
        see "Name : " + name + nl + "Age : " + age + nl

Class Employee from Human
    Job Salary
    func print
        super.print()
        see "Job : " + job + nl + "Salary : " + salary + nl

```

Output:

```

Name : test
Age  : 20
Job  : programmer
Salary : 20000000

```

40.8 Dynamic Attributes

We can write instructions after the class name to be executed when we create new objects

Example:

```

o1 = new dynamicClass
see o1.var5 + nl           # output 5

Class DynamicClass
    for x = 1 to 10
        cStr = "var" + x + " = " + x
        eval(cStr)
    next

```

Tip: in the previous example var1, var2, ..., var10 will be defined as attributes.

Tip: The problem with the previous example is that x and cStr will be defined as attributes too!

Note: we can write class definitions inside a string then using eval() we can execute the string to define the classes

40.9 Packages

We can create a package (a group of classes under a common name) using the next syntax

```
package PackageName
    Class Class1
        ...
    Class Class2
        ...
    Class Class3
        ...
    ...
```

Example

```
o1 = new System.output.console
o1.print("Hello World")

Package System.Output
    Class Console
        Func Print cText
            see cText + nl
```

Note: we can use the dot operator as part of the package name

Instead of typing the long name `PackageName.ClassName` we can use the import command

When we import a package, we can use any class inside this package directly.

Example

```
import system.output
o1 = new console {
    print("Hello World")
}

Package System.Output
    Class Console
        Func Print cText
            see cText + nl
```

40.10 Printing Objects

We can print the object state (attributes and values) using the `see` command.

Example:

```
see new point { x=10 y=20 z=30 }
class point x y z
```

Output:

```
x: 10.000000
y: 20.000000
z: 30.000000
```

40.11 Find() and List of Objects

We can use the `find()` function to search inside a list of objects.

Syntax:

```
Find(List,ItemValue,nColumn,cAttribute) ---> Item Index
```

Example:

```
myList1 = [new Company {position=3 name="Mahmoud" symbol="MHD"},
           new Company {position=2 name="Bert" symbol="BRT"},
           new Company {position=1 name="Ring" symbol="RNG"}
]

see find(myList1,"Bert",1,"name") + nl
see find(myList1,"Ring",1,"name") + nl
see find(myList1,"Mahmoud",1,"name") + nl
see find(myList1,"RNG",1,"symbol") + nl
see find(myList1,"MHD",1,"symbol") + nl
see find(myList1,"BRT",1,"symbol") + nl
see find(myList1,3,1,"position") + nl
see find(myList1,1,1,"position") + nl
see "Other" + nl
see find(myList1,"test",1,"name") + nl
see find(myList1,"test",0,"name") + nl
see find(myList1,"test",5,"name") + nl

class company position name symbol
```

Output:

```
2
3
1
3
1
2
1
3
Other
0
0
0
```

40.12 Sort() and list of objects

We can sort a list of objects based on an object attribute using the Sort() function.

Syntax:

```
Sort(List,nColumn,cAttribute) ---> Sorted List based on Object Attribute
```

Example:

```
myList1 = [
    new Company {position=3 name="Mahmoud" symbol="MHD"},
    new Company {position=2 name="Bert" symbol="BRT"},
    new Company {position=8 name="Charlie" symbol="CHR"},
    new Company {position=6 name="Easy" symbol="FEAS"},
    new Company {position=7 name="Fox" symbol="EFOX"},
    new Company {position=5 name="Dog" symbol="GDOG"},
    new Company {position=4 name="George" symbol="DGRG"},
]
```



```

        new Company {position=1 name="Ring" symbol="RNG"}
    ]

    see sort(mylist1,1,"name")
    see copy(" ",70) + nl
    see sort(mylist1,1,"symbol")
    see copy(" ",70) + nl
    see sort(mylist1,1,"position")

class company position name symbol

```

Output:

```

position: 2.000000
name: Bert
symbol: BRT
position: 8.000000
name: Charlie
symbol: CHR
position: 5.000000
name: Dog
symbol: GDOG
position: 6.000000
name: Easy
symbol: FEAS
position: 7.000000
name: Fox
symbol: EFOX
position: 4.000000
name: George
symbol: DGRG
position: 3.000000
name: Mahmoud
symbol: MHD
position: 1.000000
name: Ring
symbol: RNG
*****
position: 2.000000
name: Bert
symbol: BRT
position: 8.000000
name: Charlie
symbol: CHR
position: 4.000000
name: George
symbol: DGRG
position: 7.000000
name: Fox
symbol: EFOX
position: 6.000000
name: Easy
symbol: FEAS
position: 5.000000
name: Dog
symbol: GDOG
position: 3.000000
name: Mahmoud

```

```

symbol: MHD
position: 1.000000
name: Ring
symbol: RNG
*****
position: 1.000000
name: Ring
symbol: RNG
position: 2.000000
name: Bert
symbol: BRT
position: 3.000000
name: Mahmoud
symbol: MHD
position: 4.000000
name: George
symbol: DGRG
position: 5.000000
name: Dog
symbol: GDOG
position: 6.000000
name: Easy
symbol: FEAS
position: 7.000000
name: Fox
symbol: EFOX
position: 8.000000
name: Charlie
symbol: CHR

```

40.13 Using Self.Attribute and Self.Method()

Inside the class region (After the class name and before any method) and the class methods we can use self.attribute and self.method()

```

Class Point
    self.x = 10
    self.y = 20
    self.z = 30
    func print
        see self.x + nl + self.y + nl + self.z + nl

```

Note: using self.attribute in the class region to define the class attribute protect the class attributes from conflict with global variables.

Tip: if you typed the class attributes with self.attribute and there are a global variable with the same name it will be used and the attribute will not be defined.

Check the “Scope Rules” chapter to know about the conflict between the global variable name and the attribute name

Whay this may happens?

Because

- Because in the class region we can access global variables.

- Before defining any variable, Ring try to find the variable and use it if it's found.

Note: Try to avoid the global variables, use the main function and start their names with \$

Tip: In large programs protect your classes and define their members using self.attribute

40.14 Using This.Attribute and This.Method()

Inside class methods we have access to the object scope directly. we don't need to use Self.attribute or Self.method to read/write attribute and call methods.

But we can use braces {} while we are inside methods to access another object, In this case the current object scope will be changed while we are inside the brace.

How we can get access to our class attributes and methods while we are inside braces?

This can be done using This.Attribute and This.Method()

Example:

```
new point

class point
  x=10 y=20 z=30
  print()
  func print
    new UI {
      display(this.x,this.y,this.z)
    }
end class point

Class UI
  func display x,y,z
    see x + nl + y + nl + z + nl
```

FUNCTIONAL PROGRAMMING

In previous chapters we learned about Functions and Recursion.

In this chapter we are going to learn about more Functional Programming (FP) concepts like

- Pure Functions
- First-class functions
- Higher-order functions
- Anonymous and nested functions.
- Equality of functions

41.1 Pure Functions

We can create pure functions (functions that doesn't change the state) by the help of the assignment operator to copy variables (Lists & Objects) by value to create new variables instead of working on the original data that are passed to the function by reference.

Example:

```
Func Main
    aList = [1,2,3,4,5]
    aList2 = square(aList)
    see "aList" + nl
    see aList
    see "aList2" + nl
    see aList2

Func Square aPara
    a1 = aPara                # copy the list
    for x in a1
        x *= x
    next
    return a1                 # return new list
```

Output:

```
aList
1
2
3
4
5
```

```
aList2
1
4
9
16
25
```

41.2 First-class Functions

Functions inside the Ring programming language are first-class citizens, you can pass functions as parameters, return them as value or store them in variables.

We can pass/return the function by typing the function name as literal like “FunctionName” or :FunctionName for example.

We can pass/return functions using the variable that contains the function name.

We can call function from variables contains the function name using the Call command

Syntax:

```
Call Variable([Parameters])
```

Example:

```
Func Main
    see "before test2()" + nl
    f = Test2(:Test)
    see "after test2()" + nl
    call f()

Func Test
    see "Message from test!" + nl

Func Test2 f1
    call f1()
    See "Message from test2!" + nl
    return f1
```

Output:

```
before test2()
Message from test!
Message from test2!
after test2()
Message from test!
```

41.3 Higher-order Functions

Higher-order functions are the functions that takes other functions as parameters.

Example:

```
Func Main
    times(5,:test)
```

```

Func Test
    see "Message from the test function!" + nl

Func Times nCount,F
    for x = 1 to nCount
        Call F()
    next

```

Output:

```

Message from the test function!
Message from the test function!
Message from the test function!
Message from the test function!
Message from the test function!

```

41.4 Anonymous and Nested Functions

Anonymous Functions are functions without names that can be passed as parameters to other functions or stored in variables.

Syntax:

```

Func [Parameters] { [statements] }

```

Example:

```

test( func x,y {
    see "hello" + nl
    see "Sum : " + (x+y) + nl
} )

new great { f1() }

times(3, func { see "hello world" + nl } )

func test x
    call x(3,3)
    see "wow!" + nl

func times n,x
    for t=1 to n
        call x()
    next

Class great
    func f1
        f2( func { see "Message from f1" + nl } )

        func f2 x
            call x()

```

Output:

```

hello
Sum : 6

```

```
wow!
Message from f1
hello world
hello world
hello world
```

Example:

```
Func Main
  aList = [1,2,3,4]
  Map (aList , func x {
                                return x*x
                              } )

  see aList
  aList = [4,9,14,25]
  Map(aList, :myfilter )
  see aList
  aList = [11,12,13,14]
  Map (aList , func x {
    if x%2=0
      return "even"
    else
      return "odd"
    ok
  })
  see aList

Func myfilter x
  if x = 9
    return "True"
  else
    return "False"
  ok

Func Map aList,cFunc
  for x in aList
    x = call cFunc(x)
  next
```

Output:

```
1
4
9
16
False
True
False
False
odd
even
odd
even
```

41.5 Equality of functions

We can test if function = function or not using the '=' or '!=' operators

Example:

```
f1 = func { see "hello" + nl }  
  
f2 = func { see "how are you?" + nl }  
  
f3 = f1  
  
call f1()  
call f2()  
call f3()  
  
see (f1 = f2) + nl  
see (f2 = f3) + nl  
see (f1 = f3) + nl
```

Output:

```
hello  
how are you?  
hello  
0  
0  
1
```


REFLECTION AND META-PROGRAMMING

Since the Ring programming language is a dynamic language, we can get answers about the program code and we can modify our code during the runtime.

In this chapter we will learn about this and the available functions to use.

- `locals()`
- `globals()`
- `functions()`
- `cfunctions()`
- `islocal()`
- `isglobal()`
- `isfunction()`
- `iscfunction()`
- `packages()`
- `ispackage()`
- `classes()`
- `isclass()`
- `packageclasses()`
- `ispackageclass()`
- `classname()`
- `objectid()`
- `isobject()`
- `attributes()`
- `methods()`
- `isattribute()`
- `isprivateattribute()`
- `ismethod()`
- `isprivatemethod()`
- `addattribute()`
- `addmethod()`

- `getattribute()`
- `setattribute()`
- `mergemethods()`
- `packagename()`

42.1 locals() Function

We can get a list of variables names in the current scope using the `locals()` function.

Syntax:

```
locals() --> a list contains the variables names in the current scope
```

Example:

```
test("hello")

func test cMsg

    see cMsg + nl

    x = 10
    y = 20
    z = 30

    see locals()
```

Output:

```
hello
cmsg
x
y
z
```

42.2 globals() Function

We can get a list of variables names in the global scope using the `globals()` function.

Syntax:

```
globals() --> a list contains variables names in the global scope
```

Example:

```
x=10 y=20 z=30
test()

func test
    see "message from test()" + nl +
        "Global Variables:" + nl
    see globals()
```

Output:

```
message from test()
Global Variables:
x
y
z
```

42.3 functions() Function

We can get a list of functions names written in the Ring language using the functions() function.

Syntax:

```
functions() --> a list contains functions names
```

Example:

```
see functions()

func f1
    see "f1" + nl

func f2
    see "f2" + nl

func f3
    see "f3" + nl
```

Output:

```
f1
f2
f3
```

42.4 cfunctions() Function

We can get a list of functions names written in the C language using the cfunctions() function.

Syntax:

```
cfunctions() --> a list contains functions names
```

Example:

```
aList = cfunctions()
See "Count : " + len(aList) + nl
for x in aList
    see x + "()" + nl
next
```

Output:

```
Count : 197
len()
add()
del()
get()
```

```
clock()
...
```

Note: The complete list is removed from the previous output.

42.5 islocal() Function

We can check if a variable is defined in the local scope or not using the `islocal()` function.

Syntax:

```
islocal(cVariableName) --> returns 1 if the variable is defined in the local scope
                           returns 0 if the variable is not defined in the local scope
```

Example:

```
test()

func test
    x=10 y=20
    see islocal("x") + nl +
        islocal("y") + nl +
        islocal("z") + nl
```

Output:

```
1
1
0
```

42.6 isglobal() Function

We can check if a variable is defined in the global scope or not using the `isglobal()` function.

Syntax:

```
isglobal(cVariableName) --> returns 1 if the variable is defined in the global scope
                           returns 0 if the variable is not defined in the global scope
```

Example:

```
x=10 y=20

test()

func test
    see isglobal("x") + nl +
        isglobal("y") + nl +
        isglobal("z") + nl
```

Output:

```
1
1
0
```

42.7 isfunction() Function

We can check if a Ring function is defined or not using the `isFunction()` function.

Syntax:

```
isFunction(cFunctionName) --> returns 1 if the Ring function is defined
                             returns 0 if the Ring function is not defined
```

Example:

```
see isfunction("f1") + nl +
      isfunction("f2") + nl +
      isfunction("f3") + nl

func f1
      see "message from f1()" + nl

func f2
      see "message from f2()" + nl
```

Output:

```
1
1
0
```

42.8 iscfunfunction() Function

We can check if a C function is defined or not using the `iscfunfunction()` function.

Syntax:

```
iscfunfunction(cFunctionName) --> returns 1 if the C function is defined
                                   returns 0 if the C function is not defined
```

Example:

```
see iscfunfunction("len") + nl +
      iscfunfunction("add") + nl +
      iscfunfunction("test") + nl
```

Output:

```
1
1
0
```

42.9 packages() Function

We can get a list of packages names using the `packages()` function.

Syntax:

```
packages() --> a list contains packages names
```

Example:

```
See packages()

Package Package1
    Class class1
        Func f1

Package Package2
    Class class1
        Func f1

Package Package3
    Class class1
        Func f1

Package Package4
    Class class1
        Func f1
```

Output:

```
package1
package2
package3
package4
```

42.10 ispackage() Function

We can check if a package is defined or not using the `ispackage()` function.

Syntax:

```
ispackage(cPackageName) --> returns 1 if the Package is defined
                           returns 0 if the Package is not defined
```

Example:

```
See ispackage("package1") + nl +
    ispackage("package4") + nl +
    ispackage("package5") + nl +
    ispackage("package3") + nl

Package Package1
    Class class1
        Func f1

Package Package2
    Class class1
        Func f1

Package Package3
    Class class1
        Func f1

Package Package4
    Class class1
        Func f1
```

Output:

```
1
1
0
1
```

42.11 classes() Function

We can get a list of classes names using the classes() function.

Syntax:

```
classes() --> a list contains classes names
```

Example:

```
See classes()

Class class1
    Func f1

Class class2
    Func f1

Class class3
    Func f1
```

Output:

```
class1
class2
class3
```

42.12 isclass() Function

We can check if a class is defined or not using the isclass() function.

Syntax:

```
isclass(cClassName) --> returns 1 if the Class is defined
                        returns 0 if the Class is not defined
```

Example:

```
see isclass("class4") + nl +
    isclass("class3") + nl +
    isclass("class2") + nl

Class class1
    func f1

class class2
    func f1

class class3
    func f1
```

Output:

```
0
1
1
```

42.13 packageclasses() Function

We can get a list of classes names inside a package using the packageclasses() function.

Syntax:

```
packageclasses(cPackageName) --> a list contains classes names inside the package
```

Example:

```
see "classes in Package1" + nl
see packageclasses("Package1")
see "classes in Package2" + nl
see packageclasses("Package2")
```

```
Package Package1
  Class class1
    Func f1
```

```
Package Package2
  Class class1
    Func f1
  Class class2
    Func f1
  Class class3
    func f1
```

Output:

```
classes in Package1
class1
classes in Package2
class1
class2
class3
```

42.14 ispackageclass() Function

We can check if a class is defined inside package or not using the ispackageclass() function.

Syntax:

```
ispackageclass(cPackageName,cClassName) --> returns 1 if the Class is defined
                                             returns 0 if the Class is not defined
```

Example:

```
see ispackageclass("package1","class1") + nl +
    ispackageclass("package1","class2") + nl +
    ispackageclass("package2","class1") + nl +
```



```
ispackageclass("package2","class2") + nl

Package Package1
  Class class1
    Func f1

Package Package2
  Class class1
    Func f1
  Class class2
    Func f1
  Class class3
    func f1
```

Output:

```
1
0
1
1
```

42.15 classname() Function

We can know the class name of an object using the classname() function

Syntax:

```
classname(object) --> Returns the object class name
```

Example:

```
o1 = new point
o2 = new rect

see classname(o1) + nl      # print point
see classname(o2) + nl      # print rect

class point
class rect
```

42.16 objectid() Function

We can know the object id using the objectid() function

Syntax:

```
objectid(object) --> Returns the object id
```

Example:

```
o1 = new point
see objectid(o1) + nl
test(o1)

func test v
  see objectid(v) + nl
```

```
Class point x y z
```

Output:

```
021B5808
021B5808
```

42.17 isobject() Function

We can check the variable to know if it's an object or not using the isobject() function

Syntax:

```
isobject(variable) --> Returns True if it's an object, False if it's not
```

42.18 attributes() Function

We can get the object attributes using the attributes() function

Syntax:

```
attributes(object) --> Returns a list contains the object attributes
```

Example:

```
o1 = new point
aList = attributes(o1)           # we can use see attributes(o1)
for t in aList see t next       # print xyz
Class Point x y z
```

42.19 methods() Function

We can get the object methods using the methods() function

Syntax:

```
methods(object) --> Returns a list contains the object methods
```

Example:

```
o1 = new test
aList = methods(o1)

for x in aList
    cCode = "o1."+x+"()"
    eval(cCode)
next

Class Test
    func f1
        see "hello from f1" + nl
    func f2
        see "hello from f2" + nl
```

```
func f3
    see "hello from f3" + nl
func f4
    see "hello from f4" + nl
```

Output:

```
hello from f1
hello from f2
hello from f3
hello from f4
```

42.20 isattribute() Function

We can test if the object contains an attribute or not using the `isattribute()` function

Syntax:

```
isattribute(object, cAttributeName) --> Returns True if the object contains the attribute
```

Example:

```
o1 = new point

see isattribute(o1, "x") + nl    # print 1
see isattribute(o1, "t") + nl    # print 0
see isattribute(o1, "y") + nl    # print 1
see isattribute(o1, "z") + nl    # print 1

class point x y z
```

42.21 isprivateattribute() Function

We can test if the object contains a private attribute or not using the `isprivateattribute()` function

Syntax:

```
isprivateattribute(object, cAttributeName) --> Returns True if the object
contains the private attribute
```

Example:

```
o1 = new person

see isprivateattribute(o1, "name") + nl +
    isprivateattribute(o1, "address") + nl +
    isprivateattribute(o1, "phone") + nl +
    isprivateattribute(o1, "job") + nl +
    isprivateattribute(o1, "salary")

Class Person
    name address phone
    private
        job salary
```

Output:

```
0
0
0
1
1
```

42.22 ismethod() Function

We can test if the object class contains a method or not using the ismethod() function

Syntax:

```
ismethod(object,cMethodName) --> Returns True if the object class contains the method
```

Example:

```
o1 = new point

see ismethod(o1,"print") + nl      # print 1

mylist = []
mylist + new point

see ismethod(mylist[1],"print") + nl  # print 1

class point x y z
  func print
    see x + nl + y + nl + z + nl
```

42.23 isprivatemethod() Function

We can test if the object class contains a private method or not using the isprivatemethod() function

Syntax:

```
isprivatemethod(object,cMethodName) --> Returns True if the object class contains
the private method
```

Example:

```
o1 = new Test

see isprivatemethod(o1,"f1") + nl +
    isprivatemethod(o1,"f2")

Class Test
  func f1
    see "message from f1()" + nl
  private
    func f2
      see "message from f2()" + nl
```

Output:

```
0
1
```

42.24 addattribute() Function

We can add an attribute (or a group of attributes) to the object state (not the class) using the addattribute() function

Syntax:

```
AddAttribute(object, cAttributeName|aAttributesList)
```

Example(1):

```
see new point {x=10 y=20 z=30}
Class Point
    AddAttribute(self, ["x", "y", "z"])
```

Example(2):

```
o1 = new point
addattribute(o1, "x")
addattribute(o1, "y")
addattribute(o1, "z")
see o1 {x=10 y=20 z=30}
class point
```

Output:

```
x: 10.000000
y: 20.000000
z: 30.000000
```

42.25 addmethod() Function

We can add a method to the object class using the addmethod() function This method can be used with any object from the same class.

Syntax:

```
AddMethod(Object, cNewMethodName, cMethodName|AnonymousFunction)
```

Example:

```
o1 = new point { x=10 y=20 z=30 }

addmethod(o1, "print", func { see x + nl + y + nl + z + nl } )

o1.print()

Class point
    x y z
```

Output:

```
10
20
30
```

Instead of using anonymous function to add new method to the class, we can use the function name

Example:

```
o1 = new point { x=10 y=20 z=30 }

myfunc = func { see x + nl + y + nl + z + nl }

addmethod(o1,"print", myfunc )
addmethod(o1,"display", myfunc )
addmethod(o1,"show", myfunc )

o1.print()
o1.display()
o1.show()

Class point
  x y z
```

Output:

```
10
20
30
10
20
30
10
20
30
```

Since we add the method to the class, any object from that class can use this method

Example:

```
o1 = new point { x=10 y=20 z=30 }
o2 = new point { x=100 y=200 z=300 }
o3 = new point { x=50 y=150 z=250 }

addmethod(o1,"print", func { see x + nl + y + nl + z + nl } )

o1.print()
o2.print()
o3.print()

Class point
  x y z
```

Output:

```
10
20
30
100
200
300
```

```
50
150
250
```

42.26 getattribute() function

We can get the object attribute value using the getattribute() function

Syntax:

```
GetAttribute(oObject,cAttributeName) ---> Attribute Value
```

Example:

```
o1 = new point

see getattribute(o1,"name") + nl +
    getattribute(o1,"x") + nl +
    getattribute(o1,"y") + nl +
    getattribute(o1,"z") + nl

Class Point
    x=10 y=20 z=30
    name = "3D-Point"
```

Output:

```
3D-Point
10
20
30
```

Example:

We can Find a Class List Member using GetAttribute() using a function findclass() The Find uses the member name, rather than the column number

```
myList =
    [new Company {position=3 name="Mahmoud" symbol="MHD"},
    new Company {position=2 name="Bert" symbol="BRT"},
    new Company {position=1 name="Ring" symbol="RNG"}
    ]

see myList
see nl + "===== " + nl + nl

for i = 1 to len(myList)
    see "Pos: " + i + " | " + myList[i].position + " | " + myList[i].name +
        " | " + myList[i].symbol + " | " + nl
next

See findclass(myList, "MHD", "symbol") +nl    ### Specify Member class name

###-----

func findclass classList, cValue, classMember
```

```

    See nl + "FindClass: " + " " + cValue + nl + nl

    for i = 1 to len(classList)
        result = getattribute( classList[i], classMember )

        See "Result-Attr: " + i + " " + result +nl
        if result = cValue
            j = i
            ok
        next
    return j

###-----

class company position name symbol

```

Output:

```

Pos: 1 | 3 | Mahmoud | MHD |
Pos: 2 | 2 | Bert | BRT |
Pos: 3 | 1 | Ring | RNG |

FindClass:  MHD

Result-Attr: 1 MHD
Result-Attr: 2 BRT
Result-Attr: 3 RNG

1

```

42.27 setattribute() function

We can set the object attribute value using the setattribute() function

Syntax:

```
SetAttribute(oObject,cAttributeName,Value)
```

Example:

```

o1 = new person
setattribute(o1,"cName","Mahmoud")
setattribute(o1,"nSalary",1000000)
setattribute(o1,"aColors",["white","blue","yellow"])

see o1
see o1.aColors

Class Person
    cName
    nSalary
    aColors

```

Output:

```

cname: Mahmoud
nsalary: 1000000.000000

```



```
acolors: List...
white
blue
yellow
```

42.28 mergemethods() Function

We can share methods between classes without inheritance using the MergeMethods() function

This function merge class methods to another class.

Syntax:

```
MergeMethods(cClassNameDestination,cClassNameSource)
```

Example:

```
mergemethods("count","share")
mergemethods("count2","share")

o1 = new count { test() }
o1 = new count2 { test() }

Class Share
    func one
        see "one" + nl
    func two
        see "two" + nl
    func three
        see "three" + nl

Class Display
    func printline
        see copy(" ",20) + nl

Class Count from Display
    func test
        printline()
        one()
        two()
        three()
        printline()

Class Count2 from Display
    func test
        three()
        two()
        one()
        printline()
```

Output:

```
*****
one
two
three
*****
```

```
three
two
one
*****
```

42.29 packagename() Function

We can know the package name of the latest successful import command using the `packagename()` function

Syntax:

```
packagename() --> Returns the package name of the latest successful import
```

Example:

```
load "weblib.ring"
import System.web
see packagename()      # system.web
```

STDLIB FUNCTIONS

In this chapter we are going to learn about functions in the `stdlib.ring`
Before using the functions in the library, We must load the library first

```
load "stdlib.ring"
```

Instead of using `stdlib.ring` we can use `stdlibcore.ring`

Using `stdlibcore.ring` we can use the `StdLib` functions (Without Classes)

This is useful when developing standalone console applications

Because using `stdlib.ring` (functions & classes) will load libraries like `RingLibCurl`, `RingOpenSSL`, etc.

43.1 Puts() function

print the value then print new line (nl)

Syntax:

```
puts (expr)
```

Example:

```
Load "stdlib.ring"
```

```
Puts("Hello, World!")
```

43.2 Print() function

print string - support `\n`, `\t` and `\r`

Also we can use `#{variable_name}` to insert variables values.

Syntax:

```
print (string) ---> String
```

Example:

```
print("\nHello, World\n\nHow are you? \t\t I'm fine!\n")
x=10 y=20
print("\nx value = #{x} , y value = #{y} \n")
```

43.3 Print2Str() Function

Syntax:

```
print2Str(string) ---> String
```

Example:

```
world = "World!"
mystring = print2str("Hello, #{world} \nIn Year \n#{2000+17} \n")

see mystring + nl
```

Output:

```
Hello, World!
In Year
2017
```

43.4 GetString() function

Get input from the keyboard - return value as string

```
getString() ---> string
```

43.5 GetNumber() function

Get input from the keyboard - return value as number

```
getnumber() ---> number
```

43.6 AppPath() function

Get the path of the application folder

Syntax:

```
AppPath() ---> The path as String
```

Example:

```
Load "stdlib.ring"

# Application Path
Puts("Test AppPath() ")
See AppPath() + nl
```

43.7 JustFilePath() function

Get the path of the file, remove the file name.

Syntax:

```
JustFilePath(cFile) ---> The path as String
```

Example:

```
load "stdlib.ring"
see justfilePath("b:\\ring\\applications\\rnote\\rnote.ring")
```

Output:

```
b:\\ring\\applications\\rnote\\
```

43.8 JustFileName() function

Get the file, remove the file path.

Syntax:

```
JustFileName(cFile) ---> The file name as String
```

Example:

```
load "stdlib.ring"
see justfileName("b:\\ring\\applications\\rnote\\rnote.ring")
```

Output:

```
rnote.ring
```

43.9 Value() function

create a copy from a list or object

Syntax:

```
value(List) ---> new list
```

Example:

```
Load "stdlib.ring"
aList = 1:10
del(value(aList),1) # delete first item
see aList          # print numbers from 1 to 10
```

43.10 Times() function

Execute a Function nCount times

Syntax:

```
Times(nCount,function)
```

Example:

```
Load "stdlib.ring"

Puts("Test Times()")
Times ( 3 , func { see "Hello, World!" + nl } )
```

43.11 Map() function

Execute a Function on each list item

Syntax:

```
Map(alist,function) ---> List
```

Example:

```
Load "stdlib.ring"

Puts("Test Map()")
See Map( 1:10, func x { return x*x } )
```

43.12 Filter() function

Execute a Function on each list item to filter items

Syntax:

```
Filter(alist,function) ---> List
```

Example:

```
Load "stdlib.ring"

Puts("Test Filter()")
See Filter( 1:10 , func x { if x <= 5 return true else return false ok } )
```

43.13 Split() function

Convert string words to list items

Syntax:

```
Split(cstring,delimiter) ---> List
```

Example:

```
Load "stdlib.ring"

Puts("Test Split()")
See Split("one two three four five", " ")
```

43.14 SplitMany() function

Convert string words to list items. Allow many delimiters.

Syntax:

```
SplitMany(cstring, delimiters as string or list) --> List
```

Example:

```
Load "stdlib.ring"

Puts("Test SplitMany() ")
See SplitMany("one,two,three,four and five", " ,")
```

43.15 NewList() function

Create a two dimensional list

Syntax:

```
NewList(nRows,nColumns) ---> new list
```

Example:

```
Load "stdlib.ring"

Puts("Test Newlist() ")
a1 = 3
a2 = 5
chrArray = newlist(a1,a2)
numArray = newlist(a1,a2)
chrArray[1][1] = "Hello"
numArray[1][1] = 987.2
See chrArray[1][1] + nl
See numArray[1][1] + nl
```

43.16 Capitalized() function

Return a copy of a string with the first letter capitalized

Syntax:

```
Capitalized(string) ---> string
```

Example:

```
Load "stdlib.ring"

Puts("Test Capitalized() ")
See capitalized("welcome to the Ring Programming Language")
```

43.17 IsSpecial() function

Check whether a character is special or not

Syntax:

```
IsSpecial(char) ---> True/False
```

Example:

```
Load "stdlib.ring"

Puts("Test Isspecial() ")
See "Isspecial = " + isSpecial("%") + nl
```

43.18 IsVowel() function

Check whether a character is vowel or not

Syntax:

```
IsVowel(char) ---> True/False
```

Example:

```
Load "stdlib.ring"

Puts("Test Isvowel() ")
See "Isvowel = " + isVowel("c") + nl
```

43.19 LineCount() function

Return the lines count in a text file.

Syntax:

```
LineCount(cFileName) ---> Lines Count as number
```

Example:

```
Load "stdlib.ring"

Puts("Test Linecount() ")
See "the number of lines = " + lineCount("test.ring")
```

43.20 Factorial() function

Return the factorial of a number

Syntax:

```
Factorial(number) ---> number
```

Example:


```
Load "stdlib.ring"

Puts("Test Factorial() ")
see "6 factorial is : " + Factorial(6)
```

43.21 Fibonacci() function

Return the fibonacci number

Syntax:

```
Fibonacci(number) ---> number
```

Example:

```
Load "stdlib.ring"

Puts("Test Fibonacci() ")
see "6 Fibonacci is : " + Fibonacci(6)
```

43.22 IsPrime() function

Check whether a number is prime or not

Syntax:

```
isprime(number) ---> Number
```

Example:

```
Load "stdlib.ring"

Puts("Test Isprime() ")
if isPrime(16) see "16 is a prime number"
else see "16 is not a prime number" ok
```

43.23 Sign() function

Returns an integer value indicating the sign of a number.

Syntax:

```
Sign(number) ----> number ( -1 = negative , 0 , 1 (positive) )
```

Example:

```
Load "stdlib.ring"

Puts("Test Sign() ")
see "sign of 12 is = " + sign(12) + nl
```

43.24 List2File() function

Write list items to text file (each item in new line).

Syntax:

```
List2File(aList,cFileName)
```

Example:

```
Load "stdlib.ring"

# Test List2File
Puts("Test List2File() ")
list2file(1:100,"myfile.txt")
```

43.25 File2List() function

Read text file and convert lines to list items

Syntax:

```
File2List(cFileName) ---> List
```

Example:

```
Load "stdlib.ring"

# Test File2List
Puts("Test File2List() ")
see len(file2list("myfile.txt"))
```

43.26 StartsWith() function

Returns true if the given string starts with the specified substring.

Leading white spaces are ignored.

Syntax:

```
StartsWith(string, substring) ---> True/False
```

Example:

```
Load "stdlib.ring"

Puts("Test Startswith() ")
see Startswith("CalmoSoft", "Calmo") + nl
```

43.27 EndsWith() function

Returns true if the given string ends with the specified substring.

Trailing white spaces are ignored.

Syntax:

```
Endswith(string, substring) ---> True/False
```

Example:

```
Load "stdlib.ring"

Puts("Test Endswith()")
see endswith("CalmoSoft", "Soft") + nl
```

43.28 GCD() function

Finding of the greatest common divisor of two integers.

Syntax:

```
Gcd(number, number) ---> number
```

Example:

```
Load "stdlib.ring"

Puts("Test Gcd()")
see gcd (24, 32) + nl
```

43.29 LCM() function

Compute the least common multiple of two integers.

Syntax:

```
lcm(number, number) ---> number
```

Example:

```
Load "stdlib.ring"

Puts("Test Lcm()")
see Lcm(24, 36) + nl
```

43.30 SumList() function

Compute the sum of a list of integers.

Syntax:

```
sumlist(list) ---> number
```

Example:

```
Load "stdlib.ring"

Puts("Test Sumlist()")
```

```
aList = [1,2,3,4,5]
see Sumlist(aList) + nl
```

43.31 ProdList() function

Compute the product of a list of integers.

Syntax:

```
prodlist(list) ---> number
```

Example:

```
Load "stdlib.ring"

Puts("Test Prodlist() ")
aList = [1,2,3,4,5]
see Prodlist(aList) + nl
```

43.32 EvenOrOdd() function

Test whether an integer is even or odd.

Result of test (1=odd 2=even).

Syntax:

```
evenorodd(number) ---> 1 (odd) or 2 (even)
```

Example:

```
Load "stdlib.ring"

Puts("Test Evenorodd() ")
nr = 17
see Evenorodd(nr) + nl
```

43.33 Factors() function

Compute the factors of a positive integer.

Syntax:

```
factors(number) ---> list
```

Example:

```
Load "stdlib.ring"

Puts("Test Factors() ")
n = 45
aList = factors(n)
see "Factors of " + n + " = "
for i = 1 to len(aList)
```

```
    see "" + aList[i] + " "
next
```

43.34 Palindrome() function

Check if a sequence of characters is a palindrome or not.

Syntax:

```
Palindrome(String) ---> True/False
```

Example:

```
Load "stdlib.ring"

Puts("Test Palindrome() ")
cString = "radar"
see Palindrome(cString)
```

43.35 IsLeapYear() function

Check whether a given year is a leap year in the Gregorian calendar.

Syntax:

```
Isleapyear(number) ---> True/False
```

Example:

```
Load "stdlib.ring"

Puts("Test Isleapyear() ")
year = 2016
if Isleapyear(year) see "" + year + " is a leap year."
else see "" + year + " is not a leap year." ok
```

43.36 BinaryDigits() function

Compute the sequence of binary digits for a given non-negative integer.

Syntax:

```
binarydigits(number) ---> string
```

Example:

```
Load "stdlib.ring"

Puts("Test Binarydigits() ")
b = 35
see "Binary digits of " + b + " = " + Binarydigits(b)
```

43.37 MatrixMulti() function

Multiply two matrices together.

Syntax:

```
Matrixmulti(List,List) ---> List
```

Example:

```
Load "stdlib.ring"

# Multiply two matrices together.
Puts("Test Matrixmulti()")
A = [[1,2,3], [4,5,6], [7,8,9]]
B = [[1,0,0], [0,1,0], [0,0,1]]
see Matrixmulti(A, B)
```

43.38 MatrixTrans() function

Transpose an arbitrarily sized rectangular Matrix.

Syntax:

```
Matrixtrans(List) ---> List
```

Example:

```
Load "stdlib.ring"

# Transpose an arbitrarily sized rectangular Matrix.
Puts("Test Matrixtrans()")
matrix = [[78,19,30,12,36], [49,10,65,42,50], [30,93,24,78,10], [39,68,27,64,29]]
see Matrixtrans(matrix)
```

43.39 DayOfWeek() function

Return the day of the week of given date. (yyyy-mm-dd)

Syntax:

```
dayofweek(string) ---> string
```

Example:

```
Load "stdlib.ring"

# Return the day of the week of given date.
Puts("Test Dayofweek()")
date = "2016-04-24"
see "Data : " + date + " - Day : " + Dayofweek(date) + nl
```

43.40 Permutation() function

Generates all permutations of n different numerals.

Syntax:

```
permutation(list)
```

Example:

```
Load "stdlib.ring"

# Generates all permutations of n different numerals
Puts("Test Permutation()")
list = [1, 2, 3, 4]
for perm = 1 to 24
    for i = 1 to len(list)
        see list[i] + " "
    next
    see nl
    Permutation(list)
next
```

43.41 ReadLine() function

Read line from file

Syntax:

```
readline(fp) ---> string
```

Example:

```
Load "stdlib.ring"

# Read a file line by line.
Puts("Test Readline()")
fp = fopen("test.ring", "r")
while not feof(fp)
    See Readline(fp) end
fclose(fp)
```

43.42 SubString() function

Return a position of a substring starting from a given position in a string.

Syntax:

```
Substring(str, substr, npos) ---> string
```

Example:

```
Load "stdlib.ring"

# Return a position of a substring starting from a given position in a string.
```

```
Puts("Test Substring() ")
a = "abcxyzqweabc"
b = "abc"
i = 4
see substring(a,b,i)
```

43.43 ChangeString() function

Change substring from given position to a given position with another substring.

Syntax:

```
Changestring(cString, nPos1, nPos2, cSubstr) ---> cString
```

Example:

```
Load "stdlib.ring"

# Change substring from given position for given position with a substring.
Puts("Test Changestring() ")
see Changestring("Rmasdg",2,5,"in")      # Ring
```

43.44 Sleep() function

Sleep for the given amount of time.

Syntax:

```
sleep(nSeconds)
```

Example:

```
Load "stdlib.ring"

Puts("Test Sleep() ")
see "Wait 3 Seconds!"
Sleep(3)
see nl
```

43.45 IsMainSourceFile() function

Check if the current file is the main source file

Syntax:

```
IsMainSourceFile() ---> True/False
```

Example:

```
Load "stdlib.ring"

if ismainsourcefile()
    # code
ok
```


43.46 DirExists() function

Check if directory exists

Syntax:

```
DirExists(String) ---> True/False
```

Example:

```
Load "stdlib.ring"

see "Check dir : b:\ring "
puts( DirExists("b:\ring") )
see "Check dir : C:\ring "
puts( DirExists("C:\ring") )
```

43.47 MakeDir() function

Make Directory

Syntax:

```
MakeDir(String)
```

Example:

```
Load "stdlib.ring"

# Create Directory
puts("create Directory : myfolder")
mkdir("myfolder")
```

43.48 Fsize() function

The function return the file size in bytes.

Syntax:

```
FSize(File Handle) ---> Number (File Size in Bytes)
```

43.49 TrimAll() function

Remove all spaces and tabs characters from a string

Syntax:

```
TrimAll(cString) ---> cString # Without Spaces and Tabs
```

43.50 TrimLeft() function

Remove all spaces and tabs characters from the left side of a string

Syntax:

```
TrimLeft(cString) ---> cString # Without Spaces and Tabs from the left side
```

43.51 TrimRight() function

Remove all spaces and tabs characters from the right side of a string

Syntax:

```
TrimRight(cString) ---> cString # Without Spaces and Tabs from the right side
```

43.52 EpochTime() function

Return the Epoch Time

Syntax:

```
EpochTime(cDate,cTime) ---> nEpochTime
```

Example:

```
see EpochTime( Date(), Time() )
```

43.53 SystemCmd() Function

We can execute system commands using the SystemCmd() function that outputs to a variable

Syntax:

```
SystemCmd(cCommand)
```

Example:

```
cYou = SystemCmd("whoami")      # User Name logged in is output a variable
cThem = SystemCmd("dir c:\Users") # Directory List is output to a variable
```

43.54 ListAllFiles() Function

Using this function we can quickly do a process on a group of files in a folder and it's sub folders.

Syntax:

```
ListAllFiles(cFolder,cExtension) ---> List of Files
```

Example:

```
aList = ListAllFiles("c:/ring/ringlibs","ring") # *.ring only
aList = sort(aList)
see aList
```

Example:

```
see listallfiles("b:/ring/ringlibs/weblib","") # All Files
```

43.55 SystemSilent() Function

We can execute system commands using the SystemSilent() function to avoid displaying the output!

Syntax:

```
SystemSilent(cCommand)
```

43.56 OSCreateOpenFolder() Function

Create folder then change the current folder to this new folder

Syntax:

```
OSCreateOpenFolder(cCommand)
```

43.57 OSCopyFolder() Function

Copy folder to the current folder

Parameters : The path to the parent folder and the folder name to copy

Syntax:

```
OSCopyFolder(cParentFolder,cFolderName)
```

Example

To copy the folder b:ringringlibsstdlib to the current folder

```
OSCopyFolder("b:\ring\ringlibs\","stdlib")
```

43.58 OSDeleteFolder() Function

Delete Folder in the current Directory

Syntax:

```
OSDeleteFolder(cFolderName)
```

43.59 OSCopyFile() Function

Copy File to the current directory

Syntax:

```
OSCopyFile(cFileName)
```

43.60 OSDeleteFile() Function

Delete File

Syntax:

```
OSDeleteFile(cFileName)
```

43.61 OSRenameFile() Function

Rename File

Syntax:

```
OSRenameFile(cOldFileName, cNewFileName)
```

STDLIB CLASSES

In this chapter we are going to learn about the classes in the stdlib.ring

- StdBase Class
- String Class
- List Class
- Stack Class
- Queue Class
- HashTable Class
- Tree Class
- Math Class
- DateTime Class
- File Class
- System Class
- Debug Class
- DataType Class
- Conversion Class
- ODBC Class
- MySQL Class
- SQLite Class
- Security Class
- Internet Class

44.1 StdBase Class

Attributes:

- vValue : Object Value

Methods:

Method	Description/Output
Init(x)	Set vValue Attribute to x value
Print()	Print vValue
PrintLn()	Print vValue then New Line
Size()	return number represent the size of vValue
Value()	return vValue
Set(x)	Call Init(x)

44.2 String Class

Parent Class : StdBase Class

Methods:

Method	Description/Output
Init(String Number List)	
Lower()	New String - Lower case characters
Upper()	New String - Upper case characters
Left(x)	New String - contains x characters from the left
Right(x)	New String - contains x characters from the right
Lines()	Number - Lines count
Trim()	New String - Remove Spaces
Copy(x)	New String - repeat string x times
strcmp(cString)	Compare string with cString
tolist()	List (String Lines to String Items)
tofile(cFileName)	Write string to file
mid(nPos1,nPos2)	New String - from nPos1 to nPos2
getfrom(nPos1)	New String - from nPos1 to the end of the string
replace(cStr1,cStr2,lCase)	New String - Replace cStr1 with cStr2 , lCase (True=Match Case)
split()	List - Each Word as list item
startswith(substring)	Return true if the start starts with a substring
endswith(substring)	Return true if the start ends with a substring

Example:

```
Load "stdlib.ring"

See "Testing the String Class" + nl
oString = new string("Hello, World!")
oString.println()
oString.upper().println()
oString.lower().println()
oString.left(5).println()
oString.right(6).println()
oString = new string("Hi" + nl + "Hello" )
See oString.lines() + nl
oString = new string("    Welcome    ")
oString.println()
oString.trim().println()
oString = new string("Hello! ")
oString.copy(3).println()
see oString strcmp("Hello! ") + nl
see oString strcmp("Hello ") + nl
see oString strcmp("Hello!! ") + nl
oString = new string(["one", "two", "three"])
```

```

oString.print()
see oString.lines() + nl
oString = new String(1234)
oString.println()
oString = new String("one"+nl+"two"+nl+"three")
aList = oString.toList()
see "List Items" + nl See aList
oString = new String( "Welcome to the Ring programming language")
See "the - position : " + oString.pos("the") + nl
oString = oString.getfrom(oString.pos("Ring"))
oString.println()
oString.mid(1,4).println()
oString = oString.replace("Ring","***Ring***",true)
oString.println()
oString = oString.replace("ring","***Ring***",false)
oString.println()
oString1 = new string("First")
oString2 = new string("Second")
oString = oString1 + oString2
oString.println()
oString = oString1 * 3
oString.println()
for t in ostring see t next
oString.toFile("test.txt")
oString = new string("one two three")
see nl
see ostring.split()
oString {
    set("Hello") println()
    set("How are you?") println()
}

```

Output:

```

Testing the String Class
Hello, World!
HELLO, WORLD!
hello, world!
Hello
World!
2
    Welcome
Welcome
Hello! Hello! Hello!
0
1
-1
one
two
three
4
1234
List Items
one
two
three
the - position : 12
Ring programming language

```

```

Ring
***Ring*** programming language
*****Ring***** programming language
FirstSecond
FirstFirstFirst
FirstFirstFirst
one
two
three
Hello
How are you?

```

44.3 List Class

Parent Class : StdBase Class

Methods:

Method	Description/Output
Init(String List)	
Add(Value)	Add item to the list
Delete(nIndex)	Delete item from the list
Item(nIndex)	Get item from the list
First()	Get the first item in the list
Last()	Get the last item in the list
Set(nIndex,Value)	Set item value
FindInColumn(nCol,Value)	Find item in a column
Sort()	Sort items - return new list
Reverse()	Reverse items - return new list
Insert(nIndex,Value)	Inset Item after nIndex

example:

```

Load "stdlib.ring"

oList = new list ( [1,2,3] )
oList.Add(4)
oList.print()
see oList.item(1) + nl
oList.delete(4)
oList.print()
see oList.first() + nl
see oList.last() + nl
oList { set(1,"one") set(2,"two") set(3,"three") print() }
see oList.find("two") + nl
oList.sort().print()
oList.reverse().print()
oList.insert(2,"nice")
oList.print()
oList = new list ( [ [1,"one"], [2,"two"], [3,"three"] ] )
see copy(" ",10) + nl
oList.print()
see "Search two : " + oList.findincolumn(2,"two") + nl
see "Search 1 : " + oList.findincolumn(1,1) + nl
oList = new list ( [ "Egypt" , "USA" , "KSA" ] )
for x in oList

```



```

        see x + nl
next
oList = new list ( [1,2,3,4] )
oList + [5,6,7]
oList.print()
oList = new list ( ["one","two"] )
oList2 = new list ( ["three","four"] )
oList + oList2
oList.print()

```

output:

```

1
2
3
4
1
1
2
3
1
3
one
two
three
2
one
three
two
three
two
one
one
two
nice
three
*****
1
one
2
two
3
three
Search two : 2
Search 1 : 1
Egypt
USA
KSA
1
2
3
4
5
6
7
one
two
three

```

```
four
```

44.4 Stack Class

Parent Class : List Class

Methods:

Method	Description/Output
Init(String Number List)	
Push(Value)	Push item to the stack
Pop()	Pop item from the stack
Print()	Print the stack items

example:

```
Load "stdlib.ring"

oStack = new Stack
oStack.push(1)
oStack.push(2)
oStack.push(3)
see oStack.pop() + nl
see oStack.pop() + nl
see oStack.pop() + nl
oStack.push(4)
see oStack.pop() + nl
oStack { push("one") push("two") push("three") }
oStack.print()
```

output:

```
3
2
1
4
three
two
one
```

44.5 Queue Class

Parent Class : List Class

Methods:

Method	Description/Output
Init(String Number List)	
Remove()	Remove item from the Queue.

example:

```
Load "stdlib.ring"

oQueue = new Queue
oQueue.add(1)
```

```
oQueue.add(2)
oQueue.add(3)
see oQueue.remove() + nl
see oQueue.remove() + nl
see oQueue.remove() + nl
oQueue.add(4)
see oQueue.remove() + nl
oQueue { add("one") add("two") add("three") }
oQueue.print()
```

output:

```
1
2
3
4
one
two
three
```

44.6 HashTable Class

Parent Class : List Class

Methods:

Method	Description/Output
Init(List)	
Add(cKey,Value)	Add item to the HashTable
Set(cKey,Value)	Set item value using the Key
GetValue(cKey)	Get item value using the Key
Contains(cKey)	Check if the HashTable contains item using the Key
Index(cKey)	Get the item index using the Key

example:

```
Load "stdlib.ring"

ohashtable = new hashtable
See "Test the hashtable Class Methods" + nl
ohashtable {
    Add("Egypt", "Cairo")
    Add("KSA", "Riyadh")
    see self["Egypt"] + nl
    see self["KSA"] + nl
    see contains("Egypt") + nl
    see contains("USA") + nl
    see index("KSA") + NL
    print()
    delete(index("KSA"))
    see copy(" ", 60) + nl
    print()
}
```

output:

```

Test the hashtable Class Methods
Cairo
Riyadh
1
0
2
Egypt
Cairo
KSA
Riyadh
*****
Egypt
Cairo

```

44.7 Tree Class

Data:

Attribute	Description
Data	Node Value
Children	Children List

Methods:

Method	Description/Output
set(value)	Set the node value.
value()	Get the node value.
Add(value)	Add new child.
parent()	Get the parent node.
print()	Print the tree nodes.

example:

```

Load "stdlib.ring"

otree = new tree
See "Test the tree Class Methods" + nl
otree {
    set("The first step")    # set the root node value
    see value() + nl
    Add("one")
    Add("two")
    Add("three") {
        Add("3.1")
        Add("3.2")
        Add("3.3")
        see children
    }
    see children
    oTree.children[2] {
        Add("2.1") Add("2.2") Add("2.3") {
            Add("2.3.1") Add("2.3.2") Add("test")
        }
    }
    oTree.children[2].children[3].children[3].set("2.3.3")
}

```

```
see copy(" ", 60) + nl
oTree.print()
```

output:

```
Test the tree Class Methods
The first step
data: 3.1
parent: List...
children: List...
data: 3.2
parent: List...
children: List...
data: 3.3
parent: List...
children: List...
data: one
parent: List...
children: List...
data: two
parent: List...
children: List...
data: three
parent: List...
children: List...
*****
one
two
2.1
2.2
2.3
2.3.1
2.3.2
2.3.3
three
3.1
3.2
3.3
```

44.8 Math Class

Methods:

Method	Description
<code>sin(x)</code>	Returns the sine of an angle of x radians
<code>cos(x)</code>	Returns the cosine of an angle of x radians
<code>tan(x)</code>	Returns the tangent of an angle of x radians
<code>asin(x)</code>	Returns the principal value of the arc sine of x, expressed in radians
<code>acos(x)</code>	Returns the principal value of the arc cosine of x, expressed in radians
<code>atan(x)</code>	Returns the principal value of the arc tangent of x, expressed in radians
<code>atan2(y,x)</code>	Returns the principal arc tangent of y/x, in the interval $[-\pi, +\pi]$ radians
<code>sinh(x)</code>	Returns the hyperbolic sine of x radians
<code>cosh(x)</code>	Returns the hyperbolic cosine of x radians
<code>tanh(x)</code>	Returns the hyperbolic tangent of x radians
<code>exp(x)</code>	Returns the value of e raised to the xth power
<code>log(x)</code>	Returns the natural logarithm of x
<code>log10(x)</code>	Returns the common logarithm (base-10 logarithm) of x
<code>ceil(x)</code>	Returns the smallest integer value greater than or equal to x
<code>floor(x)</code>	Returns the largest integer value less than or equal to x
<code>fabs(x)</code>	Returns the absolute value of x.
<code>pow(x,y)</code>	Returns x raised to the power of y
<code>sqrt(x)</code>	Returns the square root of x
<code>random(x)</code>	Returns a random number in the range $[0,x]$
<code>unsigned(n,n,c)</code>	Perform operation using unsigned numbers
<code>decimals(n)</code>	Determine the decimals digits after the point in float/double numbers

example:

```
Load "stdlib.ring"

oMath = new Math

See "Test the Math Class Methods" + nl
See "Sin(0) = " + oMath.sin(0) + nl
See "Sin(90) radians = " + oMath.sin(90) + nl
See "Sin(90) degree = " + oMath.sin(90*3.14/180) + nl

See "Cos(0) = " + oMath.cos(0) + nl
See "Cos(90) radians = " + oMath.cos(90) + nl
See "Cos(90) degree = " + oMath.cos(90*3.14/180) + nl

See "Tan(0) = " + oMath.tan(0) + nl
See "Tan(90) radians = " + oMath.tan(90) + nl
See "Tan(90) degree = " + oMath.tan(90*3.14/180) + nl

See "asin(0) = " + oMath.asin(0) + nl
See "acos(0) = " + oMath.acos(0) + nl
See "atan(0) = " + oMath.atan(0) + nl
See "atan2(1,1) = " + oMath.atan2(1,1) + nl

See "sinh(0) = " + oMath.sinh(0) + nl
See "sinh(1) = " + oMath.sinh(1) + nl
See "cosh(0) = " + oMath.cosh(0) + nl
See "cosh(1) = " + oMath.cosh(1) + nl
See "tanh(0) = " + oMath.tanh(0) + nl
See "tanh(1) = " + oMath.tanh(1) + nl

See "exp(0) = " + oMath.exp(0) + nl
See "exp(1) = " + oMath.exp(1) + nl
See "log(1) = " + oMath.log(1) + nl
```

```

See "log(2) = " + oMath.log(2) + nl
See "log10(1) = " + oMath.log10(1) + nl
See "log10(2) = " + oMath.log10(2) + nl
See "log10(10) = " + oMath.log10(10) + nl

See "Ceil(1.12) = " + oMath.Ceil(1.12) + nl
See "Ceil(1.72) = " + oMath.Ceil(1.72) + nl

See "Floor(1.12) = " + oMath.floor(1.12) + nl
See "Floor(1.72) = " + oMath.floor(1.72) + nl

See "fabs(1.12) = " + oMath.fabs(1.12) + nl
See "fabs(1.72) = " + oMath.fabs(1.72) + nl

See "pow(2,3) = " + oMath.pow(2,3) + nl

see "sqrt(16) = " + oMath.sqrt(16) + nl

for x = 1 to 20
    see "Random number Max (100) : " + oMath.random(100) + nl
next

x = 1.1234567890123
for d = 0 to 14
    oMath.decimals(d)
    see x + nl
next

cKey = "hello"

h = 0
for x in cKey
    h = oMath.unsigned(h,ascii(x), "+")
    h = oMath.unsigned(h,oMath.unsigned(h,10,"<<"), "+")
    r = oMath.unsigned(h,6,">>")
    h = oMath.unsigned(h, r, "^")
next
h = oMath.unsigned(h,oMath.unsigned(h,3,"<<"), "+")
h = oMath.unsigned(h,oMath.unsigned(h,11,">>"), "^")
h = oMath.unsigned(h,oMath.unsigned(h,15,"<<"), "+")

see "Hash : " + h

```

output:

```

Test the Math Class Methods
Sin(0) = 0
Sin(90) radians = 0.89
Sin(90) degree = 1.00
Cos(0) = 1
Cos(90) radians = -0.45
Cos(90) degree = 0.00
Tan(0) = 0
Tan(90) radians = -2.00
Tan(90) degree = 1255.77
asin(0) = 0
acos(0) = 1.57
atan(0) = 0
atan2(1,1) = 0.79

```

```

sinh(0) = 0
sinh(1) = 1.18
cosh(0) = 1
cosh(1) = 1.54
tanh(0) = 0
tanh(1) = 0.76
exp(0) = 1
exp(1) = 2.72
log(1) = 0
log(2) = 0.69
log10(1) = 0
log10(2) = 0.30
log10(10) = 1
Ceil(1.12) = 2
Ceil(1.72) = 2
Floor(1.12) = 1
Floor(1.72) = 1
fabs(1.12) = 1.12
fabs(1.72) = 1.72
pow(2,3) = 8
sqrt(16) = 4
Random number Max (100) : 87
Random number Max (100) : 49
Random number Max (100) : 99
Random number Max (100) : 58
Random number Max (100) : 15
Random number Max (100) : 46
Random number Max (100) : 37
Random number Max (100) : 64
Random number Max (100) : 73
Random number Max (100) : 35
Random number Max (100) : 89
Random number Max (100) : 80
Random number Max (100) : 20
Random number Max (100) : 33
Random number Max (100) : 44
Random number Max (100) : 89
Random number Max (100) : 82
Random number Max (100) : 94
Random number Max (100) : 83
Random number Max (100) : 68
1
1.1
1.12
1.123
1.1235
1.12346
1.123457
1.1234568
1.12345679
1.123456789
1.1234567890
1.12345678901
1.123456789012
1.1234567890123
1.12345678901230
Hash : 3372029979.000000000000000

```


44.9 DateTime Class

Methods:

Method	Description/Output
clock()	The number of clock ticks from program start.
time()	Get the system time.
date()	Get the date.
timelist()	List contains the date and the time information.
adddays(cDate,nDays)	Return Date from cDate and after nDays
diffdays(cDate1,cDate2)	Return the Number of days (cDate1 - cDate2)

example:

```
Load "stdlib.ring"

oDateTime = new datetime

See "Test the datetime Class Methods" + nl

See "Calculate performance" + nl
t1 = oDateTime.clock()
for x = 1 to 1000000 next
see oDateTime.clock() - t1 + nl

See "Time : " + oDateTime.time() + nl

See "Date : " + oDateTime.date() + nl

See oDateTime.TimeList()

See "Month Name : " + oDateTime.TimeList()[4]

cDate = oDateTime.date()
see cDate + nl
cDate = oDateTime.adddays(cDate,10)
see cDate + nl

cDate1 = oDateTime.date()
see cDate1 + nl
cDate2 = oDateTime.adddays(cDate1,10)
see cDate2 + nl
see "DiffDays = " + oDateTime.diffdays(cDate1,cDate2) + nl
see "DiffDays = " + oDateTime.diffdays(cDate2,cDate1) + nl
```

output:

```
Test the datetime Class Methods
Calculate performance
85
Time : 02:53:35
Date : 31/08/2016
Wed
Wednesday
Aug
August
08/31/16 02:53:35
31
```

```
02
02
244
08
53
AM
35
35
3
08/31/16
02:53:35
16
2016
Arab Standard Time
%
Month Name : August31/08/2016
10/09/2016
31/08/2016
10/09/2016
DiffDays = -10
DiffDays = 10
```

44.10 File Class

Methods:

Method	Description/Output
<code>read(cFileName)</code>	Read the file content
<code>write(cFileName,cStr)</code>	Write string to file
<code>dir(cFolderPath)</code>	Get the folder contents (files & sub folders)
<code>rename(cOld,cNew)</code>	Rename files using the <code>Rename()</code> function
<code>remove(cFileName)</code>	Delete a file using the <code>Remove()</code> function
<code>open(cFileName,cMode)</code>	Open a file using the <code>Fopen()</code> function
<code>close()</code>	Close file
<code>flush()</code>	Flushes the output buffer of a stream
<code>reopen(cFileName,cMode)</code>	Open another file using the same file handle
<code>tempfile()</code>	Creates a temp. file (binary).
<code>seek(noffset,nwhence)</code>	Set the file position of the stream
<code>tell()</code>	Know the current file position of a stream
<code>rewind()</code>	Set the file position to the beginning of the file
<code>getpos()</code>	Get handle to the current file position
<code>setpos(poshandle)</code>	Set the current file position
<code>clearerr()</code>	Clear the EOF error and the error indicators of a stream
<code>eof()</code>	Test the end-of-file indicator
<code>error()</code>	Test the error indicator
<code>perror(cErrorMessage)</code>	Print error message to the <code>stderr</code>
<code>getc()</code>	Get the next character from the stream
<code>gets(nsize)</code>	Read new line from the stream
<code>putc(cchar)</code>	Write a character to the stream
<code>puts(cStr)</code>	Write a string to the stream
<code>ungetc(cchar)</code>	Push a character to the stream
<code>fread(nsize)</code>	Read data from a stream
<code>fwrite(cString)</code>	Write data to a stream
<code>exists(cFileName)</code>	Check if a file exists

example:

```
Load "stdlib.ring"

ofile = new file

See "Test the file Class Methods" + nl
see ofile.read(filename())

see nl
ofile.open(filename(),"r")
see ofile.gets(100) + nl
ofile.close()
```

44.11 System Class

Methods:

Method	Description/Output
system()	Execute system commands
sysget()	Get environment variables
ismsdos()	Check if the operating system is MSDOS or not
iswindows()	Check if the operating system is Windows or not
iswindows64()	Check if the operating system is Windows 64bit or not
isunix()	Check if the operating system is Unix or not
ismacosx()	Check if the operating system is macOS or not
islinux()	Check if the operating system is Linux or not
isfreebsd()	Check if the operating system is FreeBSD or not
isandroid()	Check if the operating system is Android or not
windowsnl()	Get the windows new line string
sysargv()	Get the command line arguments passed to the ring script
filename()	Get the active source file

example:

```
Load "stdlib.ring"

oSystem = new System

See "Test the System Class Methods" + nl

oSystem.system("dir")
see oSystem.sysget("path") + nl
see oSystem.ismsdos() + nl
see oSystem.iswindows() + nl
see oSystem.iswindows64() + nl
see oSystem.isunix() + nl
see oSystem.ismacosx() + nl
see oSystem.islinux() + nl
see oSystem.isfreebsd() + nl
see oSystem.isandroid() + nl
see oSystem.windowsnl() + nl
see oSystem.sysargv() + nl
see oSystem.filename() + nl
```

44.12 Debug Class

Methods:

Method	Description/Output
eval(cCode)	Execute code during the runtime from string.
raise(cError)	Raise an exception.
assert(cCondition)	Test condition before executing the code.

example:

```
Load "stdlib.ring"

oDebug = new Debug
See "Test the Debug Class Methods" + nl
oDebug.eval("see 'Hello'+nl")
try
    x = 10
    oDebug.assert(x=11)
```

```
catch see "assert" + nl done
raise("Error!")
```

44.13 DataType Class

Methods:

Method	Description/Output
isstring(vValue)	We can know if the value is a string or not.
isnumber(vValue)	We can know if the value is a number or not.
islist(vValue)	We can know if the value is a list or not.
type(vValue)	Know the type of a value
isnull(vValue)	Check the value to know if it's null or not.
isalnum(vValue)	1 if the value is digit/letter or 0 if not
isalpha(vValue)	1 if the value is a letter or 0 if not
iscntrl(vValue)	1 if the value is a control character (no printing position)
isdigit(vValue)	1 if the value is a digit or 0 if not
isgraph(vValue)	1 if the value can be printed (Except space) or 0 if not
islower(vValue)	1 if the value is lowercase letter or 0 if not
isprint(vValue)	1 if the value occupies a printing position or 0 if not
ispunct(vValue)	1 if the value is a punctuation character or 0 if not
isspace(vValue)	1 if the value is a white-space or 0 if not
isupper(vValue)	1 if the value is an uppercase alphabetic letter or 0 if not
isxdigit(vValue)	1 if the value is a hexadecimal digit character or 0 if not

example:

```
Load "stdlib.ring"

oDataType = new DataType
See "Test the DataType Class Methods" + nl
see oDataType.isstring("test") + nl
see oDataType.isnumber(1) + nl
see oDataType.islist(1:3) + nl
see oDataType.type("test") + nl
see oDataType.isnull(null) + nl
see oDataType.isalnum("Hello") + nl +      # print 1
oDataType.isalnum("123456") + nl +      # print 1
oDataType.isalnum("ABCabc123") + nl + # print 1
oDataType.isalnum("How are you") + nl      # print 0 because of spaces
see oDataType.isalpha("Hello") + nl +      # print 1
oDataType.isalpha("123456") + nl +      # print 0
oDataType.isalpha("ABCabc123") + nl + # print 0
oDataType.isalpha("How are you") + nl      # print 0
See oDataType.iscntrl("hello") + nl +      # print 0
oDataType.iscntrl(nl) + nl                  # print 1
see oDataType.isdigit("0123456789") + nl +      # print 1
oDataType.isdigit("0123a") + nl
see oDataType.isgraph("abcdef") + nl +      # print 1
oDataType.isgraph("abc def") + nl          # print 0
see oDataType.islower("abcDEF") + nl +      # print 0
oDataType.islower("ghi") + nl              # print 1
see oDataType.isprint("Hello") + nl +      # print 1
oDataType.isprint("Nice to see you") + nl + # print 1
oDataType.isprint(nl) + nl                  # print 0
```

```

see oDataType.isprint("Hello") + nl           # print 1
see oDataType.isupper("welcome") + nl +      # print 0
oDataType.isupper("WELCOME") + nl           # print 1
see oDataType.isxdigit("0123456789abcdef") + nl + # print 1
oDataType.isxdigit("123z")                  # print 0

```

Output:

```

Test the DataType Class Methods
1
1
1
STRING
1
1
1
1
1
0
1
0
0
0
0
0
11
0
1
0
0
1
1
1
1
0
1
0
1
1
0
1
1
0

```

44.14 Conversion Class

Methods:

Method	Description/Output
number(vValue)	Convert strings to numbers.
string(vValue)	Convert numbers to strings.
ascii(vValue)	Get the ASCII code for a letter.
char(vValue)	Convert the ASCII code to character.
hex(vValue)	Convert decimal to hexadecimal.
dec(vValue)	Convert hexadecimal to decimal.
str2hex(vValue)	Convert string characters to hexadecimal characters.
hex2str(vValue)	Convert hexadecimal characters to string.

example:

```
Load "stdlib.ring"
```

```

oConversion = new conversion
See "Test the conversion Class Methods" + nl
See oConversion.number("3") + 5 + nl
See oConversion.string(3) + "5" + nl
See oConversion.Ascii("m") + nl
See oConversion.char(77) + nl
see oConversion.hex(162) + nl
see oConversion.dec("a2") + nl
cHex = oConversion.str2hex("Hello")
see cHex + nl
see oConversion.hex2str(cHex) + nl

```

Output:

```

Test the conversion Class Methods
8
35
109
M
a2
162
48656c6c66
Hello

```

44.15 ODBC Class

Methods:

Method	Description/Output
drivers()	Get a list of ODBC drivers.
datasources()	Get a list of ODBC data sources.
close()	Free resources.
connect(cConString)	Connect to the database.
disconnect()	Close the connection.
execute(cSQL)	Execute SQL Statements
colcount()	Get columns count in the query result
fetch()	Fetch a row from the query result
getdata(nCol)	Get column value from the fetched row
tables()	Get a list of tables inside the database
columns(cTableName)	Get a list of columns inside the table
autocommit(lStatus)	Enable or disable the auto commit feature
commit()	Commit updates to the database
rollback()	Rollback updates to the database

example:

```

Load "stdlib.ring"

oodbc = new odbc
See "Test the odbc Class Methods" + nl
oODBC {
    see drivers()
    see datasources()
    See "Connect to database" + nl
    see connect("DBQ=test.mdb;Driver={Microsoft Access Driver (*.mdb)}") + nl
    See "Select data" + nl
}

```

```

    see execute("select * from person") + nl
    nMax = colcount()
    See "Columns Count : " + nMax + nl
    while fetch()
        See "Row data:" + nl
        for x = 1 to nMax
            see getdata(x) + " - "
        next
    end
    See "Close database..." + nl
    disconnect()
    close()
}

```

44.16 MySQL Class

Methods:

Method	Description/Output
info()	Return string contains the MySQL Client version.
error()	Get the error message from the MySQL Client.
connect(cServer,cUser,cPass,cDatabase)	Connect to the MySQL database server.
close()	Close the connection to the MySQL database.
query(cQuery)	Execute SQL queries.
insert_id()	Get the inserted row id.
result()	Get the query result (data without column names).
next_result()	Move to the next query result.
columns()	Get a list of columns names.
result2()	Get all of the column names then the query result in one list.
escape_string(cStr)	Before storing binary data and special characters in the database.
autocommit(lStatus)	Enable or disable the auto commit feature.
commit()	Commit updates to the database.
rollback()	Rollback updates to the database.

example:

```

Load "stdlib.ring"

omysql = new mysql
See "Test the MySQL Class Methods" + nl
omysql {
    see info() + nl
    connect("localhost", "root", "root", "mahdb")
    see "Execute Query" + nl
    query("SELECT * FROM Employee")
    see "Print Result" + nl
    see result2()
    see "Close database" + nl
    close()
}

```

Output:

```

Test the MySQL Class Methods
5.5.30

```



```

Execute Query
Print Result
Id
Name
Salary
1
Mahmoud
15000
2
Samir
16000
3
Fayed
17000
Close database

```

44.17 SQLite Class

Methods:

Method	Description/Output
open(cDatabase)	Open Database.
close()	Close Database.
errormessage()	Get Error Message.
execute(cSQL)	Execute Query.

example:

```

Load "stdlib.ring"

osqlite = new sqlite
See "Test the sqlite Class Methods" + nl
osqlite {
    open("test.db")
    sql = "CREATE TABLE COMPANY(" +
        "ID INT PRIMARY KEY     NOT NULL," +
        "NAME           TEXT      NOT NULL," +
        "AGE             INT       NOT NULL," +
        "ADDRESS          CHAR(50)," +
        "SALARY           REAL );"

    execute(sql)

    sql = "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) " +
        "VALUES (1, 'Mahmoud', 29, 'Jeddah', 20000.00 );" +
        "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) " +
        "VALUES (2, 'Ahmed', 27, 'Jeddah', 15000.00 );" +
        "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY)" +
        "VALUES (3, 'Mohammed', 31, 'Egypt', 20000.00 );" +
        "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY)" +
        "VALUES (4, 'Ibrahim', 24, 'Egypt ', 65000.00 );"

    execute(sql)

    aResult = execute("select * from COMPANY")
    for x in aResult
        for t in x

```

```

        see t[2] + nl
    next
next
see copy(" ",50)  + nl
for x in aResult
    see x["name"] + nl
next
close()
}

```

Output:

```

Test the sqlite Class Methods
1
Mahmoud
29
Jeddah
20000.0
2
Ahmed
27
Jeddah
15000.0
3
Mohammed
31
Egypt
20000.0
4
Ibrahim
24
Egypt
65000.0
*****
Mahmoud
Ahmed
Mohammed
Ibrahim

```

44.18 Security Class

Methods:

Method	Description/Output
md5(cString)	Calculate the MD5 hash.
sha1(cString)	Calculate the SHA1 hash.
sha256(cString)	Calculate the SHA256 hash.
sha512(cString)	Calculate the SHA512 hash.
sha384(cString)	Calculate the SHA384 hash.
sha224(cString)	Calculate the SHA224 hash.
encrypt(cString,cKey,cIV)	Cncrypts the data using the Blowfish algorithm.
decrypt(cString,cKey,cIV)	Decrypt the data encrypted using the Encrypt() method.
randbytes(nSize)	Generate a string of pseudo-random bytes.

example:

```

Load "stdlib.ring"

osecuirty = new secuirty
See "Test the secuirty Class Methods" + nl
oSecuirty {
    see md5("hello") + nl +
    sha1("hello") + nl + sha256("hello") + nl +
    sha512("hello") + nl + sha384("hello") + nl +
    sha256("hello") + nl
    list = 0:15 cKey=""    for x in list cKey += char(x) next
    list = 1:8  cIV = ""   for x in list cIV += char(x) next
    cCipher = encrypt("hello",cKey,cIV)
    see cCipher + nl + decrypt(cCipher,cKey,cIV) + nl
}

```

44.19 Internet Class

Methods:

- download(cURL)
- sendemail(cSMTPServer,cEmail,cPassword,cSender,cReceiver,cCC,cTitle,cContent)

example:

```

Load "stdlib.ring"

ointernet = new internet
See "Test the internet Class Methods" + nl
ointernet {
    see download("www.ring-lang.sf.net")
}

```

DECLARATIVE PROGRAMMING USING NESTED STRUCTURES

In this chapter we are going to learn how to build declarative programming world using nested structures on the top of object oriented.

We will learn about

- Creating Objects inside Lists
- Composition and Returning Objects and Lists by Reference
- Executing code after the end of object access
- Declarative Programming on the top of Object-Oriented

45.1 Creating Objects inside Lists

We can create objects inside lists during list definition. Also we can add objects to the list at any time using the Add() function or the + operator.

Example:

```
alist = [new point, new point, new point]      # create list contains three objects
alist + [1,2,3]                                # add another item to the list

see "Item 4 is a list contains 3 items" + nl
see alist[4]

add(alist , new point)
alist + new point

alist[5] { x = 100 y = 200 z = 300 }
alist[6] { x = 50 y = 150 z = 250 }

see "Object inside item 5" + nl
see alist[5]
see "Object inside item 6" + nl
see alist[6]

class point x y z
```

Output:

```
Item 4 is a list contains 3 items
1
2
```

```

3
Object inside item 5
x: 100.000000
y: 200.000000
z: 300.000000
Object inside item 6
x: 50.000000
y: 150.000000
z: 250.000000

```

45.2 Composition and Returning Objects and Lists by Reference

When we use composition and have object as one of the class attributes, when we return that object it will be returned by reference.

if the caller used the assignment operator, another copy of the object will be created.

The caller can avoid using the assignment operator and use the returned reference directly to access the object.

The same is done also if the attribute is a list (not object).

Note: Objects and Lists are treated using the same rules. When you pass them to function they are passed by reference,

when you return them from functions they are returned by value except if it's an object attribute where a return by reference will be done.

Example:

```

o1 = new Container
myobj = o1.addobj()      # the assignment will create another copy
myobj.x = 100
myobj.y = 200
myobj.z = 300
see o1.aobjs[1]          # print the object inside the container
see myobj                 # print the copy

Class Container
    aobjs = []
    func addobj
        aobjs + new point
        return aobjs[len(aobjs)]    # return object by reference

Class point
    x = 10
    y = 20
    z = 30

```

Output:

```

x: 10.000000
y: 20.000000
z: 30.000000
x: 100.000000
y: 200.000000
z: 300.000000

```

Example(2):

```
func main
    o1 = new screen {
        content[point()] {
            x = 100
            y = 200
            z = 300
        }
        content[point()] {
            x = 50
            y = 150
            z = 250
        }
    }
    see o1.content[1]
    see o1.content[2]

Class Screen
    content = []
    func point
        content + new point
        return len(content)

Class point
    x = 10
    y = 20
    z = 30
```

Output:

```
x: 100.000000
y: 200.000000
z: 300.000000
x: 50.000000
y: 150.000000
z: 250.000000
```

Example(3):

```
func main
    o1 = New Screen {
        point() { # access the object using reference
            x = 100
            y = 200
            z = 300
        }
        point() { # access the object using reference
            x = 50
            y = 150
            z = 250
        }
    }
    see o1.content[1]
    see o1.content[2]

Class Screen
    content = []
    func point
        content + new point
```

```

        return content[len(content)]    # return the object by reference

Class point x=10 y=20 z=30

```

Output:

```

x: 100.000000
y: 200.000000
z: 300.000000
x: 50.000000
y: 150.000000
z: 250.000000

```

45.3 Executing code after the end of object access

We can access an object using { } to use object attributes and methods.

if the object contains a method called BraceEnd(), it will be executed before the end of the object access.

Example:

```

New Point { See "How are you?" + nl }

Class Point x y z
    func braceend
        see "I'm fine, Thank you!" + nl

```

Output:

```

How are you?
I'm fine, Thank you!

```

45.4 Declarative Programming on the top of Object-Oriented

The next features enable us to build and use declarative programming environment using nested structures on the top of object oriented

- using { } to access the object attributes and methods
- BraceEnd() Method
- returning objects by reference
- Setter/Getter Methods (optional)

Example:

```

# Declarative Programming (Nested Structures)

Screen()
{
    point()
    {
        x = 100
        y = 200
        z = 300
    }
}

```

```

    }

    point()
    {
        x = 50
        y = 150
        z = 250
    }
}

# Functions and Classes

Func screen return new screen

Class Screen

    content = []

    func point
        content + new point
        return content[len(content)]

    func braceend
        see "I have " + len(content) + " points!"

Class point

    x=10 y=20 z=30

    func braceend
        see self

```

Output:

```

x: 100.000000
y: 200.000000
z: 300.000000
x: 50.000000
y: 150.000000
z: 250.000000
I have 2 points!

```

45.5 More beautiful Code

We can get better results and a more beautiful code when we can avoid writing () after the method name when the methods doesn't take parameters. This feature is not provided directly by the Ring language because there is a difference between object methods and object attributes. We can get a similar effect on the syntax of the code when we define a getter method for the object attribute. For example instead of defining the point() method, we will define the point attribute then the getpoint() method that will be executed once you try to get the value of the point attribute. since we write the variable name directly without () we can write point instead of point() and the method getpoint() will create the object and return the object reference for us.

Example:

```

new Container
{

```



```
    Point
    {
        x=10
        y=20
        z=30
    }
}

Class Container
    aObjs = []
    point
    func getpoint
        aObjs + new Point
        return aObjs[len(aObjs)]

Class Point x y z
    func braceend
        see "3D Point" + nl + x + nl + y + nl + z + nl
```

Output

```
3D Point
10
20
30
```

NATURAL LANGUAGE PROGRAMMING

Using the Ring programming language, we can create Natural programming languages based on classes and objects.

46.1 History

In 2010, I developed a new programming language called Supernova (developed using PWCT). This language uses a code that looks similar to Natural Language statements to create simple GUI applications. Now after five years, In the Ring programming language, we can get similar results, but now we have the ability to create/use code similar to Natural language statements in any domain that we like or need.

The Ring programming language comes with the Supernova sprite, but with more generalization and with mix of other languages sprites.

46.2 Example

The next example presents how to create a class that define two instructions

The first instruction is : I want window

The second instruction is : Window title = <expr>

Also keywords that can be ignored like the 'the' keyword

```
New App
{
    I want window
    The window title = "hello world"
}

Class App

    # Attributes for the instruction I want window
    i want window
    nIwantwindow = 0

    # Attributes for the instruction Window title
    # Here we don't define the window attribute again
    title
    nWindowTitle = 0

    # Keywords to ignore, just give them any value
    the=0

    func geti
```

```

        if nIwantwindow = 0
            nIwantwindow++
        ok

    func getwant
        if nIwantwindow = 1
            nIwantwindow++
        ok

    func getwindow
        if nIwantwindow = 2
            nIwantwindow= 0
            see "Instruction : I want window" + nl
        ok
        if nWindowTitle = 0
            nWindowTitle++
        ok

    func setttitle cValue
        if nWindowTitle = 1
            nWindowTitle=0
            see "Instruction : Window Title = " + cValue + nl
        ok

```

Output:

```

Instruction : I want window
Instruction : Window Title = hello world

```

46.3 Change the Ring Keyword ‘And’

What if we want to connect between the two instructions using ‘and’

We have a problem because in Ring ‘and’ is a keyword

We can change that using the ChangeRingKeyword command.

Syntax:

```
ChangeRingKeyword <oldkeyword> <newkeyword>
```

Note: remember to restore the keyword again

Tip: The ChangeRingKeyword command is executed in the scanner stage by the compiler (before parsing).

Example:

```

ChangeRingKeyword      and  _and

New App
{
    I want window and the window title = "hello world"
}

Class App

```

```

# Attributes for the instruction I want window
    i want window
    nIwantwindow = 0
# Attributes for the instruction Window title
# Here we don't define the window attribute again
    title
    nWindowTitle = 0
# Keywords to ignore, just give them any value
    the=0 and=0

ChangeRingKeyword    _and and

func geti
    if nIwantwindow = 0
        nIwantwindow++
    ok

func getwant
    if nIwantwindow = 1
        nIwantwindow++
    ok

func getwindow
    if nIwantwindow = 2
        nIwantwindow= 0
        see "Instruction : I want window" + nl
    ok
    if nWindowTitle = 0
        nWindowTitle++
    ok

func setttitle cValue
    if nWindowTitle = 1
        nWindowTitle=0
        see "Instruction : Window Title = " + cValue + nl
    ok

func getand
    see "Using : and" + nl

```

Output:

```

Instruction : I want window
Using : and
Instruction : Window Title = hello world

```

46.4 Change the Ring Operator ‘+’

What if we want to define a new behavior for any operator like the “+” operator.

We can do this change using the ChangeRingOperator command to hide operator (change it’s name)

Then we can use the operator as identifier that we can handle it’s behaviour

Syntax:

```
ChangeRingOperator <oldoperator> <newoperator>
```

Note: remember to restore the operator again

Tip: The ChangeRingOperator command is executed in the scanner stage by the compiler (before parsing).

Example:

```
ChangeRingOperator + _+

New App {
    +
}

Class App
    +
    func get+
        see "Plus operator"

ChangeRingOperator _+ +
```

Output:

```
Plus operator
```

46.5 Change the '=' operator to 'is'

Example:

```
ChangeRingKeyword      and _and
ChangeRingOperator     =    is

New App
{
    I want window and the window title is "hello world"
}

ChangeRingOperator     is    =

Class App

    # Attributes for the instruction I want window
    i want window
    nIwantwindow = 0
    # Attributes for the instruction Window title
    # Here we don't define the window attribute again
    title
    nWindowTitle = 0
    # Keywords to ignore, just give them any value
    the=0 and=0

ChangeRingKeyword      _and and

    func geti
        if nIwantwindow = 0
```

```

                                nIwantwindow++
                                ok

func getwant
    if nIwantwindow = 1
        nIwantwindow++
    ok

func getwindow
    if nIwantwindow = 2
        nIwantwindow= 0
        see "Instruction : I want window" + nl
    ok
    if nWindowTitle = 0
        nWindowTitle++
    ok

func setttitle cValue
    if nWindowTitle = 1
        nWindowTitle=0
        see "Instruction : Window Title = " + cValue + nl
    ok

```

46.6 Using Eval() with our Natural Code

Example:

```

func Main

    cProgram = ' I want window and the window title is "hello world" '

    MyLanguage(cProgram)

Func MyLanguage cCode

    # We add to the code the instructions that change keywords and operators
    # Because Eval() uses a new Compiler Object (the original keywords and operators).

    cCode = '
        ChangeRingKeyword  and  _and
        ChangeRingOperator =    is
    ' + cCode

    New App
    {
        eval(cCode)
    }

    Class App

        # Attributes for the instruction I want window
        i want window
        nIwantwindow = 0
        # Attributes for the instruction Window title
        # Here we don't define the window attribute again

```

```

        title
        nWindowTitle = 0
# Keywords to ignore, just give them any value
the=0

ChangeRingKeyword and _and
and=0
ChangeRingKeyword _and and

func geti
    if nIwantwindow = 0
        nIwantwindow++
    ok

func getwant
    if nIwantwindow = 1
        nIwantwindow++
    ok

func getwindow
    if nIwantwindow = 2
        nIwantwindow= 0
        see "Instruction : I want window" + nl
    ok
    if nWindowTitle = 0
        nWindowTitle++
    ok

func setttitle cValue
    if nWindowTitle = 1
        nWindowTitle=0
        see "Instruction : Window Title = " + cValue + nl
    ok

```

46.7 BraceStart and BraceEnd Methods

We can write code that will be executed before/after using { }

Example:

```

o1 = new test {
    see "Hello" + nl
}

o1 {}

class test

    func bracestart
        see "start" + nl

    func braceend
        see "end" + nl

```

Output:

```
start
Hello
end
start
end
```

46.8 BraceExprEval Method

The next example demonstrates how to use the “BraceExprEval” method to get expressions in Natural code.

Example:

```
new natural {
    create 5
}

class natural
    create=0
    lkeyword = false
    func braceexpr eval r
        if lkeyword lkeyword=false return ok
        see "expr eval" + nl
        see "type: " + type(r) see nl
        see "value : " see r see nl
    func getcreate
        lkeyword = true
        see "create" + nl
```

Output:

```
create
expr eval
type: NUMBER
value : 5
```

46.9 Real Natural Code

The next example is a more advanced example

```
# Natural Code
new program {
    Accept 2 numbers then print the sum
}

# Natural Code Implementation
class program
    # Keywords
        Accept=0 numbers=0 then=0 print=0 the=0 sum=0

    # Execution
    func braceexpr eval x
        value = x
    func getnumbers
        for x=1 to value
            see "Enter Number (" + x + ") : " give nNumber
```



```

                                aNumbers + nNumber
                                next
func getsum
    nSum = 0
    for x in aNumbers nSum+= x next
    see "The Sum : " + nSum
private
    value=0 aNumbers=[]

```

Output:

```

Enter Number (1) :3
Enter Number (2) :4
The Sum : 7

```

46.10 BraceError() Method

The next examples demonstrates how to use the “BraceError” method to handle errors when accessing the object using braces {}.

Example:

```

func main
    o1 = new point {
        x=10 y=20 z=30
        TEST
        SEE test
    }

class point x y z
    func braceerror
        see "Handle Error!" + nl
        SEE "Message :" + cCatchError + nl
        if ( left(cCatchError,11) = "Error (R24)" ) and not isattribute(self,"test")
            see "add attribute" + nl
            addattribute(self,"test")
            test = 10

        ok
        see "done" + nl
        return

```

Output:

```

Handle Error!
Message :Error (R24) : Using uninitialized variable : test
add attribute
done
10

```

Example:

```

new point {
    x=10 y=20 z=30
    test()
    see "mmm..." + NL
}

```

```

class point x y z
  func braceerror
    see "Handle Error!" + nl
    see "Message :" + cCatchError + nl
    see self
    see "Done" + NL

```

Output:

```

Handle Error!
Message :Error (R3) : Calling Function without definition !: test
x: 10.000000
y: 20.000000
z: 30.000000
Done
mmm...

```

46.11 Clean Natural Code

Instead of typing the literal as “literal” we can accept the words directly.

Example:

The next example accept hello world instead of “hello world”

But this example uses braceend() to check the end of the instruction

This means that this class process only one natural statement that end with literal.

```

ChangeRingKeyword      and _and

New App
{
    I want window and the window title is hello world
}

Class App

    # Attributes for the instruction I want window
    i want window
    nIwantwindow = 0
    # Attributes for the instruction Window title
    # Here we don't define the window attribute again
    title is
    nWindowTitle = 0
    # Keywords to ignore, just give them any value
    the=0 and=0
    # Data
    literal = ""

ChangeRingKeyword      _and and

    func geti
        if nIwantwindow = 0
            nIwantwindow++
        ok

    func getwant

```

```
        if nIwantwindow = 1
            nIwantwindow++
        ok

func getwindow
    if nIwantwindow = 2
        nIwantwindow= 0
        see "Instruction : I want window" + nl
    ok
    if nWindowTitle = 0
        nWindowTitle++
    ok

func gettitle
    if nWindowTitle = 1
        nWindowTitle=2
    ok

func getis
    if nWindowTitle = 2
        nWindowTitle=3
    ok

func braceend
    if nWindowTitle = 3
        see "Instruction : Window Title = " + literal + nl
        nWindowTitle = 0
    ok

func braceerror
    c= substr(cCatchError,":")
    while c > 0
        c= substr(cCatchError,":")
        cCatchError=substr(cCatchError,c+1)
    end
    literal += substr(cCatchError,1)
```

USING THE NATURAL LIBRARY

In this chapter we will learn how to use the Natural Library to quickly define a language that contains a group of commands.

To start using the library, We need to call `naturallib.ring`

```
load "naturallib.ring"
```

After loading the library, We can use the `NaturalLanguage` class that contains the next methods :-

- `SetLanguageName(cLanguageName)`
- `setCommandsPath(cFolder)`
- `SetPackageName(cPackageName)`
- `UseCommand(cCommandName)`
- `SetOperators(cOperators)`
- `RunFile(cFileName)`
- `RunString(cString)`

47.1 Natural Library - Demo Program

We will write the natural code in a Text file, for example `program.txt`

File: `program.txt`

```
Welcome to the Ring programming language!
What you are reading now is not comments, I swear!

After many years of programming I decided to think different about
programming and solve the problems in a better way.

We are writing commands or code and the Ring language is reading
it to understand us! Sure, What you are seeing now is
just ***part of the code - Not the Complete Program***
You have to write little things before and after this
part to be able to run it!

It is the natural part of our code where we can write in English,
Arabic or any Natural Language Then we will tell the computer
through the Ring language what must happens! in a way that we can scale
for large frameworks and programs.
```

Just imagine what will happens to the world of programming once we create many powerful frameworks using the Ring language that uses this way (Natural Programming).

For example When we say Hello to the Machine, It can reply! and when we say count from 1 to 5 it will understand us, Also if we said count from 5 to 1 it will understand us too! You can see the Output window!

This Goal is not new, but the Ring language comes with an innovative solution to this problem.

Output:

```
Hello, Sir!

The Numbers!

1
2
3
4
5

I will count Again!

5
4
3
2
1
```

To execute the natural code, We have start.ring

In start.ring we define the language and the commands.

File: start.ring

```
load "stdlib.ring"
load "naturallib.ring"

New NaturalLanguage {
    SetLanguageName(:MyLanguage)
    SetCommandsPath(CurrentDir()+"/../command")
    SetPackageName("MyLanguage.Natural")
    UseCommand(:Hello)
    UseCommand(:Count)
    RunFile("program.txt")
}
```

We defined a language called MyLanguage, We have folder for the language commands.

Each command will define a class that belong to the MyLanguage.Natural package.

We will define two commands, Hello and Count.

So we must have two files for defining the commands in the CurrentDir()+"/../command" folder

File: hello.ring

```
DefineNaturalCommand.SyntaxIsKeyword([
    :Package = "MyLanguage.Natural",
    :Keyword = :hello,
    :Function = func {
        See "Hello, Sir!" + nl + nl
    }
])
```

File: count.ring

```
DefineNaturalCommand.SyntaxIsKeywordNumberNumber([
    :Package = "MyLanguage.Natural",
    :Keyword = :count,
    :Function = func {
        if not isattribute(self, :count_times) {
            AddAttribute(self, :count_times)
            Count_Times = 0
        }
        if Expr(1) > Expr(2) {
            nStep = -1
        }
        else {
            nStep = 1
        }
        if Count_Times = 0 {
            see nl+"The Numbers!" + nl
            Count_Times++
        }
        else {
            see nl + "I will count Again!" +nl
        }
        for x = Expr(1) to Expr(2) step nStep {
            see nl+x+nl
        }
        CommandReturn(fabs(Expr(1)-Expr(2))+1)
    }
])
```

47.2 Defining Commands

To define new command we can use the DefineNaturalCommand object

This object provides the next methods :-

- SyntaxIsKeyword(aPara)
- SyntaxIsKeywordNumber(aPara)
- SyntaxIsKeywordNumberNumber(aPara)
- SyntaxIsKeywordNumbers(aPara,nCount)
- SyntaxIsKeywordString(aPara)
- SyntaxIsKeywordStringString(aPara)

- SyntaxIsKeywordStrings(aPara,nCount)
- SyntaxIsKeywordExpression(aPara)
- SyntaxIsKeywordExpressionExpression(aPara)
- SyntaxIsKeywordExpressions(aPara,nCount)
- SyntaxIsCommand(aPara)
- SyntaxIsCommandNumber(aPara)
- SyntaxIsCommandNumberNumber(aPara)
- SyntaxIsCommandNumbers(aPara,nCount)
- SyntaxIsCommandString(aPara)
- SyntaxIsCommandStringString(aPara)
- SyntaxIsCommandStrings(aPara,nCount)
- SyntaxIsCommandExpression(aPara)
- SyntaxIsCommandExpressionExpression(aPara)
- SyntaxIsCommandExpressions(aPara,nCount)

File: mylanguage.ring

```
load "stdlib.ring"
load "naturallib.ring"

MyLanguage = New NaturalLanguage {
    SetLanguageName (:MyLanguage)
    setCommandsPath (CurrentDir () + "/../command")
    SetPackageName ("MyLanguage.Natural")
    UseCommand (:Hello)
    UseCommand (:Count)
    UseCommand (:Print)
    UseCommand (:IWantWindow)
    UseCommand (:WindowTitleIs)
    UseCommand (:IWantButton)
}
```

Example (1)

In the next example we will define the Print command.

We will use the SyntaxIsKeywordExpression() Method.

We pass list (as Hash) to the method. We determine the package name, the keyword and the function that will be executed.

Inside this function we uses the Expr(nExprNumber) function to get the expression value that the user will write after the keyword.

File: print.ring

```
DefineNaturalCommand.SyntaxIsKeywordExpression ([
    :Package = "MyLanguage.Natural",
    :Keyword = :print,
    :Function = func {
        See Expr(1)
    }
])
```

Usage:

```
load "mylanguage.ring"

MyLanguage.RunString('
    print "Hello, World!"
')
```

Output:

```
Hello, World!
```

Example (2)

File: `iwantwindow.ring`

```
DefineNaturalCommand.SyntaxIsCommand([
    :Package = "MyLanguage.Natural",
    :Command = "i want window",
    :Function = func {
        See "Command: I want window" + nl
    }
])
```

Usage:

```
load "mylanguage.ring"

MyLanguage.RunString('
    i want window
')
```

Output:

```
Command: I want window
```

Example (3)

File: `windowtitleis.ring`

```
DefineNaturalCommand.SyntaxIsCommandString([
    :Package = "MyLanguage.Natural",
    :Command = "window title is",
    :Function = func {
        See "Command: Window title is " + Expr(1) + nl
    }
])
```

Usage:

```
load "mylanguage.ring"

MyLanguage.RunString('
    I want window and the window title is "Hello World"
')
```

Output:

```
Command: I want window
Command: Window title is Hello World
```


47.3 Natural Library - Operators

In the next example we uses the Count command without using operators

```
load "mylanguage.ring"

MyLanguage.RunString("
    Hello
    Count 1 5
    Count 5 1
")
```

We can add more description

```
load "mylanguage.ring"

MyLanguage.RunString("
    Hello, Please    Count from 1 to 5 then count from 5 to 1
")
```

Also we can use operators like “(” and ”)” around the instruction

```
load "mylanguage.ring"

MyLanguage {
    SetOperators("(" ")")
    RunString("
        Here we will play and will try something
        that looks like Lisp Syntax
        (count (count 1 5) (count 20 15))
        Just for fun!
    ")
}
```

47.4 Defining commands using classes

This section is related to the implementation details.

When we define new command, Each command is defined by the Natural Library as a class.

We have the choice to define commands using the simple interface provided by the DefineNaturalCommand object or by defining new class as in the next examples.

If we used DefineNaturalCommand (More Simple), The class will be defined during the runtime.

File: hello.ring

```
Package MyLanguage.Natural

class Hello

    func AddAttributes_Hello
        AddAttribute(self, :hello)

    func GetHello
        See "Hello, Sir!" + nl + nl
```

File: count.ring

```
Package MyLanguage.Natural
```

```
class Count
```

```
    func Getcount
```

```
        StartCommand()
        CommandData()[:name] = :Count
        CommandData()[:nExpr] = 0
        CommandData()[:aExpr] = []
```

```
    func BraceExprEval_Count nValue
```

```
        if isCommand() and CommandData()[:name] = :Count {
            if isNumber(nValue) {
                CommandData()[:nExpr]++
                CommandData()[:aExpr] + nValue
                if CommandData()[:nExpr] = 2 {
                    Count_Execute()
                }
            }
        }
```

```
    func AddAttributes_Count
```

```
        AddAttribute(self, :count)
```

```
    func Count_Execute
```

```
        if not isattribute(self, :count_times) {
            AddAttribute(self, :count_times)
            Count_Times = 0
        }
        if Expr(1) > Expr(2) {
            nStep = -1
        }
        else
            nStep = 1
        }
        if Count_Times = 0 {
            see nl+"The Numbers!" + nl
            Count_Times++
        }
        else
            see nl + "I will count Again!" +nl
        }
        for x = Expr(1) to Expr(2) step nStep {
            see nl+x+nl
        }
        CommandReturn(fabs(Expr(1)-Expr(2))+1)
```

WEB DEVELOPMENT (CGI LIBRARY)

In this chapter we will learn about developing Web applications using a CGI Library written in the Ring language.

48.1 Configure the Apache web server

We can use Ring with any web server that support CGI. In this section we will learn about using Ring with the Apache HTTP Server.

You can download Apache from : <http://httpd.apache.org/>

Or you can get it included with other projects like

XAMPP : <https://www.apachefriends.org/download.html>

Install then open the file:

```
xampp\apache\conf\httpd.conf
```

search for

```
<Directory />
```

Then after it add

```
Options FollowSymLinks +ExecCGI
```

So we have

```
<Directory />
Options FollowSymLinks +ExecCGI
```

Search for the next line and be sure that it's not commented

```
LoadModule cgi_module modules/mod_cgi.so
```

Search for : AddHandler cgi-script

Then add ".ring" to the supported cgi extensions

Example

```
AddHandler cgi-script .cgi .ring
```

Example

```
AddHandler cgi-script .cgi .pl .asp .ring
```

Run/Start the server

Create your web applications in a directory supported by the web server.

Example:

```
Apache2.2\htdocs\mywebapplicationfolder
```

Example:

```
xampp\htdocs\mywebapplicationfolder
```

Inside the source code file (*.ring), Add this line

```
#!c:\ring\bin\ring.exe -cgi
```

Note: Change the previous line based on the path to ring.exe in your machine

48.2 Ring CGI Hello World Program

The next program is the Hello World program

```
#!c:\ring\bin\ring.exe -cgi

See "content-type : text/html" +nl+nl+
    "Hello World!" + nl
```

48.3 Hello World Program using the Web Library

We can use the web library to write CGI Web applications quickly

Example (1) :

```
#!c:\ring\bin\ring.exe -cgi

Load "weblib.ring"

Import System.Web

New Page
{
    Text ("Hello World!")
}
```

Example (2) :

```
#!c:\ring\bin\ring.exe -cgi

Load "weblib.ring"

Import System.Web

WebPage ()
{
    Text ("Hello World!")
}
```

Tip: the difference between ex. 1 and ex. 2 is using `WebPage()` function to return the page object instead of creating the object using new statement.

48.4 Web Library Features

The next features are provided by the Web library to quickly create web applications.

- Generate HTML pages using functions
- Generate HTML pages using objects
- HTTP Get
- HTTP Post
- Files Upload
- URL Encode
- Templates
- CRUD MVC Sample
- Users Logic & Registration Sample

48.5 HTTP Get Example

The Page User Interface

```
#!c:\ring\bin\ring.exe -cgi
Load "weblib.ring"
Import System.Web
New Page
{
    Title = "Test HTTP Get"
    divstart([ :style = StyleSizeFull() ] )
    boxstart()
        text( "Test HTTP GET" )
        newline()
    boxend()
    divstart([ :style = Styledivcenter("600px","550px") +
        StyleGradient(21) ])
    divstart([:style = stylefloatleft() + stylesize("100px","100%") +
        stylecolor("black") + stylegradient(58)])
    formstart("ex5.ring")
        tablestart([ :style = stylesize("65%","90%") +
            stylemarginleft("35%") +
            stylemargintop("30%") ])
            rowstart([])
                cellstart([])
                    text ( "Name : " )
                cellend()
                cellstart([])
                    cTextboxStyle = StyleMarginLeft("5%") +
                        StyleWidth("250px") +
                        StyleColor("black") +
```

```

                StyleBackColor("white")
                textbox([ :name = "Name", :style = cTextboxStyle ] )
            cellend()
        rowend()
        rowstart([])
            cellstart([])
                text ( "Address : " )
            cellend()
            cellstart([])
                textbox([ :name = "Address", :style = cTextboxStyle ] )
            cellend()
        rowend()
        rowstart([])
            cellstart([])
                text ( "Phone : " )
            cellend()
            cellstart([])
                textbox([ :name = "Phone", :style = cTextboxStyle ] )
            cellend()
        rowend()
        rowstart([])
            cellstart([])
                text ( "Age : " )
            cellend()
            cellstart([])
                textbox([ :name = "Age", :style = cTextboxStyle ] )
            cellend()
        rowend()
        rowstart([])
            cellstart([])
                text ( "City: " )
            cellend()
            cellstart([])
                listbox([ :name = "City", :items = ["Cairo","Riyadh","Jeddah"],
                    :style = stylemarginleft("5%") + stylewidth("400px") ] )
            cellend()
        rowend()
        rowstart([])
            cellstart([])
                text ( "Country : " )
            cellend()
            cellstart([])
                combobox([ :name = "Country",
                    :items = ["Egypt","Saudi Arabia","USA"],
                    :style = stylemarginleft("5%") +
                        stylewidth("400px")+
                        stylecolor("black")+
                        stylebackcolor("white")+
                        stylefontsize("14px") ])
            cellend()
        rowend()
        rowstart([])
            cellstart([])
                text ( "Note : " )
            cellend()
            cellstart([])
                editbox([ :name = "Notes",
                    :style = stylemarginleft("5%") +

```

```
                stylesize("400px","100px")+
                stylecolor("black")+
                stylebackcolor("white") ,
            :value = "write comments here..." ] )
        cellend()
    rowend()
    rowstart([])
        cellstart([])
        cellend()
        cellstart([])
            submit([ :value = "Send" , :Style = stylemarginleft("5%") ])
        cellend()
    rowend()
tableend()
formend()
divend()
divend()
divend()
}
```

Screen Shot:

The screenshot shows a web browser window titled 'Test HTTP Get' with the address bar displaying 'localhost/ringapp/ex4.ring'. The page content is titled 'Test HTTP GET' in yellow text on a black background. Below this is a form with a light blue gradient background. The form fields are as follows:

- Name :** Mahmoud Fayed
- Address :** Demerdash
- Phone :** 00000000
- Age :** 28
- City:** A dropdown menu with 'Cairo' selected, and other options 'Riyadh' and 'Jeddah' visible.
- Country :** A dropdown menu with 'Egypt' selected.
- Note :** A text area containing the placeholder text 'write comments here...'. Below the text area is a 'Send' button.

The Response

```
#!c:\ring\bin\ring.exe -cgi
Load "weblib.ring"
Import System.Web
New Page
{
  divstart([ :style = styledivcenter("800px","500px") ])
  boxstart()
    text ( "HTTP GET Response" )  newline()
  boxend()
  divstart([ :style = stylefloatleft()+stylewidth("10%")+
```



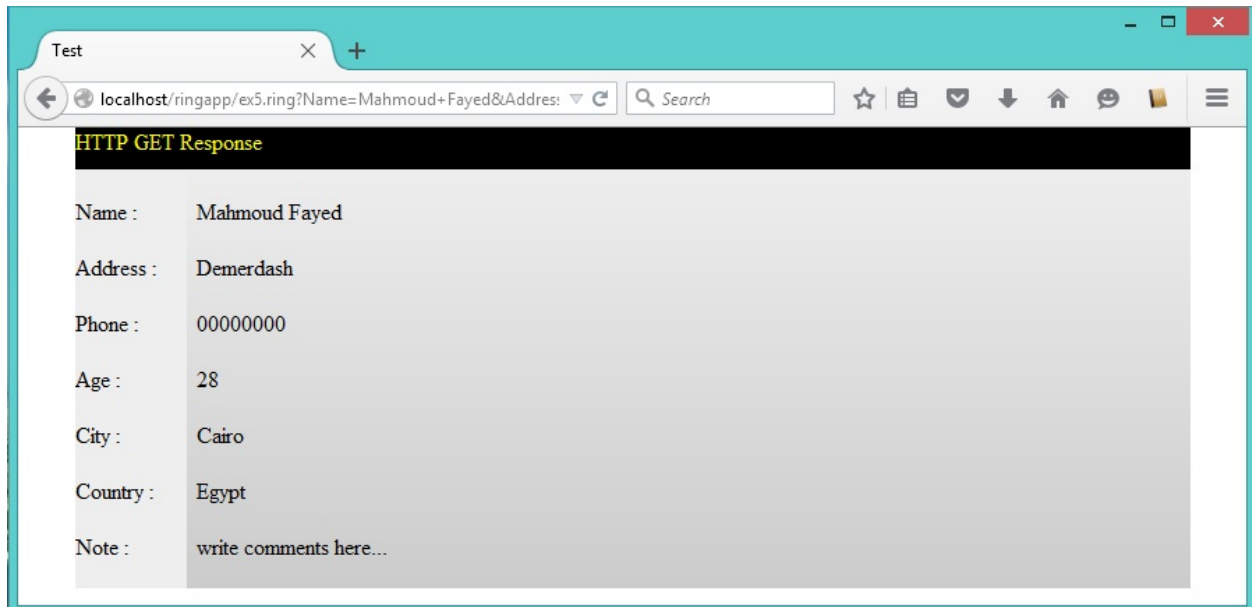
```

                                stylecolor("black")+stylegradient(58) ])

    newline()
    text ( "Name : " )
    newline() newline()
    text ( "Address : " )
    newline() newline()
    text ( "Phone : " )
    newline() newline()
    text ( "Age : " )
    newline() newline()
    text ( "City : " )
    newline() newline()
    text ( "Country : " )
    newline() newline()
    text ( "Note : " )
    newline() newline()
divend()
divstart([ :style = stylefloatleft()+stylewidth("90%")+
                                stylecolor("black")+stylegradient(47) ])
    divstart([ :style = stylefloatleft() + stylewidth("1%") ])
        newline()
    divend()
divstart([ :style = stylefloatleft() + stylewidth("95%") ])
    newline()
    text ( aPageVars["Name"] )
    newline() newline()
    text ( aPageVars["Address"] )
    newline() newline()
    text ( aPageVars["Phone"] )
    newline() newline()
    text ( aPageVars["Age"] )
    newline() newline()
    text ( aPageVars["City"] )
    newline() newline()
    text (aPageVars["Country"] )
    newline() newline()
    text ( aPageVars["Notes"] )
    newline() newline()
divend()
divend()
divend()
}

```

Screen Shot:

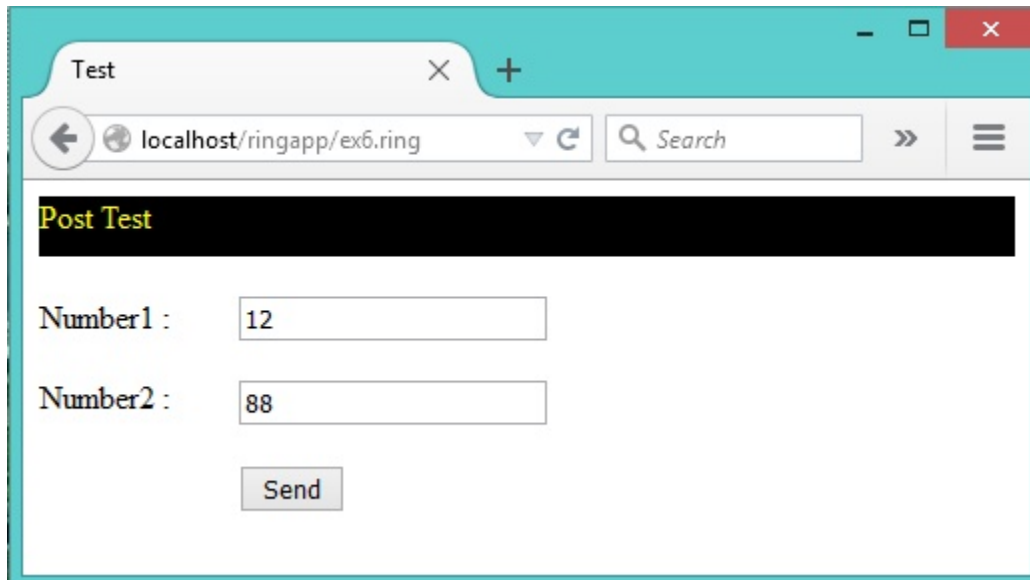


48.6 HTTP POST Example

The Page User Interface

```
#!c:\ring\bin\ring.exe -cgi
Load "weblib.ring"
Import System.Web
New Page
{
    boxstart()
        text( "Post Test")
        newline()
    boxend()
    divstart([ :style=StyleFloatLeft()+StyleWidth("100px") ])
        newline()
        text( "Number1 : " )      newline() newline()
        text( "Number2 : " )      newline() newline()
    divend()
    formpost( "ex7.ring")
        divstart([ :style = styleFloatLeft()+StyleWidth("200px") ])
            newline()
            textbox([ :name = "Number1" ]) newline() newline()
            textbox([ :name = "Number2" ]) newline() newline()
            submit([ :value = "Send" ] )
        divend()
    formend()
}
```

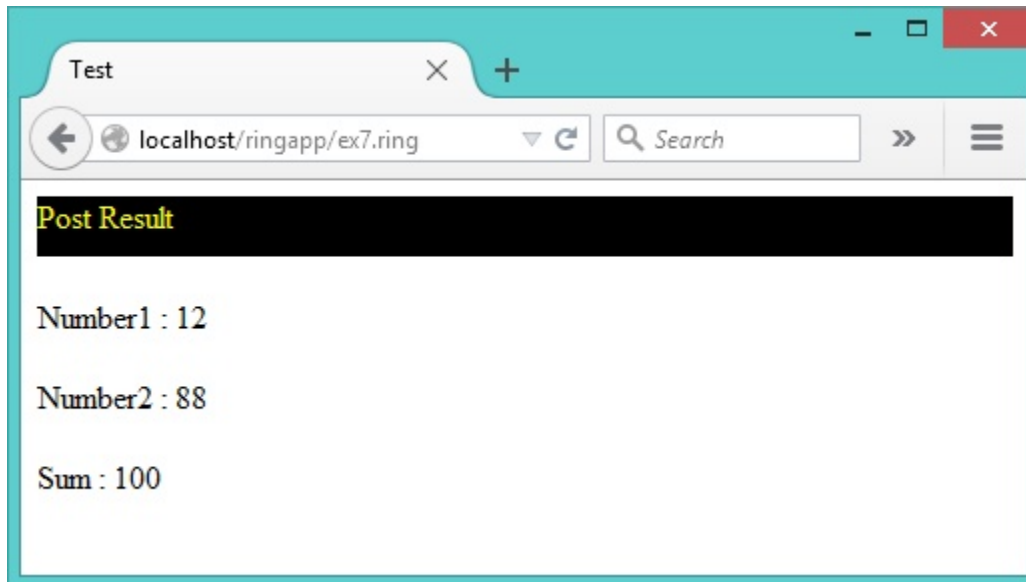
Screen Shot:



The Response

```
#!c:\ring\bin\ring.exe -cgi
Load "weblib.ring"
Import System.Web
New Page
{
    boxstart()
        text( "Post Result" )
        newline()
    boxend()
    divstart([ :style = styleFloatLeft()+styleWidth("200px") ])
        newline()
        text( "Number1 : " + aPageVars["Number1"] )
        newline() newline()
        text( "Number2 : " + aPageVars["Number2"] )
        newline() newline()
        text( "Sum : " + (0 + aPageVars["Number1"] + aPageVars["Number2"] ) )
        newline()
    divend()
}
```

Screen Shot:

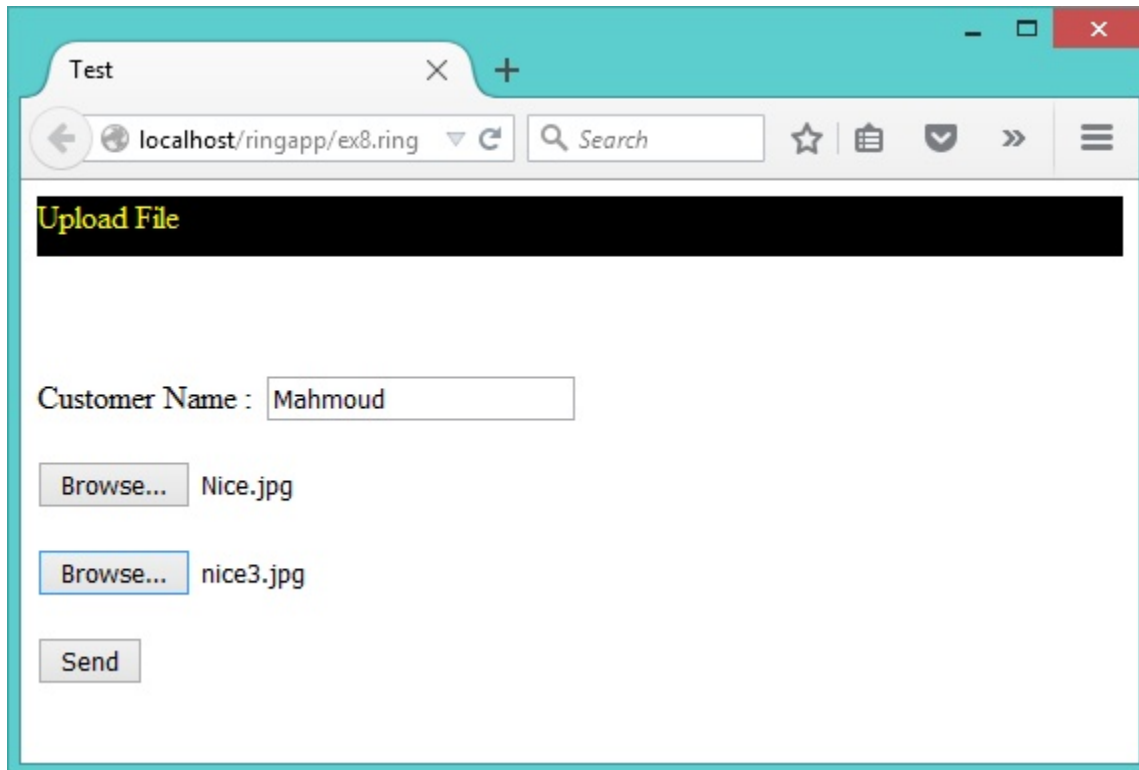


48.7 Upload Files

The Page User Interface

```
#!c:\ring\bin\ring.exe -cgi
Load "weblib.ring"
Import System.Web
New page
{
    boxstart()
        text( "Upload File" )
        newline()
    boxend()
    for x = 1 to 3 newline() next
    formupload("ex9.ring")
        text( "Customer Name : " )
        textbox([ :name = "custname" ])
        newline() newline()
        divstart([ :style = styleFloatLeft() + styleWidth("90px") ])
            uploadfile("file")  newline() newline()
            uploadfile("file2") newline() newline()
            submit([ :value = "Send" ])
        divend()
    formend()
}
```

Screen Shot:



The Response

```
#!c:\ring\bin\ring.exe -cgi
Load "weblib.ring"
Import System.Web

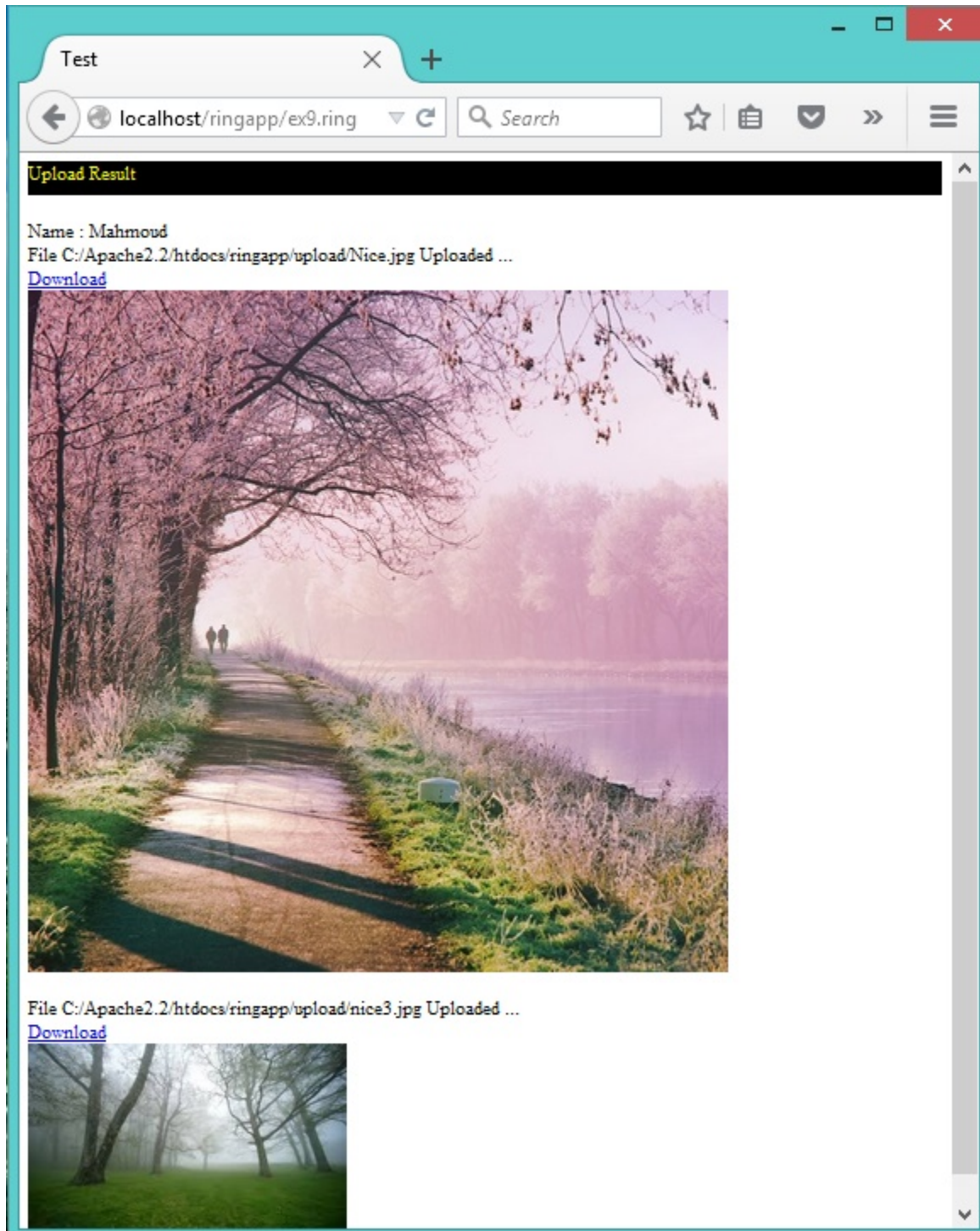
cUploadPath = "C:/Apache2.2/htdocs/ringapp/upload/"
cUploadFolder = "/ringapp/upload/"

New page
{
    boxstart()
        text( "Upload Result" )
        newline()
    boxend()
    newline()
    divstart([ :style= styleFloatLeft() + styleWidth("100px") ])
        text( "Name : " + aPageVars["custname"] )
        newline()
    divend()
    if aPageVars["file"] != char(13)
        getuploadedfile(self,"file")
    ok
    if aPageVars["file2"] != char(13)
        getuploadedfile(self,"file2")
    ok
}

Func getuploadedfile oObj,cFile
    # here we use object.property
    # instead of object { } to avoid executing braceend method
    cFileName = cUploadPath + oObj.getfilename(aPageVars,cFile)
```

```
write(cFileName,aPageVars[cFile])
system("chmod a+x "+cFileName)
oObj.newline()
oObj.text( "File "+cFileName+ " Uploaded ..." )
oObj.newline()
imageURL = cUploadFolder+oObj.getfilename(aPageVars,cFile)
oObj.link([ :url = imageURL, :title = "Download" ])
oObj.newline()
oObj.image( [ :url = imageURL , :alt = :image ] )
oObj.newline()
```

Screen Shot:



48.8 Cookies

The Page User Interface

```
#!c:\ring\bin\ring.exe -cgi
Load "weblib.ring"
```

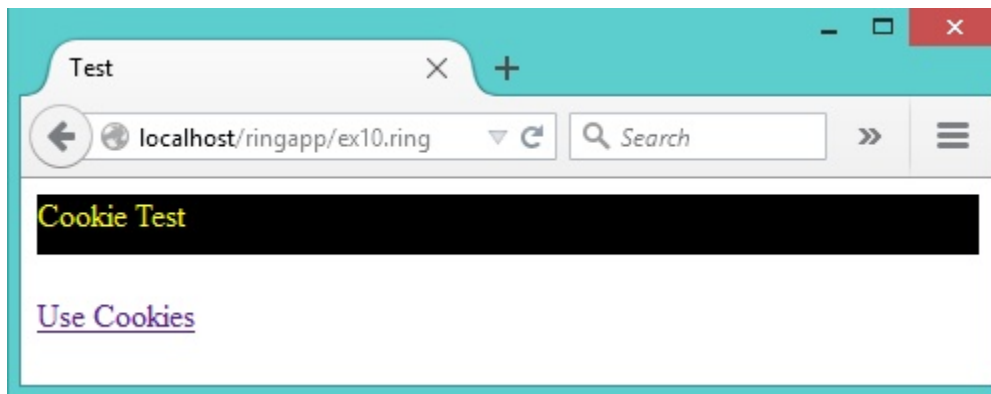
```

Import System.Web

New page
{
    boxstart()
        text( "Cookie Test" )
        newline()
    boxend()
    newline()
    link([ :url = "ex11.ring", :title = "Use Cookies" ])
    cookie("custname","Mahmoud Fayed")
    cookie("custage",28)
}

```

Screen Shot:



The Response

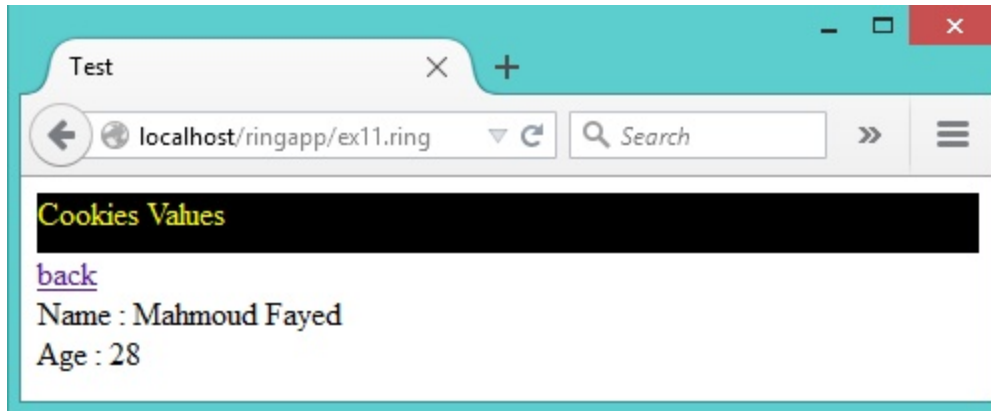
```

#!c:\ring\bin\ring.exe -cgi
Load "weblib.ring"
Import System.Web

New Page
{
    boxstart()
        text( "Cookies Values" )
        newline()
    boxend()
    link([ :url = "ex10.ring", :title = "back" ])
    newline()
    divstart([:style="float:left;width:200px"])
        text( "Name : " + aPageVars["custname"] )
        newline()
        text( "Age : " + aPageVars["custage"] )
        newline()
    divend()
}

```

Screen Shot:



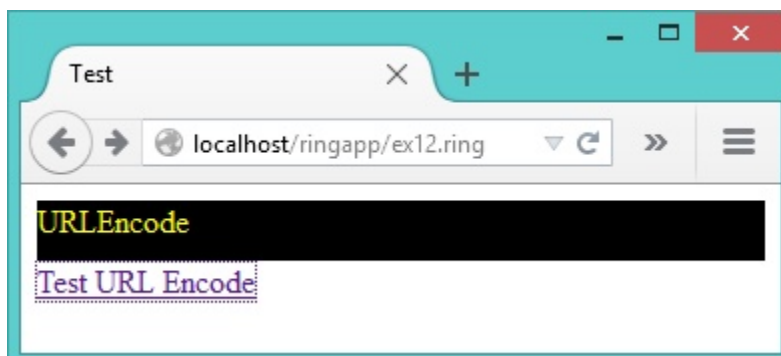
48.9 URL Encode

The Page User Interface

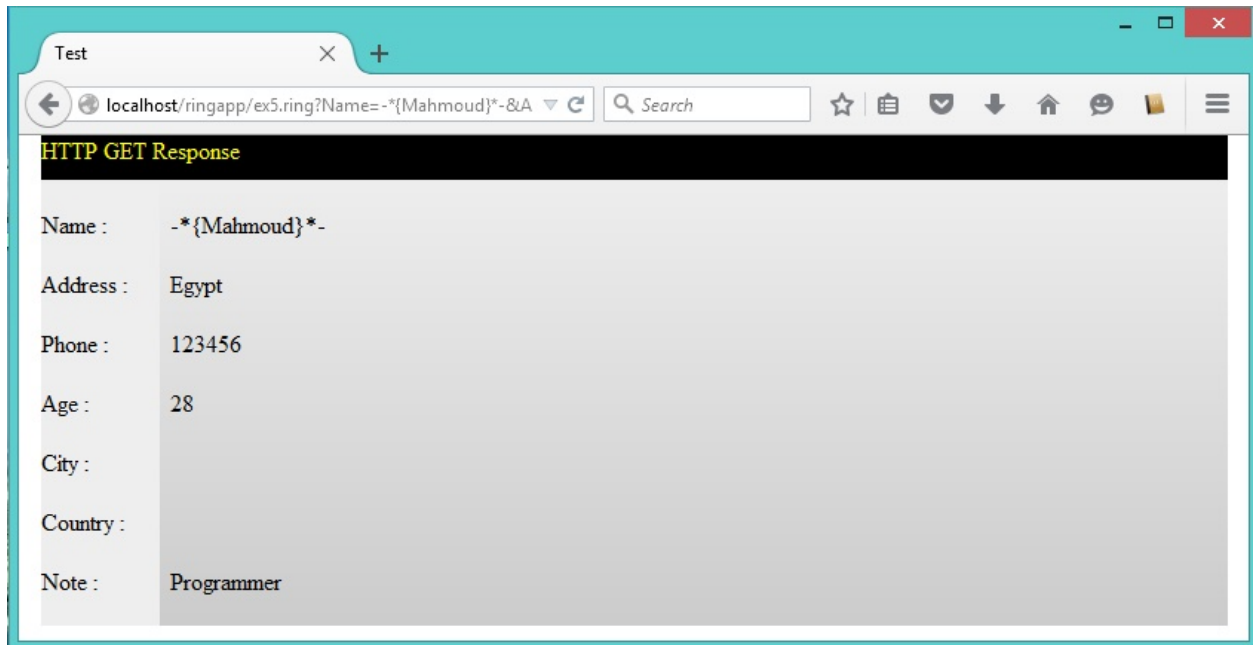
```
#!/c:\ring\bin\ring.exe -cgi
Load "weblib.ring"
Import System.Web

New Page
{
    boxstart()
        text( "URLEncode" )
        newline()
    boxend()
    link([ :url = "ex5.ring?Name="+URLEncode("-*{Mahmoud}*~")+
              "&Address=Egypt&Phone=123456&Age=28&Notes=Programmer",
          :title = "Test URL Encode" ])
}
```

Screen Shot:



Screen Shot:



48.10 Templates

Using Templates we can write Ring code inside HTML files

Syntax:

```
<%= Ring Expression %>
<% Ring Statements %>
```

The HTML Code

```
<h1>Listing Numbers</h1>
<table>
  <tr>
    <th> <%= myheader.cColumn1 %> </th>
    <th> <%= myheader.cColumn2 %> </th>
    <th></th>
    <th></th>
    <th></th>
  </tr>
  <% for x in aNumbers %>
    <tr>
      <td> <%= x.nValue %> </td>
      <td> <%= x.nSquare %> </td>
    </tr>
  <% next %>
</table>
```

The Ring Code

```
#!c:\ring\bin\ring.exe -cgi
Load "weblib.ring"
Import System.Web

New NumbersController { start() }
```

```
Class NumbersController

    MyHeader aNumbers

    Func Start

        MyHeader = New Header
        {
            cColumn1 = "Number" cColumn2 = "Square"
        }

        aNumbers = list(20)

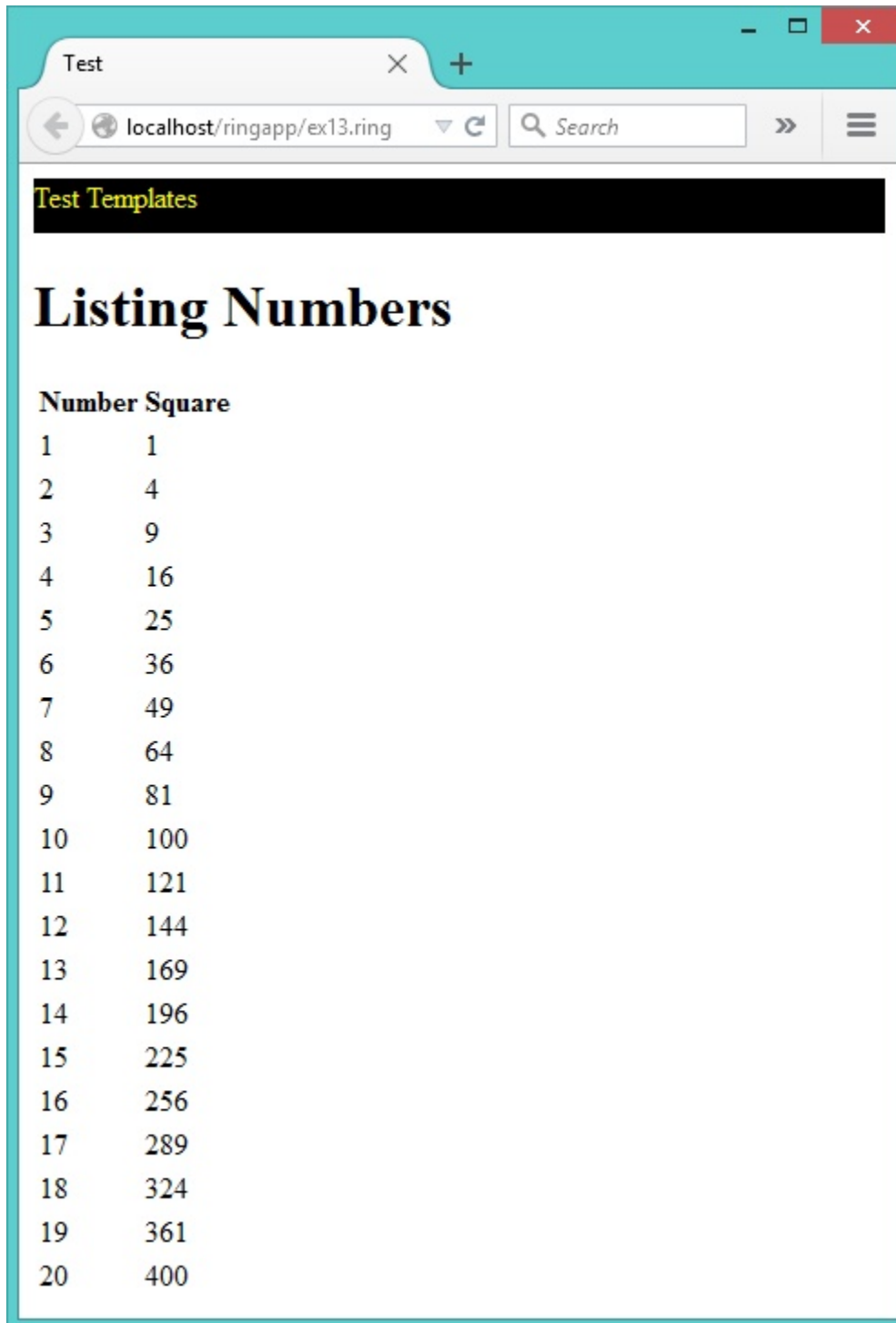
        for x = 1 to len(aNumbers)
            aNumbers[x] = new number
            {
                nValue = x    nSquare = x*x
            }
        next

        cTemp = Template("mynumbers.html",self)

    New Page
    {
        boxstart()
            text( "Test Templates" )
            newline()
        boxend()
        html(cTemp)
    }

Class Header cColumn1 cColumn2
Class Number nValue    nSquare
```

Screen Shot:



48.11 HTML Special Characters

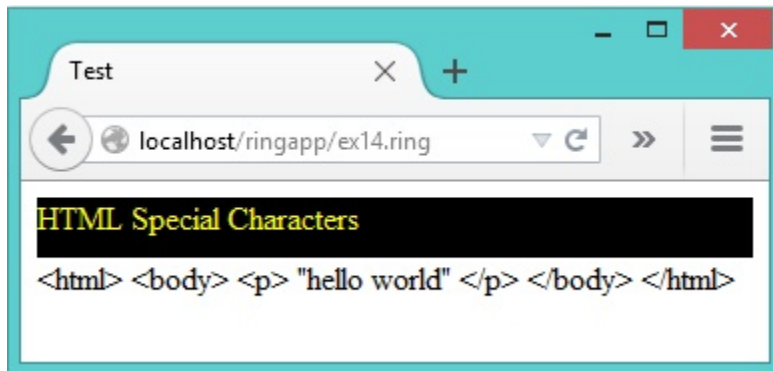
The `text()` function display HTML special characters.

If you want to write html code, use the `html()` function.

```
#!c:\ring\bin\ring.exe -cgi
Load "weblib.ring"
Import System.Web

New Page
{
    boxstart()
        text("HTML Special Characters")
        newline()
    boxend()
    text('
        <html>
            <body>
                <p> "hello world" </p>
            </body>
        </html>
    ')
}
```

Screen Shot:



48.12 Hash Functions

The Page User Interface

```
#!c:\ring\bin\ring.exe -cgi
Load "weblib.ring"
Import System.Web

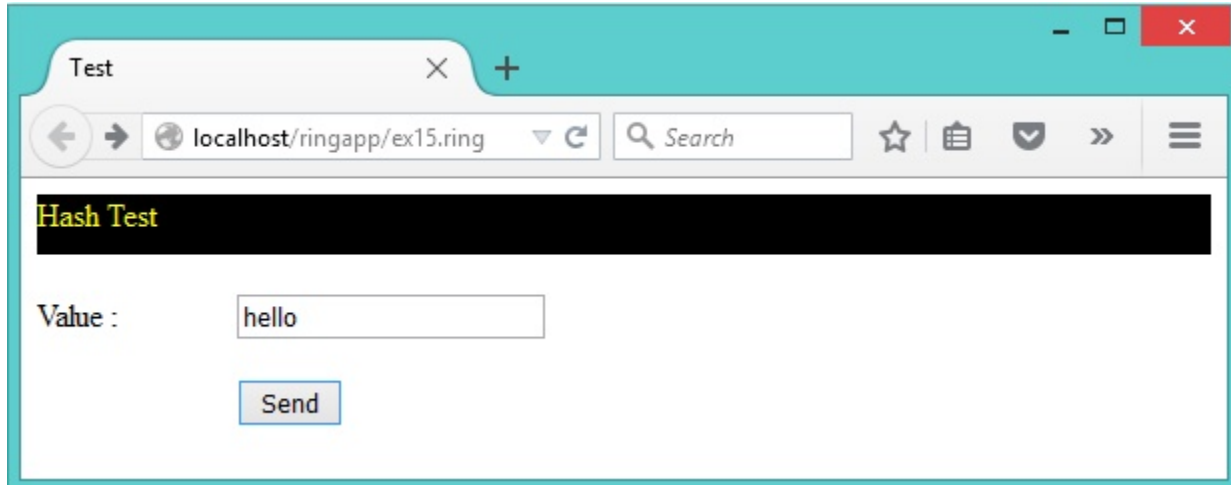
New Page
{
    boxstart()
        text("Hash Test")
        newline()
    boxend()
    divstart([ :style = StyleFloatLeft() + StyleWidth("100px") ])
        newline()
        text("Value : ")
        newline()
    divend()
    formpost("ex16.ring")
        divstart([ :style = StyleFloatLeft() + StyleWidth("300px") ])
            newline()
        }
```

```

        textbox([ :name = "Value" ])
        newline() newline()
        submit([ :value = "Send" ])
    divend()
formend()
}

```

Screen Shot:



The Response

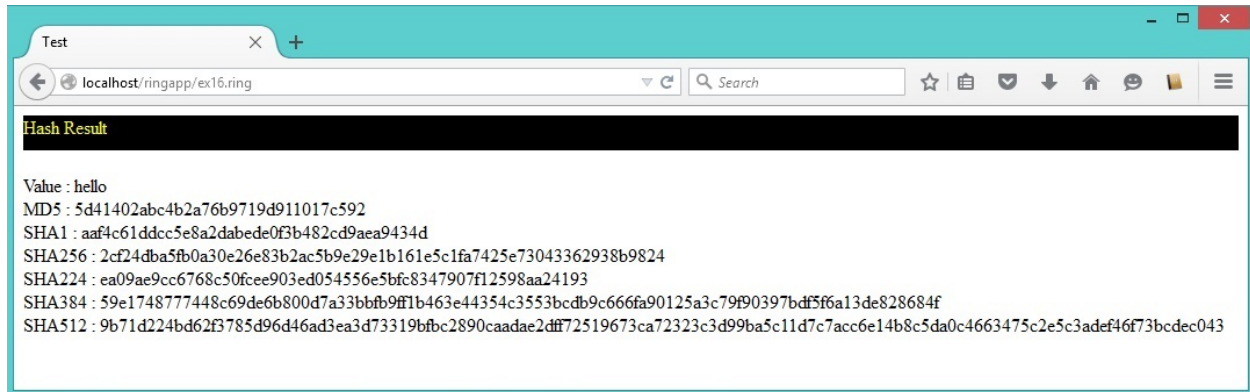
```

#!c:\ring\bin\ring.exe -cgi
Load "weblib.ring"
Import System.Web

New Page
{
    boxstart()
        text( "Hash Result" )
        newline()
    boxend()
    divstart([ :style = styleFloatLeft() + styleWidth("100%") ])
        newline()
        text( "Value : " + aPageVars["Value"] )
        newline()
        text( "MD5 : " + MD5(aPageVars["Value"]) )
        newline()
        text( "SHA1 : " + SHA1(aPageVars["Value"]) )
        newline()
        text( "SHA256 : " + SHA256(aPageVars["Value"]) )
        newline()
        text( "SHA224 : " + SHA224(aPageVars["Value"]) )
        newline()
        text( "SHA384 : " + SHA384(aPageVars["Value"]) )
        newline()
        text( "SHA512 : " + SHA512(aPageVars["Value"]) )
        newline()
    divend()
}

```

Screen Shot:



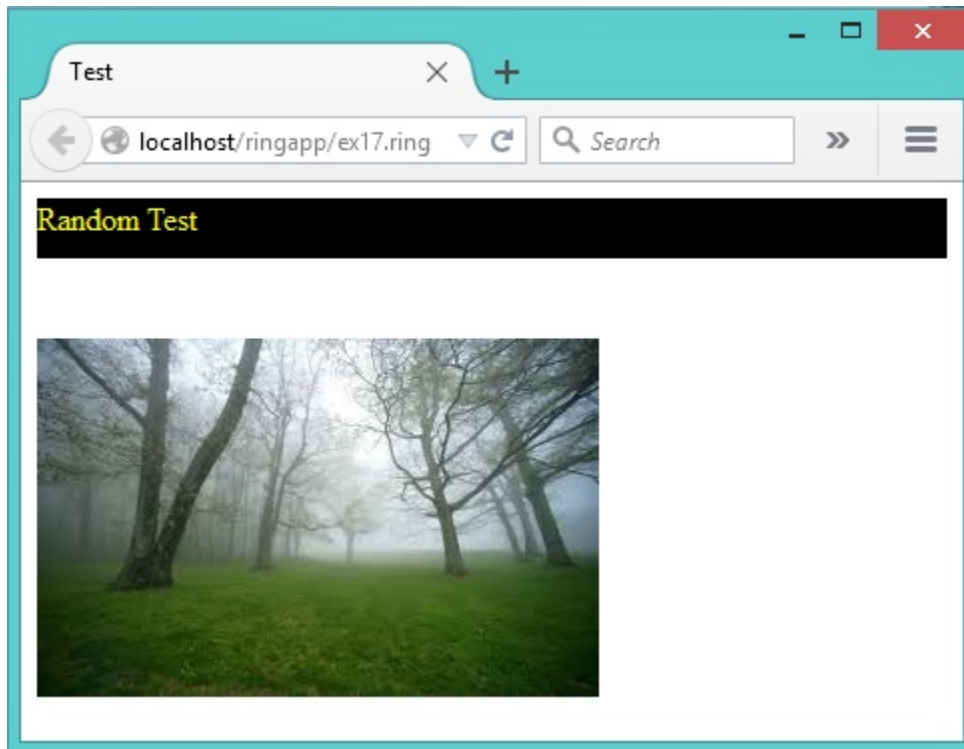
48.13 Random Image

```
#!c:\ring\bin\ring.exe -cgi
Load "weplib.ring"
Import System.Web

cUploadPath = "C:/Apache2.2/htdocs/ringapp/upload/"

New Page
{
    boxstart()
        text( "Random Test")
        newline()
    boxend()
    divstart([ :style = styleFloatLeft() + styleWidth("400px") ])
        newline()
        aList = dir(cUploadPath)
        if len(aList) > 0
            nIndex = random(len(aList))
            if nIndex = 0 nIndex = 1 ok
            cItem = "upload/" + aList[nIndex][1]
            newline()
            image( [ :url = cItem , :alt = :image ] )
        else
            text("No images!") newline()
        ok
    divid()
}
```

Screen Shot:



48.14 HTML Lists

The next example print a list contains numbers from 1 to 10

Then print a list from Ring List.

Finally we have a list of buttons and when we press on a button we get a message contains the clicked button number.

To start the list we uses the `ulstart()` function.

To end the list we uses the `ulend()` function.

We uses `liststart()` and `liend()` to determine the list item.

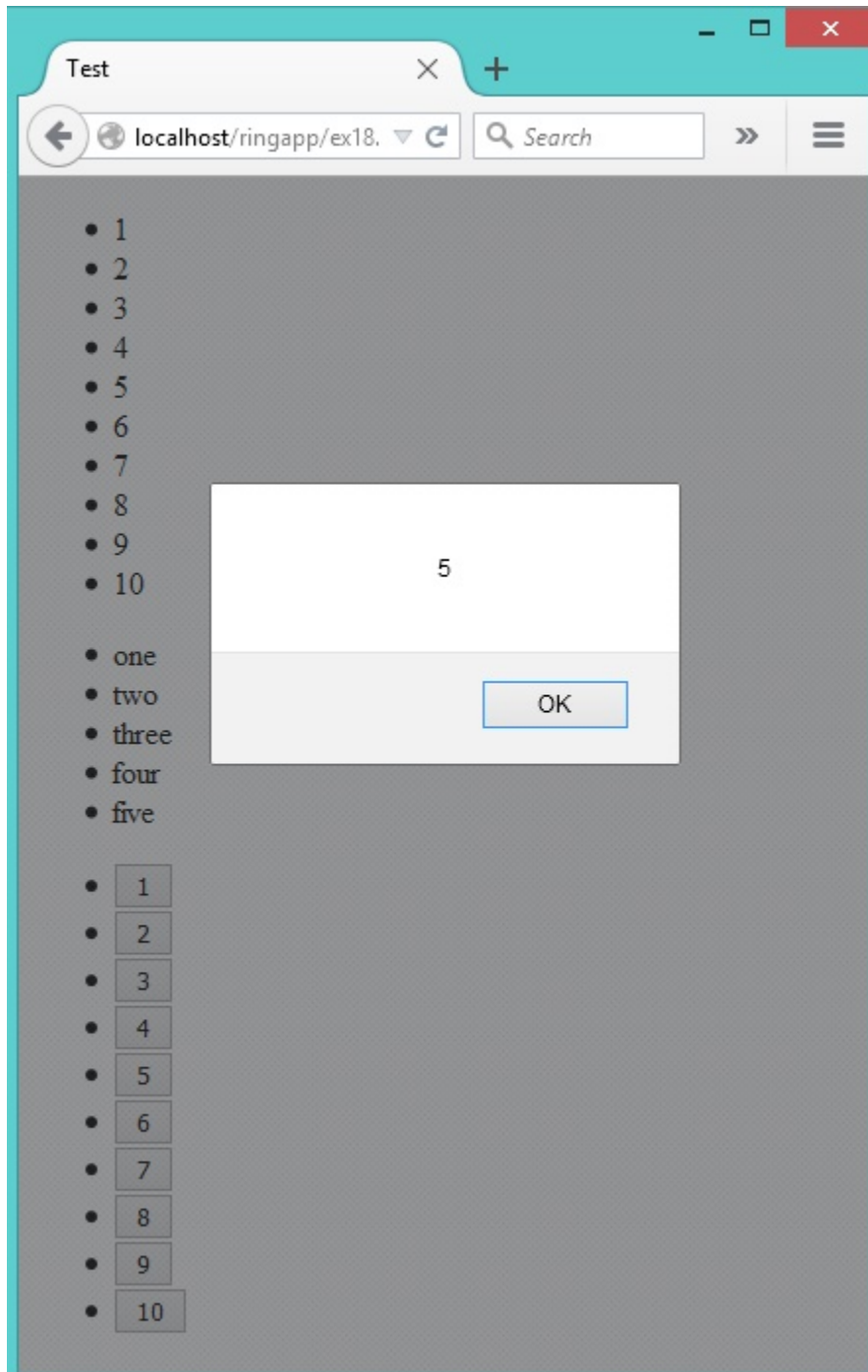
```
#!c:\ring\bin\ring.exe -cgi
Load "weblib.ring"
Import System.Web

Func Main
    New Page
    {
        ulstart([])
        for x = 1 to 10
            liststart([])
            text(x)
            liend()
        next
        ulend()
        list2ul(["one", "two", "three", "four", "five"])
        ulstart([])
        for x = 1 to 10
            liststart([])
```



```
        cFuncName = "btn"+x+"()"
        button([ :onclick = cFuncName , :value = x])
        script(scriptfunalert(cFuncName,string(x)))
    }
    liend()
    next
    ulend()
}
```

Screen Shot:



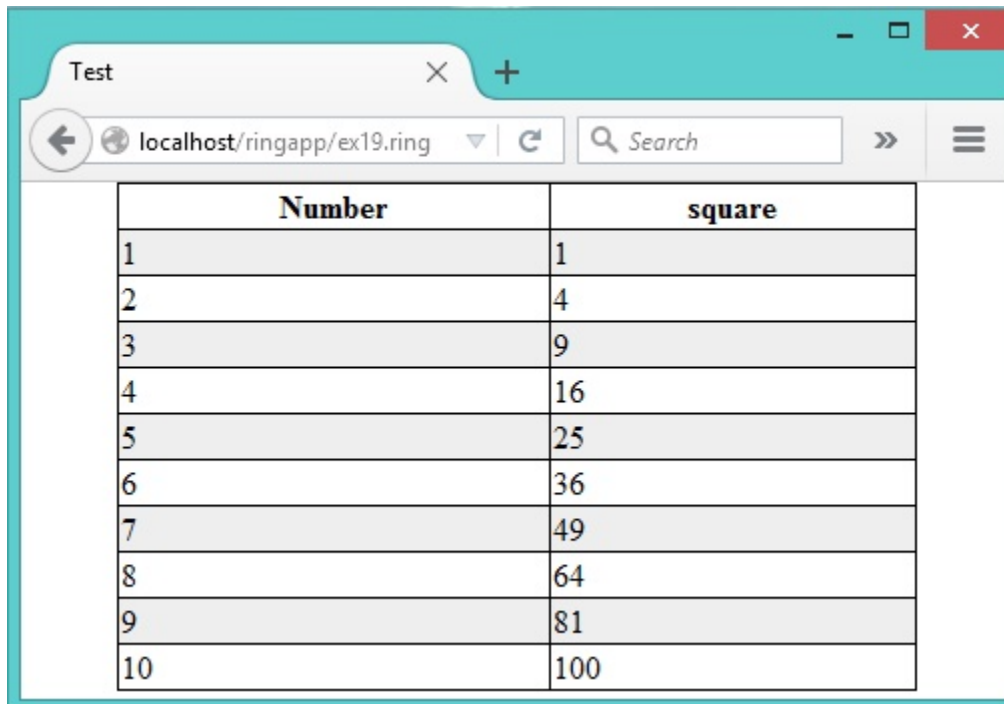
48.15 HTML Tables

In this example we will learn how to generate HTML tables using the `tablestart()`, `tableend()`, `rowstart()`, `rowend()`, `headerstart()`, `headerend()`, `cellstart()` and `cellend()` functions.

```
#!c:\ring\bin\ring.exe -cgi
Load "weblib.ring"
Import System.Web

Func Main
    New Page
    {
        divstart([ :style = styledivcenter("400px","500px") ])
        style(styletable() + styletablerows("t01"))
        tablestart([ :id = :t01 , :style = stylewidth("100%") ])
        rowstart([])
            headerstart([]) text("Number") headerend()
            headerstart([]) text("square") headerend()
        rowend()
        for x = 1 to 10
            rowstart([])
                cellstart([]) text(x) cellend()
                cellstart([]) text(x*x) cellend()
            rowend()
        next
        tableend()
    }
    divend()
```

Screen Shot:



Number	square
1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64
9	81
10	100

48.16 Gradient

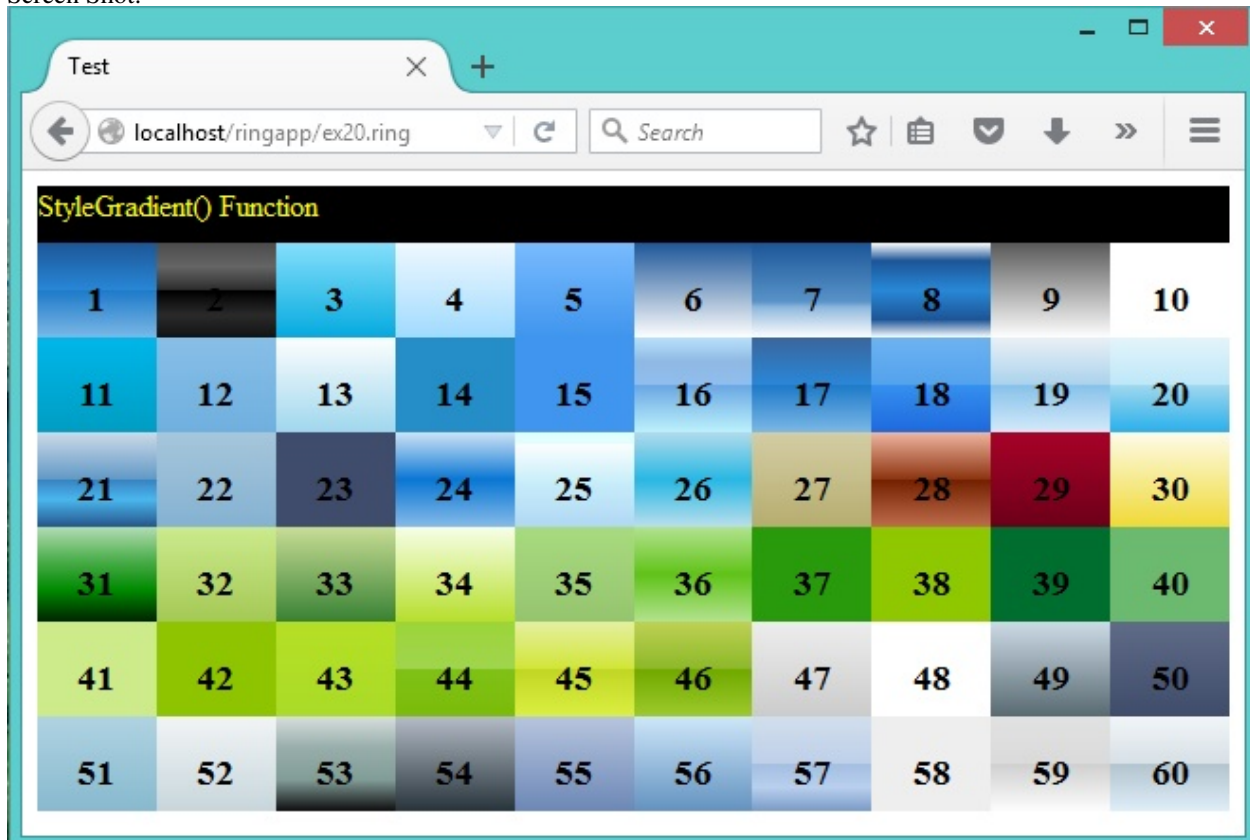
In this example we will learn how to use the `StyleGradient()` function.

The function takes the style number as input (range from 1 to 60).

```
#!c:\ring\bin\ring.exe -cgi
Load "weblib.ring"
Import System.Web

Func Main
    New Page
    {
        boxstart()
            text("StyleGradient() Function")
        boxend()
        for x = 1 to 60
            divstart([ :id = x , :align = "center" ,
                :style = stylefloatleft() +
                    stylesize(string(100/60*6)+"%", "50px") +
                    stylegradient(x) ])
                h3(x)
            divend()
        next
    }
}
```

Screen Shot:



48.17 Generating Pages using Objects

Instead of using functions/methods to generate HTML pages, we can use an object for each element in the page.

This choice means more beautiful code but slower.

The fastest method is to print HTML code directly, then using functions then using templates then using objects (slower).

```
#!c:\ring\bin\ring.exe -cgi
Load "weblib.ring"
Import System.Web

Func Main

WebPage()
{
    Title = "Using objects to create the Web Page content"
    h1 { text("welcome") }
    link
    {
        Title = "Google"
        Link = "http://www.google.com"
    }
    div
    {
        id = "div1"
        style = stylegradient(30) + stylesize("50%", "50%")
        text("Outer Div")
        div
        {
            id = "div2"
            color = "white"
            backgroundcolor = "green"
            width = "50%"
            height = "50%"
            marginleft = "5%"
            margintop = "5%"
            text("Inner Div")
        }
    }
    div
    {
        id = "div3"
        color = "black"
        backgroundcolor = "silver"
        width = "100%"
        height = "100%"
        text("Form")
        form
        {
            method = "POST"
            Action = "helloworld.ring"
            Table
            {
                style = stylewidth("100%") + stylegradient(24)
                TR
                {
                    TD { WIDTH="10%" text("Name : " ) }

```

```

        TD { Input { type = "text" } }
    }
    TR
    {
        TD { WIDTH="10%" text("Email : " ) }
        TD { Input { type = "text" } }
    }
    TR
    {
        TD { WIDTH="10%" text("Password : " ) }
        TD { Input { type = "password" } }
    }
    TR
    {
        TD { WIDTH="10%" text("Notes") }
        TD { TextArea { width="100%" rows = 10 cols = 10
            text("type text here...") } }
    }
    TR
    {
        TD { WIDTH="10%" text("Gender") }
        TD {
            select
            {
                width = "100%"
                option { text("Male") }
                option { text("Female") }
            }
        }
    }
    TR
    {
        TD { WIDTH="10%" text("Role") }
        TD
        {
            select
            {
                multiple = "multiple"
                width = "100%"
                option { text("student") }
                option { text("admin") }
            }
        }
    }
    }
    Input { type = "submit" value = "send" }
    Image { src="upload/profile1.jpg" alt="profile"}
    Input { type = "checkbox" value = "Old Member" } text("old member")
    Input { type = "range" min=1 max=100}
    Input { type = "number" min=1 max=100}
    Input { type = "radio" color="black" name="one"
        value = "one" } text("one")
    }
}
div
{
    color = "white"

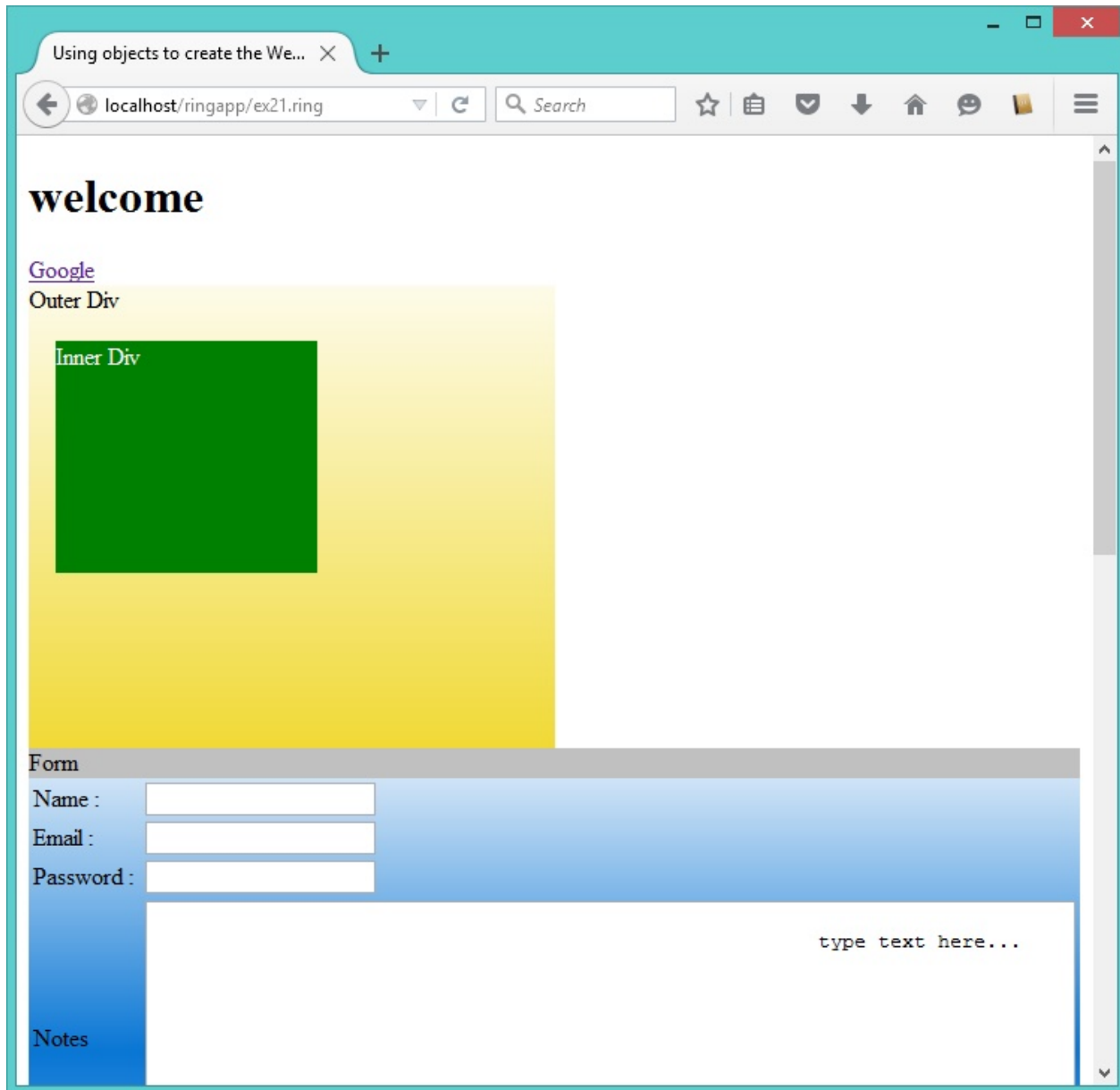
```

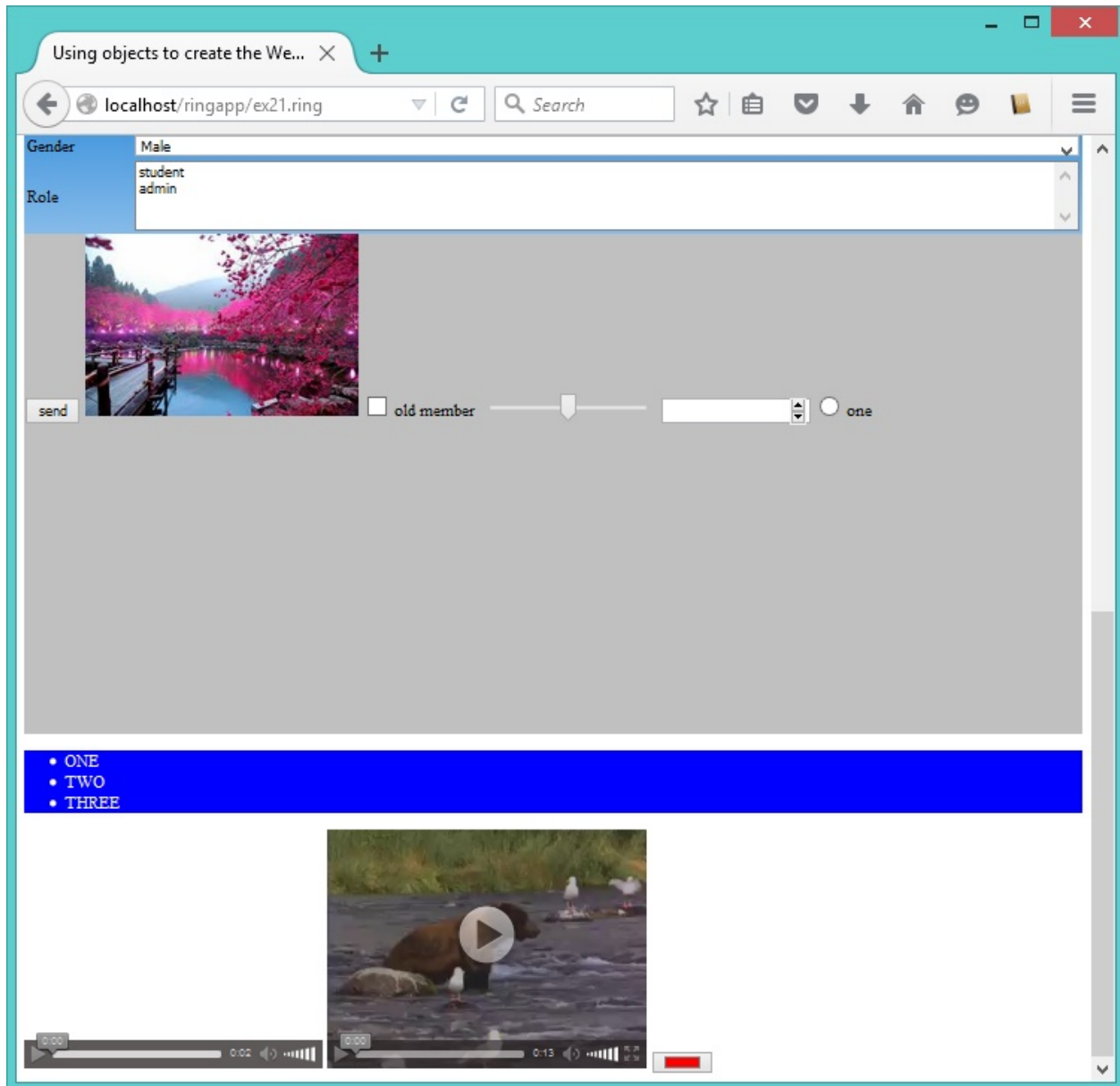
```
        backgroundColor = "blue"
        width = "100%"
        UL
        {
            LI { TEXT("ONE") }
            LI { TEXT("TWO") }
            LI { TEXT("THREE") }
        }
    }
    div
    {
        audio
        {
            src = "horse.ogg"
            type = "audio/ogg"
        }

        video
        {
            width = 320
            height = 240
            src = "movie.mp4"
            type = "video/mp4"
        }

        Input
        {
            type = "color"
            value = "#ff0000"
            onchange = "clickColor(0, -1, -1, 5)"
        }
    }
}
```

Screen Shot:





48.18 HtmlPage Class

Using this class we can create HTML documents without printing the output to the standard output

So instead of using the WebLib in Web Applications only

We can use it in Console/GUI/Mobile Applications too

Example:

```
load "stdlib.ring"
load "weblib.ring"

import System.Web
```



```

func main

  mypage = new HtmlPage {
    h1 { text("Customers Report") }
    Table
    {
      style = stylewidth("100%") + stylegradient(4)
      TR
      {
        TD { WIDTH="10%" text("Customers Count : " ) }
        TD { text (100) }
      }
    }
    Table
    {
      style = stylewidth("100%") + stylegradient(26)
      TR
      {
        style = stylewidth("100%") + stylegradient(24)
        TD { text("Name " ) }
        TD { text("Age" ) }
        TD { text("Country" ) }
        TD { text("Job" ) }
        TD { text("Company" ) }
      }
      for x = 1 to 100
        TR
        {
          TD { text("Test" ) }
          TD { text("30" ) }
          TD { text("Egypt" ) }
          TD { text("Sales" ) }
          TD { text("Future" ) }
        }
      next
    }
  }

  write("report.html",mypage.output())

```

48.19 Using Bootstrap Library using Functions

The next example uses the Bootstrap JavaScript Library when generating the HTML page.

```

#!c:\ring\bin\ring.exe -cgi
Load "weblib.ring"
Import System.Web

Func Main
  new BootstrapPage {
    divstart([ :class = "container" ])
    divstart([ :class = "jumbotron" ])
    h1("Bootstrap Page")
    divend()
  }

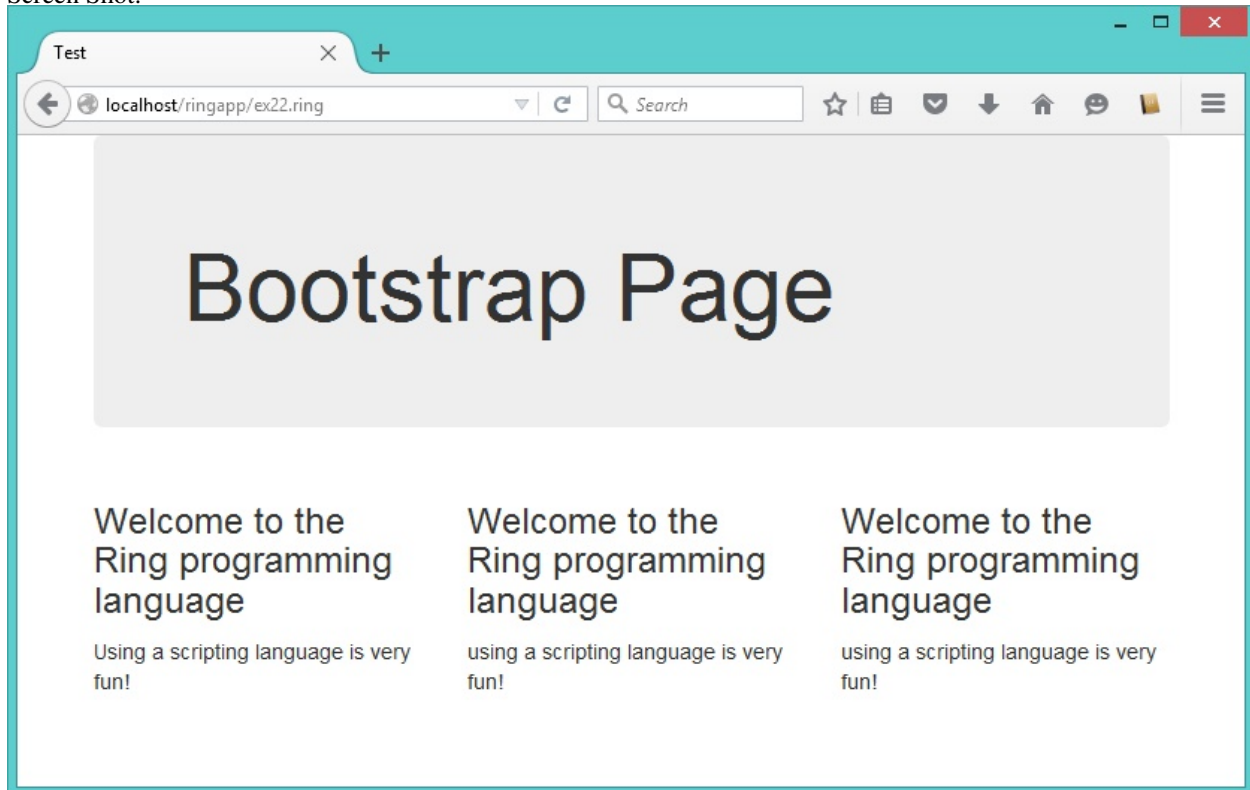
```

```

divstart([ :class = :row ])
  divstart([ :class = "col-sm-4" ])
    h3("Welcome to the Ring programming language")
    p([ :text = "Using a scripting language is very fun!" ])
  divend()
  divstart([ :class = "col-sm-4" ])
    h3("Welcome to the Ring programming language")
    p([ :text = "using a scripting language is very fun!" ])
  divend()
  divstart([ :class = "col-sm-4" ])
    h3("Welcome to the Ring programming language")
    p([ :text = "using a scripting language is very fun!" ])
  divend()
divend()
divend()
}

```

Screen Shot:



48.20 Using Bootstrap Library using Objects

The next example uses the Bootstrap JavaScript Library when generating the HTML page.

Instead of using functions to generate the HTML elements, we will use objects.

```

#!c:\ring\bin\ring.exe -cgi
Load "weblib.ring"
Import System.Web

Func Main

```

```

BootstrapWebPage()
{
  div
  {
    classname = :container
    div
    {
      classname = :jumbotron
      H1 { text("Bootstrap Page") }
    }
    div
    {
      classname = :row
      for x = 1 to 3
      div
      {
        classname = "col-sm-4"
        H3 { html("Welcome to the Ring programming language") }
        P { html("Using a scripting language is very fun!") }
      }
      next
    }
    div
    {
      classname = :row
      div
      {
        classname = "col-sm-4"
        Button
        {
          classname = "btn btn-info btn-lg"
          datatoggle= "modal"
          datatarget = "#myModal"
          text("Open Large Modal")
        }
      }
      div
      {
        classname = "col-sm-4"
        Button { classname = "btn btn-default btn-lg" text("default") }
        Button { classname = "btn btn-primary btn-md" text("primary") }
        Button { classname = "btn btn-sucess btn-sm" text("sucess") }
        Button { classname = "btn btn-info btn-xs" text("info") }
        Button { classname = "btn btn-warning" text("warning") }
        Button { classname = "btn btn-danger" text("danger") }
        Button { classname = "btn btn-link" text("link") }
      }
      div
      {
        classname = "col-sm-4"
        Button { classname = "btn btn-default btn-block" text("default") }
        Button { classname = "btn btn-primary btn-block" text("primary") }
        Button { classname = "btn btn-sucess btn-block" text("sucess") }
        Button { classname = "btn btn-info btn-block" text("info") }
        Button { classname = "btn btn-warning btn-block" text("warning") }
        Button { classname = "btn btn-danger btn-block" text("danger") }
        Button { classname = "btn btn-link btn-block" text("link") }
      }
    }
  }
}

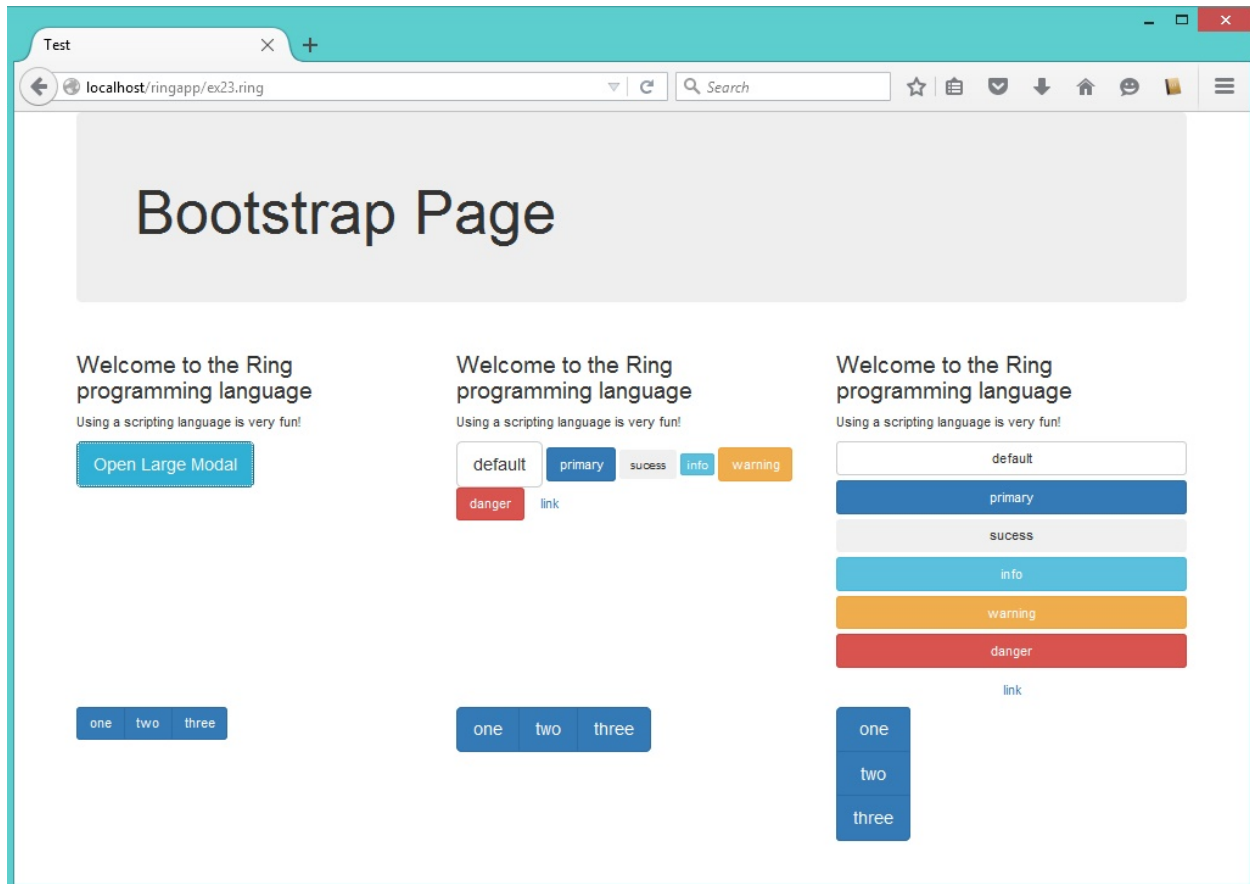
```

```

div
{
  classname = "col-sm-4"
  div { classname = "btn-group"
    button { classname="btn btn-primary" text("one") }
    button { classname="btn btn-primary" text("two") }
    button { classname="btn btn-primary" text("three") }
  }
}
div
{
  classname = "col-sm-4"
  div { classname = "btn-group btn-group-lg"
    button { classname="btn btn-primary" text("one") }
    button { classname="btn btn-primary" text("two") }
    button { classname="btn btn-primary" text("three") }
  }
}
div
{
  classname = "col-sm-4"
  div {
    classname = "btn-group-vertical btn-group-lg"
    button { classname="btn btn-primary" text("one") }
    button { classname="btn btn-primary" text("two") }
    button { classname="btn btn-primary" text("three") }
  }
}
}
div { classname="modal fade" id="myModal" role="dialog"
  div { classname = "modal-dialog modal-lg"
    div { classname="modal-content"
      div { classname="modal-header"
        button { classname="close" datadismiss="modal"
          html("&times;")
        }
        h4 { classname="modal-title"
          text("Modal Header")
        }
      }
      div { classname = "modal-body"
        p { text("This is a large model.") }
      }
      div { classname="modal-footer"
        button { classname = "btn btn-default" datadismiss="modal"
          text("close")
        }
      }
    }
  }
}
}
}

```

Screen Shot:



48.21 CRUD Example using MVC

The next example uses the weblib.ring & datalib.ring.

The datalib.ring contains classes for creating database applications using MVC pattern.

In this example we create an object from the SalaryController class then call the Routing method.

We define the website variable to contains the basic url of the page.

When we create the SalaryModel class from the ModelBase class, the salary table will be opened and the columns data will be defined as attributes in the model class.

The SalaryView class create an object from the SalaryLanguageEnglish class to be used for translation.

The method AddFuncScript is used to call the form for adding/modifying record data.

The method FormViewContent is used to determine the controls in the form when we add or modify a record.

```
#!c:\ring\bin\ring.exe -cgi
Load "weblib.ring"
Load "datalib.ring"
Import System.Web

website = "ex24.ring"

New SalaryController { Routing() }
```

```

Class SalaryModel from ModelBase

Class SalaryController From ControllerBase

Class SalaryView From ViewBase

    oLanguage = new SalaryLanguageEnglish

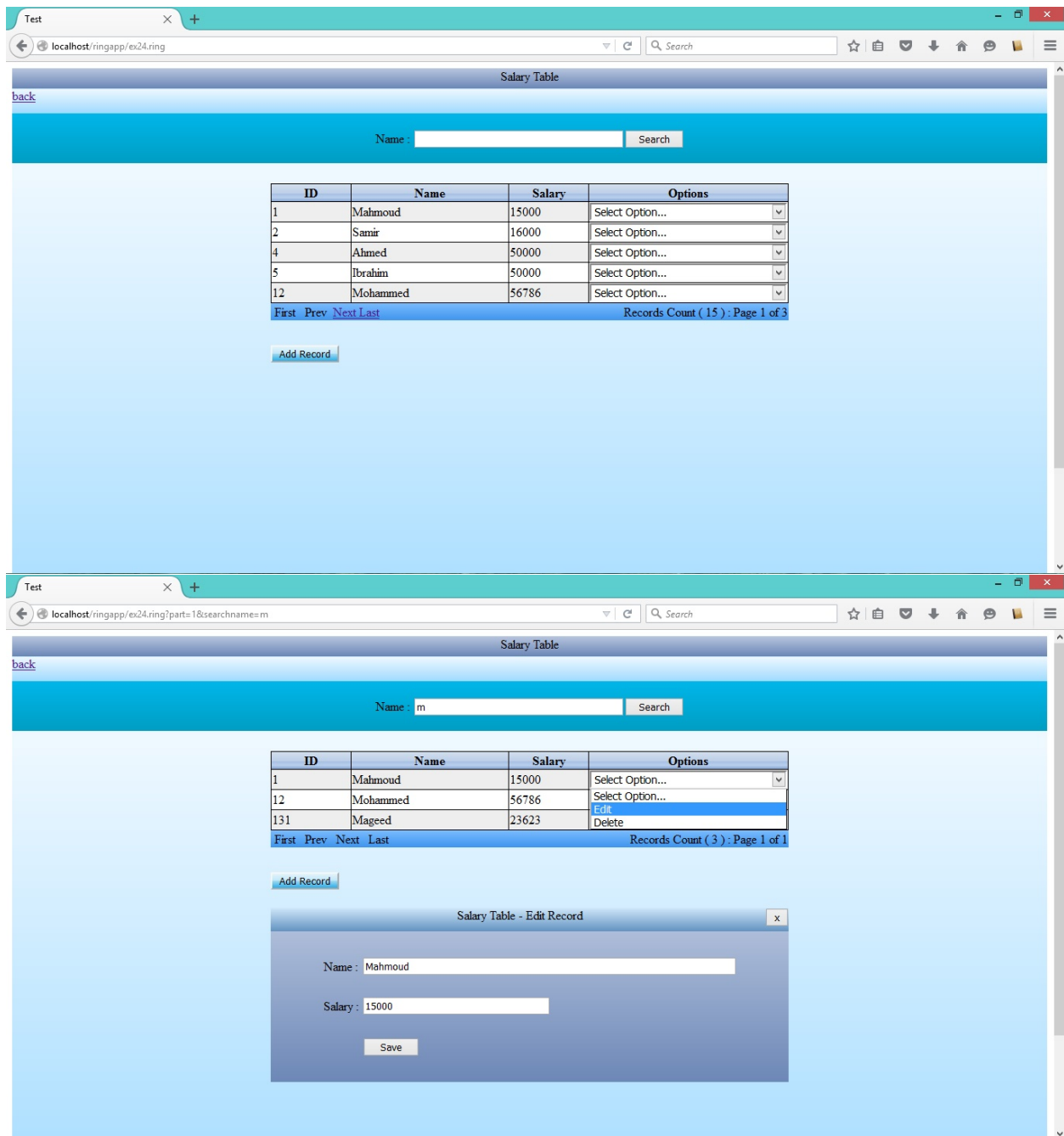
Func AddFuncScript oPage,oController
    return oPage.scriptfuncajax("myadd",oController.cMainURL+
        oController.cOperation+"=add","mysubpage")

Func FormViewContent oController,oTranslation,oPage
    return [
        [ oTranslation.aColumnsTitles[2], "textbox", "name",
          oController.oModel.Name, oPage.stylewidth("100%") ],
        [ oTranslation.aColumnsTitles[3], "textbox", "salary",
          oController.oModel.Salary, oPage.stylewidth("50%") ]
    ]

Class SalaryLanguageEnglish
    cTitle = "Salary Table"
    cBack = "back"
    aColumnsTitles = ["ID","Name","Salary"]
    cOptions = "Options"
    cSearch = "Search"
    comboitems = ["Select Option...", "Edit", "Delete"]
    cAddRecord = "Add Record"
    cEditRecord = "Edit Record"
    cRecordDeleted = "Record Deleted!"
    aMovePages = ["First","Prev","Next","Last"]
    cPage = "Page"
    cOf = "of"
    cRecordsCount = "Records Count"
    cSave = "Save"
    temp = new page
    cTextAlign = temp.StyleTextRight()
    cNoRecords = "No records!"

```

Screen Shot:



48.22 Users registration and Login

We have the users classes (Model, View & Controller) to deal with the users data like username & email.

The next code is stored in `ex25_users.ring`

```
Class UsersModel from ModelBase
    cSearchColumn = "username"

Class UsersController From ControllerBase
```

```

aColumnsNames = ["id","username","email"]

Func UpdateRecord
    oModel.id = aPageVars[cRecID]
    oModel.updatecolumn("username", aPageVars[:username] )
    oModel.updatecolumn("email", aPageVars[:email] )
    oView.UpdateView(self)

Class UsersView from ViewBase

    oLanguage = new UsersLanguageEnglish

    Func AddFuncScript oPage,oController
        return oPage.scriptfunc("myadd",oPage.scriptredirection("ex26.ring"))

    Func FormViewContent oController,oTranslation,oPage
        return [
            [oTranslation.aColumnsTitles[2],"textbox","username",
             oController.oModel.UserName,oPage.stylewidth("100%")],
            [oTranslation.aColumnsTitles[3],"textbox","email",
             oController.oModel.Email,oPage.stylewidth("50%")]
        ]

Class UsersLanguageEnglish
    cTitle = "Users Table"
    cBack = "back"
    aColumnsTitles = ["ID","User Name","Email"]
    cOptions = "Options"
    cSearch = "Search"
    comboitems = ["Select Option...", "Edit", "Delete"]
    cAddRecord = "Add Record"
    cEditRecord = "Edit Record"
    cRecordDeleted = "Record Deleted!"
    aMovePages = ["First", "Prev", "Next", "Last"]
    cPage = "Page"
    cOf = "of"
    cRecordsCount = "Records Count"
    cSave = "Save"
    temp = new page
    cTextAlign = temp.StyleTextRight()
    cNoRecords = "No records!"

```

In the file ex25.ring we load ex25_users.ring then create an object from UsersController class.

Using the created object, we call the routing method.

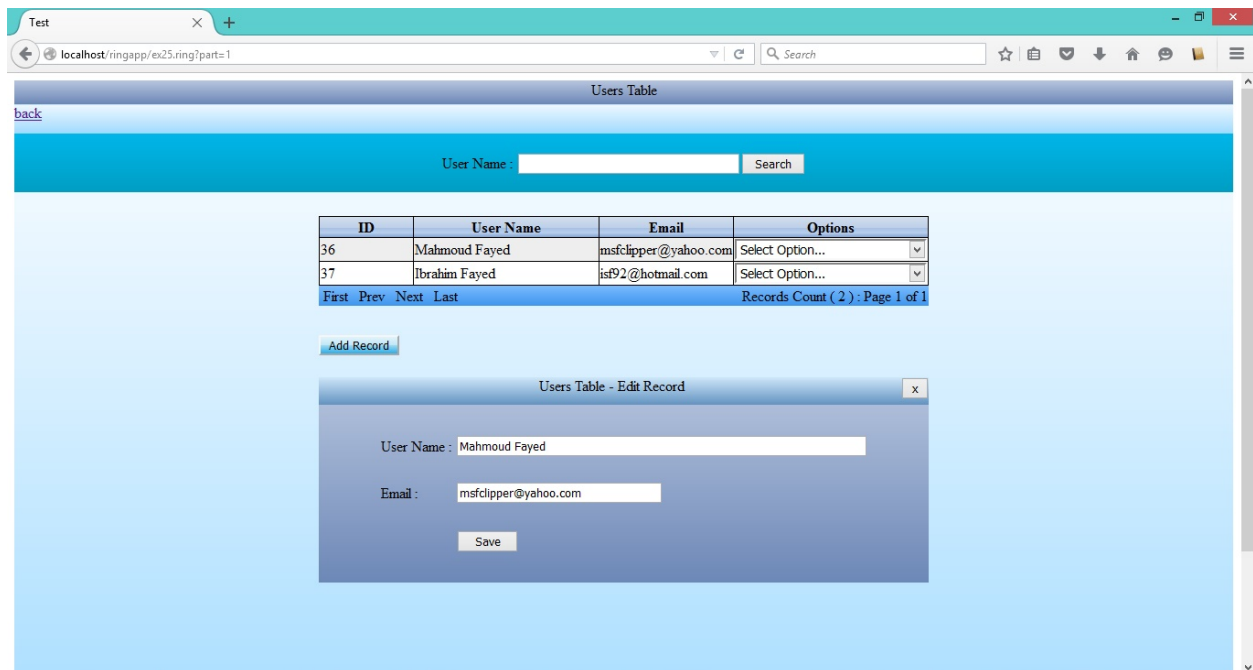
```

#!c:\ring\bin\ring.exe -cgi
Load "weblib.ring"
Load "datalib.ring"
Load "ex25_users.ring"

Import System.Web
website = "ex25.ring"
New UsersController { Routing() }

```

Screen Shot:



See the next code for the registration page

```
#!/c:/ring/bin/ring.exe -cgi
Load "weblib.ring"
Load "datalib.ring"
Import System.Web

website = "ex26.ring"

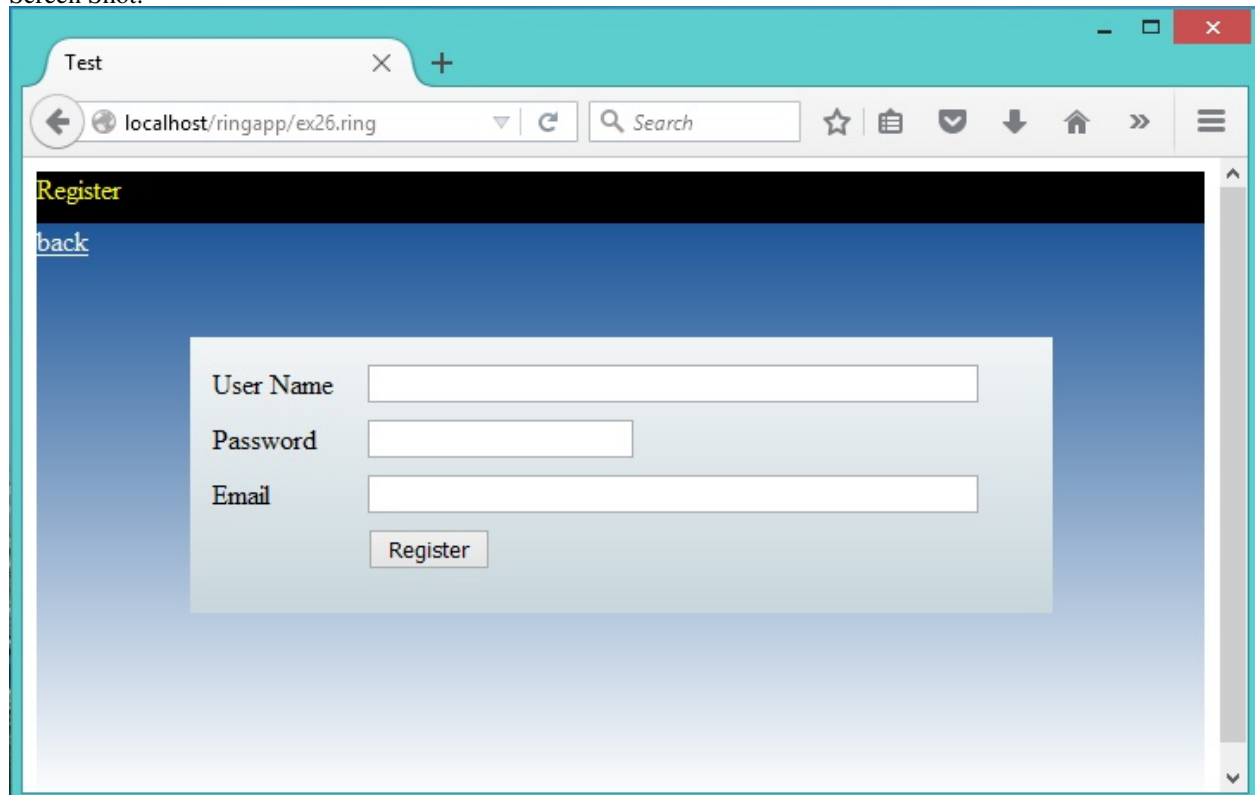
new page {
    boxstart()
        text( "Register")
        newline()
    boxend()
    divstart([:style = stylegradient(6) + stylesize("100%", "95%") ])
    link([:url = website, :title = "back" , :style = stylecolor("white")])
    newline()
    divstart([:style= styledivcenter("500", "160") + stylegradient(52) ])
    formpost("ex27.ring")
        tablestart([:Style = stylemarginleft("2%") + stylemargintop("2%") +
            stylewidth("90%") ])
            rowstart([])
                cellstart([:style = stylewidth("20%") + styleheight(30)])
                    text("User Name")
                cellend()
                cellstart([:style = stylewidth("80%") ])
                    textbox([:name = "username", :style = stylewidth("100%")])
                cellend()
            rowend()
            rowstart([])
                cellstart([:Style = styleheight(30)])
                    text("Password")
                cellend()
                cellstart([])
                    textbox([:name = "password" , :type = "password"])
```

```

        cellend()
    rowend()
    rowstart([])
        cellstart([ :style = styleheight(30)])
            text("Email")
        cellend()
        cellstart([])
            textbox([:name = "email" , :style = stylewidth("100%")])
        cellend()
    rowend()
    rowstart([])
        cellstart([ :style = styleheight(30)])
        cellend()
        cellstart([ :style = styleheight(30)])
            submit([:value = "Register" ])
        cellend()
    rowend()
tableend()
formend()
divend()
divend()
}

```

Screen Shot:



The Registration response

```

#!c:\ring\bin\ring.exe -cgi
Load "weblib.ring"
Load "datalib.ring"
Load "ex25_users.ring"

```

```

Import System.Web

oUser = new UsersModel
oUser.Connect()
if oUser.findwith("username",aPageVars["username"])
    new page {
        text("The user name is already registered")
    }
    return
ok
if oUser.findwith("email",aPageVars["email"])
    new page {
        text("This email is already registered")
    }
    return
ok

aPageVars["salt"] = str2hex(RandBytes(32))
aPageVars["pwhash"] = sha256(aPagevars["password"]+aPageVars["salt"])
aPageVars["sessionid"] = str2hex(randbytes(32))
oUser.Insert()
new page {
    cookie("sessionid",aPageVars["sessionid"])
    text("New User Created!")
    newline()
    text("User Name : " + aPageVars["username"])
    newline()
}
oUser.Disconnect()

```

See the next code for the Login page

```

#!c:\ring\bin\ring.exe -cgi
Load "weblib.ring"
Load "datalib.ring"

Import System.Web

website = "ex28.ring"

new page {
    boxstart()
        text( "Login")
        newline()
    boxend()
    divstart([:style = stylegradient(6) + stylesize("100%","95%") ])
    link([ :url = website, :title = "back" , :style = stylecolor("white")])
    newline()
    divstart([ :style= styledivcenter("500","130") + stylegradient(52) ])
    formpost("ex29.ring")
        tablestart([ :Style = stylemarginleft("2%") + stylemargintop("2%") +
            stylewidth("90%") ])
            rowstart([])
                cellstart([:style = stylewidth("20%") + styleheight(30)])
                    text("User Name")
                cellend()
                cellstart([ :style = stylewidth("80%") ])
                    textbox([:name = "username", :style = stylewidth("100%")])
                cellend()

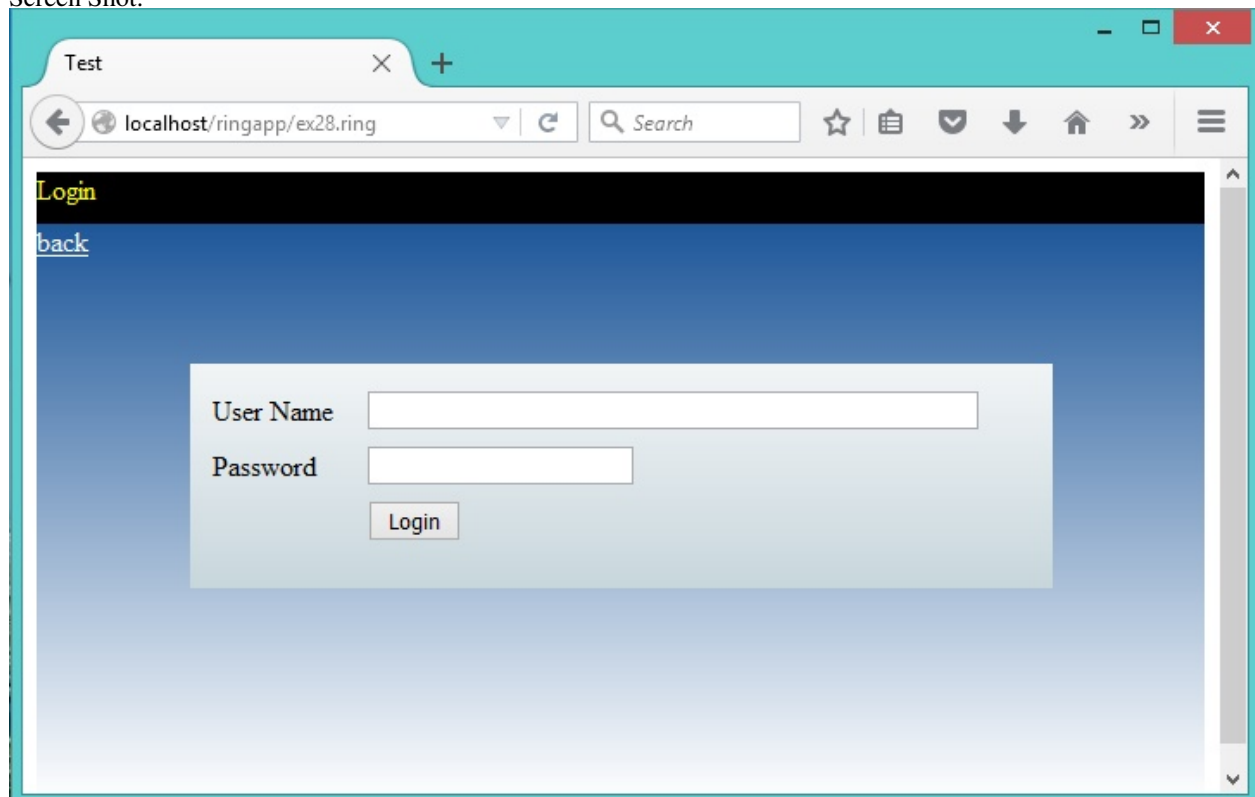
```

```

        rowend()
        rowstart([])
            cellstart([ :style = styleheight(30)])
                text("Password")
            cellend()
            cellstart([])
                textbox([:name = "password" , :type = "password"])
            cellend()
        rowend()
        rowstart([])
            cellstart([ :style = styleheight(30) ])
            cellend()
            cellstart([])
                submit([:value = "Login" ])
            cellend()
        rowend()
    tableend()
formend()
divend()
divend()
}

```

Screen Shot:



The response page

```

#!c:\ring\bin\ring.exe -cgi
Load "weblib.ring"
Load "datalib.ring"
Load "ex25_users.ring"

Import System.Web

```

```

oUser = new UsersModel
oUser.Connect()
lResult = oUser.FindWith("username",aPageVars["username"])
new page {
    if lResult
        if sha256(aPageVars["password"]+oUser.Salt) = oUser.pwhash
            text ("Correct Password!")
            aPageVars["sessionid"] = str2hex(randbytes(32))
            oUser.UpdateColumn("sessionid",aPageVars["sessionid"])
            cookie("sessionid",aPageVars["sessionid"])
        else
            text ("Bad password!")
    else
        text("Bad User Name!")
    ok
}
oUser.Disconnect()

```

The next code for checking if the user needs to login or not

```

#!c:\ring\bin\ring.exe -cgi
Load "weblib.ring"
Load "datalib.ring"
Load "ex25_users.ring"

Import System.Web

oUser = new UsersModel
oUser.Connect()
lResult = oUser.FindWith("sessionid",aPageVars["sessionid"])
new page {
    if lResult
        text("User Name : " + oUser.username )
    else
        text("Please Login First!")
    ok
}
oUser.Disconnect()

```

48.23 Database, ModelBase & ControllerBase classes

In this section we will see some code from datalib.ring

The next code presents the Database, ModelBase & ControllerBase classes

```

Import System.Web

Class Database

    cServer = "localhost"
    cUserName = "root"
    cPassword = "root"
    cDatabase = "mahdb"

    Func Connect

```

```

        con = mysql_init()
        mysql_connect(con, cServer, cUserName, cPassWord,cDatabase)

Func Disconnect

        mysql_close(con)

Func Query cQuery

        mysql_query(con,cQuery)

Func QueryResult

        return mysql_result(con)

Func QueryResultWithColumns
        # return columns names + query result
        return mysql_result2(con)

Func QueryValue
        aResult = mysql_result(con)
        if islist(aResult) and len(aResult) >= 1
            aResult = aResult[1]
            if len(aResult) >= 1
                return aResult[1]
            ok
        ok
        return 0

Func EscapeString x
        if isstring(x)
            return MySQL_Escape_String(con,x)
        else
            return MySQL_Escape_String(con,string(x))
        ok

Private
    con = NULL

Class ModelBase from Database

    cTableName = ""
    cSearchColumn = "name"
    aColumns = []
    aQueryResult = []
    ID = 0

    # set table name from class name
    classname = lower(classname(self))
    if right(classname,5) == :model
        cTablename = left(classname,len(classname)-5)
    ok

Func Insert

        cValues = ""
        for x in aColumns
            cValues += "'" + EscapeString(aPageVars[x]) + "',"

```

```

Next
cValues = left(cValues,len(cValues)-1)  # remove last comma
cColumns = ""
for x in aColumns
    cColumns += x + ","
next
cColumns = left(cColumns,len(cColumns)-1)
query("insert into " + cTableName + "("+cColumns+") values (" +
      cValues + ")")

Func Update nID

    cStr = ""
    for x in aColumns
        cStr += x + " = '" + EscapeString(aPageVars[x]) + "' , "
    # the space after comma is necessary
    Next
    cStr = left(cStr,len(cStr)-2)
    query("update " + cTableName + " set " + cStr + " where id = " + nID )

Func UpdateColumn cColumn,cValue
    query("update " + cTableName + " set " + cColumn + " = '" +
      EscapeString(cValue) + "' where id = " + self.ID )

Func Count cValue

    query("SELECT count(*) FROM " + cTableName +
          " where "+cSearchColumn+" like '" + EscapeString(cValue) + "%'")
    return queryValue()

Func Read nStart,nRecordsPerPage

    query("SELECT * FROM "+ cTableName+" limit " + EscapeString(nStart) + "," +
      EscapeString(nRecordsPerPage) )
    aQueryResult = queryResult()

Func Search cValue,nStart,nRecordsPerPage

    query("SELECT * FROM "+ cTableName+" where "+cSearchColumn+" like '" +
      EscapeString(cValue) + "%' " +
      " limit " + EscapeString(nStart) + "," + EscapeString(nRecordsPerPage) )
    aQueryResult = queryResult()

Func Find nID

    query("select * from " + cTableName + " where id = " + EscapeString(nID) )
    aResult = queryResult()[1]
    # move the result from the array to the object attributes
    ID = nID
    cCode = ""
    for x = 2 to len(aResult)
        cCode += aColumns[x-1] + " = hex2str('" + str2hex(aResult[x]) + "')" + nl
    next
    eval(cCode)

Func FindWith cColumn,cValue

```

```

query("select * from " + cTableName + " where "+cColumn+" = '" +
EscapeString(cValue) + "'" )
aResult = queryResult()
if len(aResult) > 0
    aResult = aResult[1]
else
    return 0
ok
# move the result from the array to the object attributes
ID = aResult[1]
cCode = ""
for x = 2 to len(aResult)
    cCode += aColumns[x-1] + " = hex2str('" + str2hex(aResult[x]) + "'" ) + nl
next
eval(cCode)
return 1

```

Func Delete ID

```

query("delete from " + cTableName + " where id = " + EscapeString(ID) )

```

Func Clear

```

cCode = ""
for x in aColumns
    cCode += x + ' = ""' + nl
next
eval(cCode)

```

Func LoadModel

```

# create the columns array
query("SELECT * FROM " + cTableName + " limit 0,1")
aQueryResult = QueryResultWithColumns()[1]
for x = 2 to len(aQueryResult)
    aColumns + lower(trim(aQueryResult[x]))
next

# create attribute for each column
for x in aColumns
    addattribute(self,x)
next

```

Func Connect

```

Super.Connect()
if nLoadModel = 0
    nLoadModel = 1
    LoadModel()
ok

```

private

```

nLoadModel = 0

```

Class ControllerBase


```

nRecordsPerPage = 5
nRecordsCount = 0
nPagesCount = 0
nActivePage = 0

# Dynamic creation of oView = new tablenameView and oModel = new tablename.Model
classname = lower(classname(self))
if right(classname,10) = :controller
    tablename = left(classname,len(classname)-10)
    cCode = "oView = new " + tablename+"View" + nl
    cCode += "oModel = new " + tablename+"Model" + nl
    eval(cCode)
    oModel.connect()
ok

cSearchName = "searchname"
cPart = "part"
cPageError = "The page number is not correct"
cLast = "last"
cOperation = "operation"
cRecID = "recid"

aColumnsNames = ["id"]
for t in oModel.aColumns
    aColumnsNames + t
next

cMainURL = website + "?"

func Routing

    switch      aPageVars[cOperation]
    on NULL      showtable()
    on :add      addrecord()
    on :save      saverecord()
    on :delete    deleterecord()
    on :edit      editrecord()
    on :update    updaterecord()
    off

func ShowTable

    nRecordsCount = oModel.Count( aPageVars[cSearchName] )

    nPagesCount = ceil(nRecordsCount / nRecordsPerPage)

    if aPageVars[cPart] = cLast
        aPageVars[cPart] = string(nPagesCount)
    ok

    nActivePage = number(aPageVars[cPart])
    if nActivePage = 0 nActivePage = 1 ok

    if ( nActivePage > nPagesCount ) and nRecordsCount > 0
        ErrorMsg(cPageError)
    return
    ok

```

```

nStart = (nActivePage-1)*nRecordsPerPage

if aPageVars[cSearchName] = NULL
    oModel.Read( nStart,nRecordsPerPage )
else
    oModel.Search( aPageVars[cSearchName],nStart,nRecordsPerPage )
ok

oView.GridView(self)

func AddRecord

    oModel.clear()
    oView.FormViewAdd(Self,:save,false) # false mean don't include record id

func SaveRecord

    oModel.Insert()
    oView.SaveView(self)

func EditRecord

    oModel.Find( aPageVars[cRecID] )
    oView.FormViewEdit(Self,:update,true) # true mean include record id

func UpdateRecord

    oModel.update( aPageVars[cRecID] )
    oView.UpdateView(self)

func DeleteRecord

    oModel.Delete( aPageVars[cRecID] )
    oView.DeleteView()

func braceend

    oModel.Disconnect()

```

48.24 WebLib API

In this section we will see the web library functions, classes and methods.

Function	Parameters	Description
LoadVars	None	Save the request parameters and cookies to aPageVars List
WebPage	None	Create new object from the WebPage Class
BootStrapWebPage	None	Create new object from the BootStrapWebPage Class
HTMLSpecialChars	cString	Encode Special characters to HTML equivalent
Template	cFile,oObject	Execute Ring Code in cFile after accessing oObject using { }
Alert	cMessage	Generate HTML Web Page that display cMessage using JavaScript Alert()
HTML2PDF	cString	Generate and Display PDF File from HTML String (cString)

The Package System.Web contains the next classes

Class Name	Description
Application	Contains methods for Encoding, Decoding, Cookies & More.
Page	Contains methods to generate HTML pages.
ScriptFunctions	Contains methods to generate some JavaScript Functions.
StyleFunctions	Contains methods to generate CSS.
PageBuffer	Generate HTML Page in memory (don't print the output).
HTML2PDF	Generate PDF File from HTML code.
BootstrapPage	Using Bootstrap Library.
WebPage	Generate page using objects for each element.
HtmlPage	Like WebPage but doesn't print the output to stdout.
BootstrapWebPage	Generate page using objects, using Bootstrap Library.
ObjBase	Parent Class for page objects.
NewObjectsFunctions	Methods to create new objects in the page or element.
H1	Wraps HTML H1.
H2	Wraps HTML H2.
H3	Wraps HTML H3.
H4	Wraps HTML H4.
H5	Wraps HTML H5.
H6	Wraps HTML H6.
P	Wraps HTML P.
Link	Wraps HTML link.
NewLine	Wraps HTML NewLine.
Div	Wraps HTML Div.
Form	Wraps HTML Form.
Input	Wraps HTML Input.
TextArea	Wraps HTML TextArea.
Select	Wraps HTML Select.
Option	Wraps HTML Option.
Image	Wraps HTML Image.
UL	Wraps HTML UL.
LI	Wraps HTML LI.
Table	Wraps HTML Table.
TR	Wraps HTML TR.
TD	Wraps HTML TD.
TH	Wraps HTML TH.
Audio	Wraps HTML Audio.
Video	Wraps HTML Video.
Nav	Wraps HTML Nav.
Span	Wraps HTML Span.
Button	Wraps HTML Button.

48.25 Application Class

Method	Parameters	Description
DecodeString	cString	Decode request parameters
Decode	cString	Decode multipart/form-data
GetFileName	aArray,cVar	Get File Name in aArray using cVar
SetCookie	name,value,expires,path,domain,secure	Set Cookie
Cookie	name,value	Set Cookie using name and value only
GetCookies	None	Get Cookies
URLEncode	cString	URL Encode
ScriptLibs	None	Add JavaScript Libraries like BootStrap
Print	None	Print Page Content
Style	cStyle	Add cStyle to page CSS content
StartHTML	None	Add HTTP Header to page content

The method DecodeString is used to get HTTP request parameters.

The methods Decode and GetFileName are used for uploading files.

The methods SetCookie, Cookie & GetCookies are used for adding and reading cookies.

The methods StartHTML, ScriptsLibs, Style & Print are used for page structure and JS/CSS support.

The method URLEncode is used to encode a URL to be used in HTML pages.

48.26 Page Class

Method	Parameters	Description
text	x	add HTMLSpecialChars(x) to page content (accept strings and numbers)
html	cString	add html code to page content
h1	x	add x to page content between <h1> and </h1>
h2	x	add x to page content between <h2> and </h2>
h3	x	add x to page content between <h3> and </h3>
h4	x	add x to page content between <h4> and </h4>
h5	x	add x to page content between <h5> and </h5>
h6	x	add x to page content between <h6> and </h6>
p	aPara	HTML <p> </p>, uses aPara List as Hash to get attributes
NewLine	None	add to page content
AddAttributes	aPara	Convert aPara list as hash to HTML element attributes
Link	aPara	HTML <a href> and , uses aPara List as Hash to get attributes
Image	aPara	HTML , uses aPara List as Hash to get attributes
Button	aPara	HTML <input type="button">, uses aPara List as Hash to get attributes
ButtonLink	aPara	HTML <input type="button">, uses link attribute to navigate to link
Textbox	aPara	HTML <input type="text">, uses aPara List as Hash to get attributes
Editbox	aPara	HTML <textarea> and </textarea>, uses aPara to get attributes
Combobox	aPara	HTML <select>, uses items attribute as list for <option>
Listbox	aPara	HTML <select multiple="multiple">, uses items attribute for <option>
ulstart	aPara	HTML
ulend	aPara	HTML
listart	aPara	HTML
liend	aPara	HTML

Continued on next page

Table 48.2 – continued from previous page

Method	Parameters	Description
List2UL	aList	Generate HTML including items from Ring List items
DivStart	aPara	HTML <div>, uses aPara List as Hash to get attributes
NavStart	aPara	HTML <nav>, uses aPara List as Hash to get attributes
SpanStart	aPara	HTML , uses aPara List as Hash to get attributes
BoxStart	None	Generate Div with black background to be used as page header
DivEnd	None	HTML </div>
NavEnd	None	HTML </nav>
SpanEnd	None	HTML
BoxEnd	None	HTML </div>, the same as divend()
FormStart	cAction	HTML <form>, with cAction as the action attribute or an empty value
FormPost	cAction	HTML <form method="post"> , with cAction as the action attribute
FormEnd	None	HTML </form>
Submit	aPara	HTML <input type="submit">
Hidden	cName,cValue	HTML <input type="hidden">
FormUpload	x	HTML Form, method="post" enctype="multipart/form-data" and x = action
UploadFile	x	HTML <input type="file"> and name = x
Video	aPara	HTML <video>
Audio	aPara	HTML <audio>
GetColor	aPara	Select Color
Radio	aPara	HTML <input type="radio">
Checkbox	aPara	HTML <input type="checkbox">
Spinner	aPara	HTML <input type="number">
Slider	aPara	HTML <input type="range">
TableStart	aPara	HTML <table>
TableEnd	None	HTML </table>
RowStart	aPara	HTML <tr>
RowEnd	None	HTML </tr>
CellStart	aPara	HTML <td>
CellEnd	None	HTML </td>
HeaderStart	aPara	HTML <th>
HeaderEnd	None	HTML </th>

aPara in the page methods is a list contains attributes and values. Using aPara we can set values for the next attributes

```

classname id name align style dir value onclick oncontextmenu ondblclick
onmousedown onmouseenter onmouseleave onmousemove onmouseover onmouseout
onmouseup onkeydown onkeypress onkeyup onabort onbeforeunload onerror
onhashchange onload onpageshow onpagehide onresize onscroll onunload
onblur onchange onfocus onfocusin onfocusout oninput oninvalid onreset
onsearch onselect onsubmit ondrag ondragend ondragenter ondragleave
ondragover ondragstart ondrop oncopy oncut onpaste onafterprint
onbeforeprint oncanplay oncanplaythrough ondurationchange onemptied
onended onloadeddata onloadedmetadata onloadstart onpause onplay
onplaying onprogress onratechange onseeked onseeking onstalled onsuspend
ontimeupdate onvolumechange onwaiting animationend animationiteration
animationstart transitionend onmessage onopen onmousewheel ononline
onoffline onpostate onshow onstorage ontoggle onwheel ontouchcancel
ontouchend ontouchmove ontouchstart color opacity background backgroundattachment
backgroundcolor backgroundimage backgroundposition backgroundrepeat backgroundclip
backgroundorigin backgroundsize border borderbottom borderbottomcolor
borderbottomleftradius borderbottomrightradius borderbottomstyle borderbottomwidth
bordercolor borderimage borderimageoutset borderimagerepeat borderimageslice

```

```
borderimagesource borderimagewidth borderleft borderleftcolor borderleftstyle
borderleftwidth borderradius borderright borderrightcolor borderrightstyle
borderrightwidth borderstyle bordertop bordertopcolor bordertopleftradius
bordertoprightradius bordertopstyle bordertopwidth borderwidth boxdecorationbreak
boxshadow bottom clear clip display float height left margin marginbottom marginleft
marginright margintop maxheight maxwidth minheight minwidth overflow overflowx
overflowy padding paddingbottom paddingleft paddingright paddingtop position
right top visibility width verticalalign zindex aligncontent alignitems alignself
flex flexbasis flexdirection flexflow flexgrow flexshrink flexwrap justifycontent
order hangingpunctuation hyphens letterspacing linebreak lineheight overflowwrap
tabsize textalign textalignlast textcombineupright textindent textjustify
texttransform whitespace wordbreak wordspacing wordwrap textdecoration
textdecorationcolor textdecorationline textdecorationstyle textshadow
textunderlineoffset @fontface @fontfeaturevalues font fontfamily fontfeaturesettings
fontkerning fontlanguageoverride fontsize fontsizeadjust fontstretch fontstyle
fontsynthesis fontvariant fontvariantalternates fontvariantcaps fontvarianteastasian
fontvariantligatures fontvariantnumeric fontvariantposition fontweight direction
textorientation unicodebidi writingmode bordercollapse borderspacing captionside
emptycells tablelayout counterincrement counterreset liststyle liststyleimage
liststyleposition liststyletype @keyframes animation animationdelay animationdirection
animationduration animationfillmode animationiterationcount animationname
animationplaystate animationtimingfunction backfacevisibility perspective
perspectiveorigin transform transformorigin transformstyle transition
transitionproperty transitionduration transitiontimingfunction transitiondelay
boxsizing content cursor imemode navdown navindex navleft navright navup
outline outlinecolor outlineoffset outlinestyle outlinewidth resize textoverflow
breakafter breakbefore breakinside columncount columnfill columngap columnrule
columnrulecolor columnrulestyle columnrulewidth columnspan columnwidth columns
widows orphans pagebreakafter pagebreakbefore pagebreakinside marks quotes
filter imageorientation imagerendering imageresolution objectfit objectposition
mask masktype mark markafter markbefore phonemes rest restafter restbefore
voicebalance voiceduration voicepitch voicepitchrange voicerate voicestress
voicevolume marqueeedirection marqueeplaycount marqueespeed marqueestyle datatoggle
dataride datatarget dataslideto dataslide datadismiss dataplacement datacontent
datatrigger dataspy dataoffset dataoffsettop
```

48.27 ScriptFunctions Class

This class contains methods for adding JavaScript code to the generated web page.

The class methods are merged to the Page class, so we can use the next methods with page objects directly.

Method	Parameters	Description
Script	cCode	Add cCode string between <script> and </script>
ScriptRedirection	cURL	set window.location to cURL
ScriptFunc	cFuncName,cCode	Define function cFuncName that contains cCode
ScriptFuncAlert	cFuncName,cMsg	Define function cFuncName that uses alert() to print cMsg
ScriptFuncAjax	cFuncName,cLink,cDiv	Define function cFuncName that load cLink in cDiv
ScriptFuncClean	cFuncName,cDiv	Define function cFuncName that clear the cDiv
ScriptFuncSelect	cF,aL,cD,cR,cGR,cFC,nTO,cL1,cL2	Used to Edit/Delete Grid Record
ScriptScrollFixed	cDiv,nSize	Set cDiv as Fixed Div with Size = nSize

48.28 StyleFunctions Class

This class contains methods for adding CSS to the generated web page.

Like ScriptFunctions Class, The StyleFunctions class methods are merged to the Page class, so we can use the next methods with page objects directly.

Method	Parameters	Description
StyleFloatLeft	None	Return float: left ;
StyleFloatRight	None	Return float: right ;
StyleSizeFull	None	Return width: 100% ; height: 100% ;
Stylecolor	x	Return " color: " + x + " ;"
Stylebgcolor	x	Return " background-color: " + x + " ;"
StyleTextCenter	None	Return "text-align: center ;"
StyleTextRight	None	Return "text-align: right ;"
StyleTextLeft	None	Return "text-align: left ;"
StyleSize	x,y	Return " width: " + x + " ; height: " + y + " ;"
StyleWidth	x	Return " width: " + x + " ;"
StyleHeight	x	Return " height: " + x + " ;"
StyleTop	x	Return " top: " + x + " ;"
StyleLeft	x	Return " Left: " + x + " ;"
StylePos	x,y	Return " top: " + x + " ;" + " Left: " + y + " ;"
StyleHorizontalCenter	None	Return " margin-right:auto ; margin-left:auto; "
StyleMarginTop	x	Return " margin-top: " + x + " ;"
StyleMarginRight	x	Return " margin-right: " + x + " ;"
StyleMarginLeft	x	Return " margin-left: " + x + " ;"
StyleDivCenter	nWidth,nHeight	Create Div in the center of the page
StyleAbsolute	None	Return " position:absolute ;"
StyleFixed	None	Return " position:fixed ;"
StyleZIndex	x	Return " z-index: " + x + " ;"
StyleFontSize	x	Return " font-size: " + x + " ;"
StyleGradient	x	Generate Gradient (x values from 1 to 60)
StyleTable	None	Set table properties
StyleTableRows	id	Set different color to even and odd rows in the table
StyleTableNoBorder	None	Return " border-style: none;"

48.29 WebPage Class

We use braces to access the active WebPage object attributes

Each one of these attribute will return a new object to access again using braces.

Attribute	Description
H1	Wraps HTML H1.
H2	Wraps HTML H2.
H3	Wraps HTML H3.
H4	Wraps HTML H4.
H5	Wraps HTML H5.
H6	Wraps HTML H6.
P	Wraps HTML P.
Link	Wraps HTML link.
NewLine	Wraps HTML NewLine.
Div	Wraps HTML Div.
Form	Wraps HTML Form.
Input	Wraps HTML Input.
TextArea	Wraps HTML TextArea.
Select	Wraps HTML Select.
Option	Wraps HTML Option.
Image	Wraps HTML Image.
UL	Wraps HTML UL.
LI	Wraps HTML LI.
Table	Wraps HTML Table.
TR	Wraps HTML TR.
TD	Wraps HTML TD.
TH	Wraps HTML TH.
Audio	Wraps HTML Audio.
Video	Wraps HTML Video.
Nav	Wraps HTML Nav.
Span	Wraps HTML Span.
Button	Wraps HTML Button.

48.30 HtmlPage Class

The same as the WebPage class with the next changes

1. No output to the stdout
2. Provide the Output Method to get the output

Syntax:

```
output() ---> The output as string
```


USING RINGLIBCURL

In this chapter we will learn about using RingLibCurl

49.1 Get Request

Example:

```
load "libcurl.ring"

curl = curl_easy_init()

curl_easy_setopt(curl, CURLOPT_FOLLOWLOCATION, 1)
curl_easy_setopt(curl, CURLOPT_URL, "http://ring-lang.sf.net")

curl_easy_perform(curl)

curl_easy_cleanup(curl)
```

49.2 Post Request

Example:

```
load "libcurl.ring"

curl = curl_easy_init()

cPostThis = "page=4&Number1=4&Number2=5"
curl_easy_setopt(curl, CURLOPT_URL, "http://localhost/ringapp/index.ring?page=3")
curl_easy_setopt(curl, CURLOPT_POSTFIELDS, cPostThis)

curl_easy_perform(curl)

curl_easy_cleanup(curl)
```

49.3 Facebook Login

Example:

```

load "libcurl.ring"

see "Enter Email : " give $login_email
See "Enter Password : " give $login_pass

curl = curl_easy_init()

curl_easy_setopt(curl, CURLOPT_URL, 'https://www.facebook.com/login.php')
curl_easy_setopt(curl, CURLOPT_POSTFIELDS, 'charset_test=j u s t a t e s t'+
' &email='+urlencode($login_email)+'&pass='+
urlencode($login_pass)+'&login=Login')
curl_easy_setopt(curl, CURLOPT_POST, 1)
curl_easy_setopt(curl, CURLOPT_HEADER, 0)
curl_easy_setopt(curl, CURLOPT_FOLLOWLOCATION, 1)
curl_easy_setopt(curl, CURLOPT_COOKIEJAR, "cookies.txt")
curl_easy_setopt(curl, CURLOPT_COOKIEFILE, "cookies.txt")
curl_easy_setopt(curl, CURLOPT_USERAGENT, "Mozilla/5.0 (Windows; U;" +
" Windows NT 5.1; en-US; rv:1.8.1.3) Gecko/20070309 Firefox/2.0.0.3")
curl_easy_setopt(curl, CURLOPT_REFERERER, "http://www.facebook.com")
curl_easy_setopt(curl, CURLOPT_SSL_VERIFYPEER, FALSE)
curl_easy_setopt(curl, CURLOPT_SSL_VERIFYHOST, 2)

mylist = curl_slist_append(NULL, 'Accept-Charset: utf-8')
curl_slist_append(mylist, 'Accept-Language: en-us,en;q=0.7,bn-bd;q=0.3')
curl_slist_append(mylist, 'Accept: text/xml,application/xml,' +
'application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5')
curl_easy_setopt(curl, CURLOPT_HTTPHEADER, mylist)

curl_easy_setopt(curl, CURLOPT_COOKIESESSION, false)

curl_easy_perform(curl)

curl_easy_cleanup(curl)

Func URLEncode cStr
    cOut = ""
    for x in cStr
        if isalnum(x)
            cOut += x
        but x = " "
            cOut += "+"
        else
            cOut += "%" + str2hex(x)
        ok
    next
    return cOut

```

49.4 Save Output to String

Example:

```

load "libcurl.ring"

curl = curl_easy_init()

curl_easy_setopt(curl, CURLOPT_FOLLOWLOCATION, 1)

```

```

curl_easy_setopt(curl, CURLOPT_URL, "http://ring-lang.sf.net")

cOutput = curl_easy_perform_silent(curl)

See "Output:" + nl
see cOutput

curl_easy_cleanup(curl)

```

49.5 Get Stock Data From Yahoo

Example:

```

Load "libcurl.ring"

### Part 1 --- Get Crumb and Cookie -----

See "Start curl_easy_init(): "+ nl
curl = curl_easy_init()                                ### >>> HANDLE >>> 01006BD0  CURL  0

    curl_easy_setopt(curl, CURLOPT_FOLLOWLOCATION, 1)
    curl_easy_setopt(curl, CURLOPT_COOKIEJAR, "cookies.txt")
    curl_easy_setopt(curl, CURLOPT_COOKIEFILE, "cookies.txt")
    curl_easy_setopt(curl, CURLOPT_URL, "https://finance.yahoo.com/quote/AMZN/history")

    ### HTML Data >>> STDOUT Window, Use curl_easy_perform_silent >>> String

cOutput = curl_easy_perform_silent(curl)              ### GO Get Data >>> String

### Extract Crumb from data
### "CrumbStore":{"crumb":"abcdefghijkl"},

if cOutput != NULL

    newStr1      = substr(cOutput, substr(cOutput, '"CrumbStore":{"crumb":"' ), 48 )
    nPosS       = substr(newStr1, ':' ) ; ### Start of crumb -2
    nPosE       = substr(newStr1, '"' ) ; ### End of crumb
    nCount      = nPosE - nPosS -2      ### size of crumb
    myCrumb     = substr(newStr1, nPosS +2, nCount)

    See "myCrumb.: |+ myCrumb +|"+ nl

    ### UniCode "\u002F" replace it with "/"
    if substr( myCrumb, "\u002F")
        myCrumb = substr( myCrumb, "\u002F", "/" )
        See "myCrumb2: |+ myCrumb +|"+ nl
    ok

else
    See "No Connectivity to Yahoo. Looking for Cookie and Crumb." +nl +nl
ok

### Part 2 --- Send URL with Crumb, and Cookie -----

```

```

    ### Send URL+Crumb to Yahoo to fetch 1st stock history data,

    $url = "https://query1.finance.yahoo.com/v7/finance/download/AMZN?period1=1277856000&period2=1277856000"

    curl_easy_setopt(curl, CURLOPT_URL, $url);
    cStr = curl_easy_perform_silent(curl)
    See cStr

curl_easy_cleanup(curl) ### REMEMBER to CLOSE CURL

```

Output:

```

myCrumb.: |sEEeW97mxvN|
Date,Open,High,Low,Close,Adj Close,Volume
2010-07-05,110.650002,117.480003,109.000000,117.260002,117.260002,21000400
2010-07-12,117.809998,124.879997,117.320000,118.489998,118.489998,29407300
2010-07-19,118.379997,121.250000,105.800003,118.870003,118.870003,74252100

```

USING RINGZIP

In this chapter we will learn about using RingZip

50.1 Create Zip File

Example : Create myfile.zip contains 4 files

```
load "ziplib.ring"
oZip = zip_openfile("myfile.zip", 'w')
zip_addfile(oZip, "test.c")
zip_addfile(oZip, "zip.c")
zip_addfile(oZip, "zip.h")
zip_addfile(oZip, "miniz.h")
zip_close(oZip)
```

50.2 Extract Zip File

Example : Extract myfile.zip to myfolder folder.

```
load "ziplib.ring"
zip_extract_allfiles("myfile.zip", "myfolder")
```

50.3 Print Files in Zip file

Example : Print file names in the myfile.zip

```
load "ziplib.ring"
oZip = zip_openfile("myfile.zip", 'r')
for x=1 to zip_filescount(oZip)
    see zip_getfilenamebyindex(oZip,x) + nl
next
zip_close(oZip)
```

50.4 Using RingZip Classes

The RingZip library comes with two classes. The Zip class and the ZipEntry class.

Example (1):

```
load "ziplib.ring"

new Zip {
    setFileName("myfile.zip")
    open("w")
    newEntry() {
        open("test.c")
        writefile("test.c")
        close()
    }
    close()
}
```

Example (2):

```
load "ziplib.ring"

new Zip {
    SetFileName("myfile.zip")
    Open("w")
    AddFile("test.c")
    AddFile("zip.c")
    AddFile("zip.h")
    AddFile("miniz.h")
    Close()
}
```

Example (3):

```
load "ziplib.ring"

new zip {
    SetFileName("myfile.zip")
    ExtractAllFiles("myfolder")
}
```

Example (4):

```
load "ziplib.ring"

new Zip {
    SetFileName("myfile.zip")
    Open("r")
    see FilesCount()
    Close()
}
```

Example (5):

```
load "ziplib.ring"

new Zip {
    SetFileName("myfile.zip")
    Open("r")
    for x = 1 to filescount()
        See GetFileNameByIndex(x) + nl
    next
    Close()
}
```

```
}

```

50.5 Zip Class Reference

Methods:

Method	Description/Output
SetFileName(cName)	Set the Zip file name
GetFileName()	Return the Zip file name
Open(cMode)	Open File, cMode = “a”, “w” or “r”
Close()	Close the Zip File
AddFile(cFileName)	Add file to the Zip file
ExtractAllFiles(cFolder)	Extract all files from the Zip file
FilesCount()	Return files count in the Zip file
GetFileNameByIndex(nIndex)	Return file name in the Zip file by file index
NewEntry()	Create new ZipEntry object

50.6 ZipEntry Class Reference

Methods:

Method	Description/Output
Open(cFileName)	Open new Entry
WriteFile(cFileName)	Write File to the Entry
WriteString(cString)	Write String to the Entry
Close()	Close the Entry

GRAPHICS AND 2D GAMES PROGRAMMING USING RINGALLEGRO

In this chapter we will learn how to use the allegro game programming library in our Ring applications.

We have the file gamelib.ring that load the DLL library that contains wrappers for the Allegro functions

```
Load "allegro.rh"
if iswindows()
    LoadLib("ring_allegro.dll")
but ismacosx()
    LoadLib("libringallegro.dylib")
else
    LoadLib("libringallegro.so")
ok
```

The file gamelib.ring uses the Load instruction to execute the file allegro.rh which is a ring source code file contains constants to be used in our programs. Then using the function LoadLib() we can load the DLL library "ring_allegro.dll".

To write portable code we can change the gamelib.ring to check the platform before loading the DLL/So file.

51.1 Drawing, Animation and Input

The next example uses the Allegro library for drawing, moving objects on the screen and getting input from the keyboard and the mouse.

```
Load "gamelib.ring"

al_init()
al_init_image_addon()

display = al_create_display(640,480)

al_show_native_message_box(display, "Hello", "Welcome",
    "Using Allegro from the Ring programming language",
    "", 0);

al_clear_to_color(al_map_rgb(0,0,255))

BOUNCER_SIZE = 40
bouncer_x = 10
bouncer_y = 20
bouncer = al_create_bitmap(BOUNCER_SIZE, BOUNCER_SIZE)
al_set_target_bitmap(bouncer)
al_clear_to_color(al_map_rgb(255,0,255))
```



```

for x = 1 to 30
    bouncer_x += x
    bouncer_y += x
    al_set_target_bitmap(al_get_backbuffer(display))
    al_clear_to_color(al_map_rgb(0,0,0))
    al_draw_bitmap(bouncer, bouncer_x, bouncer_y, 0)
    al_draw_bitmap(bouncer, 200+bouncer_x,200+ bouncer_y, 0)
    al_flip_display()
    al_rest(0.1)
next

al_clear_to_color(al_map_rgb(255,255,255))
image = al_load_bitmap("man2.jpg")
al_draw_bitmap(image,200,200,0)
al_flip_display()
al_rest(2)

event_queue = al_create_event_queue()
al_register_event_source(event_queue, al_get_display_event_source(display))

ev = al_new_allegro_event()
timeout = al_new_allegro_timeout()
al_init_timeout(timeout, 0.06)

FPS = 60
timer = al_create_timer(1.0 / FPS)
al_register_event_source(event_queue, al_get_timer_event_source(timer))
al_start_timer(timer)
redraw = true

SCREEN_W = 640
SCREEN_H = 480
BOUNCER_SIZE = 32
bouncer_x = SCREEN_W / 2.0 - BOUNCER_SIZE / 2.0
bouncer_y = SCREEN_H / 2.0 - BOUNCER_SIZE / 2.0
bouncer_dx = -4.0
bouncer_dy = 4.0

al_install_mouse()
al_register_event_source(event_queue, al_get_mouse_event_source())

al_install_keyboard()
al_register_event_source(event_queue, al_get_keyboard_event_source())

KEY_UP = 1
KEY_DOWN = 2
KEY_LEFT = 3
KEY_RIGHT = 4
Key = [false,false,false,false]

while true
    al_wait_for_event_until(event_queue, ev, timeout)
    switch al_get_allegro_event_type(ev)
    on ALLEGRO_EVENT_DISPLAY_CLOSE
        exit
    on ALLEGRO_EVENT_TIMER

    # Animation

```

```

    if bouncer_x < 0 or bouncer_x > SCREEN_W - BOUNCER_SIZE
        bouncer_dx = -bouncer_dx
    ok

    if bouncer_y < 0 or bouncer_y > SCREEN_H - BOUNCER_SIZE
        bouncer_dy = -bouncer_dy
    ok

    bouncer_x += bouncer_dx
    bouncer_y += bouncer_dy

    # Keyboard
    if key[KEY_UP] and bouncer_y >= 4.0
        bouncer_y -= 4.0
    ok
    if key[KEY_DOWN] and bouncer_y <= SCREEN_H - BOUNCER_SIZE - 4.0
        bouncer_y += 4.0
    ok
    if key[KEY_LEFT] and bouncer_x >= 4.0
        bouncer_x -= 4.0
    ok
    if key[KEY_RIGHT] and bouncer_x <= SCREEN_W - BOUNCER_SIZE - 4.0
        bouncer_x += 4.0
    ok

    redraw = true

on ALLEGRO_EVENT_MOUSE_AXES
    bouncer_x = al_get_allegro_event_mouse_x(ev)
    bouncer_y = al_get_allegro_event_mouse_y(ev)
on ALLEGRO_EVENT_MOUSE_ENTER_DISPLAY
    bouncer_x = al_get_allegro_event_mouse_x(ev)
    bouncer_y = al_get_allegro_event_mouse_y(ev)
on ALLEGRO_EVENT_MOUSE_BUTTON_UP
    exit
on ALLEGRO_EVENT_KEY_DOWN
    switch al_get_allegro_event_keyboard_keycode(ev)
        on ALLEGRO_KEY_UP
            key[KEY_UP] = true
        on ALLEGRO_KEY_DOWN
            key[KEY_DOWN] = true
        on ALLEGRO_KEY_LEFT
            key[KEY_LEFT] = true
        on ALLEGRO_KEY_RIGHT
            key[KEY_RIGHT] = true
    off
on ALLEGRO_EVENT_KEY_UP
    switch al_get_allegro_event_keyboard_keycode(ev)
        on ALLEGRO_KEY_UP
            key[KEY_UP] = false
        on ALLEGRO_KEY_DOWN
            key[KEY_DOWN] = false
        on ALLEGRO_KEY_LEFT
            key[KEY_LEFT] = false
        on ALLEGRO_KEY_RIGHT
            key[KEY_RIGHT] = false
        on ALLEGRO_KEY_ESCAPE
            exit

```

```
        off
        if redraw and al_is_event_queue_empty(event_queue)
            redraw = false
            al_clear_to_color(al_map_rgb(0,0,0))
            al_draw_bitmap(bouncer, bouncer_x, bouncer_y, 0)
            al_flip_display()

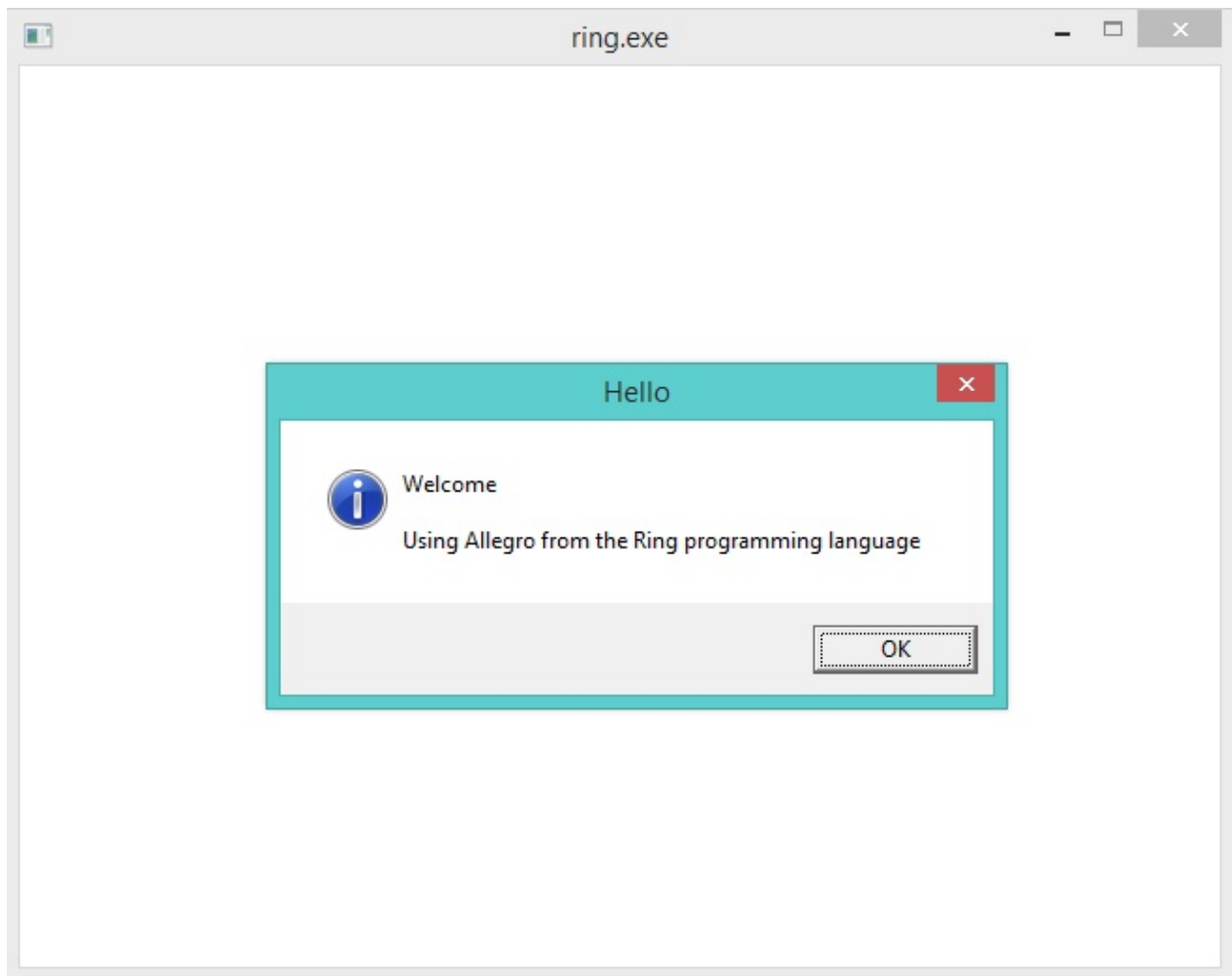
        ok
        callgc()
    end

    al_destroy_timer(timer)
    al_destroy_allegro_event(ev)
    al_destroy_allegro_timeout(timeout)
    al_destroy_event_queue(event_queue)
    al_destroy_bitmap(bouncer)
    al_destroy_bitmap(image)
    al_destroy_display(display)
```

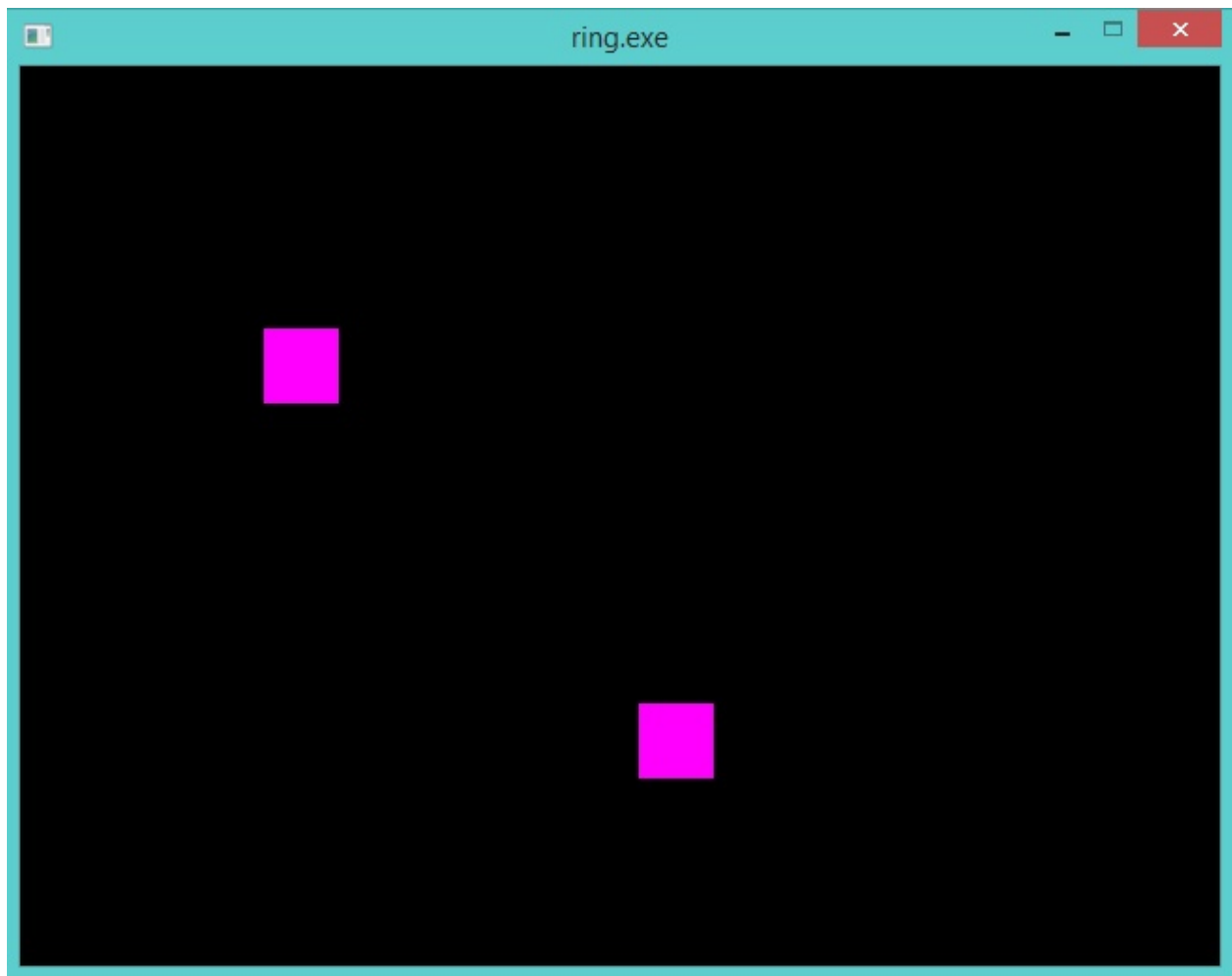
Note: In the previous example we used the function `callgc()` which is a Ring function to force calling the Garbage collector inside the While/End loop.

Program Output:

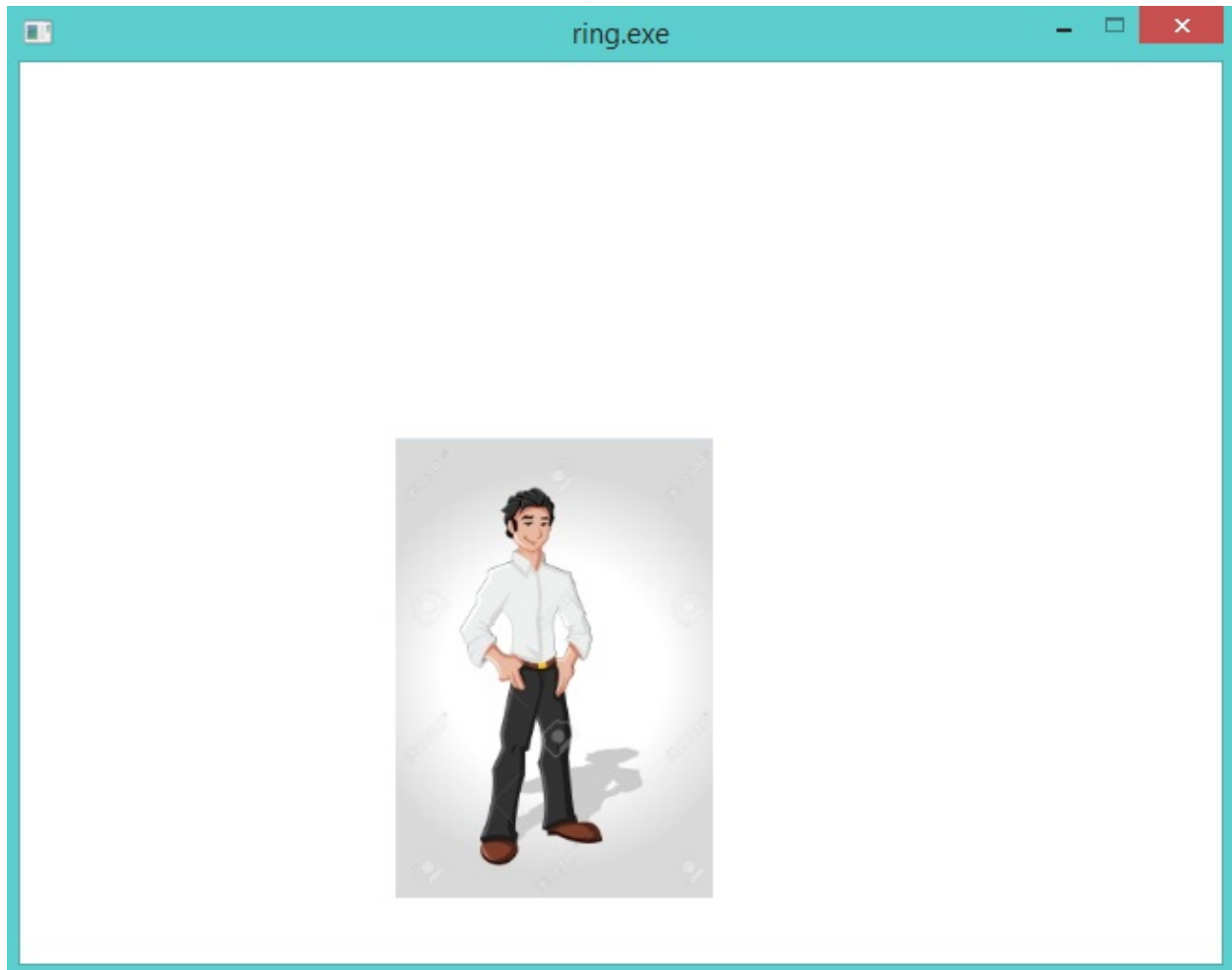
At first the program display a messagebox



Then we see two rectangles are moving on the screen



Then we see an image displayed on the screen



Finally we have one rectangle, and we see it moving all of the time on the screen but we can control it using the Mouse and/or the Keyboard



51.2 Using TrueType Fonts

In this example we will see how to use TrueType Fonts *.ttf in our Games using Allegro

```
Load "gamelib.ring"

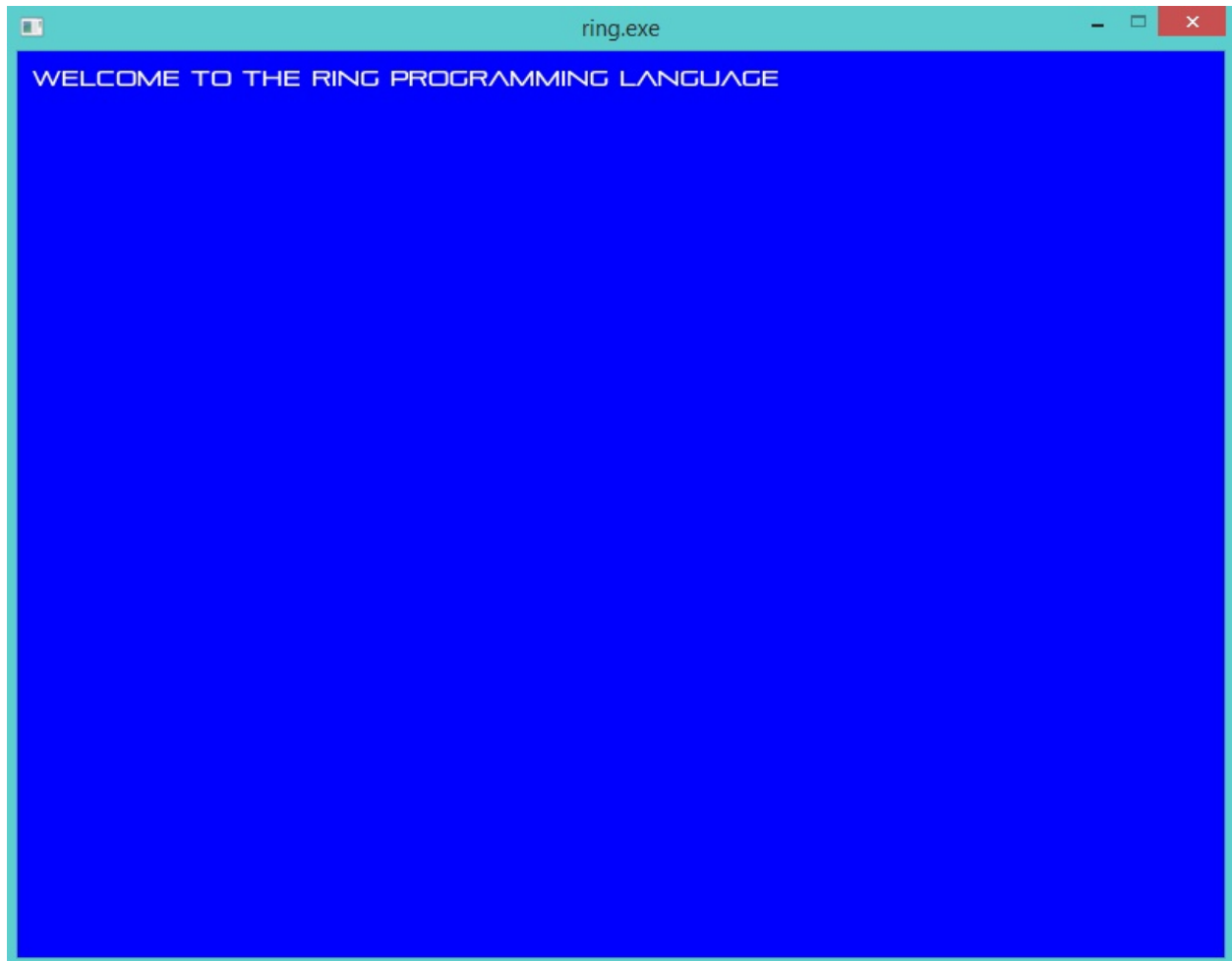
al_init()
al_init_font_addon()
al_init_ttf_addon()

display = al_create_display(800,600)

al_clear_to_color(al_map_rgb(0,0,255))
font = al_load_ttf_font("pirulen.ttf",14,0 )
al_draw_text(font, al_map_rgb(255,255,255), 10, 10,ALLEGRO_ALIGN_LEFT,
             "Welcome to the Ring programming language")
al_flip_display()
al_rest(2)

al_destroy_display(display)
```

Screen Shot:



51.3 Playing Sound Files

The next example play a sound file

```
Load "gamelib.ring"

al_init()
al_install_audio()
al_init_acodec_addon()
al_reserve_samples(1)

sample = al_load_sample( "footstep.wav" )

sampleid = al_new_allegro_sample_id()
al_play_sample(sample, 1.0, 0.0, 1.0, ALLEGRO_PLAYMODE_LOOP, sampleid)

display = al_create_display(640, 480)
al_clear_to_color(al_map_rgb(0, 0, 255))
al_flip_display()
al_rest(10)

al_destroy_allegro_sample_id(sampleid)
```



```
al_destroy_sample(sample)
al_destroy_display(display)

al_exit()
```

51.4 Scaling and Rotating Images

The next example display and rotate an image

```
Load "gamelib.ring"

al_init()
al_init_image_addon()

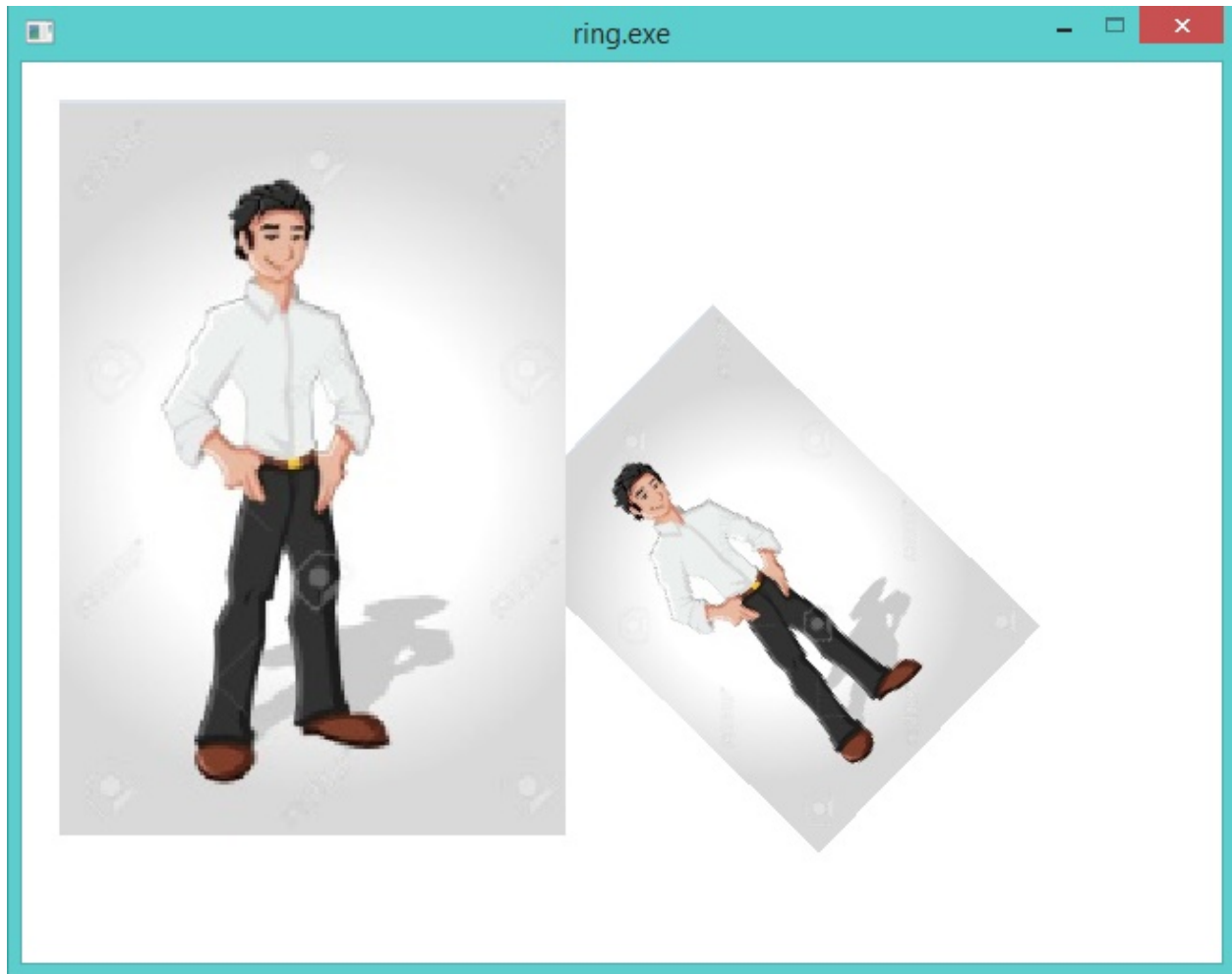
display = al_create_display(640,480)
al_set_target_bitmap(al_get_backbuffer(display))
al_clear_to_color(al_map_rgb(255,255,255))

image = al_load_bitmap("man2.jpg")
al_draw_rotated_bitmap(image,0,0,250,250,150,0)
al_draw_scaled_bitmap(image,0,0,250,250,20,20,400,400,0)

al_flip_display()
al_rest(2)

al_destroy_bitmap(image)
al_destroy_display(display)
```

Screen Shot:



51.5 Display Transparent Image

The next example display image with white background on another image

```
Load "gamelib.ring"

al_init()
al_init_image_addon()

display = al_create_display(640,480)
imageback = al_load_bitmap("palace.jpg")
al_draw_bitmap(imageback,0,0,0)

image = al_load_bitmap("man4.png")
al_convert_mask_to_alpha(image,al_map_rgb(255,255,255))
al_draw_bitmap(image,0,0,0)
al_flip_display()
al_rest(10)

al_destroy_bitmap(image)
al_destroy_display(display)
```

Screen Shot:



51.6 Using Threads

In this example we will learn how to use threads from the Allegro library

```
Load "gamelib.ring"

o1 = new mythreads

Func Main
    al_init()
    for k = 1 to 5
        al_create_thread("o1.thread1()")
        al_create_thread("o1.thread2()")
        al_create_thread("o1.thread3()")
    next
    al_rest(2)

Class Mythreads

    cAppName = "Threads Application"
```

```

Func Thread1
    for x = 1 to 5
        see x + nl
    next
See 'Thread(1) : Application Name : ' + cAppName + nl

Func Thread2
    for x = 1 to 5
        see '*****' + x + nl
    next
See 'Thread(2) : Application Name : ' + cAppName + nl

Func Thread3
    for x = 1 to 5
        see '!!!!' + x + nl
    next
See 'Thread(3) : Application Name : ' + cAppName + nl

```

Output:

```

1
2
3
4
5
Thread(1) : Application Name : Threads Application
*****1
*****2
*****3
*****4
*****5
Thread(2) : Application Name : Threads Application
!!!!1
!!!!2
!!!!3
!!!!4
!!!!5
Thread(3) : Application Name : Threads Application
1
2
3
4
5
Thread(1) : Application Name : Threads Application
!!!!1
!!!!2
!!!!3
!!!!4
!!!!5
Thread(3) : Application Name : Threads Application
*****1
*****2
*****3
*****4
*****5
Thread(2) : Application Name : Threads Application
*****1
*****2

```

```

*****3
*****4
*****5
Thread(2) : Application Name : Threads Application
!!!!1
!!!!2
!!!!3
!!!!4
!!!!5
Thread(3) : Application Name : Threads Application
1
2
3
4
5
Thread(1) : Application Name : Threads Application
*****1
*****2
*****3
*****1
*****4
*****2
!!!!1
*****5
*****3
1
!!!!2
Thread(2) : Application Name : Threads Application
1
*****4
!!!!1
2
!!!!3
!!!!4
*****5
!!!!2
3
2
!!!!5
Thread(2) : Application Name : Threads Application
!!!!3
4
3
Thread(3) : Application Name : Threads Application
!!!!4
5
4
!!!!5
Thread(1) : Application Name : Threads Application
5
Thread(3) : Application Name : Threads Application
Thread(1) : Application Name : Threads Application

```

USING RINGLIBSDL

In this chapter we will learn about using RingLibSDL to create games based on the LibSDL, SDLImage, SDLTTF and SDLMixers libraries.

Tip: RingLibSDL is not distributed with the binary releases for desktop which uses RingAllegro

Note: To use RingLibSDL, Check ring/android/ringlibsdl folder.

52.1 Create Window

Example:

```
Load "libsdl.ring"

SDL_Init(SDL_INIT_EVERYTHING)
win = SDL_CreateWindow("Hello World!", 100, 100, 640, 480, SDL_WINDOW_SHOWN)
SDL_Delay(2000)
SDL_DestroyWindow(win)
SDL_Quit()
```

52.2 Display Image

Example:

```
Load "libsdl.ring"

SDL_Init(SDL_INIT_EVERYTHING)
win = SDL_CreateWindow("Hello World!", 100, 100, 640, 480, SDL_WINDOW_SHOWN)
ren = SDL_CreateRenderer(win, -1, SDL_RENDERER_ACCELERATED | SDL_RENDERER_PRESENTVSYNC )
bmp = SDL_LoadBMP("hello.bmp")
tex = SDL_CreateTextureFromSurface(ren, bmp)
SDL_FreeSurface(bmp)
SDL_RenderClear(ren)
SDL_RenderCopy2(ren, tex)
SDL_RenderPresent(ren)
SDL_Delay(2000)
SDL_DestroyTexture(tex)
SDL_DestroyRenderer(ren)
SDL_DestroyWindow(win)
SDL_Quit()
```

52.3 Switch between two images

Example:

```
Load "libsdl.ring"

SDL_Init(SDL_INIT_EVERYTHING)
win = SDL_CreateWindow("Hello World!", 100, 100, 640, 480, SDL_WINDOW_SHOWN)
ren = SDL_CreateRenderer(win, -1, SDL_RENDERER_ACCELERATED | SDL_RENDERER_PRESENTVSYNC )
bmp = SDL_LoadBMP("hello.bmp")
tex = SDL_CreateTextureFromSurface(ren,bmp)
SDL_FreeSurface(bmp)
bmp = SDL_LoadBMP("hello2.bmp")
tex2 = SDL_CreateTextureFromSurface(ren,bmp)
SDL_FreeSurface(bmp)

for x = 1 to 10 showtex(tex) showtex(tex2) next

SDL_DestroyTexture(tex)
SDL_DestroyTexture(tex2)
SDL_DestroyRenderer(ren)
SDL_DestroyWindow(win)
SDL_Quit()

func showtex oTex
    SDL_RenderClear(ren)
    SDL_RenderCopy2(ren,oTex)
    SDL_RenderPresent(ren)
    SDL_Delay(200)
```

52.4 Draw Rectangle

Example:

```
Load "libsdl.ring"

SDL_Init(SDL_INIT_EVERYTHING)
win = SDL_CreateWindow("Hello World!", 100, 100, 640, 480, SDL_WINDOW_SHOWN)
ren = SDL_CreateRenderer(win, -1, SDL_RENDERER_ACCELERATED | SDL_RENDERER_PRESENTVSYNC )
SDL_RenderClear(ren)
rect = sdl_new_sdl_rect()
sdl_set_sdl_rect_x(rect,10)
sdl_set_sdl_rect_y(rect,10)
sdl_set_sdl_rect_w(rect,100)
sdl_set_sdl_rect_h(rect,100)
SDL_SetRenderDrawColor(ren,255,255,255,255)
SDL_RenderDrawRect(ren,rect)
sdl_destroy_sdl_rect(rect)
SDL_RenderPresent(ren)
SDL_Delay(2000)
SDL_DestroyRenderer(ren)
SDL_DestroyWindow(win)
SDL_Quit()
```

52.5 Display PNG Images

Example:

```
Load "libsdl.ring"

SDL_Init(SDL_INIT_EVERYTHING)
win = SDL_CreateWindow("Hello World!", 100, 100, 640, 480, SDL_WINDOW_SHOWN)
ren = SDL_CreateRenderer(win, -1, SDL_RENDERER_ACCELERATED | SDL_RENDERER_PRESENTVSYNC)
bmp = IMG_Load("hello3.png")
tex = SDL_CreateTextureFromSurface(ren, bmp)
SDL_FreeSurface(bmp)
SDL_RenderClear(ren)
SDL_RenderCopy2(ren, tex)
SDL_RenderPresent(ren)
SDL_Delay(2000)
SDL_DestroyTexture(tex)
SDL_DestroyRenderer(ren)
SDL_DestroyWindow(win)
SDL_Quit()
```

52.6 Use TTF Fonts

Example:

```
Load "libsdl.ring"

SDL_Init(SDL_INIT_EVERYTHING)
win = SDL_CreateWindow("Hello World!", 100, 100, 640, 480, SDL_WINDOW_SHOWN)
ren = SDL_CreateRenderer(win, -1, SDL_RENDERER_ACCELERATED | SDL_RENDERER_PRESENTVSYNC)
SDL_RenderClear(ren)

TTF_Init()
font = TTF_OpenFont("pirulen.ttf", 16)
color = sdl_new_sdl_color()
sdl_set_sdl_color_r(color, 0)
sdl_set_sdl_color_g(color, 255)
sdl_set_sdl_color_b(color, 0)
text = TTF_RenderText_Solid(font, "Welcome to the Ring language", color)
surface = SDL_GetWindowSurface(win)
SDL_BlitSurface(text, nullpointer(), surface, nullpointer())
SDL_UpdateWindowSurface(win)
SDL_Delay(2000)

SDL_Destroy_SDL_Color(color)
SDL_FreeSurface(text)
TTF_CloseFont(font)
SDL_DestroyRenderer(ren)
SDL_DestroyWindow(win)
SDL_Quit()
```

52.7 Display Transparent Images

Example:


```

Load "libsdl.ring"

SDL_Init(SDL_INIT_EVERYTHING)

flags = IMG_INIT_JPG | IMG_INIT_PNG
IMG_Init(flags)

win = SDL_CreateWindow("Hello World!", 100, 100, 800, 600, SDL_WINDOW_SHOWN)
ren = SDL_CreateRenderer(win, -1, SDL_RENDERER_ACCELERATED | SDL_RENDERER_PRESENTVSYNC )

bmp = IMG_Load("stars.jpg")
tex = SDL_CreateTextureFromSurface(ren,bmp)
SDL_FreeSurface(bmp)
SDL_RenderClear(ren)
SDL_RenderCopy(ren,tex,nullpointer(),nullpointer())
SDL_DestroyTexture(tex)

bmp = IMG_Load("player.png")
# Image - Set Transparent color (white)
myformat = sdl_get_sdl_surface_format(bmp)
white = SDL_MapRGB(myformat, 255, 255, 255)
SDL_SetColorKey(bmp, SDL_True, white)

tex = SDL_CreateTextureFromSurface(ren,bmp)
SDL_FreeSurface(bmp)
rect = sdl_new_sdl_rect()
sdl_set_sdl_rect_x(rect,0)
sdl_set_sdl_rect_y(rect,0)
sdl_set_sdl_rect_w(rect,100)
sdl_set_sdl_rect_h(rect,100)
SDL_RenderCopy(ren,tex,nullpointer(),rect)

SDL_SetTextureBlendMode(tex,2)
SDL_SetTextureAlphaMod(tex,255)
sdl_set_sdl_rect_x(rect,200)
sdl_set_sdl_rect_y(rect,200)
sdl_set_sdl_rect_w(rect,100)
sdl_set_sdl_rect_h(rect,100)
SDL_RenderCopy(ren,tex,nullpointer(),rect)

SDL_DestroyTexture(tex)
SDL_Destroy_SDL_Rect(rect)

SDL_RenderPresent(ren)
SDL_Delay(2000)
SDL_DestroyRenderer(ren)
SDL_DestroyWindow(win)
SDL_Quit()

```

52.8 Close Window Event

Example:

```

Load "libsdl.ring"

SDL_Init(SDL_INIT_EVERYTHING)

```

```

win = SDL_CreateWindow("Hello World!", 100, 100, 640, 480, SDL_WINDOW_SHOWN)

myevent = sdl_new_sdl_event()
while true
    thevent = sdl_pollevent(myevent)
    switch sdl_get_sdl_event_type(myevent)
        on sdl_get_sdl_quit()
            exit
        on sdl_get_sdl_keydown()
            Key = SDL_GET_SDL_Event_key_keysym_sym(myevent)
            if key = 27 exit ok
    end
end

SDL_DestroyWindow(win)
SDL_Quit()

```

52.9 Mouse Events

Example:

```

Load "libsdl.ring"

SDL_Init(SDL_INIT_EVERYTHING)

win = SDL_CreateWindow("Mouse Events ", 100, 100, 640, 480, SDL_WINDOW_SHOWN)

TTF_Init()
font = TTF_OpenFont("pirulen.ttf", 16)
color = sdl_new_sdl_color()
sdl_set_sdl_color_r(color,0)
sdl_set_sdl_color_g(color,255)
sdl_set_sdl_color_b(color,0)

surface = SDL_GetWindowSurface(win)

myevent = sdl_new_sdl_event()
while true
    cMsg = ""
    sdl_pollevent(myevent)
    switch sdl_get_sdl_event_type(myevent)
        on SDL_QUIT
            exit
        on SDL_KEYDOWN
            Key = SDL_GET_SDL_Event_key_keysym_sym(myevent)
            if key = 27 exit ok
        on SDL_MOUSEBUTTONDOWN
            if sdl_get_Sdl_Event_button_button(myevent) = SDL_BUTTON_LEFT
                SDL_SETWINDOWTITLE(win, " Button_Left_Down " )
            but sdl_get_Sdl_Event_button_button(myevent) = SDL_BUTTON_MIDDLE
                SDL_SETWINDOWTITLE(win, " Button_Middle_Down " )
            but sdl_get_Sdl_Event_button_button(myevent) = SDL_BUTTON_RIGHT
                SDL_SETWINDOWTITLE(win, " Button_Right_Down " )
            ok
        on SDL_MOUSEMOTION

```

```

        sdl_fillrect(surface,nullpointer(),0)
        if sdl_get_sdl_event_motion_xrel(myevent) < 0
            cMsg += " Left "
        else
            cMsg += " Right "
        ok
        if sdl_get_sdl_event_motion_yrel(myevent) < 0
            cMsg += " Up "
        else
            cMsg += " Down "
        ok
        cMsg += " x = " + sdl_get_sdl_event_motion_x(myevent)
        cMsg += " y = " + sdl_get_sdl_event_motion_y(myevent)
        showmsg(cMsg)
    off
end

SDL_Destroy_SDL_Color(Color)
TTF_CloseFont(font)
SDL_DestroyWindow(win)
SDL_Quit()

func showmsg mymsg
    text = TTF_RenderText_Solid(font,mymsg,color)
    SDL_BlitterSurface(text, nullpointer(), surface, nullpointer())
    SDL_UpdateWindowSurface(win)
    SDL_FreeSurface(text)

```

52.10 Play Sound

Example:

```

Load "libsdl.ring"

SDL_Init(SDL_INIT_EVERYTHING)
win = SDL_CreateWindow("Hello World!", 100, 100, 640, 480, SDL_WINDOW_SHOWN)
Mix_OpenAudio( 44100, MIX_DEFAULT_FORMAT , 2, 10000)
Mix_AllocateChannels(4)
sound = Mix_LoadWav( "sound.wav" )
Mix_VolumeChunk(sound,1)
Mix_PlayChannel(1,sound,0)

myevent = sdl_new_sdl_event()
while true
    thevent = sdl_pollevent(myevent)
    switch sdl_get_sdl_event_type(myevent)
        on sdl_get_sdl_quit()
            exit
        on sdl_get_sdl_keydown()
            Key = SDL_GET_SDL_Event_key_keysym_sym(myevent)
            if key = 27 exit ok
    off
end

Mix_FreeChunk( sound )
Mix_CloseAudio()

```

```
Mix_Quit()  
SDL_DestroyWindow(win)  
SDL_Quit()
```

DEMO PROJECT - GAME ENGINE FOR 2D GAMES

In this chapter we will learn about using the different programming paradigms in the same project.

We will create a simple Game Engine for 2D Games.

You can use the Engine directly to create 2D Games for Desktop or Mobile.

53.1 Project Layers

The project contains the next layers

- Games Layer (Here we will use declarative programming)
- Game Engine Classes (Here we will use the Object-Oriented Programming paradigm)
- Interface to graphics library (Here we will use procedural programming)
- Graphics Library bindings (Here we have RingAllegro and RingLibSDL)

53.2 Graphics Library bindings

We already have RingAllegro to use the Allegro game programming library and we have RingLibSDL to use the LibSDL game programming library.

Both of RingAllegro and RingLibSDL are created using the C language with the help of the Ring code generator for extensions.

Each of them is over 10,000 lines of C code which is generated after writing simple configuration files (That are processed by the code generator).

Each configuration file determines the functions names, structures information and constants then the generator process this configuration file to produce the C code and the library that can be loaded from Ring code.

Using RingAllegro and RingLibSDL is very similar to using Allegro and LibSDL from C code where you have the same functions but we can build on that using the Ring language features

- RingAllegro Source Code : <https://github.com/ring-lang/ring/tree/master/extensions/ringallegro>
- RingLibSDL Source Code : <https://github.com/ring-lang/ring/tree/master/extensions/ringsdl>

53.3 Interface to graphics library

In this layer we have `gl_allegro.ring` and `gl_libsdl.ring`

Each library provides the same functions to be used with interacting with the Graphics Library.

This layer hides the details and the difference between RingAllegro and RingLibSDL.

You have the same functions, Just use it and you can switch between Allegro and LibSDL at anytime.

Why ?

Allegro is very simple, we can use it to quickly create 2D games for Windows, Linux and MacOS X.

In Ring 1.0 we started by supporting Allegro.

Also LibSDL is very powerful and popular, very easy to use for Mobile Development.

Ring 1.1 comes with support for LibSDL so we can quickly create games for Mobile.

Note: We can use just one library for Desktop and Mobile development.

- `gl_allegro.ring` source code : https://github.com/ring-lang/ring/blob/master/ringlibs/gameengine/gl_allegro.ring
- `gl_libsdl.ring` source code : https://github.com/ring-lang/ring/blob/master/ringlibs/gameengine/gl_libsdl.ring

53.4 Game Engine Classes

The Engine comes with the next classes

- GameBase class
- Resources class
- Game class
- GameObject class
- Sprite class
- Text class
- Animate class
- Sound class
- Map class
- Source Code : <https://github.com/ring-lang/ring/blob/master/ringlibs/gameengine/gameengine.ring>

53.5 Games Layer

In this layer we create our games using the Game Engine classes

The classes are designed to be used through Declarative Programming.

In our games we will use the next classes

- Game class
- Sprite class

- Text class
- Animate class
- Sound class
- Map class

Note: Other classes in the engine are for internal use by the engine.

We will introduce some examples and three simple games :-

- Stars Fighter Game
- Flappy Bird 3000 Game
- Super Man 2016 Game

53.6 Game Class

The next table present the class attributes.

Attributes	Description
FPS	Number determines how many times the draw() method will be called per second.
FixedFPS	Number determines how many times the animate() method will be called per second.
Title	String determines the window title of the game.
aObjects	List contains all objects in the game
shutdown	True/False value to end the game loop

The next table present the class methods.

Method	Description
refresh()	Delete objects.
settitle(cTitle)	Set the window title using a string parameter.
shutdown()	Close the application.

The next table present a group of keywords defined by the class.

Keyword	Description
sprite	Create new Sprite object and add it to the game objects.
text	Create new Text object and add it to the game objects.
animate	Create new Animate object and add it to the game objects.
sound	Create new Sound object and add it to the game objects.
map	Create new Map object and add it to the game objects.

53.7 GameObject Class

The next table present the class attributes.

Attributes	Description
enabled	True/False determine the state of the object (Active/Not Active)
x	Number determine the x position of the object.
y	Number determine the y position of the object.
width	Number determine the width of the object.
height	Number determine the height of the object.
nIndex	Number determine the index of the object in objects list.
animate	True/False to animate the object or not.
move	True/False to move the object using the keyboard or not.
Scaled	True/False to scale the object image or not.
draw	Function to be called when drawing the object.
state	Function to be called for object animation.
keypress	Function to be called when a key is pressed.
mouse	Function to be called when a mouse event happens.

The next table present the class methods.

Method	Description
keyboard(oGame,nkey)	Check Keyboard Events
mouse(oGame,nType,aMouseList)	Check Mouse Events
rgb(r,g,b)	Return new color using the RGB (Red, Green and Blue) Values.

53.8 Sprite Class

Parent Class : GameObject Class

The next table present the class attributes.

Attributes	Description
image	String determine the image file name.
point	Number determine the limit of automatic movement of the object.
direction	Number determine the direction of movement.
nstep	Number determine the increment/decrement during movement.
type	Number determine the object type in the game (Optional).
transparent	True/False value determine if the image is transparent.

The next table present the class methods.

Method	Description
Draw(oGame)	Draw the object

53.9 Text Class

Parent Class : Sprite Class

The next table present the class attributes.

Attributes	Description
size	Number determine the font size
font	String determine the font file name
text	String determine the text to be displayed
color	Number determine the color

The next table present the class methods.

Method	Description
Draw(oGame)	Draw the object

53.10 Animate Class

Parent Class : Sprite Class

The next table present the class attributes.

Attributes	Description
frames	Number determine the number of frames
frame	Number determine the active frame
framewidth	Number determine the frame width.
animate	True/False determine using animate or not.
scaled	True/False determine scaling image or not.

The next table present the class methods.

Method	Description
Draw(oGame)	Draw the object

53.11 Sound Class

Parent Class : GameObject Class

The next table present the class attributes.

Attributes	Description
file	String determine the sound file name.
once	True/False determine to play the file one time or not (loop).

The next table present the class methods.

Method	Description
playsound()	Play the sound file

53.12 Map Class

Parent Class : Sprite Class

The next table present the class attributes.

Attributes	Description
aMap	List determine the map content using numbers.
aImages	List determine the image used for each number in the map.
BlockWidth	Number determine the block width (default = 32).
BlockHeight	Number determine the block height (default = 32).
Animate	True/False determine the animation status.

The next table present the class methods.

Method	Description
getvalue(x,y)	Return the item value in the Map according to the visible part

53.13 Using the Game Engine - Creating the Game Window

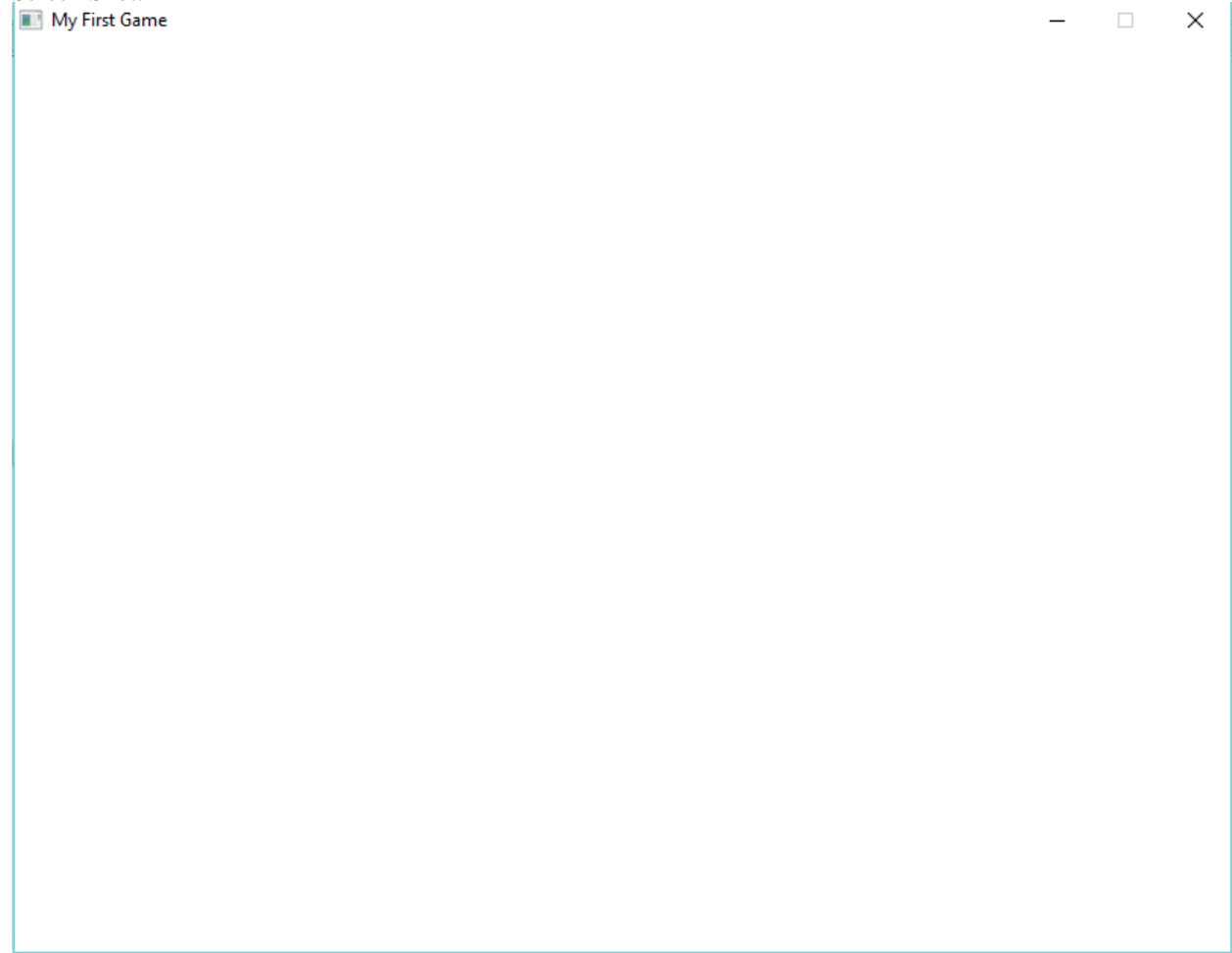
```
Load "gameengine.ring"  # Give Control to the Game Engine

func main              # Called by the Game Engine

    oGame = New Game    # Create the Game Object
    {
        title = "My First Game"
    }                  # Start the Events Loop
```

Note: if you want to define global variables, this must be before load “gameengine.ring” because this instruction will give the control to the game engine.

Screen Shot:



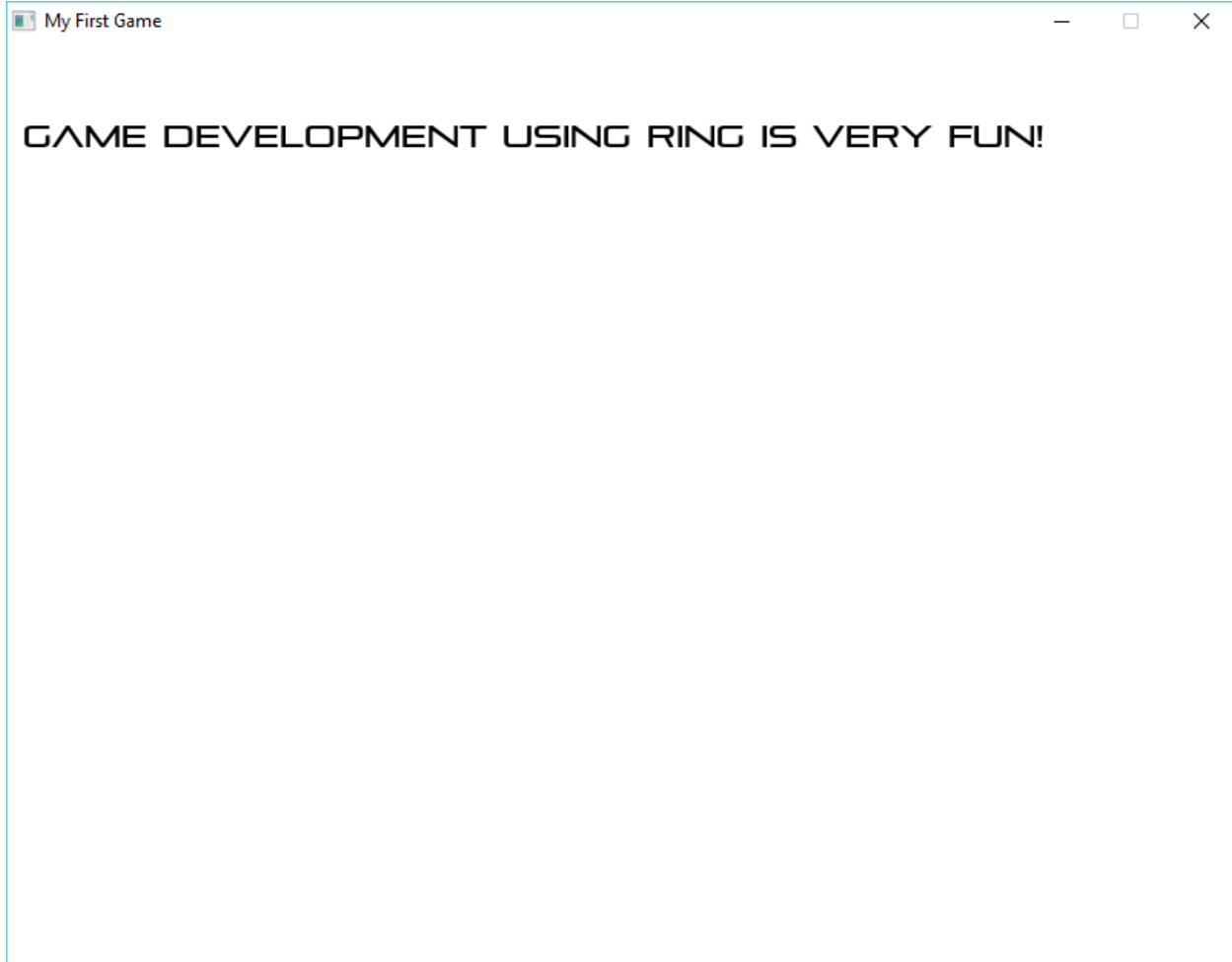
53.14 Using the Game Engine - Drawing Text

```
Load "gameengine.ring"  # Give Control to the Game Engine

func main              # Called by the Game Engine
```

```
oGame = New Game           # Create the Game Object
{
    title = "My First Game"
    text {
        x = 10  y=50
        animate = false
        size = 20
        file = "fonts/pirulen.ttf"
        text = "game development using ring is very fun!"
        color = rgb(0,0,0)
    }
}                             # Start the Events Loop
```

Screen Shot:



53.15 Using the Game Engine - Moving Text

```
Load "gameengine.ring" # Give Control to the Game Engine

func main               # Called by the Game Engine

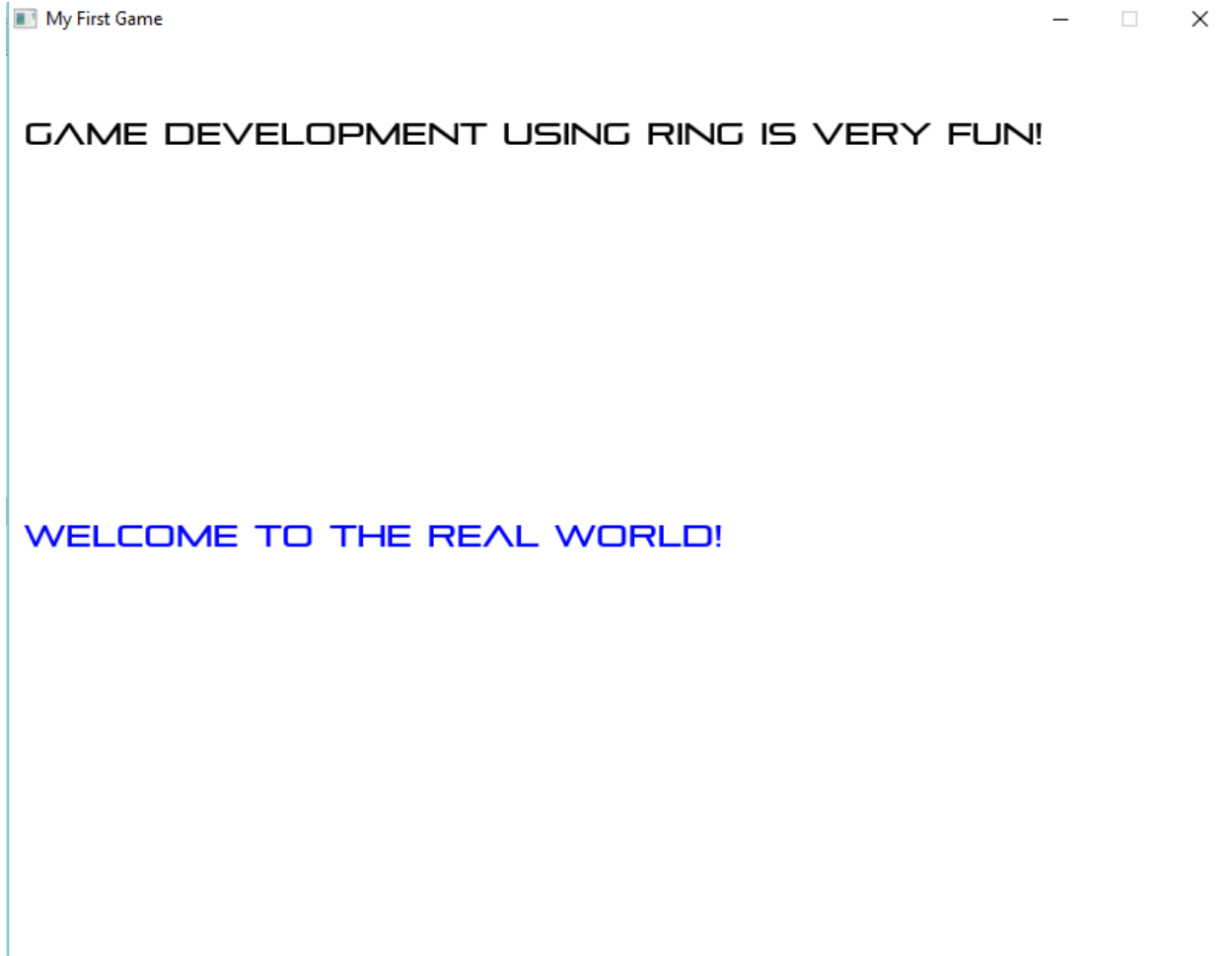
    oGame = New Game     # Create the Game Object
```

```

{
    title = "My First Game"
    text {
        x = 10  y=50
        animate = false
        size = 20
        file = "fonts/pirulen.ttf"
        text = "game development using ring is very fun!"
        color = rgb(0,0,0)      # Color = black
    }
    text {
        x = 10  y=150
        # Animation Part =====
        animate = true          # Use Animation
        direction = GE_DIRECTION_INCVERTICAL    # Increase y
        point = 400              # Continue until y=400
        nStep = 3                # Each time y+= 3
        #=====
        size = 20
        file = "fonts/pirulen.ttf"
        text = "welcome to the real world!"
        color = rgb(0,0,255)     # Color = Blue
    }
}                                # Start the Events Loop

```

Screen Shot:



53.16 Using the Game Engine - Playing Sound

```

Load "gameengine.ring"  # Give Control to the Game Engine

func main                # Called by the Game Engine

    oGame = New Game      # Create the Game Object
    {
        title = "My First Game"
        text {
            x = 10  y=50
            animate = false
            size = 20
            file = "fonts/pirulen.ttf"
            text = "game development using ring is very fun!"
            color = rgb(0,0,0)      # Color = black
        }
        text {
            x = 10  y=150
            # Animation Part =====
            animate = true          # Use Animation
            direction = GE_DIRECTION_INCVERTICAL  # Increase y
        }
    }

```

```

        point = 400          # Continue until y=400
        nStep = 3           # Each time y+= 3
        #=====
        size = 20
        file = "fonts/pirulen.ttf"
        text = "welcome to the real world!"
        color = rgb(0,0,255) # Color = Blue
    }
    Sound {                  # Play Sound
        file = "sound/music1.wav" # Sound File Name
    }
}                            # Start the Events Loop

```

53.17 Using the Game Engine - Animation

```

Load "gameengine.ring" # Give Control to the Game Engine

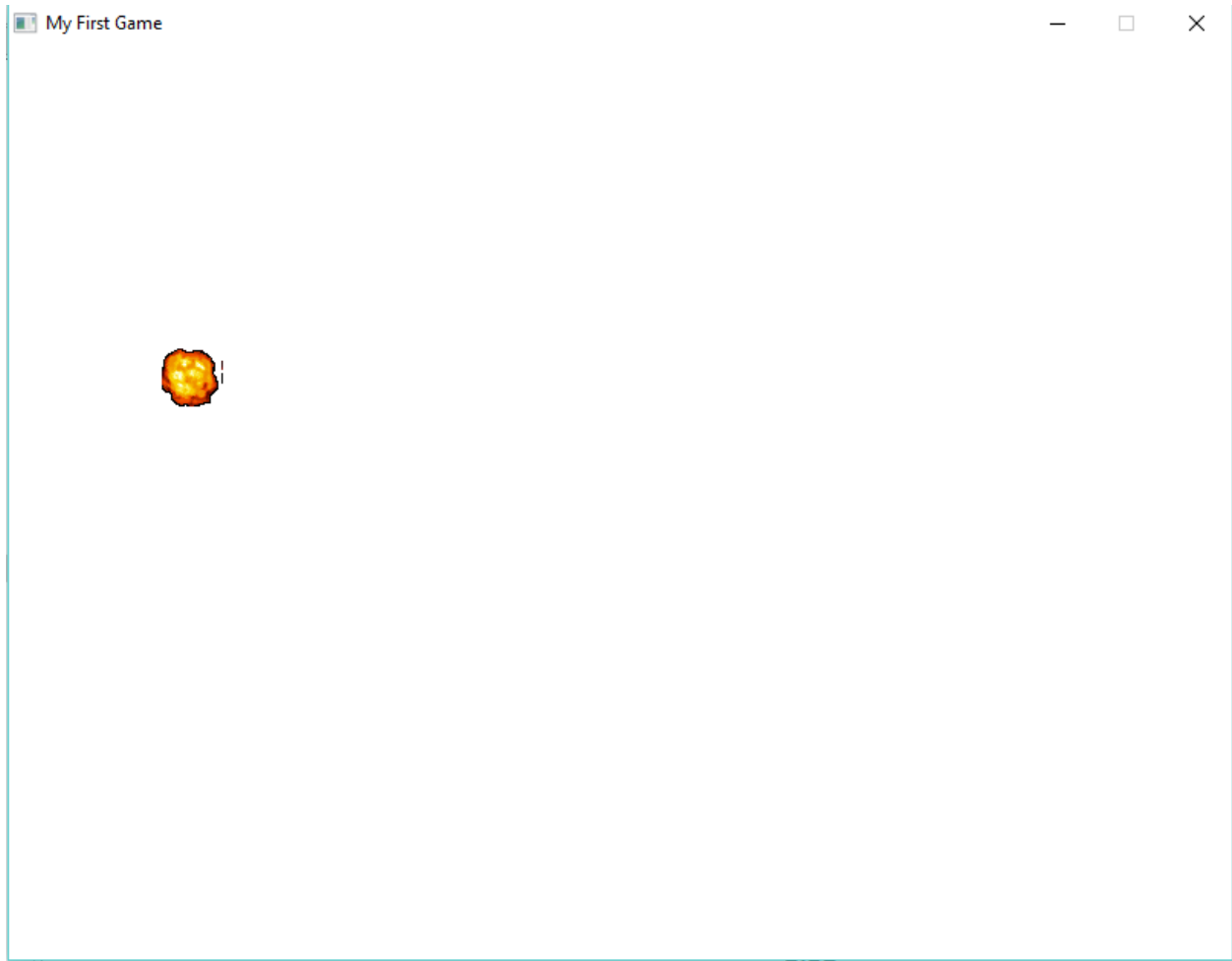
func main                # Called by the Game Engine

oGame = New Game          # Create the Game Object
{
    title = "My First Game"

    animate {
        file = "images/fire.png"
        x = 100
        y = 200
        framewidth = 40
        height = 42
        nStep = 3          # Used for delay
        transparent = true
        state = func oGame,oSelf { # Called by engine each frame
            oSelf {
                nStep--
                if nStep = 0
                    nStep = 3
                    if frame < 13      # we have 13 frames in animation
                        frame++      # move to next frame
                    else
                        oGame.remove(oself.nIndex) # remove object
                    ok
                ok
            }
        }
    }
}

}                            # Start the Events Loop

```



53.18 Using the Game Engine - Animation and Functions

```
Load "gameengine.ring"  # Give Control to the Game Engine

func main                # Called by the Game Engine

  oGame = New Game       # Create the Game Object
  {
    title = "My First Game"
    for x = 70 to 700 step 50
      for y = 70 to 500 step 50
        showfire(oGame,x,y)
      next
    next
  }                      # Start the Events Loop

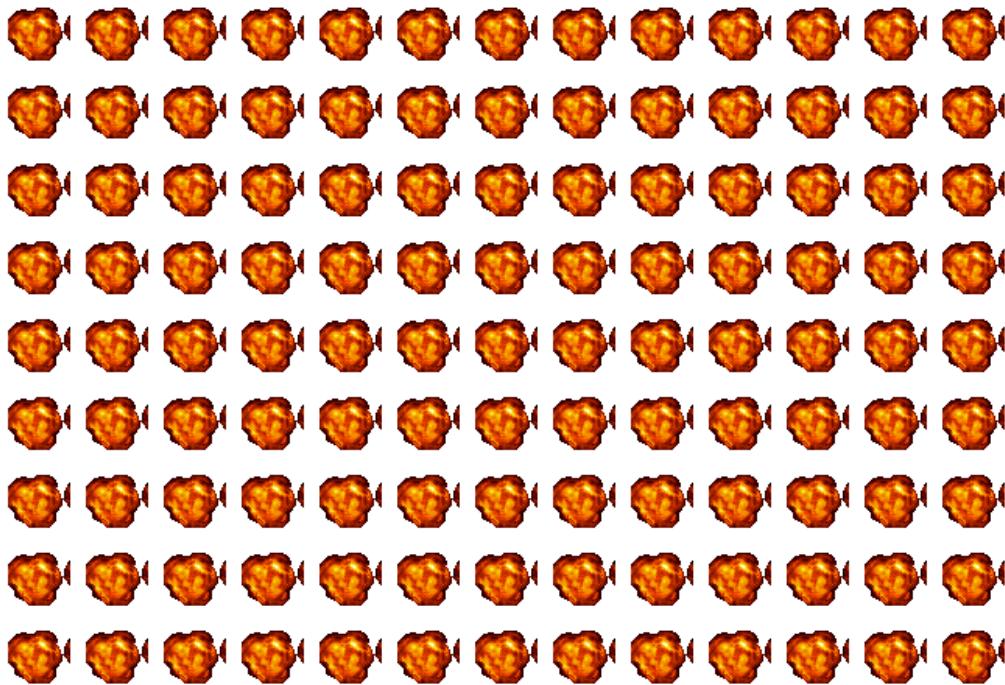
func showfire oGame,nX,nY
  oGame {
    animate {
      file = "images/fire.png"
      x = nX
```

```

y = nY
framewidth = 40
height = 42
nStep = 3      # Used for delay
transparent = true
state = func oGame,oSelf { # Called by engine each frame
  oSelf {
    nStep--
    if nStep = 0
      nStep = 3
      if frame < 13      # we have 13 frames in animation
        frame++      # move to next frame
      else
        frame=1
      ok
    }
  }
}

```

My First Game



53.19 Using the Game Engine - Sprite - Automatic Movement using Keyboard

```

Load "gameengine.ring"                # Give control to the game engine

func main                             # Called by the Game Engine

    oGame = New Game                  # Create the Game Object
    {
        title = "My First Game"
        sprite
        {
            type = GE_TYPE_PLAYER      # Just for our usage
            x=400 y=400 width=100 height=100
            file = "images/player.png"
            transparent = true
            Animate=false
            Move=true                  # we can move it using keyboard arrows
            Scaled=true
        }
    }
    }                                  # Start the Events Loop

```

My First Game



53.20 Using the Game Engine - Sprite - Keypress event

```

Load "gameengine.ring"           # Give control to the game engine

func main                        # Called by the Game Engine

    oGame = New Game             # Create the Game Object
    {
        title = "My First Game"
        sprite
        {
            type = GE_TYPE_PLAYER      # Just for our usage
            x=400 y=400 width=100 height=100
            file = "images/player.png"
            transparent = true
            Animate=false
            Move=false                # Custom Movement
            Scaled=true
            keypress = func oGame,oSelf,nKey {
                oSelf {
                    Switch nKey
                    on KEY_LEFT
                        x -= 10
                    on KEY_RIGHT
                        x += 10
                    on KEY_UP
                        y -= 10
                    on KEY_DOWN
                        y += 10
                    off
                }
            }
        }
    }

    # Start the Events Loop

```

53.21 Using the Game Engine - Sprite - Mouse event

```

Load "gameengine.ring"           # Give control to the game engine

func main                        # Called by the Game Engine

    oGame = New Game             # Create the Game Object
    {
        title = "My First Game"
        sprite
        {
            type = GE_TYPE_PLAYER      # Just for our usage
            x=400 y=400 width=100 height=100
            file = "images/player.png"
            transparent = true
            Animate=false
            Move=false                # Custom Movement
            Scaled=true
            keypress = func oGame,oSelf,nKey {
                oSelf {

```

```

        Switch nKey
        on KEY_LEFT
            x -= 10
        on KEY_RIGHT
            x += 10
        on KEY_UP
            y -= 10
        on KEY_DOWN
            y += 10
        off
    }
}
mouse = func oGame,oSelf,nType,aMouseList {
    if nType = GE_MOUSE_UP
        oSelf {
            x = aMouseList[GE_MOUSE_X]
            y = aMouseList[GE_MOUSE_Y]
        }
    ok
}
}

# Start the Events Loop

```

53.22 Using the Game Engine - Sprite - State event

```

Load "gameengine.ring" # Give control to the game engine

func main # Called by the Game Engine

oGame = New Game # Create the Game Object
{
    title = "My First Game"
    sprite
    {
        type = GE_TYPE_PLAYER # Just for our usage
        x=400 y=400 width=100 height=100
        file = "images/player.png"
        transparent = true
        Animate=false
        Move=false # Custom Movement
        Scaled=true
        keypress = func oGame,oSelf,nKey {
            oSelf {
                Switch nKey
                on KEY_LEFT
                    x -= 10
                on KEY_RIGHT
                    x += 10
                on KEY_UP
                    y -= 10
                on KEY_DOWN
                    y += 10
                off
            }
        }
    }
    mouse = func oGame,oSelf,nType,aMouseList {

```

```

        if nType = GE_MOUSE_UP
            oSelf {
                x = aMouseList[GE_MOUSE_X]
                y = aMouseList[GE_MOUSE_Y]
            }
        ok
    }
    state = func oGame,oSelf {
        oself {
            if x < 0 x = 0 ok
            if y < 0 y = 0 ok
            if x > ogame.width-width
                x= ogame.width - width ok
            if y > ogame.height-height
                y=ogame.height - height ok
        }
    }
}

# Start the Events Loop

```

53.23 Using the Game Engine - Animate - Events

```

Load "gameengine.ring"           # Give control to the game engine

func main                        # Called by the Game Engine

    oGame = New Game             # Create the Game Object
    {
        title = "My First Game"

        animate {

            file = "images/fbbird.png"
            x = 10
            y = 10
            framewidth = 20
            scaled = true
            height = 50
            width = 50
            nStep = 3
            transparent = true

            state = func oGame,oSelf {
                oSelf {

                    # Animation
                    nStep--
                    if nStep = 0
                        nStep = 3
                        if frame < 3
                            frame++
                        else
                            frame=1
                    ok
                }
            }
        }
    }
    ok

```

```

                                # Move Down
                                y += 3
                                if y > 550 y=550 ok

                                }

                                }

    keypress = func ogame,oself,nKey {
        oself {
            if nkey = key_space
                y -= 55
                if y<=0 y=0 ok

            ok

        }
    }

    mouse = func ogame,oself,nType,aMouseList {
        if nType = GE_MOUSE_UP
            cFunc = oself.keypress
            call cFunc(oGame,oSelf,Key_Space)

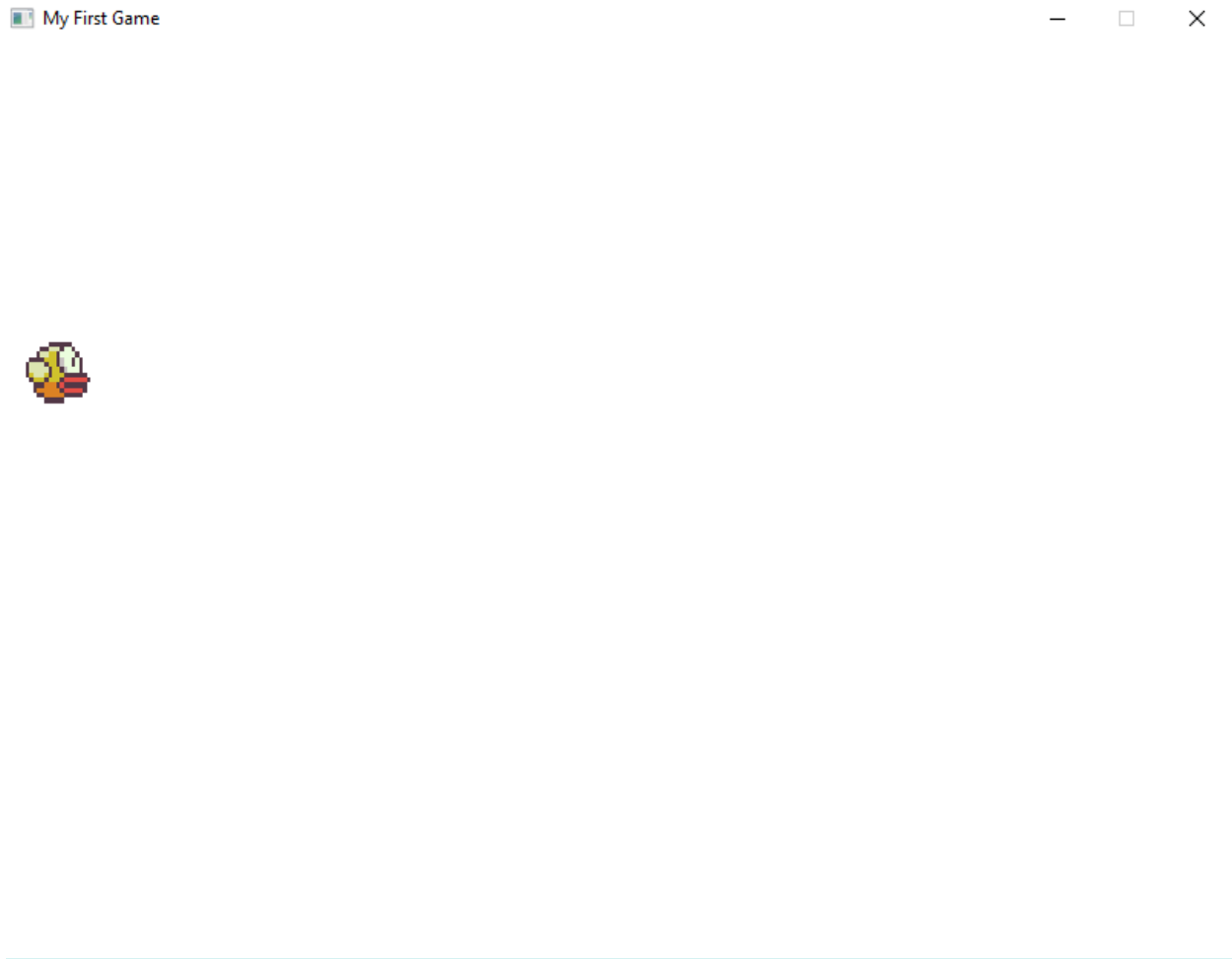
        ok

    }

}                                # Start the Events Loop

```

Screen Shot:



53.24 Using the Game Engine - Map

```
Load "gameengine.ring"           # Give control to the game engine

func main                        # Called by the Game Engine

oGame = New Game                 # Create the Game Object
{
    title = "My First Game"

    Map {

        blockwidth = 80
        blockheight = 80

        aMap = [
            [0,0,0,0,0,0,0,0,0,1,0,0,0,3,0,0,0,1,0,0,0,0,0,0,1,0,0,0],
            [0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0],
            [0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,2,0,0,0,1,0,0,0],
            [0,0,0,0,0,0,0,0,0,1,0,0,0,2,0,0,0,3,0,0,0,1,0,0,0,1,0,0,0],
            [0,0,0,0,0,0,0,0,0,3,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,3,0,0,0],
            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0],
```

```

        [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0],
        [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0]
    ]

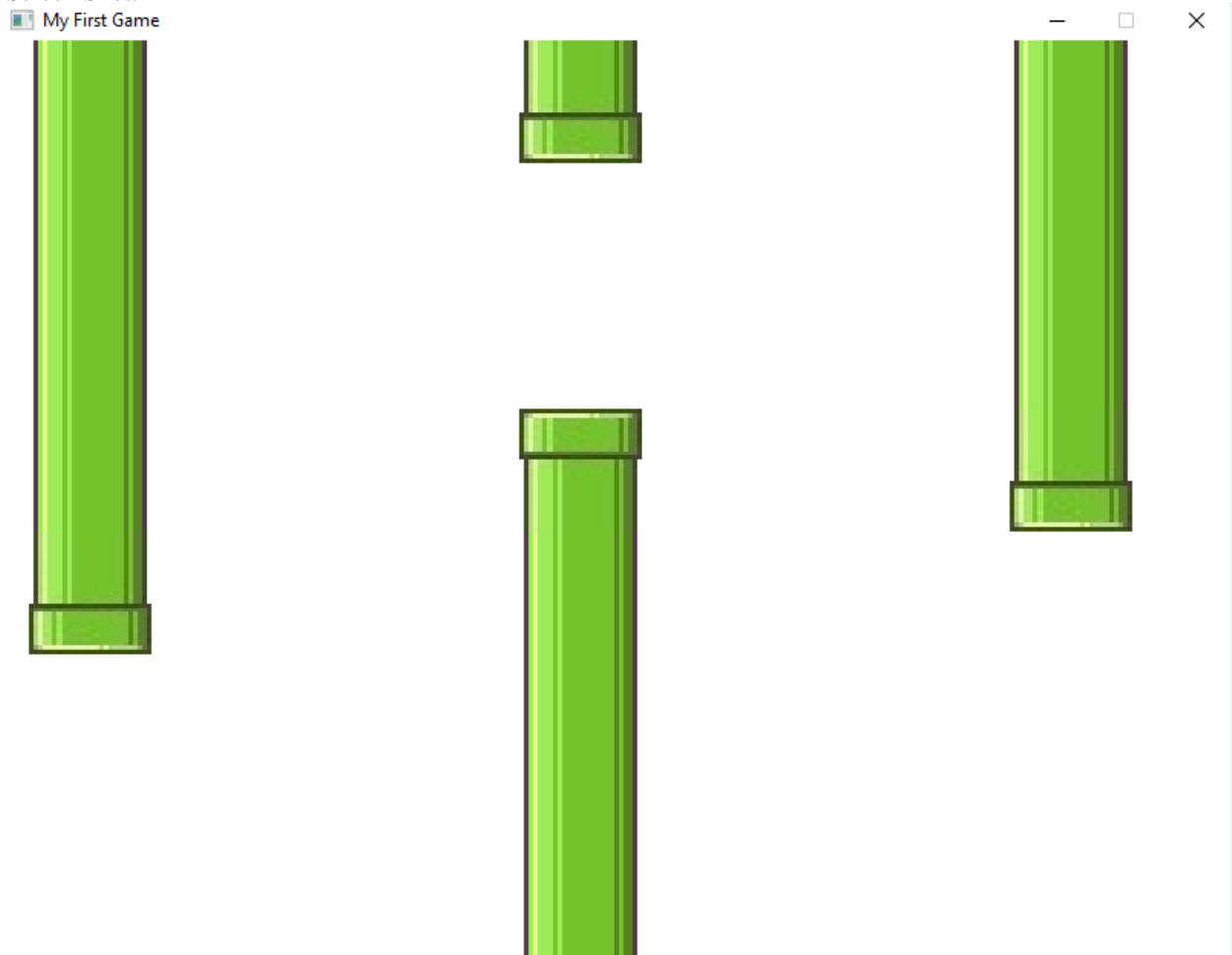
    aImages = ["images/fbwall.png",
               "images/fbwallup.png",
               "images/fbwalldown.png"]

    state = func oGame,oSelf {
        oSelf {
            x -= 3
            if x < - 2100  x = 0  ok
        }
    }
}

# Start the Events Loop

```

Screen Shot:



53.25 Using the Game Engine - Map Events

```

Load "gameengine.ring"          # Give control to the game engine

func main                      # Called by the Game Engine

oGame = New Game               # Create the Game Object
{
    title = "My First Game"

    Map {

        blockwidth = 80
        blockheight = 80

        aMap = [
            [0,0,0,0,0,0,0,0,0,0,1,0,0,0,3,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0],
            [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0],
            [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,2,0,0,0,1,0,0,0],
            [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,2,0,0,0,0,3,0,0,0,1,0,0,0,1,0,0,0],
            [0,0,0,0,0,0,0,0,0,0,3,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,3,0,0,0],
            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0],
            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0],
            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0],
            [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0]
        ]

        aImages = ["images/fbwall.png",
                   "images/fbwallup.png",
                   "images/fbwalldown.png"]

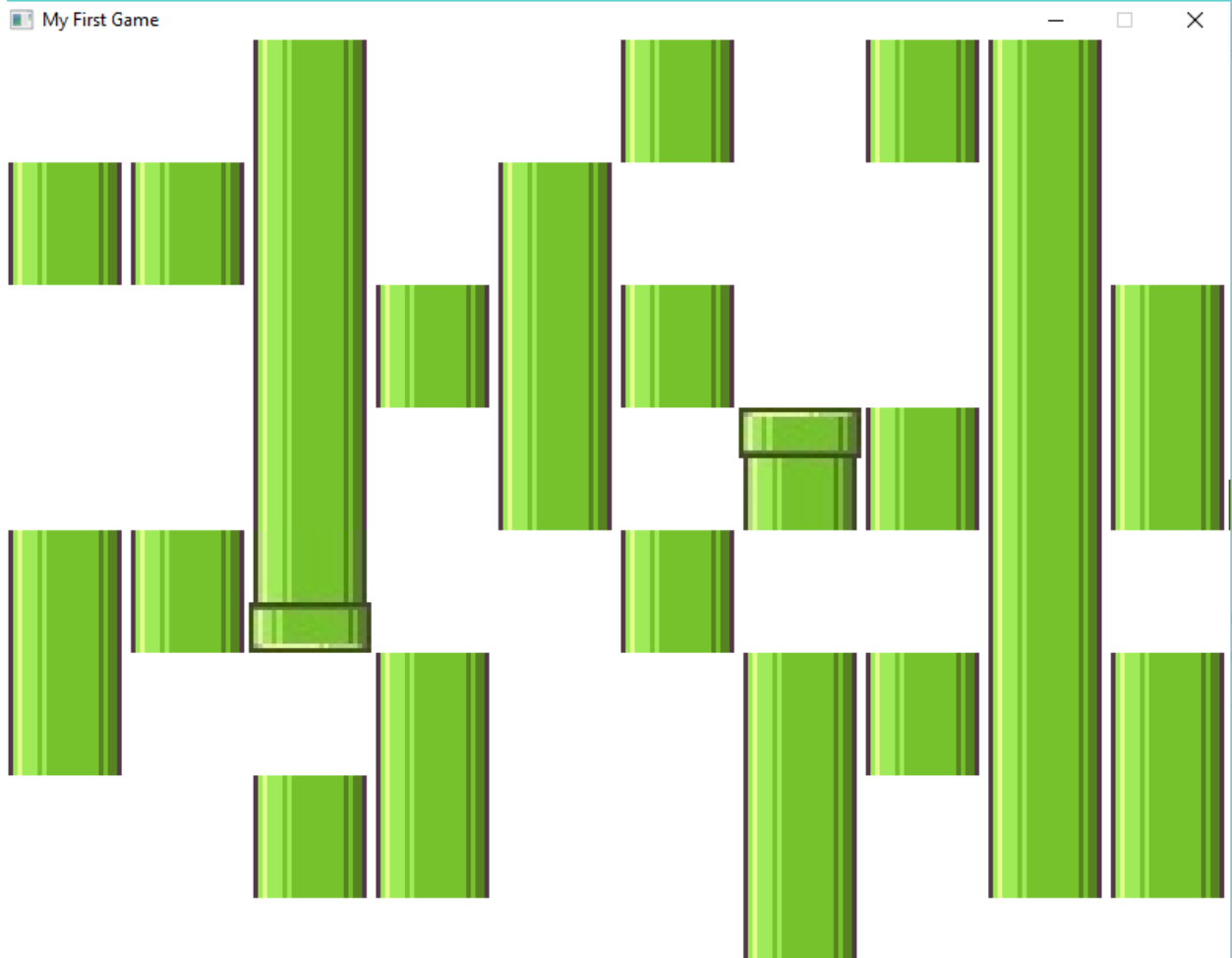
        state = func oGame,oSelf {
            oSelf {
                x -= 3
                if x < - 2100    x = 0    ok
            }
        }

        mouse = func ogame,oself,nType,aMouseList {
            if nType = GE_MOUSE_UP
            oSelf {
                mX = aMouseList[GE_MOUSE_X]
                mY = aMouseList[GE_MOUSE_Y]
                nValue = GetValue(mX,mY)
                nRow = GetRow(mX,mY)
                nCol = GetCol(mX,mY)
                Switch nValue
                On 1  aMap[nRow][nCol] = 0
                On 2  aMap[nRow][nCol] = 0
                On 3  aMap[nRow][nCol] = 0
                On 0  aMap[nRow][nCol] = 1
                Off
            }
            ok
        }
    }
}

# Start the Events Loop

```


Screen Shot:



53.26 Using the Game Engine - Object and Drawing

We can use the `Object` keyword (defined by the game engine) to create objects from the `GameObject` class.

Example:

```
Load "gameengine.ring"                                # Give control to the game engine

func main                                              # Called by the Game Engine

    oGame = New Game                                  # Create the Game Object
    {
        title = "My First Game"
        Object {
            x = 0 y=300 width = 200 height=200
            draw = func oGame,oSelf {
                oSelf {
                    for t = 1 to 210
                        gl_draw_circle(x,y,t,
                            gl_map_rgb(t*random(255),
                                t*2,t*3),1)
```

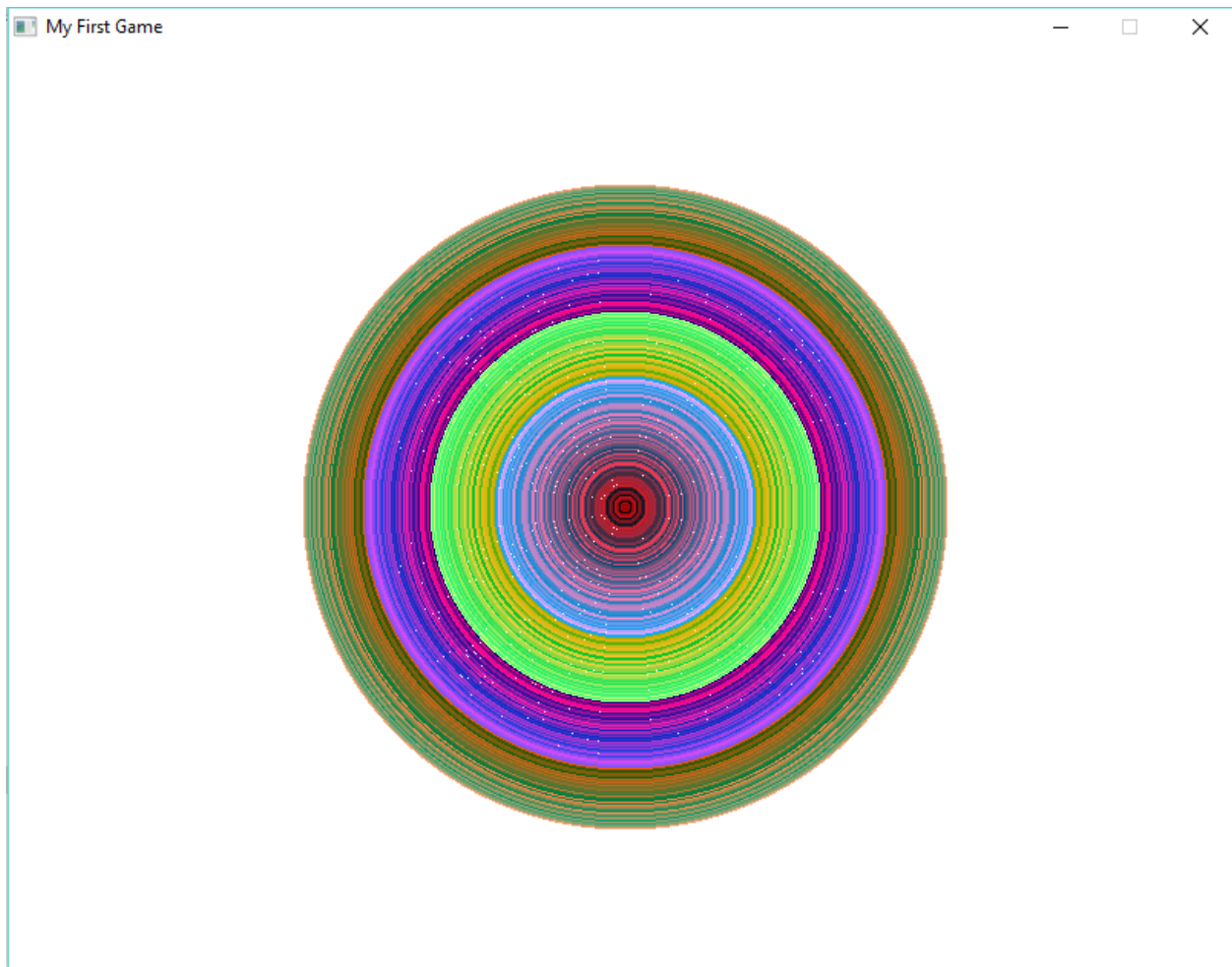
```

                                next
                                }
                                }
state = func oGame,oSelf {
    oSelf {
        if x <= 800
            x+= 3
        else
            x=0
        ok
    }
}
keypress = func oGame,oSelf,nKey {
    oSelf {
        Switch nKey
        on KEY_LEFT
            x -= 10
        on KEY_RIGHT
            x += 10
        on KEY_UP
            y -= 10
        on KEY_DOWN
            y += 10
        off
    }
}

}
                                # Start the Events Loop

```

Screen Shot:



Example:

```
Load "gameengine.ring"           # Give control to the game engine

func main                        # Called by the Game Engine

    oGame = New Game             # Create the Game Object
    {
        title = "My First Game"
        Object {
            x = 400 y=300 width = 200 height=200
            draw = func oGame,oSelf {
                oSelf {
                    for t = 1 to 210
                        gl_draw_rectangle(x+t,y+t,
                            x+t*2,y+t*2,
                            gl_map_rgb(t*random(255),
                                t*2,t*3),1)
                        gl_draw_rectangle(x+t*2,y+t*2,
                            x-t*2,y-t*2,
                            gl_map_rgb(t*random(255),
                                t*2,t*3),1)
                    next
                }
            }
        }
    }
```

```

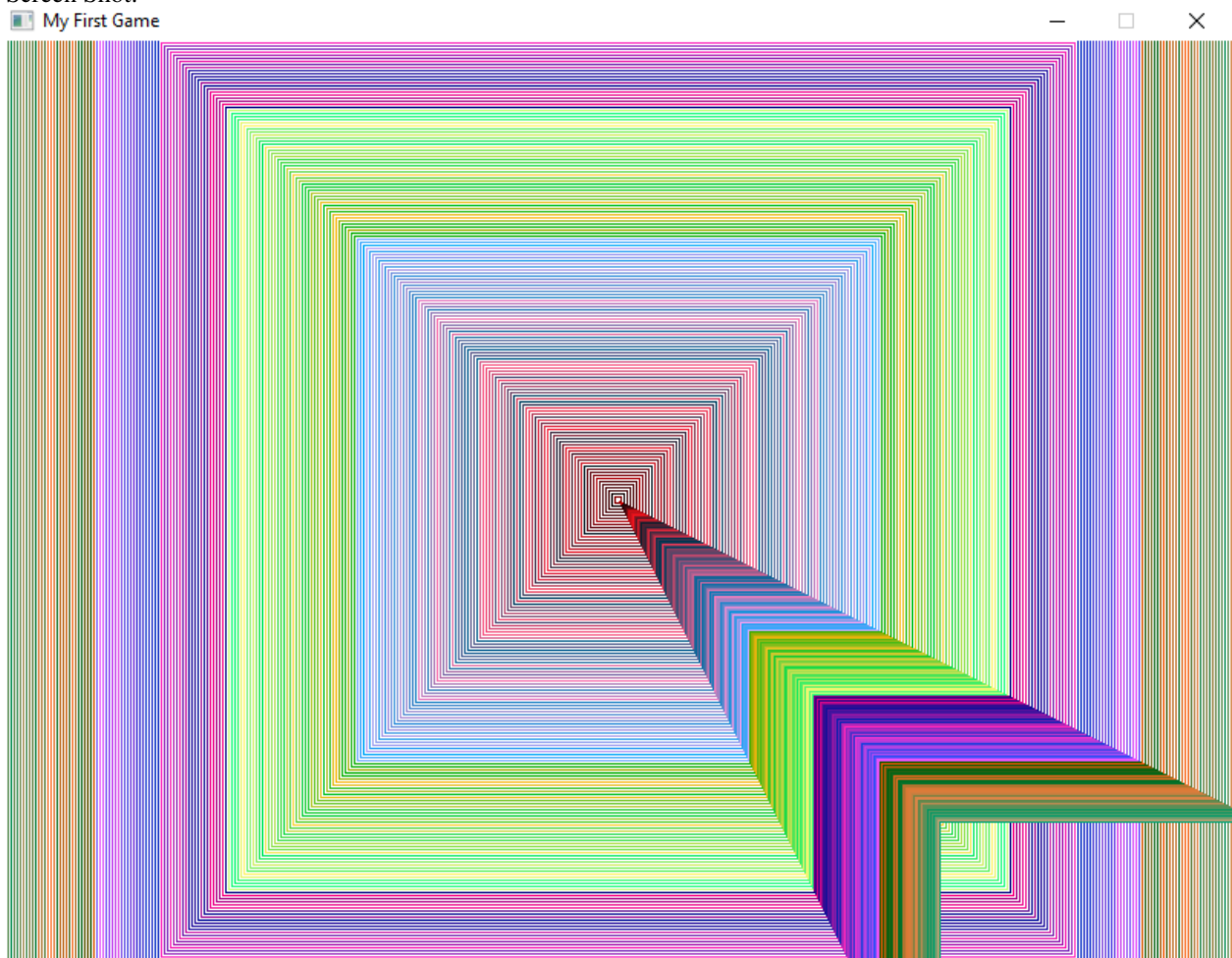
    keypress = func oGame,oSelf,nKey {
        oSelf {
            Switch nKey
            on KEY_LEFT
                x -= 10
            on KEY_RIGHT
                x += 10
            on KEY_UP
                y -= 10
            on KEY_DOWN
                y += 10
            off
        }
    }

}

# Start the Events Loop

```

Screen Shot:



53.27 Stars Fighter Game

The Stars Fighter source code

```

# The Ring Standard Library
# Game Engine for 2D Games
# 2016, Mahmoud Fayed <msfclipper@yahoo.com>

oGameState = NULL

load "gameengine.ring"

func main

  oGame = New Game

  while true

    oGameState = new GameState

    oGame {
      title = "Stars Fighter!"
      sprite
      {
        file = "images/menu1.jpg"
        x = 0 y=0 width=800 height = 600 scaled = true animate = false
        keypress = func ogame,oself,nKey {
          if nkey = key_esc or nKey = GE_AC_BACK
            ogame.shutdown()
          but nKey = key_space
            oGameState.startplay=true
            ogame.shutdown=true
          ok
        }
      }
      mouse = func ogame,oself,nType,aMouseList {
        if nType = GE_MOUSE_UP
          oGameState.startplay=true
          ogame.shutdown=true
        ok
      }
    }
    text {
      animate = false
      size = 35
      file = "fonts/pirulen.ttf"
      text = "Stars Fighter"
      x = 10 y=50
    }
    text {
      animate = false
      size = 25
      file = "fonts/pirulen.ttf"
      text = "Version 1.0"
      x = 80 y=100
    }
    text {
      animate = false
      size = 16
      file = "fonts/pirulen.ttf"
      text = "(C) 2016, Mahmoud Fayed"
      x = 45 y=140
    }
  }

```

```

        text {
            animate = false
            size = 25
            file = "fonts/pirulen.ttf"
            text = "Press Space to start"
            x = 190 y=470
        }
        text {
            animate = false
            size = 20
            file = "fonts/pirulen.ttf"
            text = "Press Esc to Exit"
            x = 260 y=510
        }
        Sound {
            file = "sound/music1.wav"
        }
    }

    if oGameState.startplay
        oGame.refresh()
        playstart(oGame)
        oGame.refresh()
    ok

end

func playstart oGame

    oSound = New Sound {
        file = "sound/music2.wav"
    }

    while true
        play(oGame)
        if ogame.shutdown = true and oGameState.value = 0
            exit
        ok
        ogame.refresh()
    end

    oSound.Delete()

func play oGame

    oGame
    {
        FPS = 60
        FixedFPS = 120
        title = "Stars Fighter!"
        sprite
        {
            file = "images/stars.jpg"
            x = 0
            y = 0
            point = -370
            direction = ge_direction_dec
            type = ge_type_background

```

```

state = func ogame,oself {
    oself {
        if x < -350
            direction = ge_direction_inc
            point = 370
        but x = 0 and direction = ge_direction_inc
            direction = ge_direction_dec
            point = -370
        ok
    }
}

sprite
{
    file = "images/player.png"
    transparent = true
    type = ge_type_player
    x = 400 y =400 width=100 height=100
    animate=false move=true Scaled=true
    mouse = func ogame,oself,nType,aMouseList {

        if not ( aMouseList[GE_MOUSE_X] >= oSelf.x and
            aMouseList[GE_MOUSE_X] <= oSelf.x+oSelf.width and
            aMouseList[GE_MOUSE_Y] >= oSelf.y and
            aMouseList[GE_MOUSE_Y] <= oSelf.y+oSelf.height )

            if nType = GE_MOUSE_DOWN
                if aMouseList[1] < oSelf.X # left
                    oSelf.X -= 100
                else
                    oSelf.X += 100
                ok
                if aMouseList[2] < oSelf.Y # up
                    oSelf.Y -= 100
                else
                    oSelf.Y += 100
                ok
            ok

        else
            if nType = GE_MOUSE_UP
                cFunc = oself.keypress
                call cFunc(oGame,oSelf,Key_Space)
            ok
        ok
    }
}

keypress = func oGame,oself,nkey {
    if nkey = key_space
        ogame {
            sprite {
                type = ge_type_fire
                file = "images/rocket.png"
                transparent = true
                x = oself.x + 30
                y = oself.y - 30
                width = 30
                height = 30
                point = -30
            }
        }
    }
}

```

```

nstep = 20
direction = ge_direction_decvertical
state = func oGame,oSelf {
    for x in oGame.aObjects
        if x.type = ge_type_enemy
            if oself.x >= x.x and oself.y >= x.y and
               oself.x <= x.x + x.width and
               oself.y <= x.y + x.height
                showfire(oGame,x.x+40,x.y+40)
                ogame.remove(x.nindex)
                oGameState.score+=10
                oGameState.enemies--
                checkwin(oGame)
                exit
            ok
        ok
    next
}

}

but nkey = key_esc or nKey = GE_AC_BACK ogame.shutdown()
ok
}

state = func oGame,oSelf {
    oself {
        if x < 0 x = 0 ok
        if y < 0 y = 0 ok
        if x > ogame.screen_w-width x= ogame.screen_w - width ok
        if y > ogame.screen_h-height y=ogame.screen_h-height ok
    }
}

}

for g = 1 to oGameState.enemies
    sprite
    {
        type = ge_type_enemy
        file = "images/enemy.png"
        transparent = true
        x = g*random(50) y =g width=100 height=100
        animate=true Scaled=true
        direction = ge_direction_random
        state = func oGame,oSelf {
            oself {
                if x < 0 x = 0 ok
                if y < 0 y = 0 ok
                if x > ogame.screen_w-width x= ogame.screen_w - width ok
                if y > ogame.screen_h-height y=ogame.screen_h-height ok
            }
            if random(100) = 1
                ogame {
                    sprite {
                        type = ge_type_fire
                        file = "images/rocket2.png"
                        transparent = true
                        x = oself.x + 30
                        y = oself.y + oself.height+ 30
                        width = 30
                        height = 30
                    }
                }
            }
        }
    }
}

```



```

        point = ogame.screen_h+30
        nstep = 10
        direction = ge_direction_incvertical
        state = func oGame,oSelf {
            x = oGame.aObjects[oGameState.playerindex]
            if oself.x >= x.x and oself.y >= x.y and
                oself.x <= x.x + x.width and
                oself.y <= x.y + x.height
                if oGameState.value > 0
                    oGameState.value-=10
                ok
            ogame.remove(oself.nindex)
            checkgameover(oGame)
        }
    }
}

ok
}

}
next
text {
    size = 30
    file = "fonts/pirulen.ttf"
    text = "Destroy All Enemies!"
    nstep = 3
    color = GE_COLOR_GREEN
    x = 100 y=50
    direction = ge_direction_incvertical
    point = 500
}
text {
    animate = false
    point = 400
    size = 30
    file = "fonts/pirulen.ttf"
    text = "Score : " + oGameState.score
    x = 500 y=10
    state = func oGame,oSelf { oSelf { text = "Score : " + oGameState.score } }
}
text {
    animate = false
    point = 400
    size = 30
    file = "fonts/pirulen.ttf"
    text = "Energy : " + oGameState.value
    x = 500 y=50
    state = func oGame,oSelf { oSelf { text = "Energy : " + oGameState.value } }
}
text {
    animate = false
    point = 400
    size = 30
    file = "fonts/pirulen.ttf"
    text = "Level : " + oGameState.level
    x = 500 y=90
}
}

```

```

func checkwin ogame
  if oGameState.gameresult return ok
  if oGameState.enemies = 0
    oGameState.gameresult = true
    oGame {
      if oGameState.level < 30
        text {
          point = 400
          size = 30
          file = "fonts/pirulen.ttf"
          text = "Level Completed!"
          nStep = 3
          x = 500 y=10
          state = func ogame,oself {
            if oself.y >= 400
              ogame.shutdown = true
              oGameState.level++
              oGameState.enemies = oGameState.level
              oGameState.gameresult = false

              ok
            }
          }
        }
      else
        text {
          point = 400
          size = 30
          nStep = 3
          file = "fonts/pirulen.ttf"
          text = "You Win !!!"
          x = 500 y=10
          state = func ogame,oself {
            if oself.y >= 400
              ogame.shutdown = true
              oGameState.value = 0

              ok
            }
          }
        }
      ok
    }
  ok

func checkgameover ogame
  if oGameState.gameresult return ok
  if oGameState.value <= 0
    oGameState.gameresult = true
    oGame {
      text {
        point = 400
        size = 30
        nStep = 3
        file = "fonts/pirulen.ttf"
        text = "Game Over !!!"
        x = 500 y=10
        state = func ogame,oself {
          if oself.y >= 400
            ogame.shutdown = true

            ok
        }
      }
    }
  
```

```

        }
    }
    showfire(oGame,oGame.aObjects[oGameState.PlayerIndex].x+40,
            oGame.aObjects[oGameState.PlayerIndex].y+40)
    oGame.aObjects[oGameState.PlayerIndex].enabled = false
    oGame.remove(oGameState.PlayerIndex)

    ok

func showfire oGame,nX,nY
oGame {
    animate {
        file = "images/fire.png"
        x = nX
        y = nY
        framewidth = 40
        height = 42
        nStep = 3
        transparent = true
        state = func oGame,oSelf {
            oSelf {
                nStep--
                if nStep = 0
                    nStep = 3
                    if frame < 13
                        frame++
                    else
                        frame=1
                        oGame.remove(oself.nIndex)
                ok
            }
        }
    }
}

class gamestate
score = 0
level = 1
enemies = 1
value = 100
playerindex = 2
gamerresult = false
startplay=false

```

Screen Shot:



53.28 Flappy Bird 3000 Game

The Flappy Bird 3000 Game source code

```
# The Ring Standard Library
# Game Engine for 2D Games
# 2016, Mahmoud Fayed <msfclipper@yahoo.com>

oGameState = NULL

Load "gameengine.ring"

func main

    oGame = New Game

    while true

        oGameState = New GameState

        oGame {
```

```

title = "Flappy Bird 3000"
sprite
{
    file = "images/fbback.png"
    x = 0 y=0 width=800 height = 600 scaled = true animate = false
    keypress = func ogame,oself,nKey {
        if nkey = key_esc or nKey = GE_AC_BACK
            ogame.shutdown()
        but nKey = key_space
            oGameState.startplay=true
            ogame.shutdown=true
        ok
    }
    mouse = func ogame,oself,nType,aMouseList {
        if nType = GE_MOUSE_UP
            cFunc = oself.keypress
            call cFunc(oGame,oSelf,Key_Space)
        ok
    }
}
text {
    animate = false
    size = 35
    file = "fonts/pirulen.ttf"
    text = "Flappy Bird 3000"
    x = 150 y=50
}
text {
    animate = false
    size = 25
    file = "fonts/pirulen.ttf"
    text = "Version 1.0"
    x = 280 y=100
}
text {
    animate = false
    size = 16
    file = "fonts/pirulen.ttf"
    text = "(C) 2016, Mahmoud Fayed"
    x = 245 y=140
}

text {
    animate = false
    size = 25
    file = "fonts/pirulen.ttf"
    text = "To Win Get Score = 3000"
    x = 150 y=270
}

text {
    animate = false
    size = 25
    file = "fonts/pirulen.ttf"
    text = "Press Space to start"
    x = 190 y=470
}
text {

```

```

        animate = false
        size = 20
        file = "fonts/pirulen.ttf"
        text = "Press Esc to Exit"
        x = 260 y=510
    }

    animate {
        file = "images/fbbird.png"
        x = 200
        y = 200
        framewidth = 20
        scaled = true
        height = 50
        width = 50
        nStep = 3
        transparent = true
        animate = true
        direction = ge_direction_random
        state = func oGame,oSelf {
            oSelf {
                nStep--
                if nStep = 0
                    nStep = 3
                    if frame < 3
                        frame++
                    else
                        frame=1
                    ok
                ok
                if x <= 0 x=0 ok
                if y <= 0 y=0 ok
                if x >= 750 x= 750 ok
                if y > 550 y=550 ok
            }
        }
    }

    Sound {
        file = "sound/music2.wav"
    }
}

if oGameState.startplay
    oGame.refresh()
    playstart(oGame)
    oGame.refresh()
ok

end

func playstart oGame

    oGame {
        FPS = 60
        FixedFPS = 120
        Title = "Flappy Bird 3000"
        Sprite {

```

```

file = "images/fbback.png"
x = 0 y=0 width=800 height = 600 scaled = true animate = false
keypress = func ogame,oself,nKey {
    if nkey = key_esc or nKey = GE_AC_BACK
        ogame.shutdown()
    ok
}
}

Map {
    blockwidth = 80
    blockheight = 80
    aMap = [
        [0,0,0,0,0,0,0,0,0,0,1,0,0,0,3,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0],
        [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0],
        [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,2,0,0,0,1,0,0,0],
        [0,0,0,0,0,0,0,0,0,0,1,0,0,0,2,0,0,0,3,0,0,0,1,0,0,0,1,0,0,0],
        [0,0,0,0,0,0,0,0,0,0,3,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,3,0,0,0],
        [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0],
        [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0],
        [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0]
    ]
    newmap(aMap)
    aImages = ["images/fbwall.png","images/fbwallup.png",
        "images/fbwalldown.png"]
    state = func oGame,oself {
        if oGameState.gameresult = false
            px = oGame.aObjects[3].x
            py = oGame.aObjects[3].y
            oSelf {
                x -= 3
                if x < - 2100
                    x = 0
                    newmap(aMap)
                ok
                nCol = getcol(px,0)
                if nCol=11 or nCol=15 or nCol=19 or nCol=23 or nCol=27
                    if nCol != oGameState.lastcol
                        oGameState.lastcol = nCol
                        oGameState.Score += 100
                        oGame { Sound {
                            once = true
                            file = "sound/sfx_point.wav"
                        } }
                        checkwin(oGame)
                    ok
                ok
            }
            if oSelf.getvalue(px+40,py) != 0 or
                oSelf.getvalue(px+40,py+40) != 0 or
                oSelf.getvalue(px,py) != 0 or
                oSelf.getvalue(px,py+40) != 0
                oGameState.gameresult = true
                oGame {
                    text {
                        point = 550
                        size = 30
                        nStep = 3

```

```

        file = "fonts/pirulen.ttf"
        text = "Game Over !!!"
        x = 500 y=10
        state = func ogame,oself {
            if oself.y >= 550
                ogame.shutdown = true

            ok

            if oself.y = 90
                ogame {
                    Sound {
                        once = true
                        file = "sound/sfx_die.wav"
                    }
                }

            ok
        }
    }
    Sound {
        once = true
        file = "sound/sfx_hit.wav"
    }
}

ok
ok

}

}

animate {
    file = "images/fbbird.png"
    x = 10
    y = 10
    framewidth = 20
    scaled = true
    height = 50
    width = 50
    nStep = 3
    transparent = true
    state = func oGame,oSelf {
        oSelf {
            nStep--
            if nStep = 0
                nStep = 3
                if frame < 3
                    frame++
                else
                    frame=1
            ok
        }
        ok
    }

    if not oGameState.playerwin
        oGameState.down --
        if oGameState.down = 0
            oGameState.down = 3
            oself {
                y += 25
                if y > 550 y=550 ok
            }

```



```

        ok
    ok

}
keypress = func ogame,oself,nKey {
    if oGameState.gameresult = false
        oself {
            if nkey = key_space
                y -= 55
                oGameState.down = 60
                if y<=0 y=0 ok
            ok
        }
    ok
}
mouse = func ogame,oself,nType,aMouseList {
    if nType = GE_MOUSE_UP
        cFunc = oself.keypress
        call cFunc(ogame,oself,Key_Space)
    ok
}
}

text {
    animate = false
    point = 400
    size = 30
    file = "fonts/pirulen.ttf"
    text = "Score : " + oGameState.score
    x = 500 y=10
    state = func oGame,oSelf {
        oSelf { text = "Score : " + oGameState.score }
    }
}
}

func newmap aMap
aV = [
    [1,1,3,0,0,2,1,1],
    [1,3,0,0,0,2,1,1],
    [1,1,1,3,0,2,1,1],
    [1,1,1,3,0,0,0,0],
    [0,0,0,0,2,1,1,1],
    [0,0,2,1,1,1,1,1],
    [0,0,0,2,1,1,1,1],
    [1,1,1,3,0,2,1,1],
    [1,1,1,1,1,3,0,0],
    [3,0,0,2,1,1,1,1],
    [3,0,0,2,3,0,0,2]
]
for x = 10 to 24 step 4
    aVar = aV[ (random(10)+1) ]
    for y = 1 to 8
        aMap[y][x] = aVar[y]
    next
next

```

```

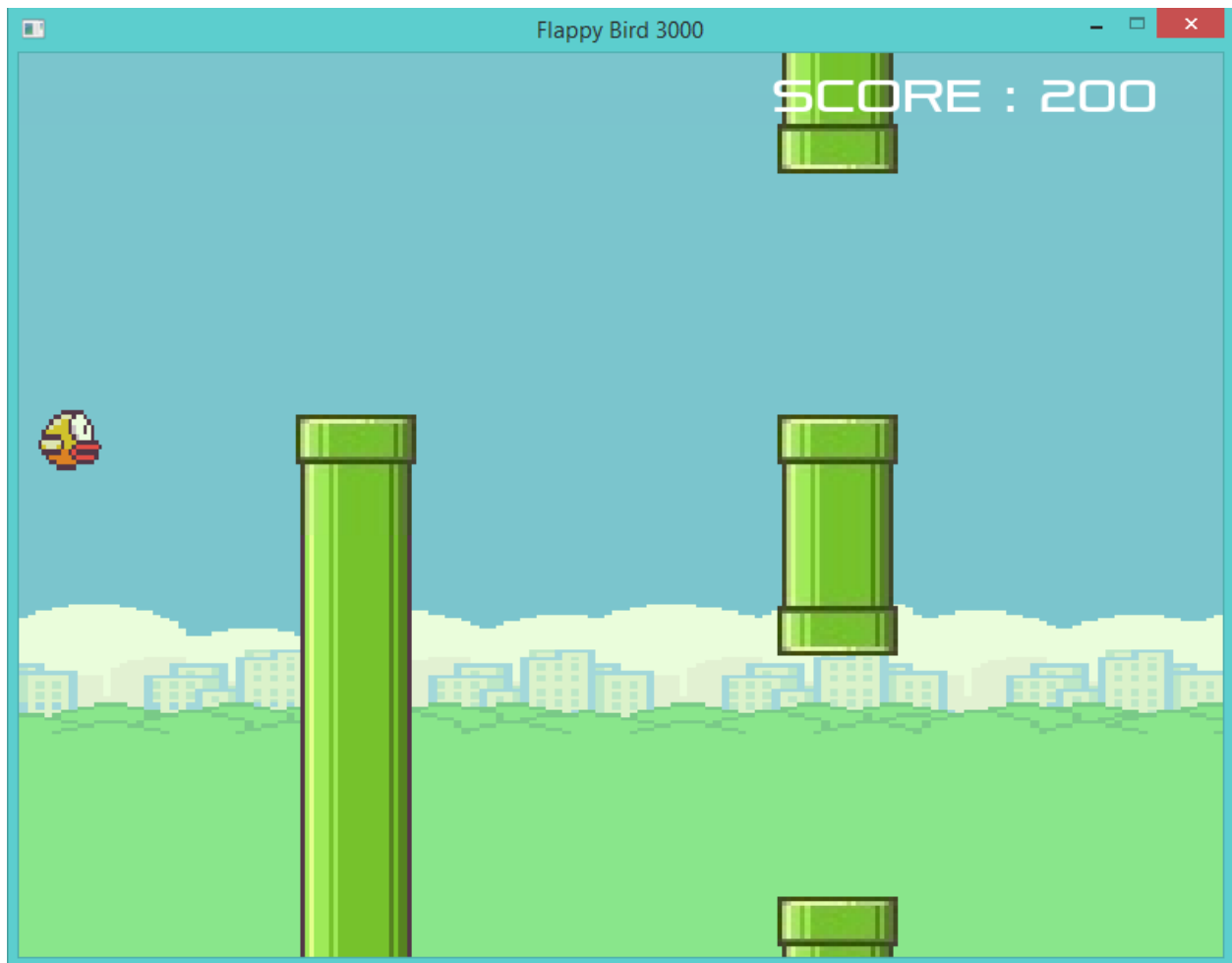
func checkwin ogame
    if oGameState.score = 3000
        oGameState.gameresult = true
        oGameState.playerwin = true
        oGame {
            text {
                point = 400
                size = 30
                nStep = 3
                file = "fonts/pirulen.ttf"
                text = "You Win !!!"
                x = 500 y=10
                state = func ogame,oself {
                    if oself.y >= 400
                        ogame.shutdown = true
                        oGameState.value = 0

                    ok
                }
            }
        }
    ok

Class GameState
    down = 3
    gameresult = false
    Score = 0
    startplay=false
    lastcol = 0
    playerwin = false

```

Screen Shot:



53.29 Super Man 2016 Game

The Super Man 2016 Game source code

```
# The Ring Standard Library
# Game Engine for 2D Games
# 2016, Mahmoud Fayed <msfclipper@yahoo.com>

oGameState = NULL

Load "gameengine.ring"

func main

    oGame = New Game

    while true

        oGameState = new GameState

        oGame {
            title = "Super Man 2016"
```

```

sprite
{
    file = "images/superman.jpg"
    x = 0 y=0 width=800 height = 600 scaled = true animate = false
    keypress = func ogame,oself,nKey {
        if nkey = key_esc or nKey = GE_AC_BACK
            ogame.shutdown()
        but nKey = key_space
            oGameState.startplay=true
            ogame.shutdown=true
        ok
    }
    mouse = func ogame,oself,nType,aMouseList {
        if nType = GE_MOUSE_UP
            oGameState.startplay=true
            ogame.shutdown=true
        ok
    }
    state = func ogame,oself {
        oself {
            if x > -500
                x-=1
                y-=1
                width +=1
                height +=4
            ok
        }
    }
}
text {
    animate = false
    size = 35
    file = "fonts/pirulen.ttf"
    text = "Super Man 2016"
    x = 20 y=30
}
text {
    animate = false
    size = 25
    file = "fonts/pirulen.ttf"
    text = "Version 1.0"
    x = 20 y=80
}
text {
    animate = false
    size = 16
    file = "fonts/pirulen.ttf"
    text = "(C) 2016, Mahmoud Fayed"
    x = 20 y=120
}
text {
    animate = false
    size = 25
    file = "fonts/pirulen.ttf"
    text = "Press Space to start"
    x = 190 y=470
}

```

```

text {
    animate = false
    size = 20
    file = "fonts/pirulen.ttf"
    text = "Press Esc to Exit"
    x = 260 y=510
}

animate {
    file = "images/superman.png"
    x = 200
    y = 200
    framewidth = 68
    scaled = true
    height = 86
    width = 60
    nStep = 10
    transparent = true
    animate = true
    direction = ge_direction_random
    state = func oGame,oSelf {
        oSelf {
            nStep--
            if nStep = 0
                nStep = 10
            if frame < 1
                frame++
            else
                frame=1
            ok
        ok
        if x <= 0 x=0 ok
        if y <= 0 y=0 ok
        if x >= 750 x= 750 ok
        if y > 550 y=550 ok
    }
}

}

Sound {
    file = "sound/music2.wav"
}

}

if oGameState.startplay
    oGame.refresh()
    playstart(oGame)
    oGame.refresh()
ok

end

func playstart oGame

oGame {
    FPS = 60
    FixedFPS = 15
    Title = "Super Man 2016"

```

```

Sprite {
  file = "images/supermancity.jpg"
  x = 0 y=0 width=800 height = 600 scaled = true animate = false
}
Map {
  blockwidth = 80
  blockheight = 80
  aMap = [
    [0,0,0,4,4,4,0,0,0,1,0,0,0,1,4,4,0,1,0,0,0,0,4,4,0,1,4,
4,4,0,0,0,0,0,0,0,0,0,0,0,0,2,0,1,0,0,0,1,0,0,0,1,0,3,3,3,5,3,3,3,3,0],
    [0,0,4,0,4,0,4,0,0,0,1,0,0,0,3,4,4,4,1,0,0,0,0,4,4,0,1,4,
4,4,0,0,4,4,4,4,4,4,4,1,4,1,0,0,0,1,0,0,0,1,0,4,4,4,4,4,4,4,0],
    [0,0,0,4,4,4,0,0,0,1,0,0,0,4,4,4,4,1,0,0,0,0,0,0,0,3,4,
4,4,0,0,4,0,0,0,0,0,0,4,2,0,0,4,1,4,1,4,2,4,1,0,2,0,1,0,4,4,4,4,4,4,4,0],
    [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,
0,0,0,0,4,4,4,4,4,4,4,4,1,0,0,4,1,4,1,4,1,4,1,0,1,0,1,0,2,2,2,2,2,2,2,0],
    [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,2,0,0,0,0,0,2,0,3,0,0,0,1,4,1,4,1,4,1,0,1,0,1,0,1,0,0,0,0,0,0,0],
    [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,2,0,0,0,1,0,0,0,2,0,0,0,0,0,0,
0,0,0,0,1,0,0,0,0,0,1,4,3,4,1,4,3,0,1,0,3,0,1,0,0,0,0,0,0,0,0],
    [0,0,2,0,0,2,0,0,2,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,0,0,
0,0,0,0,1,0,0,0,0,0,3,0,0,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0],
    [0,0,1,0,0,1,0,0,1,3,0,0,0,1,0,0,0,3,0,0,0,1,0,0,0,0,0,0,0,0,
0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0]
  ]
  aImages = ["images/small.png", "images/smallup.png",
    "images/smalldown.png", "images/smstar.png",
    "images/smkey.png", "images/smstar2.png"]
}

sprite {
  type = ge_type_enemy
  animate = false
  file = "images/smhome.png"
  x = 5000
  y = 400
  width = 290
  height = 200
  transparent = true

  state = func oGame,oSelf {
    oself {
      x = 5000 + oGame.aObjects[2].x
      if x < 0 or x > SCREEN_W return ok
    }
    if oGameState.gameresult or oGameState.DoorKey = false return ok
    if oGame.aObjects[oGameState.playerindex].x > oself.x + 100 and
oGame.aObjects[oGameState.playerindex].y > oself.y + 50
oGameState.gameresult = true
oGame {
  sprite {
    file = "images/smwin.jpg"
    x=0 y=0 width=800 height=600
    scaled = true animate=false
    state = func ogame,oself {
      oself {
        x-=5
        y-=5

```

511

```

        but nkey = key_down and checkwall(oGame,oSelf,0,40)
            file = "images/supermandown.png"
            dottransparent()
            y += 40
            if y>=500 y=500 ok
        but nKey = key_right and checkwall(oGame,oSelf,10,0)
            file = "images/supermanright.png"
            dottransparent()
            x += 10
            if x >= 440
                if oGame.aObjects[2].x > -4500
                    oGame.aObjects[2].x -= 50
                    callenemystate(oGame)
                else
                    if x <= 750
                        if checkwall(oGame,oSelf,10,0)
                            x += 10
                        ok
                    else
                        if checkwall(oGame,oSelf,-10,0)
                            x -= 10
                        ok
                    ok
                return
            ok
            x=400
        ok
    but nKey = key_left and checkwall(oGame,oSelf,-10,0)
        file = "images/supermanleft.png"
        dottransparent()
        x -= 10
        if x <= 0
            x += 10
            if oGame.aObjects[2].x != 0
                oGame.aObjects[2].x += 50
                callenemystate(oGame)
            x += 50
        ok
    ok
    but nkey = key_esc or nKey = GE_AC_BACK
        ogame.shutdown()
    ok
}
}
ok
}
mouse = func ogame,oself,nType,aMouseList {
    if nType = GE_MOUSE_DOWN
        oGameState.moveplayer = TRUE
    But nType = GE_MOUSE_UP
        oGameState.moveplayer = FALSE
    ok
    if oGameState.moveplayer = TRUE
        if aMouseList[GE_MOUSE_X] < oSelf.X # left
            cFunc = oself.keypress
            call cFunc(oGame,oSelf,Key_left)
        else
            cFunc = oself.keypress
            call cFunc(oGame,oSelf,Key_right)

```



```

        ok
        if aMouseListener[GE_MOUSE_Y] < oSelf.Y # up
            cFunc = oself.keypress
            call cFunc(oGame,oSelf,Key_up)
        else
            cFunc = oself.keypress
            call cFunc(oGame,oSelf,Key_down)

        ok
    ok
}
}

addenemy(oGame,600)
addenemy(oGame,900)
addenemy(oGame,1550)
addenemy(oGame,2350)
addenemy(oGame,3350)
addenemy(oGame,3500)
addenemy(oGame,3670)
addenemy(oGame,3840)

text {
    animate = false
    point = 400
    size = 30
    file = "fonts/pirulen.ttf"
    text = "Score : " + oGameState.score
    x = 500 y=0
    state = func oGame,oSelf {
        oSelf { text = "Score : " + oGameState.score }
    }
}

text {
    animate = false
    point = 400
    size = 30
    file = "fonts/pirulen.ttf"
    text = "Energy : " + oGameState.value
    x = 10 y=0
    state = func oGame,oSelf { oSelf { text = "Energy : " + oGameState.value } }
}

}

func inlist nValue,aList
    for x in aList
        if x = nValue
            return true
    ok
next
return false

func checkwall oGame,oself,diffx,diffy
    alist = [1,2,3]
    return checkwall2(oGame,oself,diffx,diffy,aList)

func checkwall2 oGame,oself,diffx,diffy,aList

```

```

xPos = oSelf.x + diffx
yPos = oSelf.y + diffy
nValue = oGame.aObjects[2].getvalue(xPos,yPos)
nValue = inlist(nValue,aList)
nValue = not nValue
if nValue = 0 return nValue ok

xPos = oSelf.x + diffx
yPos = oSelf.y + diffy + oSelf.height
nValue = oGame.aObjects[2].getvalue(xPos,yPos)
nValue = inlist(nValue,aList)
nValue = not nValue
if nValue = 0 return nValue ok

xPos = oSelf.x + diffx + oSelf.width
yPos = oSelf.y + diffy
nValue = oGame.aObjects[2].getvalue(xPos,yPos)
nValue = inlist(nValue,aList)
nValue = not nValue
if nValue = 0 return nValue ok

xPos = oSelf.x + diffx + oSelf.width
yPos = oSelf.y + diffy + oSelf.height
nValue = oGame.aObjects[2].getvalue(xPos,yPos)
nValue = inlist(nValue,aList)
nValue = not nValue
if nValue = 0 return nValue ok

return nValue

func checkopenwall oGame
if oGameState.score = 900
    oGame.aObjects[2].aMap[3][10] = 3
    oGame.aObjects[2].aMap[4][10] = 0
    oGame.aObjects[2].aMap[5][10] = 0
    oGame.aObjects[2].aMap[6][10] = 0
    oGame.aObjects[2].aMap[7][10] = 0
    oGame.aObjects[2].aMap[8][10] = 0
but oGameState.score = 1800
    oGame.aObjects[2].aMap[3][18] = 3
    oGame.aObjects[2].aMap[4][18] = 0
    oGame.aObjects[2].aMap[5][18] = 0
    oGame.aObjects[2].aMap[6][18] = 0
    oGame.aObjects[2].aMap[7][18] = 0
    oGame.aObjects[2].aMap[8][18] = 0
but oGameState.score = 5500
    oGame.aObjects[2].aMap[1][44] = 0
    oGame.aObjects[2].aMap[2][44] = 0
    oGame.aObjects[2].aMap[3][44] = 2
ok

func checkgameover ogame
if oGameState.gameresult return ok
if oGameState.value <= 0
    oGameState.value = 0
    oGameState.gameresult = true
    oGame {

```

```

        text {
            point = 400
            size = 30
            nStep = 9
            file = "fonts/pirulen.ttf"
            text = "Game Over !!!"
            x = 500 y=10
            state = func ogame,oself {
                if oself.y >= 400
                    ogame.shutdown = true
                ok
            }
        }
    }
    showfire(oGame,oGame.aObjects[oGameState.PlayerIndex].x+40,
            oGame.aObjects[oGameState.PlayerIndex].y+40)
    oGame.aObjects[oGameState.PlayerIndex].enabled = false
    oGame.remove(oGameState.PlayerIndex)
ok

func showfire oGame,nX,nY
oGame {
    animate {
        file = "images/fire.png"
        x = nX
        y = nY
        framewidth = 40
        height = 42
        nStep = 3
        transparent = true
        state = func oGame,oSelf {
            oSelf {
                nStep--
                if nStep = 0
                    nStep = 3
                    if frame < 13
                        frame++
                    else
                        frame=1
                        oGame.remove(oself.nIndex)
                ok
            }
        }
    }
}

func addenemy oGame,xPos
oGame {
    lbraceend = false
    sprite {
        type = ge_type_enemy
        file = "images/smenemy.png"
        transparent = true
        x = xPos y =10 width=100 height=100
        animate=true Scaled=true
        direction = GE_DIRECTION_NOMOVE
    }
}

```

```

temp = xPos
state = func oGame,oSelf {
  oself {
    x = oSelf.temp + oGame.aObjects[2].x
    if y < 0 y = 0 ok
    if y > 100 y=100 ok
    if x > SCREEN_W or x < 0 return ok
  }

  if random(10) = 1
    if oGameState.gameresult return ok
    ogame {
      sprite {
        type = ge_type_fire
        file = "images/smrocket.png"
        scaled = true
        transparent = true
        x = oself.x + 30
        y = oself.y + oself.height+ 30
        width = 30
        height = 30
        point = ogame.screen_h+30
        nstep = 30
        direction = ge_direction_incvertical
        xvalue = oGame.aObjects[2].x
        temp = oself.x + 30 - xvalue
        state = func oGame,oSelf {
          oself { x = oSelf.temp + oGame.aObjects[2].x }
          x = oGame.aObjects[oGameState.playerindex]
          if oself.x >= x.x and oself.y >= x.y and
            oself.x <= x.x + x.width and
            oself.y <= x.y + x.height
            if oGameState.value > 0
              oGameState.value-=1000
            ok
            ogame.remove(oself.nindex)
            checkgameover(oGame)
          ok
        }
      }
    }
  ok
}

}
ogame.lbraceend = true

func checkstarskey oGame,oSelf,nValue,nRow,nCol
  switch nValue
    on 4
      oGame.aObjects[2].aMap[nRow][nCol] = 6
      oGameState.Score += 100
      checkopenwall(oGame)
      oGame { Sound {
        once = true
        file = "sound/sfx_point.wav"
      } }

```

```

    on 5
        oGame.aObjects[2].aMap[nRow][nCol] = 0
        oGameState.DoorKey = true
        oGameState.Score += 500
        checkopenwall(oGame)
        oGame { Sound {
            once = true
            file = "sound/sfx_point.wav"
        } }
    off

func checkstarskeycol oGame,oSelf
    nValue = oGame.aObjects[2].getvalue(oSelf.x,oSelf.y)
    nRow = oGame.aObjects[2].getrow(oSelf.x,oSelf.y)
    nCol = oGame.aObjects[2].getcol(oSelf.x,oSelf.y)
    checkstarskey(oGame,oSelf,nValue,nRow,nCol)

    nValue = oGame.aObjects[2].getvalue(oSelf.x+oSelf.width,oSelf.y+oSelf.height)
    nRow = oGame.aObjects[2].getrow(oSelf.x+oSelf.width,oSelf.y+oSelf.height)
    nCol = oGame.aObjects[2].getcol(oSelf.x+oSelf.width,oSelf.y+oSelf.height)
    checkstarskey(oGame,oSelf,nValue,nRow,nCol)

    nValue = oGame.aObjects[2].getvalue(oSelf.x+oSelf.width,oSelf.y)
    nRow = oGame.aObjects[2].getrow(oSelf.x+oSelf.width,oSelf.y)
    nCol = oGame.aObjects[2].getcol(oSelf.x+oSelf.width,oSelf.y)
    checkstarskey(oGame,oSelf,nValue,nRow,nCol)

    nValue = oGame.aObjects[2].getvalue(oSelf.x,oSelf.y+oSelf.height)
    nRow = oGame.aObjects[2].getrow(oSelf.x,oSelf.y+oSelf.height)
    nCol = oGame.aObjects[2].getcol(oSelf.x,oSelf.y+oSelf.height)
    checkstarskey(oGame,oSelf,nValue,nRow,nCol)

func callenemystate oGame
    for t in oGame.aObjects
        t {
            if type = GE_TYPE_ENEMY
                call state(oGame,t)
            ok
        }
    next

Class GameState

    down = 3
    gameresult = false
    Score = 0
    startplay=false
    lastcol = 0
    playerwin = false
    DoorKey = false
    playerindex = 4
    value = 1000
    moveplayer = false

```

Screen Shot:



BUILDING GAMES FOR ANDROID

In this chapter we will learn about Building RingLibSDL Games for Mobile.

So we can create packages (*.apk) for the applications that are developed using Ring Game Engine for 2D Games.

54.1 Download Requirements and Update the Android SDK

- The Android SDK Tools

<https://developer.android.com/studio/index.html>

- The Android NDK (Tested using android-ndk-r10c)

<https://developer.android.com/ndk/index.html>

- Apache Ant v1.8 or later

<http://ant.apache.org/bindownload.cgi>

- Java SE Development Kit (JDK) v6 or later

<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>

- Update the Android SDK to get the API and tools packages required for development

Tested using Android 4.4.2 (API 19)

- In Windows - Define the next Environment Variables based on your system.

1. JAVA_HOME

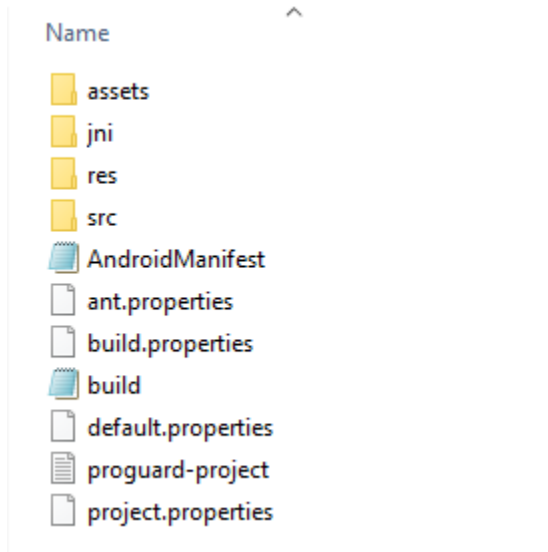
For Example : C:\Program Files (x86)\Java\jdk1.8.0_05
--

2. ANDROID_HOME

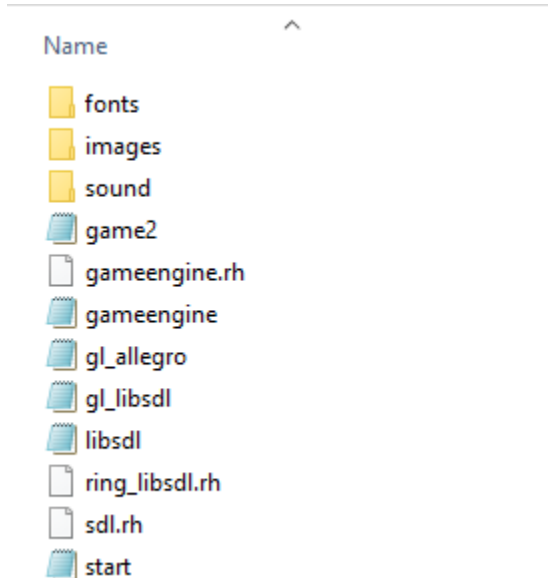
For Example : B:\mahmoud\Tools\Java-Android\adt-bundle-windows-x86-20140702\sdk
--

54.2 Project Folder

Open the project folder : ring/android/ringlibsdl/project



You can add the source code (*.ring) and Images/Sound Files to the assets folder.



You will find the Flappy Bird 3000 Game ready for building.

The execution starts from the start.ring file

```
load "game2.ring"
```

54.3 Building the project

Move to the ring/android/ringlibsdl/project folder

We can build using the next command (We need to do this for one time only).

```
ndk-build
```

Then we can create the package (*.apk) using the next command.


```
ant debug
```

USING RINGOPENGL AND RINGFREEGLUT FOR 3D GRAPHICS

In this chapter we will learn about using RingOpenGL

55.1 Samples Source (Authors)

The samples in this chapter are based on C Tutorials
from

1. <http://www.lighthouse3d.com/tutorials/glut-tutorial/>
2. <http://www.wikihow.com/Make-a-Cube-in-OpenGL>

55.2 What is RingOpenGL?

RingOpenGL contains the Ring binding to the OpenGL library

You can learn about OpenGL from : <https://www.opengl.org/>

RingOpenGL comes with support for the next versions

- OpenGL 1.1
- OpenGL 1.2
- OpenGL 1.3
- OpenGL 1.4
- OpenGL 1.5
- OpenGL 2.0
- OpenGL 2.1
- OpenGL 3.0
- OpenGL 3.2
- OpenGL 3.3
- OpenGL 4.0
- OpenGL 4.1
- OpenGL 4.2
- OpenGL 4.3

- OpenGL 4.4
- OpenGL 4.5
- OpenGL 4.6

For example, if you want to use OpenGL 2.1 then load RingOpenGL 2.1 library

```
load "opengl21lib.ring"
```

55.3 What is RingFreeGLUT?

RingFreeGLUT contains the Ring binding to the FreeGLUT library

You can learn about FreeGLUT from : <http://freeglut.sourceforge.net/>

To use the RingFreeGLUT library, Just load the library

```
load "freeglut.ring"
```

55.4 The First Window using RingFreeGLUT

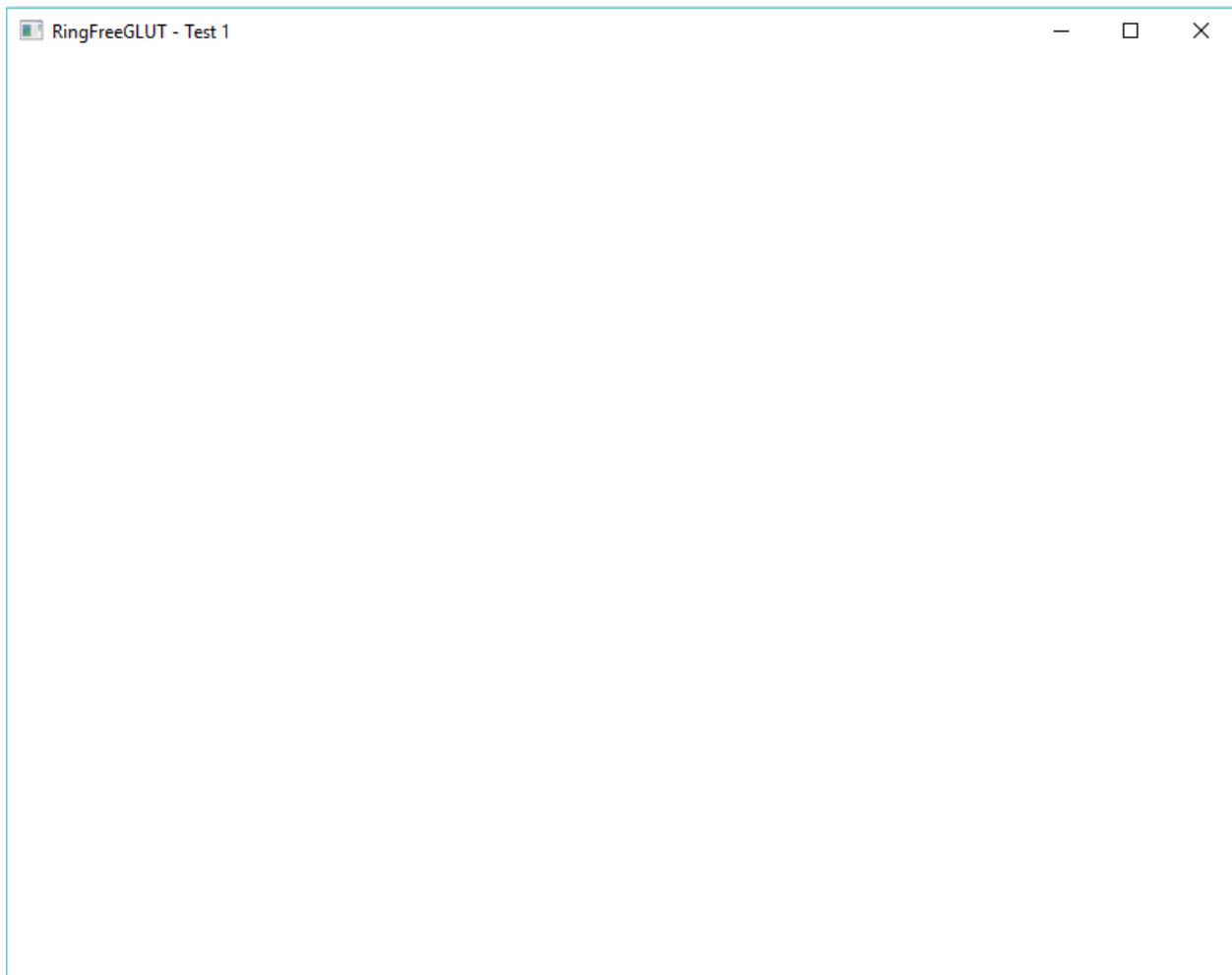
Example:

```
load "freeglut.ring"

func main
    glutInit()
    glutInitDisplayMode(GLUT_SINGLE)
    glutInitWindowSize(800, 600)
    glutInitWindowPosition(100, 10)
    glutCreateWindow("RingFreeGLUT - Test 1")
    glutDisplayFunc(:displayCode)
    glutMainLoop()

func displaycode
```

Screen Shot



55.5 Drawing using RingOpenGL

Example:

```
load "freeglut.ring"
load "opengl2llib.ring"

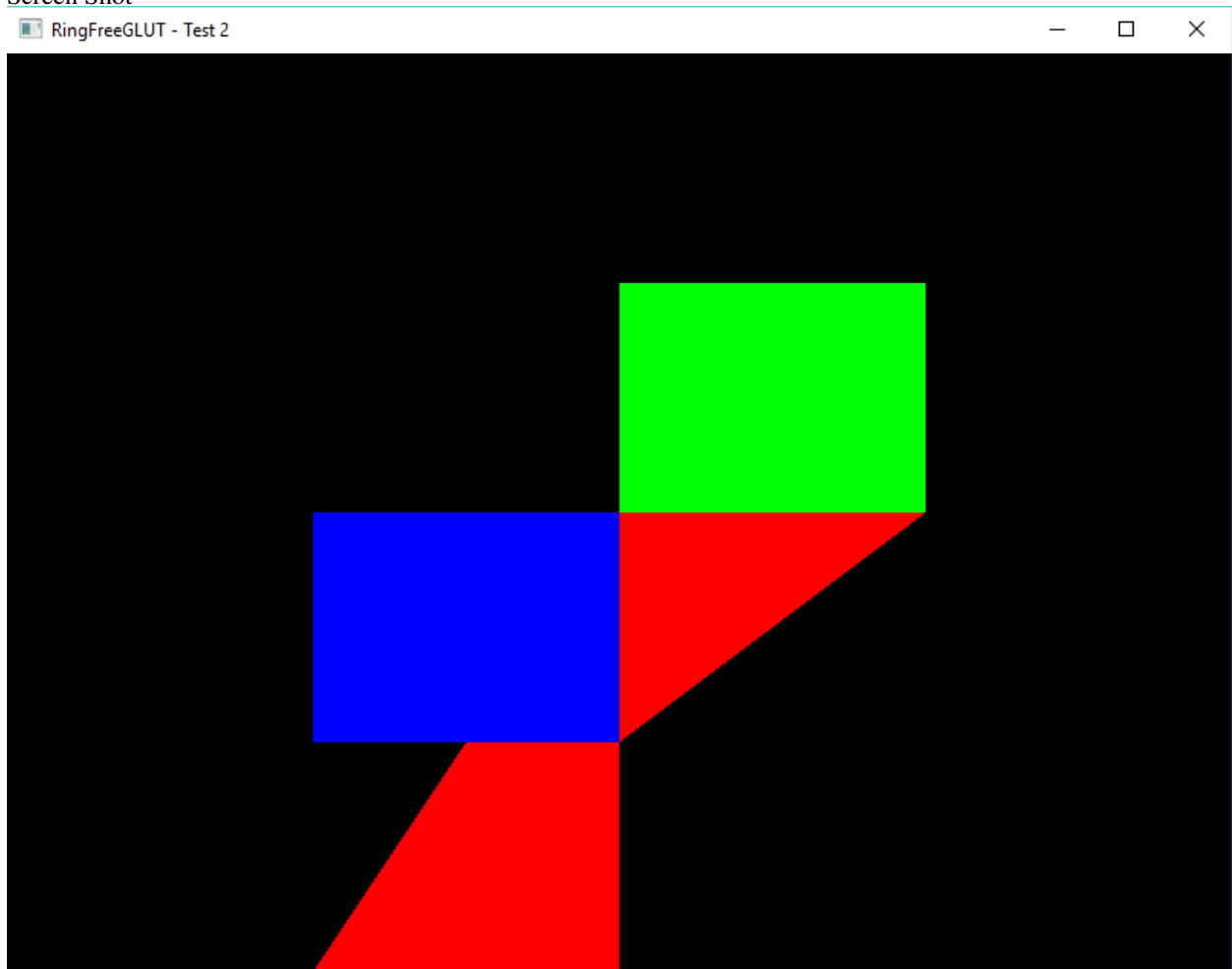
func main
    glutInit()
    glutInitDisplayMode(GLUT_SINGLE)
    glutInitWindowSize(800, 600)
    glutInitWindowPosition(100, 10)
    glutCreateWindow("RingFreeGLUT - Test 2")
    glutDisplayFunc(:displayCode)
    glutMainLoop()

func displaycode
    glClear(GL_COLOR_BUFFER_BIT)
    glColor3f(0, 255, 0)
    glBegin(GL_POLYGON)
        glVertex3f(0.0, 0.0, 0.0)
```

```
        glVertex3f(0.5, 0.0, 0.0)
        glVertex3f(0.5, 0.5, 0.0)
        glVertex3f(0.0, 0.5, 0.0)
    glEnd()
    glColor3f(255,0,0)
    glBegin(GL_POLYGON)
        glVertex3f(0.0, 0.0, 0.0)
        glVertex3f(0.5, 0.0, 0.0)
        glVertex3f(-0.5,- 1, 0.0)
        glVertex3f(0.0, -1, 0.0)
    glEnd()
    glColor3f(0,0,255)
    glBegin(GL_POLYGON)
        glVertex3f(0.0, 0.0, 0.0)
        glVertex3f(-0.5, 0.0, 0.0)
        glVertex3f(-0.5,- 0.5, 0.0)
        glVertex3f(0.0, -0.5, 0.0)
    glEnd()

    glFlush()
```

Screen Shot



55.6 The First Triangle

Example:

```
load "freeglut.ring"
load "opengl21lib.ring"

func main
    glutInit()
    glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA)
    glutInitWindowSize(320,320)
    glutInitWindowPosition(100, 10)
    glutCreateWindow("RingFreeGLUT - Test 3")
    glutDisplayFunc(:renderScene)
    glutMainLoop()

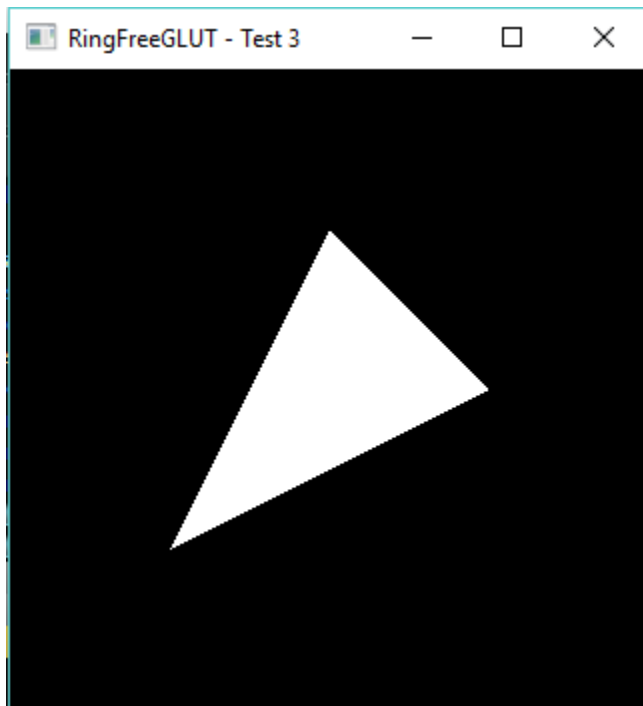
func renderScene

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

    glBegin(GL_TRIANGLES)
        glVertex3f(-0.5,-0.5,0.0)
        glVertex3f(0.5,0.0,0.0)
        glVertex3f(0.0,0.5,0.0)
    glEnd()

    glutSwapBuffers()
```

Screen Shot



55.7 Window Resize Event

Example:

```
load "freeglut.ring"
load "opengl21lib.ring"

func main

    // init GLUT and create window
    glutInit()
    glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA)
    glutInitWindowPosition(100,100)
    glutInitWindowSize(320,320)
    glutCreateWindow("RingFreeGLUT - Test 4")

    glutDisplayFunc(:renderScene)
    glutReshapeFunc(:changeSize)

    glutMainLoop()

func renderScene

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

    glBegin(GL_TRIANGLES)
        glVertex3f(-2,-2,-5.0)
        glVertex3f(2,0.0,-5.0)
        glVertex3f(0.0,2,-5.0)
    glEnd()

    glutSwapBuffers()

func changesize

    h = glutEventHeight()
    w = glutEventWidth()

    // Prevent a divide by zero, when window is too short
    // (you cant make a window of zero width).
    if (h = 0)
        h = 1
    ok

    ratio = w * 1.0 / h

    // Use the Projection Matrix
    glMatrixMode(GL_PROJECTION)

    // Reset Matrix
    glLoadIdentity()

    // Set the viewport to be the entire window
    glViewport(0, 0, w, h)

    // Set the correct perspective.
    gluPerspective(45,ratio,1,100)
```

```
// Get Back to the Modelview
glMatrixMode(GL_MODELVIEW)
```

55.8 Triangle Rotation

Example:

```
load "freeglut.ring"
load "opengl2lib.ring"

angle = 0

func main

    // init GLUT and create window
    glutInit()
    glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA)
    glutInitWindowPosition(100,100)
    glutInitWindowSize(320,320)
    glutCreateWindow("RingFreeGLUT - Test 5")

    glutDisplayFunc(:renderScene)
    glutReshapeFunc(:changeSize)
    glutIdleFunc(:renderScene)

    glutMainLoop()

func renderScene

    // Clear Color and Depth Buffers
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

    // Reset transformations
    glLoadIdentity()
    // Set the camera
    gluLookAt(      0.0, 0.0, 10.0,
                  0.0, 0.0,  0.0,
                  0.0, 1.0,  0.0)

    glRotatef(angle, 0.0, 1.0, 0.0)

    glBegin(GL_TRIANGLES)
        glVertex3f(-2.0,-2.0, 0.0)
        glVertex3f( 2.0, 0.0, 0.0)
        glVertex3f( 0.0, 2.0, 0.0)
    glEnd()

    angle+=0.1

    glutSwapBuffers();

func changeSize

    h = glutEventHeight()
    w = glutEventWidth()
```



```
// Prevent a divide by zero, when window is too short
// (you cant make a window of zero width).
if (h = 0)
    h = 1
ok

ratio = w * 1.0 / h

// Use the Projection Matrix
glMatrixMode(GL_PROJECTION)

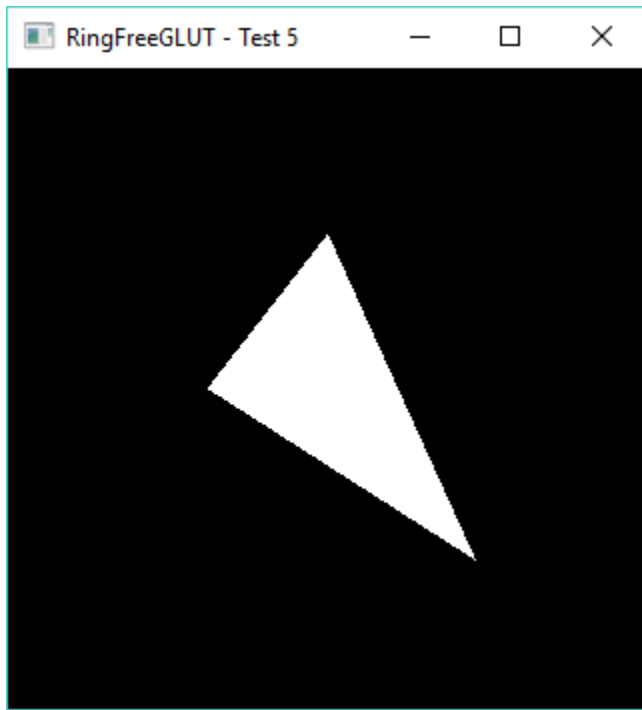
// Reset Matrix
glLoadIdentity()

// Set the viewport to be the entire window
glViewport(0, 0, w, h)

// Set the correct perspective.
gluPerspective(45, ratio, 1, 100)

// Get Back to the Modelview
glMatrixMode(GL_MODELVIEW)
```

Screen Shot



55.9 Keyboard Events and Colors

Example:

```
load "freeglut.ring"
load "opengl2llib.ring"
```

```

angle = 0

red=1.0
blue=1.0
green=1.0

func main

    // init GLUT and create window
    glutInit()
    glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA)
    glutInitWindowPosition(100,100)
    glutInitWindowSize(320,320)
    glutCreateWindow("RingFreeGLUT - Test 6")

    glutDisplayFunc(:renderScene)
    glutReshapeFunc(:changeSize)
    glutIdleFunc(:renderScene)

    // here are the new entries
    glutKeyboardFunc(:processNormalKeys)
    glutSpecialFunc(:processSpecialKeys)

    glutMainLoop()

func renderScene

    // Clear Color and Depth Buffers
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

    // Reset transformations
    glLoadIdentity()
    // Set the camera
    gluLookAt(      0.0, 0.0, 10.0,
                  0.0, 0.0,  0.0,
                  0.0, 1.0,  0.0)

    glRotatef(angle, 0.0, 1.0, 0.0)

    glColor3f(red,green,blue);

    glBegin(GL_TRIANGLES)
        glVertex3f(-2.0,-2.0, 0.0)
        glVertex3f( 2.0, 0.0, 0.0)
        glVertex3f( 0.0, 2.0, 0.0)
    glEnd()

    angle+=0.1

    glutSwapBuffers();

func changesize

    h = glutEventHeight()
    w = glutEventWidth()

    // Prevent a divide by zero, when window is too short

```

```

// (you cant make a window of zero width).
if (h = 0)
    h = 1
ok

ratio = w * 1.0 / h

// Use the Projection Matrix
glMatrixMode(GL_PROJECTION)

// Reset Matrix
glLoadIdentity()

// Set the viewport to be the entire window
glViewport(0, 0, w, h)

// Set the correct perspective.
gluPerspective(45,ratio,1,100)

// Get Back to the Modelview
glMatrixMode(GL_MODELVIEW)

func processNormalKeys
    key = GLUTEventKey()
    if key = 27
        shutdown()
    ok

func processSpecialKeys

    key = GLUTEventKey()

    switch key
        on GLUT_KEY_F1
            red = 1.0
            green = 0.0
            blue = 0.0

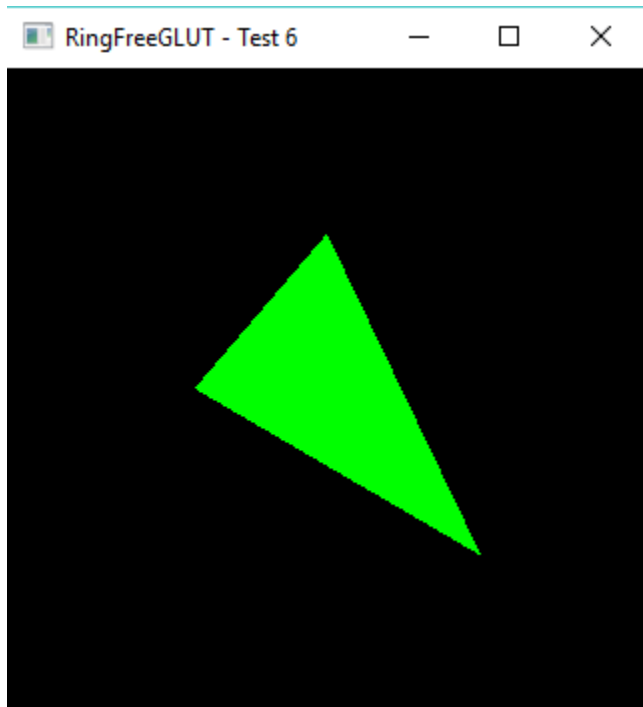
        on GLUT_KEY_F2
            red = 0.0
            green = 1.0
            blue = 0.0

        on GLUT_KEY_F3
            red = 0.0
            green = 0.0
            blue = 1.0

    off

```

Screen Shot



55.10 The Camera

Example:

```
load "freeglut.ring"
load "opengl2llib.ring"

// angle of rotation for the camera direction
angle=0.0
// actual vector representing the camera's direction
lx=0.0
lz=-1.0
// XZ position of the camera
x=0.0
z=5.0

func drawSnowMan

    glColor3f(1.0, 1.0, 1.0)

// Draw Body
    glTranslatef(0.0, 0.75, 0.0)
    glutSolidSphere(0.75, 20, 20)

// Draw Head
    glTranslatef(0.0, 1.0, 0.0)
    glutSolidSphere(0.25, 20, 20)

// Draw Eyes
    glPushMatrix()
    glColor3f(0.0, 0.0, 0.0)
    glTranslatef(0.05, 0.10, 0.18)
```

```

    glutSolidSphere(0.05,10,10)
    glTranslatef(-0.1, 0.0, 0.0)
    glutSolidSphere(0.05,10,10)

    glPopMatrix()

// Draw Nose
    glColor3f(1.0, 0.5 , 0.5)
    glutSolidCone(0.08,0.5,10,2)

func changeSize
    w = glutEventWidth()
    h = glutEventHeight()

    // Prevent a divide by zero, when window is too short
    // (you cant make a window of zero width).
    if h = 0
        h = 1

    ok

    ratio = w * 1.0 / h

    // Use the Projection Matrix
    glMatrixMode(GL_PROJECTION)

    // Reset Matrix
    glLoadIdentity()

    // Set the viewport to be the entire window
    glViewport(0, 0, w, h)

    // Set the correct perspective.
    gluPerspective(45.0, ratio, 0.1, 100.0);

    // Get Back to the Modelview
    glMatrixMode(GL_MODELVIEW)

func processNormalKeys
    key = glutEventKey()

    if key = 27
        shutdown()

    ok

func renderScene

    // Clear Color and Depth Buffers

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

    // Reset transformations
    glLoadIdentity()
    // Set the camera

```

```

gluLookAt(      x, 1.0, z,
               x+lx, 1.0, z+lz,
               0.0, 1.0, 0.0)

    // Draw ground

glColor3f(0.9, 0.9, 0.9)
glBegin(GL_QUADS)
    glVertex3f(-100.0, 0.0, -100.0)
    glVertex3f(-100.0, 0.0, 100.0)
    glVertex3f( 100.0, 0.0, 100.0)
    glVertex3f( 100.0, 0.0, -100.0)
glEnd()

    // Draw 36 SnowMen
for i = -3 to 2
    for j=-3 to 2
        glPushMatrix()
        glTranslatef(i*10.0,0,j * 10.0)
        drawSnowMan()
        glPopMatrix()
    next
next
glutSwapBuffers()

func processSpecialKeys

    key = glutEventKey()

    fraction = 0.1

    switch key
        on GLUT_KEY_LEFT
            angle -= 0.01
            lx = sin(angle)
            lz = -cos(angle)
        on GLUT_KEY_RIGHT
            angle += 0.01
            lx = sin(angle)
            lz = -cos(angle)
        on GLUT_KEY_UP
            x += lx * fraction
            z += lz * fraction
        on GLUT_KEY_DOWN
            x -= lx * fraction
            z -= lz * fraction
    off

func main

    // init GLUT and create window

    glutInit()
    glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA)

```

```

glutInitWindowPosition(100,100)
glutInitWindowSize(320,320)
glutCreateWindow("RingFreeGLUT - Test 7")

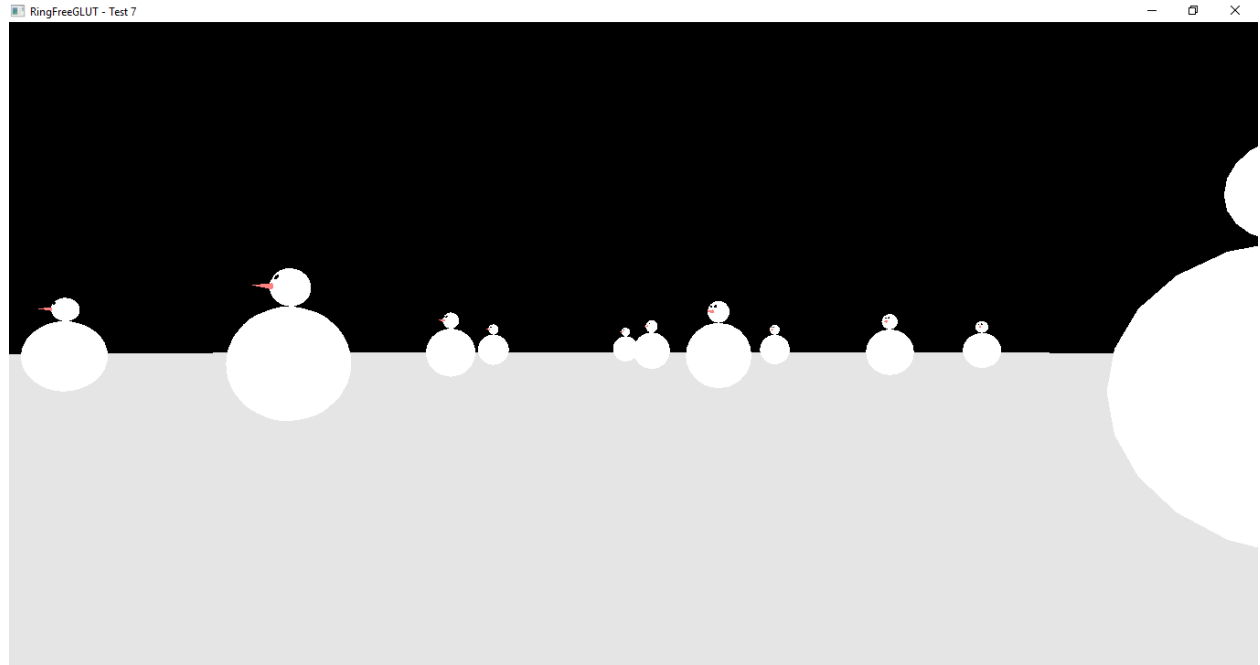
// register callbacks
glutDisplayFunc(:renderScene)
glutReshapeFunc(:changeSize)
glutIdleFunc(:renderScene)
glutKeyboardFunc(:processNormalKeys)
glutSpecialFunc(:processSpecialKeys)

// OpenGL init
glEnable(GL_DEPTH_TEST)

// enter GLUT event processing cycle
glutMainLoop()

```

Screen Shot



Another Example:

```

load "freeglut.ring"
load "opengl2llib.ring"

// angle of rotation for the camera direction
angle = 0.0

// actual vector representing the camera's direction
lx=0.0 lz=-1.0

// XZ position of the camera
x=0.0 z=5.0
// the key states. These variables will be zero
//when no key is being presses
deltaAngle = 0.0

```

```

deltaMove = 0

func changeSize
    w = glutEventWidth()
    h = glutEventHeight()

    // Prevent a divide by zero, when window is too short
    // (you cant make a window of zero width).
    if h = 0
        h = 1

    ok

    ratio = w * 1.0 / h

    // Use the Projection Matrix
    glMatrixMode(GL_PROJECTION)

    // Reset Matrix
    glLoadIdentity()

    // Set the viewport to be the entire window
    glViewport(0, 0, w, h)

    // Set the correct perspective.
    gluPerspective(45.0, ratio, 0.1, 100.0)

    // Get Back to the Modelview
    glMatrixMode(GL_MODELVIEW)

func drawSnowMan

    glColor3f(1.0, 1.0, 1.0)

    // Draw Body

    glTranslatef(0.0, 0.75, 0.0)
    glutSolidSphere(0.75, 20, 20)

    // Draw Head
    glTranslatef(0.0, 1.0, 0.0)
    glutSolidSphere(0.25, 20, 20)

    // Draw Eyes
    glPushMatrix()
    glColor3f(0.0, 0.0, 0.0)
    glTranslatef(0.05, 0.10, 0.18)
    glutSolidSphere(0.05, 10, 10)
    glTranslatef(-0.1, 0.0, 0.0)
    glutSolidSphere(0.05, 10, 10)
    glPopMatrix()

    // Draw Nose
    glColor3f(1.0, 0.5, 0.5)
    glRotatef(0.0, 1.0, 0.0, 0.0)
    glutSolidCone(0.08, 0.5, 10, 2)

```



```

func computePos deltaMove

    x += deltaMove * lx * 0.1
    z += deltaMove * lz * 0.1

func computeDir deltaAngle

    angle += deltaAngle
    lx = sin(angle)
    lz = -cos(angle)

func renderScene

    if deltaMove
        computePos(deltaMove)
    ok

    if deltaAngle
        computeDir(deltaAngle)
    ok

    // Clear Color and Depth Buffers
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

    // Reset transformations
    glLoadIdentity()
    // Set the camera
    gluLookAt(
        x, 1.0, z,
        x+lx, 1.0, z+lz,
        0.0, 1.0, 0.0)

// Draw ground

    glColor3f(0.9, 0.9, 0.9)
    glBegin(GL_QUADS)
        glVertex3f(-100.0, 0.0, -100.0)
        glVertex3f(-100.0, 0.0, 100.0)
        glVertex3f( 100.0, 0.0, 100.0)
        glVertex3f( 100.0, 0.0, -100.0)
    glEnd()

// Draw 36 SnowMen

    for i = -3 to 2
        for j=-3 to 2
            glPushMatrix()
            glTranslatef(i*10.0,0,j * 10.0)
            drawSnowMan()
            glPopMatrix()
        next
    next
    glutSwapBuffers()

func pressKey
    key = glutEventKey()
    xx = glutEventX()

```

```

yy = glutEventY()

switch key
    on GLUT_KEY_LEFT
        deltaAngle = -0.01
    on GLUT_KEY_RIGHT
        deltaAngle = 0.01
    on GLUT_KEY_UP
        deltaMove = 0.5
    on GLUT_KEY_DOWN
        deltaMove = -0.5
off

func releaseKey

    key = glutEventKey()

    switch key
        on GLUT_KEY_LEFT
            deltaAngle = 0.0
        on GLUT_KEY_RIGHT
            deltaAngle = 0.0
        on GLUT_KEY_UP
            deltaMove = 0
        on GLUT_KEY_DOWN
            deltaMove = 0
    off

func main

    // init GLUT and create window
    glutInit()
    glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA)
    glutInitWindowPosition(100,100)
    glutInitWindowSize(320,320)
    glutCreateWindow("RingFreeGLUT - Test 8")

    // register callbacks
    glutDisplayFunc(:renderScene)
    glutReshapeFunc(:changeSize)
    glutIdleFunc(:renderScene)

    glutSpecialFunc(:pressKey)

    // here are the new entries
    glutIgnoreKeyRepeat(1)
    glutSpecialUpFunc(:releaseKey)

    // OpenGL init
    glEnable(GL_DEPTH_TEST)

    // enter GLUT event processing cycle
    glutMainLoop()

```

55.11 Mouse Events

Example:

```
load "freeglut.ring"
load "opengl21lib.ring"

// angle of rotation for the camera direction
angle = 0.0

// actual vector representing the camera's direction
lx=0.0 lz=-1.0

// XZ position of the camera
x=0.0 z=5.0

// the key states. These variables will be zero
//when no key is being presses
deltaAngle = 0.0
deltaMove = 0.0
xOrigin = -1

func changeSize
    w = glutEventWidth()
    h = glutEventHeight()

    // Prevent a divide by zero, when window is too short
    // (you cant make a window of zero width).
    if h = 0
        h = 1

    ok

    ratio = w * 1.0 / h

    // Use the Projection Matrix
    glMatrixMode(GL_PROJECTION)

    // Reset Matrix
    glLoadIdentity()

    // Set the viewport to be the entire window
    glViewport(0, 0, w, h)

    // Set the correct perspective.
    gluPerspective(45.0, ratio, 0.1, 100.0)

    // Get Back to the Modelview
    glMatrixMode(GL_MODELVIEW)

func drawSnowMan

    glColor3f(1.0, 1.0, 1.0)

    // Draw Body
    glTranslatef(0.0 ,0.75, 0.0)
    glutSolidSphere(0.75,20,20)
```

```

// Draw Head
glTranslatef(0.0, 1.0, 0.0)
glutSolidSphere(0.25,20,20)

// Draw Eyes
glPushMatrix()
glColor3f(0.0,0.0,0.0)
glTranslatef(0.05, 0.10, 0.18)
glutSolidSphere(0.05,10,10)
glTranslatef(-0.1, 0.0, 0.0)
glutSolidSphere(0.05,10,10)
glPopMatrix()

// Draw Nose
glColor3f(1.0, 0.5 , 0.5)
glRotatef(0.0,1.0, 0.0, 0.0)
glutSolidCone(0.08,0.5,10,2)

func computePos deltaMove

    x += deltaMove * lx * 0.1
    z += deltaMove * lz * 0.1

func renderScene

    if deltaMove
        computePos(deltaMove)
    ok

    // Clear Color and Depth Buffers
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

    // Reset transformations
    glLoadIdentity()
    // Set the camera
    gluLookAt(
        x, 1.0, z,
        x+lx, 1.0, z+lz,
        0.0, 1.0, 0.0)

    // Draw ground

    glColor3f(0.9, 0.9, 0.9)
    glBegin(GL_QUADS)
        glVertex3f(-100.0, 0.0, -100.0)
        glVertex3f(-100.0, 0.0, 100.0)
        glVertex3f( 100.0, 0.0, 100.0)
        glVertex3f( 100.0, 0.0, -100.0)
    glEnd()

    // Draw 36 SnowMen

    for i = -3 to 2
        for j=-3 to 2

            glPushMatrix()
            glTranslatef(i*10.0,0,j * 10.0)
            drawSnowMan()
            glPopMatrix()

```

```

        next
    next
    glutSwapBuffers()

func processNormalKeys

    key = glutEventKey()

    if key = 27
        shutdown()
    ok

func pressKey
    key = glutEventKey()

    switch key
        on GLUT_KEY_UP
            deltaMove = 0.5
        on GLUT_KEY_DOWN
            deltaMove = -0.5
    off

func releaseKey
    key = glutEventKey()
    switch key
        on GLUT_KEY_UP
            deltaMove = 0
        on GLUT_KEY_DOWN
            deltaMove = 0
    off

func mouseMove
    xx = glutEventX()
    yy = glutEventY()
    // this will only be true when the left button is down
    if xOrigin >= 0

        // update deltaAngle
        deltaAngle = (xx - xOrigin) * 0.001

        // update camera's direction
        lx = sin(angle + deltaAngle)
        lz = -cos(angle + deltaAngle)
    ok

func mouseButton

    button = glutEventButton()
    state = glutEventState()
    xx = glutEventX()
    yy = glutEventY()

    // only start motion if the left button is pressed

```

```

    if button = GLUT_LEFT_BUTTON
        // when the button is released
        if state = GLUT_UP
            angle += deltaAngle
            xOrigin = -1
        else
            // state = GLUT_DOWN
            xOrigin = xx
        ok
        fflush(stdout)
    ok

func main

    // init GLUT and create window
    glutInit()
    glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA)
    glutInitWindowPosition(100,100)
    glutInitWindowSize(320,320)
    glutCreateWindow("RingFreeGLUT - Test 9")

    // register callbacks
    glutDisplayFunc(:renderScene)
    glutReshapeFunc(:changeSize)
    glutIdleFunc(:renderScene)

    glutIgnoreKeyRepeat(1)
    glutKeyboardFunc(:processNormalKeys)
    glutSpecialFunc(:pressKey)
    glutSpecialUpFunc(:releaseKey)

    // here are the two new functions
    glutMouseFunc(:mouseButton)
    glutMotionFunc(:mouseMove)

    // OpenGL init
    glEnable(GL_DEPTH_TEST)

    // enter GLUT event processing cycle
    glutMainLoop()

```

55.12 Menu Events

Example:

```

load "freeglut.ring"
load "opengl2llib.ring"

// angle of rotation for the camera direction
angle = 0.0

// actual vector representing the camera's direction
lx=0.0 lz=-1.0

// XZ position of the camera

```

```

x=0.0  z=5.0

// the key states. These variables will be zero
//when no key is being presses
deltaAngle = 0.0
deltaMove = 0
xOrigin = -1

// Constant definitions for Menus

// for RingFreeGLUT - We must have different ID for each menu item
C_RED = 1
C_GREEN = 2
C_BLUE = 3
C_ORANGE = 4

C_FILL = 5
C_LINE = 6

C_SHRINK = 7
C_NORMAL = 8

// Pop up menu identifiers
fillMenu= 0
shrinkMenu= 0
mainMenu=0
colorMenu=0

// color for the nose
red = 1.0  blue=0.5  green=0.5

// scale of snowman
scale = 1.0

// menu status
menuFlag = 0

func changeSize
    w = glutEventWidth()
    h = glutEventHeight()

    // Prevent a divide by zero, when window is too short
    // (you cant make a window of zero width).
    if h = 0
        h = 1

    ok

    ratio = w * 1.0 / h

    // Use the Projection Matrix
    glMatrixMode(GL_PROJECTION)

    // Reset Matrix
    glLoadIdentity()

    // Set the viewport to be the entire window
    glViewport(0, 0, w, h)

```

```

    // Set the correct perspective.
    gluPerspective(45.0, ratio, 0.1, 100.0)

    // Get Back to the Modelview
    glMatrixMode(GL_MODELVIEW)

func drawSnowMan

    glScalef(scale, scale, scale)
    glColor3f(1.0, 1.0, 1.0)

// Draw Body
    glTranslatef(0.0 ,0.75, 0.0)
    glutSolidSphere(0.75,20,20)

// Draw Head
    glTranslatef(0.0, 1.0, 0.0)
    glutSolidSphere(0.25,20,20)

// Draw Eyes
    glPushMatrix()
    glColor3f(0.0,0.0,0.0)
    glTranslatef(0.05, 0.10, 0.18)
    glutSolidSphere(0.05,10,10)
    glTranslatef(-0.1, 0.0, 0.0)
    glutSolidSphere(0.05,10,10)
    glPopMatrix()

// Draw Nose
    glColor3f(red, green, blue)
    glRotatef(0.0,1.0, 0.0, 0.0)
    glutSolidCone(0.08,0.5,10,2)

    glColor3f(1.0, 1.0, 1.0)

func computePos deltaMove

    x += deltaMove * lx * 0.1
    z += deltaMove * lz * 0.1

func renderScene

    if deltaMove
        computePos(deltaMove)
    ok

    // Clear Color and Depth Buffers
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

    // Reset transformations
    glLoadIdentity()
    // Set the camera
    gluLookAt(
        x, 1.0, z,
        x+lx, 1.0, z+lz,
        0.0, 1.0, 0.0)

```



```

// Draw ground

glColor3f(0.9, 0.9, 0.9)
glBegin(GL_QUADS)
    glVertex3f(-100.0, 0.0, -100.0)
    glVertex3f(-100.0, 0.0, 100.0)
    glVertex3f( 100.0, 0.0, 100.0)
    glVertex3f( 100.0, 0.0, -100.0)
glEnd()

// Draw 36 SnowMen

    for i = -3 to 2
        for j = -3 to 2
            glPushMatrix()
            glTranslatef(i*10.0, 0.0, j * 10.0)
            drawSnowMan()
            glPopMatrix()

        next
    next
glutSwapBuffers()

// -----
//          KEYBOARD
// -----

func processNormalKeys
    key = glutEventKey()
    xx = glutEventX()
    yy = glutEventY()

    glutSetMenu(mainMenu)
    switch key
        on 27
            glutDestroyMenu(mainMenu)
            glutDestroyMenu(fillMenu)
            glutDestroyMenu(colorMenu)
            glutDestroyMenu(shrinkMenu)
            shutdown()

        on 's'
            if not menuFlag
                glutChangeToSubMenu(2, "Shrink", shrinkMenu)
            ok

        on 'c'
            if not menuFlag
                glutChangeToSubMenu(2, "Color", colorMenu)
            ok

    off
    if key = 27
        shutdown()
    ok

func pressKey

    key = glutEventKey()

```

```

xx = glutEventX()
yy = glutEventY()

switch key
  on GLUT_KEY_UP
    deltaMove = 0.5
  on GLUT_KEY_DOWN
    deltaMove = -0.5
off

func releaseKey

  key = glutEventKey()

  switch key
    on GLUT_KEY_UP
      deltaMove = 0
    on GLUT_KEY_DOWN
      deltaMove = 0
  off

// -----
//          MOUSE
// -----

func mouseMove
  xx = glutEventX()
  yy = glutEventY()

  // this will only be true when the left button is down
  if xOrigin >= 0

    // update deltaAngle
    deltaAngle = (xx - xOrigin) * 0.001

    // update camera's direction
    lx = sin(angle + deltaAngle)
    lz = -cos(angle + deltaAngle)

  ok

func mouseButton

  button = glutEventButton()
  state = glutEventState()
  xx = glutEventX()
  yy = glutEventY()

  // only start motion if the left button is pressed
  if button = GLUT_LEFT_BUTTON
    // when the button is released
    if state = GLUT_UP
      angle += deltaAngle
      xOrigin = -1
    else
      // state = GLUT_DOWN

```

```

                                xOrigin = xx
                                ok
                                ok
                                ok

// -----
//             MENUS
// -----

func processMenuStatus

    status = glutEventStatus()
    xx = glutEventX()
    yy = glutEventY()

    if status = GLUT_MENU_IN_USE
        menuFlag = 1
    else
        menuFlag = 0
    ok

func processMainMenu

    // nothing to do in here
    // all actions are for submenus

func processFillMenu

    option = glutEventValue()

    switch option

        on C_FILL
            glPolygonMode(GL_FRONT, GL_FILL)
        on C_LINE
            glPolygonMode(GL_FRONT, GL_LINE)
    off

func processShrinkMenu

    option = glutEventValue()

    switch option

        on C_SHRINK
            scale = 0.5
        on C_NORMAL
            scale = 1.0
    off

func processColorMenu

    option = glutEventValue()

```

```

switch option
  on C_RED
    red = 1.0
    green = 0.0
    blue = 0.0
  on C_GREEN
    red = 0.0
    green = 1.0
    blue = 0.0
  on C_BLUE
    red = 0.0
    green = 0.0
    blue = 1.0
  on C_ORANGE
    red = 1.0
    green = 0.5
    blue = 0.5
off

func createPopupMenu

  shrinkMenu = glutCreateMenu(:processShrinkMenu)

  glutAddMenuEntry("Shrink",C_SHRINK)
  glutAddMenuEntry("NORMAL",C_NORMAL)

  fillMenu = glutCreateMenu(:processFillMenu)

  glutAddMenuEntry("Fill",C_FILL)
  glutAddMenuEntry("Line",C_LINE)

  colorMenu = glutCreateMenu(:processColorMenu)
  glutAddMenuEntry("Red",C_RED)
  glutAddMenuEntry("Blue",C_BLUE)
  glutAddMenuEntry("Green",C_GREEN)
  glutAddMenuEntry("Orange",C_ORANGE)

  mainMenu = glutCreateMenu(:processMainMenu)

  glutAddSubMenu("Polygon Mode", fillMenu)
  glutAddSubMenu("Color", colorMenu)
  // attach the menu to the right button
  glutAttachMenu(GLUT_RIGHT_BUTTON)

  // this will allow us to know if the menu is active
  glutMenuStatusFunc(:processMenuStatus)

// -----
//             MAIN
// -----

func main

  // init GLUT and create window
  glutInit()
  glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA)

```

```

glutInitWindowPosition(100,100)
glutInitWindowSize(320,320)
glutCreateWindow("RingFreeGLUT - Test 10")

// register callbacks
glutDisplayFunc(:renderScene)
glutReshapeFunc(:changeSize)
glutIdleFunc(:renderScene)

glutIgnoreKeyRepeat(1)
glutKeyboardFunc(:processNormalKeys)
glutSpecialFunc(:pressKey)
glutSpecialUpFunc(:releaseKey)

// here are the two new functions
glutMouseFunc(:mouseButton)
glutMotionFunc(:mouseMove)

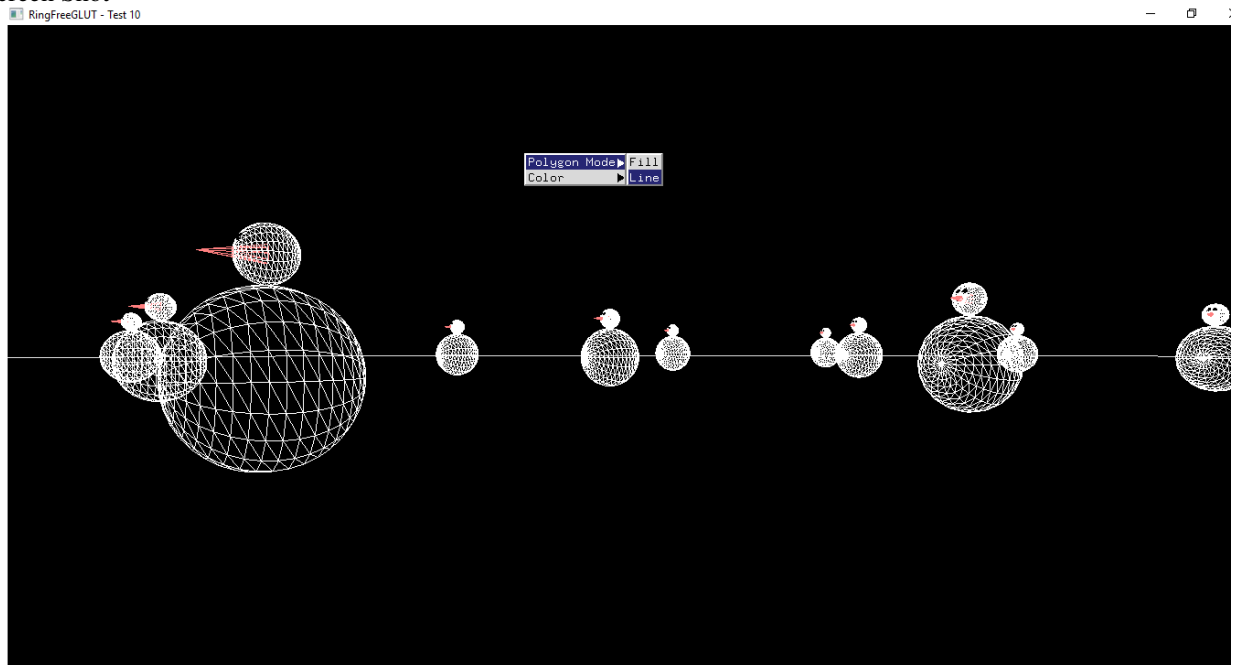
// OpenGL init
glEnable(GL_DEPTH_TEST)
glEnable(GL_CULL_FACE)

// init Menus
createPopupMenu()

// enter GLUT event processing cycle
glutMainLoop()

```

Screen Shot



55.13 Using Fonts

Example:

```

load "freeglut.ring"
load "opengl2llib.ring"

// angle of rotation for the camera direction
angle = 0.0

// actual vector representing the camera's direction
lx=0.0 lz=-1.0

// XZ position of the camera
x=0.0 z=5.0

// the key states. These variables will be zero
//when no key is being presses
deltaAngle = 0.0
deltaMove = 0
xOrigin = -1

// Constant definitions for Menus
C_RED = 1
C_GREEN = 2
C_BLUE = 3
C_ORANGE = 4

C_FILL = 5
C_LINE = 6

// Pop up menu identifiers
fillMenu=NULL
fontMenu=NULL
mainMenu=NULL
colorMenu=NULL

// color for the nose
red = 1.0
blue=0.5
green=0.5

// scale of snowman
scale = 1.0

// menu status
menuFlag = 0

// default font
font = GLUT_BITMAP_TIMES_ROMAN_24

C_INT_GLUT_BITMAP_8_BY_13 = 7
C_INT_GLUT_BITMAP_9_BY_15 = 8
C_INT_GLUT_BITMAP_TIMES_ROMAN_10 = 9
C_INT_GLUT_BITMAP_TIMES_ROMAN_24 = 10
C_INT_GLUT_BITMAP_HELVETICA_10 = 11
C_INT_GLUT_BITMAP_HELVETICA_12 = 12
C_INT_GLUT_BITMAP_HELVETICA_18 = 13

func changeSize
    w = glutEventWidth()

```

```

    h = glutEventHeight()

    // Prevent a divide by zero, when window is too short
    // (you cant make a window of zero width).
    if h = 0
        h = 1

    ok

    ratio = w * 1.0 / h

    // Use the Projection Matrix
    glMatrixMode(GL_PROJECTION)

    // Reset Matrix
    glLoadIdentity()

    // Set the viewport to be the entire window
    glViewport(0, 0, w, h)

    // Set the correct perspective.
    gluPerspective(45.0, ratio, 0.1, 100.0)

    // Get Back to the Modelview
    glMatrixMode(GL_MODELVIEW)

func drawSnowMan

    glScalef(scale, scale, scale)
    glColor3f(1.0, 1.0, 1.0)

// Draw Body
    glTranslatef(0.0, 0.75, 0.0)
    glutSolidSphere(0.75, 20, 20)

// Draw Head
    glTranslatef(0.0, 1.0, 0.0)
    glutSolidSphere(0.25, 20, 20)

// Draw Eyes
    glPushMatrix()
    glColor3f(0.0, 0.0, 0.0)
    glTranslatef(0.05, 0.10, 0.18)
    glutSolidSphere(0.05, 10, 10)
    glTranslatef(-0.1, 0.0, 0.0)
    glutSolidSphere(0.05, 10, 10)
    glPopMatrix()

// Draw Nose
    glColor3f(red, green, blue)
    glRotatef(0.0, 1.0, 0.0, 0.0)
    glutSolidCone(0.08, 0.5, 10, 2)

    glColor3f(1.0, 1.0, 1.0)

func renderBitmapString x, y, z, font, string
    glRasterPos3f(x, y, z)
    for c in string
        glutBitmapCharacter(font, ascii(c))

```

```

    next

func computePos deltaMove

    x += deltaMove * lx * 0.1
    z += deltaMove * lz * 0.1

func renderScene

    if deltaMove
        computePos(deltaMove)
    ok

    // Clear Color and Depth Buffers
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

    // Reset transformations
    glLoadIdentity()

    // Set the camera
    gluLookAt(
        x, 1.0, z,
        x+lx, 1.0, z+lz,
        0.0, 1.0, 0.0)

    // Draw ground

    glColor3f(0.9, 0.9, 0.9)
    glBegin(GL_QUADS)
        glVertex3f(-100.0, 0.0, -100.0)
        glVertex3f(-100.0, 0.0, 100.0)
        glVertex3f( 100.0, 0.0, 100.0)
        glVertex3f( 100.0, 0.0, -100.0)
    glEnd()

// Draw 36 SnowMen
for i = -3 to 2
    for j = -3 to 2
        glPushMatrix()
        glTranslatef(i*10.0, 0.0, j * 10.0)
        drawSnowMan()
        number = (i+3)*6+(j+3)
        renderBitmapString(0.0, 0.5, 0.0,font ,""+number)
        glPopMatrix()
    next
next

glutSwapBuffers()

// -----
//          KEYBOARD
// -----

func processNormalKeys
    key = glutEventKey()
    xx = glutEventX()

```



```

yy = glutEventY()

switch key
  on 27
    glutDestroyMenu(mainMenu)
    glutDestroyMenu(fillMenu)
    glutDestroyMenu(colorMenu)
    glutDestroyMenu(fontMenu)
    Shutdown()
  off

func pressKey

  key = glutEventKey()
  xx = glutEventX()
  yy = glutEventY()

  switch key
    on GLUT_KEY_UP
      deltaMove = 0.5
    on GLUT_KEY_DOWN
      deltaMove = -0.5
  off

func releaseKey

  key = glutEventKey()

  switch key
    on GLUT_KEY_UP
      deltaMove = 0
    on GLUT_KEY_DOWN
      deltaMove = 0
  off

// -----
//           MOUSE
// -----

func mouseMove
  xx = glutEventX()
  yy = glutEventY()

  // this will only be true when the left button is down
  if xOrigin >= 0

    // update deltaAngle
    deltaAngle = (xx - xOrigin) * 0.001

    // update camera's direction
    lx = sin(angle + deltaAngle)
    lz = -cos(angle + deltaAngle)
  ok

```

```

func mouseButton

    button = glutEventButton()
    state = glutEventState()
    xx = glutEventX()
    yy = glutEventY()

    // only start motion if the left button is pressed
    if button = GLUT_LEFT_BUTTON
        // when the button is released
        if state = GLUT_UP
            angle += deltaAngle
            xOrigin = -1

        else
            // state = GLUT_DOWN
            xOrigin = xx

        ok

    ok

// -----
//             MENUS
// -----

func processMenuStatus

    status = glutEventStatus()

    if status = GLUT_MENU_IN_USE
        menuFlag = 1
    else
        menuFlag = 0
    ok

func processMainMenu

    // nothing to do in here
    // all actions are for submenus

func processFillMenu

    option = glutEventValue()

    switch option

        on C_FILL
            glPolygonMode(GL_FRONT, GL_FILL)
        on C_LINE
            glPolygonMode(GL_FRONT, GL_LINE)

    off

func processFontMenu

    option = glutEventValue()

```

```

switch (option) {
    on C_INT_GLUT_BITMAP_8_BY_13
        font = GLUT_BITMAP_8_BY_13
    on C_INT_GLUT_BITMAP_9_BY_15
        font = GLUT_BITMAP_9_BY_15
    on C_INT_GLUT_BITMAP_TIMES_ROMAN_10
        font = GLUT_BITMAP_TIMES_ROMAN_10
    on C_INT_GLUT_BITMAP_TIMES_ROMAN_24
        font = GLUT_BITMAP_TIMES_ROMAN_24
    on C_INT_GLUT_BITMAP_HELVETICA_10
        font = GLUT_BITMAP_HELVETICA_10
    on C_INT_GLUT_BITMAP_HELVETICA_12
        font = GLUT_BITMAP_HELVETICA_12
    on C_INT_GLUT_BITMAP_HELVETICA_18
        font = GLUT_BITMAP_HELVETICA_18
    off

func processColorMenu

    option = glutEventValue()

    switch option
        on C_RED
            red = 1.0
            green = 0.0
            blue = 0.0
        on C_GREEN
            red = 0.0
            green = 1.0
            blue = 0.0
        on C_BLUE
            red = 0.0
            green = 0.0
            blue = 1.0
        on C_ORANGE
            red = 1.0
            green = 0.5
            blue = 0.5
        off

func createPopUpMenus

    fontMenu = glutCreateMenu(:processFontMenu)

    glutAddMenuEntry("BITMAP_8_BY_13 ",C_INT_GLUT_BITMAP_8_BY_13 )
    glutAddMenuEntry("BITMAP_9_BY_15",C_INT_GLUT_BITMAP_9_BY_15 )
    glutAddMenuEntry("BITMAP_TIMES_ROMAN_10 ",C_INT_GLUT_BITMAP_TIMES_ROMAN_10 )
    glutAddMenuEntry("BITMAP_TIMES_ROMAN_24",C_INT_GLUT_BITMAP_TIMES_ROMAN_24 )
    glutAddMenuEntry("BITMAP_HELVETICA_10 ",C_INT_GLUT_BITMAP_HELVETICA_10 )
    glutAddMenuEntry("BITMAP_HELVETICA_12",C_INT_GLUT_BITMAP_HELVETICA_12 )
    glutAddMenuEntry("BITMAP_HELVETICA_18",C_INT_GLUT_BITMAP_HELVETICA_18 )

    fillMenu = glutCreateMenu(:processFillMenu)

    glutAddMenuEntry("Fill",C_FILL)
    glutAddMenuEntry("Line",C_LINE)

```

```

colorMenu = glutCreateMenu(:processColorMenu)
glutAddMenuEntry("Red",C_RED);
glutAddMenuEntry("Blue",C_BLUE);
glutAddMenuEntry("Green",C_GREEN);
glutAddMenuEntry("Orange",C_ORANGE);

mainMenu = glutCreateMenu(:processMainMenu)

glutAddSubMenu("Polygon Mode", fillMenu)
glutAddSubMenu("Color", colorMenu)
glutAddSubMenu("Font",fontMenu)
// attach the menu to the right button
glutAttachMenu(GLUT_RIGHT_BUTTON)

// this will allow us to know if the menu is active
glutMenuStatusFunc(:processMenuStatus)

// -----
//          MAIN
// -----

func main

    // init GLUT and create window
    glutInit()
    glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA)
    glutInitWindowPosition(100,100)
    glutInitWindowSize(320,320)
    glutCreateWindow("RingFreeGLUT - Test 11")

    // register callbacks
    glutDisplayFunc(:renderScene)
    glutReshapeFunc(:changeSize)
    glutIdleFunc(:renderScene)

    glutIgnoreKeyRepeat(1)
    glutKeyboardFunc(:processNormalKeys)
    glutSpecialFunc(:pressKey)
    glutSpecialUpFunc(:releaseKey)

    // here are the two new functions
    glutMouseFunc(:mouseButton)
    glutMotionFunc(:mouseMove)

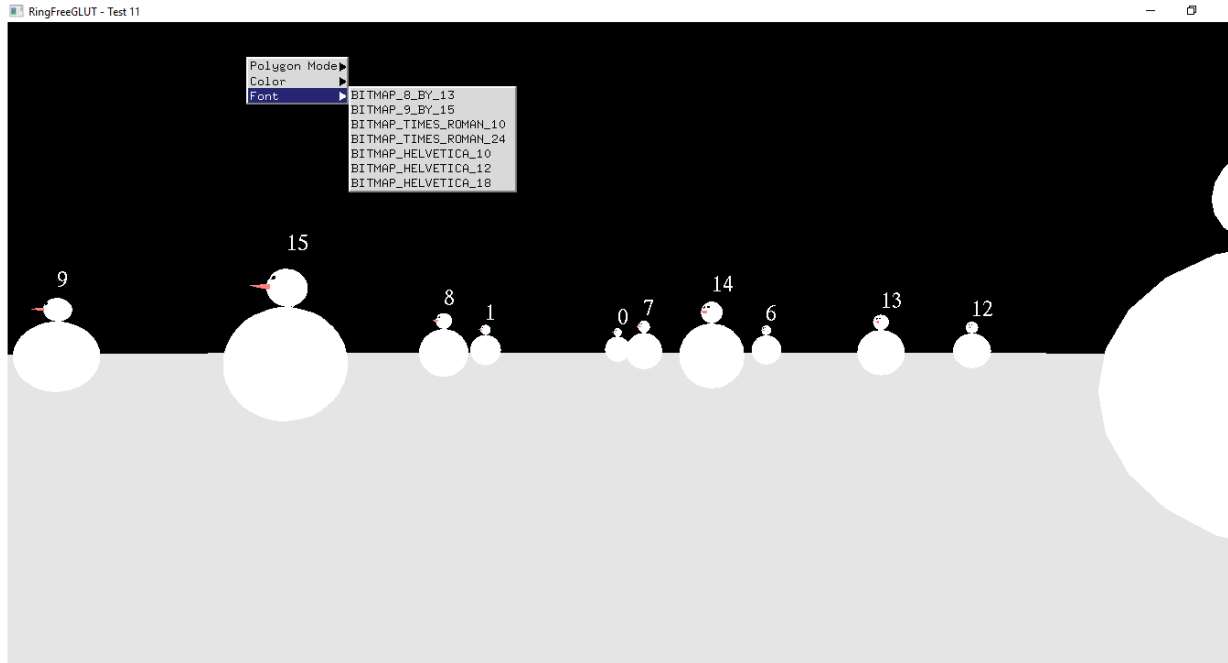
    // OpenGL init
    glEnable(GL_DEPTH_TEST)
    glEnable(GL_CULL_FACE)

    // init Menus
    createPopupMenus()

    // enter GLUT event processing cycle
    glutMainLoop()

```

Screen Shot



55.14 Frames Per Second

Example

```
load "freeglut.ring"
load "opengl21lib.ring"

// angle of rotation for the camera direction
angle = 0.0

// actual vector representing the camera's direction
lx=0.0 lz=-1.0

// XZ position of the camera
x=0.0 z=5.0

// the key states. These variables will be zero
//when no key is being presses
deltaAngle = 0.0
deltaMove = 0
xOrigin = -1

// Constant definitions for Menus
C_RED = 1
C_GREEN = 2
C_BLUE = 3
C_ORANGE = 4

C_FILL = 5
C_LINE = 6

// Pop up menu identifiers
fillMenu=NULL
```

```

fontMenu=NULL
mainMenu=NULL
colorMenu=NULL

// color for the nose
red = 1.0
blue=0.5
green=0.5

// scale of snowman
scale = 1.0

// menu status
menuFlag = 0

// default font
font = GLUT_BITMAP_TIMES_ROMAN_24

C_INT_GLUT_BITMAP_8_BY_13 = 7
C_INT_GLUT_BITMAP_9_BY_15 = 8
C_INT_GLUT_BITMAP_TIMES_ROMAN_10 = 9
C_INT_GLUT_BITMAP_TIMES_ROMAN_24 = 10
C_INT_GLUT_BITMAP_HELVETICA_10 = 11
C_INT_GLUT_BITMAP_HELVETICA_12 = 12
C_INT_GLUT_BITMAP_HELVETICA_18 = 13

// width and height of the window
h = 0
w = 0

// variables to compute frames per second
frame=0
time=0
timebase=0
s = ""

func changeSize
    w = glutEventWidth()
    h = glutEventHeight()

    // Prevent a divide by zero, when window is too short
    // (you cant make a window of zero width).
    if h = 0
        h = 1
    ok

    ratio = w * 1.0 / h

    // Use the Projection Matrix
    glMatrixMode(GL_PROJECTION)

    // Reset Matrix
    glLoadIdentity()

    // Set the viewport to be the entire window
    glViewport(0, 0, w, h)

    // Set the correct perspective.

```

```

        gluPerspective(45.0, ratio, 0.1, 100.0)

        // Get Back to the Modelview
        glMatrixMode(GL_MODELVIEW)

func drawSnowMan

        glScalef(scale, scale, scale)
        glColor3f(1.0, 1.0, 1.0)

// Draw Body
        glTranslatef(0.0 ,0.75, 0.0)
        glutSolidSphere(0.75,20,20)

// Draw Head
        glTranslatef(0.0, 1.0, 0.0)
        glutSolidSphere(0.25,20,20)

// Draw Eyes
        glPushMatrix()
        glColor3f(0.0,0.0,0.0)
        glTranslatef(0.05, 0.10, 0.18)
        glutSolidSphere(0.05,10,10)
        glTranslatef(-0.1, 0.0, 0.0)
        glutSolidSphere(0.05,10,10)
        glPopMatrix()

// Draw Nose
        glColor3f(red, green, blue)
        glRotatef(0.0,1.0, 0.0, 0.0)
        glutSolidCone(0.08,0.5,10,2)

        glColor3f(1.0, 1.0, 1.0)

func renderBitmapString x,y,z,font,string
        glRasterPos3f(x, y,z)
        for c in string
                glutBitmapCharacter(font,ascii(c))
        next

func renderStrokeFontString x,y,z,font,string
        glPushMatrix()
        glTranslatef(x, y,z)
        glScalef(0.002, 0.002, 0.002)
        for c in string
                glutStrokeCharacter(font, Ascii(c));
        next
        glPopMatrix()

func restorePerspectiveProjection

        glMatrixMode(GL_PROJECTION)
        // restore previous projection matrix
        glPopMatrix()

        // get back to modelview mode
        glMatrixMode(GL_MODELVIEW)

```

```

func setOrthographicProjection

    // switch to projection mode
    glMatrixMode(GL_PROJECTION)

    // save previous matrix which contains the
    // settings for the perspective projection
    glPushMatrix()

    // reset matrix
    glLoadIdentity()

    // set a 2D orthographic projection
    gluOrtho2D(0, w, h, 0)

    // switch back to modelview mode
    glMatrixMode(GL_MODELVIEW)

func computePos deltaMove

    x += deltaMove * lx * 0.1
    z += deltaMove * lz * 0.1

func renderScene

    if deltaMove
        computePos(deltaMove)
    ok

    // Clear Color and Depth Buffers
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

    // Reset transformations
    glLoadIdentity()

    // Set the camera
    gluLookAt(
        x, 1.0, z,
        x+lx, 1.0, z+lz,
        0.0, 1.0, 0.0)

    // Draw ground

    glColor3f(0.9, 0.9, 0.9)
    glBegin(GL_QUADS)
        glVertex3f(-100.0, 0.0, -100.0)
        glVertex3f(-100.0, 0.0, 100.0)
        glVertex3f(100.0, 0.0, 100.0)
        glVertex3f(100.0, 0.0, -100.0)
    glEnd()

    // Draw 9 SnowMen
    for i = -3 to -1
        for j = -3 to -1
            glPushMatrix()

```



```

        glTranslatef(i*10.0, 0.0, j * 10.0)
        drawSnowMan()
        number = (i+3)*3+(j+3)
        renderBitmapString(0.0, 0.5, 0.0,font ,""+number)
        glPopMatrix()

    next

next

// Code to compute frames per second
frame++

time=glutGet(GLUT_ELAPSED_TIME)
if time - timebase > 1000
    s = "RingFreeGLUT - FPS: " + (frame*1000.0/(time-timebase))
    timebase = time
    frame = 0

ok

// Code to display a string (fps) with bitmap fonts
setOrthographicProjection()

glPushMatrix()
glLoadIdentity()
renderBitmapString(5,30,0,GLUT_BITMAP_HELVETICA_18,s)
glPopMatrix()

restorePerspectiveProjection()

glutSwapBuffers()

// -----
//          KEYBOARD
// -----

func processNormalKeys
    key = glutEventKey()
    xx = glutEventX()
    yy = glutEventY()

    switch key
        on 27
            glutDestroyMenu(mainMenu)
            glutDestroyMenu(fillMenu)
            glutDestroyMenu(colorMenu)
            glutDestroyMenu(fontMenu)
            Shutdown()

    off

func pressKey

    key = glutEventKey()
    xx = glutEventX()
    yy = glutEventY()

    switch key
        on GLUT_KEY_UP

```

```

        deltaMove = 0.5
    on GLUT_KEY_DOWN
        deltaMove = -0.5
    off

func releaseKey

    key = glutEventKey()

    switch key
        on GLUT_KEY_UP
            deltaMove = 0
        on GLUT_KEY_DOWN
            deltaMove = 0
    off

// -----
//          MOUSE
// -----

func mouseMove
    xx = glutEventX()
    yy = glutEventY()

    // this will only be true when the left button is down
    if xOrigin >= 0

        // update deltaAngle
        deltaAngle = (xx - xOrigin) * 0.001

        // update camera's direction
        lx = sin(angle + deltaAngle)
        lz = -cos(angle + deltaAngle)
    ok

func mouseButton

    button = glutEventButton()
    state = glutEventState()
    xx = glutEventX()
    yy = glutEventY()

    // only start motion if the left button is pressed
    if button = GLUT_LEFT_BUTTON
        // when the button is released
        if state = GLUT_UP
            angle += deltaAngle
            xOrigin = -1
        else
            // state = GLUT_DOWN
            xOrigin = xx
    ok
ok

```

```
// -----
//          MENUS
// -----

func processMenuStatus

    status = glutEventStatus()

    if status = GLUT_MENU_IN_USE
        menuFlag = 1
    else
        menuFlag = 0
    ok

func processMainMenu

    // nothing to do in here
    // all actions are for submenus

func processFillMenu

    option = glutEventValue()

    switch option
        on C_FILL
            glPolygonMode(GL_FRONT, GL_FILL)
        on C_LINE
            glPolygonMode(GL_FRONT, GL_LINE)
    off

func processFontMenu

    option = glutEventValue()

    switch (option) {
        on C_INT_GLUT_BITMAP_8_BY_13
            font = GLUT_BITMAP_8_BY_13
        on C_INT_GLUT_BITMAP_9_BY_15
            font = GLUT_BITMAP_9_BY_15
        on C_INT_GLUT_BITMAP_TIMES_ROMAN_10
            font = GLUT_BITMAP_TIMES_ROMAN_10
        on C_INT_GLUT_BITMAP_TIMES_ROMAN_24
            font = GLUT_BITMAP_TIMES_ROMAN_24
        on C_INT_GLUT_BITMAP_HELVETICA_10
            font = GLUT_BITMAP_HELVETICA_10
        on C_INT_GLUT_BITMAP_HELVETICA_12
            font = GLUT_BITMAP_HELVETICA_12
        on C_INT_GLUT_BITMAP_HELVETICA_18
            font = GLUT_BITMAP_HELVETICA_18
    off

func processColorMenu

    option = glutEventValue()
```

```

switch option
    on C_RED
        red = 1.0
        green = 0.0
        blue = 0.0
    on C_GREEN
        red = 0.0
        green = 1.0
        blue = 0.0
    on C_BLUE
        red = 0.0
        green = 0.0
        blue = 1.0
    on C_ORANGE
        red = 1.0
        green = 0.5
        blue = 0.5
off

func createPopupMenu

    fontMenu = glutCreateMenu(:processFontMenu)

    glutAddMenuEntry("BITMAP_8_BY_13 ",C_INT_GLUT_BITMAP_8_BY_13 )
    glutAddMenuEntry("BITMAP_9_BY_15",C_INT_GLUT_BITMAP_9_BY_15 )
    glutAddMenuEntry("BITMAP_TIMES_ROMAN_10 ",C_INT_GLUT_BITMAP_TIMES_ROMAN_10 )
    glutAddMenuEntry("BITMAP_TIMES_ROMAN_24",C_INT_GLUT_BITMAP_TIMES_ROMAN_24 )
    glutAddMenuEntry("BITMAP_HELVETICA_10 ",C_INT_GLUT_BITMAP_HELVETICA_10 )
    glutAddMenuEntry("BITMAP_HELVETICA_12",C_INT_GLUT_BITMAP_HELVETICA_12 )
    glutAddMenuEntry("BITMAP_HELVETICA_18",C_INT_GLUT_BITMAP_HELVETICA_18 )

    fillMenu = glutCreateMenu(:processFillMenu)

    glutAddMenuEntry("Fill",C_FILL)
    glutAddMenuEntry("Line",C_LINE)

    colorMenu = glutCreateMenu(:processColorMenu)
    glutAddMenuEntry("Red",C_RED);
    glutAddMenuEntry("Blue",C_BLUE);
    glutAddMenuEntry("Green",C_GREEN);
    glutAddMenuEntry("Orange",C_ORANGE);

    mainMenu = glutCreateMenu(:processMainMenu)

    glutAddSubMenu("Polygon Mode", fillMenu)
    glutAddSubMenu("Color", colorMenu)
    glutAddSubMenu("Font", fontMenu)
    // attach the menu to the right button
    glutAttachMenu(GLUT_RIGHT_BUTTON)

    // this will allow us to know if the menu is active
    glutMenuStatusFunc(:processMenuStatus)

// -----
//             MAIN
// -----

```

```

func main

    // init GLUT and create window
    glutInit()
    glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA)
    glutInitWindowPosition(100,100)
    glutInitWindowSize(320,320)
    glutCreateWindow("RingFreeGLUT - Test - 9 SnowMan")

    // register callbacks
    glutDisplayFunc(:renderScene)
    glutReshapeFunc(:changeSize)
    glutIdleFunc(:renderScene)

    glutIgnoreKeyRepeat(1)
    glutKeyboardFunc(:processNormalKeys)
    glutSpecialFunc(:pressKey)
    glutSpecialUpFunc(:releaseKey)

    // here are the two new functions
    glutMouseFunc(:mouseButton)
    glutMotionFunc(:mouseMove)

    // OpenGL init
    glEnable(GL_DEPTH_TEST)
    glEnable(GL_CULL_FACE)

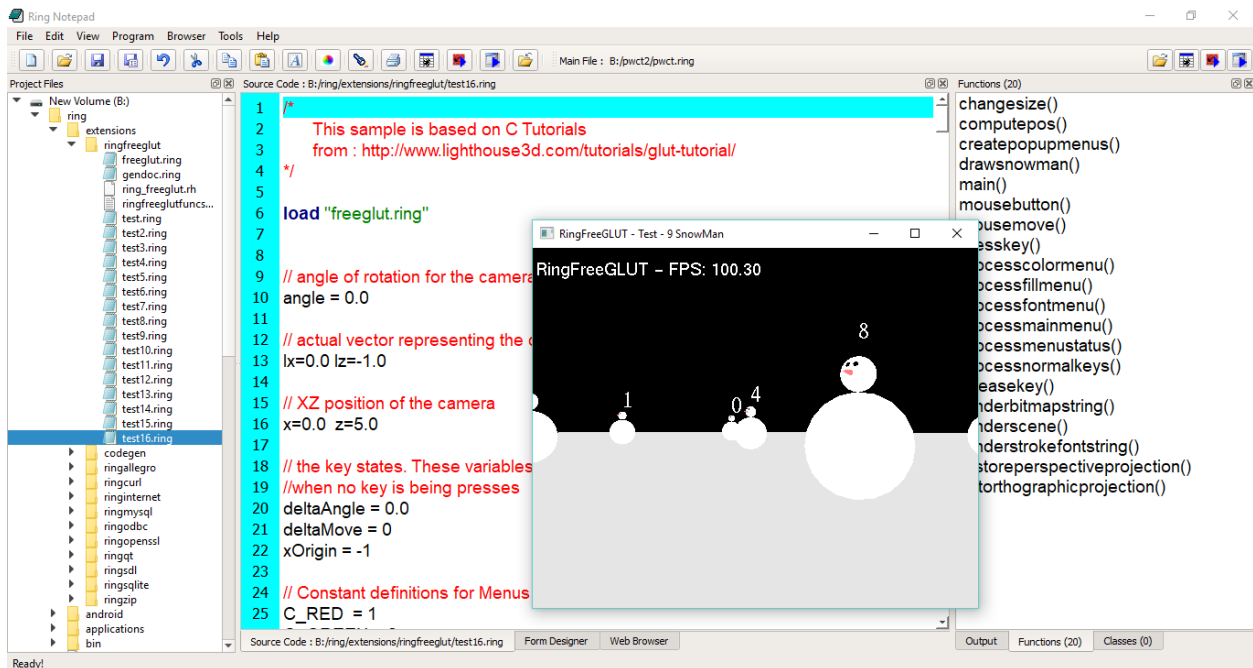
    // init Menus
    createPopupMenu()

    // enter GLUT event processing cycle
    glutMainLoop()

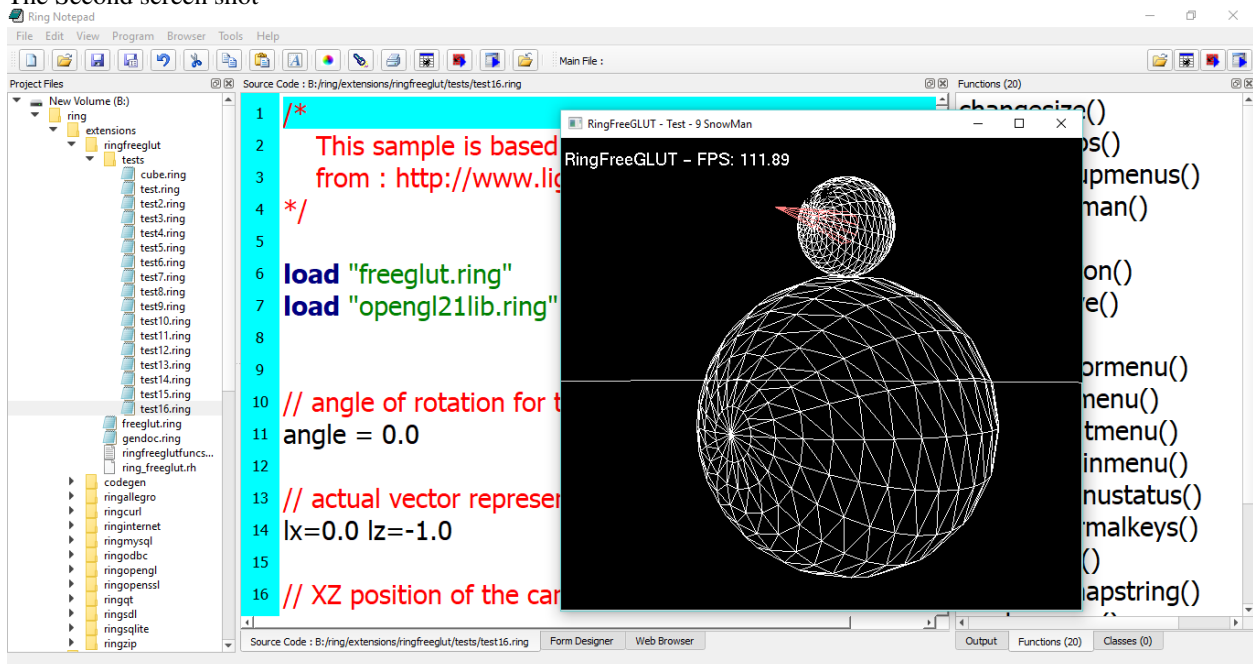
```

Screen Shots:

The First screen shot



The Second screen shot



55.15 Make a Cube using RingOpenGL and RingFreeGLUT

Example:

```
load "freelut.ring"
load "opengl21lib.ring"

// -----
```

```
// Global Variables
// -----
rotate_y=0
rotate_x=0

// -----
// display() Callback function
// -----
func display

    // Clear screen and Z-buffer
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)

    // Reset transformations
    glLoadIdentity()

    // Rotate when user changes rotate_x and rotate_y
    glRotatef( rotate_x, 1.0, 0.0, 0.0 )
    glRotatef( rotate_y, 0.0, 1.0, 0.0 )

    //Multi-colored side - FRONT
    glBegin(GL_POLYGON)

    glColor3f( 1.0, 0.0, 0.0 )    glVertex3f( 0.5, -0.5, -0.5 )    # P1 is red
    glColor3f( 0.0, 1.0, 0.0 )    glVertex3f( 0.5, 0.5, -0.5 )    # P2 is green
    glColor3f( 0.0, 0.0, 1.0 )    glVertex3f( -0.5, 0.5, -0.5 )    # P3 is blue
    glColor3f( 1.0, 0.0, 1.0 )    glVertex3f( -0.5, -0.5, -0.5 )    # P4 is purple

    glEnd()

    // White side - BACK
    glBegin(GL_POLYGON)
    glColor3f( 1.0, 1.0, 1.0 )
    glVertex3f( 0.5, -0.5, 0.5 )
    glVertex3f( 0.5, 0.5, 0.5 )
    glVertex3f( -0.5, 0.5, 0.5 )
    glVertex3f( -0.5, -0.5, 0.5 )
    glEnd()

    // Purple side - RIGHT
    glBegin(GL_POLYGON)
    glColor3f( 1.0, 0.0, 1.0 )
    glVertex3f( 0.5, -0.5, -0.5 )
    glVertex3f( 0.5, 0.5, -0.5 )
    glVertex3f( 0.5, 0.5, 0.5 )
    glVertex3f( 0.5, -0.5, 0.5 )
    glEnd()

    // Green side - LEFT
    glBegin(GL_POLYGON)
    glColor3f( 0.0, 1.0, 0.0 )
    glVertex3f( -0.5, -0.5, 0.5 )
    glVertex3f( -0.5, 0.5, 0.5 )
    glVertex3f( -0.5, 0.5, -0.5 )
    glVertex3f( -0.5, -0.5, -0.5 )
    glEnd()

    // Blue side - TOP
```

```

glBegin(GL_POLYGON)
glColor3f( 0.0, 0.0, 1.0 )
glVertex3f( 0.5, 0.5, 0.5 )
glVertex3f( 0.5, 0.5, -0.5 )
glVertex3f( -0.5, 0.5, -0.5 )
glVertex3f( -0.5, 0.5, 0.5 )
glEnd()

// Red side - BOTTOM
glBegin(GL_POLYGON)
glColor3f( 1.0, 0.0, 0.0 )
glVertex3f( 0.5, -0.5, -0.5 )
glVertex3f( 0.5, -0.5, 0.5 )
glVertex3f( -0.5, -0.5, 0.5 )
glVertex3f( -0.5, -0.5, -0.5 )
glEnd()

glFlush()
glutSwapBuffers()

// -----
// specialKeys() Callback Function
// -----
func specialKeys

    key = glutEventKey()

// Right arrow - increase rotation by 5 degree
    switch Key

    on GLUT_KEY_RIGHT
        rotate_y += 5

// Left arrow - decrease rotation by 5 degree
    on GLUT_KEY_LEFT
        rotate_y -= 5

    on GLUT_KEY_UP
        rotate_x += 5

    on GLUT_KEY_DOWN
        rotate_x -= 5

    off

// Request display update
glutPostRedisplay()

// -----
// main() function
// -----
func main

// Initialize GLUT and process user parameters
glutInit()

```



```
// Request double buffered true color window with Z-buffer
glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH)

// Create window
glutCreateWindow("Awesome Cube")

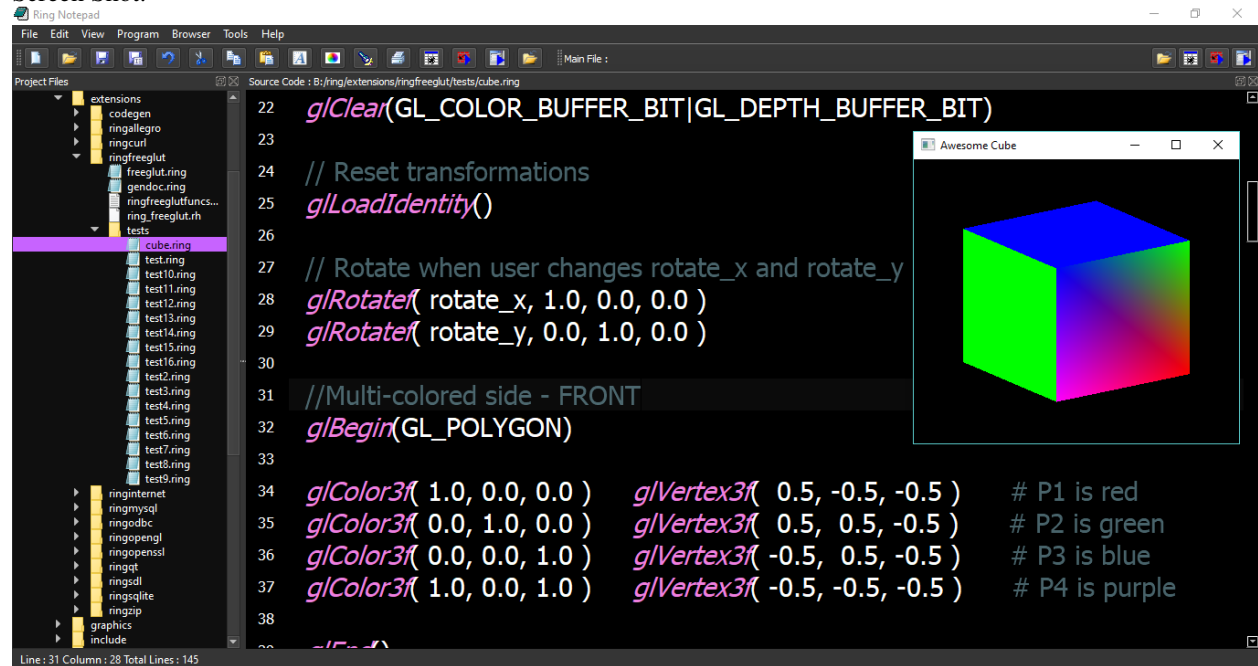
// Enable Z-buffer depth test
glEnable(GL_DEPTH_TEST)

// Callback functions
glutDisplayFunc(:display)
glutSpecialFunc(:specialKeys)

// Pass control to GLUT for events
glutMainLoop()

// Return to OS
```

Screen Shot:



USING RINGOPENGL AND RINGALLEGRO FOR 3D GRAPHICS

In this chapter we will learn about using RingOpenGL and RingAllegro

56.1 3D Cube and Texture

Source Code:

```
# Load Libraries
    load "gamelib.ring"           # RingAllegro Library
    load "opengl21lib.ring"       # RingOpenGL Library

=====
# To Support MacOS X
    al_run_main()
    func al_game_start           # Called by al_run_main()
        main()                  # Now we call our main function
=====

func main

    new GraphicsApp {
        start()
    }

class GraphicsApp from GraphicsAppBase

    TITLE = "Ring Cube"

    bitmap texture

    xrot = 0.0
    yrot = 0.0
    zrot = 0.0

    func loadresources

        bitmap = al_load_bitmap("ring.bmp")
        texture = al_get_opengl_texture(bitmap)

    func destroyResources

        al_destroy_bitmap(bitmap)
```

```

func drawScene

    w = 800 h = 600
    ratio = w / h

    glViewport(0, 0, w, h)
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()

    gluPerspective(45, ratio, 1, 100)
    glMatrixMode(GL_MODELVIEW)
    glLoadIdentity()

    glEnable(GL_TEXTURE_2D)
    glShadeModel(GL_SMOOTH)
    glClearColor(0.0, 0.0, 0.0, 0.5)
    glClearDepth(1.0)
    glEnable(GL_DEPTH_TEST)
    glEnable(GL_CULL_FACE)
    glDepthFunc(GL_LEQUAL)
    glHint(GL_PERSPECTIVE_CORRECTION_HINT, GL_NICEST)

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    glLoadIdentity()
    glTranslatef(0.0, 0.0, -5.0)

    glRotatef(xrot, 1.0, 0.0, 0.0)
    glRotatef(yrot, 0.0, 1.0, 0.0)
    glRotatef(zrot, 0.0, 0.0, 1.0)

    glBindTexture(GL_TEXTURE_2D, texture)

    glBegin(GL_QUADS)
        // Front Face
        glTexCoord2f(0.0, 0.0) glVertex3f(-1.0, -1.0, 1.0)
        glTexCoord2f(1.0, 0.0) glVertex3f( 1.0, -1.0, 1.0)
        glTexCoord2f(1.0, 1.0) glVertex3f( 1.0,  1.0, 1.0)
        glTexCoord2f(0.0, 1.0) glVertex3f(-1.0,  1.0, 1.0)
        // Back Face
        glTexCoord2f(1.0, 0.0) glVertex3f(-1.0, -1.0, -1.0)
        glTexCoord2f(1.0, 1.0) glVertex3f(-1.0,  1.0, -1.0)
        glTexCoord2f(0.0, 1.0) glVertex3f( 1.0,  1.0, -1.0)
        glTexCoord2f(0.0, 0.0) glVertex3f( 1.0, -1.0, -1.0)
        // Top Face
        glTexCoord2f(0.0, 1.0) glVertex3f(-1.0,  1.0, -1.0)
        glTexCoord2f(0.0, 0.0) glVertex3f(-1.0,  1.0,  1.0)
        glTexCoord2f(1.0, 0.0) glVertex3f( 1.0,  1.0,  1.0)
        glTexCoord2f(1.0, 1.0) glVertex3f( 1.0,  1.0, -1.0)
        // Bottom Face
        glTexCoord2f(1.0, 1.0) glVertex3f(-1.0, -1.0, -1.0)
        glTexCoord2f(0.0, 1.0) glVertex3f( 1.0, -1.0, -1.0)
        glTexCoord2f(0.0, 0.0) glVertex3f( 1.0, -1.0,  1.0)
        glTexCoord2f(1.0, 0.0) glVertex3f(-1.0, -1.0,  1.0)
        // Right face
        glTexCoord2f(1.0, 0.0) glVertex3f( 1.0, -1.0, -1.0)
        glTexCoord2f(1.0, 1.0) glVertex3f( 1.0,  1.0, -1.0)
        glTexCoord2f(0.0, 1.0) glVertex3f( 1.0,  1.0,  1.0)
        glTexCoord2f(0.0, 0.0) glVertex3f( 1.0, -1.0,  1.0)
    glEnd()

```

```

        // Left Face
        glTexCoord2f(0.0, 0.0) glVertex3f(-1.0, -1.0, -1.0)
        glTexCoord2f(1.0, 0.0) glVertex3f(-1.0, -1.0, 1.0)
        glTexCoord2f(1.0, 1.0) glVertex3f(-1.0, 1.0, 1.0)
        glTexCoord2f(0.0, 1.0) glVertex3f(-1.0, 1.0, -1.0)

    glEnd()

    xrot += 0.3
    yrot += 0.2
    zrot += 0.4

```

```
class GraphicsAppBase
```

```

    display event_queue ev timeout
    timer  redraw    = true

```

```
FPS                = 60
```

```
SCREEN_W           = 800
```

```
SCREEN_H           = 600
```

```
KEY_UP             = 1
```

```
KEY_DOWN           = 2
```

```
KEY_LEFT           = 3
```

```
KEY_RIGHT          = 4
```

```
Key = [false, false, false, false]
```

```
TITLE = "Graphics Application"
```

```
func start
```

```

    SetUp()
    loadResources()
    eventsLoop()
    destroy()

```

```
func setup
```

```

    al_init()
    al_init_image_addon()
    al_set_new_display_flags(ALLEGRO_OPENGL)
    display = al_create_display(SCREEN_W, SCREEN_H)
    al_set_window_title(display, TITLE)
    al_clear_to_color(al_map_rgb(0, 0, 0))
    event_queue = al_create_event_queue()
    al_register_event_source(event_queue,
        al_get_display_event_source(display))
    ev = al_new_allegro_event()
    timeout = al_new_allegro_timeout()
    al_init_timeout(timeout, 0.06)
    timer = al_create_timer(1.0 / FPS)
    al_register_event_source(event_queue,
        al_get_timer_event_source(timer))
    al_start_timer(timer)
    al_install_mouse()
    al_register_event_source(event_queue,

```

```

        al_get_mouse_event_source())
    al_install_keyboard()
    al_register_event_source(event_queue,
        al_get_keyboard_event_source())

func eventsLoop

    while true
        al_wait_for_event_until(event_queue, ev, timeout)
        switch al_get_allegro_event_type(ev)
        on ALLEGRO_EVENT_DISPLAY_CLOSE
            exit
        on ALLEGRO_EVENT_TIMER
            redraw = true
        on ALLEGRO_EVENT_MOUSE_AXES
            mouse_x = al_get_allegro_event_mouse_x(ev)
            mouse_y = al_get_allegro_event_mouse_y(ev)
        on ALLEGRO_EVENT_MOUSE_ENTER_DISPLAY
            mouse_x = al_get_allegro_event_mouse_x(ev)
            mouse_y = al_get_allegro_event_mouse_y(ev)
        on ALLEGRO_EVENT_MOUSE_BUTTON_UP
            exit
        on ALLEGRO_EVENT_KEY_DOWN
            switch al_get_allegro_event_keyboard_keycode(ev)
            on ALLEGRO_KEY_UP
                key[KEY_UP] = true
            on ALLEGRO_KEY_DOWN
                key[KEY_DOWN] = true
            on ALLEGRO_KEY_LEFT
                key[KEY_LEFT] = true
            on ALLEGRO_KEY_RIGHT
                key[KEY_RIGHT] = true
            off
        on ALLEGRO_EVENT_KEY_UP
            switch al_get_allegro_event_keyboard_keycode(ev)
            on ALLEGRO_KEY_UP
                key[KEY_UP] = false
            on ALLEGRO_KEY_DOWN
                key[KEY_DOWN] = false
            on ALLEGRO_KEY_LEFT
                key[KEY_LEFT] = false
            on ALLEGRO_KEY_RIGHT
                key[KEY_RIGHT] = false
            on ALLEGRO_KEY_ESCAPE
                exit
            off
        off
        if redraw and al_is_event_queue_empty(event_queue)
            redraw = false
            drawScene()
            al_flip_display()
        ok
        callgc()

    end

func destroy

    destroyResources()

```

```

al_destroy_timer(timer)
al_destroy_allegro_event(ev)
al_destroy_allegro_timeout(timeout)
al_destroy_event_queue(event_queue)
al_destroy_display(display)
al_exit()

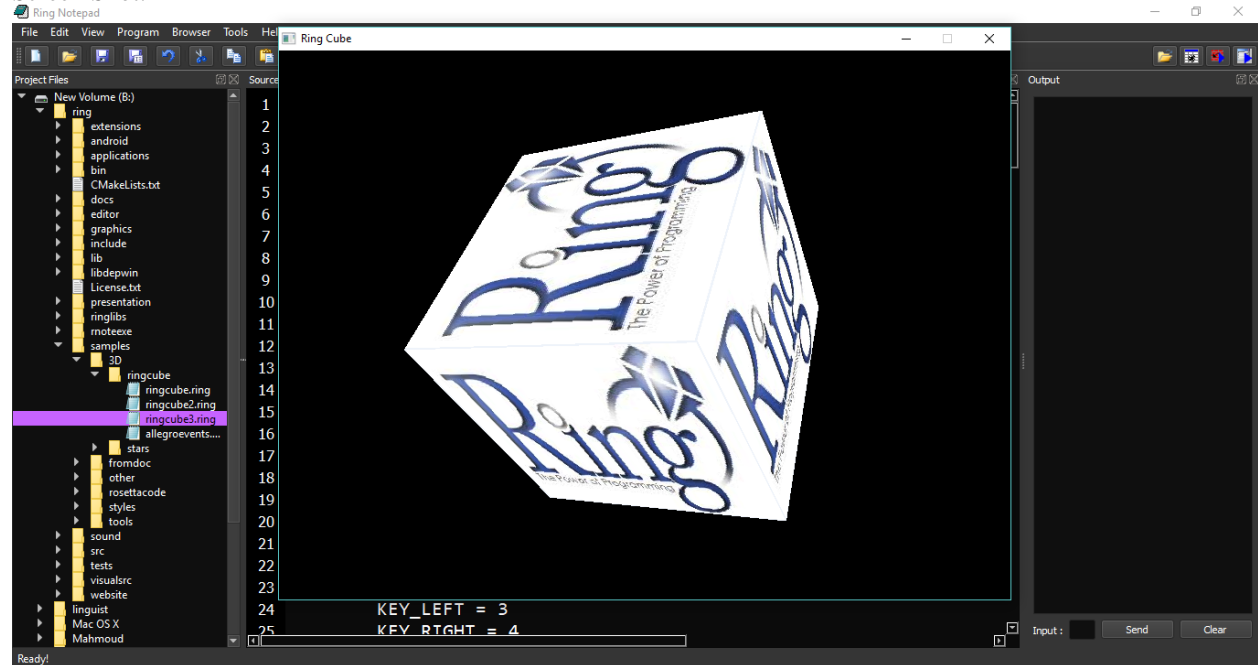
func loadresources

func drawScene

func destroyResources

```

Screen Shot:



56.2 Many Cubes

Source Code:

```

# Load Libraries
load "gamelib.ring"           # RingAllegro Library
load "opengl21lib.ring"      # RingOpenGL Library

=====
# To Support MacOS X
al_run_main()
func al_game_start           # Called by al_run_main()
    main()                   # Now we call our main function
=====

func main

```

```

new GraphicsApp {
    start()
}

class GraphicsApp from GraphicsAppBase

    TITLE = "Many Cubes"

    bitmap bitmap2 bitmap3
    texture texture2 texture3

    fps = 120
    xrot = 0.0
    yrot = 0.0
    zrot = 0.0

    nPerspective = 100

    func loadresources

        bitmap = al_load_bitmap("sky1.jpg")
        texture = al_get_opengl_texture(bitmap)

        bitmap2 = al_load_bitmap("sky2.jpg")
        texture2 = al_get_opengl_texture(bitmap2)

        bitmap3 = al_load_bitmap("sky3.jpg")
        texture3 = al_get_opengl_texture(bitmap3)

    func destroyResources

        al_destroy_bitmap(bitmap)
        al_destroy_bitmap(bitmap2)
        al_destroy_bitmap(bitmap3)

    func drawScene

        prepare()
        cubes()
        rotate()

    func Prepare
        w = 800 h = 600
        ratio = w / h
        glViewport(0, 0, w, h)
        glMatrixMode(GL_PROJECTION)
        glLoadIdentity()
        gluPerspective(-nPerspective, ratio, 1, nPerspective)
        glMatrixMode(GL_MODELVIEW)
        glLoadIdentity()
        glEnable(GL_TEXTURE_2D)
        glShadeModel(GL_SMOOTH)
        glClearColor(0.0, 0.0, 0.0, 0.5)
        glClearDepth(1.0)
        glEnable(GL_DEPTH_TEST)
        glEnable(GL_CULL_FACE)
        glDepthFunc(GL_LEQUAL)

```

```

glHint(GL_PERSPECTIVE_CORRECTION_HINT, GL_NICEST)
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

func Cubes
    cube(5,-3.4,-5,:sky1)
    cube(0,-3,-5,:sky1)
    cube(-5,-3,-5,:sky1)
    cube(5,0.5,-5,:sky2)
    cube(0,0.5,-5,:sky2)
    cube(-5,0.5,-5,:sky2)
    cube(5,4,-5,:sky3)
    cube(0,4,-5,:sky3)
    cube(-5,4,-5,:sky3)

func Rotate
    xrot += 0.3 * 5
    yrot += 0.2 * 5
    zrot += 0.4 * 5
    nPerspective += 0.5

func cube(x,y,z,nTexture)
    glLoadIdentity()
    glTranslatef(x,y,z)
    glRotatef(xrot,1.0,0.0,0.0)
    glRotatef(yrot,0.0,1.0,0.0)
    glRotatef(zrot,0.0,0.0,1.0)
    drawcube(nTexture)

func drawcube(cTexture)

    switch cTexture
        on :sky1
            glBindTexture(GL_TEXTURE_2D, texture)
        on :sky2
            glBindTexture(GL_TEXTURE_2D, texture2)
        on :sky3
            glBindTexture(GL_TEXTURE_2D, texture3)
    off

    glBegin(GL_QUADS)
        // Front Face
        glTexCoord2f(0.0, 0.0) glVertex3f(-1.0, -1.0, 1.0)
        glTexCoord2f(1.0, 0.0) glVertex3f( 1.0, -1.0, 1.0)
        glTexCoord2f(1.0, 1.0) glVertex3f( 1.0,  1.0, 1.0)
        glTexCoord2f(0.0, 1.0) glVertex3f(-1.0,  1.0, 1.0)
        // Back Face
        glTexCoord2f(1.0, 0.0) glVertex3f(-1.0, -1.0, -1.0)
        glTexCoord2f(1.0, 1.0) glVertex3f(-1.0,  1.0, -1.0)
        glTexCoord2f(0.0, 1.0) glVertex3f( 1.0,  1.0, -1.0)
        glTexCoord2f(0.0, 0.0) glVertex3f( 1.0, -1.0, -1.0)
        // Top Face
        glTexCoord2f(0.0, 1.0) glVertex3f(-1.0,  1.0, -1.0)
        glTexCoord2f(0.0, 0.0) glVertex3f(-1.0,  1.0,  1.0)
        glTexCoord2f(1.0, 0.0) glVertex3f( 1.0,  1.0,  1.0)
        glTexCoord2f(1.0, 1.0) glVertex3f( 1.0,  1.0, -1.0)
        // Bottom Face

```



```

glTexCoord2f(1.0, 1.0) glVertex3f(-1.0, -1.0, -1.0)
glTexCoord2f(0.0, 1.0) glVertex3f( 1.0, -1.0, -1.0)
glTexCoord2f(0.0, 0.0) glVertex3f( 1.0, -1.0,  1.0)
glTexCoord2f(1.0, 0.0) glVertex3f(-1.0, -1.0,  1.0)

// Right face
glTexCoord2f(1.0, 0.0) glVertex3f( 1.0, -1.0, -1.0)
glTexCoord2f(1.0, 1.0) glVertex3f( 1.0,  1.0, -1.0)
glTexCoord2f(0.0, 1.0) glVertex3f( 1.0,  1.0,  1.0)
glTexCoord2f(0.0, 0.0) glVertex3f( 1.0, -1.0,  1.0)

// Left Face
glTexCoord2f(0.0, 0.0) glVertex3f(-1.0, -1.0, -1.0)
glTexCoord2f(1.0, 0.0) glVertex3f(-1.0, -1.0,  1.0)
glTexCoord2f(1.0, 1.0) glVertex3f(-1.0,  1.0,  1.0)
glTexCoord2f(0.0, 1.0) glVertex3f(-1.0,  1.0, -1.0)

glEnd()

```

```
class GraphicsAppBase
```

```

    display event_queue ev timeout
    timer  redraw  = true

```

```
FPS = 60
```

```

SCREEN_W = 800
SCREEN_H = 600

```

```

KEY_UP      = 1
KEY_DOWN    = 2
KEY_LEFT    = 3
KEY_RIGHT   = 4

```

```
Key = [false, false, false, false]
```

```
TITLE = "Graphics Application"
```

```
func start
```

```

    SetUp()
    loadResources()
    eventsLoop()
    destroy()

```

```
func setup
```

```

    al_init()
    al_init_image_addon()
    al_set_new_display_flags(ALLEGRO_OPENGL)
    display = al_create_display(SCREEN_W, SCREEN_H)
    al_set_window_title(display, TITLE)
    al_clear_to_color(al_map_rgb(0, 0, 0))
    event_queue = al_create_event_queue()
    al_register_event_source(event_queue,
        al_get_display_event_source(display))
    ev = al_new_allegro_event()
    timeout = al_new_allegro_timeout()

```

```

al_init_timeout(timeout, 0.06)
timer = al_create_timer(1.0 / FPS)
al_register_event_source(event_queue,
    al_get_timer_event_source(timer))
al_start_timer(timer)
al_install_mouse()
al_register_event_source(event_queue,
    al_get_mouse_event_source())
al_install_keyboard()
al_register_event_source(event_queue,
    al_get_keyboard_event_source())

func eventsLoop

    while true
        al_wait_for_event_until(event_queue, ev, timeout)
        switch al_get_allegro_event_type(ev)
        on ALLEGRO_EVENT_DISPLAY_CLOSE
            exit
        on ALLEGRO_EVENT_TIMER
            redraw = true
        on ALLEGRO_EVENT_MOUSE_AXES
            mouse_x = al_get_allegro_event_mouse_x(ev)
            mouse_y = al_get_allegro_event_mouse_y(ev)
        on ALLEGRO_EVENT_MOUSE_ENTER_DISPLAY
            mouse_x = al_get_allegro_event_mouse_x(ev)
            mouse_y = al_get_allegro_event_mouse_y(ev)
        on ALLEGRO_EVENT_MOUSE_BUTTON_UP
            exit
        on ALLEGRO_EVENT_KEY_DOWN
            switch al_get_allegro_event_keyboard_keycode(ev)
            on ALLEGRO_KEY_UP
                key[KEY_UP] = true
            on ALLEGRO_KEY_DOWN
                key[KEY_DOWN] = true
            on ALLEGRO_KEY_LEFT
                key[KEY_LEFT] = true
            on ALLEGRO_KEY_RIGHT
                key[KEY_RIGHT] = true
            off
        on ALLEGRO_EVENT_KEY_UP
            switch al_get_allegro_event_keyboard_keycode(ev)
            on ALLEGRO_KEY_UP
                key[KEY_UP] = false
            on ALLEGRO_KEY_DOWN
                key[KEY_DOWN] = false
            on ALLEGRO_KEY_LEFT
                key[KEY_LEFT] = false
            on ALLEGRO_KEY_RIGHT
                key[KEY_RIGHT] = false
            on ALLEGRO_KEY_ESCAPE
                exit
            off
        off
        if redraw and al_is_event_queue_empty(event_queue)
            redraw = false
            drawScene()
            al_flip_display()

```

```
        ok
        callgc()
    end

func destroy

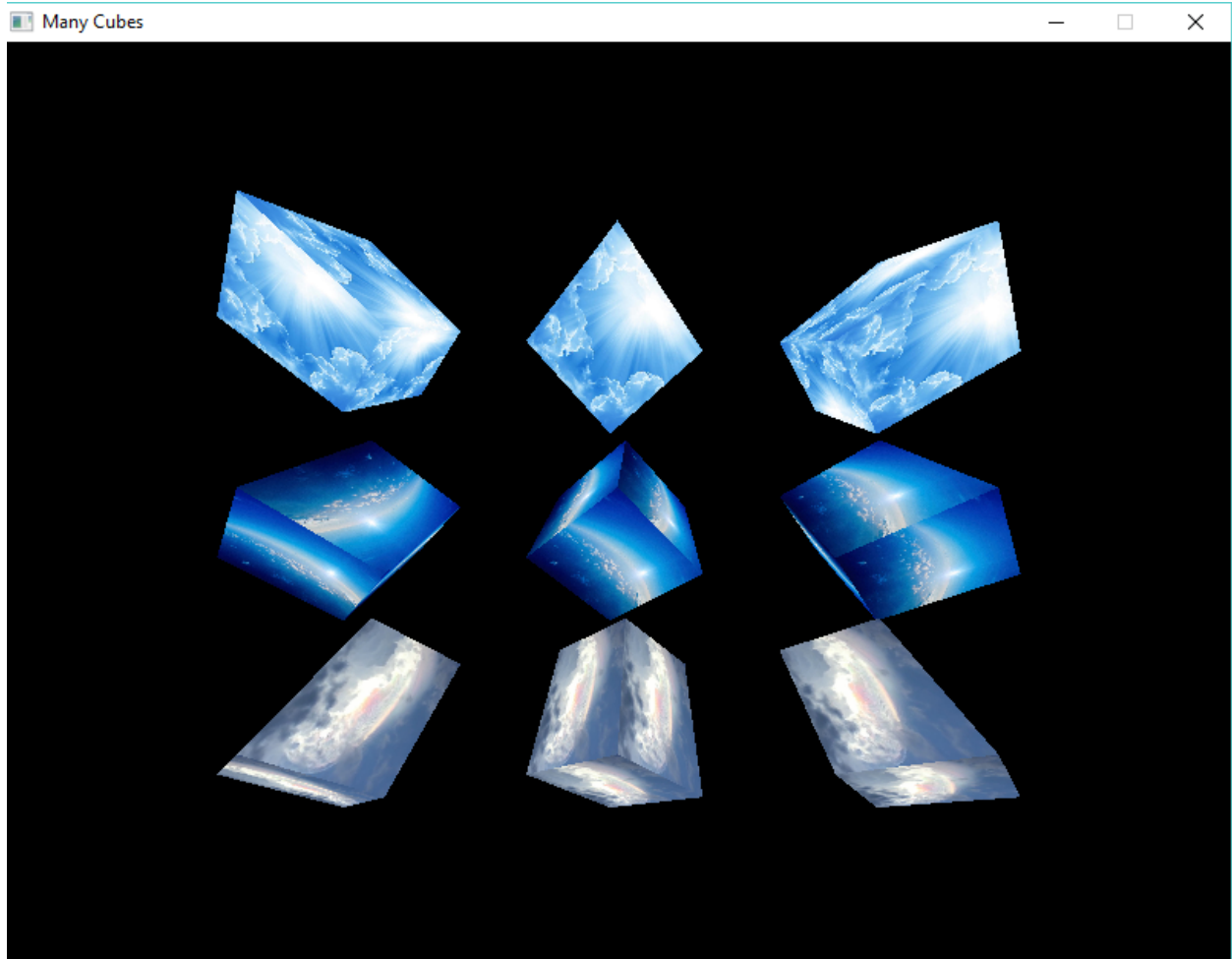
    destroyResources()
    al_destroy_timer(timer)
    al_destroy_allegro_event(ev)
    al_destroy_allegro_timeout(timeout)
    al_destroy_event_queue(event_queue)
    al_destroy_display(display)
    al_exit()

func loadresources

func drawScene

func destroyResources
```

Screen Shot:



56.3 TicTacToe 3D Game

Source Code:

```
# Load Libraries
    load "gamelib.ring"           # RingAllegro Library
    load "opengl21lib.ring"       # RingOpenGL Library

=====
# To Support MacOS X
    al_run_main()
    func al_game_start            # Called by al_run_main()
        main()                   # Now we call our main function
=====

func main
    new TicTacToe3D {
        start()
    }

class TicTacToe3D from GameLogic

    FPS = 60
    TITLE = "TicTacToe 3D"

    oBackground = new GameBackground
    oGameSound = new GameSound
    oGameCube = new GameCube
    oGameOver = new GameOver
    oGameInterface = new GameInterface

    func loadresources
        oGameOver.loadresources()
        oGameSound.loadresources()
        oBackGround.loadresources()
        oGameCube.loadresources()

    func destroyResources
        oGameOver.destroyResources()
        oGameSound.destroyResources()
        oBackGround.destroyResources()
        oGameCube.destroyResources()

    func drawScene
        oBackground.update()
        oGameInterface.update(self)

    func MouseClickEvent
        oGameInterface.MouseClickEvent(self)

class GameInterface

    func Update oGame
        prepare()
        cubes(oGame)

    func Prepare
        w = 1024 h = 768
```

```

ratio = w / h
glViewport(0, 0, w, h)
glMatrixMode(GL_PROJECTION)
glLoadIdentity()
gluPerspective(-120, ratio, 1, 120)
glMatrixMode(GL_MODELVIEW)
glLoadIdentity()
glEnable(GL_TEXTURE_2D)
glShadeModel(GL_SMOOTH)
glClearColor(0.0, 0.0, 0.0, 0.5)
glClearDepth(1.0)
glEnable(GL_DEPTH_TEST)
glEnable(GL_CULL_FACE)
glDepthFunc(GL_LEQUAL)
glHint(GL_PERSPECTIVE_CORRECTION_HINT, GL_NICEST)

func Cubes oGame
    oGame.oGameCube {
        aGameMap = oGame.aGameMap
        cube( 5 , -3 , -5 , aGameMap[1][1] )
        cube( 0 , -3 , -5 , aGameMap[1][2] )
        cube( -5 , -3 , -5 , aGameMap[1][3] )
        cube( 5 , 1 , -5 , aGameMap[2][1] )
        cube( 0 , 1 , -5 , aGameMap[2][2] )
        cube( -5 , 1 , -5 , aGameMap[2][3] )
        cube( 5 , 5 , -5 , aGameMap[3][1] )
        cube( 0 , 5 , -5 , aGameMap[3][2] )
        cube( -5 , 5 , -5 , aGameMap[3][3] )
        rotate()
    }

func MouseClickEvent oGame
    oGame {
        aBtn = Point2Button(Mouse_X, Mouse_Y)
        nRow = aBtn[1]
        nCol = aBtn[2]
        if nRow != 0 and nCol != 0
            if aGameMap[nRow][nCol] = :n
                aGameMap[nRow][nCol] = cActivePlayer
                ChangeActivePlayer()
                CheckGameOver()
            ok
        ok
    }

Class GameLogic from GraphicsAppBase

    aGameMap = [
        [ :n , :n , :n ] ,
        [ :n , :n , :n ] ,
        [ :n , :n , :n ]
    ]

    aGameButtons = [
        [176, 88, 375, 261], # x1, y1, x2, y2
        [176, 88, 375, 261], # [1, 1]
        [423, 88, 591, 261], # [1, 2]
        [645, 88, 876, 261], # [1, 3]
        [176, 282, 375, 428], # [2, 1]
    ]

```

```

[423,282,591,428],      # [2,2]
[645,282,876,428],      # [2,3]
[176,454,375,678],      # [3,1]
[423,454,591,678],      # [3,2]
[645,454,876,678]       # [3,3]
]

cActivePlayer = :x

func point2button x,y
  nRow = 0
  nCol = 0
  for t = 1 to len(aGameButtons)
    rect = aGameButtons[t]
    if x >= rect[1] and x <= rect[3] and
      y >= rect[2] and y <= rect[4]
      switch t
        on 1 nRow = 1 nCol = 1
        on 2 nRow = 1 nCol = 2
        on 3 nRow = 1 nCol = 3
        on 4 nRow = 2 nCol = 1
        on 5 nRow = 2 nCol = 2
        on 6 nRow = 2 nCol = 3
        on 7 nRow = 3 nCol = 1
        on 8 nRow = 3 nCol = 2
        on 9 nRow = 3 nCol = 3
      off
      exit
    ok
  next
  return [nRow,nCol]

func ChangeActivePlayer()
  if cActivePlayer = :x
    cActivePlayer = :o
  else
    cActivePlayer = :x
  ok

func CheckGameOver
  aList = [
    aGameMap[1][1],
    aGameMap[1][2],
    aGameMap[1][3],
    aGameMap[2][1],
    aGameMap[2][2],
    aGameMap[2][3],
    aGameMap[3][1],
    aGameMap[3][2],
    aGameMap[3][3]
  ]
  for item in aList
    switch item
      on :x item = 1
      on :o item = 2
      on :n item = 0
    off
  next

```

```

nStatus = CheckWinner(aList)
if nStatus
    oGameOver {
        Switch nStatus
            on 1 Player1Win(this)
            on 2 Player2Win(this)
            on 3 NoOneWin(this)

        off
    }
    refreshGame()

ok

func refreshGame
    aGameMap = [
        [ :n , :n , :n ] ,
        [ :n , :n , :n ] ,
        [ :n , :n , :n ]
    ]
    cActivePlayer = :x

func CheckWinner lst
    //vertical check
    for v=1 to 9 step 3
        if lst[v]!=0 and lst[v+1]!=0 and lst[v+2]!=0
            if lst[v]=lst[v+1] and lst[v+1]=lst[v+2]
                return lst[v]
            ok
        ok
    next
    //horzintal
    for h=1 to 3
        if lst[h]!=0 and lst[h+3]!=0 and lst[h+6]!=0
            if lst[h]=lst[h+3] and lst[h+3]=lst[h+6]
                return lst[h]
            ok
        ok
    next
    //Cross
    if lst[1]!=0 and lst[5]!=0 and lst[9]!=0
        if lst[1]=lst[5] and lst[5]=lst[9] return lst[1] ok
    ok
    if lst[3]!=0 and lst[5]!=0 and lst[7]!=0
        if lst[3]=lst[5] and lst[5]=lst[7] return lst[3] ok
    ok
    //tie
    tie=true
    for i=1 to 9
        if lst[i]=0 tie=false exit ok
    next
    if tie=true return 3 ok return 0

class GameOver

    font bitmap

    func loadresources

```

```

        font = al_load_ttf_font("font/pirulen.ttf",54,0 )
        bitmap = al_load_bitmap("image/ballon.png")

func destroyResources
    al_destroy_bitmap(bitmap)
    al_destroy_font(font)

func Player1Win oGame
    showMsg(oGame,80,430,"Good job X you won!")

func Player2Win oGame
    showMsg(oGame,80,430,"Good job O you won!")

func NoOneWin oGame
    showMsg(oGame,150,430,"Oh no it's a tie!")

func ShowMsg oGame,x,y,cMsg
    oGame {
        drawScene()
        al_flip_display()
        al_rest(0.3)
        newdisplay = al_create_display(SCREEN_W,SCREEN_H)
        al_set_window_title(newdisplay,TITLE)
        al_clear_to_color(al_map_rgb(255,255,255))
        al_draw_bitmap(this.bitmap,200,50,1)
        al_draw_text(this.font,
                     al_map_rgb(0,0,255), x,y,
                     ALLEGRO_ALIGN_LEFT,cMsg)
        al_flip_display()
        al_rest(2)
        al_destroy_display(newdisplay)
        al_set_target_backbuffer(display)
    }

```

```
class GameCube
```

```

    bitmap bitmap2 bitmap3
    textureX textureO textureN

    xrot = 0.0
    yrot = 0.0
    zrot = 0.0

func loadresources
    bitmap = al_load_bitmap("image/o.png")
    textureO = al_get_opengl_texture(bitmap)
    bitmap2 = al_load_bitmap("image/x.png")
    textureX = al_get_opengl_texture(bitmap2)
    bitmap3 = al_load_bitmap("image/empty.png")
    textureN = al_get_opengl_texture(bitmap3)

func destroyResources
    al_destroy_bitmap(bitmap)
    al_destroy_bitmap(bitmap2)
    al_destroy_bitmap(bitmap3)

func cube(x,y,z,nTexture)
    glLoadIdentity()

```



```

glTranslatef(x,y,z)
glRotatef(xrot,1.0,0.0,0.0)
glRotatef(yrot,0.0,1.0,0.0)
glRotatef(zrot,0.0,0.0,1.0)
setCubeTexture(nTexture)
drawCube()

func setCubeTexture cTexture
    switch cTexture
        on :x
            glBindTexture(GL_TEXTURE_2D, textureX)
        on :o
            glBindTexture(GL_TEXTURE_2D, textureO)
        on :n
            glBindTexture(GL_TEXTURE_2D, textureN)
    off

func Rotate
    xrot += 0.3 * 5
    yrot += 0.2 * 5
    zrot += 0.4 * 5

func drawcube
    glBegin(GL_QUADS)
        // Front Face
        glTexCoord2f(0.0, 0.0) glVertex3f(-1.0, -1.0, 1.0)
        glTexCoord2f(1.0, 0.0) glVertex3f( 1.0, -1.0, 1.0)
        glTexCoord2f(1.0, 1.0) glVertex3f( 1.0,  1.0, 1.0)
        glTexCoord2f(0.0, 1.0) glVertex3f(-1.0,  1.0, 1.0)
        // Back Face
        glTexCoord2f(1.0, 0.0) glVertex3f(-1.0, -1.0, -1.0)
        glTexCoord2f(1.0, 1.0) glVertex3f(-1.0,  1.0, -1.0)
        glTexCoord2f(0.0, 1.0) glVertex3f( 1.0,  1.0, -1.0)
        glTexCoord2f(0.0, 0.0) glVertex3f( 1.0, -1.0, -1.0)
        // Top Face
        glTexCoord2f(0.0, 1.0) glVertex3f(-1.0,  1.0, -1.0)
        glTexCoord2f(0.0, 0.0) glVertex3f(-1.0,  1.0,  1.0)
        glTexCoord2f(1.0, 0.0) glVertex3f( 1.0,  1.0,  1.0)
        glTexCoord2f(1.0, 1.0) glVertex3f( 1.0,  1.0, -1.0)
        // Bottom Face
        glTexCoord2f(1.0, 1.0) glVertex3f(-1.0, -1.0, -1.0)
        glTexCoord2f(0.0, 1.0) glVertex3f( 1.0, -1.0, -1.0)
        glTexCoord2f(0.0, 0.0) glVertex3f( 1.0, -1.0,  1.0)
        glTexCoord2f(1.0, 0.0) glVertex3f(-1.0, -1.0,  1.0)

        // Right face
        glTexCoord2f(1.0, 0.0) glVertex3f( 1.0, -1.0, -1.0)
        glTexCoord2f(1.0, 1.0) glVertex3f( 1.0,  1.0, -1.0)
        glTexCoord2f(0.0, 1.0) glVertex3f( 1.0,  1.0,  1.0)
        glTexCoord2f(0.0, 0.0) glVertex3f( 1.0, -1.0,  1.0)

        // Left Face
        glTexCoord2f(0.0, 0.0) glVertex3f(-1.0, -1.0, -1.0)
        glTexCoord2f(1.0, 0.0) glVertex3f(-1.0, -1.0,  1.0)
        glTexCoord2f(1.0, 1.0) glVertex3f(-1.0,  1.0,  1.0)
        glTexCoord2f(0.0, 1.0) glVertex3f(-1.0,  1.0, -1.0)
    glEnd()

```

```

class GameBackground

    nBackX = 0
    nBackY = 0
    nBackDiffx = -1
    nBackDiffy = -1
    nBackMotion = 1
    aBackMotionList = [
        [ -1, -1 ] ,      # Down - Right
        [ 0 , 1 ] ,      # Up
        [ -1, -1 ] ,      # Down - Right
        [ 0 , 1 ] ,      # Up
        [ 1 , -1 ] ,      # Down - Left
        [ 0 , 1 ] ,      # Up
        [ 1 , -1 ] ,      # Down - Left
        [ 0 , 1 ] ,      # Up
    ]

    bitmap

    func Update
        draw()
        motion()

    func draw
        al_draw_bitmap(bitmap,nBackX,nBackY,1)

    func motion
        nBackX += nBackDiffx
        nBackY += nBackDiffy
        if (nBackY = -350) or (nBackY = 0)
            nBackMotion++
            if nBackMotion > len(aBackMotionList)
                nBackMotion = 1

            ok
            nBackDiffx = aBackMotionList[nBackMotion][1]
            nBackDiffy = aBackMotionList[nBackMotion][2]

        ok

    func loadResources
        bitmap = al_load_bitmap("image/back.jpg")

    func destroyResources
        al_destroy_bitmap(bitmap)

class GameSound

    sample sampleid

    func loadresources
        sample = al_load_sample( "sound/music1.wav" )
        sampleid = al_new_allegro_sample_id()
        al_play_sample(sample, 1.0, 0.0,1.0,ALLEGRO_PLAYMODE_LOOP,sampleid)

    func destroyResources
        al_destroy_allegro_sample_id(sampleid)
        al_destroy_sample(sample)

```

```

class GraphicsAppBase

    display event_queue ev timeout
    timer
    redraw                = true
    FPS                    = 60
    SCREEN_W               = 1024
    SCREEN_H               = 700
    KEY_UP                 = 1
    KEY_DOWN               = 2
    KEY_LEFT               = 3
    KEY_RIGHT              = 4
    Key                    = [false, false, false, false]
    Mouse_X                = 0
    Mouse_Y                = 0
    TITLE                  = "Graphics Application"
    PRINT_MOUSE_XY        = False

    func start
        SetUp()
        loadResources()
        eventsLoop()
        destroy()

    func setup
        al_init()
        al_init_font_addon()
        al_init_ttf_addon()
        al_init_image_addon()
        al_install_audio()
        al_init_acodec_addon()
        al_reserve_samples(1)
        al_set_new_display_flags(ALLEGRO_OPENGL)
        display = al_create_display(SCREEN_W, SCREEN_H)
        al_set_window_title(display, TITLE)
        al_clear_to_color(al_map_rgb(0, 0, 0))
        event_queue = al_create_event_queue()
        al_register_event_source(event_queue,
                                al_get_display_event_source(display))
        ev = al_new_allegro_event()
        timeout = al_new_allegro_timeout()
        al_init_timeout(timeout, 0.06)
        timer = al_create_timer(1.0 / FPS)
        al_register_event_source(event_queue,
                                al_get_timer_event_source(timer))
        al_start_timer(timer)
        al_install_mouse()
        al_register_event_source(event_queue,
                                al_get_mouse_event_source())
        al_install_keyboard()
        al_register_event_source(event_queue,
                                al_get_keyboard_event_source())

    func eventsLoop
        while true
            al_wait_for_event_until(event_queue, ev, timeout)
            switch al_get_allegro_event_type(ev)

```

```

on ALLEGRO_EVENT_DISPLAY_CLOSE
    CloseEvent ()
on ALLEGRO_EVENT_TIMER
    redraw = true
on ALLEGRO_EVENT_MOUSE_AXES
    mouse_x = al_get_allegro_event_mouse_x(ev)
    mouse_y = al_get_allegro_event_mouse_y(ev)
    if PRINT_MOUSE_XY
        see "x = " + mouse_x + nl
        see "y = " + mouse_y + nl
    ok
on ALLEGRO_EVENT_MOUSE_ENTER_DISPLAY
    mouse_x = al_get_allegro_event_mouse_x(ev)
    mouse_y = al_get_allegro_event_mouse_y(ev)
on ALLEGRO_EVENT_MOUSE_BUTTON_UP
    MouseClickEvent ()
on ALLEGRO_EVENT_KEY_DOWN
    switch al_get_allegro_event_keyboard_keycode(ev)
        on ALLEGRO_KEY_UP
            key[KEY_UP] = true
        on ALLEGRO_KEY_DOWN
            key[KEY_DOWN] = true
        on ALLEGRO_KEY_LEFT
            key[KEY_LEFT] = true
        on ALLEGRO_KEY_RIGHT
            key[KEY_RIGHT] = true
    off
on ALLEGRO_EVENT_KEY_UP
    switch al_get_allegro_event_keyboard_keycode(ev)
        on ALLEGRO_KEY_UP
            key[KEY_UP] = false
        on ALLEGRO_KEY_DOWN
            key[KEY_DOWN] = false
        on ALLEGRO_KEY_LEFT
            key[KEY_LEFT] = false
        on ALLEGRO_KEY_RIGHT
            key[KEY_RIGHT] = false
        on ALLEGRO_KEY_ESCAPE
            exit
    off
off
if redraw and al_is_event_queue_empty(event_queue)
    redraw = false
    drawScene()
    al_flip_display()
ok
callgc()

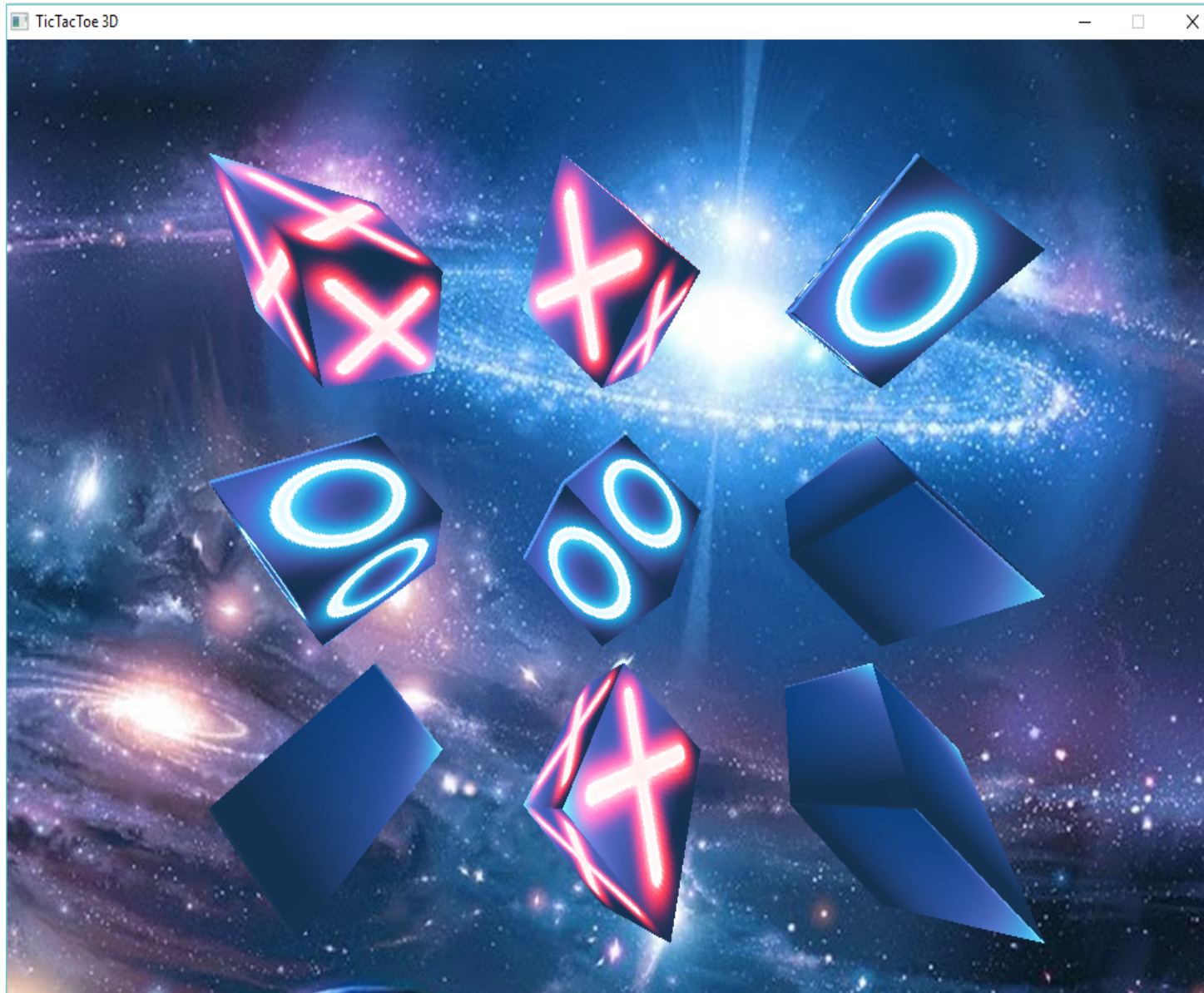
end

func destroy
    destroyResources()
    al_destroy_timer(timer)
    al_destroy_allegro_event(ev)
    al_destroy_allegro_timeout(timeout)
    al_destroy_event_queue(event_queue)
    al_destroy_display(display)
    al_exit()

```

```
func loadresources  
  
func drawScene  
  
func destroyResources  
  
func MouseClickEvent  
    exit # Exit from the Events Loop  
  
func CloseEvent  
    exit # Exit from the Events Loop
```

Screen Shot:



DESKTOP AND MOBILE DEVELOPMENT USING RINGQT

In this chapter we will learn how to use the Qt framework classes in our Ring applications to create Desktop and Mobile Applications.

57.1 The First GUI Application

In this example we will create an application to ask the user about his/her name. When the user type the name in the textbox then click on “Say Hello” button, the textbox value will be updated by adding “Hello ” to the name.

```
Load "guilib.ring"

MyApp = New qApp {

    win1 = new QWidget() {

        setWindowTitle("Hello World")
        setGeometry(100,100,370,250)

        label1 = new QLabel(win1) {
            settext("What is your name ?")
            setGeometry(10,20,350,30)
            setalignment(Qt_AlignHCenter)
        }

        btn1 = new QPushButton(win1) {
            setGeometry(10,200,100,30)
            settext("Say Hello")
            setclideanvent("pHello() ")
        }

        btn1 = new QPushButton(win1) {
            setGeometry(150,200,100,30)
            settext("Close")
            setclideanvent("pClose() ")
        }

        linedit1 = new QLineEdit(win1) {
            setGeometry(10,100,350,30)
        }

        show()
    }
}
```

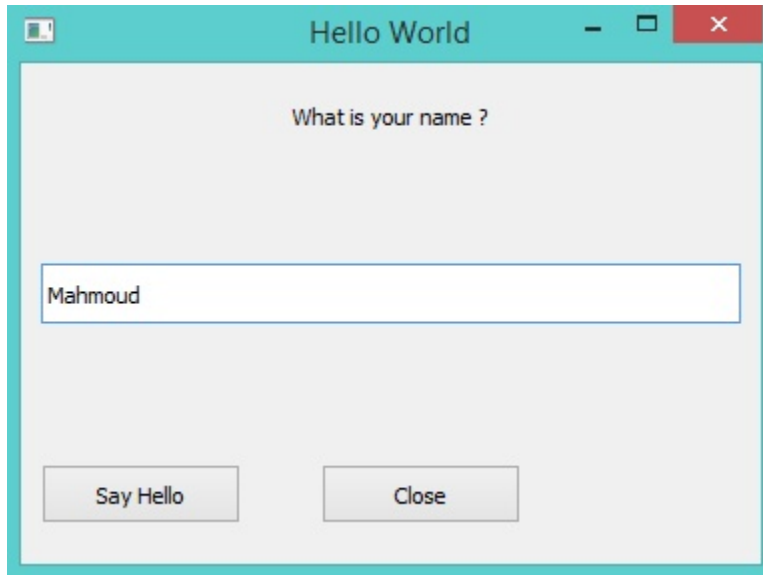
```
        exec()
    }

    Func pHello
        linedit1.settext( "Hello " + linedit1.text())

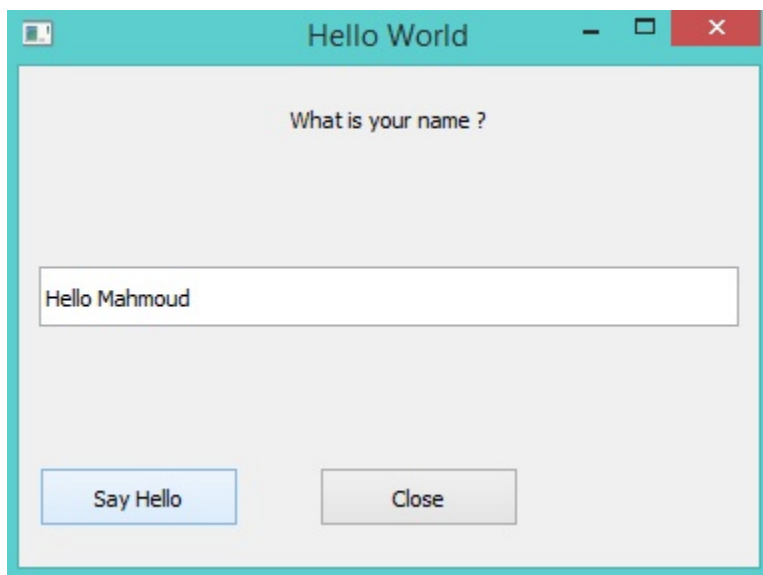
    Func pClose
        MyApp.quit()
```

Program Output:

At first we type the name in the textbox



Then we click on the say hello button



57.2 Using Layout

The next example is just an upgrade to the previous application to use the vertical layout.

```
Load "guilib.ring"

MyApp = New qApp {

    win1 = new QWidget() {

        setWindowTitle("Hello World")
        setGeometry(100,100,400,130)
        label1 = new QLabel(win1) {
            setText("What is your name ?")
            setGeometry(10,20,350,30)
            setAlignment(Qt_AlignHCenter)
        }
        btn1 = new QPushButton(win1) {
            setGeometry(10,200,100,30)
            setText("Say Hello")
            setClickedEvent("pHello()")
        }
        btn2 = new QPushButton(win1) {
            setGeometry(150,200,100,30)
            setText("Close")
            setClickedEvent("pClose()")
        }
        lineedit1 = new QLineEdit(win1) {
            setGeometry(10,100,350,30)
        }
        layout1 = new QVBoxLayout() {
            addWidget(label1)
            addWidget(lineedit1)
            addWidget(btn1)
            addWidget(btn2)
        }
        win1.setLayout(layout1)
        show()

    }

    exec()

}

Func pHello
    lineedit1.setText( "Hello " + lineedit1.text() )

Func pClose
    MyApp.quit()
```

The application during the runtime!



57.3 Using the QTextEdit Class

In this example we will use the QTextEdit Class

```
Load "guilib.ring"

New qApp {

    win1 = new QWidget() {

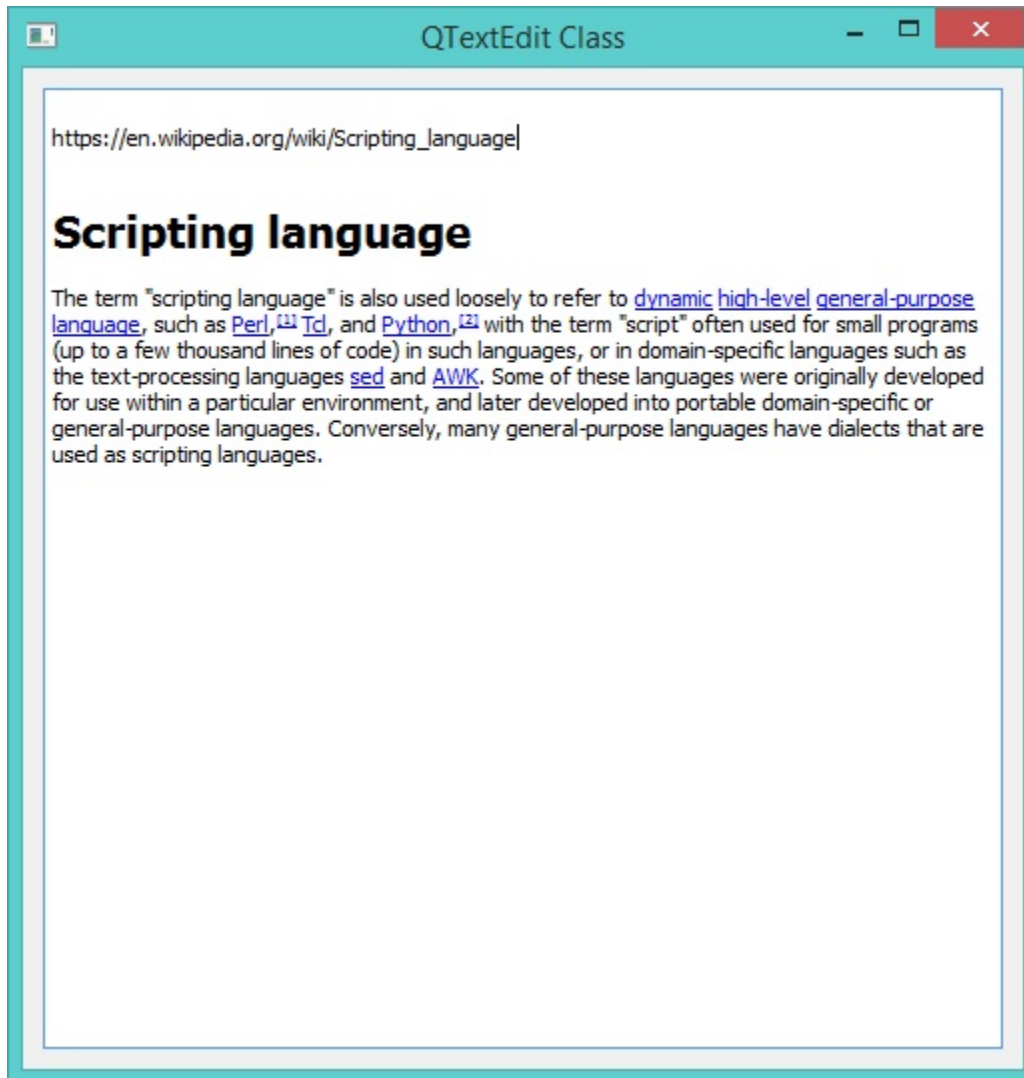
        setWindowTitle("QTextEdit Class")
        setGeometry(100,100,500,500)

        new qtextedit(win1) {
            setGeometry(10,10,480,480)
        }

        show()
    }

    exec()
}
```

During the runtime we can paste rich text in the qtextedit widget



57.4 Using the QListWidget Class

In this example we will use the QListWidget Class

```
Load "guilib.ring"

New qApp {

    win1 = new QWidget() {

        setGeometry(100,100,400,400)

        list1 = new QListWidget(win1) {
            setGeometry(150,100,200,200)
            alist = ["one","two","three","four","five"]
            for x in alist additem(x) next
            setcurrentrow(3,2)
            win1.setwindowtitle("Items Count : " + count() )
        }

    }
```

```

        btn1 = new QPushButton(win1) {
            setGeometry(10,200,100,30)
            setText("selected item")
            setClickedEvent("pWork() ")
        }

        btn2 = new QPushButton(win1) {
            setGeometry(10,240,100,30)
            setText("Delete item")
            setClickedEvent("pWork2() ")
        }

        show()

    }

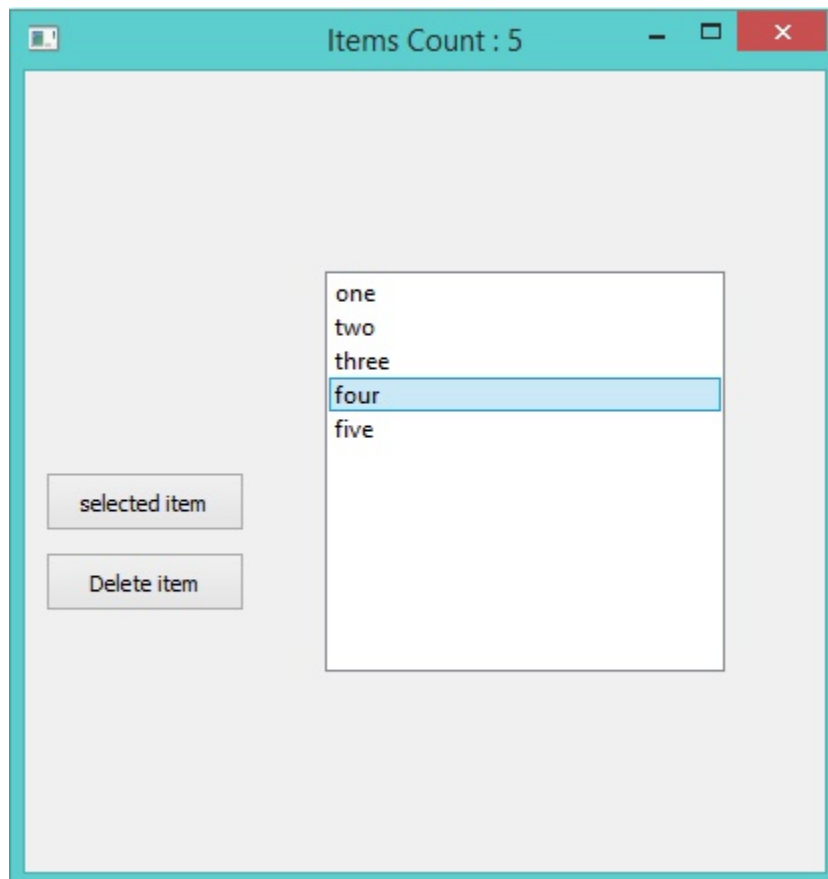
    exec()
}

func pWork
    btn1.setText(string(list1.currentrow()))

func pWork2
    list1 {
        takeitem(currentrow())
    }
}

```

The application during the runtime



Another Example:

```
Load "guilib.ring"

New qApp {

    win1 = new QWidget() {

        setGeometry(100,100,500,400)

        list1 = new QListWidget(win1) {
            setGeometry(150,100,200,200)
            alist = ["one","two","three","four","five"]
            for x in alist additem(x) next

            setcurrentrow(3,2)
            win1.setwindowtitle("Items Count : " + count() )
        }

        btn1 = new QPushButton(win1) {
            setGeometry(10,200,100,30)
            settext("selected item")
            setclickevent("pWork()")
        }

        btn2 = new QPushButton(win1) {
            setGeometry(10,240,100,30)
            settext("Delete item")
            setclickevent("pWork2()")
        }

        show()

    }

    exec()
}

func pWork

    nbrOfItems = list1.count()
    curItemNbr = list1.currentrow()
    curValue   = list1.item(list1.currentrow()).text()

    win1.setwindowtitle( "After Select - NbrOfItems: " + nbrOfItems +
        " CurItemNbr: " + curItemNbr + " CurValue: " + curValue )

    btn1.settext( string(list1.currentrow() ) + " --- " +
        list1.item(list1.currentrow()).text() )

func pWork2
    list1 {
        takeitem(currentrow())

        nbrOfItems = count()
        curItemNbr = currentrow()
        curValue   = item(currentrow()).text()
    }
```

```
win1.setwindowtitle("After Delete - NbrOfItems: " + nbrOfItems +
    " CurItemNbr: " + curItemNbr + " CurValue: " + curValue )
}
```

57.5 Using QTreeView and QFileSystemModel

In this example we will learn how to use the QTreeView widget to represent the File System

```
Load "guilib.ring"

New qApp {

    win1 = New QWidget() {

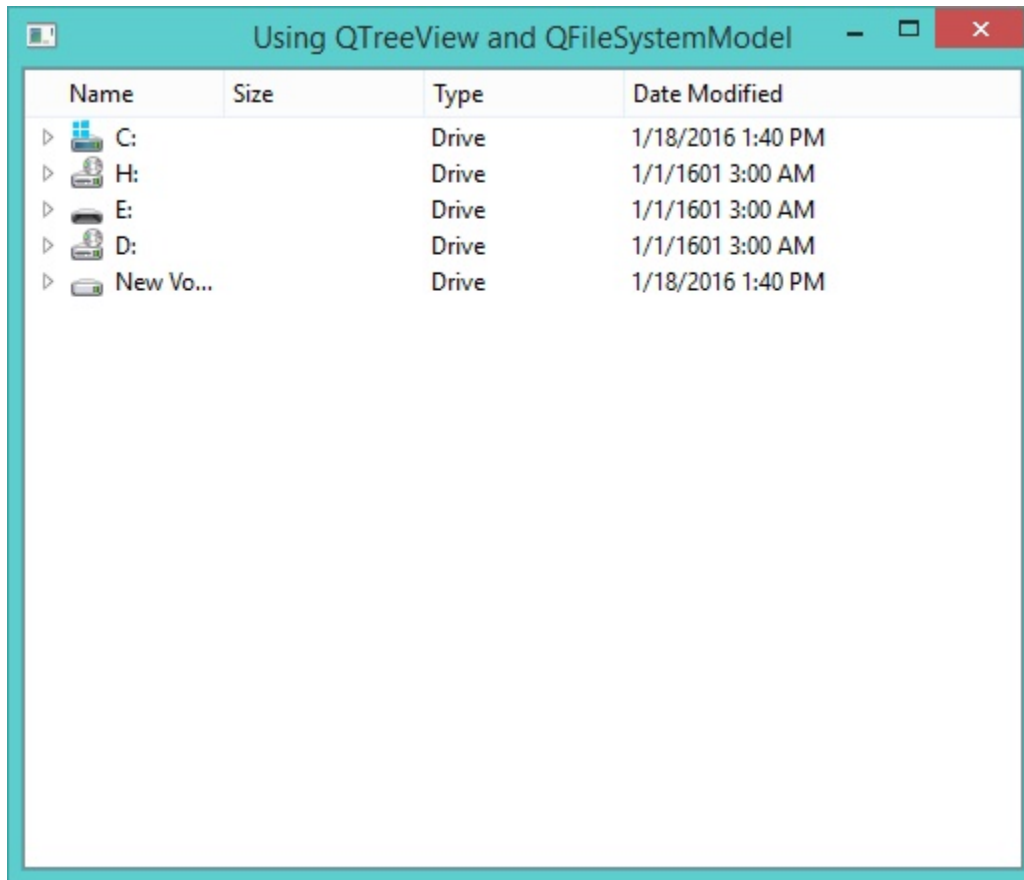
        setwindowtitle("Using QTreeView and QFileSystemModel")
        setGeometry(100,100,500,400)

        New qtreeview(win1) {
            setGeometry(00,00,500,400)
            oDir = new QDir()
            ofile = new QFileSystemModel()
            ofile.setrootpath(oDir.currentpath())
            setmodel(ofile)
        }

        show()
    }

    exec()
}
```

The application during the runtime



57.6 Using QTreeWidget and QTreeWidgetItem

In this example we will learn about using the QTreeWidget and QTreeWidgetItem classes

```
Load "guilib.ring"
```

```
New qApp {
```

```
    win1 = new QWidget() {
```

```
        setWindowTitle("TreeWidget")
```

```
        setGeometry(100,100,400,400)
```

```
        layout1 = new QVBoxLayout()
```

```
        tree1 = new QTreeWidget(win1) {
```

```
            setGeometry(00,00,400,400)
```

```
            setColumnCount(1)
```

```
            myitem = new QTreeWidgetItem()
```

```
            myitem.setText(0,"The First Step")
```

```
            addTopLevelItem(myitem)
```

```
            for x = 1 to 10
```

```
                myitem2 = new QTreeWidgetItem()
```

```
                myitem2.setText(0,"hello"+x)
```

```
                myitem.addChild(myitem2)
```

```

        for y = 1 to 10
            myitem3 = new qtreeswidgetitem()
            myitem3.settext(0,"hello"+x)
            myitem2.addchild(myitem3)

        next
    next
    setheaderlabel("Steps Tree")
}

layout1.addwidget(tree1)
setLayout(layout1)

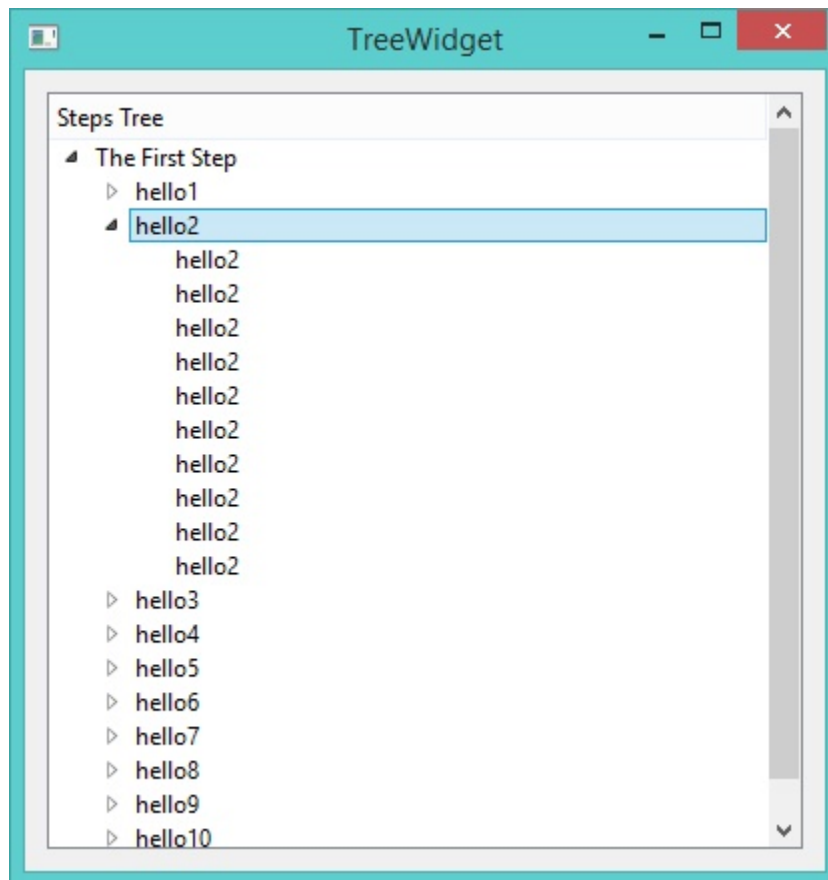
show()

}

exec()
}

```

The application during the runtime



57.7 Using QComboBox Class

In this example we will learn about using the QComboBox class

```
Load "guilib.ring"

New QApplication {

    win1 = new QWidget() {

        setWindowTitle("Using QComboBox")
        setGeometry(100,100,400,400)

        New QComboBox(win1) {
            setGeometry(150,100,200,30)
            alist = ["one","two","three","four","five"]
            for x in alist additem(x,0) next
        }

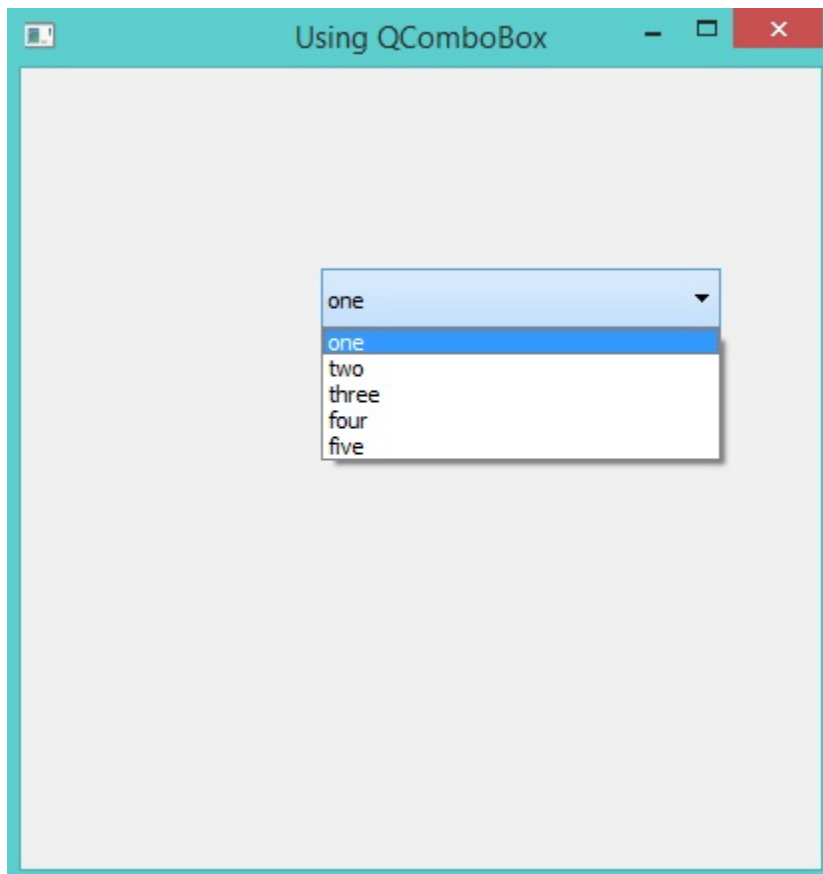
        show()

    }

    exec()

}
```

The application during the runtime



57.8 Creating Menubar

In this example we will learn about using the QMenuBar class


```

Load "guilib.ring"

MyApp = New qApp {

    win1 = new QWidget() {

        setWindowTitle("Using QMenubar")
        setGeometry(100,100,400,400)

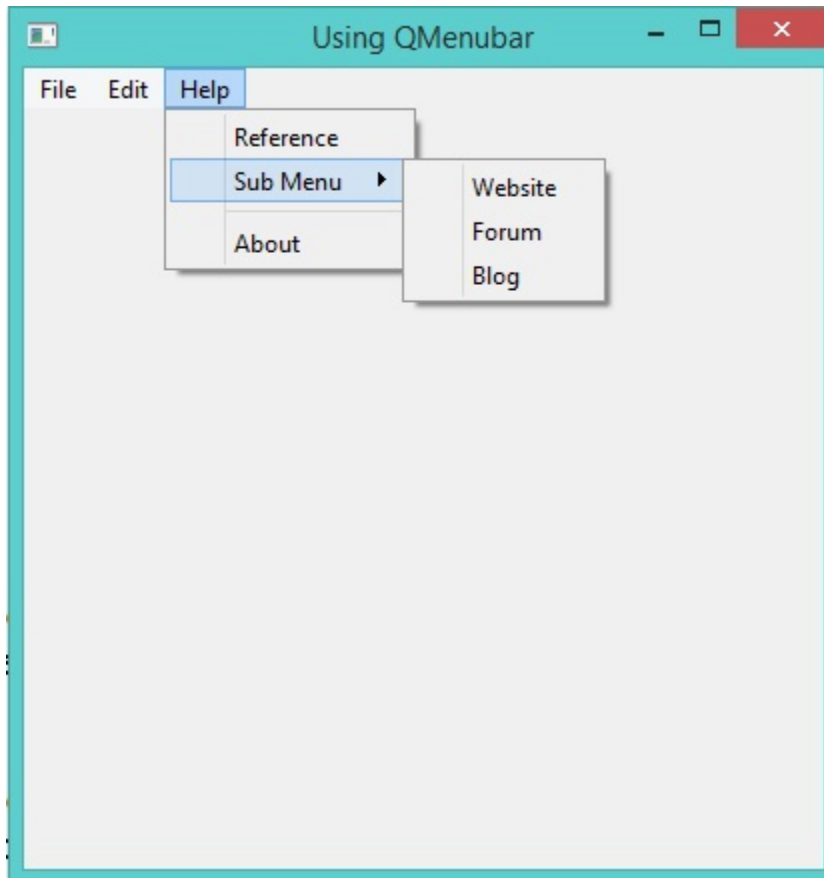
        menu1 = new qmenubar(win1) {
            sub1 = addmenu("File")
            sub2 = addmenu("Edit")
            sub3 = addmenu("Help")
            sub1 {
                oAction = new QAction(win1) {
                    settext("New")
                }
                addaction(oAction)
                oAction = new QAction(win1) {
                    settext("Open")
                }
                addaction(oAction)
                oAction = new QAction(win1) {
                    settext("Save")
                }
                addaction(oAction)
                oAction = new QAction(win1) {
                    settext("Save As")
                }
                addaction(oAction)
                addseparator()
                oAction = new QAction(win1) {
                    settext("Exit")
                    setclideanvent("myapp.quit()")
                }
                addaction(oAction)
            }
            sub2 {
                oAction = new QAction(win1) {
                    settext("Cut")
                }
                addaction(oAction)
                oAction = new QAction(win1) {
                    settext("Copy")
                }
                addaction(oAction)
                oAction = new QAction(win1) {
                    settext("Paste")
                }
                addaction(oAction)
                addseparator()
                oAction = new QAction(win1) {
                    settext("Select All")
                }
                addaction(oAction)
            }
            sub3 {
                oAction = new QAction(win1) {

```

```
        settext("Reference")
    }
    addaction(oAction)
    sub4 = addmenu("Sub Menu")
    sub4 {
        oAction = new QAction(win1) {
            settext("Website")
        }
        addaction(oAction)
        oAction = new QAction(win1) {
            settext("Forum")
        }
        addaction(oAction)
        oAction = new QAction(win1) {
            settext("Blog")
        }
        addaction(oAction)
    }
    addseparator()
    oAction = new QAction(win1) {
        settext("About")
    }
    addaction(oAction)
}

}
show()
}
exec()
}
```

The application during the runtime



57.9 Context Menu

Example:

```
load "guilib.ring"

new QApplication {
    win = new QWidget() {
        setWindowTitle("Context Menu")
        resize(400,400)
        myfilter = new QAllEvents(win) {
            setContextMenuEvent("myMenu()")
        }
        installEventFilter(myfilter)
        show()
    }
    exec()
}

func myMenu

    new QMenu(win) {
        QAction = new QAction(win) {
            setText("new")
            SetClickedEvent("See :New")
        }
    }
}
```

```

    }
    addaction(oAction)
    oAction = new QAction(win) {
        settext("open")
        SetCLickevent("See :Open")
    }
    addaction(oAction)
    oAction = new QAction(win) {
        settext("save")
        SetCLickevent("See :Save")
    }
    addaction(oAction)
    oAction = new QAction(win) {
        settext("close")
        SetCLickevent("See :Close")
    }
    addaction(oAction)
    oCursor = new QCursor()
    exec(oCursor.pos())
}

```

57.10 Creating Toolbar

In this example we will learn about using the `QToolBar` class

```

Load "guilib.ring"

New QApplication {

    win1 = new QMainWindow() {

        setwindowtitle("Using QToolBar")
        setGeometry(100,100,600,400)

        abtns = [
            new QPushButton(win1) { settext("Add") } ,
            new QPushButton(win1) { settext("Edit") } ,
            new QPushButton(win1) { settext("Find") } ,
            new QPushButton(win1) { settext("Delete") } ,
            new QPushButton(win1) { settext("Exit")
                                setclickevent("win1.close()") }
        ]

        tool1 = new QToolBar(win1) {
            for x in abtns addwidget(x) addseparator() next
            setmovable(true)
            setGeometry(0,0,500,30)
            setFloatable(true)
        }

        show()

    }

    exec()
}

```

The application during the runtime



57.11 Creating StatusBar

In this example we will learn about using the `QStatusBar` class

```
Load "guilib.ring"

New qApp {

    win1 = new QMainWindow() {

        setWindowTitle("Using QStatusBar")
        setGeometry(100,100,400,400)

        status1 = new qstatusbar(win1) {
            showMessage("Ready!",0)
        }

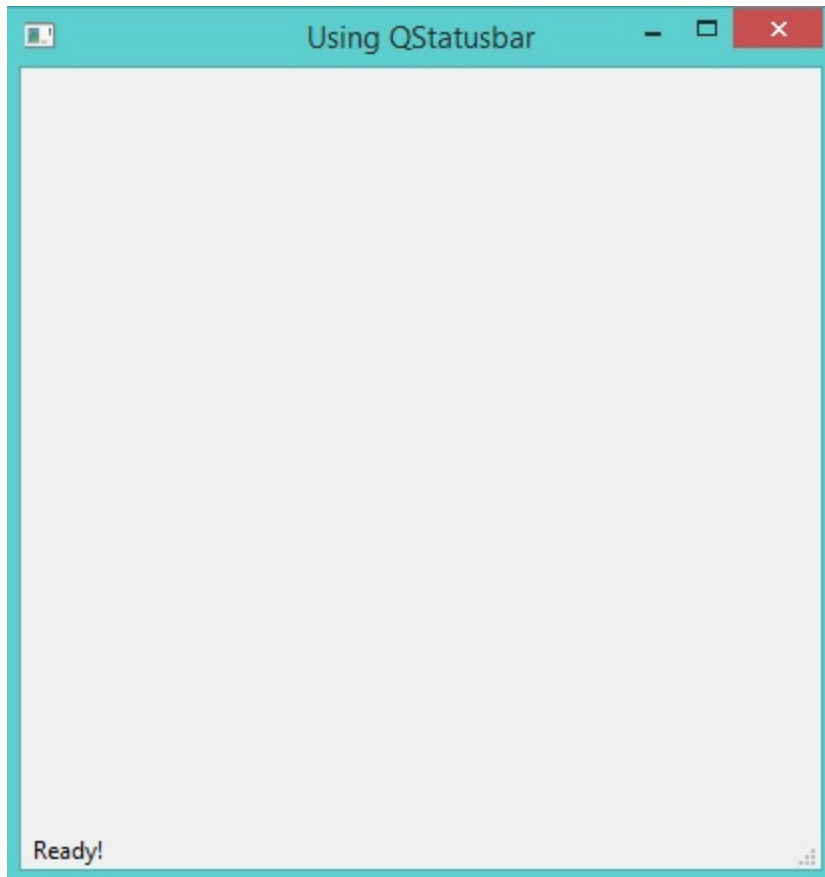
        setstatusbar(status1)
        show()

    }

    exec()

}
```

The application during the runtime



57.12 Using QDockWidget

In this example we will learn about using the QDockWidget class

```
Load "guilib.ring"

New QApplication {

    win1 = new QMainWindow() {

        setWindowTitle("QDockWidget")
        setGeometry(100,100,400,400)

        label1 = new QLabel(win1) {
            setText("Hello")
            setGeometry(300,300,100,100)
        }

        label2 = new QLabel(win1) {
            setText("How are you ?")
            setGeometry(100,100,100,100)
        }

        dock1 = new QDockWidget(win1,0) {
```

```

        setwidget (label1)
        SetAllowedAreas (1)
    }

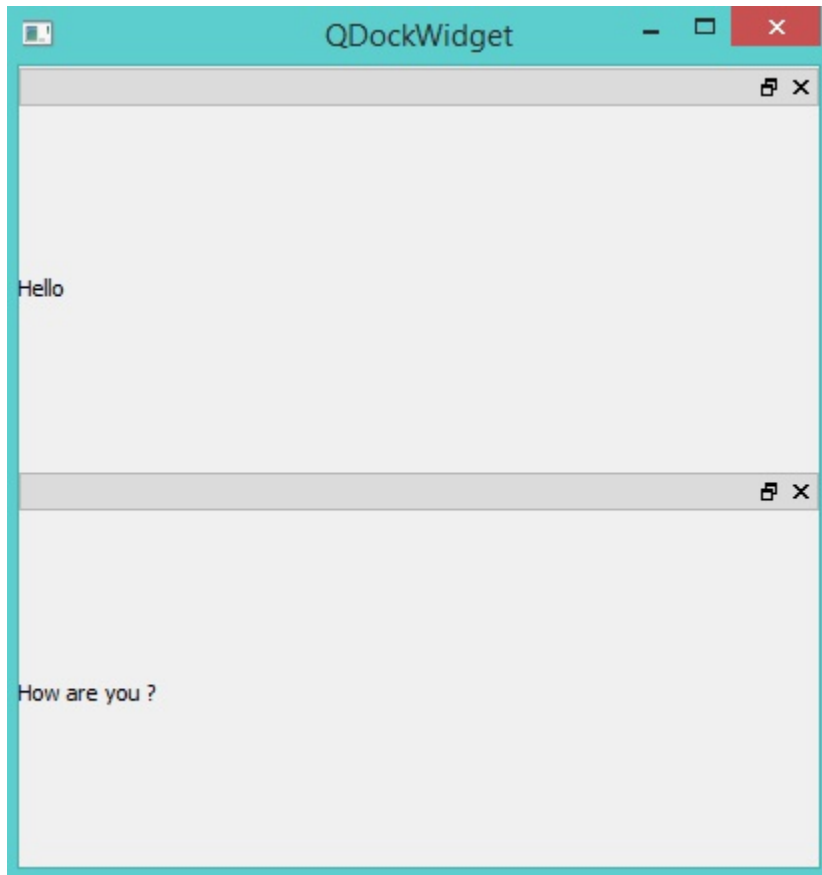
    dock2 = new qdockwidget (win1,0) {
        setwidget (label2)
        SetAllowedAreas (2)
    }

    adddockwidget (Qt_LeftDockWidgetArea,dock1,Qt_Horizontal)
    adddockwidget (Qt_LeftDockWidgetArea,dock2,Qt_Vertical)

    show ()
}
exec ()
}

```

The application during the runtime



57.13 Using QTabWidget

In this example we will learn about using the QTabWidget class

```

Load "guilib.ring"

New qApp {

```

```
win1 = new QMainWindow() {

    setWindowTitle("Using QTabWidget")
    setGeometry(100,100,400,400)

    page1 = new QWidget() {
        new QPushButton(page1) {
            setText("The First Page")
        }
    }

    page2 = new QWidget() {
        new QPushButton(page2) {
            setText("The Second Page")
        }
    }

    page3 = new QWidget() {
        new QPushButton(page3) {
            setText("The Third Page")
        }
    }

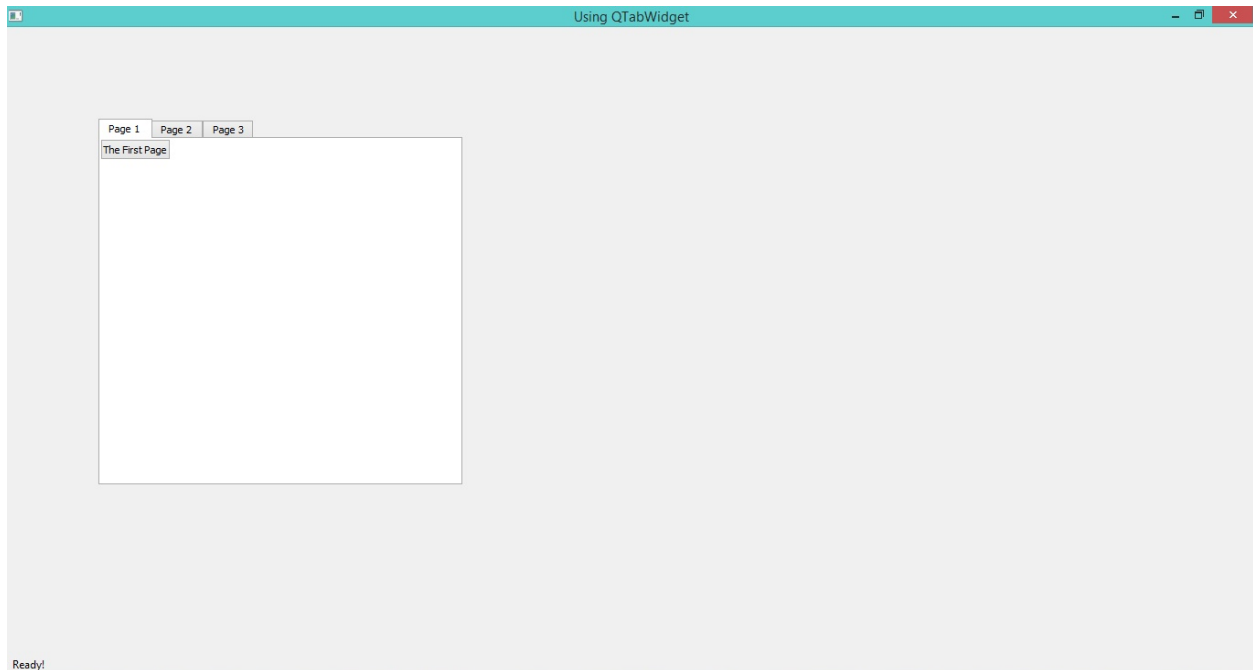
    tab1 = new QTabWidget(win1) {
        insertTab(0,page1,"Page 1")
        insertTab(1,page2,"Page 2")
        insertTab(2,page3,"Page 3")
        setGeometry(100,100,400,400)
    }

    status1 = new QStatusBar(win1) {
        showMessage("Ready!",0)
    }

    setStatusBar(status1)
    showMaximized()
}

exec()
```

The application during the runtime



57.14 Using QTableWidgetItem

In this example we will learn about using the QTableWidgetItem class

```
Load "guilib.ring"

New QApplication {

    win1 = new QMainWindow() {

        setGeometry(100,100,1100,370)
        setWindowTitle("Using QTableWidgetItem")

        Table1 = new QTableWidgetItem(win1) {

            setRowCount(10) setColumnCount(10)
            setGeometry(0,0,800,400)
            setSelectionBehavior(QAbstractItemView_SelectRows)

            for x = 1 to 10
                for y = 1 to 10
                    item1 = new QTableWidgetItem("R"+X+"C"+Y)
                    setItem(x-1,y-1,item1)
                next
            next

        }

        setCentralWidget(table1)
        show()

    }

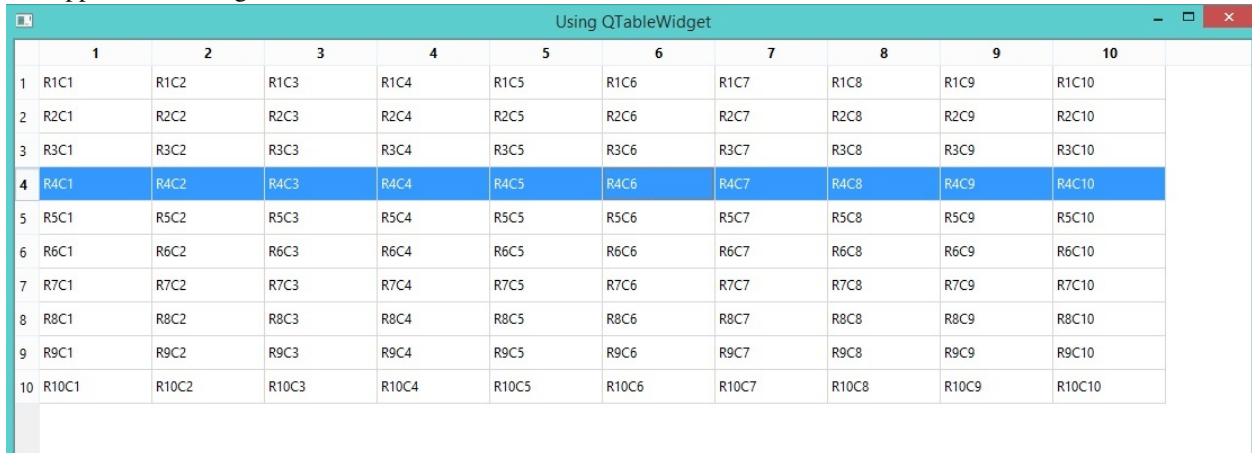
}
```

```

    exec()
}

```

The application during the runtime



	1	2	3	4	5	6	7	8	9	10
1	R1C1	R1C2	R1C3	R1C4	R1C5	R1C6	R1C7	R1C8	R1C9	R1C10
2	R2C1	R2C2	R2C3	R2C4	R2C5	R2C6	R2C7	R2C8	R2C9	R2C10
3	R3C1	R3C2	R3C3	R3C4	R3C5	R3C6	R3C7	R3C8	R3C9	R3C10
4	R4C1	R4C2	R4C3	R4C4	R4C5	R4C6	R4C7	R4C8	R4C9	R4C10
5	R5C1	R5C2	R5C3	R5C4	R5C5	R5C6	R5C7	R5C8	R5C9	R5C10
6	R6C1	R6C2	R6C3	R6C4	R6C5	R6C6	R6C7	R6C8	R6C9	R6C10
7	R7C1	R7C2	R7C3	R7C4	R7C5	R7C6	R7C7	R7C8	R7C9	R7C10
8	R8C1	R8C2	R8C3	R8C4	R8C5	R8C6	R8C7	R8C8	R8C9	R8C10
9	R9C1	R9C2	R9C3	R9C4	R9C5	R9C6	R9C7	R9C8	R9C9	R9C10
10	R10C1	R10C2	R10C3	R10C4	R10C5	R10C6	R10C7	R10C8	R10C9	R10C10

57.15 Using QProgressBar

In this example we will learn about using the QProgressBar class

```

Load "guilib.ring"

New qApp {
    win1 = new QMainWindow() {

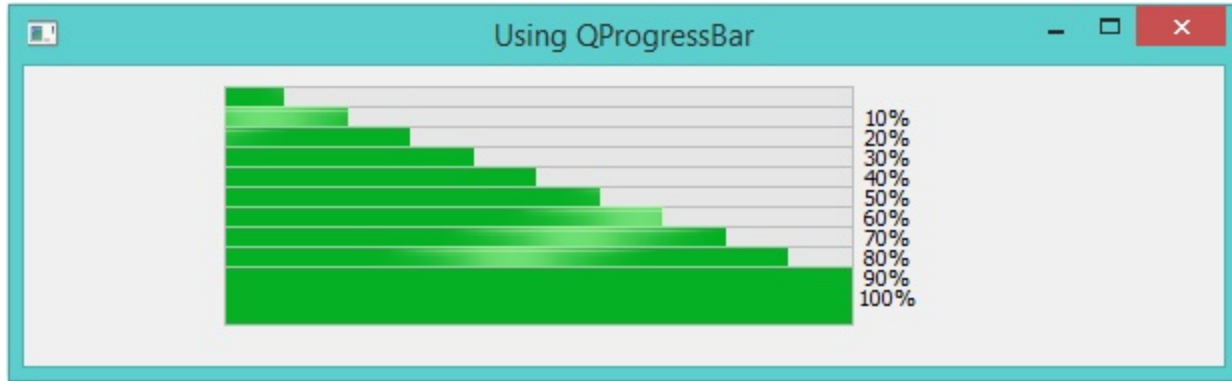
        setGeometry(100,100,600,150)
        setWindowTitle("Using QProgressBar")

        for x = 10 to 100 step 10
            new qprogressbar(win1) {
                setGeometry(100,x,350,30)
                setvalue(x)
            }
        next

        show()
    }
    exec()
}

```

The application during the runtime



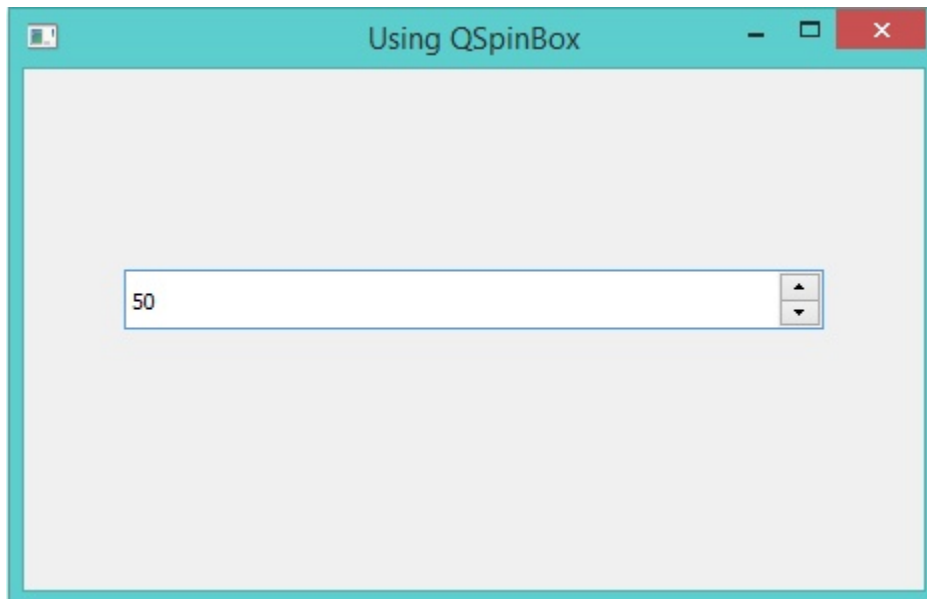
57.16 Using QSpinBox

In this example we will learn about using the QSpinBox class

```
Load "guilib.ring"

New qApp {
    win1 = new QMainWindow() {
        setGeometry(100,100,450,260)
        setWindowTitle("Using QSpinBox")
        new qspinbox(win1) {
            setGeometry(50,100,350,30)
            setvalue(50)
        }
        show()
    }
    exec()
}
```

The application during the runtime



57.17 Using QSlider

In this example we will learn about using the QSlider class

```
Load "guilib.ring"

New QApplication {

    win1 = new QMainWindow() {

        setGeometry(100,100,500,400)
        setWindowTitle("Using QSlider")

        new QSlider(win1) {
            setGeometry(100,100,50,130)
            setTickInterval(50)
        }

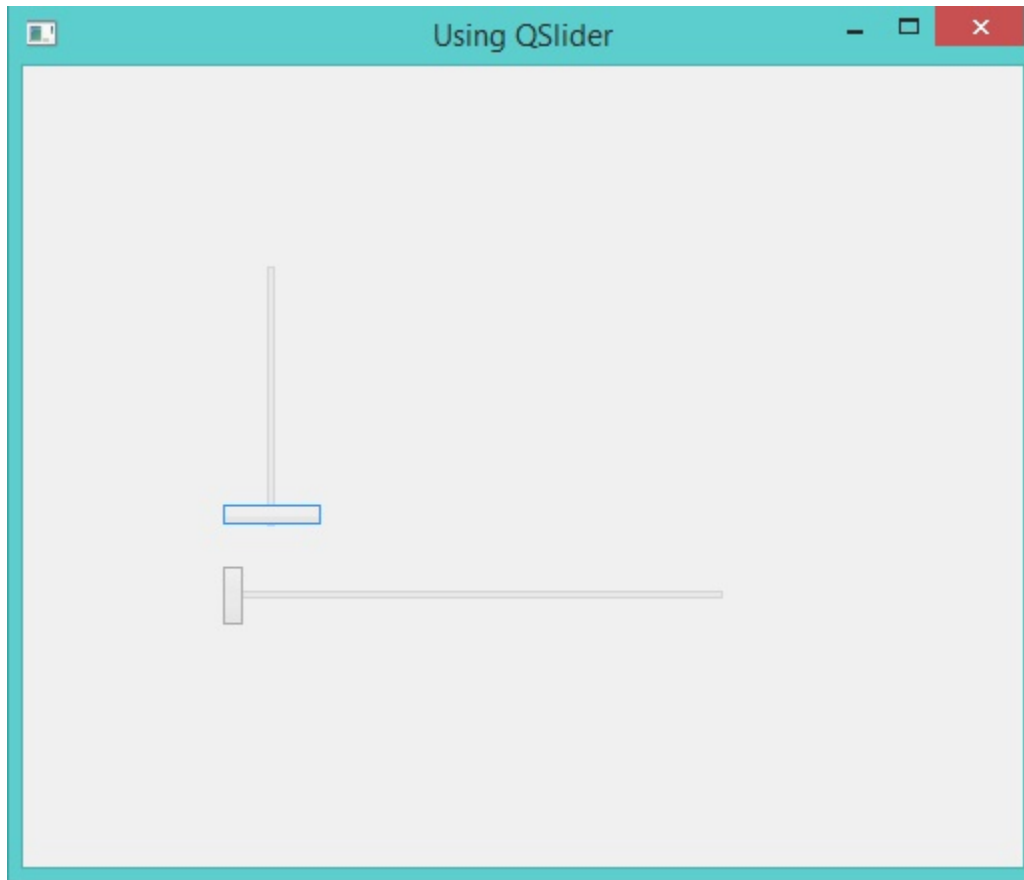
        new QSlider(win1) {
            setGeometry(100,250,250,30)
            setTickInterval(50)
            setOrientation(Qt_Horizontal)
        }

        show()

    }

    exec()
}
```

The application during the runtime



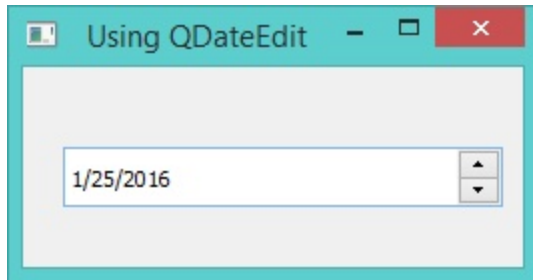
57.18 Using QDateEdit

In this example we will learn about using the QDateEdit class

```
Load "guilib.ring"

New QApplication {
    win1 = new QMainWindow() {
        setWindowTitle("Using QDateEdit")
        setGeometry(100,100,250,100)
        new QDateEdit(win1) {
            setGeometry(20,40,220,30)
        }
        show()
    }
    exec()
}
```

The application during the runtime



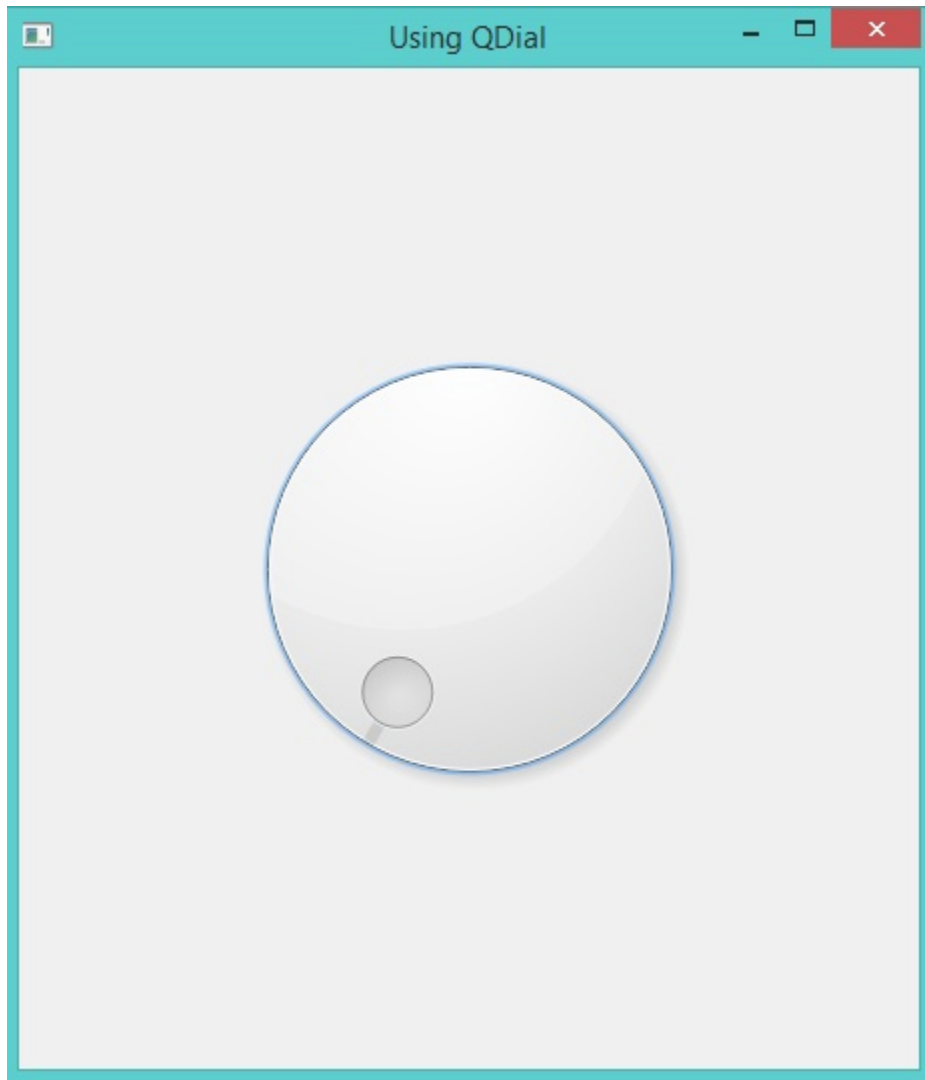
57.19 Using QDial

In this example we will learn about using the QDial class

```
Load "guilib.ring"

New qApp {
    win1 = new QMainWindow() {
        setGeometry(100,100,450,500)
        setWindowTitle("Using QDial")
        new qdial(win1) {
            setGeometry(100,100,250,300)
        }
        show()
    }
    exec()
}
```

The application during the runtime



Another Example

```
Load "guilib.ring"

New qApp {
    win1 = new QMainWindow()
    {
        setGeometry(100,100,450,500)
        setWindowTitle("Using QDial")
        button1 = new QPushButton(win1){
            setGeometry(100,350,100,30)
            setText("Increment")
            setClickEvent("pIncrement()")
        }

        button2 = new QPushButton(win1){
            setGeometry(250,350,100,30)
            setText("Decrement")
            setClickEvent("pDecrement()")
        }
        pdial = new qdial(win1) {
```

```

        setGeometry(100,50,250,300)
        setNotchesVisible(true)
        setValue(50)
        SetValueChangedEvent("pDialMove()")
    }
    linedit1 = new QLineEdit(win1) {
        setGeometry(200,400,50,30)
        setAlignment(Qt_AlignHCenter)
        setText(string(pdial.value()))
        setReturnPressedEvent("pPress()")
    }
    show()
}

exec()
}

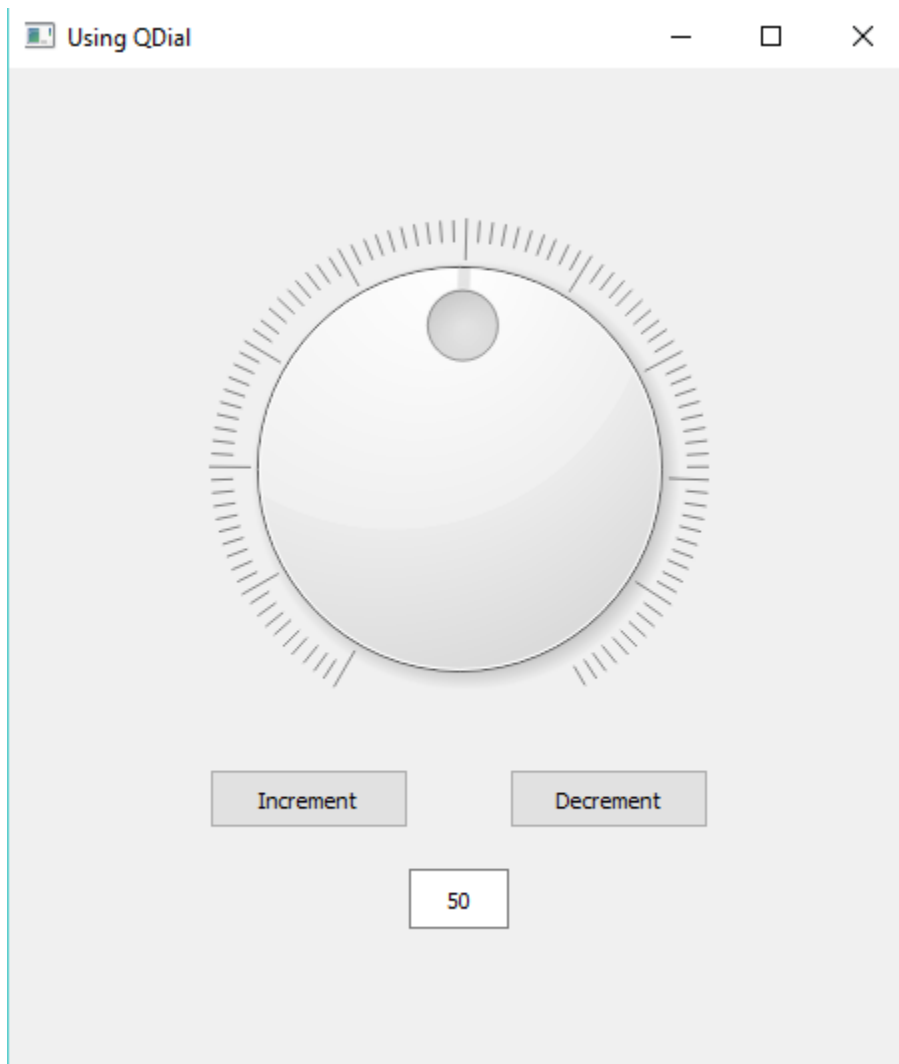
func pIncrement
    pdial{val=value()}
    pdial.setvalue(val+1)
    linedit1{setText(string(val+1))}

func pDecrement
    pdial{val=value()}
    pdial.setvalue(val-1)
    linedit1{setText(string(val-1))}

func pPress
    linedit1{val=text()}
    pdial.setvalue(number(val))

func pDialMove
    linedit1.setText(""+pdial.value())

```

57.20 Using QWebView

In this example we will learn about using the QWebView class

```
Load "guilib.ring"
```

```
New qApp {  
    win1 = new QMainWindow() {  
        setWindowTitle("QWebView")  
        myweb = new QWebView(win1) {  
            setGeometry(10,10,600,600)  
            loadPage(new QUrl("http://google.com"))  
        }  
        setCentralWidget(myweb)  
        showMaximized()  
    }  
    exec()  
}
```

The application during the runtime



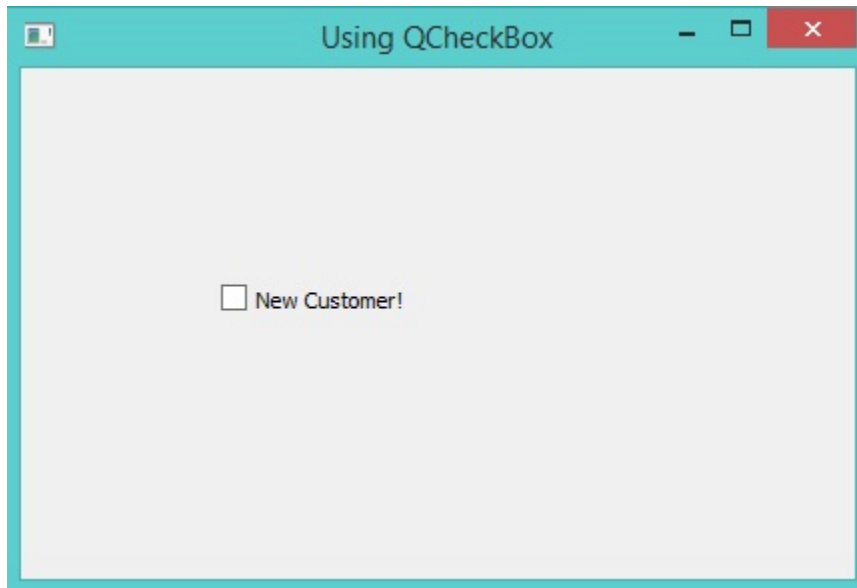
57.21 Using QCheckBox

In this example we will learn about using the QCheckBox class

```
Load "guilib.ring"

New qApp {
    win1 = new QMainWindow() {
        setWindowTitle("Using QCheckBox")
        new qcheckbox(win1) {
            setGeometry(100,100,100,30)
            settext("New Customer!")
        }
        showMaximized()
    }
    exec()
}
```

The application during the runtime



Another Example:

```
Load "guilib.ring"

New qApp {
    win1 = new QMainWindow() {
        setGeometry(100,100,400,300)
        setWindowTitle("Using QCheckBox")

        ### 0-Unchecked 1-Checked

        CheckBox = new qcheckbox(win1) {
            setGeometry(100,100,160,30)
            setText("New Customer!")
            setClickedEvent("HandleClickEvent()")
        }

        show()
    }
    exec()
}

Func HandleClickEvent

    if CheckBox.isChecked() = 1
        CheckBox.setText("New Customer. Check 1-ON")
    else
        CheckBox.setText("New Customer. Check 0-OFF")
    ok
```

57.22 Using QRadioButton and QButtonGroup

In this example we will learn about using the QRadioButton and QButtonGroup classes

```
Load "guilib.ring"
```

```
New QApplication {  
    win1 = new QMainWindow() {  
        setWindowTitle("Using QRadioButton")  
  
        new qradiobutton(win1) {  
            setGeometry(100,100,100,30)  
            settext("One")  
        }  
        new qradiobutton(win1) {  
            setGeometry(100,150,100,30)  
            settext("Two")  
        }  
        new qradiobutton(win1) {  
            setGeometry(100,200,100,30)  
            settext("Three")  
        }  
  
        group2 = new qbuttongroup(win1) {  
            btn4 = new qradiobutton(win1) {  
                setGeometry(200,150,100,30)  
                settext("Four")  
            }  
            btn5 = new qradiobutton(win1) {  
                setGeometry(200,200,100,30)  
                settext("Five")  
            }  
            addbutton(btn4,0)  
            addbutton(btn5,0)  
        }  
  
        showMaximized()  
    }  
    exec()  
}
```

The application during the runtime



57.23 Adding Hyperlink to QLabel

In this example we will learn about creating Hyperlink using the QLabel class

```
Load "guilib.ring"

New qApp {
    win1 = new QMainWindow() {
        setWindowTitle("QLabel - Hyperlink")
        new QLabel(win1) {
            setGeometry(100,100,100,30)
            setOpenExternallinks(true)
            setText('<a href="http://google.com">Google</a>')
        }
        showMaximized()
    }
    exec()
}
```

The application during the runtime



57.24 QVideoWidget and QMediaPlayer

In this example we will learn about using the `QVideoWidget` and `QMediaPlayer` classes to play a group of movies from different positions at the same time

```
Load "guilib.ring"

New QApplication {

    win1 = new QMainWindow() {

        setWindowTitle("QVideoWidget")

        btn1 = new QPushButton(win1) {
            setGeometry(0,0,100,30)
            setText("play")
            setClickedEvent("player.play() player2.play()
                           player3.play() player4.play()")
        }

        videowidget = new QVideoWidget(win1) {
            setGeometry(50,50,600,300)
            setStyleSheet("background-color: black")
        }

        videowidget2 = new QVideoWidget(win1) {
            setGeometry(700,50,600,300)
            setStyleSheet("background-color: black")
        }

        videowidget3 = new QVideoWidget(win1) {
            setGeometry(50,370,600,300)
            setStyleSheet("background-color: black")
        }

        videowidget4 = new QVideoWidget(win1) {
```

```

        setGeometry(700,370,600,300)
        setstylesheet("background-color: black")
    }

    player = new QMediaPlayer() {
        setmedia(new QUrl("1.mp4"))
        setvideowidget(videowidget)
        setposition(35*60*1000)
    }

    player2 = new QMediaPlayer() {
        setmedia(new QUrl("2.mp4"))
        setvideowidget(videowidget2)
        setposition(23*60*1000)
    }

    player3 = new QMediaPlayer() {
        setmedia(new QUrl("3.mp4"))
        setvideowidget(videowidget3)
        setposition(14.22*60*1000)
    }

    player4 = new QMediaPlayer() {
        setmedia(new QUrl("4.avi"))
        setvideowidget(videowidget4)
        setposition(8*60*1000)
    }

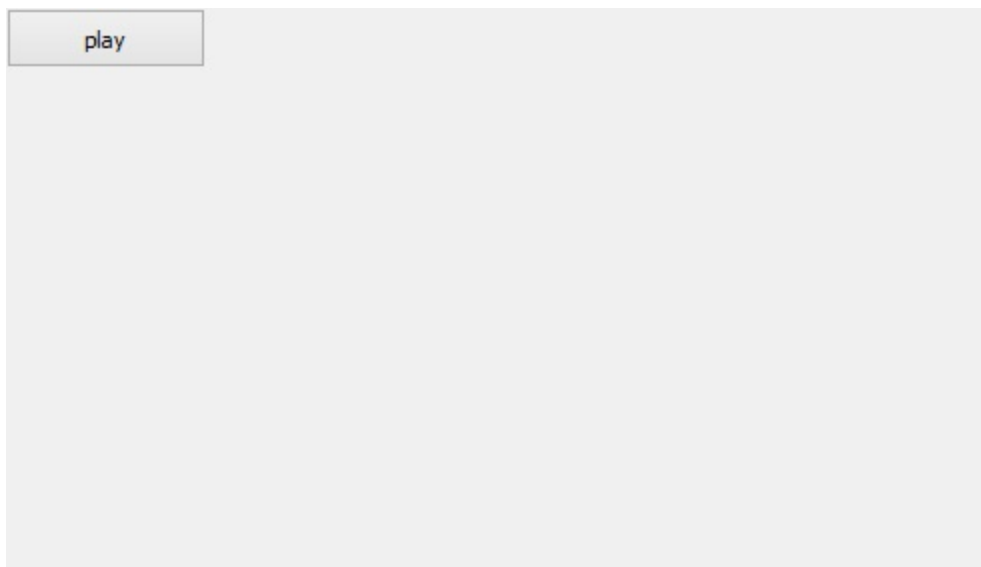
    showfullscreen()

    }

    exec()
}

```

The application during the runtime



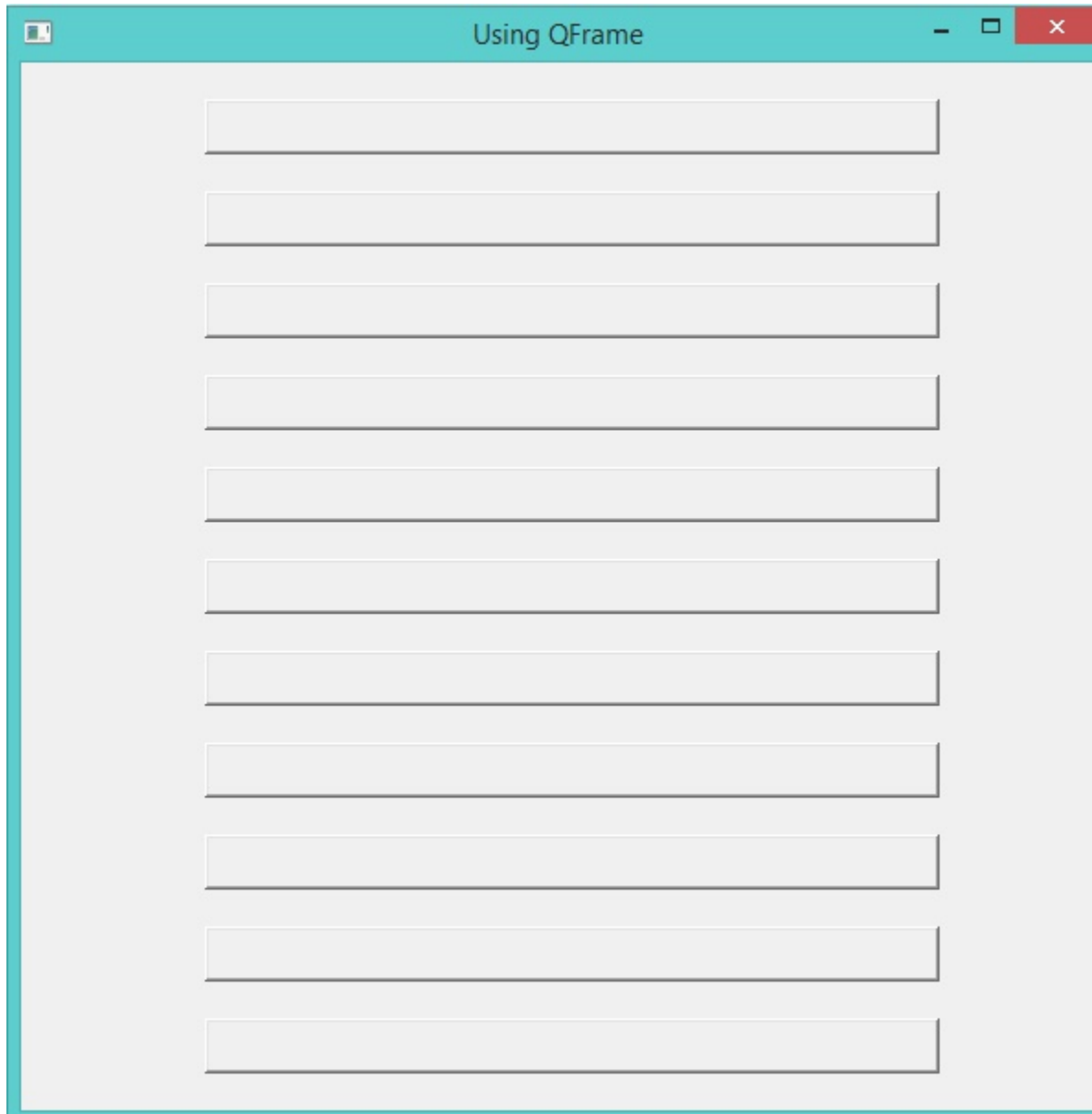
57.25 Using QFrame

In this example we will learn about using the QFrame class

```
Load "guilib.ring"

New qApp {
    win1 = new QMainWindow() {
        setWindowTitle("Using QFrame")
        for x = 0 to 10
            frame1 = new qframe(win1,0) {
                setGeometry(100,20+50*x,400,30)
                setFrameStyle(QFrame_Raised | QFrame_WinPanel)
            }
        next
        showMaximized()
    }
    exec()
}
```

The application during the runtime



57.26 Display Image using QLabel

In this example we will learn about displaying an image using the QLabel widget

```
Load "guilib.ring"

New QApplication {
    win1 = new QMainWindow() {
        setWindowTitle("QLabel - Display image")
        new QLabel(win1) {
            image = new QPixmap("b:/mahmoud/photo/advice.jpg")
            setPixmap(image)
            setGeometry(0,0,image.width(),image.height())
        }
    }
}
```

```

        showMaximized()
    }
    exec()
}

```

The application during the runtime



57.27 Menubar and StyleSheet Example

In this example we will learn about creating menubar and setting the window stylesheet

```
Load "guilib.ring"
```

```
New qApp {
```

```

win1 = new QMainWindow() {

    setwindowtitle("Menubar")

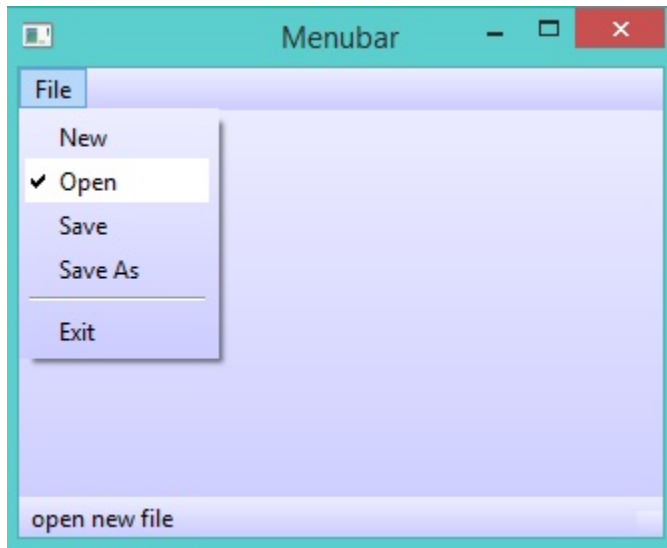
    menu1 = new qmenubar(win1) {
        sub1 = addmenu("File")
        sub1 {
            oAction = new QAction(win1) {
                settext("New")
                setenabled(false)
            }
            addaction(oAction)
            oAction = new QAction(win1) {
                settext("Open")
                setcheckable(true)
                setchecked(true)
                setstatustip("open new file")
            }
            addaction(oAction)
            oAction = new QAction(win1) {
                settext("Save")
            }
            addaction(oAction)
            oAction = new QAction(win1) {
                settext("Save As")
            }
            addaction(oAction)

            addseparator()
            oAction = new QAction(win1)
            oAction.settext("Exit")
            oAction.setclickevent("myapp.quit()")
            addaction(oAction)
        }
    }

    status1 = new qstatusbar(win1) {
        showmessage("Ready!", 0)
    }
    setmenubar(menu1)
    setmousetracking(true)
    setstatusbar(status1)
    setStyleSheet("color: black; selection-color: black;
selection-background-color:white ;
background: QLinearGradient(x1: 0, y1: 0, x2: 0, y2: 1,
stop: 0 #eef, stop: 1 #ccf);")
    showmaximized()
}
exec()
}

```

The application during the runtime



57.28 QLineEdit Events and QMessageBox

In this example we will learn about using QLineEdit Events and displaying a MessageBox

Load "guilib.ring"

```
MyApp = New qApp {

    win1 = new QWidget() {

        setWindowTitle("Welcome")
        setGeometry(100,100,400,300)

        label1 = new QLabel(win1) {
            settext("What is your name ?")
            setGeometry(10,20,350,30)
            setalignment(Qt_AlignHCenter)
        }

        btn1 = new QPushButton(win1) {
            setGeometry(10,200,100,30)
            settext("Say Hello")
            setclickevent("pHello()")
        }

        btn1 = new QPushButton(win1) {
            setGeometry(150,200,100,30)
            settext("Close")
            setclickevent("pClose()")
        }

        lineedit1 = new QLineEdit(win1) {
            setGeometry(10,100,350,30)
            settextchangedevent("pChange()")
            setreturnpressedevent("penter()")
        }
    }
}
```

```

        show()
    }

    exec()
}

Func pHello
    lineedit1.setText( "Hello " + lineedit1.text() )

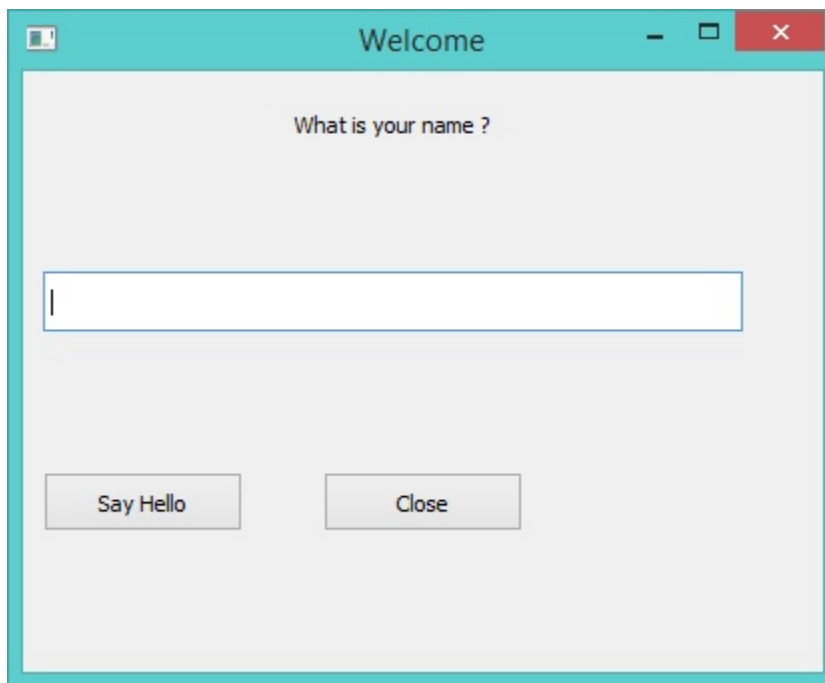
Func pClose
    MyApp.quit()

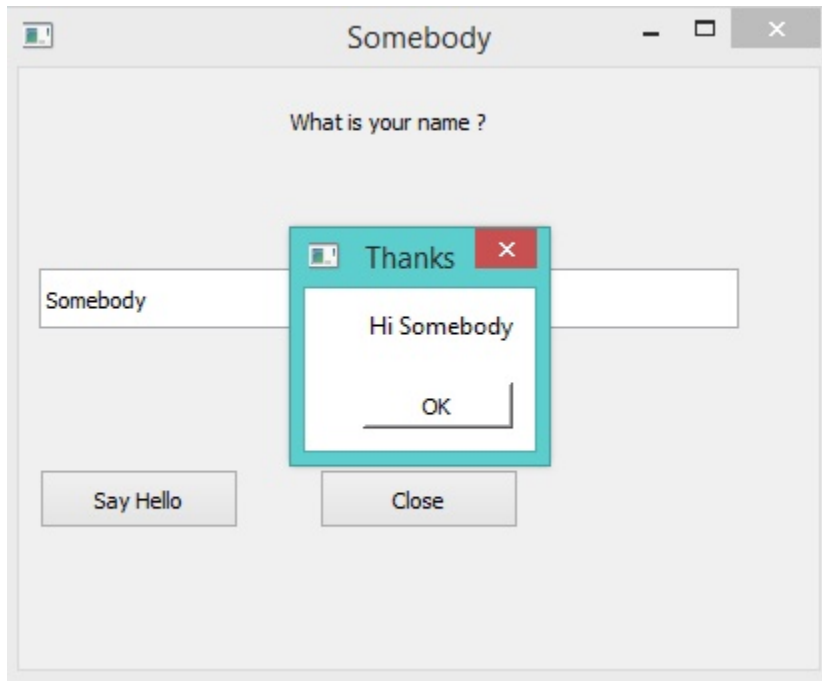
Func pChange
    win1 { setwindowtitle( lineedit1.text() ) }

Func pEnter
    new QMessageBox(win1) {
        setwindowtitle("Thanks")
        setText("Hi " + lineedit1.text() )
        setstylesheet("background-color : white")
        show()
    }

```

The application during the runtime





57.29 Other Widgets Events

Each Qt signal can be used in RingQt, just add Set before the signal name and add event after the signal name to get the method that can be used to determine the event code.

For example the `QProgressBar` class contains a signal named `valueChanged()` To use it just use the function `setValueChangedEvent()`

Example:

```
Load "guilib.ring"

New qApp {
    win1 = new QMainWindow() {

        setWindowTitle("QProgressBar valueChanged Event")

        progress1 = new qprogressbar(win1) {
            setGeometry(100,100,350,30)
            setValue(10)
            setValueChangedEvent("pChange() ")
        }

        new QPushButton(win1) {
            setGeometry(10,10,100,30)
            setText("increase")
            setClickedEvent("pIncrease() ")
        }

        showMaximized()
    }
}
```

```

        exec()
    }

    func pIncrease
        progress1 { setvalue(value()+1) }

    func pchange
        win1.setwindowtitle("value : " + progress1.value() )

```

The application during the runtime



Another example for the stateChanged event of the QCheckBox class

```

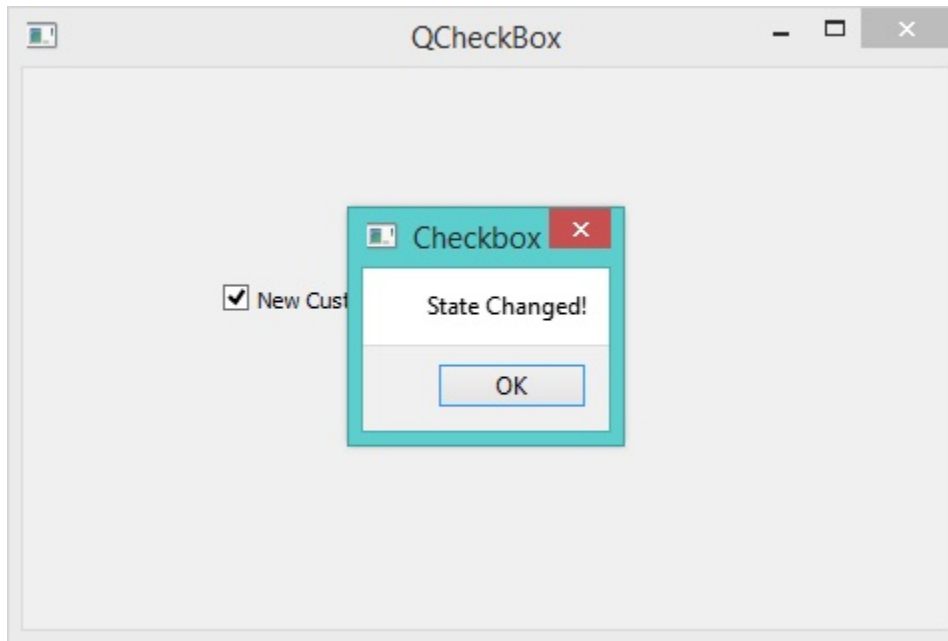
Load "guilib.ring"

New qApp {
    win1 = new QMainWindow() {
        setwindowtitle("QCheckBox")
        new qcheckbox(win1) {
            setGeometry(100,100,100,30)
            settext("New Customer!")
            setstatechangedevent("pchange() ")
        }
        showMaximized()
    }
    exec()
}

Func pChange
    new QMessageBox(Win1) {
        setWindowTitle("Checkbox")
        settext("State Changed!")
        show()
    }

```

The application during the runtime



57.30 Using the QTimer Class

In this example we will learn about using the QTimer class

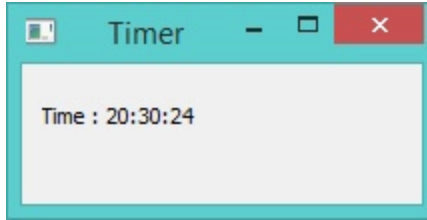
```
Load "guilib.ring"

new QApplication {
    win1 = new QWidget() {
        setGeometry(100,100,200,70)
        setWindowTitle("Timer")
        label1 = new QLabel(win1) {
            setGeometry(10,10,200,30)
            setText(thetime())
        }
        new QTimer(win1) {
            setInterval(1000)
            setTimeoutEvent("pTime()")
            start()
        }
        show()
    }
    exec()
}

func ptime
    label1.setText(thetime())

Func thetime
    return "Time : " + Time()
```

The application during the runtime



57.31 Using QProgressBar and Timer

In this example we will learn about using the “animated” QProgressBar class and Timer

```
###-----
### ProgressBar and Timer Example

Load "guilib.ring"

new qApp
{
    win1 = new QWidget()
    {
        setGeometry(100,100,400,100)
        setWindowTitle("Timer and ProgressBar")

        LabelMan = new QLabel(win1)
        {
            setGeometry(10,10,200,30)
            setText(theTime())      ### ==>> func
        }

        TimerMan = new QTimer(win1)
        {
            setInterval(1000)
            setTimeoutEvent("pTime()")  ### ==>> func
            start()
        }

        BarMan = new QProgressBar(win1)
        {
            setGeometry(100,50,300,10)  ### Position X y, Length, Thickness
            setValue(0)                  ### Percent filled
        }

        show()
    }
    exec()
}

func pTime
    LabelMan.setText(theTime())      ### ==>> func

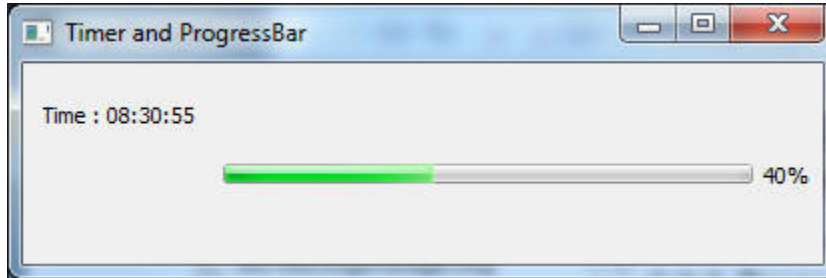
    Increment = 10
    if BarMan.value() >= 100          ### ProgressBar start over.
        BarMan.setValue(0)

    ok
    BarMan{ setValue(value() + Increment) }
```

```

Func theTime
    return "Time : " + Time()

```



57.32 Display Scaled Image using QLabel

In this example we will learn about displaying and scaling an image so that it looks “animated” using the QLabel widget

```

Load "guilib.ring"

#-----
# REQUIRES: image = "C:\RING\bin\stock.jpg"
# imageStock: start dimensions for growing image

imageW = 200 ; imageH = 200 ; GrowBy = 4

###-----
### Window and Box Size dimensions

WinWidth = 1280 ; WinHeight = 960
BoxWidth = WinWidth -80 ; BoxHeight = WinHeight -80

###-----

New qapp {

    win1 = new QWidget() {

        setGeometry(50,50, WinWidth,WinHeight)
        setWindowTitle("Animated Image - Display Image Scaled and Resized")

        imageStock = new QLabel(win1) {

            image = new QPixmap("C:\RING\bin\stock.jpg")
            AspectRatio = image.width() / image.height()

            imageW = 200
            imageH = imageH / AspectRatio

            ### Size-H, Size-V, Aspect, Transform
            setPixmap(image.scaled(imageW , imageH ,0,0))

            PosLeft = (BoxWidth - imageW ) / 2
            PosTop  = (BoxHeight - imageH ) / 2
        }
    }
}

```

```

        setGeometry(PosLeft,PosTop,imageW,imageH)

    }

    TimerMan = new QTimer(win1) {
        setInterval(100)          ### interval 100 millisecs.
        setTimeoutEvent("pTime()")  ### ==> func
        start()

    }

    show()

}

exec()
}

###-----
### Fuction TimerMan: calling interval 100 milliseconds

func pTime

    ### Stop Timer when image is size of Window area
    if imageW > BoxWidth
        TimerMan.stop()
        imageStock.clear()      ### Will clear the image

    ok

    ### Grow image
    imageW += GrowBy
    imageH = imageW / AspectRatio

    ### Scaled Image: Size-H, Size-V, Aspect, Transform
    imageStock.setPixmap(image.scaled(imageW , imageH ,0,0))

    ### Center the image
    PosLeft = (WinWidth - imageW ) / 2
    PosTop  = (WinHeight - imageH ) / 2
    imageStock.setGeometry(PosLeft,PosTop,imageW,imageH)

```

57.33 Using the QFileDialog Class

Example

```

Load "guilib.ring"

New qapp {
    win1 = new QWidget() {
        setWindowTitle("open file")
        setGeometry(100,100,400,400)
        new QPushButton(win1) {
            setGeometry(10,10,200,30)
            setText("open file")
            setClickedEvent("pOpen() ")
        }
        show()
    }
}

```

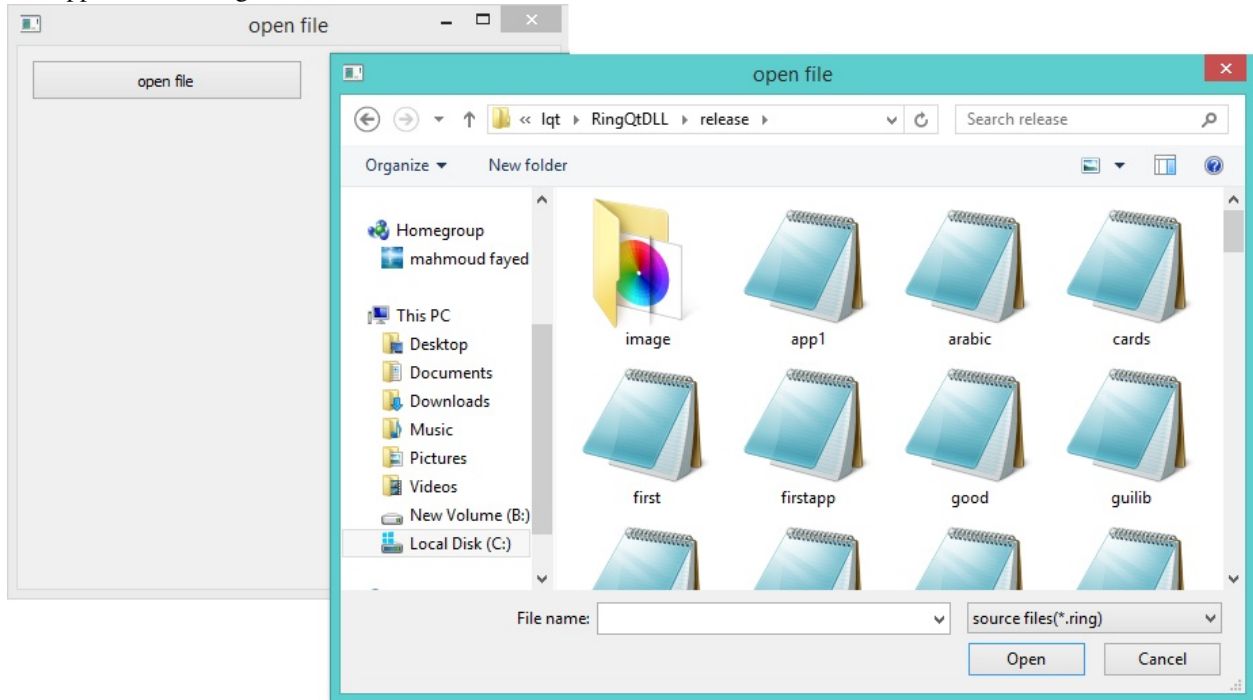
```

    exec()
}

Func pOpen
    new QFileDialog(win1) {
        cName = getOpenFileName(win1,"open file","c:\\","source files(*.ring)")
        win1.setWindowTitle(cName)
    }
}

```

The application during the runtime



57.34 Drawing using QPainter

In this example we will learn about drawing using the QPainter class

```

Load "guilib.ring"
New qapp {
    win1 = new QWidget() {
        setWindowTitle("Drawing using QPainter")
        setGeometry(100,100,500,500)
        label1 = new QLabel(win1) {
            setGeometry(10,10,400,400)
            setText("")
        }
        new QPushButton(win1) {
            setGeometry(200,400,100,30)
            setText("draw")
            setClickedEvent("draw() ")
        }

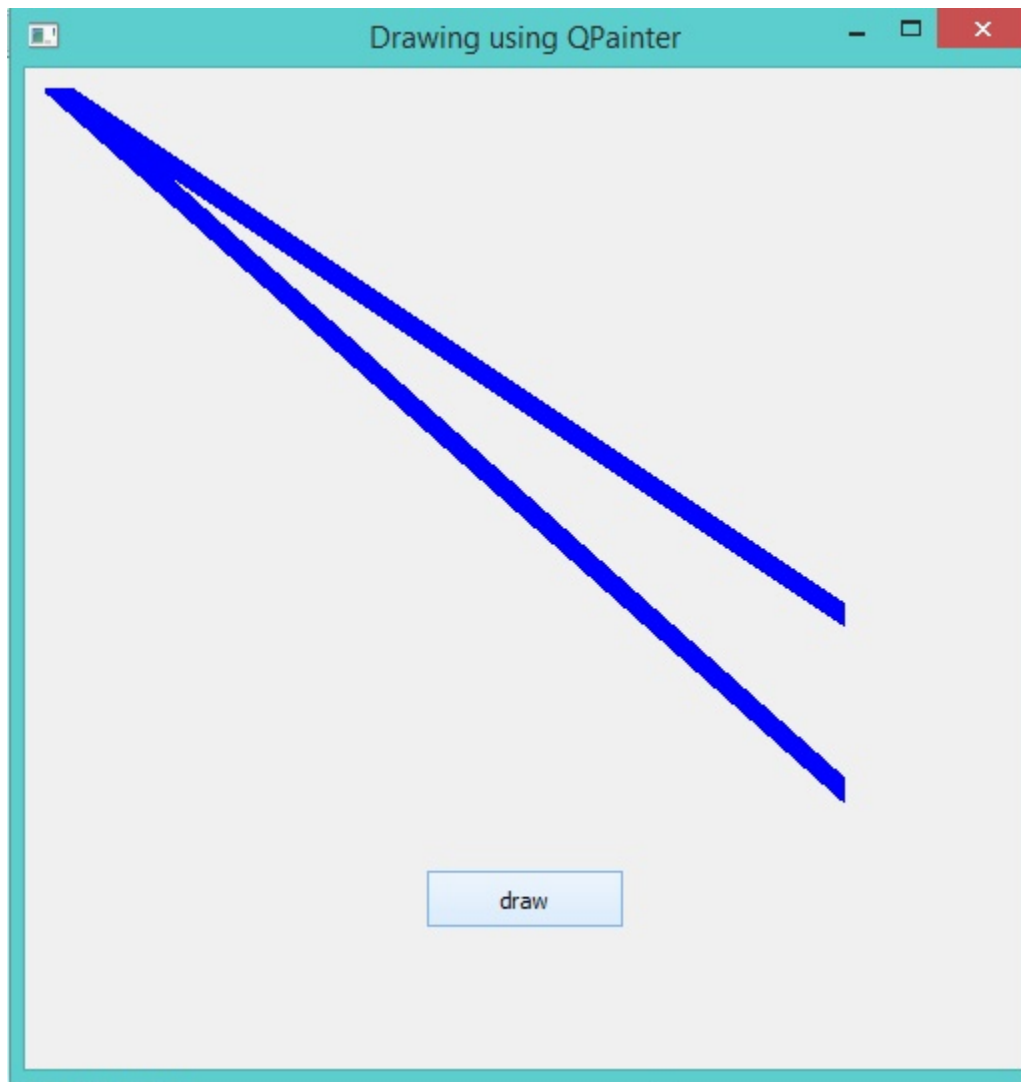
        show()
    }
}

```

```
        exec()
    }

    Func draw
        p1 = new qpicture()
        color = new qcolor() {
            setrgb(0,0,255,255)
        }
        pen = new qpen() {
            setcolor(color)
            setwidth(10)
        }
        new QPainter() {
            begin(p1)
            setpen(pen)
            drawline(500,150,950,450)
            drawline(950,550,500,150)
            endpoint()
        }
        label1 { setpicture(p1) show() }
```

The application during the runtime



57.35 Printing using QPainter

In this example we will learn how to print to PDF file using QPainter

```
Load "guilib.ring"
new QApplication {
    win1 = new QWidget() {
        setWindowTitle("Printer")
        setGeometry(100,100,500,500)
        myweb = new QWebView(win1) {
            setGeometry(100,100,1000,500)
            loadPage(new QUrl("http://google.com"))
        }
        new QPushButton(win1) {
            setGeometry(20,20,100,30)
            setText("Print")
            setClickedEvent("print()")
        }
    }
    showMaximized()
}
```

```

    }
    exec()
}

func print
    printer1 = new QPainter() {
        setoutputformat(1)      # 1 = pdf
        setoutputfilename("test.pdf")
        painter = new QPainter() {
            begin(printer1)
            myfont = new QFont("Times", 50, -1, 0)
            setfont(myfont)
            drawtext(100, 100, "test")
            printer1.newpage()
            drawtext(100, 100, "test2")
            endpoint()
        }
    }

    printer1 = new QPainter() {
        setoutputformat(1)
        setoutputfilename("test2.pdf")
        myweb.print(printer1)
        myweb.show()
    }

    system ("test.pdf")
    system ("test2.pdf")

```

57.36 Creating More than one Window

The next example demonstrates how to create more than one window

```

Load "guilib.ring"
appl = new qapp {
    win1 = new QWidget() {
        setwindowtitle("First")
        setgeometry(100, 100, 500, 500)

        new QPushButton(win1) {
            setgeometry(100, 100, 100, 30)
            settext("close")
            setclideanvent("appl.quit()")
        }

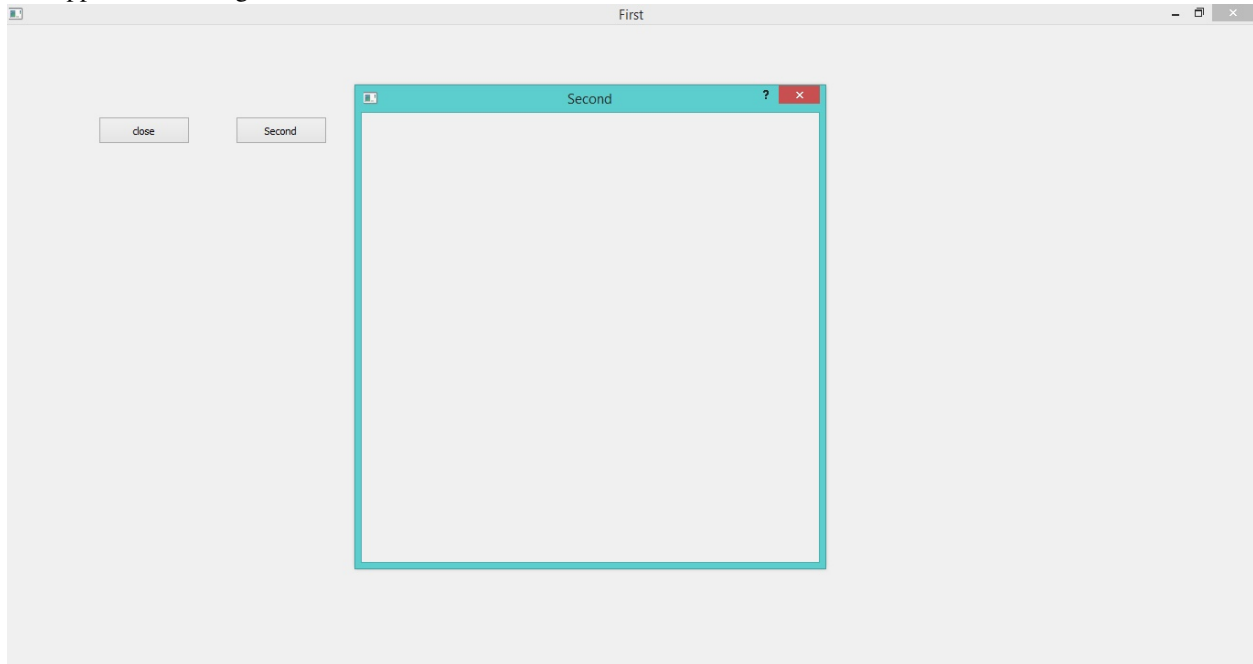
        new QPushButton(win1) {
            setgeometry(250, 100, 100, 30)
            settext("Second")
            setclideanvent("second()")
        }

        showmaximized()
    }
    exec()
}

```

```
func second
    win2 = new QWidget() {
        setWindowTitle("Second")
        setGeometry(100,100,500,500)
        setWindowFlags(Qt_dialog)
        show()
    }
```

The application during the runtime



57.37 Playing Sound

Example:

```
Load "guilib.ring"
new qapp {
    win1 = new QWidget() {
        setWindowTitle("play sound!") show()
    }
    new QMediaPlayer() {
        setMedia(new QUrl("footstep.wav"))
        setVolume(50) play()
    }
    exec()
}
```

57.38 Using the QColorDialog Class

Example:


```

Load "guilib.ring"

oApp = new myapp { start() }

Class MyApp

    oColor win1

    Func start

        myapp = new qapp

        win1 = new QMainWindow() {
            setwindowtitle("Color Dialog")
            setgeometry(100,100,400,400)
        }

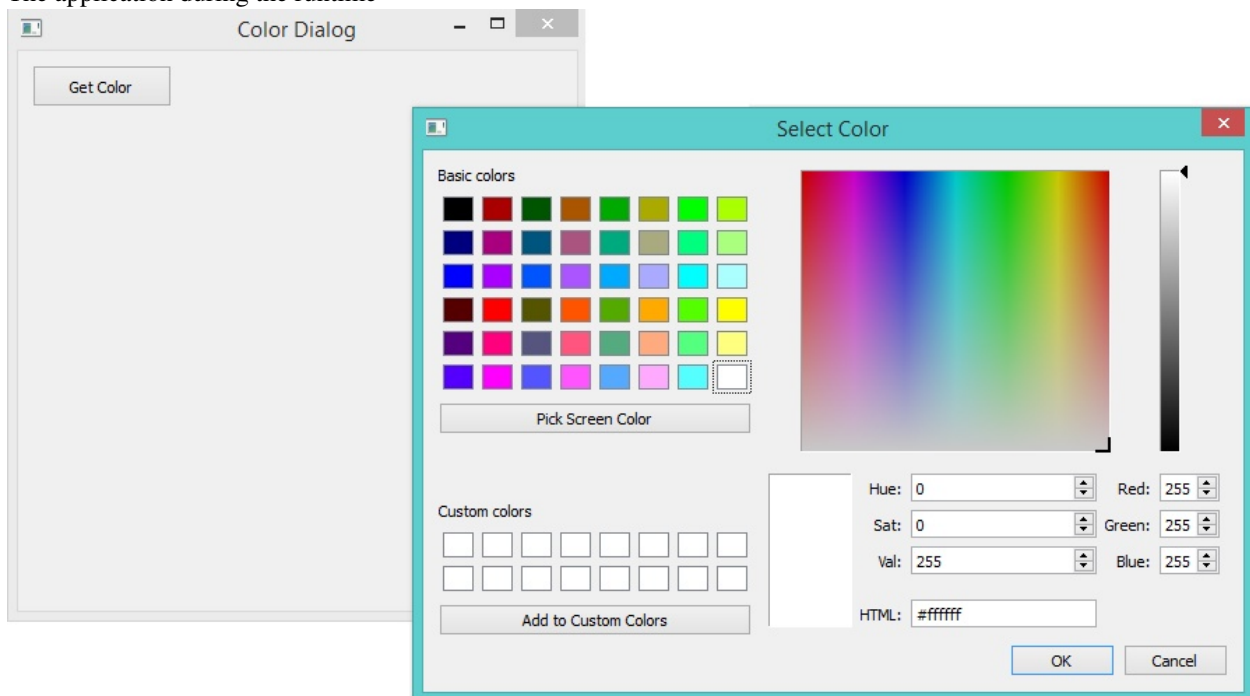
        new QPushButton(win1) {
            setgeometry(10,10,100,30)
            settext("Get Color")
            setclideanvent("oApp.pColor()")
        }

        win1.show()
        myapp.exec()

    Func pColor
        myobj = new QColorDialog()
        aColor = myobj.GetColor()
        r=acolor[1] g=acolor[2] b=acolor[3]
        win1.setStyleSheet("background-color: rgb("+r+", " + g+ ", " + b + ")")

```

The application during the runtime



57.39 Using qLCDNumber Class

In this example we will learn about using the qLCDNumber class

```
Load "guilib.ring"

New QApplication
{
    win1 = new QWidget()
    {
        setWindowTitle("LCD Number")
        setGeometry(100,100,250,120)

        new qLCDNumber(win1)
        {
            setGeometry(10,10,100,40)
            display(100)

        }

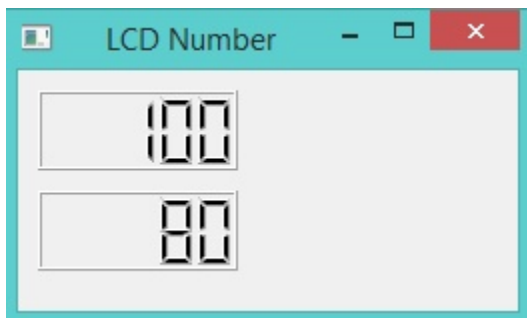
        new qLCDNumber(win1)
        {
            setGeometry(10,60,100,40)
            display(80)

        }

        show()
    }

    exec()
}
```

The application during the runtime



57.40 Movable Label Example

```
Load "guilib.ring"

new QApplication {
    win1 = new QWidget()
    {
```

```

        label1 = new QLabel(win1)
        {
            setText("Welcome")
            setGeometry(10,10,200,50)
            setStyleSheet("color: purple ; font-size: 30pt;")
        }

        new QTimer(win1)
        {
            setInterval(10)
            setTimeoutEvent("pMove()")
            start()
        }

        setWindowTitle("Movable Label")
        setGeometry(100,100,600,80)
        setStyleSheet("background-color: white;")
        show()
    }

    exec()
}

Func pMove
    label1
    {
        move(x()+1,y())
        if x() > 600
            move(10,y())
        ok
    }

```

The application during the runtime



57.41 QMessageBox Example

In this section we will learn how to check the output of the Message box

```

Load "guilib.ring"

new QApplication {
    win1 = new QWidget()
    {
        label1 = new QPushButton(win1)
        {
            setText("Test")
            setGeometry(10,10,200,50)
        }
    }
}

```

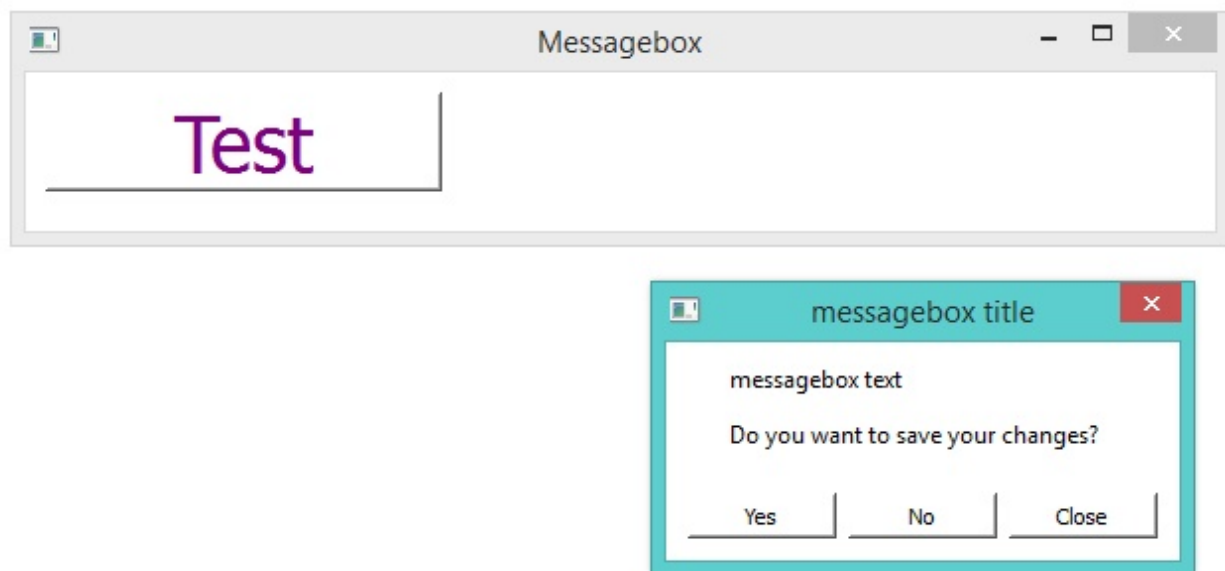
```

        stylesheet("color: purple ; font-size: 30pt;")
        setclickevent("pWork()")
    }
    setWindowTitle("Messagebox")
    setgeometry(100,100,600,80)
    setStyleSheet("background-color: white;")
    show()
}
exec()
}

func pWork
    new QMessageBox(win1)
    {
        setWindowTitle("messagebox title")
        setText("messagebox text")
        setInformativeText("Do you want to save your changes?")
        setStandardButtons(QMessageBox_Yes | QMessageBox_No | QMessageBox_Close)
        result = exec()
        win1 {
            if result = QMessageBox_Yes
                setWindowTitle("Yes")
            but result = QMessageBox_No
                setWindowTitle("No")
            but result = QMessageBox_Close
                setWindowTitle("Close")
            ok
        }
    }
}

```

The application during the runtime



57.42 Using QInputDialog Class

In the next example we will learn about using the QInputDialog class

```

Load "guilib.ring"

New QApp {

    Win1 = New QWidget () {

        SetGeometry(100,100,400,400)
        SetWindowTitle("Input Dialog")

        New QPushButton(win1)
        {

            SetText ("Input Dialog")
            SetGeometry(100,100,100,30)
            SetClickEvent ("pWork() ")

        }

        Show()

    }

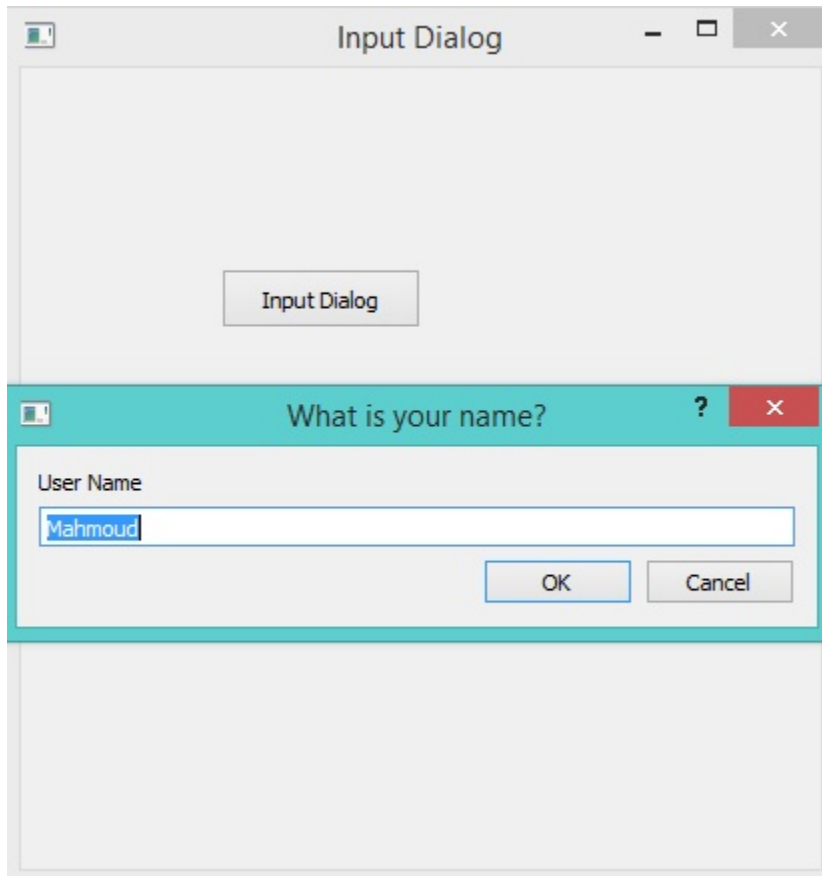
    exec()

}

Func pWork
    oInput = New QDialog(win1)
    {
        setwindowtitle("What is your name?")
        setgeometry(100,100,400,50)
        setlabeltext("User Name")
        settextvalue("Mahmoud")
        lcheck = exec()
        if lCheck win1.setwindowtitle(oInput.textvalue()) ok
    }
}

```

The application during the runtime



57.43 Dialog Functions

We have the next functions

```
SetDialogIcon(cIconFile)
MsgInfo(cTitle,cMessage)
ConfirmMsg(cTitle,cMessage) --> lResult
InputBox(cTitle,cMessage) --> cValue
InputBoxInt(cTitle,cMessage) --> nValue
InputBoxNum(cTitle,cMessage) --> nValue
InputBoxPass(cTitle,cMessage) --> cValue
```

Example

```
load "guilib.ring"

new qApp
{
    SetDialogIcon("notepad.png")
    msginfo(:Ring, :Welcome)
    see confirmMsg(:Ring, "Are you sure?") + nl
    see InputBoxNum(:Ring, "Enter Number(double) :") + nl
    see InputBox(:Ring, "Enter Value :") + nl
    see InputBoxInt(:Ring, "Enter Number(int)") + nl
    see InputBoxPass(:Ring, "Enter Password") +nl
}
```

57.44 KeyPress and Mouse Move Events

In this example we will learn how to use the Events Filter to know about KeyPress and Mouse Move Events

```
Load "guilib.ring"

new qApp {

    win1 = new QWidget()
    {
        setWindowTitle("Test using Event Filter!")
        setGeometry(100,100,400,400)
        setMouseTracking(true)
        myfilter = new QEventFilter(win1)
        myfilter.setKeyPressEvent("pWork()")
        myfilter.setMouseButtonPressEvent("pClick()")
        myfilter.setMouseMoveEvent("pMove()")

        installEventFilter(myfilter)

        show()
    }

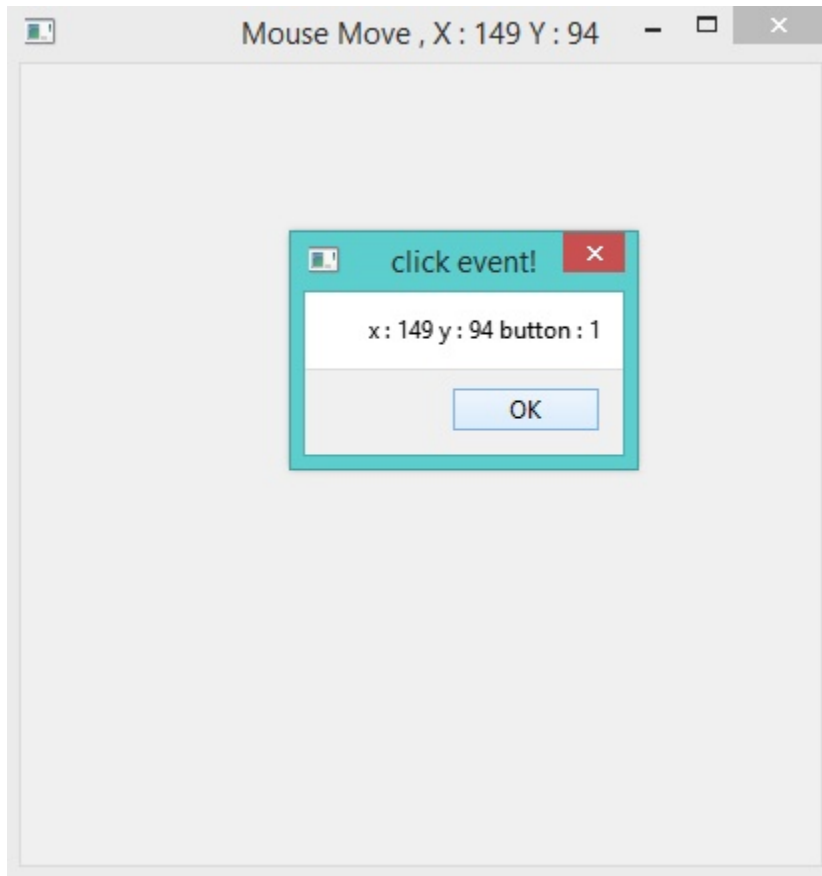
    exec()
}

func pWork
    win1.setWindowTitle('KeyPress! : ' + myfilter.getKeyCode())

func pClick
    new QMessageBox(win1) {
        setGeometry(100,100,400,100)
        setWindowTitle("click event!")
        setText("x : " + myfilter.getX() +
                " y : " + myfilter.getY() + " button : " +
                myfilter.getButton() )
        show()
    }

func pMove
    win1.setWindowTitle("Mouse Move , X : " + myfilter.getX() +
                        " Y : " + myfilter.getY() )
```

The application during the runtime



57.45 Moving Objects using the Mouse

In the next example we will learn how to program movable objects where the user can move a label

```
Load "guilib.ring"

lPress = false
nX = 0
nY = 0

new QApplication {

    win1 = new QWidget()
    {

        setWindowTitle("Move this label!")
        setGeometry(100,100,400,400)
        setstylesheet("background-color:white;")

        Label1 = new QLabel(win1){
            setGeometry(100,100,200,50)
            setText("Welcome")
            setstylesheet("font-size: 30pt")
            myfilter = new QFilterEvents(label1)
            myfilter.setEnterEvent("pEnter() ")
            myfilter.setLeaveEvent("pLeave() ")
        }
    }
}
```



```

        myfilter.setMouseButtonPressEvent("pPress()")
        myfilter.setMouseButtonReleaseEvent("pRelease()")
        myfilter.setMouseMoveEvent("pMove()")
        installeventfilter(myfilter)
    }

    show()
}

exec()
}

Func pEnter
    Labell.setStyleSheet("background-color: purple; color:white;font-size: 30pt;")

Func pLeave
    Labell.setStyleSheet("background-color: white; color:black;font-size: 30pt;")

Func pPress
    lPress = True
    nX = myfilter.getglobalx()
    ny = myfilter.getglobaly()

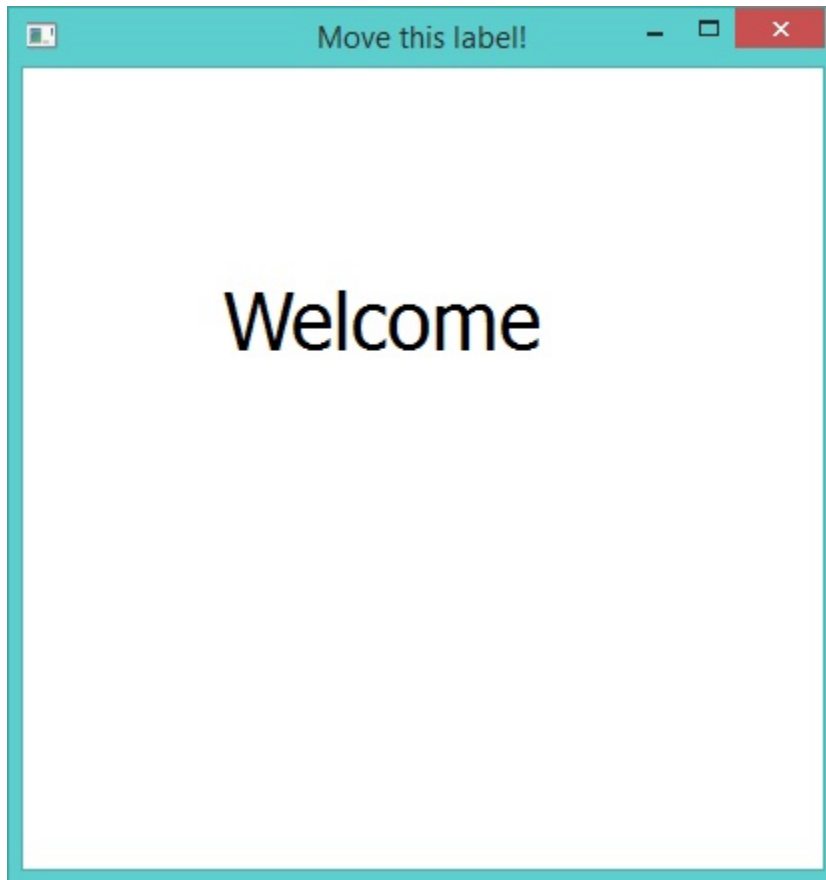
Func pRelease
    lPress = False
    pEnter()

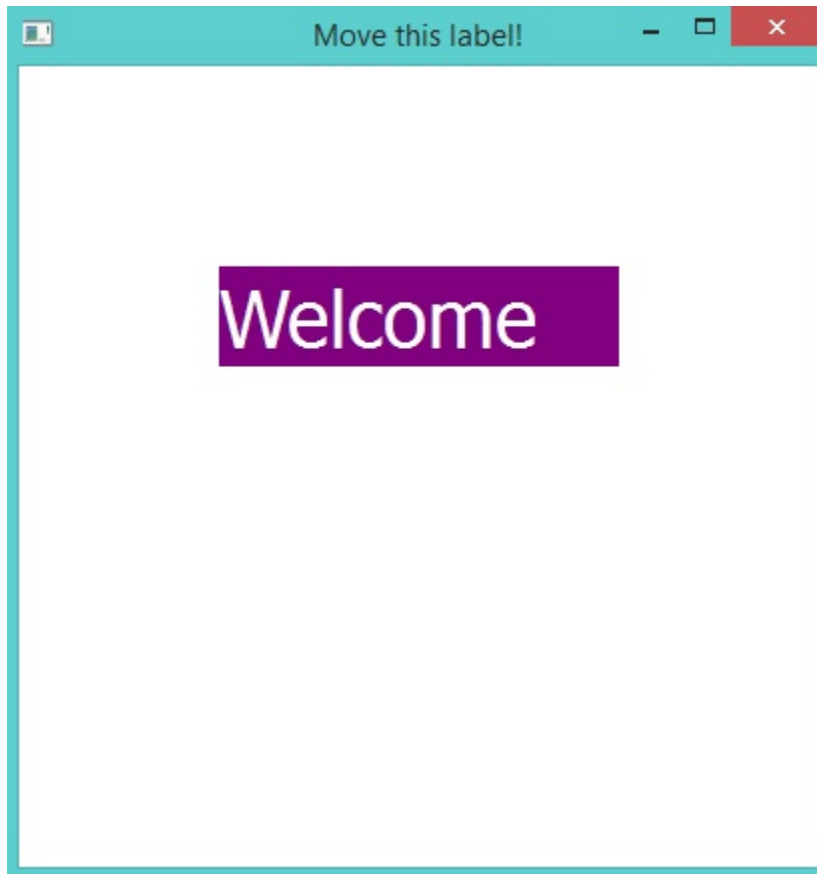
Func pMove
    nX2 = myfilter.getglobalx()
    ny2 = myfilter.getglobaly()
    ndiffx = nX2 - nX
    ndiffy = ny2 - ny
    if lPress
        Labell {
            move(x()+ndiffx,y()+ndiffy)
            setStyleSheet("background-color: Green;
                           color:white;font-size: 30pt;")
            nX = nX2
            ny = ny2
        }

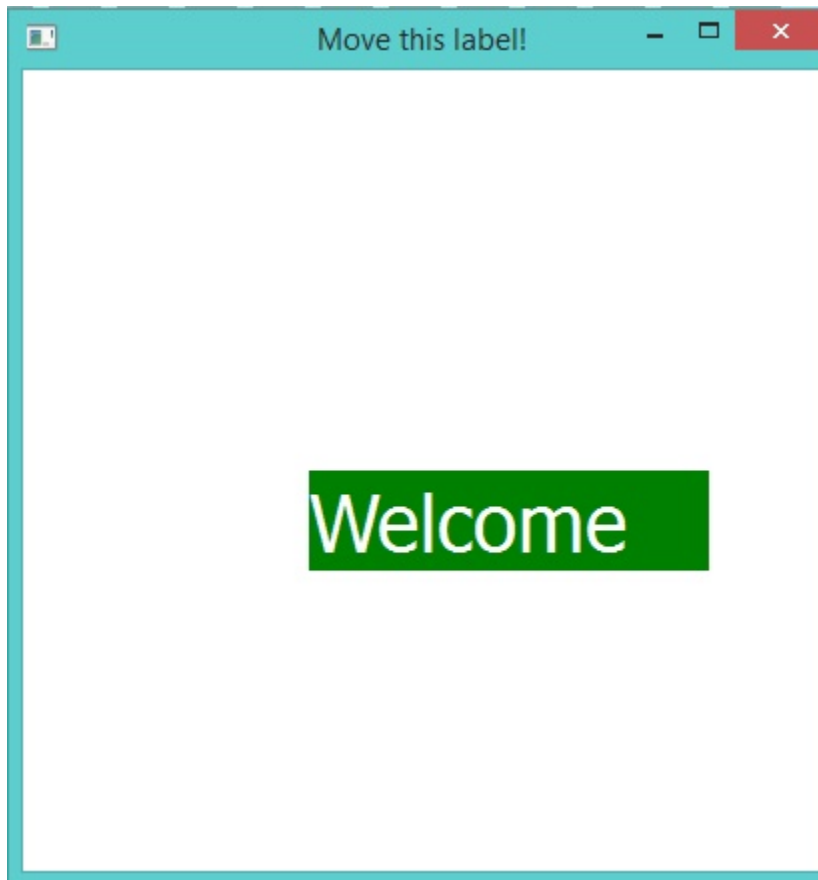
    ok

```

The application during the runtime







57.46 Inheritance from GUI Classes

Example :

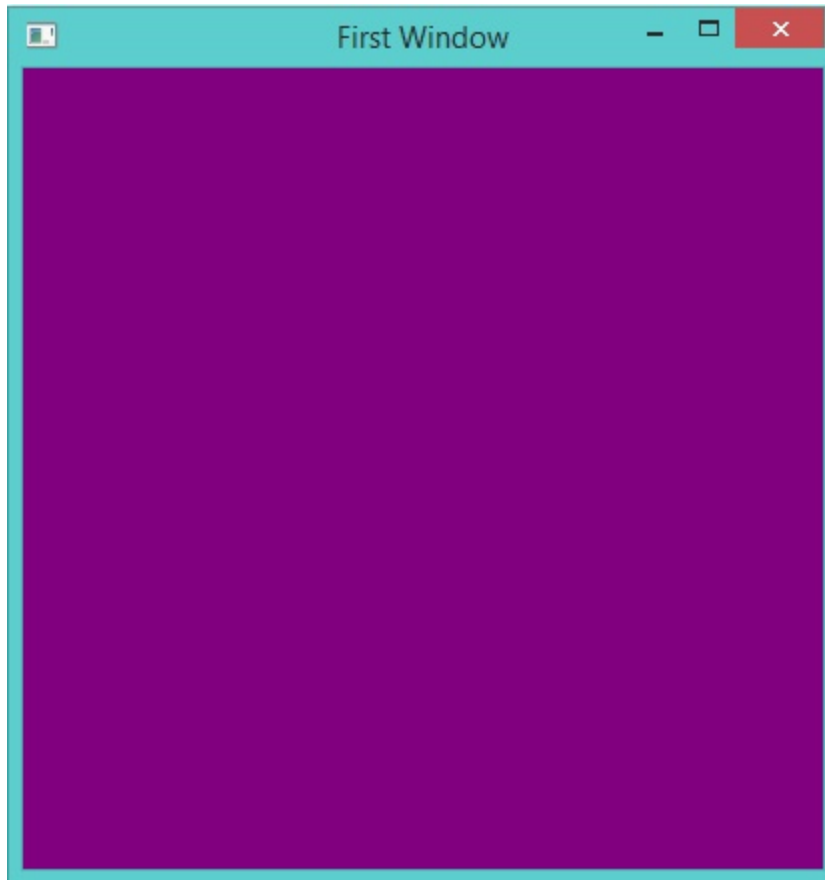
```
Load "guilib.ring"

New MyWindow()

new QApplication { exec() }

class mywindow from QWidget
    Func init
        super.init()
        setWindowTitle("First Window")
        setGeometry(100,100,400,400)
        setStyleSheet("background-color: purple;")
        setToolTip("my first window!")
        show()
```

The application during the runtime



57.47 Using QDesktopWidget Class

In the next example we will learn about using the QDesktopWidget class

```
Load "guilib.ring"

New qApp {
    win1 = New QWidget()
    {
        resize(400,400)
        btn1 = new QPushButton(win1)
        {
            setText("Center")
            move(100,100)
            resize(100,30)
            setClickEvent("pCenter()")
        }

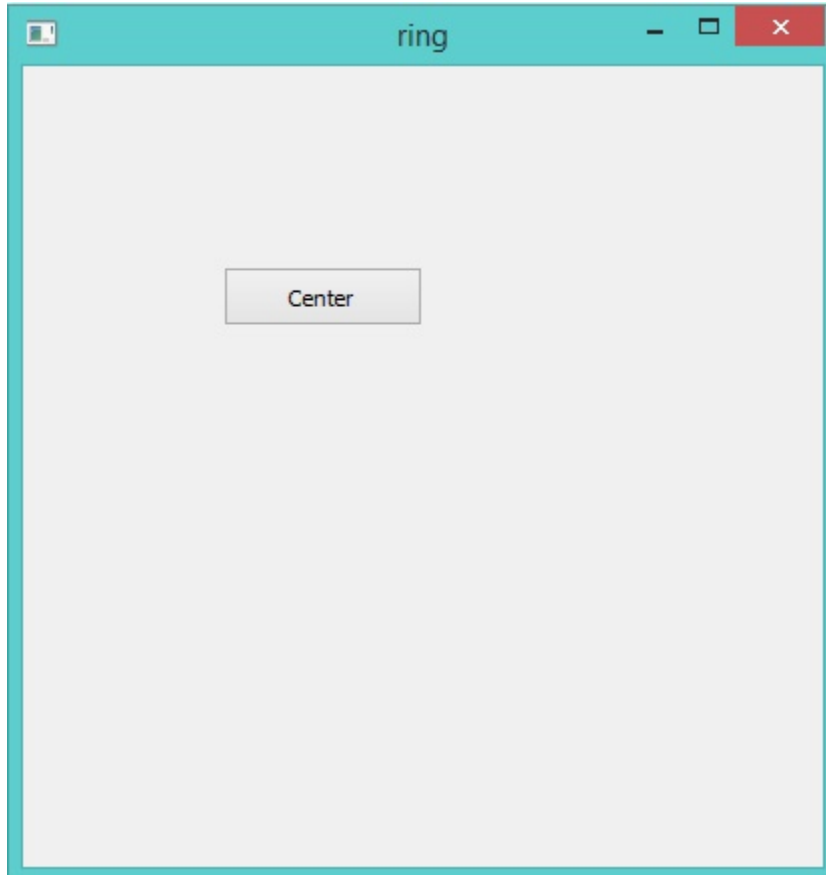
        Show()
    }

    exec()
}

Func pCenter
    oDesktop = new QDesktopWidget()
```

```
oRect = oDesktop.screenGeometry( oDesktop.primaryScreen() )
win1.move((oRect.width()-win1.width()) /2 , (oRect.Height()-win1.Height())/2 )
win1.show()
```

The application during the runtime



57.48 Rotate Text

The next example rotate text using a Timer.

```
Load "guilib.ring"

nAngle = 0

New qapp {
    win1 = new QWidget() {
        setWindowTitle("Rotate Text")
        resize(800,600)
        labell = new QLabel(win1) {
            setText("")
            myfilter = new QEventFilter(win1)
            myfilter.setMouseButtonPressEvent("pClick()")
            installEventFilter(myfilter)
        }
        new QTimer(win1) {
            setInterval(50)
        }
    }
}
```

```

        settimeoutevent("pTime()")
        start()
    }
    pDraw()
    L1 = new QVBoxLayout() { AddWidget(Label1) } SetLayout(L1)
    showMaximized()
}
exec()
}

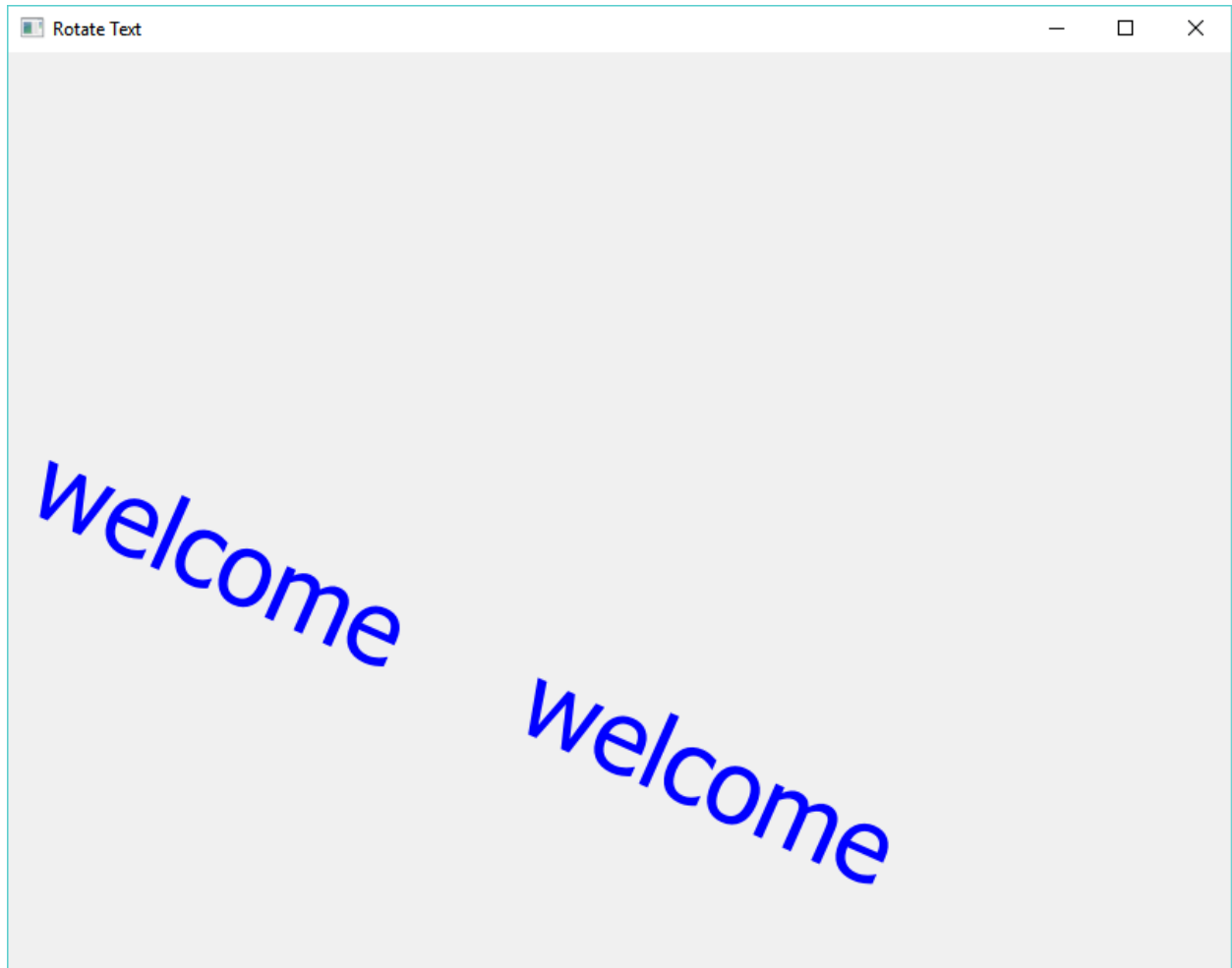
Func pDraw
    p1 = new qpicture()
    color = new qcolor() {
        setrgb(0,0,255,255)
    }
    pen = new qpen() {
        setcolor(color)
        setwidth(50)
    }
    painter = new QPainter() {
        begin(p1)
            setpen(pen)
            myfont = font()
            myfont.setpointsize(50)
            setfont(myfont)
            rotate(nAngle)
            drawtext(350,0*nAngle,"welcome")
            drawtext(0,0*nAngle,"welcome")
        endpoint()
    }
    label1 {
        setpicture(p1)
        show()
    }
}

Func pClick
    win1 { setwindowtitle("Click Event") }

Func pTime
    nAngle++
    if nAngle = 90
        nAngle = 10
    ok
    pDraw()

```

The application during the runtime



57.49 Change Focus

The next example change the focus using the ENTER key.

```
load "guilib.ring"

new QApplication {
    win = new QWidget() {
        resize(600,600)
        SetWindowTitle("Change Focus")
        text1 = new QLineEdit(win)
        text2 = new QLineEdit(win)
        text3 = new QLineEdit(win)
        text4 = new QLineEdit(win)
        layout1 = new QVBoxLayout() {
            AddWidget(text1)
            AddWidget(text2)
            AddWidget(text3)
            AddWidget(text4)
        }
        setLayout(layout1)
    }
}
```



```

        aList = [text1,text2,text3,text4]
        oFilter = new qallevents(win)
        oFilter.setKeyPressEvent("pWork()")
        installeventfilter(oFilter)
        show()
    }
    exec()
}

func pWork
    nCode = oFilter.getkeycode()
    if nCode = 16777220 # ENTER Key
        for x=1 to len(aList)
            if aList[x].HasFocus()
                t = x+1
                if t > len(aList) t=1 ok
                aList[t].SetFocus(0)
                exit
            ok
        next
    ok

```

57.50 Regular Expressions

The next example uses the Regular Expressions classes.

```

load "guilib.ring"

new qApp
{
    see "Using Regular Expressions" + nl

    exp = new qregexexpression() {
        setPattern("\d\d \w+")
        see pattern() + nl
        match = match("33 one",0,0,0)
        see match.hasmatch() + nl
        match = match("3 one",0,0,0)
        see match.hasmatch() + nl
        match = match("welcome 11 one",0,0,0)
        see match.hasmatch() + nl
        matched = match.captured(0)
        see matched + nl
    }

    exp = new qregexexpression() {
        setPattern("^(\\d\\d)/(\\d\\d)/(\\d\\d\\d\\d)$")
        see pattern() + nl
        match = match("08/12/1985",0,0,0)
        see match.hasmatch() + nl
        day = match.captured(1)
        month = match.captured(2)
        year = match.captured(3)
        see day + nl + month + nl + year + nl
        see "(" + match.capturedStart(1) + "," + match.capturedEnd(1)+ ")" + nl
        see "(" + match.capturedStart(2) + "," + match.capturedEnd(2)+ ")" + nl
        see "(" + match.capturedStart(3) + "," + match.capturedEnd(3)+ ")" + nl
    }
}

```

```

    }
}

```

Output

```

Using Regular Expressions
\d\d \w+
1
0
1
11 one
^(\d\d)/(\d\d)/(\d\d\d\d)$
1
08
12
1985
(0,2)
(3,5)
(6,10)

```

57.51 Simple Client and Server Example

In this section we will learn about creating simple Client and Server Application

```

Load "guilib.ring"

new qApp {
    oClient = new Client { client() }
    oServer = new Server { server() }
    exec()
}

Class Client

    win1 lineedit1 cOutput=""
    oTcpSocket

    func client

        win1 = new QWidget()

        new QPushButton(win1) {
            setGeometry(50,50,100,30)
            setText("connect")
            setClickedEvent("oClient.Connect()")
        }

        lineedit1 = new QTextEdit(win1) {
            setGeometry(150,50,200,300)
        }

        win1 {
            setWindowTitle("client")
            setGeometry(10,100,400,400)
            show()
        }
    }

```

```

    }

    func connect
        cOutput = "Connect to host 127.0.0.1 port 9999" + nl
        linedit1.setText(cOutput)
        oTcpSocket = new qTcpSocket(win1) {
            setconnectedevent("oClient.pConnected()")
            setreadyreadevent("oClient.pRead()")
            connecttohost("127.0.0.1", 9999, 3, 0)
            waitforconnected(5000)
        }

    func pConnected

        cOutput += "Connected!" + nl
        linedit1.setText(cOutput)

    func pRead

        cOutput += "Ready Read!" + nl
        linedit1.setText(cOutput)
        cOutput += oTcpSocket.readall().data() + nl
        linedit1.setText(cOutput)

```

Class Server

```

win1 linedit1
oTcpServer oTcpClient
cOutput = ""

func server

    win1 = new QWidget()

    linedit1 = new QTextEdit(win1) {
        setGeometry(150, 50, 200, 300)
    }

    win1 {
        setWindowTitle("Server")
        setGeometry(450, 100, 400, 400)
        show()
    }

    oTcpServer = new QTcpServer(win1) {
        setNewConnectionEvent("oServer.pNewConnection()")
        oHostAddress = new QHostAddress()
        oHostAddress.SetAddress("127.0.0.1")
        listen(oHostAddress, 9999)
    }

    cOutput = "Server Started" + nl +
        "listen to port 9999" + nl

    linedit1.setText(cOutput)

    Func pNewConnection

        oTcpClient = oTcpServer.nextPendingConnection()

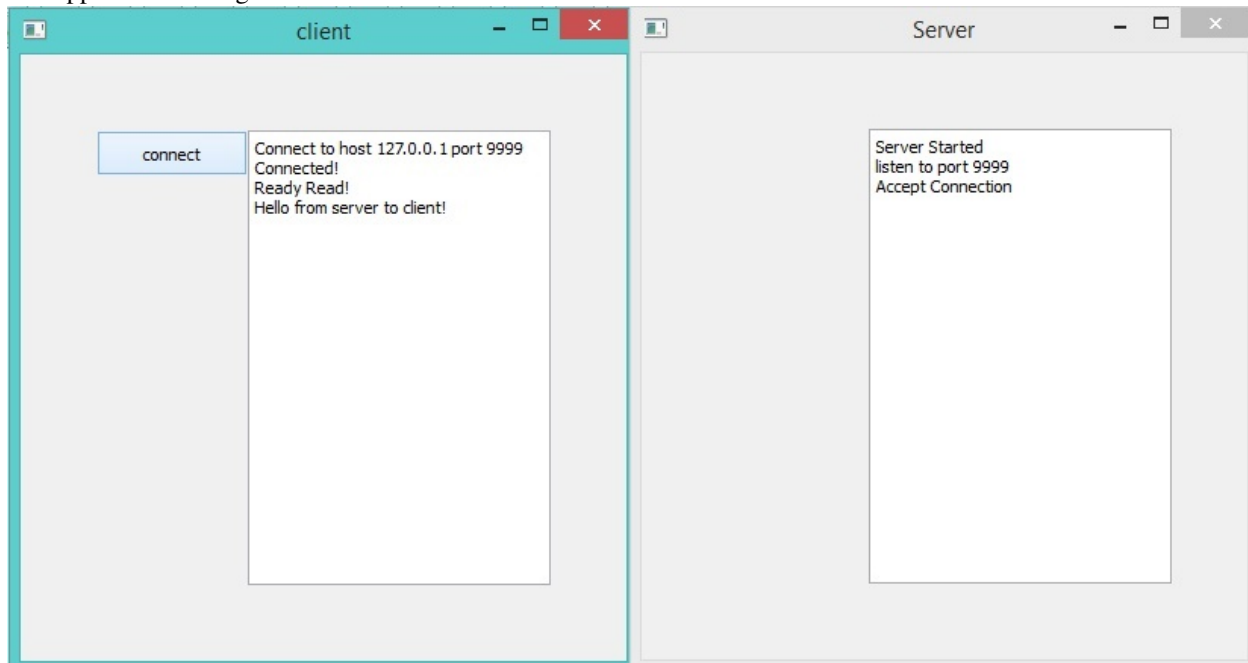
```

```

cOutput += "Accept Connection" + nl
lineEdit1.setText(cOutput)
oTcpClient {
    cStr = "Hello from server to client!" + char(13) + char(10)
    write(cStr, len(cStr))
    flush()
    waitforbyteswritten(300000)
    close()
}

```

The application during the runtime



57.52 Dynamic Objects

We may create objects in the runtime and add them to windows.

Example:

```

load "guilib.ring"

oFormDesigner = new FormDesigner { start("oFormDesigner") }

Class FormDesigner
    winToolBox  winForm

    aObjects = []

    func start cObjectName
        oApp = new qApp

        winToolBox = new QWidget()
        winToolBox.setWindowTitle("ToolBox")

```

```

winToolBox.move(10,10)
winToolBox.resize(300,600)

btn = new QPushButton(winToolBox)
btn.resize(300,30)
btn.setText("Create Button")
btn.setClickEvent(cObjectName+".pCreateButton()")
btn.show()

winToolBox.show()

winForm = new QWidget() {
    move(400,50)
    setWindowTitle("Form Designer")
    resize(600,600)
    show()
}

oApp.exec()

func pCreateButton

    nCount = len(aObjects)

    aObjects + new MyButton(winForm)
    {
        nIndex = nCount + 1
        setText("Button"+ nIndex)
        Move(30*nIndex,30*nIndex)
        resize(100,30)
        show()
    }

Class MyButton from QPushButton
    nIndex = 0

```

57.53 Weight History Application

The next sample help in recording (Date, Time and Weight).

```

Load "guilib.ring"

MyApp = new QApplication
{
    $ApplicationObject = "oApp"    # To be used when calling events
    oApp = new App
    exec()
    oApp.CloseDatabase()
}

class App

    cDir = currentdir() + "/"
    oCon

```

```

aIDs = []

win1 = new QWidget()
{
    setWindowTitle("Weight History")
    resize(600,600)
    layoutButtons = new QHBoxLayout()
    {
        label1 = new QLabel(win1) { setText("Weight") }
        text1 = new QLineEdit(win1)
        btnAdd = new QPushButton(win1) {
            setText("Add")
            setClickEvent($ApplicationObject+".AddWeight()")
        }
        btnDelete = new QPushButton(win1) {
            setText("Delete")
            setClickEvent($ApplicationObject+".Deleteweight()")
        }
        addWidget(label1)
        addWidget(text1)
        addWidget(btnAdd)
        addWidget(btnDelete)
    }
    layoutData = new QHBoxLayout()
    {
        Table1 = new QTableWidgetItem(win1) {
            setRowCount(0)
            setColumnCount(3)
            setSelectionBehavior(QAbstractItemView_SelectRows)
            setHorizontalHeaderItem(0, new QTableWidgetItem("Date"))
            setHorizontalHeaderItem(1, new QTableWidgetItem("Time"))
            setHorizontalHeaderItem(2, new QTableWidgetItem("Weight"))
            setItemChangedEvent($ApplicationObject+".ItemChanged()")
            setAlternatingRowColors(true)
            horizontalHeader().setStyleSheet("color: blue")
            verticalHeader().setStyleSheet("color: red")
        }
        addWidget(Table1)
    }
    layoutClose = new QHBoxLayout()
    {
        btnClose = new QPushButton(win1) {
            setText("Close")
            setClickEvent("MyApp.Quit()")
        }
        addWidget(btnClose)
    }
    layoutMain = new QVBoxLayout()
    {
        addLayout(layoutButtons)
        addLayout(layoutData)
        addLayout(layoutClose)
    }
    setLayout(layoutMain)
    self.OpenDatabase()
    self.ShowRecords()
    show()
}

```

```

Func OpenDatabase
    lCreate = False
    if not fexists(cDir + "weighthistory.db")
        lCreate = True
    ok
    new QSqlDatabase() {
        this.oCon = addDatabase("SQLITE") {
            setDatabaseName("weighthistory.db")
            Open()
        }
    }
    if lCreate
        new QSqlQuery( ) {
            exec("create table weighthistory (id integer primary key, "+
                " f_date varchar(10), "+
                " f_time varchar(8), f_weight varchar(8) );")
            delete()
        }
    ok

Func CloseDatabase
    oCon.Close()

Func AddWeight
    cWeight = text1.text()
    AddRecord(cWeight)

Func DeleteWeight
    Table1 {
        nRow = CurrentRow()
        if nRow >= 0
            nID = this.aIDs[nRow+1]
            new QSqlQuery( ) {
                exec("delete from weighthistory where id = " + nID )
            }
            Del(this.aIDs,nRow+1)
            removerow(nRow)
            selectrow(nRow)
        ok
    }

Func AddRecord cWeight
    new QSqlQuery( ) {
        cStr = "insert into weighthistory (f_date,f_time,f_weight) values"+
            " ('%f1','%f2','%f3') "
        cDate = Date()
        cTime = Time()
        cStr = substr(cStr,"%f1",cDate)
        cStr = substr(cStr,"%f2",cTime)
        cStr = substr(cStr,"%f3",cWeight)
        exec(cStr)
        delete()
    }
    ShowRecords()
    Table1.selectrow(table1.rowcount()-1)

```

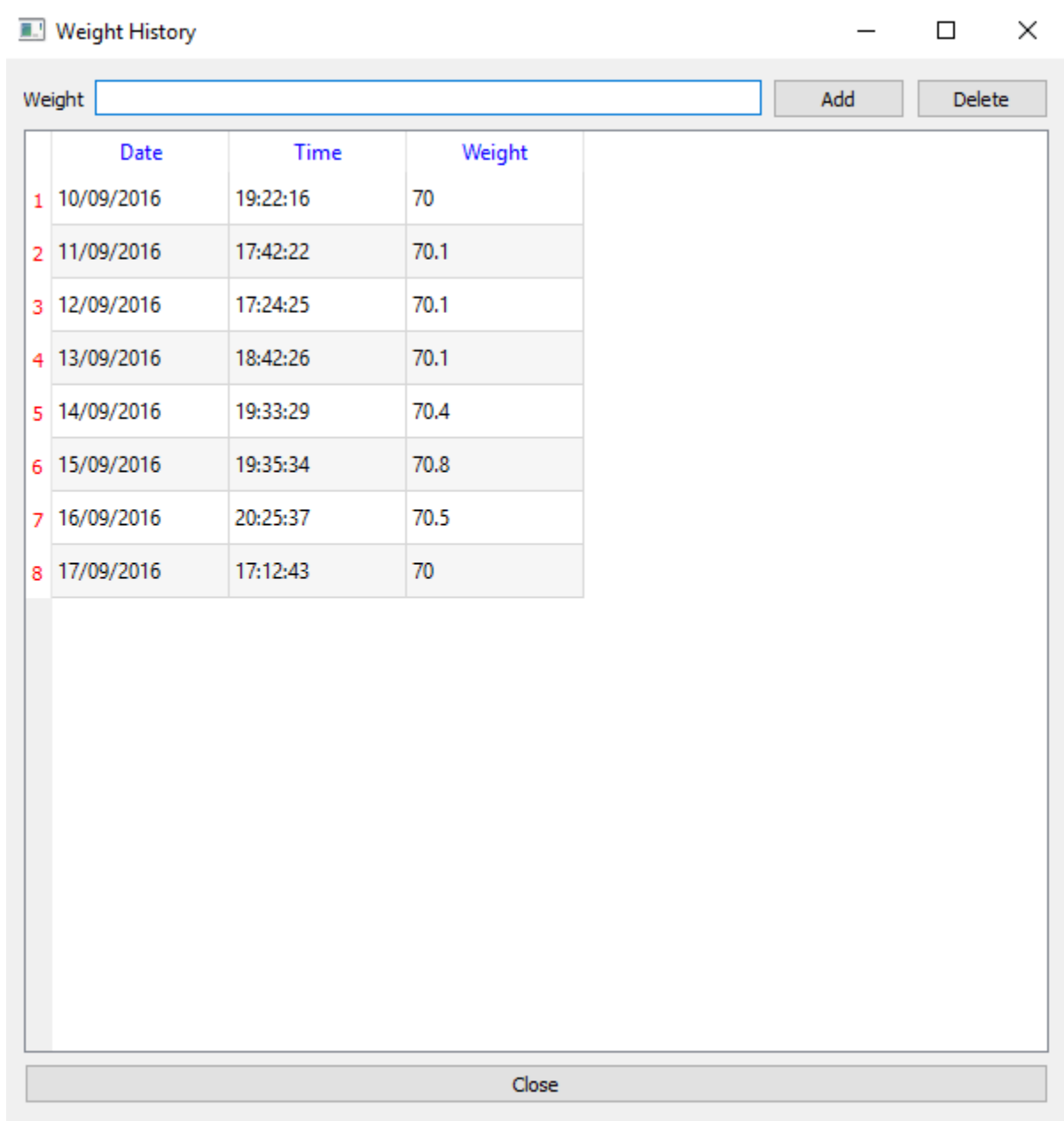
```

Func ShowRecords
    table1.setItemChangedEvent("")
    aIDs = []
    query = new QSqlQuery() {
        exec("select * from weighthistory")
        nRows = 0
        this.Table1.setrowcount(0)
        while movenext()
            this.table1 {
                insertRow(nRows)
                this.aIDs + query.value(0).toString()
                for x = 1 to 3
                    cStr = query.value(x).toString()
                    item = new QTableWidgetItem(cStr)
                    setItem(nRows,x-1,item)
                next
            }
            nRows++
        end
        delete()
    }
    table1.setItemChangedEvent($ApplicationObject+".ItemChanged()")

Func ItemChanged
    nRow = table1.currentrow()
    if nRow >= 0
        myitem = Table1.item(table1.currentrow(),0)
        cDate = myitem.text()
        myitem = Table1.item(table1.currentrow(),1)
        cTime = myitem.text()
        myitem = Table1.item(table1.currentrow(),2)
        cWeight = myitem.text()
        new QSqlQuery( ) {
            cStr = "update weighthistory set f_date ='%f1' , f_time = '%f2' , "+
                "f_weight ='%f3' where id = " + this.aIDs[nRow+1]
            cStr = substr(cStr,"%f1",cDate)
            cStr = substr(cStr,"%f2",cTime)
            cStr = substr(cStr,"%f3",cWeight)
            exec(cStr)
            delete()
        }
    ok

```

The next screen shot for the application during the runtime



57.54 Notepad Application

In the next example we will see simple Notepad developed using the RingQt

```
Load "guilib.ring"

cActiveFileName = ""
aTextColor = [0,0,0]
aBackColor = [255,255,255]
cFont = "MS Shell Dlg 2,14,-1,5,50,0,0,0,0,0"
cWebsite = "http://www.google.com"
```

```

oSearch = NULL
oSearchValue = NULL
oSearchCase = NULL
oSearchFilter = NULL
oReplaceValue = NULL

lAskToSave = false

MyApp = New qApp {
    win1 = new QMainWindow() {

        setWindowTitle("Ring Notepad")
        setGeometry(100,100,400,400)
        aBtns = [

            new QPushButton(win1) {
                setBtnImage(self,"image/new.png")
                setClickedEvent("pNew()")
                setToolTip("New File")
            },
            new QPushButton(win1) {
                setBtnImage(self,"image/open.png")
                setClickedEvent("pOpen()")
                setToolTip("Open File")
            },
            new QPushButton(win1) {
                setBtnImage(self,"image/save.png")
                setClickedEvent("pSave()")
                setToolTip("Save")
            },
            new QPushButton(win1) {
                setBtnImage(self,"image/saveas.png")
                setClickedEvent("pSaveAs()")
                setToolTip("Save As")
            },
            new QPushButton(win1) {
                setBtnImage(self,"image/cut.png")
                setClickedEvent("pCut()")
                setToolTip("Cut")
            },
            new QPushButton(win1) {
                setBtnImage(self,"image/copy.png")
                setClickedEvent("pCopy()")
                setToolTip("Copy")
            },
            new QPushButton(win1) {
                setBtnImage(self,"image/paste.png")
                setClickedEvent("pPaste()")
                setToolTip("Paste")
            },
            new QPushButton(win1) {
                setBtnImage(self,"image/font.png")
                setClickedEvent("pFont()")
                setToolTip("Font")
            },
            new QPushButton(win1) {
                setBtnImage(self,"image/colors.jpg")
                setClickedEvent("pColor()")
                setToolTip("Text Color")
            }
        ]
    }
}

```

```

    } ,
    new QPushButton(win1) {
        setbtnimage(self, "image/search.png")
        setclideanvent("pFind() ")
        settooltip("Find and Replace")
    } ,
    new QPushButton(win1) {
        setbtnimage(self, "image/print.png")
        setclideanvent("pPrint() ")
        settooltip("Print")
    } ,
    new QPushButton(win1) {
        setbtnimage(self, "image/debug.png")
        setclideanvent("pDebug() ")
        settooltip("Debug (Run then wait!)")
    } ,
    new QPushButton(win1) {
        setbtnimage(self, "image/run.png")
        setclideanvent("pRun() ")
        settooltip("Run the program")
    } ,
    new QPushButton(win1) {
        setbtnimage(self, "image/close.png")
        setclideanvent("pQuit() ")
        settooltip("Quit")
    }
}

tool1 = addtoolbar("files") {
    for x in aBtns addwidget(x) addseparator() next
}

menu1 = new QMenuBar(win1) {
    sub1 = addmenu("File")
    sub2 = addmenu("Edit")
    sub3 = addmenu("View")
    sub4 = addmenu("Help")
    sub1 {
        oAction = new QAction(win1) {
            setShortcut(new QKeySequence("Ctrl+n"))
            setbtnimage(self, "image/new.png")
            settext("New")
            setclideanvent("pNew() ")
        }
        addaction(oAction)
        oAction = new QAction(win1) {
            setShortcut(new QKeySequence("Ctrl+o"))
            setbtnimage(self, "image/open.png")
            settext("Open")
            setclideanvent("pOpen() ")
        }
        addaction(oAction)
        addseparator()
        oAction = new QAction(win1) {
            setShortcut(new QKeySequence("Ctrl+s"))
            setbtnimage(self, "image/save.png")
            settext("Save")
            setclideanvent("pSave() ")
        }
    }
}

```

```

    }
    addaction(oAction)
    addseparator()
    oAction = new QAction(win1) {
        setShortcut(new QKeySequence("Ctrl+e"))
        setBtnImage(self, "image/saveas.png")
        setText("Save As")
        setClickedEvent("pSaveAs() ")
    }
    addaction(oAction)
    addseparator()
    oAction = new QAction(win1) {
        setShortcut(new QKeySequence("Ctrl+p"))
        setBtnImage(self, "image/print.png")
        setText("Print to PDF")
        setClickedEvent("pPrint() ")
    }
    addaction(oAction)
    addseparator()
    oAction = new QAction(win1) {
        setShortcut(new QKeySequence("Ctrl+d"))
        setBtnImage(self, "image/debug.png")
        setText("Debug (Run then wait!)")
        setClickedEvent("pDebug() ")
    }
    addaction(oAction)
    addseparator()
    oAction = new QAction(win1) {
        setShortcut(new QKeySequence("Ctrl+r"))
        setBtnImage(self, "image/run.png")
        setText("Run")
        setClickedEvent("pRun() ")
    }
    addaction(oAction)
    addseparator()
    oAction = new QAction(win1) {
        setShortcut(new QKeySequence("Ctrl+F5"))
        setBtnImage(self, "image/run.png")
        setText("Run GUI Application (No Console)")
        setClickedEvent("pRunNoConsole() ")
    }
    addaction(oAction)
    addseparator()
    oAction = new QAction(win1) {
        setShortcut(new QKeySequence("Ctrl+q"))
        setBtnImage(self, "image/close.png")
        setText("Exit")
        setStatusTip("Exit")
        setClickedEvent("pQuit() ")
    }
    addaction(oAction)
}
sub2 {
    oAction = new QAction(win1) {
        setShortcut(new QKeySequence("Ctrl+x"))
        setBtnImage(self, "image/cut.png")
        setText("Cut")
        setClickedEvent("pCut() ")
    }
}

```

```

    }
    addaction(oAction)
    oAction = new QAction(win1) {
        setShortcut(new QKeySequence("Ctrl+c"))
        setBtnImage(self, "image/copy.png")
        setText("Copy")
        setClickedEvent("pCopy()")
    }
    addaction(oAction)
    oAction = new QAction(win1) {
        setShortcut(new QKeySequence("Ctrl+v"))
        setBtnImage(self, "image/paste.png")
        setText("Paste")
        setClickedEvent("pPaste()")
    }
    addaction(oAction)
    addseparator()
    oAction = new QAction(win1) {
        setShortcut(new QKeySequence("Ctrl+i"))
        setBtnImage(self, "image/font.png")
        setText("Font")
        setClickedEvent("pFont()")
    }
    addseparator()
    addaction(oAction)
    oAction = new QAction(win1) {
        setShortcut(new QKeySequence("Ctrl+t"))
        setBtnImage(self, "image/colors.jpg")
        setText("Text Color")
        setClickedEvent("pColor()")
    }
    addaction(oAction)
    oAction = new QAction(win1) {
        setShortcut(new QKeySequence("Ctrl+b"))
        setBtnImage(self, "image/colors.jpg")
        setText("Back Color")
        setClickedEvent("pColor2()")
    }
    addaction(oAction)
    addseparator()
    oAction = new QAction(win1) {
        setShortcut(new QKeySequence("Ctrl+g"))
        setText("Go to line")
        setClickedEvent("pGoto()")
    }
    addaction(oAction)
    oAction = new QAction(win1) {
        setShortcut(new QKeySequence("Ctrl+f"))
        setBtnImage(self, "image/search.png")
        setText("Find and Replace")
        setClickedEvent("pFind()")
    }
    addaction(oAction)
}
sub3 {
    oAction = new QAction(win1) {
        setShortcut(new QKeySequence("Ctrl+p"))
        setBtnImage(self, "image/project.png")
    }
}

```

```

        settext("Project Files")
        setclideanvent("pProject()")
    }
    addaction(oAction)
    oAction = new QAction(win1) {
        setShortcut(new QKeySequence("Ctrl+u"))
        setbtnimage(self,"image/source.png")
        setclideanvent("pSourceCode()")
        settext("Source Code")
    }
    addaction(oAction)
    oAction = new QAction(win1) {
        setShortcut(new QKeySequence("Ctrl+w"))
        setbtnimage(self,"image/richtext.png")
        setclideanvent("pWebBrowser()")
        settext("Web Browser")
    }
    addaction(oAction)
}
sub4 {
    sub5 = addmenu("Development Tools")
    sub5 {

        oAction = new QAction(win1) {
            settext("Programming Language")
            setclideanvent("pLang()")
        }
        addaction(oAction)
        oAction = new QAction(win1) {
            settext("GUI Library")
            setclideanvent("pGUI()")
        }
        addaction(oAction)
    }
    addseparator()
    oAction = new QAction(win1) {
        settext("About")
        setclideanvent("pAbout()")
    }
    addaction(oAction)
}

setmenubar(menu1)

status1 = new qstatusbar(win1) {
    showmessage("Ready!",0)
}

setstatusbar(status1)

tree1 = new qtreeview(win1) {
    setclickedevent("pChangeFile()")
    setGeometry(00,00,200,400)
    oDir = new QDir()
    ofile = new QFileSystemModel() {
        setrootpath(oDir.currentpath())
    }
    myfiles = new qstringlist()
}

```

```

        myfile.append("*.ring")
        myfile.append("*.rh")
        setnamefilters(myfiles)
        setNameFilterDisables(false)
    }
    setmodel(ofile)
    myindex = ofile.index(oDir.currentpath(),0)
    for x = 1 to ofile.columncount()
        hidecolumn(x)
    next
    setcurrentindex(myindex)
    setexpanded(myindex,true)
    header().hide()
}

oDock1 = new qdockwidget(win1,0) {
    setGeometry(00,00,200,200)
    setwindowtitle("Project Files")
    setwidget(tree1)
}

textedit1 = new qtextedit(win1) {
    setCursorPositionChangedEvent("pCursorPositionChanged()")
    setLineWrapMode(QTextEdit_NoWrap)
    setAcceptRichText(false)
    setTextChangedEvent("lAskToSave = true")
}

oDock2 = new qdockwidget(win1,0) {
    setwidget(textedit1)
    setwindowtitle("Source Code")
}

oWebBrowser = new QWidget() {
    setWindowFlags(Qt_SubWindow)
    oWLabel = new QLabel(win1) {
        setText("Website: ")
    }
    oWBText = new QLineEdit(win1) {
        setText(cWebSite)
        setReturnPressedEvent("pWebGo()")
    }
    oWBGo = new QPushButton(win1) {
        setText("Go")
        setClickEvent("pWebGo()")
    }
    oWBBack = new QPushButton(win1) {
        setText("Back")
        setClickEvent("pWebBack()")
    }
    oWLayout1 = new QHBoxLayout() {
        addWidget(oWLabel)
        addWidget(oWBText)
        addWidget(oWBGo)
        addWidget(oWBBack)
    }
    oWebView = new QWebView(win1) {

```

```

        loadpage(new qurl(cWebSite))
    }
    oWLayout2 = new QVBoxLayout() {
        addLayout(oWLayout1)
        addWidget(oWebView)
    }
    setLayout(oWLayout2)
}

oDock3 = new QDockWidget(win1,0) {
    setWidget(oWebBrowser)
    setWindowTitle("Web Browser")
    setFeatures(QDockWidget_DocWidgetClosable)
}

addDockWidget(1,oDock1,1)
addDockWidget(2,oDock2,2)
addDockWidget(2,oDock3,1)

setWinIcon(self,"image/notepad.png")

showMaximized()
}
RestoreSettings()
exec()
}

func pWebGo
    cWebsite = oWBText.text()
    oWebView.LoadPage( new qurl( cWebSite ) )

func pWebBack
    oWebView.Back()

func pProject
    oDock1.Show()

func pSourceCode
    oDock2.Show()

func pWebBrowser
    oDock3.Show()

func pChangeFile
    myitem = tree1.currentIndex()
    if ofile.isdir(myitem)
        return

    ok
    cActiveFileName = ofile.filepath(myitem)
    textedit1.setText(read(cActiveFileName))
    textedit1.setFocus(0)
    pCursorPositionChanged()
    pSetActiveFileName()

func pSetActiveFileName
    oDock2.setWindowTitle("Source Code : " + cActiveFileName)

func pCursorPositionChanged

```



```

        status1.showMessage(" Line : "+(textedit1.textcursor().blocknumber()+1)+
            " Column : " + (textedit1.textcursor().columnnumber()+1) +
            " Total Lines : " + textedit1.document().linecount() ,0)

func pGoto
    oInput = New QInputDialog(win1)
    {
        setwindowtitle("Enter the line number?")
        setgeometry(100,100,400,50)
        setlabeltext("Line")
        settextvalue("1")
        exec()
        nLine = 0 + oInput.textvalue()
        oBlock = textedit1.document().findBlockByLineNumber(nLine-1)
        oCursor = textedit1.textcursor()
        oCursor.setPosition(oBlock.position(),0)
        textedit1.setTextcursor(oCursor)
    }

func pFind
    if isobject(oSearch)
        oSearch.activatewindow()
        return
    ok
    oSearch = new QWidget()
    {
        new QLabel(oSearch)
        {
            setText("Find What : ")
            setgeometry(10,10,50,30)
        }
        oSearchValue = new QLineEdit(oSearch)
        {
            setgeometry(80,10,460,30)
            setReturnPressedEvent("pFindValue()")
        }
        new QLabel(oSearch)
        {
            setText("Replace with ")
            setgeometry(10,45,80,30)
        }
        oReplaceValue = new QLineEdit(oSearch)
        {
            setgeometry(80,45,460,30)
        }
        oSearchCase = new QCheckBox(oSearch)
        {
            setText("Case Sensitive")
            setgeometry(80,85,100,30)
        }
        new QPushButton(oSearch)
        {
            setText("Find/Find Next")
            setgeometry(80,120,100,30)
            setClickedEvent("pFindValue()")
        }
        new QPushButton(oSearch)
        {

```

```

        setText("Replace")
        setGeometry(200,120,100,30)
        setClickedvent("pReplace()")
    }
    new QPushButton(oSearch)
    {
        setText("Replace All")
        setGeometry(320,120,100,30)
        setClickedvent("pReplaceAll()")
    }
    new QPushButton(oSearch)
    {
        setText("Close")
        setGeometry(440,120,100,30)
        setClickedvent("pSearchClose()")
    }

    setWinIcon(oSearch,"image/notepad.png")
    setWindowTitle("Find/Replace")
    setStyleSheet("background-color:white;")
    setFixedSize(550,160)
    setWindowFlags( Qt_CustomizeWindowHint |
                    Qt_WindowTitleHint | Qt_WindowStaysOnTopHint)

    oSearchFilter = new Qallevnts(oSearch)
    oSearchFilter.setKeyPressEvent("pSearchKeyPress()")
    installEventFilter(oSearchFilter)

    show()
}

Func pReplace
oCursor = textedit1.textCursor()
if oCursor.HasSelection() = false
    new QMessageBox(oSearch)
    {
        SetWindowTitle("Replace")
        SetText("No Selection")
        show()
    }
    return false

ok
cValue = oSearchValue.text()
cSelected = oCursor.SelectedText()
if oSearchCase.checkState() = Qt_Unchecked
    cValue = lower(cValue)
    cSelected = lower(cSelected)

ok
if cSelected != cValue
    new QMessageBox(oSearch)
    {
        SetWindowTitle("Replace")
        SetText("No Match")
        show()
    }
    return false

ok
cValue = oReplaceValue.text()

```

```

nStart = oCursor.SelectionStart()
nEnd = oCursor.SelectionEnd()
cStr = textedit1.toPlainText()
cStr = left(cStr,nStart)+cValue+substr(cStr,nEnd+1)
textedit1.setText(cStr)
return pFindValue()

Func pReplaceAll
    cStr = textedit1.toPlainText()
    cOldValue = oSearchValue.text()
    cNewValue = oReplaceValue.text()
    if oSearchCase.checkState() = Qt_Unchecked
        # Not Case Sensitive
        cStr = SubStr(cStr,cOldValue,cNewValue,true)
    else
        # Case Sensitive
        cStr = SubStr(cStr,cOldValue,cNewValue)
    ok
    textedit1.setText(cStr)
    new QMessageBox(oSearch)
    {
        SetWindowTitle("Replace All")
        SetText("Operation Done")
        show()
    }

Func pSearchClose
    oSearch.close()
    oSearch = NULL

func pSearchKeyPress
    if oSearchFilter.getKeyCode() = Qt_Key_Escape
        pSearchClose()
    ok

func pFindValue
    oCursor = textedit1.textcursor()
    nPosStart = oCursor.Position() + 1
    cValue = oSearchValue.text()
    cStr = textedit1.toplaintext()
    cStr = substr(cStr,nPosStart)
    if oSearchCase.checkState() = Qt_Unchecked
        cStr = lower(cStr)  cValue = lower(cValue)
    ok
    nPos = substr(cStr,cValue)
    if nPos > 0
        nPos += nPosStart - 2
        oCursor = textedit1.textcursor()
        oCursor.setposition(nPos,0)
        textedit1.settextcursor(oCursor)
        oCursor = textedit1.textcursor()
        oCursor.setposition(nPos+len(cValue),1)
        textedit1.settextcursor(oCursor)
        return true
    else
        new QMessageBox(oSearch)
        {
            SetWindowTitle("Search")

```

```

        SetText("Cannot find :" + cValue)
        show()
    }
    return false
ok

func pNofileopened
    New qMessageBox(win1) {
        setWindowTitle("Sorry")
        setText("Save the file first!")
        show()
    }

func pDebug
    if cActiveFileName = Null return pNofileopened() ok
    cCode = "start run " + cActiveFileName + nl
    system(cCode)

func pRun
    if cActiveFileName = Null return pNofileopened() ok
    cCode = "start ring " + cActiveFileName + nl
    system(cCode)

func pRunNoConsole
    if cActiveFileName = Null return pNofileopened() ok
    cCode = "start /b ring " + cActiveFileName + nl
    system(cCode)

func pSave
    if cActiveFileName = NULL return pSaveAs() ok
    writefile(cActiveFileName, textedit1.toplaintext())
    status1.showmessage("File : " + cActiveFileName + " saved!", 0)
    lAskToSave = false

func pSaveAs
    new qfiledialog(win1) {
        cName = getsavefilename(win1, "Save As", "", "source files(*.ring)")
        if cName != NULL
            cActiveFileName = cName
            writefile(cActiveFileName, textedit1.toplaintext())
            status1.showmessage("File : " + cActiveFileName + " saved!", 0)
            pSetActiveFileName()
            lAskToSave = false
        ok
    }

func pPrint
    status1.showmessage("Printing to File : RingDoc.pdf", 0)
    printer1 = new qPrinter(0) {
        setoutputformat(1) # 1 = pdf
        setoutputfilename("RingDoc.pdf")
        textedit1.print(printer1)
    }
    status1.showmessage("Done!", 0)
    system("RingDoc.pdf")

func pCut
    textedit1.cut()

```

```

        status1.showmessage("Cut!",0)

func pCopy
    textedit1.copy()
    status1.showmessage("Copy!",0)

func pPaste
    textedit1.paste()
    status1.showmessage("Paste!",0)

func pFont
    oFontDialog = new qfontdialog() {
        aFont = getfont()
    }
    textedit1.selectAll()
    cFont = aFont[1]
    pSetFont()

Func pSetFont
    myfont = new qfont("",0,0,0)
    myfont.fromstring(cFont)
    textedit1.setcurrentfont(myfont)

Func pColor
    new qcolordialog() { aTextColor = GetColor() }
    pSetColors()

Func pColor2
    new qcolordialog() { aBackColor = GetColor() }
    pSetColors()

Func pSetColors
    textedit1.setStyleSheet("color: rgb(" + aTextColor[1] + "," + aTextColor[2] +
        "," + aTextColor[3] + ");" + "background-color: rgb(" +
        aBackColor[1] + "," + aBackColor[2] + "," +
        aBackColor[3] + ")")

func pOpen
    new qfiledialog(win1) {
        cName = getopenfilename(win1,"open file","c:\\","source files(*.ring)")
        if cName != NULL
            cActiveFileName = cName
            textedit1.settext(read(cActiveFileName))

        ok
    }

func pNew
    new qfiledialog(win1) {
        cName = getsavefilename(win1,"New file","", "source files(*.ring)")
        if cName != NULL
            write(cName,"")
            cActiveFileName = cName
            textedit1.settext(read(cActiveFileName))

        ok
    }

Func WriteFile cFileName,cCode

```

```

aCode = str2list(cCode)
fp = fopen(cFileName,"wb")
for cLine in aCode
    fwrite(fp,cLine+char(13)+char(10))
next
fclose(fp)

Func MsgBox cTitle,cMessage
    new qMessageBox(win1) {
        setwindowtitle(cTitle)
        setText(cMessage)
        show()
    }

Func pLang
    MsgBox("Programming Language",
        "This application developed using the Ring programming language")

Func pGUI
    MsgBox("GUI Library",
        "This application uses the Qt GUI Library through RingQt")

Func pAbout
    MsgBox("About",
        "2016, Mahmoud Fayed <msfclipper@yahoo.com>")

Func pSaveSettings
    cSettings = "aTextColor = ["+aTextColor[1]+","aTextColor[2]+
        ","aTextColor[3]+"]" + nl +
        "aBackColor = ["+aBackColor[1]+","aBackColor[2]+
        ","aBackColor[3]+"]" + nl +
        "cFont = '" + cFont + "'" + nl +
        "cWebSite = '" + cWebsite + "'" + nl
    cSettings = substr(cSettings,nl,char(13)+char(10))
    write("ringnotepad.ini",cSettings)
    if lAsktoSave
        new qmessagebox(win1)
        {
            setwindowtitle("Save Changes?")
            settext("Some changes are not saved!")
            setInformativeText("Do you want to save your changes?")
            setstandardbuttons(QMessageBox_Yes |
                QMessageBox_No | QMessageBox_Cancel)
            result = exec()
            win1 {
                if result = QMessageBox_Yes
                    pSave()
                but result = QMessageBox_Cancel
                    return false
            }
            ok
        }
    }
    ok
    return true

Func pSetWebsite
    oWebView { loadpage(new qurl(cWebSite)) }

```

```

        oWBText { setText(cWebSite) }

Func RestoreSettings
    eval(read("ringnotepad.ini"))
    pSetColor()
    pSetFont()
    pSetWebsite()

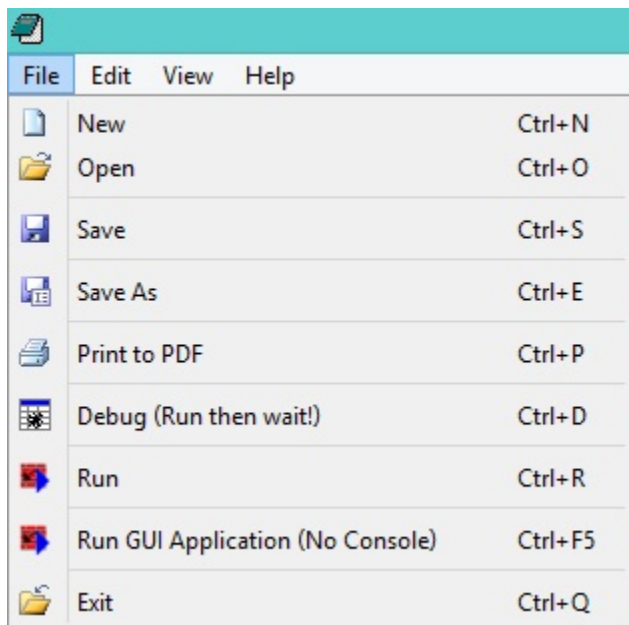
Func pQuit
    if pSaveSettings()
        myapp.quit()

    ok

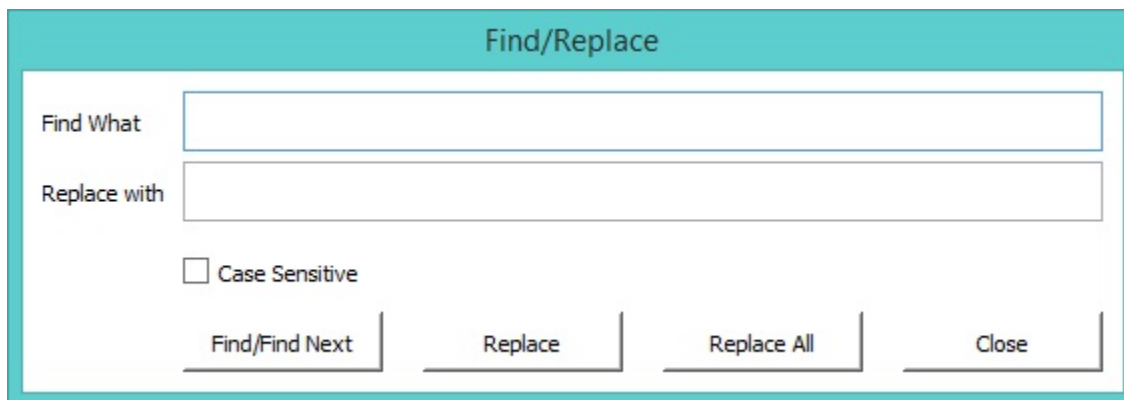
```

The application during the runtime

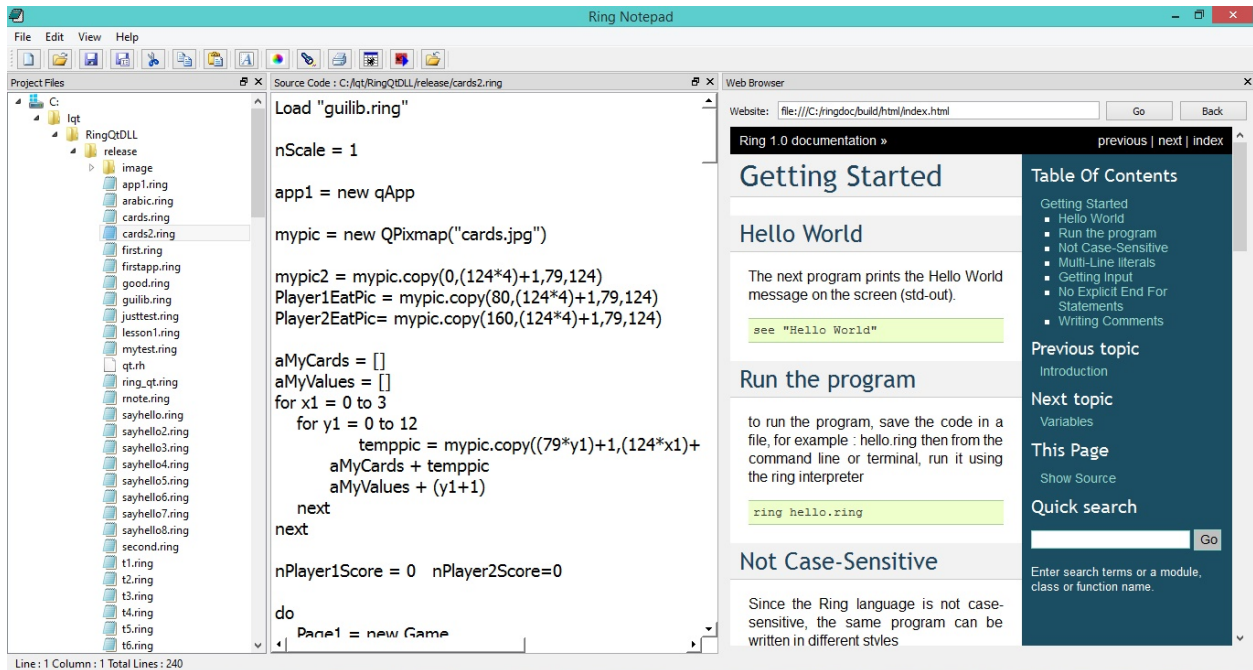
The next screen shot demonstrates the “File” menu



The next window for “search and replace”



The next screen shot demonstrates the application main window



Note: the functions `pDebug()`, `pRun()` and `pRunNoConsole()` in the previous sample are not portable! They are written in this sample for MS-Windows and we can update them for other operating systems.

57.55 The Cards Game

In the next example we will see a simple Cards game developed using RingQt

Each player get 5 cards, the cards are unknown to any one. each time one player click on one card to see it. if the card is identical to another card the play get point for each card. if the card value is "5" the player get points for all visible cards.

```

Load "guilib.ring"

nScale = 1

app1 = new qApp

mypic = new QPixmap("cards.jpg")

mypic2 = mypic.copy(0, (124*4)+1, 79, 124)
Player1EatPic = mypic.copy(80, (124*4)+1, 79, 124)
Player2EatPic= mypic.copy(160, (124*4)+1, 79, 124)

aMyCards = []
aMyValues = []
for x1 = 0 to 3
    for y1 = 0 to 12
        temppic = mypic.copy((79*y1)+1, (124*x1)+1, 79, 124)
        aMyCards + temppic
        aMyValues + (y1+1)
    next
next

```



```

nPlayer1Score = 0    nPlayer2Score=0

do
    Page1 = new Game
    Page1.Start()
again Page1.lnewgame

mypic.delete()
mypic2.delete()
Player1EatPic.delete()
Player2EatPic.delete()

for t in aMyCards
    t.delete()
next

func gui_setbtnpixmap pBtn,pPixmap
    pBtn {
        setIcon(new QIcon(pPixmap.scaled(width(),height(),0,0)))
        setIconSize(new QSize(width(),height()))
    }

Class Game

    nCardsCount = 10
    win1 layout1 label1 label2 layout2 layout3 aBtns aBtns2
    aCards nRole=1 aStatus = list(nCardsCount) aStatus2 = aStatus
    aValues          aStatusValues = aStatus aStatusValues2 = aStatus
    Player1EatPic    Player2EatPic
    lnewgame = false
    nDelayEat = 0.5
    nDelayNewGame = 1

    func start

        win1 = new QWidget() {
            setWindowTitle("Five")
            setStyleSheet("background-color: White")
            showFullScreen()
        }

        layout1 = new QVBoxLayout()

        label1 = new QLabel(win1) {
            setText("Player (1) - Score : " + nPlayer1Score)
            setAlignment(Qt_AlignHCenter | Qt_AlignVCenter)
            setStyleSheet("color: White; background-color: Purple;
                           font-size:20pt")
            setFixedHeight(200)
        }

        closebtn = new QPushButton(win1) {
            setText("Close Application")
            setStyleSheet("font-size: 18px ; color : white ;
                           background-color: black ;")
            setClickedEvent("Page1.win1.close()")
        }

```

```

aCards = aMyCards
aValues = aMyValues

layout2 = new QHBoxLayout()

aBtns = []

for x = 1 to nCardsCount
    aBtns + new QPushButton(win1)
    aBtns[x].setFixedWidth(79*nScale)
    aBtns[x].setFixedHeight(124*nScale)
    gui_setbtnpixmap(aBtns[x],mypic2)
    layout2.addWidget(aBtns[x])
    aBtns[x].setClickedEvent("Page1.Player1click("+x+")")
next

layout1.addWidget(label1)
layout1.addLayout(layout2)

label2 = new QLabel(win1) {
    setText("Player (2) - Score : " + nPlayer2Score)
    setAlignment(Qt_AlignHCenter | Qt_AlignVCenter)
    setStyleSheet("color: white; background-color: red;
                  font-size:20pt")
    setFixedHeight(200)
}

layout3 = new QHBoxLayout()

aBtns2 = []
for x = 1 to nCardsCount
    aBtns2 + new QPushButton(win1)
    aBtns2[x].setFixedWidth(79*nScale)
    aBtns2[x].setFixedHeight(124*nScale)
    gui_setbtnpixmap(aBtns2[x],mypic2)
    layout3.addWidget(aBtns2[x])
    aBtns2[x].setClickedEvent("Page1.Player2click("+x+")")
next

layout1.addWidget(label2)
layout1.addLayout(layout3)
layout1.addWidget(closebtn)

win1.setLayout(layout1)

app1.exec()

Func Player1Click x
    if nRole = 1 and aStatus[x] = 0
        nPos = ((random(100)+clock())%(len(aCards)-1)) + 1
        gui_setbtnpixmap(aBtns[x],aCards[nPos])
        del(aCards,nPos)
        nRole = 2
        aStatus[x] = 1
        aStatusValues[x] = aValues[nPos]
        del(aValues,nPos)
        Player1Eat(x,aStatusValues[x])
        checknewgame()

```

```

    ok

Func Player2Click x
    if nRole = 2 and aStatus2[x] = 0
        nPos = ((random(100)+clock())%(len(aCards)-1)) + 1
        gui_setbtnpixmap(aBtns2[x],aCards[nPos])
        del(aCards,nPos)
        nRole = 1
        aStatus2[x] = 1
        aStatusValues2[x] = aValues[nPos]
        del(aValues,nPos)
        Player2Eat(x,aStatusValues2[x])
        checknewgame()

    ok

Func Player1Eat nPos,nValue

    appl.processEvents()

    delay(nDelayEat)
    lEat = false
    for x = 1 to nCardsCount
        if aStatus2[x] = 1 and (aStatusValues2[x] = nValue or nValue=5)
            aStatus2[x] = 2
            gui_setbtnpixmap(aBtns2[x],Player1EatPic)
            lEat = True
            nPlayer1Score++

        ok
        if (x != nPos) and (aStatus[x] = 1) and
            (aStatusValues[x] = nValue or nValue=5)
            aStatus[x] = 2
            gui_setbtnpixmap(aBtns[x],Player1EatPic)
            lEat = True
            nPlayer1Score++

        ok
    next
    if lEat
        nPlayer1Score++
        gui_setbtnpixmap(aBtns[nPos],Player1EatPic)
        aStatus[nPos] = 2
        label1.setText("Player (1) - Score : " + nPlayer1Score)

    ok

Func Player2Eat nPos,nValue

    appl.processEvents()

    delay(nDelayEat)
    lEat = false
    for x = 1 to nCardsCount
        if aStatus[x] = 1 and (aStatusValues[x] = nValue or nValue = 5)
            aStatus[x] = 2
            gui_setbtnpixmap(aBtns[x],Player2EatPic)
            lEat = True
            nPlayer2Score++

        ok

        if (x != nPos) and (aStatus2[x] = 1) and

```

```

(aStatusValues2[x] = nValue or nValue=5 )
aStatus2[x] = 2
gui_setbtnpixmap(aBtns2[x],Player2EatPic)
lEat = True
nPlayer2Score++
                                ok
        next
        if lEat
                                nPlayer2Score++
                                gui_setbtnpixmap(aBtns2[nPos],Player2EatPic)
                                aStatus2[nPos] = 2
                                label2.setText("Player (2) - Score : " + nPlayer2Score)
                                ok
Func checknewgame
    if isnewgame()
                                lnewgame = true

                                if nPlayer1Score > nPlayer2Score
                                    label1.setText("Player (1) Wins!!!")
                                ok
                                if nPlayer2Score > nPlayer1Score
                                    label2.setText("Player (2) Wins!!!")
                                ok

                                appl.processEvents()
                                delay(nDelayNewGame)

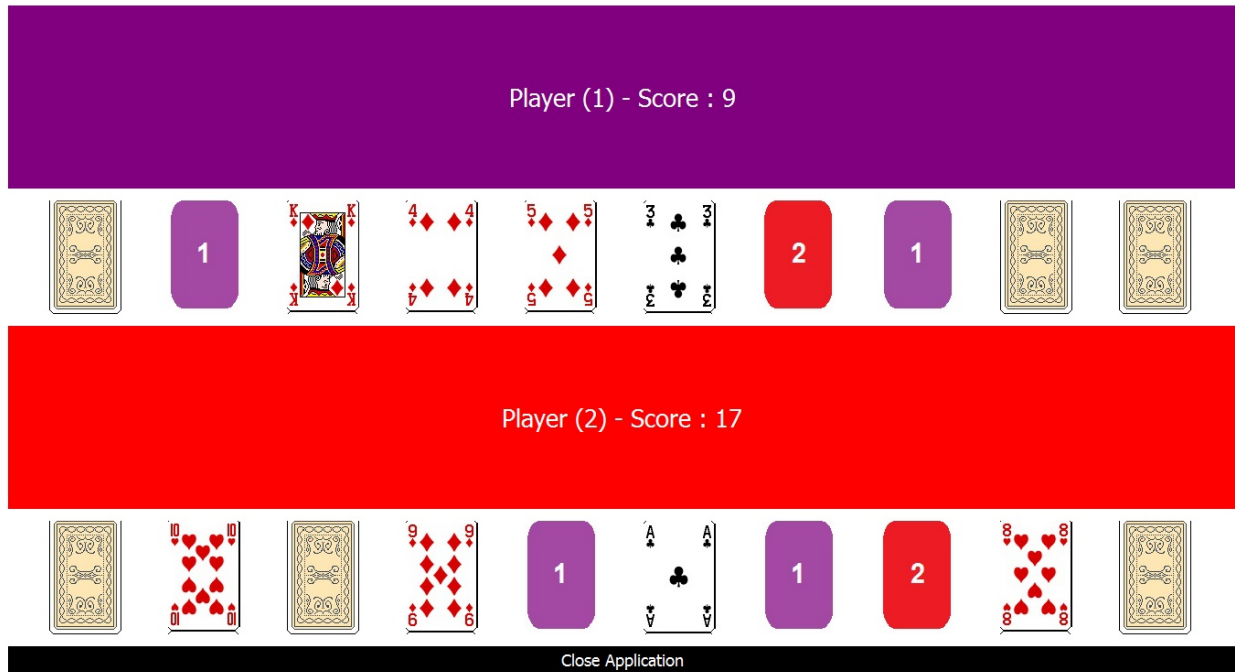
                                win1.delete()
                                appl.quit()

                                ok
Func isnewgame
    for t in aStatus
        if t = 0
                                return false
                                ok
    next
    for t in aStatus2
        if t = 0
                                return false
                                ok
    next
    return true

Func delay x
nTime = x * 1000
oTest = new QTest
oTest.qsleep(nTime)

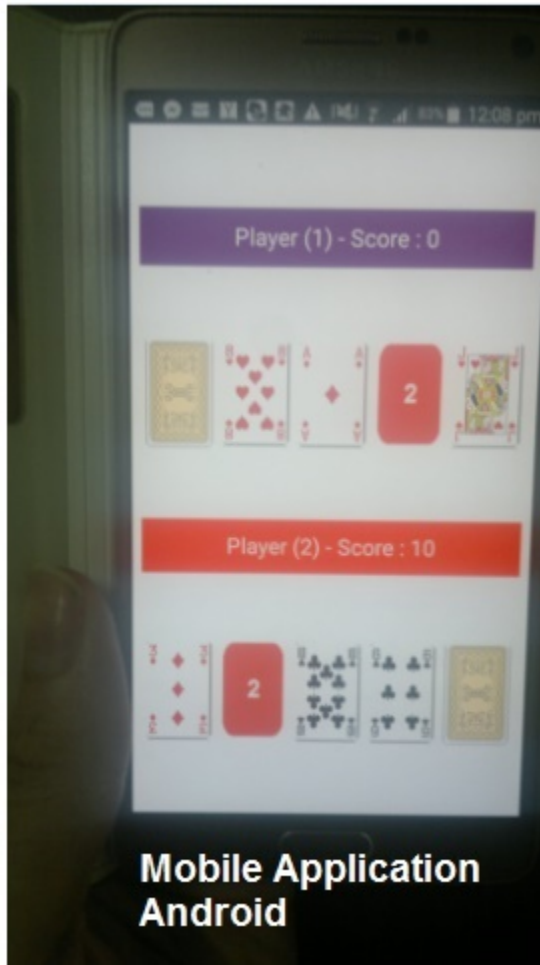
```

The application during the runtime



Note: in the previous screen shot the player get the card number '5' but his score is not increased because he opened this card while no other cards are visible!

The next screen shot while running the game using a Mobile (Android)



Note: using Qt we can run the same application on other Mobile systems

57.56 Classes and their Methods to use the default events

The next table present the class name and the methods that we have to use the default events.

Class Name	Methods to use the default Events
QPushButton	SetClickEvent()
QAction	SetClickEvent()
QLineEdit	SetTextChangedEvent()
	SetCursorPositionChangedEvent()
	SetEditingFinishedEvent()
	SetReturnPressedEvent()
	SetSelectionChangedEvent()
	SetTextEditedEvent()
QTextEdit	SetCopyAvailableEvent()
	SetCurrentCharFormatChangedEvent()
	SetCursorPositionChangedEvent()
	SetRedoAvailableEvent()
Continued on next page	

Table 57.1 – continued from previous page

Class Name	Methods to use the default Events
	SetSelectionChangedEvent()
	SetTextChangedEvent()
	SetUndoAvailableEvent()
QListWidget	SetCurrentItemChangedEvent()
	SetCurrentRowChangedEvent()
	SetCurrentTextChangedEvent()
	SetItemActivatedEvent()
	SetItemChangedEvent()
	SetItemClickedEvent()
	SetItemDoubleClickedEvent()
	SetItemEnteredEvent()
	SetItemPressedEvent()
	SetItemSelectionChangedEvent()
QTreeView	SetCollapseEvent()
	SetExpandedEvent()
	SetActivatedEvent()
	SetClickedEvent()
	SetDoubleClickedEvent()
	SetEnteredEvent()
	SetPressedEvent()
	SetViewportEnteredEvent()
QTreeWidget	SetCollapsedEvent()
	SetExpandedEvent()
	SetActivatedEvent()
	SetClickedEvent()
	SetDoubleClickedEvent()
	SetEnteredEvent()
	SetPressedEvent()
	SetViewportEnteredEvent()
	SetCurrentItemChangedEvent()
	SetItemActivatedEvent()
	SetItemChangedEvent()
	SetItemClickedEvent()
	SetItemCollapsedEvent()
	SetItemDoubleClickedEvent()
	SetItemEnteredEvent()
	SetItemExpandedEvent()
	SetItemPressedEvent()
	SetItemSelectionChangedEvent()
QComboBox	SetActivatedEvent()
	SetCurrentIndexChangedEvent()
	SetEditTextChangedEvent()
	SetHighlightedEvent()
QTabWidget	SetCurrentChangedEvent()
	SetTabCloseRequestedEvent()
QTableWidget	SetCellActivatedEvent()
	SetCellChangedEvent()
	SetCellClickedEvent()
	SetCellDoubleClickedEvent()

Continued on next page

Table 57.1 – continued from previous page

Class Name	Methods to use the default Events
	SetCellEnteredEvent()
	SetCellPressedEvent()
	SetCurrentCellChangedEvent()
	SetCurrentItemChangedEvent()
	SetItemActivatedEvent()
	SetItemChangedEvent()
	SetItemClickedEvent()
	SetItemDoubleClickedEvent()
	SetItemEnteredEvent()
	SetItemPressedEvent()
	SetItemSelectionChangedEvent()
QProgressBar	SetValueChangedEvent()
QSpinBox	SetValueChangedEvent()
QSlider	SetActionTriggeredEvent()
	SetRangeChangedEvent()
	SetSliderMovedEvent()
	SetSliderPressedEvent()
	SetSliderReleasedEvent()
	SetValueChangedEvent()
QDial	SetActionTriggeredEvent()
	SetRangeChangedEvent()
	SetSliderMovedEvent()
	SetSliderPressedEvent()
	SetSliderReleasedEvent()
	SetValueChangedEvent()
QWebView	SetLoadFinishedEvent()
	SetLoadProgressEvent()
	SetLoadStartedEvent()
	SetSelectionChangedEvent()
	SetTitleChangedEvent()
	SetUrlChangedEvent()
QCheckBox	SetStateChangedEvent()
	SetClickedEvent()
	SetPressedEvent()
	SetReleasedEvent()
	SetToggledEvent()
QRadioButton	SetClickedEvent()
	SetPressedEvent()
	SetReleasedEvent()
	SetToggledEvent()
QButtonGroup	SetButtonClickedEvent()
	SetButtonPressedEvent()
	SetButtonReleasedEvent()
QVideoWidget	SetBrightnessChangedEvent()
	SetContrastChangedEvent()
	SetFullScreenChangedEvent()
	SetHueChangedEvent()
	SetSaturationChangedEvent()
QTimer	SetTimeoutEvent()

Continued on next page

Table 57.1 – continued from previous page

Class Name	Methods to use the default Events
QTcpServer	SetAcceptErrorEvent()
	SetNewConnectionEvent()
QIODevice	SetAboutToCloseEvent()
	SetBytesWrittenEvent()
	SetReadChannelFinishedEvent()
	SetReadyReadEvent()
QAbstractSocket	SetConnectedEvent()
	SetDisconnectedEvent()
	SetErrorEvent()
	SetHostFoundEvent()
	SetProxyAuthenticationRequiredEvent()
	SetStateChangedEvent()
QTcpSocket	SetConnectedEvent()
	SetDisconnectedEvent()
	SetErrorEvent()
	SetHostFoundEvent()
	SetProxyAuthenticationRequiredEvent()
	SetStateChangedEvent()
	SetAboutToCloseEvent()
	SetBytesWrittenEvent()
	SetReadChannelFinishedEvent()
	SetReadyReadEvent()
QColorDialog	SetColorSelectedEvent()
	SetCurrentColorChangedEvent()
QNetworkAccessManager	SetFinishedEvent()
QThread	SetStartedEvent()
	SetFinishedEvent()

57.57 Methods to use Events with Events Filter

RingQt define a new class called QAllEvents that help you in using Events Filter

The next table presents the methods that we have

Methods to get parameters	Class Name
getKeyCode() → Number	QAllEvents
getx() → Number	
gety() → Number	
getglobalx() → Number	
getglobaly() → Number	
getbutton() → Number	
getbuttons() → Number	

The next table presents the methods that we have to use events.

Method Name	Class Name
setKeyPressEvent(cEvent)	QAllEvents
setMouseButtonPressEvent(cEvent)	
setMouseButtonReleaseEvent(cEvent)	
setMouseButtonDbClickEvent(cEvent)	
setMouseMoveEvent(cEvent)	
setCloseEvent(cEvent)	
setContextMenuEvent(cEvent)	
setDragEnterEvent(cEvent)	
setDragLeaveEvent(cEvent)	
setDragMoveEvent(cEvent)	
setDropEvent(cEvent)	
setEnterEvent(cEvent)	
setFocusInEvent(cEvent)	
setFocusOutEvent(cEvent)	
setKeyReleaseEvent(cEvent)	
setLeaveEvent(cEvent)	
setNonClientAreaMouseButtonDbClickEvent(cEvent)	
setNonClientAreaMouseButtonPressEvent(cEvent)	
setNonClientAreaMouseButtonReleaseEvent(cEvent)	
setNonClientAreaMouseMoveEvent(cEvent)	
setMoveEvent(cEvent)	
setResizeEvent(cEvent)	
setWindowActivateEvent(cEvent)	
setWindowBlockedEvent(cEvent)	
setWindowDeactivateEvent(cEvent)	
setWindowStateChangeEvent(cEvent)	
setWindowUnblockedEvent(cEvent)	

57.58 The Difference between Qt and RingQt

1. RingQt use simple methods to set the code that will be executed for events.

Syntax:

```
Set<Event_Name>Event (cEventCode)
```

2. RingQt change the name of some methods to avoid conflict with Ring Keywords.

The next table present these little changes

Class Name	Qt Method Name	RingQt Method Name
QWebView	load	loadpage
QMediaPlaylist	load	loadfile
QMediaPlaylist	next	movenext
QPainter	end	endpaint
QPicture	load	loadfile
QLineEdit	end	endtext
QDialog	done	donedialog
QTextDocument	end	enddoc
QTextBlock	next	nextblock
QSqlQuery	next	movenext
QImage	load	loadimage
QNetworkAccessManager	get	getvalue
QNetworkAccessManager	put	putvalue
QThread	exit	exitfromthread
QRegularExpressionMatchIterator	next	nextitem
QCamera	load	loadcamera

57.59 RingQt Classes and their Qt Documentation

Qt Documentation : <http://doc.qt.io/qt-5/classes.html>

See the “RingQt Classes and Methods Reference” chapter for supported classes and methods.

57.60 New Classes names - Index Start from 1

We added new classes to RingQt - another version of classes where the class names doesn't start with the “q” letter
Also updated methods so the index start from 1 when we deal with the GUI controls like

- ComboBox
- ListWidget
- TableWidget
- TreeWidget

These classes are inside guilib.ring under the package name : System.GUI

To use it

```
load "guilib.ring"
import System.GUI
```

This doesn't have any effect on our previous code, It's just another choice for better code that is consistent with Ring rules.

Also the form designer is updated to provide us the choice between using classes where (index start from 0) or (index start from 1)

Example (Uses the Form Designer)

1. <https://github.com/ring-lang/ring/blob/master/applications/formdesigner/tests/indexstart/indexstartView.ring>
2. <https://github.com/ring-lang/ring/blob/master/applications/formdesigner/tests/indexstart/indexstartController.ring>

57.61 Creating Reports using the WebLib and the GUIlib

The WebLib comes with a class called HtmlPage

Using this class we can create reports quickly using WebLib & GUIlib together

Example:

```
load "stdlib.ring"
load "weblib.ring"
load "guilib.ring"

import System.Web
import System.GUI

new qApp {
    open_window(:CustomersReportController)
    exec()
}

class CustomersReportController

    oView = new CustomersReportView

    func Start
        CreateReport()

    func CreateReport
        mypage = new HtmlPage {
            h1 { text("Customers Report") }
            Table
            {
                style = stylewidth("100%") + stylegradient(4)
                TR
                {
                    TD { WIDTH="10%"
                        text("Customers Count : " ) }
                    TD { text(100) }
                }
            }
            Table
            {
                style = stylewidth("100%") + stylegradient(26)
                TR
                {
                    style = stylewidth("100%") +
                        stylegradient(24)
                    TD { text("Name " ) }
                    TD { text("Age" ) }
                    TD { text("Country" ) }
                    TD { text("Job" ) }
                    TD { text("Company" ) }
                }
                for x = 1 to 100
                    TR
                    {
                        TD { text("Test" ) }
                        TD { text("30" ) }
                        TD { text("Egypt" ) }
```

```

                                TD { text("Sales" ) }
                                TD { text("Future" ) }
                                }
                                next
                                }
                                }
                                write("report.html",mypage.output())

func PrintEvent
    printer1 = new qPrinter(0) {
        setoutputformat(1)
        setoutputfilename("report.pdf")
    }
    oView {
        web.print(printer1)
        web.show()
    }
    system ("report.pdf")

class CustomersReportView

    win = new window() {
        setwindowtitle("Report Window")
        setgeometry(100,100,500,500)
        web = new webview(win) {
            setgeometry(100,100,1000,500)
            loadpage(new qurl("file:///"+
                currentdir()+"/report.html"))
        }
        new pushbutton(win) {
            setGeometry(100,20,100,30)
            setText("Print")
            setclickevent(Method(:PrintEvent))
        }
        showMaximized()
    }

```

Screen Shot:

BUILDING RINGQT APPLICATIONS FOR MOBILE

In this chapter we will learn about Building RingQt Applications for Mobile.

58.1 Download Requirements

Check the next link : <http://doc.qt.io/qt-5/androidgs.html>

Download

- The Android SDK Tools
<https://developer.android.com/studio/index.html>
- The Android NDK (Tested using android-ndk-r10c)
<https://developer.android.com/ndk/index.html>
- Apache Ant v1.8 or later
<http://ant.apache.org/bindownload.cgi>
- Java SE Development Kit (JDK) v6 or later
<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>

58.2 Update the Android SDK

Update the Android SDK to get the API and tools packages required for development

Tested using Android 4.4.2 (API 19)

- In Windows - Define the next Environment Variables based on your system.

1. JAVA_HOME

For Example : C:\Program Files (x86)\Java\jdk1.8.0_05
--

2. ANDROID_HOME

For Example : B:\mahmoud\Tools\Java-Android\adt-bundle-windows-x86-20140702\sdk
--

58.3 Install Qt for Android

- You can install Qt for Android from the next link

<https://download.qt.io/archive/qt/5.5/5.5.1/>

- **Run Qt Creator, Select Tools > Options > Android to add the** Android NDK and SDK paths.

<http://doc.qt.io/qtcreator/creator-developing-android.html>

- Using Qt Creator Open the project

Folder : ring/android/ringqt/project

Project file : project.pro

- Using Qt Creator, You will find the compiled Ring application in resources/ringapp.ringo

This file (Ring Object File) is generated by the Ring compiler using

```
ring ringapp.ring -go -norun
```

- To run your application instead of the default application

1. Using Qt Creator, Add your application images to resources

Or You can use any text editor (Notepad) and modify : project.qrc

2. To find images from your Ring application, You need to use the file name in resources

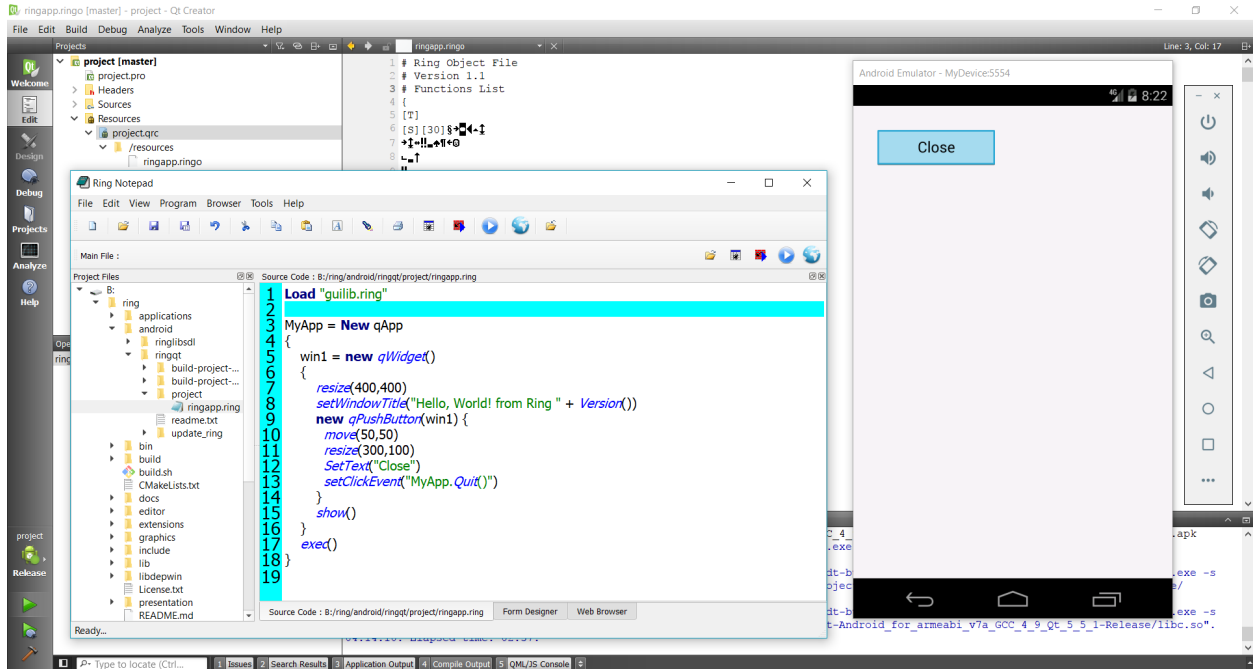
Example

```
if isandroid()
    mypic = new QPixmap(":/cards.jpg")
else
    mypic = new QPixmap("cards.jpg")
ok
```

3. In your Ring application folder (Using the command prompt)

```
ring myapp.ring -go -norun
```

4. Add your file to the project project/myapp.ringo
5. Update main.cpp and project.qrc and replace ringapp.ringo with myapp.ringo
6. Build and Run your Application using Qt Creator



58.4 Comments about developing for Android using RingQt

1. The main project file is main.cpp

This file load Ring Compiler/Virtual Machine and RingQt

Then copy the Ring Object File during the runtime from the resources to temp. folder

Then run the Ring Object File (ringapp.ringo) using the Ring VM

Through main.cpp you can extract more files from the resources to temp. folder once you add them (create projects with many files).

2. The next functions are missing from this Ring edition

- Database (ODBC, SQLite & MySQL)
- Security and Internet functions (LibCurl & OpenSSL)
- RingAllegro (Allegro Library)
- RingLibSDL (LibSDL Library)

Just use Qt Classes through RingQt.

For database access use the QSqlDatabase Class

Note: All of the missing libraries ((LibCurl, OpenSSL & Allegro) can be compiled for Android, but they are not included in this Qt project.

3. use if isandroid() when you want to modify the code just for android

Example:

```
if isandroid()
    // Android code
```

```
else
    // other platforms
ok
```

(4) Sometimes you will find that the button text/image is repeated in drawing ! it's Qt problem that you can avoid using the next code.

```
if isandroid()
    setStyleSheet ("
        border-style: outset;
        border-width: 2px;
        border-radius: 4px;
        border-color: black;
        padding: 6px;")
ok
```

5. Always use Layouts instead of manual setting of controls position and size.

This is the best way to get the expected user interface to avoid problems like (controls with small/extra size)

6. When you deal with Qt Classes you can determine the images from resources (you don't need to copy them using main.cpp)

Example:

```
if isandroid()
    mypic = new QPixmap(":/cards.jpg")
else
    mypic = new QPixmap("cards.jpg")
ok
```

Now RingQt comes with the AppFile() function to determine the file name

Example:

```
mypic = new QPixmap(AppFile("cards.jpg")) # Desktop or Android
```

58.5 Using Ring2EXE

Starting from Ring 1.6 we can use Ring2EXE to quickly prepare Qt project for our application

Example:

```
ring2exe myapp.ring -dist -mobileqt
```

OBJECTS LIBRARY FOR RINGQT APPLICATION

In this chapter we will learn about the objects library for RingQt applications.

Ring comes with the Objects library for RingQt applications. Instead of using global variables for windows objects and connecting events to objects using the object name, the Objects Library will manage the GUI objects and will provide a more natural API to quickly create one or many windows from the same class and the library provide a way to quickly set methods to be executed when an event is fired. Also the library provide a natural interface to quickly use the parent or the caller windows from the child or sub windows.

The Objects Library is designed to be used with the MVC Design Pattern.

The Objects Library is merged in RingQt so you can use it directly when you use RingQt

59.1 Library Usage

- Use the `Open_Window(cWindowControllerClassName)` function to open new Windows
- Create at least Two Classes for each window, The Controller Class and the View Class
- Create each controller class from the `WindowsControllerParent` Class
- Create each view class from the `WindowsViewParent` Class
- Use the `Last_Window()` function to get the object of the last window created (The Controller object).
- When you call a sub window, use the `SetParentObject()` method and pass the self object.
- In the View Class, To determine the event method use the `Method(cMethodName)` function.
- The `Method(cMethodName)` function determine the method in the controller class that will be executed.
- Each controller class contains by default the `CloseAction()` method that you can call to close the window.
- You don't need to call the `Show()` Method for each window, When you use `Open_Window()` It will be called.
- In the view class, Define the GUI window object as an attribute called `win`.
- You can use `Open_WindowNoShow()` to avoid displaying the window.
- You can use `Open_WindowAndLink()` to quickly get methods to access the windows.

59.2 Example

In the next example we will create two types of windows.

- Main Window contains a button. When the user click on the button a sub window will be opened.

- The User Can click on the button many times to open many sub windows.
- Each Sub Window contains Two buttons.
- The first button in the sub window change the Main and the Sub Windows Titles.
- The second button in the sub window close the Sub Window.

```
load "guilib.ring"

new qApp {
    open_window( :MainWindowController )
    exec()
}

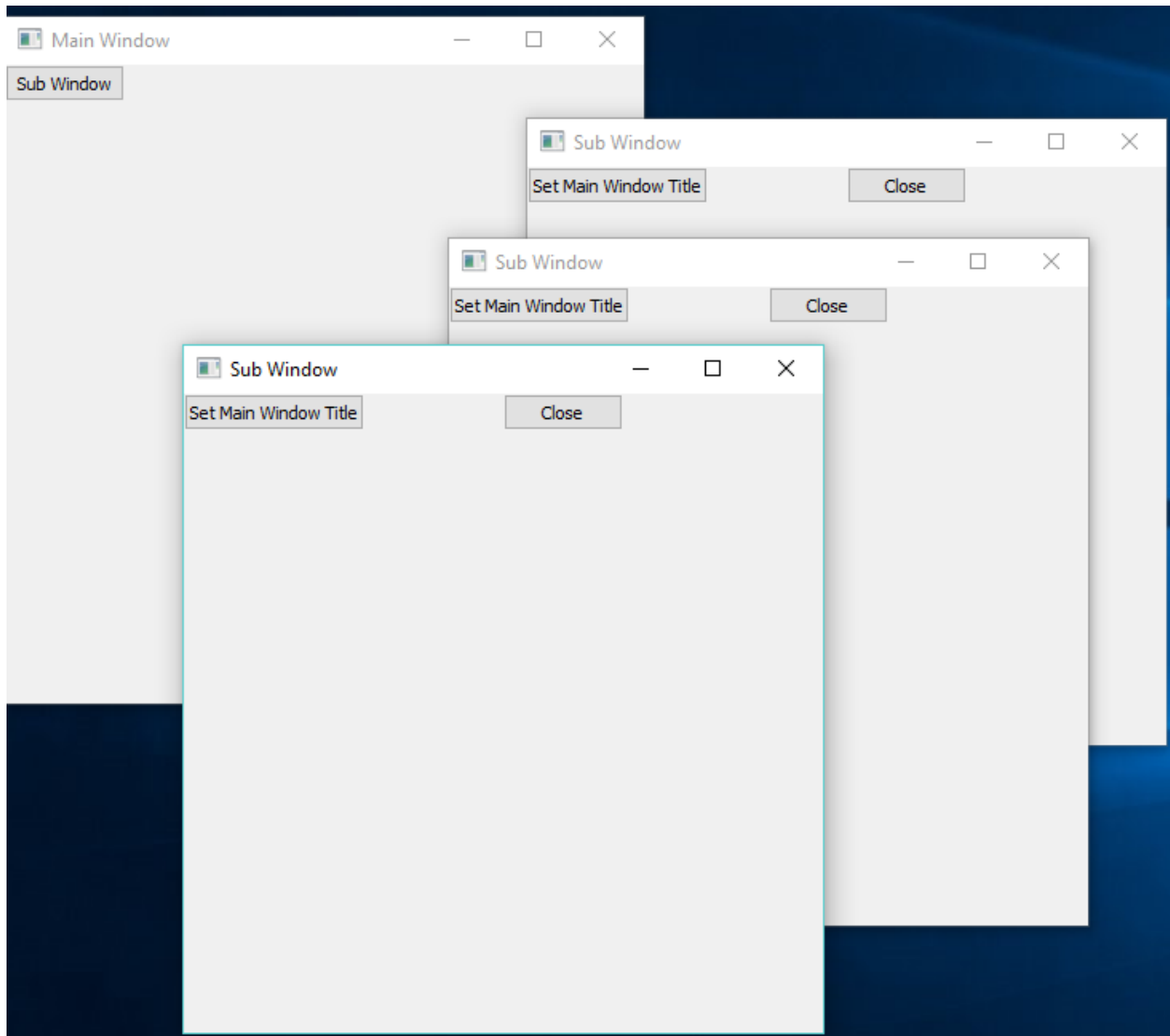
class MainWindowController from WindowsControllerParent
    oView = new MainWindowView
    func SubWindowAction
        Open_window( :SubWindowController )
        Last_Window().SetParentObject(self)

class MainWindowView from WindowsViewParent
    win = new QWidget() {
        SetWindowTitle("Main Window")
        btnSub = new QPushButton(win) {
            setText("Sub Window")
            setClickEvent( Method( :SubWindowAction ) )
        }
        resize(400,400)
    }

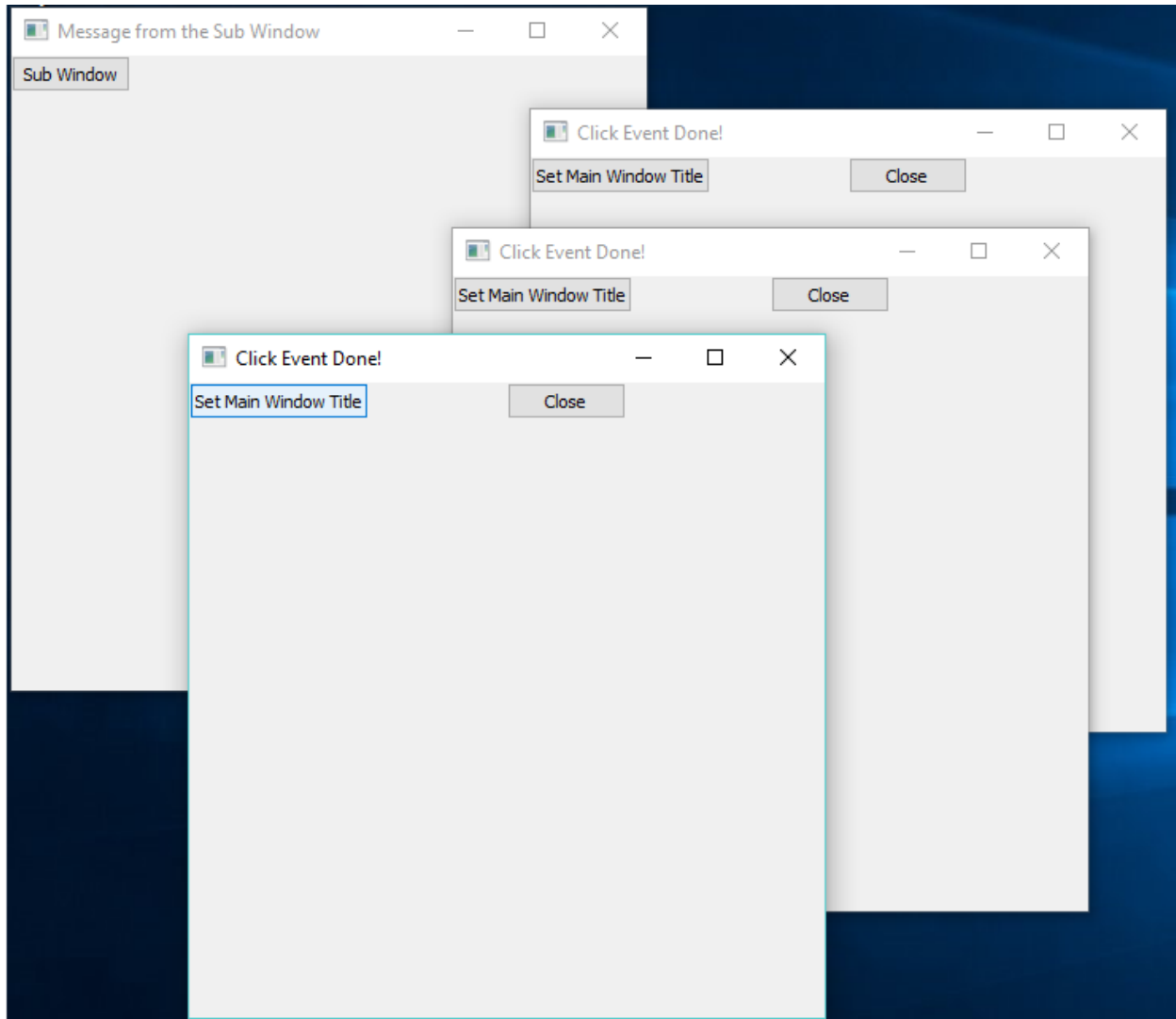
class SubWindowController from WindowsControllerParent
    oView = new SubWindowView
    func SetMainWindowTitleAction
        Parent().oView.win.SetWindowTitle("Message from the Sub Window")
        oView.win.SetWindowTitle("Click Event Done!")

class SubWindowView from WindowsViewParent
    win = new QWidget() {
        SetWindowTitle("Sub Window")
        btnMsg = new QPushButton(win) {
            setText("Set Main Window Title")
            setClickEvent( Method( :SetMainWindowTitleAction ) )
        }
        btnClose = new QPushButton(win) {
            Move(200,0)
            setText("Close")
            setClickEvent( Method( :CloseAction ) )
        }
        resize(400,400)
    }
}
```

The next screen shot after creating three sub windows.



The next screen shot after clicking on the button in each sub window.



59.3 Open_WindowAndLink() Function

We can use the `Open_WindowAndLink()` function to connect between the application windows, pass messages (call methods) between the objects.

This function uses Meta-programming to define dynamic methods in the Caller Class to use the dynamic objects of other windows that we create.

Example : (Uses the Form Designer)

First Window

1. <https://github.com/ring-lang/ring/blob/master/applications/formdesigner/tests/twowindowpart5/firstwindowView.ring>
2. <https://github.com/ring-lang/ring/blob/master/applications/formdesigner/tests/twowindowpart5/firstwindowController.ring>

Second Window

1. <https://github.com/ring-lang/ring/blob/master/applications/formdesigner/tests/twowindowpart5/secondwindowView.ring>
2. <https://github.com/ring-lang/ring/blob/master/applications/formdesigner/tests/twowindowpart5/secondwindowController.ring>

In the next code for example (from FirstWindowController.ring)

The `Open_WindowAndLink()` will create an object from the `SecondWindowController` Class

Then will add the Method : `SecondWindow()`, `IsSecondWindow()` Methods to the `FirstWindowController` Class

Also will add the Method : `FirstWindow()`, `IsFirstWindow()` Methods to the `SecondWindowController` Class

So the `SendMessage()` method in `FirstWindowController` class can use the `SecondWindow()` method to access the object.

This is more simple than using `Last_Window()`, `Parent()` and `SetParentObject()` methods.

```
class firstwindowController from windowsControllerParent

    oView = new firstwindowView

    func OpenSecondWindow
        Open_WindowAndLink(:SecondWindowController,self)

    func SendMessage
        if IsSecondWindow()
            SecondWindow().setMessage("Message from the first window")
        ok

    func setMessage cMessage
        oView.Label1.setText(cMessage)
```

59.4 Open_WindowInPackages() Function

The `Open_WindowInPackages()` function is the same as `Open_Window()` but takes an extra list that determine the packages to import before opening the window.

Syntax:

```
Open_WindowInPackages(cClassName,aPackagesList)
```

Example:

The next example from the Form Designer source code, Open the Window Flags window using the `open_windowInPackages()` function.

We determine the class name “`WindowFlagsController`” and the packages name.

The Window Flags window uses the `FormDesigner` and `System.GUI` packages.

```
open_windowInPackages(:WindowFlagsController,[
    "formdesigner",
    "System.GUI"
])
```

59.5 Objects Library Source Code

The library source code is very simple, You can check the source code files

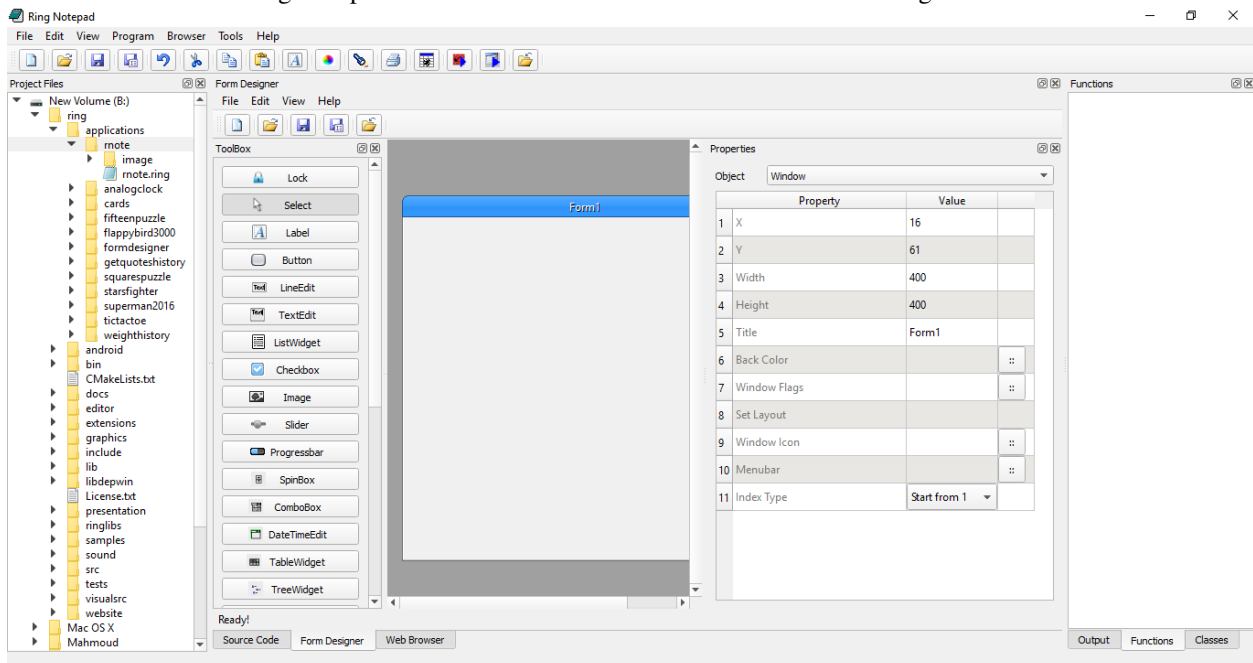
- <https://github.com/ring-lang/ring/blob/master/extensions/ringqt/objectslib/objects.ring>
- <https://github.com/ring-lang/ring/blob/master/extensions/ringqt/objectslib/subwindows.ring>

USING THE FORM DESIGNER

In this chapter we will learn about using the Form Designer.

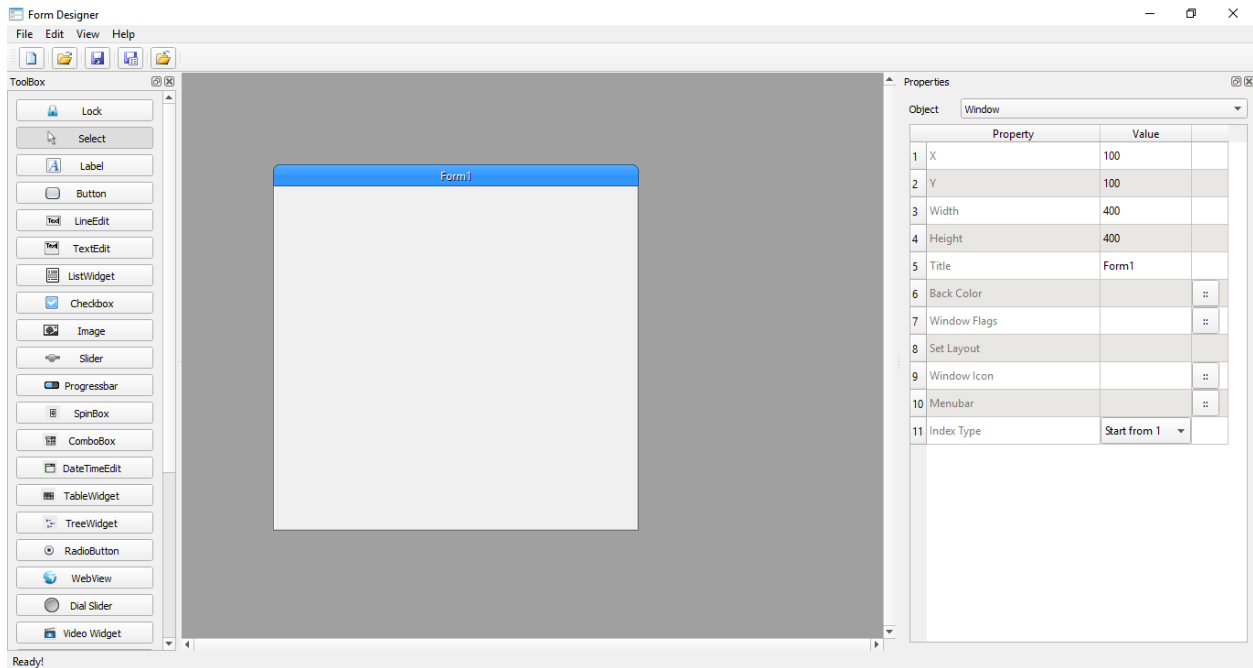
We can run the From Designer from Ring Notepad

From the Menubar in Ring Notepad - View Menu - We can Show/Hide the Form Designer window.



Also we can run the Form Designer in another window.

From the Ring Notepad - Tools Menu - Select the Form Designer.



60.1 The Designer Windows

- **Toolbox** : To select controls to be added to the window.
- **Properties** : To set the properties of the active window or controls.
- **Design Region** : To select, move and resize the window and the controls.

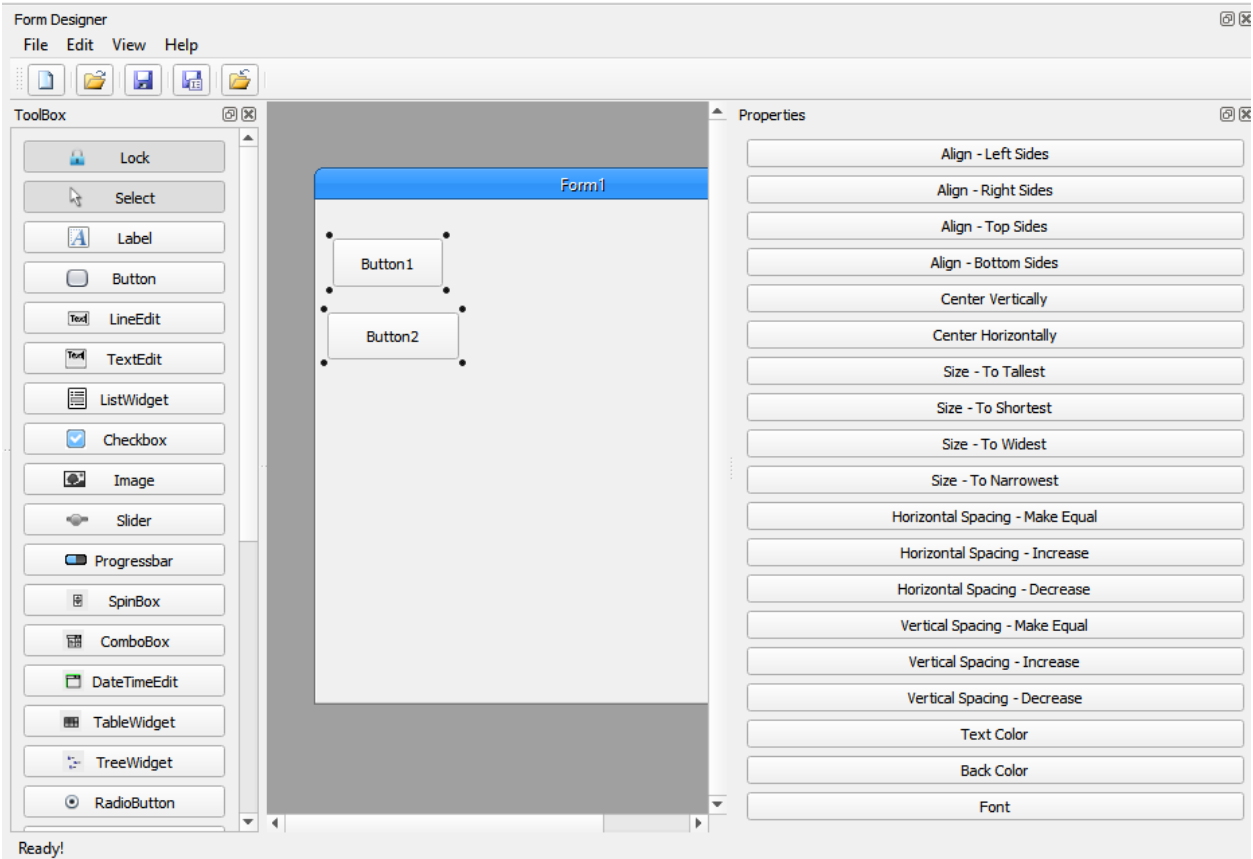
60.2 The Toolbox

We have many buttons.

- **Lock** : We can use it to draw many controls of the same type quickly.
- **Select** : We can use it to select a control in the Design Region
- **Controls Buttons** : Select a control to be added to the window.

60.3 The Properties

- When we select the window or one control, We will have the selected object properties.
- Also In the properties window we have a combobox to select the active control.
- Some properties provide a button next to the property value. We can click on the button to get more options.
- When we select more than one control, We will have options for multi-selection



60.4 Running Forms

When we save the form file (*.rform), The Form Designer will create two Ring files

- The Controller Class
- The View Class

For example, if the form file is helloworld.rform

The form designer will generate two files

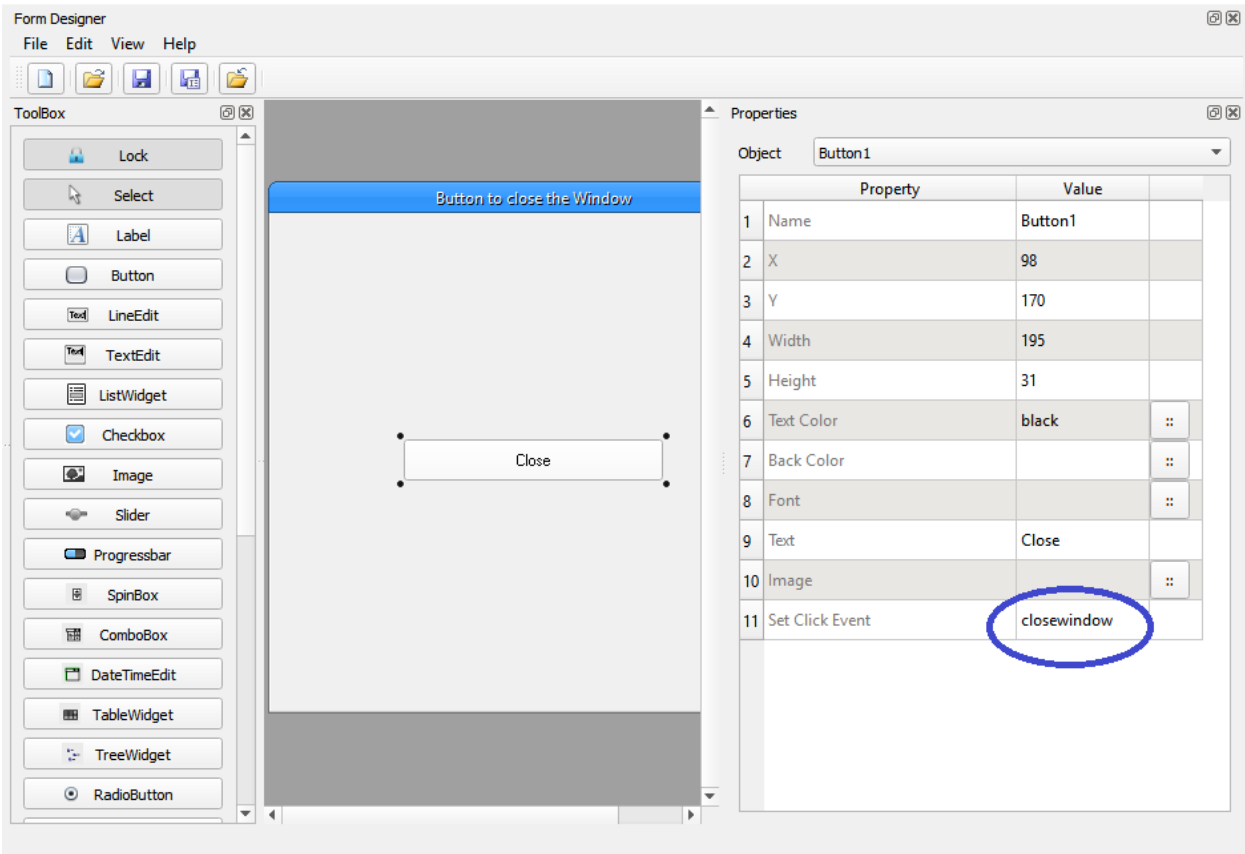
- helloworldcontroller.ring
- helloworldview.ring

To run the program, Open the controller class file then click the Run button (CTRL+F5)

Tip: When you open a form using Ring Notepad, the controller class will be opened automatically, So we can press (CTRL+F5) or click on the Run button while the form designer window is active.

60.5 Events Code

1. Just type the method name in the event property.



(2) Then write the method code in the controller class.

```

Source Code : B:/ring/applications/formdesigner/tests/buttontoclosethewindow/buttontoclosethewindowController.ring
1  # Form/Window Controller - Source Code File
2
3  load "buttontoclosethewindowView.ring"
4
5  if IsMainSourceFile() {
6      new qApp {
7          StyleFusion()
8          open_window(:buttontoclosethewindowController)
9          exec()
10     }
11 }
12
13 class buttontoclosethewindowController from windowsControllerParent
14
15     oView = new buttontoclosethewindowView
16
17     func CloseWindow
18         oView.win.close()
19
20

```

In this example we write

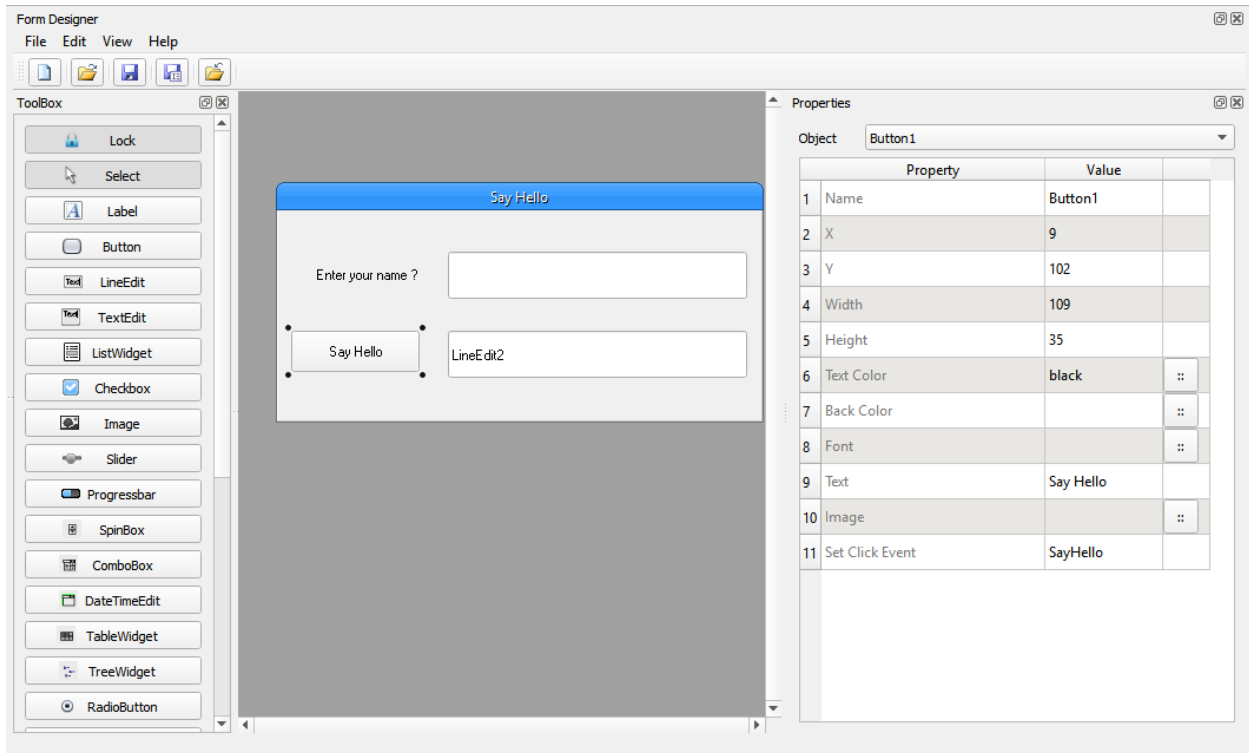
```

func CloseWindow
    oView.win.close()

```

Where inside the controller class, We uses the oView object to access the form.

Another Example :



The Event Code

```
func SayHello
    oView {
        LineEdit2.setText("Hello " + LineEdit1.text() )
    }
```

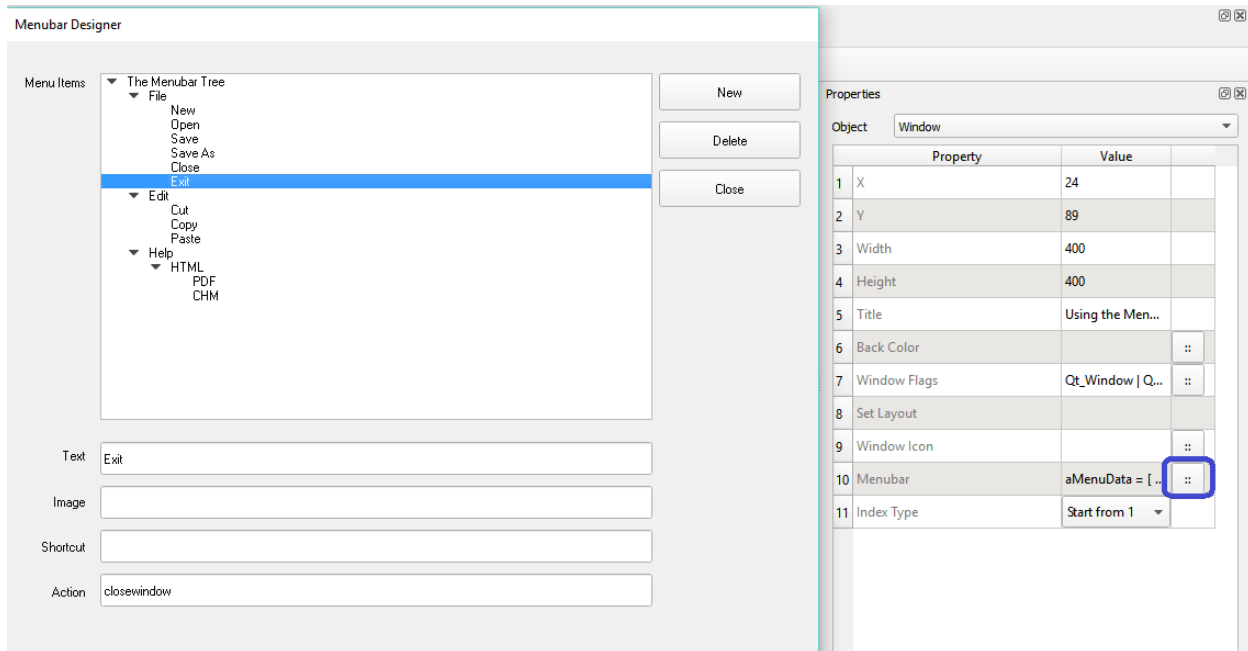
60.6 Keyboard Shortcuts

After selecting one or group of controls

- Use the Arrows (Up, Down, Left and Right) to move them around.
- Shift + the Arrows (Up, Down, Left and Right) to Resize the controls.
- Del button to delete the controls.
- CTRL+SHIFT+V to Duplicate the controls.

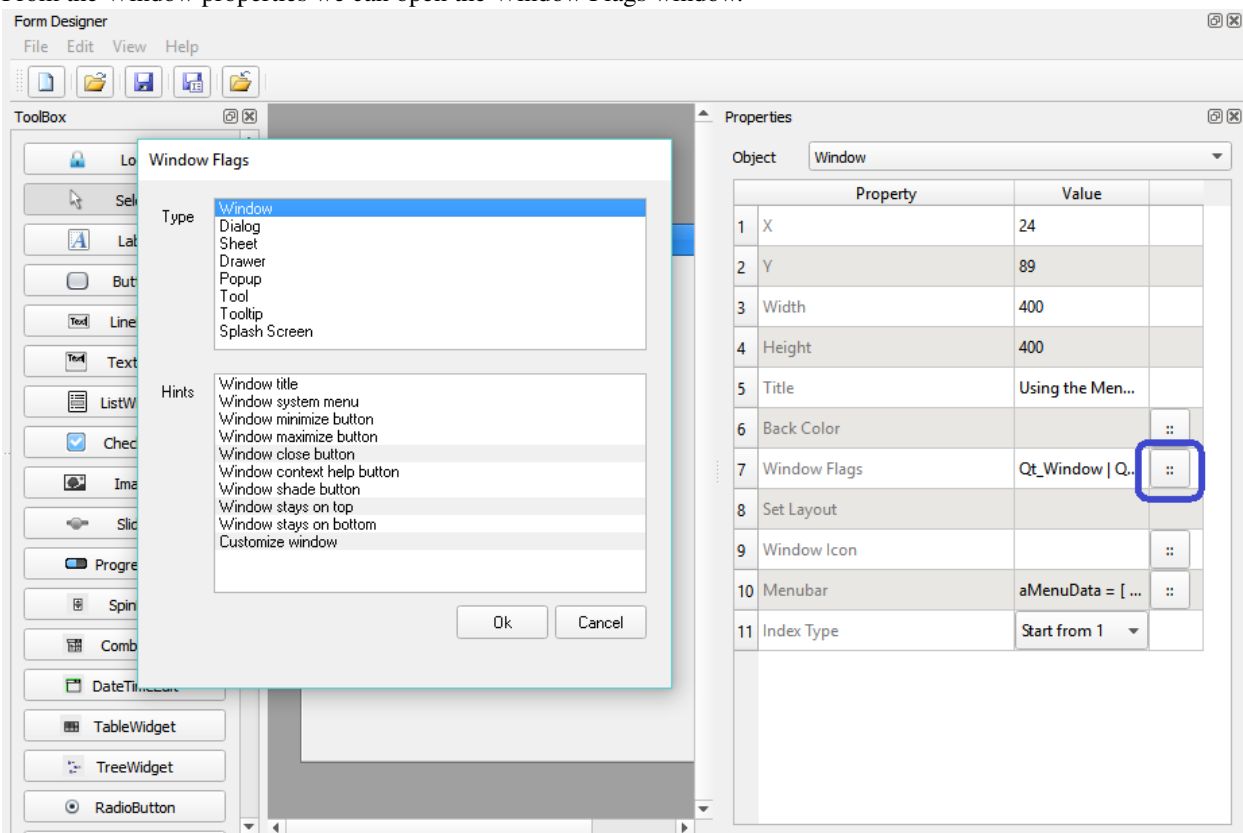
60.7 Menubar Designer

From the Window properties we can open the Menubar Designer



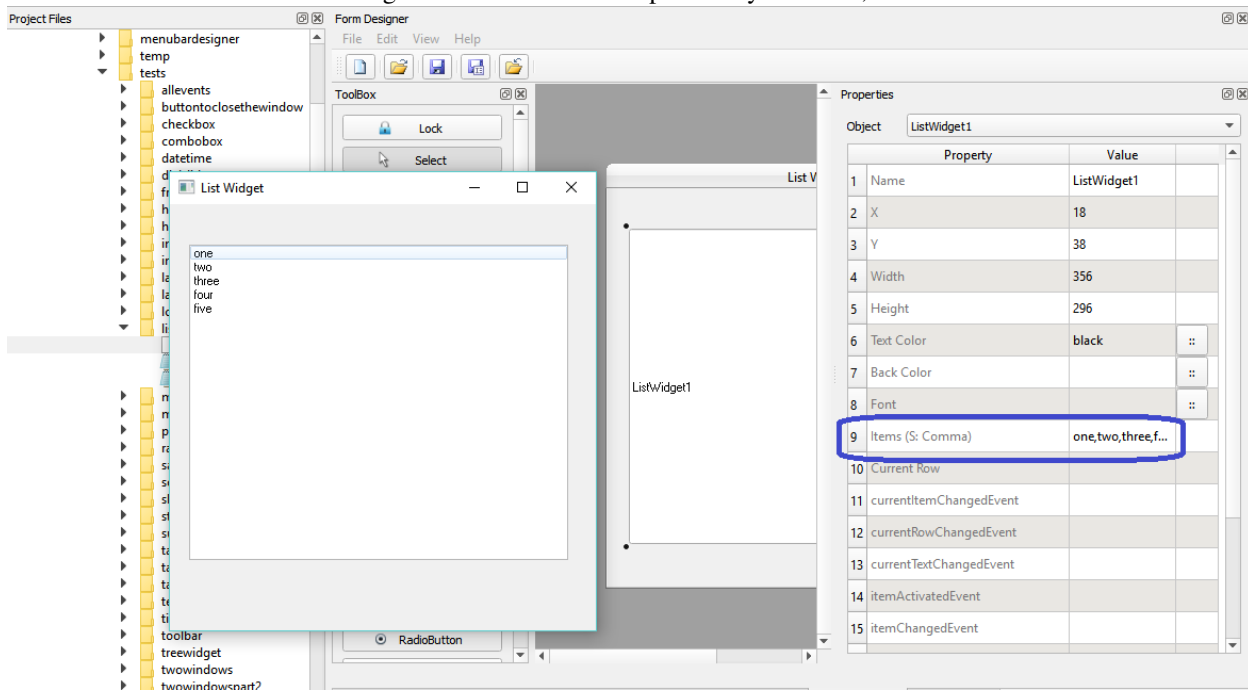
60.8 Window Flags

From the Window properties we can open the Window Flags window.



60.9 Entering Items

For some controls like the List Widget we can enter items separated by comma ‘,’



60.10 Using Layouts

1. To use layouts, At first add the layout control to the window.
2. Use the window “Set Layout” property to determine the main layout.
3. From the layout properties determine the controls and the layout type.

60.11 More Samples and Tests

Check the folder : ring/applications/formdesigner/tests

Online : <https://github.com/ring-lang/ring/tree/master/applications/formdesigner/tests>

SCOPE RULES FOR VARIABLES AND ATTRIBUTES

In this chapter we will learn about scope rules and how Ring find variables.

Also we will learn about conflicts and how to solve/avoid them.

The next information are important once you start developing large applications using Ring

These application may uses

- Global variables (Try to avoid them)
- Classes (Object-Oriented)
- braces { } to access objects
- Declarative Programming
- Natural Programming

61.1 Three Scopes

In Ring we have three scopes :-

1. Public/Global Scope - Each variable you define in the statements part (before functions and classes)
2. Object Scope - When you are inside an object (Inside class method or using { } to access the object)
3. Local Scope - Related to functions and methods

61.2 Defining Variables and Variables Access

1. Ring uses lexical scoping, i.e. the scope of the variable is based on where we defined the variable.
2. Inside braces { } when you access an object, You will change the current active object scope to this object scope but you still can access the global scope and the local scope.
3. After the 'Class' keyword and the class name, when you write variable names to be defined as attributes, You still can access the global scope.

In this region (class region - after the class name and before methods) we have

- Global Scope —> The Global Scope
- Object Scope —> The Object Scope
- Local Scope —> The Object Scope

Note: Since the local scope in the class region point also to the object scope in this region, we can use nested braces and still have access to the object scope of the class through the local scope.

Tip: You can create windows and controls as attributes by defining them in this region.

Tip: In the class region if you created objects and used braces { } to access them then using self.attribute inside braces will use the class (not the object that you access) because you have access to the class through the local scope.

4. Function Parameters are automatically defined in the local scope.

61.3 How Ring find the variable?

1 - Search First in the Local Scope

if not found !

2 - Search in the Object Scope

if not found !

3 - Search in the public scope

if not found —> Runtime Error

if found —> Check if we can do optimization to avoid searching next time (Cache / Pointers for performance).

61.4 Using Object.Attribute

When we use object.attribute the search will be in the object attributes only.

I.e. no search will be done in the local scope or in the global scope for the object attribute.

Note: Using self.attribute will search for the first self before searching for attributes.

61.5 The Self Object

The self object is a reference to the current object that we can use from the class methods.

When we are inside class method and use Self we mean the object that will be created from this class.

Inside the class methods if we used Braces { } this will change the current object scope and self will be changed also inside braces to reference the object that we access using Braces.

Inside the Class Region (after the class name and before any method) we have access to the object through the object scope and the local scope also. In this region using Self will always be a reference to the class object. if we used Braces to change the object scope then used Self inside Braces, Also self will be a reference to the class object (not the object that we already access using braces) because in the class region we have :-

- Global Scope —> Global Scope
- Object Scope —> Object Scope

- Local Scope —> Object Scope

And using Braces changes the object scope only (not the local scope) and when Ring search for variables it will search in the Local Scope first so it will find self in the class that we are inside.

61.6 How Ring Define Variables and Attributes

Ring will use the variable name in the Assignment operation

1 - Search using the variable name

2 - If not found —> Avoid the runtime error and define the variable in the current scope

3 - If found —> Use the variable and don't define anything in the current scope

- In the global region (before any function or class) the current scope is the global scope.
- In the class region (after the class name and before any method) the current scope is the object attributes.
- In Functions and methods the current scope is the local scope.

61.7 Conflict between Global Variables and Class Attributes

Look at this example:

```
name = "test"
o1 = new person
see o1

class person
  name
  address
  phone
```

In the previous example we have a global variable called 'name' inside the class person.

when we use the variable 'name', Ring will start the search operation and will try to find it.

if found —> Use it

if not found —> Define new attribute

But the variable name is a global variable, so it will be found and used!

We will not have the attribute name! added to the object.

Solution (1) - Use the Main Function

```
func main
  name = "test"
  o1 = new person
  see o1

class person
  name
  address
  phone
```

Solution (2) - Use special mark for global variable names like \$

```
$name = "test"
o1 = new person
see o1

class person
    name
    address
    phone
```

Solution (3) - Use the AddAttribute() Method

```
name = "test"
o1 = new person
see o1

class person
    AddAttribute(self, "name")
    address
    phone
```

Solution (4) - Use self before the attribute name

```
name = "test"
o1 = new person
see o1

class person
    self.name
    address
    phone
```

So what is the best solution to this conflict?

- 1 - Use the \$ Mark for global variables
- 2 - Optional : Try to avoid global variables and use the Main function

In practice i do both of them.

The other solution

- Use self before the attribute name or use AddAttribute()

61.8 Conflict between Class Attributes and Local Variables

This conflict may happen when we access the object using braces

Example:

```
func main
    name = "nice"
    o1 = new person {name="mahmoud" address="Egypt" phone = 000 }
    see o1

class person
    name
    address
    phone
```

In the previous example we have the local variable name.

The value of this variable will be set to “mahmoud” instead of the object attribute.

Solution (1) : Just use Self

```
func main
    name = "nice"
    o1 = new person {self.name="mahmoud" address="Egypt" phone = 000 }
    see o1

class person
    name
    address
    phone
```

Solution (2) : Change the Local variable name

```
func main
    cName = "nice"
    o1 = new person {name="mahmoud" address="Egypt" phone = 000 }
    see o1

class person
    name
    address
    phone
```

Solution (3) : Change Braces and use the Dot operator

```
func main
    name = "nice"
    o1 = new person
    o1.name = "mahmoud"
    o1.address = "Egypt"
    o1.phone = 000
    see o1

class person
    name
    address
    phone
```

61.9 Using Braces to access objects inside Class Methods

Remember that we have Three scopes (Local Scope, Object Scope and Global Scope) and when we are inside a class method, we expect that we have access to the object attributes and methods and this is true until we use braces to access another object attributes and methods because in this case our object scope will be switched to another object.

```
new point { test() }

class point
    x=10 y=20
    func test
        see x + nl + y + nl # works fine
        myobj = new otherclass {
            see name + nl
            see x + nl + y + nl # error !
```

```

    }

class otherclass
    name = "test"

```

Output:

```

10
20
test

Line 8 Error (R24) : Using uninitialized variable : x
In method test() in file methodbraceerror.ring
called from line 5 in file methodbraceerror.ring

```

Now what we will do to solve the previous problem?

Solution (1) : Write the code that access the class attributes outside braces.

```

new point { test() }

class point
    x=10 y=20
    func test
        see x + nl + y + nl # works fine
        myobj = new otherclass {
            see name + nl
        }
        see x + nl + y + nl # Outside braces - works fine

class otherclass
    name = "test"

```

Output:

```

10
20
test
10
20

```

Solution (2) : Don't Use Braces

```

new point { test() }

class point
    x=10 y=20
    func test
        see x + nl + y + nl
        myobj = new otherclass
        see myobj.name
        see x + nl + y + nl

class otherclass
    name = "test"

```

Solution (3) : Copy the self object

We may use this solution if we want to use braces and get access to the class attributes (Just Reading).

```

new point { test() }

class point
  x=10 y=20
  func test
    oSelf = self
    see x + nl + y + nl
    myobj = new otherclass {
      see name + nl
      see oself.x + nl + oself.y + nl
    }

class otherclass
  name = "test"

```

Output:

```

10
20
test
10
20

```

Now look at this line

```
oself = self
```

The problem with the previous line is that we will have a new copy from the object. Because in Ring the assignment operator copy lists and objects by value (not by reference).

When we access the new object attributes (reading) we don't have problems

But if we modified the object attributes (Then we will modify the copy!).

Note: We can use braces again with the copy

```

new point { test() }

class point
  x=10 y=20
  func test
    oSelf = self
    see x + nl + y + nl
    myobj = new otherclass {
      see name + nl
      oSelf {
        see x + nl + y + nl
      }
    }

class otherclass
  name = "test"

```

In a GUI application, we may create a class contains the window objects as attributes to be able to access the controls from different methods. Remember the previous information when you try to access objects using braces inside methods because in this case you can't access the object attributes directly and if you copied the self object you will work on a copy and the new controls that you create will be related to the copy and you can't access them.

61.10 Accessing the class attributes from braces inside class methods

We access the class attributes directly from the class methods, also we have the choice to use the Self reference before the attribute/method name. Using Braces {} inside class method change the active object scope and prevent us from getting direct access to the class attributes. Also using Self will not help because the Self reference will be changed to the object that we access using Braces.

In this case if you want to read an attribute you have to copy the Self object before using Braces and if you want to modify an attribute you have to copy from local variable to the object attribute after using Braces.

This case happens when you want to read/modify attribute inside braces.

```
Class MyApp

  oCon    # Attribute

  # some code here

  Func OpenDatabase
    # some code here
    new QSqlDatabase() {
      oCon = addDatabase("QSQLITE") {
        setDatabaseName("weighthistory.db")
        open()
      }
    }
    self.oCon = oCon
    # some code here
```

In the previous example we want to create the connection object and save it inside the oCon attribute.

The object is an output from the addDatabase() method that we use after accessing the QSqlDatabase() object.

Inside braces we can't use the Self reference to use the object created from the MyApp class, Because the Self reference here will be to the object that we access using Braces.

We solved the problem in the previous example by creating a local variable called oCon then after Braces we copied that variable to the oCon attribute.

The next code is another solution.

```
Class MyApp

  oCon    # Attribute

  # some code here

  Func OpenDatabase
    # some code here
    oCon = new QSqlDatabase()
    oCon = oCon.addDatabase("QSQLITE") {
      setDatabaseName("weighthistory.db")
      Open()
    }
    # some code here
```

The next code is a better solution.

```

Class MyApp

    oCon    # Attribute

    # some code here

    Func OpenDatabase
        # some code here
        new QSqlDatabase() {
            this.oCon = addDatabase("QSQLITE") {
                setDatabaseName("weighthistory.db")
                Open()
            }
        }
        # some code here

```

Note: We used this.attribute to access the class attribute (oCon) while we are inside Braces.

61.11 Creating a Class for each Window in GUI applications

A good way for creating classes for windows is to define the window directly after the class name

In this area you can use nested braces without problems to define the window and the controls, and they will be attributes that you can access from methods.

Example:

```

Load "guilib.ring"

new qApp
{
    $ObjectName = "oFirstWindow"
    oFirstWindow = new FirstWindow

    $ObjectName = "oSecondWindow"
    oSecondWindow = new SecondWindow

    exec()
}

Class FirstWindow

    win = new QWidget() {
        setGeometry(0,50,300,200)
        setWindowTitle("First Window")
        label1 = new QLabel(win)
        {
            setGeometry(10,10,300,30)
            setText("0")
        }
        btn1 = new QPushButton(win)
        {
            move(100,100)
            setText("Increment")
            setClickEvent($ObjectName+".increment()")
        }
    }

```



```

        show()
    }

    Func Increment
        label1 {
            setText( "" + ( 0 + text() + 1 ) )
        }
    }

Class SecondWindow

    win = new QWidget() {
        setGeometry(400,50,300,200)
        setWindowTitle("Second Window")
        label1 = new QLabel(win)
        {
            setGeometry(10,10,300,30)
            setText("0")
        }
        btn1 = new QPushButton(win)
        {
            move(100,100)
            setText("Decrement")
            setClickEvent($ObjectName+".decrement()")
        }
        show()
    }

    Func Decrement
        label1 {
            setText( "" + ( 0 + text() - 1 ) )
        }
    }

```

61.12 Conflict between self inside braces and self in the class region

In the class region (after the class name and before any methods) we define the attributes.

In this region we have access to the global scope and the local scope will point to the object scope.

Three Scopes

- Global Scope —> Global Scope
- Object Scope —> Object Scope
- Local Scope —> Object Scope

Look at this example

```

New Account {
    see aFriends
}

Class Account
    name = "Mahmoud"
    aFriends = []
    aFriends + new Friend {
        name = "Gal"
    }

```

```

    }
    aFriends + new Friend {
        name = "Bert"
    }

Class Friend
    name

```

Output:

```

name: NULL
name: NULL

```

The problem in the previous example is that the Class account contains an attribute called “name” and the Friend class contains an attribute called “name” also.

If you tried using self.name inside braces you will get the same result!

```

New Account {
    see aFriends
}

Class Account
    name = "Mahmoud"
    aFriends = []
    aFriends + new Friend {
        self.name = "Gal"
    }
    aFriends + new Friend {
        self.name = "Bert"
    }
}

Class Friend
    name

```

So why using self.name inside braces doesn’t solve this conflict?

Because after the class region we have

- global scope —> global scope
- object scope —> object scope (Account Class)
- local scope —> local scope (Account Class)

When we use braces we change the object scope, so we have

- global scope —> global scope
- object scope —> object scope (Friend Class)
- local scope —> local scope (Account Class)

Ring search in the local scope first, so using self.name will use the Account class.

There are many solution

Solution (1) : Access the object through the list

```

New Account {
    see aFriends
}

Class Account

```

```

name = "Mahmoud"
aFriends = []
aFriends + new Friend
aFriends[len(aFriends)] {
    aFriends[len(aFriends)].name = "Gal"
}
aFriends + new Friend
aFriends[len(aFriends)] {
    aFriends[len(aFriends)].name = "Bert"
}

Class Friend
    name

```

Solution (2) : Create Method in the friend class to set the name attribute.

```

New Account {
    see aFriends
}

Class Account
    name = "Mahmoud"
    aFriends = []
    aFriends + new Friend {
        setname("Gal")
    }
    aFriends + new Friend {
        setname("Bert")
    }
}

Class Friend
    name
    func setname cName
        name = cName

```

Solution (3) : Create a method in the account class to set the attribute

```

New Account {
    see aFriends
}

Class Account
    name = "Mahmoud"
    aFriends = []
    friend("Gal")
    friend("Bert")

    func friend cName
        aFriends + new Friend {
            name = cName
        }
}

Class Friend
    name

```

Solution (4) : Declarative Programming

```

New Account {
    name = "mahmoud"
    friend {

```

```

        name = "Gal"
    }
    friend {
        name = "Bert"
    }
    see aFriends
}

Class Account
    name
    aFriends = []
    friend
    func getfriend
        aFriends + new Friend
        return aFriends[len(aFriends)]

Class Friend
    name

```

Output:

```

name: Gal
name: Bert

```

61.13 Using braces to escape from the current object scope

Since braces change the current object scope to another object, we can use it to do some work without modifying the class attributes and using the same variable names.

```

new point {x=10 y=20 z=30 start() }
class point x y z
    func start
        see self # print the x y z values (10,20,30)
        new Local {
            x = 100
            y = 200
            z = 300
        }
        see self # print the x y z values (10,20,30)
        see x + nl # will print 100
        see y + nl # will print 200
        see z + nl # will print 300
        Self { # NO Advantage - Search is done in local scope first
            see x + nl # will print 100
            see y + nl # will print 200
            see z + nl # will print 300
        }
        see self.x + nl # will print 10
        see self.y + nl # will print 20
        see self.z + nl # will print 30
    }
class Local

```

Output:

```

x: 10.000000
y: 20.000000

```

```

z: 30.000000
x: 10.000000
y: 20.000000
z: 30.000000
100
200
300
100
200
300
10
20
30

```

61.14 Summary of Scope Rules

At first remember that

- 1 - Each programming language comes with it's scope rules based on the language goals
- 2 - Programming in the small is different than Programming in the Large
- 3 - Some programming language are designed for developing small programs while others are designed for large programs
- 4 - In programming, If we have access to more than one scope - Then problems may come if we don't manage things correctly
- 5 - It's always more secure to reduce the number of visible scopes
- 6 - Some programming languages force you to manage the scope in some way, while others not!

In Ring

- 1 - Special and *very simple* scope rules that are designed for Flexibility first then Security
- 2 - Ring is designed to support programming in the small and programming in the large.
- 3 - The language provide the different programming paradigms that you may select from based on the project size. Errors comes only if you selected a bad paradigm for the target project or you are using the paradigm in a way that is not correct or at least not common.
- 4 - In Ring you have the choice, you can use global variables or avoid them. you can give them a special \$ mark or leave them. you can use object-oriented or stay with procedures. you can use the class region (after the class name and before any method) just for attributes or use it for code too.
- 5 - Just read the next scope rules and think about them then use them in your favorite way.

Scope Rules:

- 1 - At any place in our program code we have only at maximum Three Scopes (Local Scope, Object Scope and Global Scope).
- 2 - When Ring find a variable it will search in the local scope first then in the object scope then in the global scope.
- 3 - At any time inside procedures or methods you can use braces { } to access an object and change the current object scope.
- 4 - In the class region (After the class name and before any method) this is a special region where both of the object scope and the local scope point to the object scope. I.e. No local variables where each variable you define in this region will become an attribute.

5 - Before defining any variable (in any scope and in the class region too) a search process will be done to use the variable if it's found.

6 - Functions and Methods parameters are defined automatically as local variables to these functions or methods.

7 - Using `Object.Attribute` will search in the object attributes only.

8 - Using `Self.Attribute` will lead to a search for `Self` first then search in `Self Attributes`.

9 - The `Self` reference inside class region (after the class name and before any method) always point to the object scope created from the class.

10- The `Self` reference inside methods will be changed when we uses Braces to be a reference to the object that we access.

11- Writing variable names directly in the class region (after the class name and before any method) means using them or define then (in order).

12- Using `self.attribute` in the class region reduce search to the object scope (avoid conflict with global scope).

From these rules you can understand all types of conflicts and why you may have them and how to avoid them

Simple advices to avoid any conflict and use the scope rules in a better way

1 - Try to avoid global variables

2 - Use the Main Function - This will help you to avoid global variables

3 - If you are going to use many global variables use the \$ mark before the variable name

4 - In the class region if you don't respect the advice number three (\$) then use `self.attribute` when you define your attributes

5 - You can use `object.attribute` and `object.method()` instead of `object { attribute }` and `object { method() }` if you don't like changing the object scope.

6 - If you will use nested braces in a class - think about using the class region if possible because in this region you will have access to the object that you access using `{ }` + access to the class attributes

7 - If you are inside a class method and used nested braces you will change the object scope with each brace and you will loss the access to the class attributes directly but you have access to the local scope before and after using brace `{ }` , if you will read/modify the class attribute from braces then use `This.Attribute` because using 'This' means (The object created from this class) while using 'Self' means (The object in the current object scope).

After understanding all of the previous points, You will master this topic.

SCOPE RULES FOR FUNCTIONS AND METHODS

In this chapter we will learn about the scope rules for functions and methods.

You need to know the next information once you started using Ring for large applications.

These applications may contains and use

- Many Packages and Classes written in Ring
- Many Functions written in Ring
- Standard Ring Functions (Written in C language)
- Functions and Classes written in C/C++ languages

62.1 How Ring find the Functions and Methods?

When you call a method or function, Ring will start a search process to find this function

If found → Call the function and store the function pointer in the cache so Ring can use it again with doing another search.

If not found → Runtime error message (That you can avoid using Try/Catch)

How the search process is done?

Search for functions/methods follow the next order

- 1 - Search in methods (if we are inside class method or object using braces { })
- 2 - Search in functions written by the programmer using Ring Code
- 3 - Search in functions written in C/C++ like standard Ring functions

This enable us to write clean code inside classes methods and avoid any conflict with functions.

If we want to call a function with the same name as a method in the class we will need a wrapper function or we will access a temp. object using { } then call that function there.

We can replace C/C++ Functions with Ring Functions.

We can replace Ring Functions with Ring Methods.

Note: Using self.method() is not necessary in any use case.

Tip: We can use this.method() to escape from the current active scope that we access using braces { } and call a method in the class that we are inside.

62.2 Example about Sharing Names between Functions and Methods

Look at the next example

```
func main
    o1 = new myclass { test() test2() }
    test2()

func f1
    see "f1 function" + nl

func f2
    see "f2 function" + nl

func f3
    see "f3 function" + nl

func test2
    myline()
    see "test2 function" + nl
    new myclass {
        f1()
        f2()
        f3()
        self.f3()
    }
    myobj = new myclass
    myobj.f3()
    myline()

func myline
    see copy("=",40) + nl

Class myclass

    func test
        myline()
        see "test method" + nl
        f1()
        f2()
        f3()
        myline()

    func f3
        see "f3 method" + nl

    func test2
        myline()
        see "test2 method" + nl
        self {
            f1()
            f2()
            f3()
        }
        myline()
```

Output:


```

=====
test method
f1 function
f2 function
f3 method
=====
=====
test2 method
f1 function
f2 function
f3 method
=====
=====
test2 function
f1 function
f2 function
f3 method
f3 method
f3 method
=====
=====

```

62.3 Calling a function sharing the name with a method in the current class

In the previous example we have a function called f3() and we have a method called f3()

How we can call the f3() function from the test() method ?

Solution (1) : Change the current object scope to another object scope

In this solution we will have an empty class called local that we will use to change the current object scope.

```

func main
    o1 = new myclass { test() }

func f1
    see "f1 function" + nl

func f2
    see "f2 function" + nl

func f3
    see "f3 function" + nl

func myline
    see copy("=", 40) + nl

Class myclass

    func test
        myline()
        see "test method" + nl
        f1()
        f2()
        f3() # call f3() method
        new local { f3() } # call f3() function
        myline()

```

```
    func f3
        see "f3 method" + nl
class local
```

Output:

```
=====
test method
f1 function
f2 function
f3 method
f3 function
=====
```

SYNTAX FLEXIBILITY

In this chapter we will learn about some options that are provided automatically by the Ring compiler for syntax flexibility.

63.1 Change Language Keywords

We can change any keyword using the `ChangeRingKeyword` command.

Note: Remember to restore the keyword again if the team will mix between styles in the same project.

Tip: The `ChangeRingKeyword` command is executed in the scanner stage by the compiler (before parsing).

Syntax:

```
ChangeRingKeyword <oldkeyword> <newkeyword>
```

Example:

```
ChangeRingKeyword see print
print "welcome" + nl
ChangeRingKeyword print see
see "Welcome" + nl
```

Example:

```
ChangeRingKeyword func function
ChangeRingKeyword see print
ChangeRingKeyword ok endif
ChangeRingKeyword next endfor
ChangeRingKeyword end endwhile

x = 10
while x > 0
    print "x = " + x + nl
    for t = 1 to 10
        if t = 3
            print "number three" + nl
        endif
    endfor
```

```
x--
endwhile

test()

function test
    print "message from test" + nl

ChangeRingKeyword function func
ChangeRingKeyword print see
ChangeRingKeyword endif ok
ChangeRingKeyword endfor next
ChangeRingKeyword endwhile end
```

63.2 Change Language Operators

We can change any operator using the `ChangeRingOperator` command.

Note: Remember to restore the operator again if the team will mix between styles in the same project.

Tip: The `ChangeRingOperator` command is executed in the scanner stage by the compiler (before parsing).

Syntax:

```
ChangeRingOperator <oldkeyword> <newkeyword>
```

Example:

The next program hide the `+` operator by changing it to `_+`

```
changingoperator + _+
changingkeyword SEE PRINT

try
    print 5 + 10
catch
    print nl print "error" print nl
done

changingoperator _+ +
```

The next program change the `+` operator to “plus”.

```
changingoperator + plus
changingkeyword SEE PRINT

Print 5 plus 5

changingoperator plus +
changingkeyword PRINT SEE
```

63.3 Load Syntax Files

You may store a group of `ChangeRingKeyword` and `ChangeRingOperator` commands in a file to use later in many source files. You can't use the `Load` command to call these files because

- `ChangeRingKeyword` and `ChangeRingOperator` commands are executed in the scanner phase by the compiler (before parsing).
- The `load` command is executed in the parsing phase (after the scanner phase).

Solution: Use the `LoadSyntax` Command which is executed in the scanner phase.

Syntax:

```
LoadSyntax "syntaxfile.ring"
```

Example:

File : `StyleBasicOn.ring`

```
ChangeRingKeyword  see    print
ChangeRingKeyword  ok     endif
ChangeRingKeyword  next   endfor
ChangeRingKeyword  end    endwhile
```

File : `StyleBasicOff.ring`

```
ChangeRingKeyword  print    see
ChangeRingKeyword  endif    ok
ChangeRingKeyword  endfor   next
ChangeRingKeyword  endwhile end
```

File : `UseStyleBasic.ring`

```
LoadSyntax "stylebasicon.ring"

x = 10
while x > 0
    print "x = " + x + nl
    for t = 1 to 10
        if t = 3
            print "number three" + nl
        endif
    endfor
    x--
endwhile

LoadSyntax "stylebasicoff.ring"

see "done" + nl
```

Note: files called by the `LoadSyntax` command must contain `ChangeRingKeyword` and `ChangeRingOperator` commands only.

Tip: files called by the `LoadSyntax` command doesn't support functions, packages and classes. just imperative commands only.

Note: Using this feature you can create many styles that you can use in the same project and you can support Ring

translation to other languages like Arabic, French and so on.

Tip: The effect of LoadSyntax command is related to the current source code file only.

63.4 Using “()” around the function parameters

We can use () around the function parameters (optional).

Example:

```
hello()
sum(3,4)

func hello()
    see "Hello" + nl

func sum(x,y)
    see x+y+nl
```

Output:

```
Hello
7
```

Example:

```
myfunc = func x,y { see x + y + nl }

call myfunc (3,4)

myfunc2 = func (x,y) { see x+y+nl }

call myfunc(3,4)
```

Output:

```
7
7
```

63.5 Using Semi-colon after and between statements

In Ring we can use semi-colon after and between statements (optional).

Example:

```
# Using semi-colon is optional

see "Hello" + nl ; see "How are you?" + nl ; see "Welcome to Ring" + nl ;
one() ; two() ; three() ;
func one ; see "one" + nl ;
func two ; see "two" + nl ;
func three ; see "three" + nl ;
```

Output:

```
Hello
How are you?
Welcome to Ring
one
two
three
```

63.6 Using \$ and @ in the start of the variable name

You can use any unicode character in the variable name also we can use \$ and @ in the name.

This feature may help, for example we can start global variables with \$ and the object attributes with @.

In other languages like Ruby this is the rule, In the Ring language this is just an option without any force from the Compiler.

example:

```
$global_variable = 5

new test { hello() }

class test

    @instance_variable = 10

    func hello

        local_variable = 15

        see "Global    : " + $global_variable + nl +
           "Instance  : " + @instance_variable + nl +
           "Local     : " + local_variable + nl
```

Output:

```
Global    : 5
Instance  : 10
Local     : 15
```

63.7 Using the ‘elseif’ keyword as ‘but’ in if statement

if you don't like the ‘but’ keyword in if statement Then you can use the ‘elseif’ keyword.

Example:

```
give x
if x = 1 see "one"
elseif x=2 see "two"
elseif x=3 see "three"
elseif x=4 see "four"
else see "other"
ok
see nl
```

63.8 Using the ‘else’ keyword as ‘other’ in switch statement

if you don't like the ‘other’ keyword in switch statement Then you can use the ‘else’ keyword.

Also you can replace ‘else’ with ‘other’ in if statement.

i.e. ‘other’ keyword is the same as ‘else’ keyword.

Example:

```
x = 1
switch x
  on 10
    see "10" + nl
  else
    see "not 10" + nl
end
```

Output:

```
not 10
```

63.9 Using the ‘end’ keyword in different control structures

We can use the ‘end’ keyword to close different control structures

- If statement
- For loop
- Switch
- While
- Try-Catch

Example:

```
see "if statement.." + nl
x = 1
if x = 1
  see "one" + nl
elseif x=2
  see "two" + nl
elseif x=3
  see "three" + nl
end
see "for loop.." + nl
for t = 1 to 10
  see t
end
see nl
see "switch..." + nl
x = 1

switch x
  on 1 see "one" + nl
  on 2 see "two" + nl
end
```



```

see "try catch..." + nl
try
    x = 1 / 0
catch
    see "catching error" + nl
end

```

Output:

```

if statement..
one
for loop..
12345678910
switch...
one
try catch...
catching error

```

63.10 Using braces to start and end different control structures

We can use braces { } to start and end different control structures

- If statement
- For loop
- Switch
- While
- Try-Catch

Example:

```

see "if statement.." + nl
x = 1
if x = 1 {
    see "one" + nl
elseif x=2
    see "two" + nl
elseif x=3
    see "three" + nl
}
see "for loop.." + nl
for t = 1 to 10 {
    see t
}
see nl
see "switch..." + nl
x = 1

switch x {
    on 1 see "one" + nl
    on 2 see "two" + nl
}

see "try catch..." + nl
try {

```

```

    x = 1 / 0
catch
    see "catching error" + nl
}

```

Output:

```

if statement..
one
for loop..
12345678910
switch...
one
try catch...
catching error

```

63.11 Using ‘put’ and ‘get’ as ‘see’ and ‘give’

We can replace the ‘see’ keyword with the ‘put’ keyword.

Also we can replace the ‘give’ keyword with the ‘get’ keyword.

Example:

```

put "Hello World" + nl
put "Enter Your Name ? " Get Name
Put "Hello " + Name

```

63.12 Using ‘case’ as ‘on’ in switch statements

We can replace the ‘on’ keyword with ‘case’ keyword in the switch statement.

Example (1) :

```

for x=1 to 10
    switch x
    case 1 put "one" + nl
    case 2 put "two" + nl
    case 3 put "thre" + nl
    else put "else" + nl
end

```

Example (2) :

```

for x=1 to 10 {
    switch x {
    case 1 put "one" + nl
    case 2 put "two" + nl
    case 3 put "thre" + nl
    else put "else" + nl
    }
}

```

63.13 Using ‘def’ as ‘func’ in functions/methods definition

We can use the ‘def’ keyword as the ‘func’ keyword to define functions and methods.

Example:

```
one() two()

def one put "one" + nl
def two put "two" + nl
```

63.14 Using braces { } in Packages/Classes/Functions

Example:

```
load "stdlib.ring"

import mypackage

new myclass {
    myfunc()
}

package mypackage
{
    class myclass
    {
        func myfunc
        {
            print("Hello, World!\n")
        }
    }
}
```

63.15 Using ‘end’ keyword after Packages/Classes/Functions

Example:

```
import mypackage

new myclass {
    myfunc()
}

package mypackage
    class myclass
        def myfunc
            put "Hello, World!"
        end
    end
end
```

63.16 Using ‘endpackage’/’endclass’/’endfunc’ keywords after Packages/Classes/Functions

Example:

```
import mypackage

new myclass { myfunc() }

package mypackage
  class myclass
    func myfunc
      see "welcome" + nl
    endfunc
  endclass
endpackage
```

INTRODUCTION TO THE TYPE HINTS LIBRARY

In this chapter we will learn about the Type Hints Library

64.1 Why Type Hints?

Using this library we can add the type information to the source code which will be very useful for tools like

- Code Editors
- Static-Analysis

Note: Ring is a dynamic language, No type checking will be done by the compiler.

64.2 Example

The next example will use the Type Hints library

```
load "typehints.ring"

see sum(3,4) + nl ;
see sayHello("Mahmoud");

int func sum(int x,int y) {
    return x+y ;
}

string func sayHello(string name) {
    return "Hello " + name ;
}
```

64.3 User Types

The Type Hints library is very powerful and will support user types (Classes) automatically

Example:

```
load "typehints.ring"

import mypackage
```

```
test() { main([:one,:two,:three]) }

myclass func test() {
    see "Testing User Types!" + nl
    return new myclass
}

package mypackage {
    public class myclass {
        public static void func main(list args) {
            see "welcome" + nl
            see args
        }
    }
}
```

64.4 Using Types inside Code

Also you can use the types inside the code (not only the function prototype)

Example:

```
load "typehints.ring"

int    sum = sum(3,4)
string msg = sayHello("Mahmoud")

see "Sum = " + sum + nl + msg + nl

int func sum(int x,int y) {
    return x+y ;
}

string func sayHello(string name) {
    return "Hello " + name ;
}
```

64.5 Rules

- To use the types in the function prototype you must use ‘(‘ and ‘)’ around parameters
- To use the types in the function code, You must set the variable value (Assignment).

The next types are defined by the library

```
# Low Level Types
char
unsigned
signed
int
short
long
float
```

```
double
void

# High Level Types
string
list
number
object

# Other
public
static
abstract
protected
override
```

COMMAND LINE OPTIONS

The ring language takes source code file (*.ring*) or the *object file* (*.ringo*) as input to execute, also the language provide other options like

Option	Description
-tokens	Print a list of tokens in the source code file
-rules	Print grammar rules applied on the tokens
-ic	Print the intermediate byte code (before execution)
-icfinal	Print the final byte code (after execution)
-cgi	Print http response header before error messages
-norun	Don't run the program after compiling
-ins	Print instruction operation code before execution
-performance	Print clock before and after program execution
-go	Generate Object File
-w	Display Warnings

65.1 Printing Tokens

Example:

```
Func Main
    See "Hello World" + nl
    for x = 1 to 10
        see x + nl
    next
    test()

func test
    see "welcome" + nl
    o1 = new point { x=10 y=20 z=30 }
    see o1

class point x y z
```

Command:

```
ring test.ring -tokens -norun
```

Output:

```
=====
Tokens - Generated by the Scanner
=====
```



```

Keyword : FUNC
Identifier : main
EndLine
Keyword : SEE
Literal : Hello World
Operator : +
Identifier : nl
EndLine
Keyword : FOR
Identifier : x
Operator : =
Number : 1
Keyword : TO
Number : 10
EndLine
Keyword : SEE
Identifier : x
Operator : +
Identifier : nl
EndLine
Keyword : NEXT
EndLine
Identifier : test
Operator : (
Operator : )
EndLine
Keyword : FUNC
Identifier : test
EndLine
Keyword : SEE
Literal : welcome
Operator : +
Identifier : nl
EndLine
Identifier : ol
Operator : =
Keyword : NEW
Identifier : point
Operator : {
Identifier : x
Operator : =
Number : 10
Identifier : y
Operator : =
Number : 20
Identifier : z
Operator : =
Number : 30
Operator : }
EndLine
Keyword : SEE
Identifier : ol
EndLine
Keyword : CLASS
Identifier : point
Identifier : x
Identifier : y
Identifier : z

```

```
EndLine
```

65.2 Printing Rules

Command:

```
ring test.ring -rules -norun
```

Output:

```
=====
Grammar Rules Used by The Parser
=====

Rule : Program --> {Statement}

Line 1
Rule : Statement --> 'Func' Identifier [ParaList]

Line 2
Rule : Factor --> Literal
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : Factor --> Identifier [ {Mixer} | Assignment | PlusPlus | MinusMinus]
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : Arithmetic --> Arithmetic + Arithmetic
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR --> BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot --> EqualOrNot
Rule : Expr --> LogicNot
Rule : Statement --> 'See' Expr

Line 3
Rule : Factor --> Number
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR --> BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot --> EqualOrNot
Rule : Expr --> LogicNot
Rule : Factor --> Number
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
```

```

Rule : BitAnd --> BitShift
Rule : BitOrXOR --> BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot --> EqualOrNot
Rule : Expr --> LogicNot
Rule : Statement --> 'For' Identifier '=' Expr to Expr ['step' Expr]

Line 4
Rule : Factor --> Identifier [ {Mixer} | Assignment | PlusPlus | MinusMinus]
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : Factor --> Identifier [ {Mixer} | Assignment | PlusPlus | MinusMinus]
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : Arithmetic --> Arithmetic + Arithmetic
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR --> BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot --> EqualOrNot
Rule : Expr --> LogicNot
Rule : Statement --> 'See' Expr

Line 5
Rule : Next --> 'Next'

Line 6
Rule : Mixer --> '(' [Expr { ',' Expr } ] ')'

Line 8
Rule : Factor --> Identifier [ {Mixer} | Assignment | PlusPlus | MinusMinus]
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR --> BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot --> EqualOrNot
Rule : Expr --> LogicNot
Rule : Statement --> Expr
Rule : Statement --> 'Func' Identifier [ParaList]

Line 9
Rule : Factor --> Literal
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : Factor --> Identifier [ {Mixer} | Assignment | PlusPlus | MinusMinus]
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : Arithmetic --> Arithmetic + Arithmetic

```

```

Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR --> BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Statement --> 'See' Expr

Line 10
Rule : Factor --> New Identifier { '.' Identifier }
Rule : Mixer --> '{' {Statement} BraceEnd
Rule : Factor --> Number
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR --> BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Assignment -> '=' Expr
Rule : Factor --> Identifier [ {Mixer} | Assignment | PlusPlus | MinusMinus]
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR --> BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Statement --> Expr
Rule : Factor --> Number
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR --> BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Assignment -> '=' Expr
Rule : Factor --> Identifier [ {Mixer} | Assignment | PlusPlus | MinusMinus]
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR --> BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot

```

```

Rule : Expr --> LogicNot
Rule : Statement --> Expr
Rule : Factor --> Number
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR --> BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Assignment -> '=' Expr
Rule : Factor --> Identifier [ {Mixer} | Assignment | PlusPlus | MinusMinus]
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR --> BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Statement --> Expr
Rule : BraceEnd --> '}'
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR --> BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Assignment -> '=' Expr
Rule : Factor --> Identifier [ {Mixer} | Assignment | PlusPlus | MinusMinus]
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR --> BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Statement --> Expr

Line 11
Rule : Factor --> Identifier [ {Mixer} | Assignment | PlusPlus | MinusMinus]
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift

```

```

Rule : BitOrXOR --> BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Statement --> 'See' Expr

Line 13
Rule : Statement --> 'Class' Identifier
Rule : Factor --> Identifier [ {Mixer} | Assignment | PlusPlus | MinusMinus]
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR --> BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Statement --> Expr
Rule : Factor --> Identifier [ {Mixer} | Assignment | PlusPlus | MinusMinus]
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR --> BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Statement --> Expr
Rule : Factor --> Identifier [ {Mixer} | Assignment | PlusPlus | MinusMinus]
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR --> BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Statement --> Expr
=====

```

65.3 Printing Intermediate Code

Command:

```
ring test.ring -ic -norun
```

Output:

```
=====
Byte Code - Before Execution by the VM
=====
```

PC	OPCode	Data
1	ReturnNull	
2	Func	main
3	NewLine	2
4	FuncExE	
5	PushC	Hello World
6	LoadA	nl 0
7	PushV	
8	SUM	0
9	Print	
10	NewLine	3
11	ExitMark	29 28
12	LoadAFirst	x
13	PushN	1.000000
14	BeforeEqual	0
15	Assignment	
16	PushN	1.000000
17	StepNumber	
18	JumpVarLENum	x 10.000000 29
19	NewLine	4
20	FuncExE	
21	LoadA	x 0
22	PushV	
23	LoadA	nl 0
24	PushV	
25	SUM	0
26	Print	
27	NewLine	5
28	IncJump	x 18
29	POPExitMark	
30	POPStep	
31	NewLine	6
32	LoadFunc	test
33	Call	0
34	NoOperation	
35	NewLine	8
36	PushV	
37	FreeStack	
38	ReturnNull	
39	Func	test
40	NewLine	9
41	FuncExE	
42	PushC	welcome
43	LoadA	nl 0
44	PushV	
45	SUM	0
46	Print	
47	NewLine	10
48	LoadA	o1 0
49	AssignmentPointer	
50	New	point
51	SetScope	
52	PushV	

```

53 BraceStart
54     LoadA      x      0      58
55 AssignmentPointer
56     PushN      10.000000
57 BeforeEqual    0
58 Assignment      0      0
59 FreeStack
60     LoadA      y      0      64
61 AssignmentPointer
62     PushN      20.000000
63 BeforeEqual    0
64 Assignment      0      0
65 FreeStack
66     LoadA      z      0      70
67 AssignmentPointer
68     PushN      30.000000
69 BeforeEqual    0
70 Assignment      0      0
71 FreeStack
72     LoadFunc   ismethod
73     LoadA      self    0
74     PushV
75     PushC      braceend
76     Call
77 NoOperation
78     PushV
79     JumpZ      85
80     LoadFunc   braceend
81     Call
82 NoOperation
83     PushV
84 FreeStack
85 BraceEnd
86 FreeStack
87     NewLine      11
88     FuncExE
89     LoadA      o1      0
90     PushV
91     Print
92     NewLine      13
93 ReturnNull
94     Class      point  006E8BC0
95     NewLabel
96     LoadA      x      0
97     PushV
98 FreeStack
99     LoadA      y      0
100    PushV
101 FreeStack
102    LoadA      z      0
103    PushV
104 FreeStack
105 ReturnNull

```

```
=====
```


65.4 Printing Final Intermediate Code

Command:

```
ring test.ring -icfinal
```

Output:

Hello World
1
2
3
4
5
6
7
8
9
10
welcome
x: 10.000000
y: 20.000000
z: 30.000000

=====
Byte Code - After Execution by the VM
=====

PC	OPCode	Data
1	ReturnNull	
2	Func	main
3	NewLine	2
4	FuncExE	
5	PushC	Hello World
6	PushP	007D3670 0
7	PushV	
8	SUM	0
9	Print	
10	NewLine	3
11	ExitMark	29 28
12	LoadAFirst	x
13	PushN	1.000000
14	BeforeEqual	0
15	Assignment	
16	PushN	1.000000
17	StepNumber	
18	JumpVarLPLENum	x 10.000000 29
19	NewLine	4
20	FuncExE	
21	PushPLocal	x 0
22	PushV	
23	PushP	007D3670 0
24	PushV	
25	SUM	0
26	Print	
27	NewLine	5

```

28  IncLPJump      x      18
29  POPExitMark
30  POPStep
31  NewLine        6
32  LoadFuncP     test
33  Call           0
34  NoOperation
35  NewLine        8
36  PushV
37  FreeStack
38  ReturnNull
39  Func           test
40  NewLine        9
41  FuncExE
42  PushC          welcome
43  PushP          007D3670      0
44  PushV
45  SUM            0
46  Print
47  NewLine        10
48  PushPLocal     o1      0
49  AssignmentPointer
50  New            point
51  SetScope
52  PushV
53  BraceStart
54  LoadA          x      0      58
55  AssignmentPointer
56  PushN          10.000000
57  BeforeEqual    0
58  SetProperty     0      106
59  FreeStack
60  LoadA          y      0      64
61  AssignmentPointer
62  PushN          20.000000
63  BeforeEqual    0
64  SetProperty     0      141
65  FreeStack
66  LoadA          z      0      70
67  AssignmentPointer
68  PushN          30.000000
69  BeforeEqual    0
70  SetProperty     0      176
71  FreeStack
72  LoadFunc      ismethod
73  LoadA         self      0
74  PushV
75  PushC          braceend
76  Call
77  NoOperation
78  PushV
79  JumpZ          85
80  LoadFunc      braceend
81  Call
82  NoOperation
83  PushV
84  FreeStack
85  BraceEnd

```

```

86   FreeStack
87   NewLine      11
88   FuncExE
89   PushPLocal   o1      0
90   PushV
91   Print
92   NewLine      13
93   ReturnNull
94   Class        point   007D8470
95   NewLabel
96   LoadA       x        0
97   PushV
98   FreeStack
99   LoadA       y        0
100  PushV
101  FreeStack
102  LoadA       z        0
103  PushV
104  FreeStack
105  ReturnNull
106  LoadFunc     ismethod
107  LoadA       ring_gettemp_var      0
108  PushV
109  PushC        setx
110  Call         0
111  NoOperation
112  PushV
113  JumpZ        132
114  NewLine      2
115  LoadA       ring_gettemp_var      0
116  LoadMethod   setx
117  LoadA       ring_settemp_var      0
118  PushV
119  Call         0      1
120  AfterCallMethod
121  PushV
122  FreeStack
123  NewLine      3
124  LoadA       ring_tempflag_var      0      128
125  AssignmentPointer
126  PushN        0.000000
127  BeforeEqual   0
128  Assignment    0      0
129  FreeStack
130  NewLine      4
131  Jump         140
132  NewLine      5
133  PushP        007D37D8      0      137
134  AssignmentPointer
135  PushN        1.000000
136  BeforeEqual   0
137  Assignment    0      0
138  FreeStack
139  NewLine      6
140  Return
141  LoadFunc     ismethod
142  LoadA       ring_gettemp_var      0
143  PushV

```

```

144     PushC     sety
145     Call      0
146 NoOperation
147     PushV
148     JumpZ     167
149     NewLine   2
150     LoadA    ring_gettemp_var    0
151 LoadMethod    sety
152     LoadA    ring_settemp_var    0
153     PushV
154     Call      0      1
155 AfterCallMethod
156     PushV
157 FreeStack
158     NewLine   3
159     LoadA    ring_tempflag_var    0      163
160 AssignmentPointer
161     PushN    0.000000
162 BeforeEqual    0
163 Assignment     0      0
164 FreeStack
165     NewLine   4
166     Jump     175
167     NewLine   5
168     PushP    007D37D8      0      172
169 AssignmentPointer
170     PushN    1.000000
171 BeforeEqual    0
172 Assignment     0      0
173 FreeStack
174     NewLine   6
175     Return
176 LoadFunc     ismethod
177     LoadA    ring_gettemp_var    0
178     PushV
179     PushC     setz
180     Call      0
181 NoOperation
182     PushV
183     JumpZ     202
184     NewLine   2
185     LoadA    ring_gettemp_var    0
186 LoadMethod    setz
187     LoadA    ring_settemp_var    0
188     PushV
189     Call      0      1
190 AfterCallMethod
191     PushV
192 FreeStack
193     NewLine   3
194     LoadA    ring_tempflag_var    0      198
195 AssignmentPointer
196     PushN    0.000000
197 BeforeEqual    0
198 Assignment     0      0
199 FreeStack
200     NewLine   4
201     Jump     210

```

```

202      NewLine      5
203      PushP      007D37D8      0      207
204      AssignmentPointer
205      PushN      1.000000
206      BeforeEqual      0
207      Assignment      0      0
208      FreeStack
209      NewLine      6
210      Return
=====

```

65.5 CGI Support

Command:

```
ring test.ring -cgi
```

65.6 No Run

Command:

```
ring test.ring -norun
```

65.7 Printing Instruction Operation Code

Command:

```
ring test.ring -ins
```

Output:

```

=====
Operation   : ReturnNull
PC          : 1
Line Number : 1 , File test.ring

SP (After) : 0 - FuncSP : 0
  LineNumber 1
=====
.....
.....
.....

```

Tip: Output removed from the previous example because it's very large!

65.8 Performance

Command:

```
ring test.ring -performance
```

Output:

```
=====
Date   : 2015/09/15 Time : 15:56:17
Clock  : 0
=====
Hello World
1
2
3
4
5
6
7
8
9
10
welcome
x: 10.000000
y: 20.000000
z: 30.000000

=====
Date   : 2015/09/15 Time : 15:56:17
Clock  : 0
=====
```

65.9 Generate Object File

You can generate object file (*.ringo*) from your source code file (*.ring*) using `-go` option

Tip: You will get one object file to use for distributing/running your application which may contains one or many ring source files that you can keep or distribute based on the application (commercial or open source).

Command:

```
ring test.ring -go
```

To run the compiled object file

```
ring test.ringo
```

DISTRIBUTING RING APPLICATIONS

In this chapter we will learn about distributing Ring applications.

The next method is old and was used in Ring 1.5 and previous versions!

Starting from Ring 1.6 we have a nice tool called Ring2EXE

Using Ring2EXE we can distribute applications quickly for Windows, Linux and macOS

Check the Ring2EXE chapter for more information!

66.1 Distributing Applications for Microsoft Windows

Step 1:

```
Copy c:\ring\bin folder to be for example c:\myapp
```

Step 2:

```
Rename c:\myapp\ring.exe to c:\myapp\myapp.exe
```

Step 3:

```
Create a file c:\myapp\ring.ring
```

And write

```
Load "myapp.ring"
```

When you run myapp.exe the file ring.ring will be executed automatically

So your file myapp.ring will be called and executed

Or just rename myapp.ring to ring.ring

It's a fast way to distribute applications.

66.2 Protecting the Source Code

Step 1:

Execute the next command

```
ring myapp.ring -go
```

This will generate one object file (myapp.ringo) from the project files (*.ring)

Step 2:

`Rename myapp.ringo to ring.ringo`

When you run the executable file (ring.exe) or (myapp.exe) the file ring.ringo will be executed.

66.3 Creating Windows Installer

There are many tools that you can use to distribute your application.

Check : nullsoft scriptable install system

URL : http://nsis.sourceforge.net/Main_Page

66.4 Using C/C++ Compiler and Linker

Another method to distribute applications is to use a C/C++ compiler.

Ring can be embedded in C/C++ projects, We can create executable files using a C/C++ compiler by embedding the Ring language in our project.

Check the “Embedding Ring Language in C/C++ Programs” chapter.

Using this way we will avoid using ring.ring or ring.ringo files.

66.5 Distributing Applications and Games for Mobile

Ring can be embedded in a Qt projects or LibSDL projects to build Mobile applications and Games.

You can build the Qt project or the LibSDL project and get the Android package directly (*.apk)

Check Ring distributions for Mobile development using Qt or LibSDL.

DISTRIBUTING RING APPLICATIONS USING RING2EXE

In this chapter we will learn about distributing Ring applications.

Starting from Ring 1.6 we have a nice tool called Ring2EXE (Written in Ring itself)

Using Ring2EXE we can distribute applications quickly for Windows, Linux, macOS and Mobile devices

67.1 Using Ring2EXE

```
ring2exe filename.ring [Options]
```

This will set filename.ring as input to the program

The next files will be generated

filename.ringo	(The Ring Object File - by Ring Compiler)
filename.c	(The C Source code file Contains the ringo file content Will be generated by this program)
filename_buildvc.bat	(Will be executed to build filename.c using Visual C/C++)
filename_buildgcc.bat	(Will be executed to build filename.c using GNU C/C++)
filename_buildclang.bat	(Will be executed to build filename.c using CLang C/C++)
filename.obj	(Will be generated by the Visual C/C++ compiler)
filename.exe	(Will be generated by the Visual C/C++ Linker)
filename	(Executable File - On Linux & MacOS X platforms)

67.2 How Ring2EXE works?

At first the Ring compiler will be used to generate the Ring object file (*.ringo)

If we have a C compiler (optional), This object file will be embedded inside a C source code file

Then using the C compiler and the Ring library (Contains the Ring Virtual Machine) the executable file will be generated!

If we don't have a C compiler, the Ring executable will be copied and renamed to your application name

And your Ring object file (*.ringo) will become ring.ringo to be executed at startup of the executable file.

So it's better and easy to have a C compiler on your machine to be used by Ring2EXE.

67.3 Example

We have test.ring contains the next code

```
see "Hello, World!" + nl
```

To build th executable file for Windows, Linux or macOS

```
ring2exe test.ring
```

To run the program (Windows)

```
test
```

To run the program (Linux and macOS)

```
./test
```

67.4 Options

-keep	: Don't delete Temp. Files
-static	: Build Standalone Executable File (Don't use ring.dll/ring.so/ring.dylib)
-gui	: Build GUI Application (Hide the Console Window)
-dist	: Prepare application for distribution
-allruntime	: Include all libraries in distribution
-mobileqt	: Prepare Qt Project to distribute Ring Application for Mobile
-noqt	: Remove RingQt from distribution
-noallegro	: Remove RingAllegro from distribution
-noopenssl	: Remove RingOpenSSL from distribution
-nolibcurl	: Remove RingLibCurl from distribution
-nomysql	: Remove RingMySQL from distribution
-noodbc	: Remove RingODBC from distribution
-nosqlite	: Remove RingSQLite from distribution
-noopengl	: Remove RingOpenGL from distribution
-nofreeglut	: Remove RingFreeGLUT from distribution
-nolibzip	: Remove RingLibZip from distribution
-noconsolecolors	: Remove RingConsoleColors from distribution
-nomurmurhash	: Remove RingMurmurHash from distribution
-nocruntime	: Remove C Runtime from distribution
-qt	: Add RingQt to distribution
-allegro	: Add RingAllegro to distribution
-openssl	: Add RingOpenSSL to distribution
-libcurl	: Add RingLibCurl to distribution
-mysql	: Add RingMySQL to distribution
-odbc	: Add RingODBC to distribution
-sqlite	: Add RingSQLite to distribution
-opengl	: Add RingOpenGL to distribution
-freeglut	: Add RingFreeGLUT to distribution
-libzip	: Add RingLibZip to distribution
-consolecolors	: Add RingConsoleColors to distribution
-murmurhash	: Add RingMurmurHash to distribution
-cruntime	: Add C Runtime to distribution

67.5 Building standalone console application

Using the “-static” option we can build executable console application

So we don’t have to use ring.dll, ring.so or ring.dylib

This avoid only the need to Ring dynamic link library

If you are using another libraries, You will need to include it with your application.

```
ring2exe test.ring -static
```

67.6 Distributing RingAllegro Applications

We have test2.ring contains the next code

```
# Just a simple program to test Ring2EXE Tool!
# Using RingAllegro

load "gameengine.ring" # Give Control to the Game Engine

func main                # Called by the Game Engine

    oGame = New Game      # Create the Game Object
    {
        title = "My First Game"
    }
```

To build the executable file and prepare for distributing the Game

We use “-dist” option and “-allruntime” to include all libraries

```
ring2exe test2.ring -dist -allruntime
```

After executing the previous command

On Windows we will have : target/windows folder

On Linux we will have : target/linux folder

On macOS we will have : target/macos folder

The previous command will add all of the Ring runtime libraries to our distribution

But we may need only RingAllegro, So it’s better to use the next command

```
ring2exe test2.ring -dist -allegro -cruntime
```

This will produce smaller size distribution and will avoid the runtime files that we don’t need!

Also we could use the “-gui” option to hide the console window

So it’s better to use the next command

```
ring2exe test2.ring -dist -gui -allegro -cruntime
```

67.7 Distributing RingQt Applications

We have test3.ring contains the next code

```
# Just a simple program to test Ring2EXE Tool!
# Using RingQt

load "guilib.ring"

new qApp {
    new QWidget() {
        setwindowtitle("Hello, World!")
        resize(400,400)
        show()
    }
    exec()
}
```

To build the executable file and prepare for distributing the GUI application

We use “-dist” option and “-allruntime” to include all libraries

```
ring2exe test3.ring -dist -allruntime
```

After executing the previous command

On Windows we will have : target/windows folder

On Linux we will have : target/linux folder

On macOS we will have : target/macos folder

The previous command will add all of the Ring runtime libraries to our distribution

But we may need only RingQt, So it's better to use the next command

```
ring2exe test3.ring -dist -qt -cruntime
```

This will produce smaller size distribution and will avoid the runtime files that we don't need!

Also we could use the “-gui” option to hide the console window

So it's better to use the next command

```
ring2exe test3.ring -dist -gui -qt -cruntime
```

67.8 Distributing Applications for Mobile using RingQt

To prepare a Qt project for your RingQt application (test3.ring) use the “-mobileqt” option

Example :

```
ring2exe test3.ring -dist -mobileqt
```

After executing the previous command, We will have the Qt project in target/mobile/qtproject folder

The main project file will be project.pro which we can open using the Qt Creator IDE.

Also we will have the resource file : project.qrc

Another important file is our C++ main file : main.cpp

67.9 Building the Cards Game for Mobile using RingQt

For a better example, consider building an Android package for the Cards game that comes with the

Ring language in this folder : ring/application/cards

The Cards game folder contains three files

cards.ring : The Game source code

cards.jpg : The image file used by the game

project.qrc : Resource file to be used with the Qt project

The resource file contains the next content

```
<RCC>
    <qresource>
        <file>cards.ringo</file>
        <file>cards.jpg</file>
    </qresource>
</RCC>
```

We have two files in the resource file

The first file is cards.ringo (The Ring Object File) and the second file is cards.jpg (The image file)

As a start, Ring2EXE will generate this resource file in target/mobile/qtproject/project.qrc

But this file will contains only cards.ringo (That Ring2EXE will generate by calling Ring compiler)

We need to update this resource file to add the image file : cards.jpg

After this update, we copy the resource file to the main application folder

So when we use Ring2EXE again, Our updated resource file will be used!

Now to build the cards game for Mobile

1. Run the next command

```
ring2exe cards.ring -dist -mobileqt
```

2. Open target/mobile/qtproject/project.pro using Qt creator
3. Build and Run using Qt Creator

How the Cards game will find the image file ?

RingQt comes with a simple function : AppFile() that we can use to determine the files that we may access on Desktop or Mobile platforms

The next code from cards.ring

```
mypic = new QPixmap(AppFile("cards.jpg"))
```

So all what you need is using AppFile() function around your image files!

67.10 Building the Weight History Application for Mobile using RingQt

Another example to distribute your application for Mobile Devices using Ring2EXE and Qt

consider building an Android package for the Weight History application that comes with the

Ring language in this folder : ring/application/weighthistory

The Weight History application folder contains four files

weighthistory.ring : The application source code

weighthistory.db : The SQLite database

project.qrc : The resource file for the Qt project

main.cpp : The main C++ source file for the Qt project

To build the Weight History application for Mobile

1. Run the next command

```
ring2exe weighthistory.ring -dist -mobileqt
```

2. Open target/mobile/qtproject/project.pro using Qt creator

3. Build and Run using Qt Creator

The resource file (project.qrc) contains two files

```
<RCC>
    <qresource>
        <file>weighthistory.ringo</file>
        <file>weighthistory.db</file>
    </qresource>
</RCC>
```

The first file is weighthistory.ringo (Ring Object File - Generated by Ring2EXE by calling Ring compiler)

The database file : weighthistory.db

The main.cpp contains the next little update, To copy the database file from resources to a writable location on the mobile device

```
QString path3 ;
path3 = path+"/weighthistory.db";
QFile::copy(":/weighthistory.db",path3);
```

You will need to do this with database files only!

When we use Ring2EXE, the tool will check for project.qrc and main.cpp, if they exist then your updated files will be used in target/mobile/qtproject instead of the default version generated by Ring2EXE

So Use Ring2EXE to generate these files, Then copy them to your application folder when you update them.

67.11 Building the Form Designer for Mobile using RingQt

To build the Form Designer application (ring/applications/formdesigner) for Mobile

1. Run the next command

```
ring2exe formdesigner.ring -dist -mobileqt
```

2. Open target/mobile/qtproject/project.pro using Qt creator

3. Build and Run using Qt Creator

in the folder ring/application/formdesigner You will find the resource file : project.qrc

It will be used automatically by Ring2EXE

```
<RCC>
    <qresource>
        <file>formdesigner.ringo</file>
        <file>image/allevnts.png</file>
        <file>image/checkbox.png</file>
        <file>image/close.png</file>
        <file>image/combobox.bmp</file>
        <file>image/datepicker.bmp</file>
        <file>image/dial.png</file>
        <file>image/formdesigner.png</file>
        <file>image/frame.png</file>
        <file>image/grid.bmp</file>
        <file>image/hyperlink.png</file>
        <file>image/image.png</file>
        <file>image/label.png</file>
        <file>image/layout.png</file>
        <file>image/lcdnumber.png</file>
        <file>image/listview.png</file>
        <file>image/lock.png</file>
        <file>image/new.png</file>
        <file>image/open.png</file>
        <file>image/progressbar.png</file>
        <file>image/project.png</file>
        <file>image/pushbutton.png</file>
        <file>image/radiobutton.png</file>
        <file>image/save.png</file>
        <file>image/saveas.png</file>
        <file>image/select.png</file>
        <file>image/slider.png</file>
        <file>image/spinner.bmp</file>
        <file>image/statusbar.png</file>
        <file>image/tab.png</file>
        <file>image/textarea.png</file>
        <file>image/textfield.png</file>
        <file>image/timer.png</file>
        <file>image/toolbar.png</file>
        <file>image/tree.bmp</file>
        <file>image/videowidget.png</file>
        <file>image/webview.png</file>
    </qresource>
</RCC>
```

As we did in the Cards game, The Form Designer will use the AppFile() function to determine the name of the Image files.

The next code from ring/applications/formdesigner/mainwindow/formdesignernview.ring

```
func CreateToolBar
    aBtns = [
        new qtoolbutton(win) {
            setbtnimage(self, AppFile("image/new.png"))
            setclideanvent(Method(:NewAction))
            settooltip("New File")
        },
        new qtoolbutton(win) {
            setbtnimage(self, AppFile("image/open.png"))
```

```

        setclickevent (Method (:OpenAction))
        settooltip ("Open File")
    } ,
    new qtoolbutton(win) {
        setbtnimage (self, AppFile ("image/save.png"))
        setclickevent (Method (:SaveAction))
        settooltip ("Save")
    } ,
    new qtoolbutton(win) {
        setbtnimage (self, AppFile ("image/saveas.png"))
        setclickevent (Method (:SaveAsAction))
        settooltip ("Save As")
    } ,
    new qtoolbutton(win) {
        setbtnimage (self, AppFile ("image/close.png"))
        setclickevent (Method (:ExitAction))
        settooltip ("Exit")
    }
}

tool1 = win.addtoolbar("files") {
    for x in aBtns { addwidget(x) addseparator() }
}

```

From this example, We know that we can use sub folders for images.

67.12 Creating the Qt resource file using Folder2qrc

When we have large RingQt project that contains a lot of images and files, We need to add these files to the resource file (*.qrc) when distributing applications for Mobile devices.

Instead of adding these files one by one, Ring 1.6 comes with a simple tool that save our time, It's called Folder2qrc.

Example:

```
folder2qrc formdesigner.ring
```

We determine the main source file while we are in the application folder, and Folder2qrc will check all of the files in the current folder and sub folders, Then add them to the resource file after the mainfile.ringo (In our example this will be formdesigner.ringo)

The output file will be : project.qrc

You can open it and remove the files that you don't need in the resources!

67.13 Important Information about Ring2EXE

- Using Ring2EXE to prepare distribution will delete all of the files in the old distribution

for example, if you have target/windows folder then used

```
ring2exe test3.ring -dist -allruntime
```

The files in target/windows will be deleted before adding the files again

This is important when you prepare a distribution for Mobile devices


```
ring2exe test3.ring -dist -mobileqt
```

If you modified the resource file : project.qrc or the main file : main.cpp

Don't forget to copy them to the application folder!

So Ring2EXE can use the updated version if you tried the previous command again!

- Ring2EXE is written in Ring, and you can read the source code from

<https://github.com/ring-lang/ring/blob/master/ring2exe/ring2exe.ring>

- The libraries information are stored in a separated file, So this file can be updated in the future automatically to support new libraries

<https://github.com/ring-lang/ring/blob/master/ring2exe/ring2exe.data>

LOW LEVEL FUNCTIONS

In this chapter we will learn about the low level functions provided by Ring

- `callgc()`
- `varptr()`
- `space()`
- `nullpointer()`
- `object2pointer()`
- `pointer2object()`
- `ptrcmp()`
- `ringvm_cfunctionslist()`
- `ringvm_functionslist()`
- `ringvm_classeslist()`
- `ringvm_packageslist()`
- `ringvm_memorylist()`
- `ringvm_calllist()`
- `ringvm_fileslist()`
- `ringvm_settrace()`
- `ringvm_tracedata()`
- `ringvm_traceevent()`
- `ringvm_tracefunc()`
- `ringvm_scopescount()`
- `ringvm_evalinscope()`
- `ringvm_passerror()`
- `ringvm_hideerrorMsg()`
- `ringvm_callfunc()`

68.1 callgc() function

Use this function to force calling the garbage collector during function execution when you use a loop that create temp. variables that you don't free using the assignment operation.

It's very rare to need this function but it's useful when you create something like event-loop for your game engine and start creating lists on the fly when you call functions.

Example

```
While True
    # process events
    # call functions using temp. lists like myfunc(["temp list"])

    # call the garbage collector
    callgc()
End
```

Tip: In Ring the garbage collector works automatically in the end of function execution or when you use the assignment statement.

68.2 varptr() function

Use the varptr() function when you need to pass a pointer to a C/C++ function.

Syntax:

varptr(cVariableName,cPointerType) —> Low Level Object (C Pointer)

example:

```
r = 10
z = 20
see r + nl
see varptr("r","int")
see varptr("z","int")
```

Output:

```
10
00E3C740
int
2
00E3BEC0
int
2
```

Note: the low level object is a list contains three items (The Pointer, The Type, The Status)

68.3 space() function

Use the space function to allocate a specific number of bytes in Memory.

Syntax:

```
Space(nBytesCount) ---> String
```

Example:

```
mystring = space(200)
See "String Size : " + len(mystring) + nl
See "String : " + mystring + nl
See "String Pointer : "
See varptr("mystring", "char *")
```

Output:

```
String Size : 200
String :
String Pointer : 00FF8FE8
char *
2
```

Note: You may need the space() and VarPtr() functions to pass buffers to C functions.

68.4 nullpointer() function

You may need to pass the NULL pointer to a C function that may expect a pointer as parameter and accept NULL pointers for optional parameters.

Example:

The next example uses the SDL_BlitterSurface() function from the LibSDL Library through RingSDL. The function accepts SDL_Rect pointers in the second and the last parameter. Also the function accepts NULL pointers, so we can pass them using the NULLPointer() Function.

```
SDL_BlitterSurface(text, nullpointer(), surface, nullpointer())
```

Note: The previous code doesn't work alone, you need to learn how to use RingSDL first.

Tip: We can pass NULL as parameter instead of using NULLPointer()

68.5 object2pointer() function

Use this function to get a C pointer for Ring lists and objects

Syntax:

```
object2pointer(List|Object) --> Low Level Object ( C Pointer )
```

68.6 pointer2object() function

Use this function to get the Ring list and/or object from the low level object (C Pointer)

Syntax:

```
pointer2object(Low Level Object) ---> List|Object
```

Example:

```
# Create the list
mylist = 1:5

# Create pointer to the list
x = object2pointer(mylist)
see x

see nl

# Add items to the list
mylist + "welcome"

# print the list items
y = pointer2object(x)
see y
```

Output:

```
0069A5D8
OBJECTPOINTER
0

1
2
3
4
5
welcome
```

Note: In Ring the assignment operator copy lists and objects by value, to copy by reference Just use the `object2pointer()` and `pointer2object()` functions.

Tip: The `object2pointer()` and `pointer2object()` are used in the `stdlib - Tree Class` implementation to create a reference for the parent node (object) in the child node (another object).

68.7 ptrcmp() function

We can compare between two pointers (C Objects) using the `ptrcmp()` function.

Syntax:

```
ptrcmp(oObject1,oObject2) ---> value = 1 if oObject1 = oObject2
                                value = 0 if oObject1 != oObject2
```

Example:

```
fp = fopen("ptrcmp.ring", "r")
fp2 = fp
fp3 = fopen("ptrcmp.ring", "r")
```

```

see ptrcmp(fp,fp2) + nl
see ptrcmp(fp,fp3) + nl

fclose(fp)
fclose(fp3)

```

Output:

```

1
0

```

68.8 ringvm_cfunctionslist() function

The Function return a list of functions written in C.

Syntax:

```
RingVM_CFunctionsList() ---> List
```

Example:

```
See RingVM_CFunctionsList()
```

68.9 ringvm_functionslist() function

The Function return a list of functions written in Ring.

Each List Member is a list contains the next items

- Function Name
- Program Counter (PC) - Function Position in Byte Code.
- Source Code File Name
- Private Flag (For Private Methods in Classes)

Syntax:

```
RingVM_FunctionsList() ---> List
```

Example:

```

test()

func test
    see ringvm_functionslist()

```

Output:

```

test
8
B:/ring/tests/scripts/functionslist.ring
0

```

68.10 ringvm_classeslist() function

The Function return a list of Classes.

Each List Member is a list contains the next items

- Class Name
- Program Counter (PC) - Class Position in Byte Code.
- Parent Class Name
- Methods List
- Flag (Is parent class information collected)
- Pointer to the package (or NULL if no package is used)

Syntax:

```
RingVM_ClassesList() ---> List
```

Example:

```
see ringvm_classeslist()

class class1
    func f1
class class2 from class1
class class3 from class1
```

Output:

```
class1
9

f1
13
B:/ring/tests/scripts/classeslist.ring
0
0
00000000
class2
16
class1
0
00000000
class3
20
class1
0
00000000
```

68.11 ringvm_packageslist() function

The Function return a list of Packages.

Each List Member is a list contains the next items

- Package Name

- Classes List

Syntax:

```
RingVM_PackagesList() ---> List
```

Example:

```
see ringvm_packageslist()

package package1
    class class1

package package2
    class class1

package package3
    class class1
```

Output:

```
package1
class1
11

0
00FEF838
package2
class1
17

0
00FEF978
package3
class1
23

0
00FEFF68
```

68.12 ringvm_memorylist() function

The Function return a list of Memory Scopes and Variables.

Each List Member is a list contains variables in a different scope.

Each Item in the scope list is a list contains the next items

- Variable Name
- Variable Type
- Variable Value
- Pointer Type (List/Item) if the value is a list
- Private Flag (if the variable is an attribute in a Class)

Syntax:


```
RingVM_MemoryList() ---> List
```

Example:

```
x = 10
test()
func test
    y = 20
    see ringvm_memorylist()
```

Output:

```
true
2
1
0
0
false
2
0
0
0
nl
1

0
0
null
1

0
0
ring_gettemp_var
4
00000000
0
0
ccatcherror
1
NULL
0
0
ring_settemp_var
4
00000000
0
0
ring_tempflag_var
2
0
0
0
stdin
3
50512DB8
file
0
0
```

```

0
stdout
3
50512DD8
file
0
0
0
stderr
3
50512DF8
file
0
0
0
this
4
00000000
0
0
sysargv
3
B:\ring\bin\ring
B:/ring/tests/scripts/memorylist.ring
0
0
x
2
10
0
0
Y
2
20
0
0

```

68.13 ringvm_calllist() function

The Function return a list of the functions call list.

Each List Member is a list contains the next items

- Function Type
- Function Name
- Program Counter (PC)
- Stack Pointer (SP)
- Temp. Memory List
- Method or Function Flag
- Caller PC
- FuncExec Flag
- ListStart Flag

- Nested Lists Pointer
- State List

Syntax:

```
RingVM_CallList() ---> List
```

Example:

```
hello()
func hello
    test()

func test
    mylist = ringvm_calllist()
    for t in mylist see t[2] + nl next
```

Output:

```
function hello() in file B:/ring/tests/scripts/calllist.ring
called from line 1
function test() in file B:/ring/tests/scripts/calllist.ring
called from line 3
ringvm_calllist
```

68.14 ringvm_fileslist() function

Function return a list of the Ring Files.

Syntax:

```
RingVM_FilesList() ---> List
```

Example:

```
load "stdlib.ring"
see ringvm_fileslist()
```

Output:

```
B:/ring/tests/scripts/fileslist.ring
B:\ring\bin\stdlib.ring
eval
stdlib.ring
stdlib.rh
stdclasses.ring
stdfunctions.ring
stdbase.ring
stdstring.ring
stdlist.ring
stdstack.ring
stdqueue.ring
stdmath.ring
stddatetime.ring
stdfile.ring
stdsystem.ring
stddebug.ring
stddatatype.ring
```

```
stdconversion.ring
stdodbc.ring
stdmysql.ring
stdsecurity.ring
stdinternet.ring
stdhashtable.ring
stdtree.ring
```

68.15 ringvm_settrace()

The function ringvm_settrace() determine the Trace function name

The trace function is a Ring function that will be called for each event

Syntax:

```
RingVM_SetTrace(cCode)
```

68.16 ringvm_tracedata()

Inside the function that we will use for tracing events

We can use the ringvm_tracedata() function to get the event data.

The event data is a list contains the next items

- The Source Code Line Number
- The Source File Name
- The Function/Method Name
- Method or Function (Bool : True=Method, False=Function/File)

Syntax:

```
RingVM_TraceData() ---> aDataList
```

68.17 ringvm_traceevent()

Inside the function that we will use for tracing events

We can use ringvm_traceevent() to know the event type

- New Line
- Before Function
- After Function
- Runtime Error
- Before C Function
- After C Function

Syntax:

```
RingVM_TraceEvent () ---> nTraceEvent
```

68.18 ringvm_tracefunc()

The function return the name of the function that we are using for tracing events.

Syntax:

```
RingVM_TraceEvent () ---> cCode
```

68.19 ringvm_scopescount()

We can use the RingVM_ScopesCount() function to know the number of scopes used in the application.

In the start of the program, We have the (global scope only)

When we call a function, A new scope is created.

When the function execution is done, the function scope is deleted.

Syntax:

```
RingVM_ScopesCount () ---> nScopes
```

68.20 ringvm_evalinscope()

The function ringvm_evalinscope() is similar to the eval() function

Unlike eval() which execute the code in the current scope

Using RingVM_EvalInScope() we can execute the scope in a specific scope.

Syntax:

```
RingVM_EvalInScope (nScope, cCode)
```

68.21 ringvm_passerror()

When we have runtime error, After printing the Error message, Ring will end the execution of the program.

Using ringvm_passerror() we can avoid that, and continue the execution of our program.

Syntax:

```
RingVM_PassError ()
```

68.22 ringvm_hideerrormsg()

We can disable/enable displaying the runtime error messages using the RingVM_HideErrorMsg() function.

Syntax:

```
RingVM_HideErrorMsg(lStatus)
```

68.23 ringvm_callfunc()

We can call a function from a string without using eval() using the ringvm_callfunc()

Syntax:

```
RingVM_CallFunc(cFuncName)
```

68.24 Example - Using the Trace Functions

The next example use the Trace Functions to trace the program Events!

In practical, We will use the Trace Library instead of these low level functions!

```
load "tracelib.ring"

ringvm_settrace("mytrace() ")

see "Hello, world!" + nl
see "Welcome" + nl
see "How are you?" +nl
mytest()
new myclass { mymethod() }

func mytest
    see "Message from mytest" + nl

func mytrace
    see "==== The Trace function is Active =====" + nl +
        "Trace Function Name : " + ringvm_TraceFunc() + nl +
        "Trace Event : "
    switch ringvm_TraceEvent()
        on TRACEEVENT_NEWLINE          see "New Line"
        on TRACEEVENT_NEWFUNC          see "New Function"
        on TRACEEVENT_RETURN           see "Return"
        on TRACEEVENT_ERROR            see "Error"
        on TRACEEVENT_BEFORECFUNC      see "Before C Function"
        on TRACEEVENT_AFTERCFUNC       see "After C Function"
    off
    see nl +
        "Line Number : " + ringvm_tracedata()[TRACEDATA_LINENUMBER] + nl +
        "File Name   : " + ringvm_tracedata()[TRACEDATA_FILENAME] + nl +
        "Function Name : " + ringvm_tracedata()[TRACEDATA_FUNCNAME] + nl +
        "Method or Function : "
    if ringvm_tracedata()[TRACEDATA_METHODORFUNC] =
        TRACEDATA_METHODORFUNC_METHOD
        see "Method"
    else
        if ringvm_tracedata()[TRACEDATA_FUNCNAME] = NULL
            see "Command"
        else
            see "Function"
```

```

                                ok
                                ok
                                see nl + Copy("=",42) + nl

class myclass
    func mymethod
        see "Message from mymethod" + nl

```

Output:

```

===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : After C Function
Line Number : 3
File Name : test1.ring
Function Name : ringvm_settrace
Method or Function : Function
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 5
File Name : test1.ring
Function Name :
Method or Function : Command
=====
Hello, world!
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 6
File Name : test1.ring
Function Name :
Method or Function : Command
=====
Welcome
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 7
File Name : test1.ring
Function Name :
Method or Function : Command
=====
How are you?
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 8
File Name : test1.ring
Function Name :
Method or Function : Command
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : New Function
Line Number : 8
File Name : test1.ring

```

```

Function Name : mytest
Method or Function : Function
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 12
File Name : test1.ring
Function Name : mytest
Method or Function : Function
=====
Message from mytest
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 14
File Name : test1.ring
Function Name : mytest
Method or Function : Function
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : Return
Line Number : 8
File Name : test1.ring
Function Name :
Method or Function : Command
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 9
File Name : test1.ring
Function Name :
Method or Function : Command
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 43
File Name : test1.ring
Function Name :
Method or Function : Command
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : Before C Function
Line Number : 9
File Name : test1.ring
Function Name : ismethod
Method or Function : Function
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : After C Function
Line Number : 9
File Name : test1.ring
Function Name : ismethod

```



```

Method or Function : Function
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : New Function
Line Number : 9
File Name : test1.ring
Function Name : mymethod
Method or Function : Method
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 44
File Name : test1.ring
Function Name : mymethod
Method or Function : Method
=====
Message from mymethod
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : Return
Line Number : 9
File Name : test1.ring
Function Name :
Method or Function : Command
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : Before C Function
Line Number : 9
File Name : test1.ring
Function Name : ismethod
Method or Function : Function
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : After C Function
Line Number : 9
File Name : test1.ring
Function Name : ismethod
Method or Function : Function
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : Before C Function
Line Number : 9
File Name : test1.ring
Function Name : ismethod
Method or Function : Function
=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : After C Function
Line Number : 9
File Name : test1.ring
Function Name : ismethod
Method or Function : Function

```

```

=====
===== The Trace function is Active =====
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 11
File Name : test1.ring
Function Name :
Method or Function : Command
=====

```

68.25 Example - The Trace Library

The next example uses the Trace functions provided by the Ring language to create the Trace library.

Using the Trace library we have nice Tracing tools and Interaction debugger too.

```

# Trace Events
TRACEEVENT_NEWLINE      = 1
TRACEEVENT_NEWFUNC      = 2
TRACEEVENT_RETURN       = 3
TRACEEVENT_ERROR        = 4
TRACEEVENT_BEFORECFUNC  = 5
TRACEEVENT_AFTERCFUNC   = 6

# Trace Data
TRACEDATA_LINENUMBER    = 1
TRACEDATA_FILENAME      = 2
TRACEDATA_FUNCNAME      = 3
TRACEDATA_METHODORFUNC  = 4

# Method of Function
TRACEDATA_METHODORFUNC_METHOD      = TRUE
TRACEDATA_METHODORFUNC_NOTMETHOD   = FALSE

TRACE_BREAKPOINTS = TRUE

TRACE_TEMPLIST = []

func Trace cType
    switch trim(lower(cType))
    on :AllEvents
        ringvm_settrace("TraceLib_AllEvents()")
    on :Functions
        ringvm_settrace("TraceLib_Functions()")
    on :PassError
        ringvm_settrace("TraceLib_PassError()")
    on :Debugger
        ringvm_settrace("TraceLib_Debugger()")
    on :LineByLine
        ringvm_settrace("TraceLib_LineByLine()")
    off

func TraceLib_AllEvents
    if right(ringvm_tracedata()[TRACEDATA_FILENAME],13) = "tracelib.ring"
        return
    ok
    see "===== The Trace function is Active =====" + nl +

```

```

        "Trace Function Name : " + ringvm_TraceFunc() + nl +
        "Trace Event : "
switch ringvm_TraceEvent()
    on TRACEEVENT_NEWLINE          see "New Line"
    on TRACEEVENT_NEWFUNC          see "New Function"
    on TRACEEVENT_RETURN           see "Return"
    on TRACEEVENT_ERROR            see "Error"
    on TRACEEVENT_BEFORECFUNC      see "Before C Function"
    on TRACEEVENT_AFTERCFUNC       see "After C Function"
off
see nl +
    "Line Number : " + ringvm_tracedata()[TRACEDATA_LINENUMBER] + nl +
    "File Name   : " + ringvm_tracedata()[TRACEDATA_FILENAME] + nl +
    "Function Name : " + ringvm_tracedata()[TRACEDATA_FUNCNAME] + nl +
    "Method or Function : "
    if ringvm_tracedata()[TRACEDATA_METHODORFUNC] =
        TRACEDATA_METHODORFUNC_METHOD
        see "Method"
    else
        if ringvm_tracedata()[TRACEDATA_FUNCNAME] = NULL
            see "Command"
        else
            see "Function"
    ok
ok
see nl + Copy("=",42) + nl

func TraceLib_Functions
    if right(ringvm_tracedata()[TRACEDATA_FILENAME],13) = "tracelib.ring"
        return
    ok
    switch ringvm_TraceEvent()
        on TRACEEVENT_NEWFUNC
            see "Open Func : " +
            ringvm_TraceData()[TRACEDATA_FUNCNAME] + nl
        on TRACEEVENT_RETURN
            see "Return to Func : " +
            ringvm_TraceData()[TRACEDATA_FUNCNAME] + nl
    off

func TraceLib_PassError
    if right(ringvm_tracedata()[TRACEDATA_FILENAME],13) = "tracelib.ring"
        return
    ok
    switch ringvm_TraceEvent()
        on TRACEEVENT_ERROR
            see nl
            see "TraceLib : After Error !" + nl
            ringvm_passerror()
    off

func TraceLib_Debugger
    if right(ringvm_tracedata()[TRACEDATA_FILENAME],13) = "tracelib.ring"
        return
    ok
    switch ringvm_TraceEvent()
        on TRACEEVENT_ERROR
            _BreakPoint()

```

```

off

func TraceLib_LineByLine
    if right(ringvm_tracedata()[TRACEDATA_FILENAME],13) = "tracelib.ring" or
        ringvm_TraceEvent() != TRACEEVENT_NEWLINE
        return

    ok
    aList = ringvm_tracedata()
    see "Before Line : " + aList[TRACEDATA_LINENUMBER] + nl
    _BreakPoint()

func BreakPoint
    if not TRACE_BREAKPOINTS
        return

    ok
    _BreakPoint()

func _BreakPoint
    see nl+nl+Copy("=",60) + nl +
    Copy(" ",20)+"Interactive Debugger" + nl +
    Copy("=",60) + nl +
    "Command (Exit)          : End Program" + nl +
    "Command (Cont)         : Continue Execution" + nl +
    "Command (Locals)        : Print local variables names" + nl +
    "Command (LocalsData)    : Print local variables data" + nl +
    "Command (Globals)       : Print global variables names" + nl +
    "We can execute Ring code" + nl +
    Copy("=",60) + nl
    while true
        see nl + "code:> "
        give cCode
        cmd = trim(lower(cCode))
        if cmd = "exit" or cmd = "bye"
            shutdown()

        ok
        nScope = ringvm_scopescount()-2
        switch cmd
            on "locals"
                ringvm_EvalInScope(nScope,"see locals() callgc()")
                loop
            on "localsdata"
                PrintLocalsData(nScope)
                loop
            on "globals"
                ringvm_EvalInScope(nScope,"see globals() callgc()")
                loop
            on "cont"
                ringvm_passerror()
                exit

        off
        Try
            ringvm_EvalInScope(nScope,cCode)
            catch
                see cCatchError

        done

    end

func NoBreakPoints

```

```

TRACE_BREAKPOINTS = FALSE

func PrintLocalsData nScope
  if nScope = 1 # Global
    ringvm_Evalinscope(nScope, 'TRACE_TEMPLIST = globals()')
  else
    ringvm_Evalinscope(nScope, 'TRACE_TEMPLIST = locals() callgc()')
  ok
  see nl
  aTempList = TRACE_TEMPLIST
  TRACE_TEMPLIST = []
  nSpaces = 5
  for TRACE_ITEM in aTempList
    if len(TRACE_ITEM) + 5 > nSpaces
      nSpaces = len(TRACE_ITEM) + 5
    ok
  next
  for TRACE_ITEM in aTempList
    see "Variable : " + TRACE_ITEM
    cVarName = TRACE_ITEM
    see copy(" ", nSpaces - len(cVarName)) + " Type : "
    ringvm_Evalinscope(nScope, "see type(" + TRACE_ITEM + ")")
    ringvm_Evalinscope(nScope, "see Copy(' ', fabs(15 - len(type(" +
      TRACE_ITEM + "))))")
    see " Value : "
    ringvm_Evalinscope(nScope, "see " + TRACE_ITEM)
    see nl
  next

```

THE TRACE LIBRARY AND THE INTERACTIVE DEBUGGER

In this chapter we will learn about the Trace Library and the Interactive Debugger

69.1 Loading the Trace library

To start using the Trace library, We must load it first!

```
load "tracelib.ring"
```

69.2 Trace All Events

The next example demonstrates the Trace library usage to trace all events.

```
# Trace All Events
trace(:AllEvents)

see "Hello, world!" + nl
see "Welcome" + nl
see "How are you?" +nl

mytest()

new myclass { mymethod() }

func mytest
    see "Message from mytest" + nl

class myclass
    func mymethod
        see "Message from mymethod" + nl
```

69.3 Trace control flow between functions

The next example demonstrates the Trace library usage to trace the control flow between functions.

```
Trace(:Functions)

test1()
```

```

func test1
    see :test1 + nl
    test2()

func test2
    see :test2 + nl
    see test3() + nl

func test3
    see :test3 + nl
    return "test 3 output"

```

69.4 Pass Error

The next example demonstrates the Trace library usage to pass an error!

```

Trace(:PassError)

test1()

func test1
    x = 10
    see :test1 + nl
    test2() # Runtime Error!
    see "We can continue!"

```

69.5 Interactive Debugger

The next example demonstrates the Trace library usage to use the Interactive Debugger

```

Trace(:Debugger)

test1()
see "good bye!" + nl

func test1
    x = 10
    see :test1 + nl
    t = 12
    test2() # Runtime Error!
    see "After Error!" +nl
    see "t = " see t see nl
    see "x = " see x see nl

```

69.6 Execute Program Line by Line

The next example demonstrates the Trace library usage to execute the program line by line!

```

Trace(:LineByLine)

test1()

```

```
func test1
  x = 10
  see :test1 + nl
  t = 12
  test2()
  see "After Error!" +nl
  see "t = " + t + nl
```

69.7 BreakPoint

The next example demonstrates the Trace library usage to stop at a breakpoint!

```
test1()

func test1
  x = 10
  see :test1 + nl
  t = 12
  BreakPoint()
  see "After breakpoint!" +nl
  see "t = " + t + nl
  see "End of program!" + nl
```

69.8 Disable BreakPoints

The next example demonstrates the Trace library usage and how to disable the Breakpoints!

```
NoBreakPoints()

test1()

func test1
  x = 10
  see :test1 + nl
  t = 12
  BreakPoint()
  see "After breakpoint!" +nl
  see "t = " + t + nl
  see "End of program!" + nl
```

69.9 Using the Interactive Debugger

The next example uses a Breakpoint to open the Interactive Debugger!

```
load "tracelib.ring"

test1()

func test1
  x = 10
  see :test1 + nl
  t = 12
```



```

BreakPoint()
see "After breakpoint!" + nl
see "t = " + t + nl
see "End of program!" + nl

```

Screen Shots:

We have the Interactive Debugger at the Breakpoint!

The screenshot shows the Ring Interactive Debugger window. The source code on the left is as follows:

```

5 # BreakPoint
6
7 load "tracelib" test1
8
9 test1()
10
11 func test1
12     x = 1
13     see : Command (Exit)      : End Program
14     t = 1 Command (Cont)     : Continue Execution
15     Break Command (Locals)   : Print local variables names
16     see "Command (LocalsData) : Print local variables data
17     see "Command (Globals)   : Print global variables names
18     see "We can execute Ring code
19
20

```

The debugger window on the right displays the following menu:

```

=====
Interactive Debugger
=====
Command (Exit)      : End Program
Command (Cont)     : Continue Execution
Command (Locals)    : Print local variables names
Command (LocalsData) : Print local variables data
Command (Globals)   : Print global variables names
We can execute Ring code
=====
code:>

```

The status bar at the bottom indicates the source code file is `B:/ring/ringlibs/tracelib/samples/sample6.ring`.

We can print the variables values

The screenshot shows the Ring Interactive Debugger window. The source code on the left is as follows:

```

6
7 load "tracelib.ring"
8
9 test1()
10
11 func test1
12     x = 10
13     see :test1 + nl
14     t = 12
15     BreakPoint()
16     see "After breakpoint!" + nl
17     see "t = " + t + nl
18     see "End of program!" + nl
19
20

```

The debugger window on the right displays the following menu:

```

=====
Command (Exit)      : End Program
Command (Cont)     : Continue Execution
Command (Locals)    : Print local variables names
Command (LocalsData) : Print local variables data
Command (Globals)   : Print global variables names
We can execute Ring code
=====
code:> localsdata

```

The output of the `localsdata` command is shown below:

Variable	Type	Value
x	NUMBER	10
t	NUMBER	12

The status bar at the bottom indicates the source code file is `B:/ring/ringlibs/tracelib/samples/sample6.ring`.

We can change the variables values then continue execution

```

5 # BreakPoint
6
7 load "tracelib.ring"
8
9 test1()
10
11 func test1
12     x = 10
13     see :test1 + nl
14     t = 12
15     BreakPoint()
16     see "After breakpoint!" +nl
17     see "t = " + t + nl
18     see "End of program!" + nl
19
20

```

```

C:\WINDOWS\system32\cmd.exe - B:\ring\applications\rnote/run "B:/ring/ringlibs/tracelib/samples/sample6.ring"

code:> localsdata

Variable : x                Type : NUMBER        Value : 10
Variable : t                Type : NUMBER        Value : 12

code:> x = 100

code:> t = 200

code:> cont
After breakpoint!
t = 200
End of program!

```

We can run the Interactive Debugger in the Output Window

```

1 # BreakPoint
2
3 load "tracelib.ring"
4
5 test1()
6
7 func test1
8     x = 10
9     see :test1 + nl
10    t = 12
11    BreakPoint()
12    see "After breakpoint!" +nl
13    see "t = " + t + nl
14    see "End of program!" + nl
15
16

```

```

=====
Interactive Debugger
=====
Command (Exit)      : End Program
Command (Cont)     : Continue Execution
Command (Locals)    : Print local variables names
Command (LocalsData): Print local variables data
Command (Globals)   : Print global variables names
We can execute Ring code
=====

code:> localsdata

Variable : x      Type : NUMBER    Value : 10
Variable : t      Type : NUMBER    Value : 12

code:> t = 100

code:> cont
After breakpoint!
t = 100
End of program!

```

EMBEDDING RING IN RING

In this chapter we will learn about embedding Ring in Ring programs and applications.

70.1 Embedding Ring in Ring without sharing the State

From Ring 1.0 we already have functions for embedding Ring in the C language. Also we can execute Ring code inside Ring programs using the `eval()` function. In this release we provide functions for embedding Ring in Ring programs without sharing the state.

Advantages:

1. Quick integration for Ring programs and applications together without conflicts.
2. Execute and run Ring code in safe environments that we can trace.

Example:

```
pState = ring_state_init()
ring_state_runcode(pState, "See 'Hello, World!'+nl")
ring_state_runcode(pState, "x = 10")

pState2 = ring_state_init()
ring_state_runcode(pState2, "See 'Hello, World!'+nl")
ring_state_runcode(pState2, "x = 20")

ring_state_runcode(pState, "see x +nl")
ring_state_runcode(pState2, "see x +nl")

v1 = ring_state_findvar(pState, "x")
v2 = ring_state_findvar(pState2, "x")

see v1[3] + nl
see v2[3] + nl

ring_state_delete(pState)
ring_state_delete(pState2)
```

Output:

```
Hello, World!
Hello, World!
10
20
10
20
```

70.2 Serial Execution of Programs

We can execute application after another application using `ring_state_main()`

Example:

```
chdir(exefolder()+"/../applications/formdesigner")
ring_state_main('formdesigner.ring')
chdir(exefolder()+"/../applications/cards")
ring_state_main('cards.ring')
```

70.3 ring_state_setvar()

Using `ring_state_setvar()` we can set variables value

The value could be (String, Number, List or C Pointer)

We need this function to quickly pass lists and C pointers to the Sub Ring Environment

Syntax:

```
ring_state_setvar(oState,cVariableName,Value)
```

Example:

```
load "guilib.ring"

myapp  = null
win    = null

func main
    myapp = new QApplication {
        win = new QWidget() {
            setWindowTitle("Advanced Example on using ring_state_setvar()")
            move(100,100)
            resize(600,400)
            new QPushButton(win) {
                setText("Test")
                setClickEvent("Test()")
            }
            # We need this because using load 'guilib.ring' in the sub environment
            # Will create timers by Qt and closing the window will not be enough
            # To close the application
            oFilter = new QAllEvents(win)
            oFilter.setCloseEvent("myapp.quit()")
            win.installEventFilter(oFilter)
            show()
        }
        exec()
    }

func test
    pState = ring_state_init()
    ring_state_runcode(pstate,"load 'guilib.ring'")
    ring_state_runcode(pState,"x = NULL")
    # Pass String
    ring_state_setvar(pState,"x","hello")
```

```

        ring_state_runcode(pState, "? x")
# Pass Number
        ring_state_setvar(pState, "x", 100)
        ring_state_runcode(pState, "? x")
# Pass List
        ring_state_setvar(pState, "x", ["one", "two", "three"])
        ring_state_runcode(pState, "? x")
# Pass Object
# We can't pass the Ring Object (win)
# Because Objects store pointers to the Class Information
# And the class is related to the Parent Ring Environment
# And the sub Ring environment can't access it
# But we can pass C pointers like win.pObject
        ring_state_setvar(pState, "x", win.pObject)
# Now we create the object again but using the same C pointer
# So we have access to the Same window in the parent Ring enviroment
        ring_state_runcode(pState, "
            new QWidget {
                pObject = x
                setWindowTitle('Message from the Sub Ring Environment')
            }
        ")
ring_state_delete(pState)

```

EXTENSION USING THE C/C++ LANGUAGES

We can extend the Ring Virtual Machine (RingVM) by adding new functions written in the C programming language or C++. The RingVM comes with many functions written in C that we can call like any Ring function.

We can extend the language by writing new functions then rebuilding the RingVM again, or we can create shared library (DLL/So) file to extend the RingVM without the need to rebuild it.

The Ring language source code comes with two files to add new modules to the RingVM, ring_ext.h and ring_ext.c

71.1 ring_ext.h

The file ring_ext.h contains constants that we can change to include/exclude modules during the build process.

```
#ifndef ringext_h
#define ringext_h
/* Constants */
#define RING_VM_LISTFUNCS      1
#define RING_VM_REFMETA       1
#define RING_VM_MATH          1
#define RING_VM_FILE          1
#define RING_VM_OS            1
#define RING_VM_MYSQL         1
#define RING_VM_ODBC          1
#define RING_VM_OPENSSL       1
#define RING_VM_CURL          1
#define RING_VM_DLL           1
#endif
```

71.2 ring_ext.c

The file ring_ext.c check constants defined in ring_ext.h before calling the start-up function in each module.

Each module contains a function that register the module functions in the RingVM.

```
#include "ring.h"

void ring_vm_extension ( RingState *pRingState )
{
    /* Reflection and Meta-programming */
    #if RING_VM_REFMETA
        ring_vm_refmeta_loadfunctions(pRingState);
    #endif
}
```

```

/* List Functions */
#if RING_VM_LISTFUNCS
    ring_vm_listfuncs_loadfunctions(pRingState);
#endif
/* Math */
#if RING_VM_MATH
    ring_vm_math_loadfunctions(pRingState);
#endif
/* File */
#if RING_VM_FILE
    ring_vm_file_loadfunctions(pRingState);
#endif
/* OS */
#if RING_VM_OS
    ring_vm_os_loadfunctions(pRingState);
#endif
/* MySQL */
#if RING_VM_MYSQL
    ring_vm_mysql_loadfunctions(pRingState);
#endif
/* ODBC */
#if RING_VM_ODBC
    ring_vm_odbc_loadfunctions(pRingState);
#endif
/* OPENSSL */
#if RING_VM_OPENSSL
    ring_vm_openssl_loadfunctions(pRingState);
#endif
/* CURL */
#if RING_VM_CURL
    ring_vm_curl_loadfunctions(pRingState);
#endif
/* DLL */
#if RING_VM_DLL
    ring_vm_dll_loadfunctions(pRingState);
#endif
}

```

71.3 Module Organization

Each module starts by include the ring header file (ring.h). This files contains the Ring API that we can use to extend the RingVM.

Each module comes with a function to register the module functions in the RingVM The registration is done by using ring_vm_funcregister() function.

The ring_vm_funcregister() function takes two parameters, the first is the function name that will be used by Ring programs to call the function. The second parameter is the function pointer in the C program.

for example, the ring_vmmath.c module contains the next code to register the module functions

```

#include "ring.h"

void ring_vm_math_loadfunctions ( RingState *pRingState )
{
    ring_vm_funcregister("sin", ring_vm_math_sin);
    ring_vm_funcregister("cos", ring_vm_math_cos);
}

```

```

ring_vm_funcregister("tan", ring_vm_math_tan);
ring_vm_funcregister("asin", ring_vm_math_asin);
ring_vm_funcregister("acos", ring_vm_math_acos);
ring_vm_funcregister("atan", ring_vm_math_atan);
ring_vm_funcregister("atan2", ring_vm_math_atan2);
ring_vm_funcregister("sinh", ring_vm_math_sinh);
ring_vm_funcregister("cosh", ring_vm_math_cosh);
ring_vm_funcregister("tanh", ring_vm_math_tanh);
ring_vm_funcregister("exp", ring_vm_math_exp);
ring_vm_funcregister("log", ring_vm_math_log);
ring_vm_funcregister("log10", ring_vm_math_log10);
ring_vm_funcregister("ceil", ring_vm_math_ceil);
ring_vm_funcregister("floor", ring_vm_math_floor);
ring_vm_funcregister("fabs", ring_vm_math_fabs);
ring_vm_funcregister("pow", ring_vm_math_pow);
ring_vm_funcregister("sqrt", ring_vm_math_sqrt);
ring_vm_funcregister("unsigned", ring_vm_math_unsigned);
ring_vm_funcregister("decimals", ring_vm_math_decimals);
ring_vm_funcregister("murmur3hash", ring_vm_math_murmur3hash);
}

```

Tip: Remember that the function `ring_vm_math_loadfunctions()` will be called by the `ring_vm_extension()` function (in the `ring_ext.c` file).

71.4 Function Structure

Each module function may contains the next steps

- 1 - Check Parameters Count
- 2 - Check Parameters Type
- 3 - Get Parameters Values
- 4 - Execute Code/Call Functions
- 5 - Return Value

The structure is very similar to any function (Input - Process - Output) But here we will use the Ring API for the steps 1,2,3 and 5.

71.5 Check Parameters Count

We can check the parameters count using the `RING_API_PARACOUNT` macro.

We can compare `RING_API_PARACOUNT` with any numeric value using `==` or `!=` operators.

Example:

```

if ( RING_API_PARACOUNT != 1 ) {
    /* code */
}

```

Example:


```
if ( RING_API_PARACOUNT == 1 ) {
    /* code */
}
```

71.6 Display Error Message

We can display error messages using the `RING_API_ERROR()` function.

The function will display the error and end the execution of the program.

Note: the behaviour of this function can be changed by the Ring code using Try/Catch/Done statements, so in your C code, use Return after this function.

Syntax:

```
RING_API_ERROR(const char *cErrorMsg);
```

The Ring API comes with some of predefined error messages that we can use

```
#define RING_API_MISS1PARA "Bad parameters count, the function expect one parameter"
#define RING_API_MISS2PARA "Bad parameters count, the function expect two parameters"
#define RING_API_MISS3PARA "Bad parameters count, the function expect three parameters"
#define RING_API_MISS4PARA "Bad parameters count, the function expect four parameters"
#define RING_API_BADPARATYPE "Bad parameter type!"
#define RING_API_BADPARACOUNT "Bad parameters count!"
#define RING_API_BADPARARANGE "Bad parameters value, error in range!"
#define RING_API_NOTPOINTER "Error in parameter, not pointer!"
#define RING_API_NULLPOINTER "Error in parameter, NULL pointer!"
#define RING_API_EMPTYLIST "Bad parameter, empty list!"
```

71.7 Check Parameters Type

We can check the parameter type using the next functions

```
int RING_API_ISNUMBER(int nParameterNumber);
int RING_API_ISSTRING(int nParameterNumber);
int RING_API_ISLIST(int nParameterNumber);
int RING_API_ISPOINTER(int nParameterNumber);
```

The output of these functions will be 1 (True) or 0 (False).

71.8 Get Parameters Values

We can get parameters values using the next functions

```
double RING_API_GETNUMBER(int nParameterNumber);
const char *RING_API_GETSTRING(int nParameterNumber);
int RING_API_GETSTRINGSIZE(int nParameterNumber);
List *RING_API_GETLIST(int nParameterNumber);
void *RING_API_GETCPOINTER(int nParameterNumber, const char *cPoinerType);
int RING_API_GETPOINTERTYPE(int nParameterNumber);
```

71.9 Return Value

We can return values from our function using the next functions.

```
RING_API_RETNUMBER(double nValue);
RING_API_RETSTRING(const char *cString);
RING_API_RETSTRING2(const char *cString,int nStringSize);
RING_API_RETLIST(List *pList);
RING_API_RETCPINTER(void *pValue,const char *cPointerType);
```

71.10 Function Prototype

When we define new function to be used for RingVM extension, we use the next prototype

```
void my_function_name( void *pPointer );
```

or we can use the RING_FUNC() Macro

```
RING_FUNC(my_function_name);
```

71.11 Sin() Function Implementation

The next code represents the sin() function implementation using the Ring API and the sin() C function.

```
void ring_vm_math_sin ( void *pPointer )
{
    if ( RING_API_PARACOUNT != 1 ) {
        RING_API_ERROR(RING_API_MISS1PARA);
        return ;
    }
    if ( RING_API_ISNUMBER(1) ) {
        RING_API_RETNUMBER(sin(RING_API_GETNUMBER(1)));
    } else {
        RING_API_ERROR(RING_API_BADPARATYPE);
    }
}
```

71.12 Fopen() and Fclose() Functions Implementation

The next code represents the fopen() function implementation using the Ring API and the fopen() C Function.

The function takes two parameters, the first parameter is the file name as string. The second parameter is the mode as string.

In the file ring_vmfile.h we have some constants to use as the pointer type like

```
#define RING_VM_POINTER_FILE      "file"
#define RING_VM_POINTER_FILEPOS  "filepos"
```

The function implementation in ring_vmfile.c

```

void ring_vm_file_fopen ( void *pPointer )
{
    FILE *fp ;
    if ( RING_API_PARACOUNT != 2 ) {
        RING_API_ERROR(RING_API_MISS2PARA);
        return ;
    }
    if ( RING_API_ISSTRING(1) && RING_API_ISSTRING(2) ) {
        fp = fopen(RING_API_GETSTRING(1),RING_API_GETSTRING(2));
        RING_API_RETCPINTER(fp,RING_VM_POINTER_FILE);
    } else {
        RING_API_ERROR(RING_API_BADPARATYPE);
    }
}

```

The next code represents the fclose() function implementation

```

void ring_vm_file_fclose ( void *pPointer )
{
    FILE *fp ;
    if ( RING_API_PARACOUNT != 1 ) {
        RING_API_ERROR(RING_API_MISS1PARA);
        return ;
    }
    if ( RING_API_ISPOINTER(1) ) {
        fp = (FILE *) RING_API_GETCPINTER(1,RING_VM_POINTER_FILE) ;
        if ( fp != NULL ) {
            RING_API_RETNUMBER(fclose(fp));
            RING_API_SETNULLPOINTER(1);
        }
    } else {
        RING_API_ERROR(RING_API_BADPARATYPE);
    }
}

```

From fopen() and fclose() implementation we learned

- 1 - how to return C pointer using RING_API_RETCPINTER() function
- 2 - how to check if the parameter is a pointer using the RING_API_ISPOINTER() function
- 3 - how to get C pointer value using the RING_API_GETCPINTER() function
- 4 - how to set the C pointer variable (in RingVM) to NULL using the RING_API_SETNULLPOINTER() function

71.13 Ring API - List Functions

In this section we will learn about the list functions provided by the Ring API to create new lists and manipulate the list items.

```

List * ring_list_new ( int nSize ) ;
void ring_list_newitem ( List *pList ) ;
Item * ring_list_getitem ( List *pList,int index ) ;
List * ring_list_delete ( List *pList ) ;
void ring_list_deleteitem ( List *pList,int index ) ;
void ring_list_print ( List *pList ) ;
int ring_list_gettype ( List *pList, int index ) ;
void ring_list_setint ( List *pList, int index ,int number ) ;

```

```

void ring_list_addint ( List *pList, int x ) ;
void ring_list_setpointer ( List *pList, int index, void *pValue ) ;
void ring_list_addpointer ( List *pList, void *pValue ) ;
void ring_list_setfuncpointer ( List *pList, int index, void (*pFunc)(void *) ) ;
void ring_list_addfuncpointer ( List *pList, void (*pFunc)(void *) ) ;
int ring_list_isfuncpointer ( List *pList, int index ) ;
void ring_list_setdouble ( List *pList, int index, double number ) ;
void ring_list_adddouble ( List *pList, double x ) ;
void ring_list_setstring ( List *pList, int index, const char *str ) ;
void ring_list_setstring2 ( List *pList, int index, const char *str, int nStrSize ) ;
void ring_list_addstring ( List *pList, const char *str ) ;
void ring_list_addstring2 ( List *pList, const char *str, int nStrSize ) ;
List * ring_list_newlist ( List *pList ) ;
List * ring_list_getlist ( List *pList, int index ) ;
void ring_list_setlist ( List *pList, int index ) ;
void ring_list_setactiveitem ( List *pList, Items *pItems, int index ) ;
void ring_list_copy ( List *pNewList, List *pList ) ;
int ring_list_isnumber ( List *pList, int index ) ;
int ring_list_isstring ( List *pList, int index ) ;
int ring_list_islist ( List *pList, int index ) ;
int ring_list_ispointer ( List *pList, int index ) ;
void ring_list_deleteallitems ( List *pList ) ;
void ring_list_insertitem ( List *pList, int x ) ;
void ring_list_insertint ( List *pList, int nPos, int x ) ;
void ring_list_insertdouble ( List *pList, int nPos, double x ) ;
void ring_list_insertpointer ( List *pList, int nPos, void *pValue ) ;
void ring_list_insertstring ( List *pList, int nPos, const char *str ) ;
void ring_list_insertstring2 ( List *pList, int nPos, const char *str, int nStrSize ) ;
void ring_list_insertfuncpointer ( List *pList, int nPos, void (*pFunc)(void *) ) ;
List * ring_list_insertlist ( List *pList, int nPos ) ;
int ring_list_isiteminsidelist ( List *pList, Item *pItem ) ;
int ring_list_findstring ( List *pList, const char *str, int nColumn ) ;
int ring_list_finddouble ( List *pList, double nNum1, int nColumn ) ;
void ring_list_sortnum ( List *pList, int left, int right, int nColumn ) ;
void ring_list_sortstr ( List *pList, int left, int right, int nColumn ) ;
int ring_list_binarysearchnum ( List *pList, double nNum1, int nColumn ) ;
int ring_list_binarysearchstr ( List *pList, const char *cFind, int nColumn ) ;
void ring_list_swap ( List *pList, int x, int y ) ;
double ring_list_getdoublecolumn ( List *pList, int nIndex, int nColumn ) ;
char * ring_list_getstringcolumn ( List *pList, int nIndex, int nColumn ) ;
void ring_list_genarray ( List *pList ) ;
void ring_list_deletearray ( List *pList ) ;
void ring_list_genhashtable ( List *pList ) ;
void ring_list_genhashtable2 ( List *pList ) ;
void ring_list_refcopy ( List *pNewList, List *pList ) ;
void ring_list_clear ( List *pList ) ;
/* Macro */
ring_list_isdouble(pList, index)
ring_list_isint(pList, index)
ring_list_deletelastitem(x)
ring_list_gethashtable(x)
ring_list_getint(pList, index)
ring_list_getpointer(pList, index)
ring_list_getfuncpointer(pList, index)
ring_list_callfuncpointer(pList, index, x)
ring_list_getdouble(pList, index)
ring_list_getstring(pList, index)
ring_list_getstringobject(pList, index)

```

```
ring_list_getstringsize(pList,index)
ring_list_getsize(x) (x->nSize)
```

71.14 Ring API - String Functions

In this section we will learn about the string functions provided by the Ring API to create new string and manipulate the string content.

```
String * ring_string_new ( const char *str ) ;
String * ring_string_new2 ( const char *str,int nStrSize ) ;
String * ring_string_delete ( String *pString ) ;
int ring_string_size ( String *pString ) ;
void ring_string_set ( String *pString,const char *str ) ;
void ring_string_set2 ( String *pString,const char *str,int nStrSize ) ;
void ring_string_add ( String *pString,const char *str ) ;
void ring_string_add2 ( String *pString,const char *str,int nStrSize ) ;
void ring_string_print ( String *pString ) ;
void ring_string_setfromint ( String *pString,int x ) ;
char * ring_string_lower ( char *cStr ) ;
char * ring_string_upper ( char *cStr ) ;
char * ring_string_lower2 ( char *cStr,int nStrSize ) ;
char * ring_string_upper2 ( char *cStr,int nStrSize ) ;
char * ring_string_find ( char *cStr1,char *cStr2 ) ;
char * ring_string_find2 ( char *cStr1,int nStrSize1,char *cStr2,int nStrSize2 ) ;
/* Macro */
ring_string_tolower(x)
ring_string_toupper(x)
ring_string_get(x)
```

71.15 MySQL_Columns() Function Implementation

The next code presents the MySQL_Columns() function implementation.

This function returns table columns information.

```
void ring_vm_mysql_columns ( void *pPointer )
{
    MYSQL *con ;
    MYSQL_RES *result ;
    int nColumns,x ;
    MYSQL_ROW row ;
    MYSQL_FIELD *field ;
    List *pList, *pList2 ;
    if ( RING_API_PARACOUNT != 1 ) {
        RING_API_ERROR(RING_API_MISS1PARA);
        return ;
    }
    if ( RING_API_ISPOINTER(1) ) {
        con = (MYSQL *) RING_API_GETCPOINTER(1,RING_VM_POINTER_MYSQL) ;
        if ( con == NULL ) {
            return ;
        }
        result = mysql_store_result(con);
        if ( result == NULL ) {
```

```

        RING_API_RETNUMBER(0);
        return ;
    }
    pList = RING_API_NEWLIST ;
    nColumns = mysql_num_fields(result);
    if ( row = mysql_fetch_row(result) ) {
        while ( field = mysql_fetch_field(result) ) {
            pList2 = ring_list_newlist(pList);
            ring_list_addstring(pList2, field->name);
            ring_list_adddouble(pList2, field->length);
            ring_list_adddouble(pList2, field->type);
            ring_list_adddouble(pList2, field->flags);
        }
    }
    mysql_free_result(result);
    RING_API_RETLIST(pList);
} else {
    RING_API_ERROR(RING_API_BADPARATYPE);
}
}

```

Lists are of type List, in the previous function we declared two pointers of type List using List *pList, *pList2;

Note: The function uses RING_API_NEWLIST to create new list instead of ring_list_new() to create the list in Temp. Memory related to the function scope. This way we can return the list from the function. Also we don't delete the list, if it's stored in a variable by Ring Code it will be saved, if not it will be automatically deleted by RingVM.

The list can contains sub lists, we used the function ring_list_newlist() to create a sublist.

The function ring_list_addstring() is used to add string items to the list/sublist.

The function ring_list_adddouble() is used to add numeric items to the list/sublist.

Note: All numeric items in lists returned from RingVM extension functions must be of type double and added to the list using ring_list_adddouble() function.

We return the list from the extension function using the RING_API_RETLIST() function.

71.16 Dynamic/Shared Libraries (DLL/So/Dylib) and LoadLib() function

Instead of rebuilding the RingVM after writing new functions using C/C++ and the Ring API, we can create a DLL/So/Dylib file and dynamically use the functions provided by this file in the runtime using the LoadLib() function.

Dynamic library example in C

```

#include "ring.h"

RING_DLL __declspec(dllexport)

RING_FUNC(ring_ringlib_dlfunc)
{
    printf("Message from dlfunc");
}

```

```
RING_DLL void ringlib_init(RingState *pRingState)
{
    ring_vm_funcregister("dlfunc", ring_ringlib_dlfunc);
}
```

the idea is to create the ringlib_init() function, this function will be called by the RingVM when we use the generated DLL file though the LoadLib() function.

Inside the ringlib_init() function we can register the module function or call a function that do the registration process for all of the module functions.

The next Ring code demonstrates how to use the DLL library during the runtime.

```
See "Dynamic DLL" + NL
LoadLib("ringlib.dll")
dlfunc()
```

Output:

```
Dynamic DLL
Message from dlfunc
```

EMBEDDING RING LANGUAGE IN C/C++ PROGRAMS

We can use the Ring language from C/C++ programs using the next functions

```
RingState *ring_state_init();
ring_state_runcode(RingState *pState, const char *cCode);
ring_state_delete(RingState *pState);
```

72.1 Ring State

The idea is to use the ring_state_init() to create new state for the Ring Language then call the ring_state_runcode() function to execut Ring code using the same state. When we are done, we call the ring_state_delete() to free the memory.

Example:

```
#include "ring.h"
#include "stdlib.h"
int main(int argc, char *argv[])
{
    RingState *pState = ring_state_init();
    printf("welcome\n");
    ring_state_runcode(pState, "see 'hello world from the ring programming language'+nl");
    ring_state_delete(pState);
}
```

Output:

```
welcome
hello world from the ring programming language
```

72.2 Ring State Functions

The Ring API comes with the next functions to create and delete the state. Also we have functions to create new variables and get variables values.

```
RingState * ring_state_init ( void ) ;
RingState * ring_state_delete ( RingState *pRingState ) ;
void ring_state_runcode ( RingState *pRingState, const char *cStr ) ;
List * ring_state_findvar ( RingState *pRingState, const char *cStr ) ;
List * ring_state_newvar ( RingState *pRingState, const char *cStr ) ;
void ring_state_main ( int argc, char *argv[] ) ;
```



```
void ring_state_runfile ( RingState *pRingState, const char *cFileName ) ;
void ring_state_runobjectfile ( RingState *pRingState, const char *cFileName ) ;
```

72.3 Ring State Variables

We can create more than one ring state in the same program and we can create and modify variable values.

To get the variable list we can use the ring_state_findvar() function.

To create new variable we can use the ring_state_newvar() function.

Example:

```
#include "ring.h"
#include "stdlib.h"

int main(int argc, char *argv[])
{
    List *pList;

    RingState *pState = ring_state_init();
    RingState *pState2 = ring_state_init();

    printf("welcome\n");
    ring_state_runcode(pState, "see 'hello world from the ring programming language'+nl");

    printf("Again from C we will call ring code\n");
    ring_state_runcode(pState, "for x = 1 to 10 see x + nl next");

    ring_state_runcode(pState2, "for x = 1 to 5 see x + nl next");

    printf("Now we will display the x variable value from ring code\n");
    ring_state_runcode(pState, "see 'x value : ' + x + nl ");
    ring_state_runcode(pState2, "see 'x value : ' + x + nl ");

    pList = ring_state_findvar(pState, "x");

    printf("Printing Ring variable value from C , %.0f\n",
           ring_list_getdouble(pList, RING_VAR_VALUE));

    printf("now we will set the ring variable value from C\n");
    ring_list_setdouble(pList, RING_VAR_VALUE, 20);

    ring_state_runcode(pState, "see 'x value after update : ' + x + nl ");

    pList = ring_state_newvar(pState, "v1");
    ring_list_setdouble(pList, RING_VAR_VALUE, 10);

    pList = ring_state_newvar(pState, "v2");
    ring_list_setdouble(pList, RING_VAR_VALUE, 20);

    ring_state_runcode(pState, "see 'v1 + v2 = ' see v1+v2 see nl");

    ring_state_runcode(pState, "see 'end of test' + nl");

    ring_state_delete(pState);
```

```
ring_state_delete(pState2);  
}
```

Output:

```
welcome  
hello world from the ring programming language  
Again from C we will call ring code  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
1  
2  
3  
4  
5  
Now we will display the x variable value from ring code  
x value : 11  
x value : 6  
Printing Ring variable value from C , 11  
now we will set the ring variable value from C  
x value after update : 20  
v1 + v2 = 30  
end of test
```

CODE GENERATOR FOR WRAPPING C/C++ LIBRARIES

In this chapter we will learn how to use the code generator to wrap C/C++ Libraries to use it in our Ring applications.

73.1 Using the tool

The code generator program is `parsec.ring` that can be executed as any ring code using the ring language.

URL : <https://github.com/ring-lang/ring/tree/master/extensions/codegen>

for example to read a configuration file called `test.cf` to generate the source code file `test.c` run `parsec.ring` as in the next command

```
ring parsec.ring test.cf test.c
```

73.2 Configuration file

The configuration file (`*.cf`) is the input file that we pass to the code generator. This file determine the functions prototypes that we need to use from a C/C++ library.

Writing configuration files is simple according to the next rules

73.3 Using the function prototype

- To generate code that wraps a C function, we just write the C function prototype

Example:

```
ALLEGRO_DISPLAY *al_create_display(int w, int h)
void al_destroy_display(ALLEGRO_DISPLAY *display)
int al_get_new_display_flags(void)
void al_set_new_display_flags(int flags)
int al_get_new_display_option(int option, int *importance)
```

The previous example will guide the code generator to generate 5 functions that wraps the `al_create_display()`, `al_destroy_display()`, `al_get_new_display_flags()`, `al_set_new_display_flags()` and `al_get_new_display_option()` functions.

The generated code will be as in the next example

```

RING_FUNC(ring_al_create_display)
{
    if ( RING_API_PARACOUNT != 2 ) {
        RING_API_ERROR(RING_API_MISS2PARA);
        return ;
    }
    if ( ! RING_API_ISNUMBER(1) ) {
        RING_API_ERROR(RING_API_BADPARATYPE);
        return ;
    }
    if ( ! RING_API_ISNUMBER(2) ) {
        RING_API_ERROR(RING_API_BADPARATYPE);
        return ;
    }
    RING_API_RETCPINTER(al_create_display( (int ) RING_API_GETNUMBER(1),
                                           (int ) RING_API_GETNUMBER(2)), "ALLEGRO_DISPLAY");
}

RING_FUNC(ring_al_destroy_display)
{
    if ( RING_API_PARACOUNT != 1 ) {
        RING_API_ERROR(RING_API_MISS1PARA);
        return ;
    }
    if ( ! RING_API_ISPOINTER(1) ) {
        RING_API_ERROR(RING_API_BADPARATYPE);
        return ;
    }
    al_destroy_display((ALLEGRO_DISPLAY *) RING_API_GETCPINTER(1, "ALLEGRO_DISPLAY"));
}

RING_FUNC(ring_al_get_new_display_flags)
{
    if ( RING_API_PARACOUNT != 0 ) {
        RING_API_ERROR(RING_API_BADPARACOUNT);
        return ;
    }
    RING_API_RETNUMBER(al_get_new_display_flags());
}

RING_FUNC(ring_al_set_new_display_flags)
{
    if ( RING_API_PARACOUNT != 1 ) {
        RING_API_ERROR(RING_API_MISS1PARA);
        return ;
    }
    if ( ! RING_API_ISNUMBER(1) ) {
        RING_API_ERROR(RING_API_BADPARATYPE);
        return ;
    }
    al_set_new_display_flags( (int ) RING_API_GETNUMBER(1));
}

RING_FUNC(ring_al_get_new_display_option)

```

```
{
    if ( RING_API_PARACOUNT != 2 ) {
        RING_API_ERROR(RING_API_MISS2PARAM);
        return ;
    }
    if ( ! RING_API_ISNUMBER(1) ) {
        RING_API_ERROR(RING_API_BADPARATYPE);
        return ;
    }
    if ( ! RING_API_ISSTRING(2) ) {
        RING_API_ERROR(RING_API_BADPARATYPE);
        return ;
    }
    RING_API_RETNUMBER(al_get_new_display_option( (int ) RING_API_GETNUMBER(1),
                                                RING_API_GETINTPOINTER(2)));
    RING_API_ACCEPTINTVALUE(2) ;
}
```

from the previous example we can see how much of time and effort is saved using the Code Generator.

73.4 Adding code to the generated code

- To generate code directly type it between `<code>` and `</code>`

Example :

```
<code>
/* some C code will be written here */
</code>
```

We use this feature when we need to do something without the help of the code generator. for example including header files and defining constants using Macro.

73.5 Prefix for Functions Names

- To determine a prefix in all of the functions names type it between `<funcstart>` and `</funcstart>` for example when we wrap the Allegro game programming library and we need all of the library functions to start with “al” we type the next code in the configuration file

```
<funcstart>
al
</funcstart>
```

73.6 Generate function to wrap structures

- To generate functions that wrap structures (create/delete/get structure members)

just type the structures names between `<struct>` and `</struct>` also after the structure name you can type the structure members between `{ }` separated by comma.

Example

```
<struct>
ALLEGRO_COLOR
ALLEGRO_EVENT { type , keyboard.keycode , mouse.x , mouse.y }
</struct>
```

from the previous example we will generate two function to create/delete the structure ALLEGRO_COLOR Also we will generate two functions to create/delete the structure ALLEGRO_EVENT and four functions to get the structure ALLEGRO_EVENT members (type, keyboard.keycode, mouse.x, mouse.y).

73.7 Determine Structure Members Types

You can determine the pointer name before the structure member name.

Example:

```
SDL_Surface {flags,SDL_PixelFormat *format,w,h,pitch,void *pixels}
```

73.8 Defining Constants

You can define constants using <constant> and </constant>

The generator will generate the required functions to get the constant values

And will define the constants to be used with the same name in Ring code using *.rh file that will be generated too.

rh = Ring Header

Example:

```
<constant>
MIX_DEFAULT_FORMAT
SDL_QUIT
SDL_BUTTON_LEFT
SDL_BUTTON_MIDDLE
SDL_BUTTON_RIGHT
</constant>
```

Note: You will need to pass the *.rh file name to parsec.ring after the generated source file name.

Example:

```
ring ..\codegen\parsec.ring libSDL.cf ring_libSDL.c ring_libSDL.rh
```

73.9 Register New Functions

We can register functions by typing the function prototype between <register> and </register> We need this feature only when we don't provide the function prototype as input directly where we need to write the code of this function.

Example:

```
<register>
void al_exit(void)
</register>
```

```

<code>
RING_FUNC (ring_al_exit)
{
    if ( RING_API_PARACOUNT != 0 ) {
        RING_API_ERROR (RING_API_BADPARACOUNT);
        return ;
    }
    exit(0);
}
</code>

```

In the previous example we register the `al_exit()` function. This function is not part of the Allegro Library, it's just an extra function that we need to add. Then the code if this function is written inside `<code>` and `</code>`. This function call the `exit()` function from the C language library.

73.10 Writing comments in the configuration file

- To type comments just type it between `<comment>` and `</comment>`

Example:

```

<comment>
configuration files
</comment>

```

73.11 Executing code during code generation

- To ask from the code generator to execute Ring code during reading the configuration file, just

write the code between `<runcode>` and `</runcode>`

Example:

```

<runcode>
aNumberTypes + "al_fixed"
</runcode>

```

The previous line of code add the string "al_fixed" to the list `aNumberTypes`, This list contains types that can be considered as numbers when the code generator find it in the function prototype.

73.12 Enum and Numbers

We have the list `aEnumTypes` to use for adding each Enumeration we uses in the functions prototype.

Example:

```

<runcode>
aNumberTypes + "qreal"
aNumberTypes + "qint64"
aEnumTypes + "Qt::GestureType"
aEnumTypes + "Qt::GestureFlag"
</runcode>

```

73.13 Filtering using Expressions

using `<filter>` and `</filter>` we can include/exclude parts of the configuration file based on a condition, for example

```
<filter> iswindows()
    ... functions related to windows
</filter>
```

73.14 Constants Type

The default type for constant is Number But Some constants may be another type, for example (pointer : void *)

before using `<constant>` and `</constant>` we can use `<runcode>` and `</runcode>` to determine the constant type using two global variables used by the code generator.

The first variable is `$nDefaultConstantType` which can be `* C_CONSTANT_TYPE_NUMBER * C_CONSTANT_TYPE_STRING * C_CONSTANT_TYPE_POINTER`

if we are using `C_CONSTANT_TYPE_POINTER` then we will need the second global variable which is `$cDefaultConstantPointerType` to determine the pointer type.

Example :

The next example uses this feature to define constants in the FreeGLUT library

```
<runcode>
$nDefaultConstantType = C_CONSTANT_TYPE_POINTER
$cDefaultConstantPointerType = "void"
</runcode>
<constant>
    GLUT_STROKE_ROMAN
    GLUT_STROKE_MONO_ROMAN
    GLUT_BITMAP_9_BY_15
    GLUT_BITMAP_8_BY_13
    GLUT_BITMAP_TIMES_ROMAN_10
    GLUT_BITMAP_TIMES_ROMAN_24
    GLUT_BITMAP_HELVETICA_10
    GLUT_BITMAP_HELVETICA_12
    GLUT_BITMAP_HELVETICA_18
</constant>
```

73.15 Configuration file for the Allegro Library

The next configuration file enable us to use the Allegro library functions. The configuration file size is less than 1000 lines. when the code generator take this file as input the generated source code file in the C language will be 12000 lines of code!

We can see this configuration file as a complete example about using the code generator Also we can use it to know the functions that can be used from RingAllegro when you use it to create 2D games!

```
<code>
#define ALLEGRO_NO_MAGIC_MAIN

#include <allegro5/allegro.h>
#include "allegro5/allegro_image.h"
```



```

#include <allegro5/allegro_font.h>
#include <allegro5/allegro_ttf.h>
#include <allegro5/allegro_audio.h>
#include <allegro5/allegro_acodec.h>
#include <allegro5/allegro_opengl.h>
#include <allegro5/allegro_direct3d.h>
#include <allegro5/allegro_color.h>
#include <allegro5/allegro_memfile.h>
#include "allegro5/allegro_native_dialog.h"
#include <allegro5/allegro_physfs.h>
#include <allegro5/allegro_primitives.h>
</code>

<funcstart>
al
</funcstart>

<struct>
ALLEGRO_EVENT { type , keyboard.keycode , mouse.x , mouse.y }
ALLEGRO_TIMEOUT
ALLEGRO_SAMPLE_ID
ALLEGRO_COLOR
</struct>

<register>
void al_exit(void)
</register>

<code>
RING_FUNC(ring_al_exit)
{
    if ( RING_API_PARACOUNT != 0 ) {
        RING_API_ERROR(RING_API_BADPARACOUNT);
        return ;
    }
    exit(0);
}
</code>

int al_init(void)

<comment>
configuration files
</comment>

<runcode>
aNumberTypes + "al_fixed"
</runcode>

ALLEGRO_CONFIG *al_create_config(void)
void al_destroy_config(ALLEGRO_CONFIG *config)
ALLEGRO_CONFIG *al_load_config_file(const char *filename)
ALLEGRO_CONFIG *al_load_config_file_f(ALLEGRO_FILE *file)
bool al_save_config_file(const char *filename, const ALLEGRO_CONFIG *config)
bool al_save_config_file_f(ALLEGRO_FILE *file, const ALLEGRO_CONFIG *config)
void al_add_config_section(ALLEGRO_CONFIG *config, const char *name)

```

Note: we just provided part of the configuration file, for complete copy check the Ring source code distribution.

73.16 Threads Support

Next, another part of the configuration file, it's important because we can learn from it how to add threads to our Ring applications by using a threads library.

The idea is using `ring_vm_mutexfunctions()` and `ring_vm_runcodefromthread()` to execute Ring code.

```
<comment>
Threads
</comment>

<code>
void *al_func_thread(ALLEGRO_THREAD *thread, void *pPointer)
{
    List *pList;
    VM *pVM;
    const char *cStr;
    pList = (List *) pPointer ;
    pVM = (VM *) ring_list_getpointer(pList,2);
    cStr = ring_list_getstring(pList,1);
    ring_vm_runcodefromthread(pVM,cStr);
    ring_list_delete(pList);
    return NULL;
}

RING_FUNC(ring_al_create_thread)
{
    ALLEGRO_THREAD *pThread;
    List *pList;
    if ( RING_API_PARACOUNT != 1 ) {
        RING_API_ERROR(RING_API_MISS1PARA);
        return ;
    }
    if ( ! RING_API_ISSTRING(1) ) {
        RING_API_ERROR(RING_API_BADPARATYPE);
        return ;
    }
    pList = ring_list_new(0);
    ring_list_addstring(pList,RING_API_GETSTRING(1));
    ring_list_addpointer(pList,pPointer);
    ring_vm_mutexfunctions((VM *) pPointer,al_create_mutex,
        al_lock_mutex,al_unlock_mutex,al_destroy_mutex);
    pThread = al_create_thread(al_func_thread, pList);
    al_start_thread(pThread);
    RING_API_RETCPINTER(pThread,"ALLEGRO_THREAD");
}

RING_FUNC(ring_al_run_detached_thread)
{
    List *pList;
    if ( RING_API_PARACOUNT != 1 ) {
        RING_API_ERROR(RING_API_MISS1PARA);
        return ;
    }
    if ( ! RING_API_ISSTRING(1) ) {
```

```

        RING_API_ERROR(RING_API_BADPARATYPE);
        return ;
    }
    pList = ring_list_new(0);
    ring_list_addstring(pList,RING_API_GETSTRING(1));
    ring_list_addpointer(pList,pPointer);
    ring_vm_mutexfunctions((VM *) pPointer,al_create_mutex,
        al_lock_mutex,al_unlock_mutex,al_destroy_mutex);
    al_run_detached_thread(al_func_thread, pList);
}
</code>

<register>
ALLEGRO_THREAD *al_create_thread(void)
void al_run_detached_thread(void)
</register>

void al_start_thread(ALLEGRO_THREAD *thread)
void al_join_thread(ALLEGRO_THREAD *thread, void **ret_value)
void al_set_thread_should_stop(ALLEGRO_THREAD *thread)
bool al_get_thread_should_stop(ALLEGRO_THREAD *thread)
void al_destroy_thread(ALLEGRO_THREAD *thread)
ALLEGRO_MUTEX *al_create_mutex(void)
ALLEGRO_MUTEX *al_create_mutex_recursive(void)
void al_lock_mutex(ALLEGRO_MUTEX *mutex)
void al_unlock_mutex(ALLEGRO_MUTEX *mutex)
void al_destroy_mutex(ALLEGRO_MUTEX *mutex)
ALLEGRO_COND *al_create_cond(void)
void al_destroy_cond(ALLEGRO_COND *cond)
void al_wait_cond(ALLEGRO_COND *cond, ALLEGRO_MUTEX *mutex)

```

73.17 Code Generator Rules for Wrapping C++ Classes

- We can define classes between <class> and </class>
- Between <class> and </class> we set attributes like “name, new, para, parent, codename, passvmpointer and abstract”
- we set the attributes using the style attributename:value or attributename only if no values are required
- The “name” attribute determine the class name in C++ code and this name will be the default name in the Ring code
- The new instruction means that we don’t need new/delete methods
- The parent attribute determine the parent class name
- The codename attribute determine another class name in C++ code
- The passvmpointer instruction means passing the Ring VM pointer to the class constructor when we create new objects, this happens when we set the codename attribute to a class the we will define and this class need the Virtual Machine pointer (for example to use it to execute Ring code from C++ code).
- The abstract instruction means that no new method is required for this class “no objects will be created”.
- Using <nodllstartup> we can avoid #include “ring.h”, We need this to write our startup code.
- Using <libinitfunc> we can change the function name that register the library functions
- Using <ignorepointertype> we can ignore pointer type check

- Using the aStringTypes list when can defined new types that treated like const char *
- Using the aBeforeReturn list when can define code that is inserted after the variable name when we return that variable from a function
- Using the aNewMethodName list we can define another method name to be used in Ring code when we call the C++ method. this feature is required because some C++ method may be identical to Ring Keywords like “load”, “next”, “end” and “done”.
- in method prototype - when we use @ in the method name, we mean that we have the same method with different parameters (As in C++)

73.18 Using configuration file that wrap C++ Library

To run the code generator to generate code for using C++ library in the Ring application, we can do that as we did with using C libraries but here we will generate *.cpp file instead of *.c file. Also we will determine another file to be generated (.ring). This file will contains classes in Ring code that wraps C++ functions for using C++ classes and objects.*

```
ring parsec.ring qt.cf ring_qt.cpp ring_qt.ring
```

73.19 Configuration file for the Qt Framework

The next configuration file is used to wrap many Qt classes The configuration file is around 3500 lines and generate C++ code around 56000 lines and generate also Ring code around 9000 lines.

```
<nodllstartup>

<libinitfunc> ring_qt_start

<ignorecpointertype>

<code>

extern "C" {
    #include "ring.h"
}

#include "ring_qt.h"
#include "gpushbutton.h"
#include "gaction.h"
#include "glineedit.h"
#include "gttextedit.h"
#include "glistwidget.h"
#include "gtreeview.h"
#include "gtreewidget.h"
#include "gcombobox.h"
#include "gtabwidget.h"
#include "gtablewidget.h"
#include "gprogressbar.h"
#include "gspinbox.h"
#include "gslider.h"
#include "gdial.h"
#include "gwebview.h"
#include "gcheckbox.h"
```

```

#include "gradiobutton.h"
#include "gbuttongroup.h"
#include "gvideowidget.h"
#include "gtimer.h"
#include "gtcpserver.h"
#include "giodevice.h"
#include "gabstracksocket.h"
#include "gtcpsocket.h"
#include "gcolordialog.h"
#include "gallevents.h"
#include <QApplication>
#include <QObject>
#include <QWidget>
#include <QLabel>
#include <QPixmap>
#include <QIcon>
#include <QSize>
#include <QPushButton>
#include <QMainWindow>
#include <QVBoxLayout>
#include <QHBoxLayout>
#include <QLineEdit>
#include <QTextEdit>
#include <QListWidget>
#include <QTreeView>
#include <QDir>
#include <QFileSystemModel>
#include <QTreeWidget>
#include <QTreeWidgetItem>
#include <QComboBox>
#include <QVariant>
#include <QMenuBar>
#include <QMenu>
#include <QToolBar>
#include <QMainWindow>
#include <QStatusBar>
#include <QDockWidget>
#include <QTabWidget>
#include <QTableWidget>
#include <QTableWidgetItem>
#include <QSizePolicy>
#include <QFrame>
#include <QAbstractScrollArea>
#include <QAbstractItemView>
#include <QProgressBar>
#include <QSpinBox>
#include <QSlider>
#include <QAbstractSlider>
#include <QDateEdit>
#include <QDateTimeEdit>
#include <QAbstractSpinBox>
#include <QDial>
#include <QWebView>
#include <QUrl>
#include <QCheckBox>
#include <QRadioButton>
#include <QButtonGroup>
#include <QMediaPlayer>

```

```

#include <QMediaPlaylist>
#include <QVideoWidget>
#include <QPrinter>
#include <QAction>
#include <QEvent>
#include <QMessageBox>
#include <QTimer>
#include <QFileDialog>
#include <QPainter>
#include <QPicture>
#include <QPen>
#include <QColor>
#include <QPrinter>
#include <QFont>
#include <QWebSettings>
#include <QBrush>
#include <QByteArray>
#include <QIODevice>
#include <QAbstractSocket>
#include <QTcpSocket>
#include <QTcpServer>
#include <QNetworkProxy>
#include <QHostAddress>
#include <QHostInfo>
#include <QList>
#include <QFileInfo>
#include <QDirModel>
#include <QModelIndex>
#include <QFontDialog>
#include <QDialog>
#include <QTextCursor>
#include <QTextBlock>
#include <QTextDocumentFragment>
#include <QColorDialog>
#include <QHeaderView>
#include <QStringList>
#include <QKeySequence>
#include <QLCDNumber>
#include <QInputDialog>
#include <QDesktopWidget>
#include <QRect>
#include <QTextDocument>

extern "C" {

    #define RING_DLL __declspec(dllexport)

    RING_DLL void ringlib_init(RingState *pRingState)
    {

        new QApplication(pRingState->argc,pRingState->argv);
        ring_qt_start(pRingState) ;

    }

}
</code>

```

```

<runcode>
aStringTypes + "QString"
aBeforeReturn + ["QString", ".toStdString().c_str()"]
aNewMethodName + ["QWebView", "load", "loadpage"]
aNewMethodName + ["QMediaPlaylist", "load", "loadfile"]
aNewMethodName + ["QMediaPlaylist", "next", "movenext"]
aNewMethodName + ["QPainter", "end", "endpaint"]
aNewMethodName + ["QPicture", "load", "loadfile"]
aNewMethodName + ["QLineEdit", "end", "endtext"]
aNewMethodName + ["QDialog", "done", "donedialog"]
aNewMethodName + ["QTextDocument", "end", "enddoc"]
aNewMethodName + ["QTextBlock", "next", "nextblock"]
</runcode>

<class>
name: qApp
nonew
</class>

<register>
void exec(void)
void quit(void)
void processEvents(void)
</register>

<code>

RING_FUNC(ring_qApp_quit)
{
    qApp->quit();
}

RING_FUNC(ring_qApp_exec)
{
    qApp->exec();
}

RING_FUNC(ring_qApp_processEvents)
{
    qApp->processEvents();
}

</code>

<class>
name: QObject
para: void
</class>

bool blockSignals(bool block)
QObjectList children(void)
void dumpObjectInfo(void)
void dumpObjectTree(void)
bool inherits(const char *className)
void installEventFilter(QObject *filterObj)
bool isWidgetType(void)
void killTimer(int id)
void moveToThread(QThread *targetThread)

```

```

QString objectName(void)
QObject *parent(void)
QVariant property(const char *name)
void removeEventFilter(QObject *obj)
void setObjectName(QString)
void setParent(QObject *parent)
bool setProperty(const char *name, QVariant)
bool signalsBlocked(void)
int startTimer(int interval)
QThread *thread(void)
void deleteLater(void)

<class>
name: QWidget
para: void
parent: QObject
</class>

bool acceptDrops(void)
QString accessibleDescription(void)
QString accessibleName(void)
void activateWindow(void)
void addAction(QAction *action)
void adjustSize(void)
bool autoFillBackground(void)
int backgroundRole(void)
QSize baseSize(void)
QWidget *childAt(int x, int y)
QRect childrenRect(void)
QRegion childrenRegion(void)
void clearFocus(void)
void clearMask(void)
QMargins contentsMargins(void)
QRect contentsRect(void)
int contextMenuPolicy(void)
QCursor cursor(void)
int effectiveWinId(void)
void ensurePolished(void)
int focusPolicy(void)
QWidget *focusProxy(void)
QWidget *focusWidget(void)
QFont font(void)
QFontInfo fontInfo(void)
QFontMetrics fontMetrics(void)
int foregroundRole(void)
QRect frameGeometry(void)
QSize frameSize(void)
QRect geometry(void)
void getContentsMargins(int *left, int *top, int *right, int *bottom)
void grabGesture(int gesture, int flags)
void grabKeyboard(void)
void grabMouse(void)
int grabShortcut(QKeySequence , int context)
QGraphicsEffect *graphicsEffect(void)
QGraphicsProxyWidget *graphicsProxyWidget(void)
bool hasFocus(void)
bool hasMouseTracking(void)
int height(void)

```



```

int heightForWidth(int w)
int inputMethodHints(void)
QVariant inputMethodQuery(int query)
void insertAction(QAction *before, QAction *action)
bool isActiveWindow(void)
bool isAncestorOf(QWidget *child)
bool isEnabled(void)
bool isEnabledTo(QWidget *ancestor)
bool isFullScreen(void)
bool isHidden(void)
bool isMaximized(void)
bool isMinimized(void)
bool isModal(void)
bool isVisible(void)
bool isVisibleTo(QWidget *ancestor)
bool isWindow(void)
bool isWindowModified(void)
QLayout *layout(void)
int layoutDirection(void)
QLocale locale(void)
QPoint mapFrom(QWidget *parent, QPoint)
QPoint mapFromGlobal(QPoint)
QPoint mapFromParent(QPoint)
QPoint mapTo(QWidget *parent, QPoint)
QPoint mapToGlobal(QPoint pos)
QPoint mapToParent(QPoint pos)
QRegion mask(void)
int maximumHeight(void)
QSize maximumSize(void)
int maximumWidth(void)
int minimumHeight(void)
QSize minimumSize(void)
int minimumWidth(void)
void move(int x, int y)
QWidget *nativeParentWidget(void)
QWidget *nextInFocusChain(void)
QRect normalGeometry(void)
void overrideWindowFlags(int flags)
QPalette palette(void)
QWidget *parentWidget(void)
QPoint pos(void)
QWidget *previousInFocusChain(void)
QRect rect(void)
void releaseKeyboard(void)
void releaseMouse(void)
void releaseShortcut(int id)
void removeAction(QAction *action)
void render(QPaintDevice *target, QPoint, QRegion, int)
void repaint(int x, int y, int w, int h)
void resize(int w, int h)
bool restoreGeometry(QByteArray)
QByteArray saveGeometry(void)
void scroll(int dx, int dy)
void setAcceptDrops(bool on)
void setAccessibleDescription(QString)
void setAccessibleName(QString)
void setAttribute(int attribute, bool on)
void setAutoFillBackground(bool enabled)

```

```

void setBackgroundRole(int role)
void setBaseSize(int basew, int baseh)
void setContentsMargins(int left, int top, int right, int bottom)
void setContextMenuPolicy(int policy)
void setCursor(QCursor)
void setFixedHeight(int h)
void setFixedSize(int w, int h)
void setFixedWidth(int w)
void setFocus(int reason)
void setFocusPolicy(int policy)
void setFocusProxy(QWidget *w)
void setFont(QFont)
void setForegroundRole(int role)
void setGeometry(int x, int y, int w, int h)
void setGraphicsEffect(QGraphicsEffect *effect)
void setInputMethodHints(int hints)
void setLayout(QLayout *layout)
void setLayoutDirection(int direction)
void setLocale(QLocale)
void setMask(QBitmap)
void setMaximumHeight(int maxh)
void setMaximumSize(int maxw, int maxh)
void setMaximumWidth(int maxw)
void setMinimumHeight(int minh)
void setMinimumSize(int minw, int minh)
void setMinimumWidth(int minw)
void setMouseTracking(bool enable)
void setPalette(QPalette)
void setParent(QWidget *parent)
void setShortcutAutoRepeat(int id, bool enable)
void setShortcutEnabled(int id, bool enable)
void setSizeIncrement(int w, int h)
void setSizePolicy(int horizontal, int vertical)
void setStatusTip(QString)
void setStyle(QStyle *style)
void setToolTip(QString)
void setUpdatesEnabled(bool enable)
void setWhatsThis(QString)
void setWindowFilePath(QString)
void setWindowFlags(int type)
void setWindowIcon(QIcon)
void setWindowIconText(QString)
void setWindowModality(int windowModality)
void setWindowOpacity(double level)
void setWindowRole(QString)
void setWindowState(int windowState)
QSize size(void)
QSize sizeIncrement(void)
QSizePolicy sizePolicy(void)
void stackUnder(QWidget *w)
QString statusTip(void)
QStyle *style(void)
QString styleSheet(void)
bool testAttribute(int attribute)
QString toolTip(void)
bool underMouse(void)
void ungrabGesture(int gesture)
void unsetCursor(void)

```

```

void unsetLayoutDirection(void)
void unsetLocale(void)
void update(int x, int y, int w, int h)
void updateGeometry(void)
bool updatesEnabled(void)
QRegion visibleRegion(void)
QString whatsThis(void)
int width(void)
int winId(void)
QWidget *window(void)
QString windowFilePath(void)
int windowFlags(void)
QIcon windowIcon(void)
QString windowIconText(void)
int windowModality(void)
double windowOpacity(void)
QString windowRole(void)
int windowState(void)
QString windowTitle(void)
int windowType(void)
int x(void)
int y(void)
bool close(void)
void hide(void)
void lower(void)
void raise(void)
void setDisabled(bool disable)
void setEnabled(bool)
void setHidden(bool hidden)
void setStyleSheet(QString)
void setWindowModified(bool)
void setWindowTitle(QString)
void show(void)
void showFullScreen(void)
void showMaximized(void)
void showMinimized(void)
void showNormal(void)
QWidget *find(int id)
QWidget *keyboardGrabber(void)
QWidget *mouseGrabber(void)
void setTabOrder(QWidget *first, QWidget *second)

```

```
<class>
```

```
name: QLabel
```

```
para: QWidget *
```

```
parent: QWidget
```

```
</class>
```

```
int alignment(void)
```

```
QWidget *buddy(void)
```

```
bool hasScaledContents(void)
```

```
bool hasSelectedText(void)
```

```
int indent(void)
```

```
int margin(void)
```

```
QMovie *movie(void)
```

```
bool openExternalLinks(void)
```

```
QPicture *picture(void)
```

```
QPixmap *pixmap(void)
```

```

QString selectedText(void)
int  selectionStart(void)
void setAlignment(int)
void setBuddy(QWidget *buddy)
void setIndent(int)
void setMargin(int)
void setOpenExternalLinks(bool open)
void setScaledContents(bool)
void setSelection(int start, int length)
void setTextFormat(int)
void setTextInteractionFlags(int flags)
void setWordWrap(bool on)
QString text(void)
int  textFormat(void)
int  textInteractionFlags(void)
bool wordWrap(void)
void clear(void)
void setMovie(QMovie *movie)
void setNum(double num)
void setPicture(QPicture)
void setPixmap(QPixmap)
void setText(QString)

<class>
name: QPushButton
para: QWidget *
parent: QWidget
codename: GPushButton
passvmpointer
</class>

void setText(const char *)
void setClickEvent(const char *)
void setIcon(QIcon)
void setIconSize(QSize)

<class>
name: QLineEdit
para: QWidget *
parent: QWidget
codename: GLineEdit
passvmpointer
</class>

int  alignment(void)
void backspace(void)
QCompleter *completer(void)
QMenu *createStandardContextMenu(void)
void cursorBackward(bool mark, int steps)
void cursorForward(bool mark, int steps)
int  cursorMoveStyle(void)
int  cursorPosition(void)
int  cursorPositionAt(QPoint)
void cursorWordBackward(bool mark)
void cursorWordForward(bool mark)
void del(void)
void deselect(void)
QString displayText(void)

```

```

bool dragEnabled(void)
int echoMode(void)
void end(bool mark)
void getTextMargins(int *left, int *top, int *right, int *bottom)
bool hasAcceptableInput(void)
bool hasFrame(void)
bool hasSelectedText(void)
void home(bool mark)
QString inputMask(void)
void insert(QString)
bool isModified(void)
bool isReadOnly(void)
bool isRedoAvailable(void)
bool isUndoAvailable(void)
int maxLength(void)
QString placeholderText(void)
QString selectedText(void)
int selectionStart(void)
void setAlignment(int flag)
void setCompleter(QCompleter *c)
void setCursorMoveStyle(int style)
void setCursorPosition(int)
void setDragEnabled(bool b)
void setEchoMode(int)
void setFrame(bool)
void setInputMask(QString)
void setMaxLength(int)
void setModified(bool)
void setPlaceholderText(QString)
void setReadOnly(bool)
void setSelection(int start, int length)
void setTextMargins(int left, int top, int right, int bottom)
void setValidator(QValidator *v)
QString text(void)
QMargins textMargins(void)
QValidator *validator(void)

void clear(void)
void copy(void)
void cut(void)
void paste(void)
void redo(void)
void selectAll(void)
void setText(QString)
void undo(void)

void setTextChangedEvent(const char *)
void setCursorPositionChangedEvent(const char *)
void seteditingFinishedEvent(const char *)
void setreturnPressedEvent(const char *)
void setselectionChangedEvent(const char *)
void settextEditedEvent(const char *)

```

Note: Most of the content of the previous configuration file is removed from this documentation, for a complete version see the Ring source code distribution.

73.20 Configuration Files Examples

You can learn from the next examples

- RingAllegro : <https://github.com/ring-lang/ring/blob/master/extensions/ringallegro/allegro.cf>
- RingQt : <https://github.com/ring-lang/ring/blob/master/extensions/ringqt/qt.cf>
- RingLibSDL : <https://github.com/ring-lang/ring/blob/master/extensions/ringsdl/libSDL.cf>

After modifying the configuration file, You will need to generate the code, You can learn from the next examples

- RingAllegro : <https://github.com/ring-lang/ring/blob/master/extensions/ringallegro/gencode.bat>
- RingQt : <https://github.com/ring-lang/ring/blob/master/extensions/ringqt/gencode.bat>
- RingLibSDL : <https://github.com/ring-lang/ring/blob/master/extensions/ringsdl/gencode.bat>

After generating the code, You will need to build the library, You can learn from the next examples

- RingAllegro : <https://github.com/ring-lang/ring/blob/master/extensions/ringallegro/buildvc.bat>
- RingQt : <https://github.com/ring-lang/ring/blob/master/extensions/ringqt/buildmingw32.bat>
- RingLibSDL : <https://github.com/ring-lang/ring/blob/master/extensions/ringsdl/buildvc.bat>

RINGLIBCURL FUNCTIONS REFERENCE

- CURLOPT_VERBOSE
- CURLOPT_HEADER
- CURLOPT_NOPROGRESS
- CURLOPT_NOSIGNAL
- CURLOPT_WILDCARDMATCH
- CURLOPT_WRITEFUNCTION
- CURLOPT_WRITEDATA
- CURLOPT_READFUNCTION
- CURLOPT_READDATA
- CURLOPT_IOCTLFUNCTION
- CURLOPT_IOCTLDATA
- CURLOPT_SEEKFUNCTION
- CURLOPT_SEEKDATA
- CURLOPT_SOCKOPTFUNCTION
- CURLOPT_SOCKOPTDATA
- CURLOPT_OPEN_SOCKET_FUNCTION
- CURLOPT_OPEN_SOCKET_DATA
- CURLOPT_CLOSE_SOCKET_FUNCTION
- CURLOPT_CLOSE_SOCKET_DATA
- CURLOPT_PROGRESSFUNCTION
- CURLOPT_PROGRESSDATA
- CURLOPT_HEADERFUNCTION
- CURLOPT_HEADERDATA
- CURLOPT_DEBUGFUNCTION
- CURLOPT_DEBUGDATA
- CURLOPT_SSL_CTX_FUNCTION
- CURLOPT_SSL_CTX_DATA

- CURLOPT_CONV_TO_NETWORK_FUNCTION
- CURLOPT_CONV_FROM_NETWORK_FUNCTION
- CURLOPT_CONV_FROM_UTF8_FUNCTION
- CURLOPT_INTERLEAVEFUNCTION
- CURLOPT_INTERLEAVEDATA
- CURLOPT_CHUNK_BGN_FUNCTION
- CURLOPT_CHUNK_END_FUNCTION
- CURLOPT_CHUNK_DATA
- CURLOPT_FNMATCH_FUNCTION
- CURLOPT_FNMATCH_DATA
- CURLOPT_ERRORBUFFER
- CURLOPT_STDERR
- CURLOPT_FAILONERROR
- CURLOPT_URL
- CURLOPT_PROTOCOLS
- CURLOPT_REDIR_PROTOCOLS
- CURLOPT_PROXY
- CURLOPT_PROXYPORT
- CURLOPT_PROXYTYPE
- CURLOPT_NOPROXY
- CURLOPT_HTTPPROXYTUNNEL
- CURLOPT_SOCKS5_GSSAPI_SERVICE
- CURLOPT_SOCKS5_GSSAPI_NEC
- CURLOPT_INTERFACE
- CURLOPT_LOCALPORT
- CURLOPT_LOCALPORTRANGE
- CURLOPT_DNS_CACHE_TIMEOUT
- CURLOPT_DNS_USE_GLOBAL_CACHE
- CURLOPT_BUFFERSIZE
- CURLOPT_PORT
- CURLOPT_TCP_NODELAY
- CURLOPT_ADDRESS_SCOPE
- CURLOPT_NETRC
- CURLOPT_NETRC_FILE
- CURLOPT_USERPWD
- CURLOPT_PROXYUSERPWD

- CURLOPT_USERNAME
- CURLOPT_PASSWORD
- CURLOPT_PROXYUSERNAME
- CURLOPT_PROXYPASSWORD
- CURLOPT_HTTPAUTH
- CURLOPT_TLSAUTH_USERNAME
- CURLOPT_TLSAUTH_PASSWORD
- CURLOPT_TLSAUTH_TYPE
- CURLOPT_PROXYAUTH
- CURLOPT_AUTOREFERER
- CURLOPT_ACCEPT_ENCODING
- CURLOPT_TRANSFER_ENCODING
- CURLOPT_FOLLOWLOCATION
- CURLOPT_UNRESTRICTED_AUTH
- CURLOPT_MAXREDIRS
- CURLOPT_POSTREDIR
- CURLOPT_PUT
- CURLOPT_POST
- CURLOPT_POSTFIELDS
- CURLOPT_POSTFIELDSIZE
- CURLOPT_POSTFIELDSIZE_LARGE
- CURLOPT_COPYPOSTFIELDS
- CURLOPT_HTTPPOST
- CURLOPT_REFERER
- CURLOPT_USERAGENT
- CURLOPT_HTTPHEADER
- CURLOPT_HTTP200ALIASES
- CURLOPT_COOKIE
- CURLOPT_COOKIEFILE
- CURLOPT_COOKIEJAR
- CURLOPT_COOKIESESSION
- CURLOPT_COOKIELIST
- CURLOPT_HTTPGET
- CURLOPT_HTTP_VERSION
- CURLOPT_IGNORE_CONTENT_LENGTH
- CURLOPT_HTTP_CONTENT_DECODING

- CURLOPT_HTTP_TRANSFER_DECODING
- CURLOPT_MAIL_FROM
- CURLOPT_MAIL_RCPT
- CURLOPT_TFTP_BLKSIZE
- CURLOPT_FTPPORT
- CURLOPT_QUOTE
- CURLOPT_POSTQUOTE
- CURLOPT_PREQUOTE
- CURLOPT_APPEND
- CURLOPT_FTP_USE_EPRT
- CURLOPT_FTP_USE_EPSV
- CURLOPT_FTP_USE_PRET
- CURLOPT_FTP_CREATE_MISSING_DIRS
- CURLOPT_FTP_RESPONSE_TIMEOUT
- CURLOPT_FTP_ALTERNATIVE_TO_USER
- CURLOPT_FTP_SKIP_PASV_IP
- CURLOPT_FTPSSLAUTH
- CURLOPT_FTP_SSL_CCC
- CURLOPT_FTP_ACCOUNT
- CURLOPT_FTP_FILEMETHOD
- CURLOPT_RTSP_REQUEST
- CURLOPT_RTSP_SESSION_ID
- CURLOPT_RTSP_STREAM_URI
- CURLOPT_RTSP_TRANSPORT
- CURLOPT_RTSP_CLIENT_CSEQ
- CURLOPT_RTSP_SERVER_CSEQ
- CURLOPT_TRANSFERTEXT
- CURLOPT_PROXY_TRANSFER_MODE
- CURLOPT_CRLF
- CURLOPT_RANGE
- CURLOPT_RESUME_FROM
- CURLOPT_RESUME_FROM_LARGE
- CURLOPT_CUSTOMREQUEST
- CURLOPT_FILETIME
- CURLOPT_DIRLISTONLY
- CURLOPT_NOBODY

- CURLOPT_INFILESIZE
- CURLOPT_INFILESIZE_LARGE
- CURLOPT_UPLOAD
- CURLOPT_MAXFILESIZE
- CURLOPT_MAXFILESIZE_LARGE
- CURLOPT_TIMECONDITION
- CURLOPT_TIMEVALUE
- CURLOPT_TIMEOUT
- CURLOPT_TIMEOUT_MS
- CURLOPT_LOW_SPEED_LIMIT
- CURLOPT_LOW_SPEED_TIME
- CURLOPT_MAX_SEND_SPEED_LARGE
- CURLOPT_MAX_RECV_SPEED_LARGE
- CURLOPT_MAXCONNECTS
- CURLOPT_FRESH_CONNECT
- CURLOPT_FORBID_REUSE
- CURLOPT_CONNECTTIMEOUT
- CURLOPT_CONNECTTIMEOUT_MS
- CURLOPT_IPRESOLVE
- CURLOPT_CONNECT_ONLY
- CURLOPT_USE_SSL
- CURLOPT_RESOLVE
- CURLOPT_SSLCERT
- CURLOPT_SSLCERTTYPE
- CURLOPT_SSLKEY
- CURLOPT_SSLKEYTYPE
- CURLOPT_KEYPASSWD
- CURLOPT_SSLENGINE
- CURLOPT_SSLENGINE_DEFAULT
- CURLOPT_SSLVERSION
- CURLOPT_SSL_VERIFYHOST
- CURLOPT_SSL_VERIFYPEER
- CURLOPT_CAINFO
- CURLOPT_ISSUERCERT
- CURLOPT_CAPATH
- CURLOPT_CRLFILE

- CURLOPT_CERTINFO
- CURLOPT_RANDOM_FILE
- CURLOPT_EGDSOCKET
- CURLOPT_SSL_CIPHER_LIST
- CURLOPT_SSL_SESSIONID_CACHE
- CURLOPT_KRBLEVEL
- CURLOPT_GSSAPI_DELEGATION
- CURLOPT_SSH_AUTH_TYPES
- CURLOPT_SSH_HOST_PUBLIC_KEY_MD5
- CURLOPT_SSH_PUBLIC_KEYFILE
- CURLOPT_SSH_PRIVATE_KEYFILE
- CURLOPT_SSH_KNOWNHOSTS
- CURLOPT_SSH_KEYFUNCTION
- CURLOPT_SSH_KEYDATA
- CURLOPT_PRIVATE
- CURLOPT_SHARE
- CURLOPT_NEW_FILE_PERMS
- CURLOPT_NEW_DIRECTORY_PERMS
- CURLOPT_TELNETOPTIONS
- CURLE_OK
- CURLE_UNKNOWN_OPTION
- CURLE_NOT_BUILT_IN
- CURLINFO_EFFECTIVE_URL
- CURLINFO_RESPONSE_CODE
- CURLINFO_HTTP_CONNECTCODE
- CURLINFO_FILETIME
- CURLINFO_TOTAL_TIME
- CURLINFO_NAMELOOKUP_TIME
- CURLINFO_CONNECT_TIME
- CURLINFO_APPCONNECT_TIME
- CURLINFO_PRETRANSFER_TIME
- CURLINFO_STARTTRANSFER_TIME
- CURLINFO_REDIRECT_TIME
- CURLINFO_REDIRECT_COUNT
- CURLINFO_REDIRECT_URL
- CURLINFO_SIZE_UPLOAD

- CURLINFO_SIZE_DOWNLOAD
- CURLINFO_SPEED_DOWNLOAD
- CURLINFO_SPEED_UPLOAD
- CURLINFO_HEADER_SIZE
- CURLINFO_REQUEST_SIZE
- CURLINFO_SSL_VERIFYRESULT
- CURLINFO_SSL_ENGINES
- CURLINFO_CONTENT_LENGTH_DOWNLOAD
- CURLINFO_CONTENT_LENGTH_UPLOAD
- CURLINFO_CONTENT_TYPE
- CURLINFO_PRIVATE
- CURLINFO_HTTPAUTH_AVAIL
- CURLINFO_PROXYAUTH_AVAIL
- CURLINFO_OS_ERRNO
- CURLINFO_NUM_CONNECTS
- CURLINFO_PRIMARY_IP
- CURLINFO_PRIMARY_PORT
- CURLINFO_LOCAL_IP
- CURLINFO_LOCAL_PORT
- CURLINFO_COOKIELIST
- CURLINFO_LASTSOCKET
- CURLINFO_FTP_ENTRY_PATH
- CURLINFO_CERTINFO
- CURLINFO_CONDITION_UNMET
- CURLINFO_RTSP_SESSION_ID
- CURLINFO_RTSP_CLIENT_CSEQ
- CURLINFO_RTSP_SERVER_CSEQ
- CURLINFO_RTSP_CSEQ_RECV
- CURLFORM_COPYNAME
- CURLFORM_PTRNAME
- CURLFORM_COPYCONTENTS
- CURLFORM_PTRCONTENTS
- CURLFORM_CONTENTSLENGTH
- CURLFORM_FILECONTENT
- CURLFORM_FILE
- CURLFORM_CONTENTTYPE

- CURLFORM_FILENAME
- CURLFORM_BUFFER
- CURLFORM_BUFFERPTR
- CURLFORM_BUFFERLENGTH
- CURLFORM_STREAM
- CURLFORM_ARRAY
- CURLFORM_CONTENTHEADER
- CURL **curl_easy_init(void)*
- void *curl_easy_cleanup(CURL * handle)*
- CURLcode *curl_easy_setopt_1(CURL *handle, CURLOPToption option, int)*
- CURLcode *curl_easy_setopt_2(CURL *handle, CURLOPToption option, const char *)*
- CURLcode *curl_easy_setopt_3(CURL *handle, CURLOPToption option, void *)*
- CURLcode *curl_easy_setopt_4(CURL *handle, CURLOPToption option, CURLLIST *)*
- CURLcode *curl_easy_perform(CURL * easy_handle)*
- String **curl_easy_perform_silent(CURL * easy_handle)*
- CURLcode *curl_easy_getinfo_1(CURL *handle, CURLINFO info, char **urlp)*
- CURLcode *curl_easy_getinfo_2(CURL *handle, CURLINFO info, long *codep)*
- CURLcode *curl_easy_getinfo_3(CURL *handle, CURLINFO info, double *timep)*
- CURLcode *curl_easy_getinfo_4(CURL *handle, CURLINFO info, CURLLIST **engine_list)*
- CURLcode *curl_easy_getinfo_5(CURL *handle, CURLINFO info, struct curl_certinfo *chainp)*
- CURLcode *curl_easy_getinfo_6(CURL *handle, CURLINFO info, struct curl_tlssessioninfo **session)*
- char **curl_version(void)*
- time_t *curl_getdate(char * datestring , time_t *now)*
- CURLFORMcode *curl_formadd_1(struct curl_httppost **firstitem, struct curl_httppost **lastitem, CURLformoption, const char *, CURLformoption, const char *, CURLformoption)*
- CURLFORMcode *curl_formadd_2(struct curl_httppost **firstitem, struct curl_httppost **lastitem, CURLformoption, const char *, CURLformoption, const char *,CURLformoption, const char *, CURLformoption)*
- CURLFORMcode *curl_formadd_3(struct curl_httppost **firstitem, struct curl_httppost **lastitem, CURLformoption, const char *, CURLformoption, void *, CURLformoption)*
- CURLFORMcode *curl_formadd_4(struct curl_httppost **firstitem, struct curl_httppost **lastitem, CURLformoption, const char *, CURLformoption, void *,CURLformoption, long , CURLformoption)*
- CURLFORMcode *curl_formadd_5(struct curl_httppost **firstitem, struct curl_httppost **lastitem, CURLformoption, const char , CURLformoption, void *,CURLformoption, long , CURLformoption, const char , CURLformoption)*
- CURLFORMcode *curl_formadd_6(struct curl_httppost **firstitem, struct curl_httppost **lastitem, CURLformoption, const char *, CURLformoption, const char *,CURLformoption, void * , CURLformoption, long , CURLformoption)*
- CURLFORMcode *curl_formadd_7(struct curl_httppost **firstitem, struct curl_httppost **lastitem, CURLformoption, const char *, CURLformoption, struct curl_forms [], CURLformoption)*

- void curl_formfree(struct curl_htppost * form)
- CURLLIST *curl_slist_append(CURLLIST * list, const char * string)
- void curl_slist_free_all(CURLLIST * list)
- char *curl_easy_escape(CURL * curl , const char * string , int length)
- char *curl_easy_unescape(CURL * curl , const char * url , int inlength , int * outlength)

RINGLIBZIP FUNCTIONS REFERENCE

- `ZIP_T *zip_openfile(const char *, const char *)`
- `int zip_entry_open(ZIP_T*, const char *)`
- `int zip_entry_write(ZIP_T*, const char *,int)`
- `int zip_entry_fwrite(ZIP_T*, const char *)`
- `int zip_entry_read(ZIP_T*, void *, size_t *)`
- `int zip_entry_fread(ZIP_T*, const char *cFile)`
- `int zip_entry_close(ZIP_T*)`
- `void zip_extract_file(const char *cZIPFile,const char *cFile)`
- `void zip_extract_allfiles(const char *cFile, const char *cFolder)`
- `void zip_close(ZIP_T*)`
- `int zip_filescount(ZIP_T *)`
- `const char *zip_getfilenamebyindex(ZIP_T *pZip,int index)`

RINGCONSOLECOLORS FUNCTIONS REFERENCE

- CC_FG_NONE
- CC_FG_BLACK
- CC_FG_DARK_RED
- CC_FG_DARK_GREEN
- CC_FG_DARK_YELLOW
- CC_FG_DARK_BLUE
- CC_FG_DARK_MAGENTA
- CC_FG_DARK_CYAN
- CC_FG_GRAY
- CC_FG_DARK_GRAY
- CC_FG_RED
- CC_FG_GREEN
- CC_FG_YELLOW
- CC_FG_BLUE
- CC_FG_MAGENTA
- CC_FG_CYAN
- CC_FG_WHITE
- CC_BG_NONE
- CC_BG_BLACK
- CC_BG_DARK_RED
- CC_BG_DARK_GREEN
- CC_BG_DARK_YELLOW
- CC_BG_DARK_BLUE
- CC_BG_DARK_MAGENTA
- CC_BG_DARK_CYAN
- CC_BG_GRAY
- CC_BG_DARK_GRAY

- CC_BG_RED
- CC_BG_GREEN
- CC_BG_YELLOW
- CC_BG_BLUE
- CC_BG_MAGENTA
- CC_BG_CYAN
- CC_BG_WHITE
- void cc_print(int color,const char *string)

RINGMURMURHASH FUNCTIONS REFERENCE

” **MurmurHash is a non-cryptographic hash function suitable for general hash-based lookup.** It was created by Austin Appleby in 2008 and is currently hosted on Github along with its test suite named ‘SMHasher’. It also exists in a number of variants,[5] all of which have been released into the public domain. The name comes from two basic operations, multiply (MU) and rotate (R), used in its inner loop. “

Murmurhash extension is an extension written to implement a full implementation for the MurmurHash library.

Developer: Hassan Ahmed

77.1 MurmurHash1 functions

```
uint32_t murmurhash1(string key, int seed, [bool return_type]);  
uint32_t murmurhash1_aligned(string key, int seed, [bool return_type]);
```

77.2 MurmurHash2 functions

```
uint32_t murmurhash2(string key, int seed, [bool return_type]);  
uint32_t murmurhash2a(string key, int seed, [bool return_type]);  
uint64_t murmurhash64a(string key, int seed, [bool return_type]);  
uint64_t murmurhash64b(string key, int seed, [bool return_type]);  
uint32_t murmurhash_neutral2(string key, int seed, [bool return_type]);  
uint32_t murmurhash_aligned2(string key, int seed, [bool return_type]);
```

77.3 MurmurHash3 functions

```
uint32_t murmurhash3_x86_32(string key, int seed, [bool return_type]);  
list murmurhash3_x86_128(string key, int seed, [bool return_type]);  
list murmurhash3_x64_128(string key, int seed, [bool return_type]);
```

The third optional parameter is to set the type of the returned value, this parameter accepts a bool value [true, false], true will return a Hex value, while false will return a integer value.

77.4 Example

```
load "murmurhashlib.ring"

key = "Ring Language"

see murmurhash3_x86_32(key, 0, 0) + nl // Output: 1894444853
see murmurhash3_x86_32(key, 0, 1) + nl // Output: 70eaef35
```

RINGALLEGRO FUNCTIONS REFERENCE

- void al_exit(void)
- void al_run_main(void)
- int al_init(void)
- ALLEGRO_CONFIG *al_create_config(void)
- void al_destroy_config(ALLEGRO_CONFIG *config)
- ALLEGRO_CONFIG *al_load_config_file(const char *filename)
- ALLEGRO_CONFIG *al_load_config_file_f(ALLEGRO_FILE *file)
- bool al_save_config_file(const char *filename, const ALLEGRO_CONFIG *config)
- bool al_save_config_file_f(ALLEGRO_FILE *file, const ALLEGRO_CONFIG *config)
- void al_add_config_section(ALLEGRO_CONFIG *config, const char *name)
- void al_add_config_comment(ALLEGRO_CONFIG *config, const char *section, const char *comment)
- const char *al_get_config_value(const ALLEGRO_CONFIG *config, const char *section, const char *key)
- void al_set_config_value(ALLEGRO_CONFIG *config, const char *section, const char *key, const char *value)
- char const *al_get_first_config_section(ALLEGRO_CONFIG const *config, ALLEGRO_CONFIG_SECTION **iterator)
- char const *al_get_next_config_section(ALLEGRO_CONFIG_SECTION **iterator)
- char const *al_get_first_config_entry(ALLEGRO_CONFIG const *config, char const *section, ALLEGRO_CONFIG_ENTRY **iterator)
- char const *al_get_next_config_entry(ALLEGRO_CONFIG_ENTRY **iterator)
- ALLEGRO_CONFIG *al_merge_config(const ALLEGRO_CONFIG *cfg1, const ALLEGRO_CONFIG *cfg2)
- void al_merge_config_into(ALLEGRO_CONFIG *master, const ALLEGRO_CONFIG *add)
- ALLEGRO_DISPLAY *al_create_display(int w, int h)
- void al_destroy_display(ALLEGRO_DISPLAY *display)
- int al_get_new_display_flags(void)
- void al_set_new_display_flags(int flags)
- int al_get_new_display_option(int option, int *importance)
- void al_set_new_display_option(int option, int value, int importance)
- void al_reset_new_display_options(void)

- void al_get_new_window_position(int *x, int *y)
- void al_set_new_window_position(int x, int y)
- int al_get_new_display_refresh_rate(void)
- void al_set_new_display_refresh_rate(int refresh_rate)
- ALLEGRO_EVENT_SOURCE *al_get_display_event_source(ALLEGRO_DISPLAY *display)
- ALLEGRO_BITMAP *al_get_backbuffer(ALLEGRO_DISPLAY *display)
- void al_flip_display(void)
- void al_update_display_region(int x, int y, int width, int height)
- bool al_wait_for_vsync(void)
- int al_get_display_width(ALLEGRO_DISPLAY *display)
- int al_get_display_height(ALLEGRO_DISPLAY *display)
- bool al_resize_display(ALLEGRO_DISPLAY *display, int width, int height)
- bool al_acknowledge_resize(ALLEGRO_DISPLAY *display)
- void al_get_window_position(ALLEGRO_DISPLAY *display, int *x, int *y)
- void al_set_window_position(ALLEGRO_DISPLAY *display, int x, int y)
- int al_get_display_flags(ALLEGRO_DISPLAY *display)
- bool al_set_display_flag(ALLEGRO_DISPLAY *display, int flag, bool onoff)
- int al_get_display_option(ALLEGRO_DISPLAY *display, int option)
- int al_get_display_format(ALLEGRO_DISPLAY *display)
- int al_get_display_refresh_rate(ALLEGRO_DISPLAY *display)
- void al_set_window_title(ALLEGRO_DISPLAY *display, const char *title)
- void al_set_display_icon(ALLEGRO_DISPLAY *display, ALLEGRO_BITMAP *icon)
- void al_set_display_icons(ALLEGRO_DISPLAY *display, int num_icons, ALLEGRO_BITMAP *icons[])
- bool al_inhibit_screensaver(bool inhibit)
- void al_acknowledge_drawing_halt(ALLEGRO_DISPLAY *display)
- void al_acknowledge_drawing_resume(ALLEGRO_DISPLAY *display)
- int al_get_display_orientation(ALLEGRO_DISPLAY *display)
- void al_set_display_option(ALLEGRO_DISPLAY *display, int option, int value)
- bool al_get_window_constraints(ALLEGRO_DISPLAY *display, int *min_w, int *min_h, int *max_w, int *max_h)
- bool al_set_window_constraints(ALLEGRO_DISPLAY *display, int min_w, int min_h, int max_w, int max_h)
- ALLEGRO_EVENT_QUEUE *al_create_event_queue(void)
- void al_destroy_event_queue(ALLEGRO_EVENT_QUEUE *queue)
- void al_register_event_source(ALLEGRO_EVENT_QUEUE *queue, ALLEGRO_EVENT_SOURCE *source)
- void al_unregister_event_source(ALLEGRO_EVENT_QUEUE *queue, ALLEGRO_EVENT_SOURCE *source)
- bool al_is_event_queue_empty(ALLEGRO_EVENT_QUEUE *queue)

- `bool al_get_next_event(ALLEGRO_EVENT_QUEUE *queue, ALLEGRO_EVENT *ret_event)`
- `bool al_peek_next_event(ALLEGRO_EVENT_QUEUE *queue, ALLEGRO_EVENT *ret_event)`
- `bool al_drop_next_event(ALLEGRO_EVENT_QUEUE *queue)`
- `void al_flush_event_queue(ALLEGRO_EVENT_QUEUE *queue)`
- `void al_wait_for_event(ALLEGRO_EVENT_QUEUE *queue, ALLEGRO_EVENT *ret_event)`
- `bool al_wait_for_event_timed(ALLEGRO_EVENT_QUEUE *queue, ALLEGRO_EVENT *ret_event, float secs)`
- `bool al_wait_for_event_until(ALLEGRO_EVENT_QUEUE *queue, ALLEGRO_EVENT *ret_event, ALLEGRO_TIMEOUT *timeout)`
- `void al_init_user_event_source(ALLEGRO_EVENT_SOURCE *src)`
- `void al_destroy_user_event_source(ALLEGRO_EVENT_SOURCE *src)`
- `intptr_t al_get_event_source_data(const ALLEGRO_EVENT_SOURCE *source)`
- `void al_set_event_source_data(ALLEGRO_EVENT_SOURCE *source, intptr_t data)`
- `ALLEGRO_FILE *al_fopen(const char *path, const char *mode)`
- `ALLEGRO_FILE *al_fopen_interface(const ALLEGRO_FILE_INTERFACE *drv, const char *path, const char *mode)`
- `ALLEGRO_FILE *al_fopen_slice(ALLEGRO_FILE *fp, size_t initial_size, const char *mode)`
- `void al_fclose(ALLEGRO_FILE *f)`
- `size_t al_fread(ALLEGRO_FILE *f, void *ptr, size_t size)`
- `size_t al_fwrite(ALLEGRO_FILE *f, const void *ptr, size_t size)`
- `bool al_fflush(ALLEGRO_FILE *f)`
- `int64_t al_ftell(ALLEGRO_FILE *f)`
- `bool al_fseek(ALLEGRO_FILE *f, int64_t offset, int whence)`
- `bool al_feof(ALLEGRO_FILE *f)`
- `bool al_ferror(ALLEGRO_FILE *f)`
- `void al_fclearerr(ALLEGRO_FILE *f)`
- `int al_fungetc(ALLEGRO_FILE *f, int c)`
- `int64_t al_fsize(ALLEGRO_FILE *f)`
- `int al_fgetc(ALLEGRO_FILE *f)`
- `int al_fputc(ALLEGRO_FILE *f, int c)`
- `int16_t al_fread16le(ALLEGRO_FILE *f)`
- `int16_t al_fread16be(ALLEGRO_FILE *f)`
- `size_t al_fwrite16le(ALLEGRO_FILE *f, int16_t w)`
- `size_t al_fwrite16be(ALLEGRO_FILE *f, int16_t w)`
- `int32_t al_fread32le(ALLEGRO_FILE *f)`
- `int32_t al_fread32be(ALLEGRO_FILE *f)`
- `size_t al_fwrite32le(ALLEGRO_FILE *f, int32_t l)`

- `size_t al_fwrite32be(ALLEGRO_FILE *f, int32_t l)`
- `char *al_fgets(ALLEGRO_FILE *f, char * const buf, size_t max)`
- `ALLEGRO_USTR *al_fget_ustr(ALLEGRO_FILE *f)`
- `int al_fputs(ALLEGRO_FILE *f, char const *p)`
- `ALLEGRO_FILE *al_fopen_fd(int fd, const char *mode)`
- `ALLEGRO_FILE *al_make_temp_file(const char *template, ALLEGRO_PATH **ret_path)`
- `void al_set_new_file_interface(const ALLEGRO_FILE_INTERFACE *file_interface)`
- `void al_set_standard_file_interface(void)`
- `const ALLEGRO_FILE_INTERFACE *al_get_new_file_interface(void)`
- `ALLEGRO_FILE *al_create_file_handle(const ALLEGRO_FILE_INTERFACE *drv, void *userdata)`
- `void *al_get_file_userdata(ALLEGRO_FILE *f)`
- `ALLEGRO_FS_ENTRY *al_create_fs_entry(const char *path)`
- `void al_destroy_fs_entry(ALLEGRO_FS_ENTRY *fh)`
- `const char *al_get_fs_entry_name(ALLEGRO_FS_ENTRY *e)`
- `bool al_update_fs_entry(ALLEGRO_FS_ENTRY *e)`
- `uint32_t al_get_fs_entry_mode(ALLEGRO_FS_ENTRY *e)`
- `time_t al_get_fs_entry_atime(ALLEGRO_FS_ENTRY *e)`
- `time_t al_get_fs_entry_ctime(ALLEGRO_FS_ENTRY *e)`
- `time_t al_get_fs_entry_mtime(ALLEGRO_FS_ENTRY *e)`
- `off_t al_get_fs_entry_size(ALLEGRO_FS_ENTRY *e)`
- `bool al_fs_entry_exists(ALLEGRO_FS_ENTRY *e)`
- `bool al_remove_fs_entry(ALLEGRO_FS_ENTRY *e)`
- `bool al_filename_exists(const char *path)`
- `bool al_remove_filename(const char *path)`
- `bool al_open_directory(ALLEGRO_FS_ENTRY *e)`
- `ALLEGRO_FS_ENTRY *al_read_directory(ALLEGRO_FS_ENTRY *e)`
- `bool al_close_directory(ALLEGRO_FS_ENTRY *e)`
- `char *al_get_current_directory(void)`
- `bool al_change_directory(const char *path)`
- `bool al_make_directory(const char *path)`
- `ALLEGRO_FILE *al_open_fs_entry(ALLEGRO_FS_ENTRY *e, const char *mode)`
- `void al_set_fs_interface(const ALLEGRO_FS_INTERFACE *fs_interface)`
- `void al_set_standard_fs_interface(void)`
- `const ALLEGRO_FS_INTERFACE *al_get_fs_interface(void)`
- `al_fixed al_itofix(int x);`
- `int al_fixtoi(al_fixed x);`

- `int al_fixfloor(al_fixed x);`
- `int al_fixceil(al_fixed x);`
- `al_fixed al_ftofix(double x);`
- `double al_fixtof(al_fixed x);`
- `al_fixed al_fixmul(al_fixed x, al_fixed y);`
- `al_fixed al_fixdiv(al_fixed x, al_fixed y);`
- `al_fixed al_fixadd(al_fixed x, al_fixed y);`
- `al_fixed al_fixsub(al_fixed x, al_fixed y);`
- `al_fixed al_fixsin(al_fixed x);`
- `al_fixed al_fixcos(al_fixed x);`
- `al_fixed al_fixtan(al_fixed x);`
- `al_fixed al_fixasin(al_fixed x);`
- `al_fixed al_fixacos(al_fixed x);`
- `al_fixed al_fixatan(al_fixed x)`
- `al_fixed al_fixatan2(al_fixed y, al_fixed x)`
- `al_fixed al_fixsqrt(al_fixed x)`
- `al_fixed al_fixhypot(al_fixed x, al_fixed y)`
- `ALLEGRO_DISPLAY_MODE *al_get_display_mode(int index, ALLEGRO_DISPLAY_MODE *mode)`
- `int al_get_num_display_modes(void)`
- `ALLEGRO_COLOR al_map_rgb(unsigned char r, unsigned char g, unsigned char b)`
- `ALLEGRO_COLOR al_map_rgb_f(float r, float g, float b)`
- `ALLEGRO_COLOR al_map_rgba(unsigned char r, unsigned char g, unsigned char b, unsigned char a)`
- `ALLEGRO_COLOR al_map_rgba_f(float r, float g, float b, float a)`
- `void al_unmap_rgb(ALLEGRO_COLOR color, unsigned char *r, unsigned char *g, unsigned char *b)`
- `void al_unmap_rgb_f(ALLEGRO_COLOR color, float *r, float *g, float *b)`
- `void al_unmap_rgba(ALLEGRO_COLOR color, unsigned char *r, unsigned char *g, unsigned char *b, unsigned char *a)`
- `void al_unmap_rgba_f(ALLEGRO_COLOR color, float *r, float *g, float *b, float *a)`
- `int al_get_pixel_size(int format)`
- `int al_get_pixel_format_bits(int format)`
- `ALLEGRO_LOCKED_REGION *al_lock_bitmap(ALLEGRO_BITMAP *bitmap, int format, int flags)`
- `ALLEGRO_LOCKED_REGION *al_lock_bitmap_region(ALLEGRO_BITMAP *bitmap, int x, int y, int width, int height, int format, int flags)`
- `void al_unlock_bitmap(ALLEGRO_BITMAP *bitmap)`
- `ALLEGRO_BITMAP *al_create_bitmap(int w, int h)`
- `ALLEGRO_BITMAP *al_create_sub_bitmap(ALLEGRO_BITMAP *parent, int x, int y, int w, int h)`
- `ALLEGRO_BITMAP *al_clone_bitmap(ALLEGRO_BITMAP *bitmap)`

- `void al_destroy_bitmap(ALLEGRO_BITMAP *bitmap)`
- `int al_get_new_bitmap_flags(void)`
- `int al_get_new_bitmap_format(void)`
- `void al_set_new_bitmap_flags(int flags)`
- `void al_add_new_bitmap_flag(int flag)`
- `void al_set_new_bitmap_format(int format)`
- `int al_get_bitmap_flags(ALLEGRO_BITMAP *bitmap)`
- `int al_get_bitmap_format(ALLEGRO_BITMAP *bitmap)`
- `int al_get_bitmap_height(ALLEGRO_BITMAP *bitmap)`
- `int al_get_bitmap_width(ALLEGRO_BITMAP *bitmap)`
- `ALLEGRO_COLOR al_get_pixel(ALLEGRO_BITMAP *bitmap, int x, int y)`
- `bool al_is_bitmap_locked(ALLEGRO_BITMAP *bitmap)`
- `bool al_is_compatible_bitmap(ALLEGRO_BITMAP *bitmap)`
- `bool al_is_sub_bitmap(ALLEGRO_BITMAP *bitmap)`
- `ALLEGRO_BITMAP *al_get_parent_bitmap(ALLEGRO_BITMAP *bitmap)`
- `void al_clear_to_color(ALLEGRO_COLOR color)`
- `void al_draw_bitmap(ALLEGRO_BITMAP *bitmap, float dx, float dy, int flags)`
- `void al_draw_tinted_bitmap(ALLEGRO_BITMAP *bitmap, ALLEGRO_COLOR tint, float dx, float dy, int flags)`
- `void al_draw_bitmap_region(ALLEGRO_BITMAP *bitmap, float sx, float sy, float sw, float sh, float dx, float dy, int flags)`
- `void al_draw_tinted_bitmap_region(ALLEGRO_BITMAP *bitmap, ALLEGRO_COLOR tint, float sx, float sy, float sw, float sh, float dx, float dy, int flags)`
- `void al_draw_pixel(float x, float y, ALLEGRO_COLOR color)`
- `void al_draw_rotated_bitmap(ALLEGRO_BITMAP *bitmap, float cx, float cy, float dx, float dy, float angle, int flags)`
- `void al_draw_tinted_rotated_bitmap(ALLEGRO_BITMAP *bitmap, ALLEGRO_COLOR tint, float cx, float cy, float dx, float dy, float angle, int flags)`
- `void al_draw_scaled_rotated_bitmap(ALLEGRO_BITMAP *bitmap, float cx, float cy, float dx, float dy, float xscale, float yscale, float angle, int flags)`
- `void al_draw_tinted_scaled_rotated_bitmap(ALLEGRO_BITMAP *bitmap, ALLEGRO_COLOR tint, float cx, float cy, float dx, float dy, float xscale, float yscale, float angle, int flags)`
- `void al_draw_tinted_scaled_rotated_bitmap_region(ALLEGRO_BITMAP *bitmap, float sx, float sy, float sw, float sh, ALLEGRO_COLOR tint, float cx, float cy, float dx, float dy, float xscale, float yscale, float angle, int flags)`
- `void al_draw_scaled_bitmap(ALLEGRO_BITMAP *bitmap, float sx, float sy, float sw, float sh, float dx, float dy, float dw, float dh, int flags)`
- `void al_draw_tinted_scaled_bitmap(ALLEGRO_BITMAP *bitmap, ALLEGRO_COLOR tint, float sx, float sy, float sw, float sh, float dx, float dy, float dw, float dh, int flags)`
- `ALLEGRO_BITMAP *al_get_target_bitmap(void)`

- void al_put_pixel(int x, int y, ALLEGRO_COLOR color)
- void al_put_blended_pixel(int x, int y, ALLEGRO_COLOR color)
- void al_set_target_bitmap(ALLEGRO_BITMAP *bitmap)
- void al_set_target_backbuffer(ALLEGRO_DISPLAY *display)
- ALLEGRO_DISPLAY *al_get_current_display(void)
- void al_get_blender(int *op, int *src, int *dst)
- void al_get_separate_blender(int *op, int *src, int *dst, int *alpha_op, int *alpha_src, int *alpha_dst)
- void al_set_blender(int op, int src, int dst)
- void al_set_separate_blender(int op, int src, int dst, int alpha_op, int alpha_src, int alpha_dst)
- void al_get_clipping_rectangle(int *x, int *y, int *w, int *h)
- void al_set_clipping_rectangle(int x, int y, int width, int height)
- void al_reset_clipping_rectangle(void)
- void al_convert_mask_to_alpha(ALLEGRO_BITMAP *bitmap, ALLEGRO_COLOR mask_color)
- void al_hold_bitmap_drawing(bool hold)
- bool al_is_bitmap_drawing_held(void)
- ALLEGRO_BITMAP *al_load_bitmap_f(ALLEGRO_FILE *fp, const char *ident)
- bool al_save_bitmap(const char *filename, ALLEGRO_BITMAP *bitmap)
- bool al_save_bitmap_f(ALLEGRO_FILE *fp, const char *ident, ALLEGRO_BITMAP *bitmap)
- bool al_install_joystick(void)
- void al_uninstall_joystick(void)
- bool al_is_joystick_installed(void)
- bool al_reconfigure_joysticks(void)
- int al_get_num_joysticks(void)
- ALLEGRO_JOYSTICK *al_get_joystick(int num)
- void al_release_joystick(ALLEGRO_JOYSTICK *joy)
- bool al_get_joystick_active(ALLEGRO_JOYSTICK *joy)
- const char *al_get_joystick_name(ALLEGRO_JOYSTICK *joy)
- const char *al_get_joystick_stick_name(ALLEGRO_JOYSTICK *joy, int stick)
- const char *al_get_joystick_axis_name(ALLEGRO_JOYSTICK *joy, int stick, int axis)
- const char *al_get_joystick_button_name(ALLEGRO_JOYSTICK *joy, int button)
- int al_get_joystick_stick_flags(ALLEGRO_JOYSTICK *joy, int stick)
- int al_get_joystick_num_sticks(ALLEGRO_JOYSTICK *joy)
- int al_get_joystick_num_axes(ALLEGRO_JOYSTICK *joy, int stick)
- int al_get_joystick_num_buttons(ALLEGRO_JOYSTICK *joy)
- void al_get_joystick_state(ALLEGRO_JOYSTICK *joy, ALLEGRO_JOYSTICK_STATE *ret_state)
- ALLEGRO_EVENT_SOURCE *al_get_joystick_event_source(void)

- `bool al_install_keyboard(void)`
- `bool al_is_keyboard_installed(void)`
- `void al_uninstall_keyboard(void)`
- `void al_get_keyboard_state(ALLEGRO_KEYBOARD_STATE *ret_state)`
- `bool al_key_down(const ALLEGRO_KEYBOARD_STATE *state, int keycode)`
- `const char *al_keycode_to_name(int keycode)`
- `bool al_set_keyboard_leds(int leds)`
- `ALLEGRO_EVENT_SOURCE *al_get_keyboard_event_source(void)`
- `void *al_malloc_with_context(size_t n, int line, const char *file, const char *func)`
- `void al_free_with_context(void *ptr, int line, const char *file, const char *func)`
- `void *al_realloc_with_context(void *ptr, size_t n, int line, const char *file, const char *func)`
- `void *al_calloc_with_context(size_t count, size_t n, int line, const char *file, const char *func)`
- `void al_set_memory_interface(ALLEGRO_MEMORY_INTERFACE *memory_interface)`
- `int al_get_new_display_adapter(void)`
- `void al_set_new_display_adapter(int adapter)`
- `bool al_get_monitor_info(int adapter, ALLEGRO_MONITOR_INFO *info)`
- `int al_get_num_video_adapters(void)`
- `bool al_install_mouse(void)`
- `bool al_is_mouse_installed(void)`
- `void al_uninstall_mouse(void)`
- `unsigned int al_get_mouse_num_axes(void)`
- `unsigned int al_get_mouse_num_buttons(void)`
- `void al_get_mouse_state(ALLEGRO_MOUSE_STATE *ret_state)`
- `int al_get_mouse_state_axis(const ALLEGRO_MOUSE_STATE *state, int axis)`
- `bool al_mouse_button_down(const ALLEGRO_MOUSE_STATE *state, int button)`
- `bool al_set_mouse_xy(ALLEGRO_DISPLAY *display, int x, int y)`
- `bool al_set_mouse_z(int z)`
- `bool al_set_mouse_w(int w)`
- `bool al_set_mouse_axis(int which, int value)`
- `ALLEGRO_EVENT_SOURCE *al_get_mouse_event_source(void)`
- `ALLEGRO_MOUSE_CURSOR *al_create_mouse_cursor(ALLEGRO_BITMAP *bmp, int x_focus, int y_focus)`
- `void al_destroy_mouse_cursor(ALLEGRO_MOUSE_CURSOR *cursor)`
- `bool al_set_mouse_cursor(ALLEGRO_DISPLAY *display, ALLEGRO_MOUSE_CURSOR *cursor)`
- `bool al_set_system_mouse_cursor(ALLEGRO_DISPLAY *display, ALLEGRO_SYSTEM_MOUSE_CURSOR cursor_id)`
- `bool al_get_mouse_cursor_position(int *ret_x, int *ret_y)`

- `bool al_hide_mouse_cursor(ALLEGRO_DISPLAY *display)`
- `bool al_show_mouse_cursor(ALLEGRO_DISPLAY *display)`
- `bool al_grab_mouse(ALLEGRO_DISPLAY *display)`
- `bool al_ungrab_mouse(void)`
- `ALLEGRO_PATH *al_create_path(const char *str)`
- `ALLEGRO_PATH *al_create_path_for_directory(const char *str)`
- `void al_destroy_path(ALLEGRO_PATH *path)`
- `ALLEGRO_PATH *al_clone_path(const ALLEGRO_PATH *path)`
- `bool al_join_paths(ALLEGRO_PATH *path, const ALLEGRO_PATH *tail)`
- `bool al_rebase_path(const ALLEGRO_PATH *head, ALLEGRO_PATH *tail)`
- `const char *al_get_path_drive(const ALLEGRO_PATH *path)`
- `int al_get_path_num_components(const ALLEGRO_PATH *path)`
- `const char *al_get_path_component(const ALLEGRO_PATH *path, int i)`
- `const char *al_get_path_tail(const ALLEGRO_PATH *path)`
- `const char *al_get_path_filename(const ALLEGRO_PATH *path)`
- `const char *al_get_path_basename(const ALLEGRO_PATH *path)`
- `const char *al_get_path_extension(const ALLEGRO_PATH *path)`
- `void al_set_path_drive(ALLEGRO_PATH *path, const char *drive)`
- `void al_append_path_component(ALLEGRO_PATH *path, const char *s)`
- `void al_insert_path_component(ALLEGRO_PATH *path, int i, const char *s)`
- `void al_replace_path_component(ALLEGRO_PATH *path, int i, const char *s)`
- `void al_remove_path_component(ALLEGRO_PATH *path, int i)`
- `void al_drop_path_tail(ALLEGRO_PATH *path)`
- `void al_set_path_filename(ALLEGRO_PATH *path, const char *filename)`
- `bool al_set_path_extension(ALLEGRO_PATH *path, const char *extension)`
- `const char *al_path_cstr(const ALLEGRO_PATH *path, char delim)`
- `bool al_make_path_canonical(ALLEGRO_PATH *path)`
- `void al_restore_state(ALLEGRO_STATE const *state)`
- `void al_store_state(ALLEGRO_STATE *state, int flags)`
- `int al_get_errno(void)`
- `void al_set_errno(int errnum)`
- `void al_uninstall_system(void)`
- `bool al_is_system_installed(void)`
- `uint32_t al_get_allegro_version(void)`
- `ALLEGRO_PATH *al_get_standard_path(int id)`
- `void al_set_exe_name(const char *path)`

- void al_set_app_name(const char *app_name)
- void al_set_org_name(const char *org_name)
- const char *al_get_app_name(void)
- const char *al_get_org_name(void)
- ALLEGRO_CONFIG *al_get_system_config(void)
- ALLEGRO_THREAD *al_create_thread(void)
- void al_run_detached_thread(void)
- void al_start_thread(ALLEGRO_THREAD *thread)
- void al_join_thread(ALLEGRO_THREAD *thread, void **ret_value)
- void al_set_thread_should_stop(ALLEGRO_THREAD *thread)
- bool al_get_thread_should_stop(ALLEGRO_THREAD *thread)
- void al_destroy_thread(ALLEGRO_THREAD *thread)
- ALLEGRO_MUTEX *al_create_mutex(void)
- ALLEGRO_MUTEX *al_create_mutex_recursive(void)
- void al_lock_mutex(ALLEGRO_MUTEX *mutex)
- void al_unlock_mutex(ALLEGRO_MUTEX *mutex)
- void al_destroy_mutex(ALLEGRO_MUTEX *mutex)
- ALLEGRO_COND *al_create_cond(void)
- void al_destroy_cond(ALLEGRO_COND *cond)
- void al_wait_cond(ALLEGRO_COND *cond, ALLEGRO_MUTEX *mutex)
- int al_wait_cond_until(ALLEGRO_COND *cond, ALLEGRO_MUTEX *mutex, const ALLEGRO_TIMEOUT *timeout)
- void al_broadcast_cond(ALLEGRO_COND *cond)
- void al_signal_cond(ALLEGRO_COND *cond)
- double al_get_time(void)
- void al_init_timeout(ALLEGRO_TIMEOUT *timeout, double seconds)
- void al_rest(double seconds)
- ALLEGRO_TIMER *al_create_timer(double speed_secs)
- void al_start_timer(ALLEGRO_TIMER *timer)
- void al_stop_timer(ALLEGRO_TIMER *timer)
- bool al_get_timer_started(const ALLEGRO_TIMER *timer)
- void al_destroy_timer(ALLEGRO_TIMER *timer)
- int64_t al_get_timer_count(const ALLEGRO_TIMER *timer)
- void al_set_timer_count(ALLEGRO_TIMER *timer, int64_t new_count)
- void al_add_timer_count(ALLEGRO_TIMER *timer, int64_t diff)
- double al_get_timer_speed(const ALLEGRO_TIMER *timer)

- void al_set_timer_speed(ALLEGRO_TIMER *timer, double new_speed_secs)
- ALLEGRO_EVENT_SOURCE *al_get_timer_event_source(ALLEGRO_TIMER *timer)
- void al_copy_transform(ALLEGRO_TRANSFORM *dest, const ALLEGRO_TRANSFORM *src)
- void al_use_transform(const ALLEGRO_TRANSFORM *trans)
- const ALLEGRO_TRANSFORM *al_get_current_transform(void)
- void al_invert_transform(ALLEGRO_TRANSFORM *trans)
- int al_check_inverse(const ALLEGRO_TRANSFORM *trans, float tol)
- void al_identity_transform(ALLEGRO_TRANSFORM *trans)
- void al_build_transform(ALLEGRO_TRANSFORM *trans, float x, float y, float sx, float sy, float theta)
- void al_translate_transform(ALLEGRO_TRANSFORM *trans, float x, float y)
- void al_rotate_transform(ALLEGRO_TRANSFORM *trans, float theta)
- void al_scale_transform(ALLEGRO_TRANSFORM *trans, float sx, float sy)
- void al_transform_coordinates(const ALLEGRO_TRANSFORM *trans, float *x, float *y)
- void al_compose_transform(ALLEGRO_TRANSFORM *trans, const ALLEGRO_TRANSFORM *other)
- ALLEGRO_USTR *al_ustr_new(const char *s)
- ALLEGRO_USTR *al_ustr_new_from_buffer(const char *s, size_t size)
- void al_ustr_free(ALLEGRO_USTR *us)
- const char *al_cstr(const ALLEGRO_USTR *us)
- void al_ustr_to_buffer(const ALLEGRO_USTR *us, char *buffer, int size)
- char *al_cstr_dup(const ALLEGRO_USTR *us)
- ALLEGRO_USTR *al_ustr_dup(const ALLEGRO_USTR *us)
- ALLEGRO_USTR *al_ustr_dup_substr(const ALLEGRO_USTR *us, int start_pos, int end_pos)
- const ALLEGRO_USTR *al_ustr_empty_string(void)
- const ALLEGRO_USTR *al_ref_cstr(ALLEGRO_USTR_INFO *info, const char *s)
- const ALLEGRO_USTR *al_ref_buffer(ALLEGRO_USTR_INFO *info, const char *s, size_t size)
- const ALLEGRO_USTR *al_ref_ustr(ALLEGRO_USTR_INFO *info, const ALLEGRO_USTR *us, int start_pos, int end_pos)
- size_t al_ustr_size(const ALLEGRO_USTR *us)
- size_t al_ustr_length(const ALLEGRO_USTR *us)
- int al_ustr_offset(const ALLEGRO_USTR *us, int index)
- bool al_ustr_next(const ALLEGRO_USTR *us, int *pos)
- bool al_ustr_prev(const ALLEGRO_USTR *us, int *pos)
- int32_t al_ustr_get(const ALLEGRO_USTR *ub, int pos)
- int32_t al_ustr_get_next(const ALLEGRO_USTR *us, int *pos)
- int32_t al_ustr_prev_get(const ALLEGRO_USTR *us, int *pos)
- bool al_ustr_insert(ALLEGRO_USTR *us1, int pos, const ALLEGRO_USTR *us2)

- `bool al_ustr_insert_cstr(ALLEGRO_USTR *us, int pos, const char *s)`
- `size_t al_ustr_insert_chr(ALLEGRO_USTR *us, int pos, int32_t c)`
- `bool al_ustr_append(ALLEGRO_USTR *us1, const ALLEGRO_USTR *us2)`
- `bool al_ustr_append_cstr(ALLEGRO_USTR *us, const char *s)`
- `size_t al_ustr_append_chr(ALLEGRO_USTR *us, int32_t c)`
- `bool al_ustr_remove_chr(ALLEGRO_USTR *us, int pos)`
- `bool al_ustr_remove_range(ALLEGRO_USTR *us, int start_pos, int end_pos)`
- `bool al_ustr_truncate(ALLEGRO_USTR *us, int start_pos)`
- `bool al_ustr_ltrim_ws(ALLEGRO_USTR *us)`
- `bool al_ustr_rtrim_ws(ALLEGRO_USTR *us)`
- `bool al_ustr_trim_ws(ALLEGRO_USTR *us)`
- `bool al_ustr_assign(ALLEGRO_USTR *us1, const ALLEGRO_USTR *us2)`
- `bool al_ustr_assign_substr(ALLEGRO_USTR *us1, const ALLEGRO_USTR *us2, int start_pos, int end_pos)`
- `bool al_ustr_assign_cstr(ALLEGRO_USTR *us1, const char *s)`
- `size_t al_ustr_set_chr(ALLEGRO_USTR *us, int start_pos, int32_t c)`
- `bool al_ustr_replace_range(ALLEGRO_USTR *us1, int start_pos1, int end_pos1, const ALLEGRO_USTR *us2)`
- `int al_ustr_find_chr(const ALLEGRO_USTR *us, int start_pos, int32_t c)`
- `int al_ustr_rfind_chr(const ALLEGRO_USTR *us, int end_pos, int32_t c)`
- `int al_ustr_find_set(const ALLEGRO_USTR *us, int start_pos, const ALLEGRO_USTR *accept)`
- `int al_ustr_find_set_cstr(const ALLEGRO_USTR *us, int start_pos, const char *accept)`
- `int al_ustr_find_cset(const ALLEGRO_USTR *us, int start_pos, const ALLEGRO_USTR *reject)`
- `int al_ustr_find_cset_cstr(const ALLEGRO_USTR *us, int start_pos, const char *reject)`
- `int al_ustr_find_str(const ALLEGRO_USTR *haystack, int start_pos, const ALLEGRO_USTR *needle)`
- `int al_ustr_find_cstr(const ALLEGRO_USTR *haystack, int start_pos, const char *needle)`
- `int al_ustr_rfind_str(const ALLEGRO_USTR *haystack, int end_pos, const ALLEGRO_USTR *needle)`
- `int al_ustr_rfind_cstr(const ALLEGRO_USTR *haystack, int end_pos, const char *needle)`
- `bool al_ustr_find_replace(ALLEGRO_USTR *us, int start_pos, const ALLEGRO_USTR *find, const ALLEGRO_USTR *replace)`
- `bool al_ustr_find_replace_cstr(ALLEGRO_USTR *us, int start_pos, const char *find, const char *replace)`
- `int al_ustr_compare(const ALLEGRO_USTR *us1, const ALLEGRO_USTR *us2)`
- `int al_ustr_ncompare(const ALLEGRO_USTR *us1, const ALLEGRO_USTR *us2, int n)`
- `bool al_ustr_equal(const ALLEGRO_USTR *us1, const ALLEGRO_USTR *us2)`
- `bool al_ustr_has_prefix(const ALLEGRO_USTR *us1, const ALLEGRO_USTR *us2)`
- `bool al_ustr_has_prefix_cstr(const ALLEGRO_USTR *us1, const char *s2)`
- `bool al_ustr_has_suffix(const ALLEGRO_USTR *us1, const ALLEGRO_USTR *us2)`
- `bool al_ustr_has_suffix_cstr(const ALLEGRO_USTR *us1, const char *s2)`

- ALLEGRO_USTR *al_ustr_new_from_utf16(uint16_t const *s)
- size_t al_ustr_size_utf16(const ALLEGRO_USTR *us)
- size_t al_ustr_encode_utf16(const ALLEGRO_USTR *us, uint16_t *s, size_t n)
- size_t al_utf8_width(int c)
- size_t al_utf8_encode(char s[], int32_t c)
- size_t al_utf16_width(int c)
- LPDIRECT3DDEVICE9 al_get_d3d_device(ALLEGRO_DISPLAY *display)
- LPDIRECT3DTEXTURE9 al_get_d3d_system_texture(ALLEGRO_BITMAP *bitmap)
- LPDIRECT3DTEXTURE9 al_get_d3d_video_texture(ALLEGRO_BITMAP *bitmap)
- bool al_have_d3d_non_pow2_texture_support(void)
- bool al_have_d3d_non_square_texture_support(void)
- void al_get_d3d_texture_position(ALLEGRO_BITMAP *bitmap, int *u, int *v)
- bool al_is_d3d_device_lost(ALLEGRO_DISPLAY *display)
- ALLEGRO_OGL_EXT_LIST *al_get_opengl_extension_list(void)
- void *al_get_opengl_proc_address(const char *name)
- GLuint al_get_opengl_texture(ALLEGRO_BITMAP *bitmap)
- void al_get_opengl_texture_size(ALLEGRO_BITMAP *bitmap, int *w, int *h)
- void al_get_opengl_texture_position(ALLEGRO_BITMAP *bitmap, int *u, int *v)
- GLuint al_get_opengl_fbo(ALLEGRO_BITMAP *bitmap)
- void al_remove_opengl_fbo(ALLEGRO_BITMAP *bitmap)
- bool al_have_opengl_extension(const char *extension);
- uint32_t al_get_opengl_version(void)
- int al_get_opengl_variant(void)
- void al_set_current_opengl_context(ALLEGRO_DISPLAY *display)
- bool al_install_audio(void)
- void al_uninstall_audio(void)
- bool al_is_audio_installed(void)
- bool al_reserve_samples(int reserve_samples)
- uint32_t al_get_allegro_audio_version(void)
- size_t al_get_audio_depth_size(ALLEGRO_AUDIO_DEPTH depth)
- size_t al_get_channel_count(ALLEGRO_CHANNEL_CONF conf)
- ALLEGRO_VOICE *al_create_voice(unsigned int freq, ALLEGRO_AUDIO_DEPTH depth, ALLEGRO_CHANNEL_CONF chan_conf)
- void al_destroy_voice(ALLEGRO_VOICE *voice)
- void al_detach_voice(ALLEGRO_VOICE *voice)
- bool al_attach_audio_stream_to_voice(ALLEGRO_AUDIO_STREAM *stream, ALLEGRO_VOICE *voice)

- `bool al_attach_mixer_to_voice(ALLEGRO_MIXER *mixer, ALLEGRO_VOICE *voice)`
- `bool al_attach_sample_instance_to_voice(ALLEGRO_SAMPLE_INSTANCE *spl, ALLEGRO_VOICE *voice)`
- `unsigned int al_get_voice_frequency(const ALLEGRO_VOICE *voice)`
- `ALLEGRO_CHANNEL_CONF al_get_voice_channels(const ALLEGRO_VOICE *voice)`
- `ALLEGRO_AUDIO_DEPTH al_get_voice_depth(const ALLEGRO_VOICE *voice)`
- `bool al_get_voice_playing(const ALLEGRO_VOICE *voice)`
- `bool al_set_voice_playing(ALLEGRO_VOICE *voice, bool val)`
- `unsigned int al_get_voice_position(const ALLEGRO_VOICE *voice)`
- `bool al_set_voice_position(ALLEGRO_VOICE *voice, unsigned int val)`
- `ALLEGRO_SAMPLE *al_create_sample(void *buf, unsigned int samples, unsigned int freq, ALLEGRO_AUDIO_DEPTH depth, ALLEGRO_CHANNEL_CONF chan_conf, bool free_buf)`
- `void al_destroy_sample(ALLEGRO_SAMPLE *spl)`
- `bool al_play_sample(ALLEGRO_SAMPLE *spl, float gain, float pan, float speed, int loop, ALLEGRO_SAMPLE_ID *ret_id)`
- `void al_stop_sample(ALLEGRO_SAMPLE_ID *spl_id)`
- `void al_stop_samples(void)`
- `ALLEGRO_CHANNEL_CONF al_get_sample_channels(const ALLEGRO_SAMPLE *spl)`
- `ALLEGRO_AUDIO_DEPTH al_get_sample_depth(const ALLEGRO_SAMPLE *spl)`
- `unsigned int al_get_sample_frequency(const ALLEGRO_SAMPLE *spl)`
- `unsigned int al_get_sample_length(const ALLEGRO_SAMPLE *spl)`
- `void *al_get_sample_data(const ALLEGRO_SAMPLE *spl)`
- `ALLEGRO_SAMPLE_INSTANCE *al_create_sample_instance(ALLEGRO_SAMPLE *sample_data)`
- `void al_destroy_sample_instance(ALLEGRO_SAMPLE_INSTANCE *spl)`
- `bool al_play_sample_instance(ALLEGRO_SAMPLE_INSTANCE *spl)`
- `bool al_stop_sample_instance(ALLEGRO_SAMPLE_INSTANCE *spl)`
- `ALLEGRO_CHANNEL_CONF al_get_sample_instance_channels(const ALLEGRO_SAMPLE_INSTANCE *spl)`
- `ALLEGRO_AUDIO_DEPTH al_get_sample_instance_depth(const ALLEGRO_SAMPLE_INSTANCE *spl)`
- `unsigned int al_get_sample_instance_frequency(const ALLEGRO_SAMPLE_INSTANCE *spl)`
- `unsigned int al_get_sample_instance_length(const ALLEGRO_SAMPLE_INSTANCE *spl)`
- `bool al_set_sample_instance_length(ALLEGRO_SAMPLE_INSTANCE *spl, unsigned int val)`
- `unsigned int al_get_sample_instance_position(const ALLEGRO_SAMPLE_INSTANCE *spl)`
- `bool al_set_sample_instance_position(ALLEGRO_SAMPLE_INSTANCE *spl, unsigned int val)`
- `float al_get_sample_instance_speed(const ALLEGRO_SAMPLE_INSTANCE *spl)`
- `bool al_set_sample_instance_speed(ALLEGRO_SAMPLE_INSTANCE *spl, float val)`
- `float al_get_sample_instance_gain(const ALLEGRO_SAMPLE_INSTANCE *spl)`

- `bool al_set_sample_instance_gain(ALLEGRO_SAMPLE_INSTANCE *spl, float val)`
- `float al_get_sample_instance_pan(const ALLEGRO_SAMPLE_INSTANCE *spl)`
- `bool al_set_sample_instance_pan(ALLEGRO_SAMPLE_INSTANCE *spl, float val)`
- `float al_get_sample_instance_time(const ALLEGRO_SAMPLE_INSTANCE *spl)`
- `ALLEGRO_PLAYMODE al_get_sample_instance_playmode(const ALLEGRO_SAMPLE_INSTANCE *spl)`
- `bool al_set_sample_instance_playmode(ALLEGRO_SAMPLE_INSTANCE *spl, ALLEGRO_PLAYMODE val)`
- `bool al_get_sample_instance_playing(const ALLEGRO_SAMPLE_INSTANCE *spl)`
- `bool al_set_sample_instance_playing(ALLEGRO_SAMPLE_INSTANCE *spl, bool val)`
- `bool al_get_sample_instance_attached(const ALLEGRO_SAMPLE_INSTANCE *spl)`
- `bool al_detach_sample_instance(ALLEGRO_SAMPLE_INSTANCE *spl)`
- `ALLEGRO_SAMPLE *al_get_sample(ALLEGRO_SAMPLE_INSTANCE *spl)`
- `bool al_set_sample(ALLEGRO_SAMPLE_INSTANCE *spl, ALLEGRO_SAMPLE *data)`
- `ALLEGRO_MIXER *al_create_mixer(unsigned int freq, ALLEGRO_AUDIO_DEPTH depth, ALLEGRO_CHANNEL_CONF chan_conf)`
- `void al_destroy_mixer(ALLEGRO_MIXER *mixer)`
- `ALLEGRO_MIXER *al_get_default_mixer(void)`
- `bool al_set_default_mixer(ALLEGRO_MIXER *mixer)`
- `bool al_restore_default_mixer(void)`
- `bool al_attach_mixer_to_mixer(ALLEGRO_MIXER *stream, ALLEGRO_MIXER *mixer)`
- `bool al_attach_sample_instance_to_mixer(ALLEGRO_SAMPLE_INSTANCE *spl, ALLEGRO_MIXER *mixer)`
- `bool al_attach_audio_stream_to_mixer(ALLEGRO_AUDIO_STREAM *stream, ALLEGRO_MIXER *mixer)`
- `unsigned int al_get_mixer_frequency(const ALLEGRO_MIXER *mixer)`
- `bool al_set_mixer_frequency(ALLEGRO_MIXER *mixer, unsigned int val)`
- `ALLEGRO_CHANNEL_CONF al_get_mixer_channels(const ALLEGRO_MIXER *mixer)`
- `ALLEGRO_AUDIO_DEPTH al_get_mixer_depth(const ALLEGRO_MIXER *mixer)`
- `float al_get_mixer_gain(const ALLEGRO_MIXER *mixer)`
- `bool al_set_mixer_gain(ALLEGRO_MIXER *mixer, float new_gain)`
- `ALLEGRO_MIXER_QUALITY al_get_mixer_quality(const ALLEGRO_MIXER *mixer)`
- `bool al_set_mixer_quality(ALLEGRO_MIXER *mixer, ALLEGRO_MIXER_QUALITY new_quality)`
- `bool al_get_mixer_playing(const ALLEGRO_MIXER *mixer)`
- `bool al_set_mixer_playing(ALLEGRO_MIXER *mixer, bool val)`
- `bool al_get_mixer_attached(const ALLEGRO_MIXER *mixer)`
- `bool al_detach_mixer(ALLEGRO_MIXER *mixer)`
- `void al_destroy_audio_stream(ALLEGRO_AUDIO_STREAM *stream)`

- ALLEGRO_EVENT_SOURCE `*al_get_audio_stream_event_source(ALLEGRO_AUDIO_STREAM *stream)`
- void `al_drain_audio_stream(ALLEGRO_AUDIO_STREAM *stream)`
- bool `al_rewind_audio_stream(ALLEGRO_AUDIO_STREAM *stream)`
- unsigned int `al_get_audio_stream_frequency(const ALLEGRO_AUDIO_STREAM *stream)`
- ALLEGRO_CHANNEL_CONF `al_get_audio_stream_channels(const ALLEGRO_AUDIO_STREAM *stream)`
- ALLEGRO_AUDIO_DEPTH `al_get_audio_stream_depth(const ALLEGRO_AUDIO_STREAM *stream)`
- unsigned int `al_get_audio_stream_length(const ALLEGRO_AUDIO_STREAM *stream)`
- float `al_get_audio_stream_speed(const ALLEGRO_AUDIO_STREAM *stream)`
- bool `al_set_audio_stream_speed(ALLEGRO_AUDIO_STREAM *stream, float val)`
- float `al_get_audio_stream_gain(const ALLEGRO_AUDIO_STREAM *stream)`
- bool `al_set_audio_stream_gain(ALLEGRO_AUDIO_STREAM *stream, float val)`
- float `al_get_audio_stream_pan(const ALLEGRO_AUDIO_STREAM *stream)`
- bool `al_set_audio_stream_pan(ALLEGRO_AUDIO_STREAM *stream, float val)`
- bool `al_get_audio_stream_playing(const ALLEGRO_AUDIO_STREAM *stream)`
- bool `al_set_audio_stream_playing(ALLEGRO_AUDIO_STREAM *stream, bool val)`
- ALLEGRO_PLAYMODE `al_get_audio_stream_playmode(const ALLEGRO_AUDIO_STREAM *stream)`
- bool `al_set_audio_stream_playmode(ALLEGRO_AUDIO_STREAM *stream, ALLEGRO_PLAYMODE val)`
- bool `al_get_audio_stream_attached(const ALLEGRO_AUDIO_STREAM *stream)`
- bool `al_detach_audio_stream(ALLEGRO_AUDIO_STREAM *stream)`
- void `*al_get_audio_stream_fragment(const ALLEGRO_AUDIO_STREAM *stream)`
- bool `al_set_audio_stream_fragment(ALLEGRO_AUDIO_STREAM *stream, void *val)`
- unsigned int `al_get_audio_stream_fragments(const ALLEGRO_AUDIO_STREAM *stream)`
- unsigned int `al_get_available_audio_stream_fragments(const ALLEGRO_AUDIO_STREAM *stream)`
- bool `al_seek_audio_stream_secs(ALLEGRO_AUDIO_STREAM *stream, double time)`
- double `al_get_audio_stream_position_secs(ALLEGRO_AUDIO_STREAM *stream)`
- double `al_get_audio_stream_length_secs(ALLEGRO_AUDIO_STREAM *stream)`
- bool `al_set_audio_stream_loop_secs(ALLEGRO_AUDIO_STREAM *stream, double start, double end)`
- ALLEGRO_SAMPLE `*al_load_sample(const char *filename)`
- ALLEGRO_SAMPLE `al_load_sample_f(ALLEGRO_FILE fp, const char *ident)`
- ALLEGRO_AUDIO_STREAM `*al_load_audio_stream(const char *filename, size_t buffer_count, unsigned int samples)`
- ALLEGRO_AUDIO_STREAM `al_load_audio_stream_f(ALLEGRO_FILE fp, const char *ident, size_t buffer_count, unsigned int samples)`
- bool `al_save_sample(const char *filename, ALLEGRO_SAMPLE *spl)`
- bool `al_save_sample_f(ALLEGRO_FILE *fp, const char *ident, ALLEGRO_SAMPLE *spl)`

- `bool al_init_acodec_addon(void)`
- `uint32_t al_get_allegro_acodec_version(void)`
- `ALLEGRO_COLOR al_color_cmyk(float c, float m, float y, float k)`
- `void al_color_cmyk_to_rgb(float cyan, float magenta, float yellow, float key, float *red, float *green, float *blue)`
- `ALLEGRO_COLOR al_color_hsl(float h, float s, float l)`
- `void al_color_hsl_to_rgb(float hue, float saturation, float lightness, float *red, float *green, float *blue)`
- `ALLEGRO_COLOR al_color_hsv(float h, float s, float v)`
- `void al_color_hsv_to_rgb(float hue, float saturation, float value, float *red, float *green, float *blue)`
- `ALLEGRO_COLOR al_color_html(char const *string)`
- `void al_color_html_to_rgb(char const *string, float *red, float *green, float *blue)`
- `void al_color_rgb_to_html(float red, float green, float blue, char *string)`
- `ALLEGRO_COLOR al_color_name(char const *name)`
- `bool al_color_name_to_rgb(char const *name, float *r, float *g, float *b)`
- `void al_color_rgb_to_cmyk(float red, float green, float blue, float *cyan, float *magenta, float *yellow, float *key)`
- `void al_color_rgb_to_hsl(float red, float green, float blue, float *hue, float *saturation, float *lightness)`
- `void al_color_rgb_to_hsv(float red, float green, float blue, float *hue, float *saturation, float *value)`
- `char const *al_color_rgb_to_name(float r, float g, float b)`
- `void al_color_rgb_to_yuv(float red, float green, float blue, float *y, float *u, float *v)`
- `ALLEGRO_COLOR al_color_yuv(float y, float u, float v)`
- `void al_color_yuv_to_rgb(float y, float u, float v, float *red, float *green, float *blue)`
- `uint32_t al_get_allegro_color_version(void)`
- `void al_init_font_addon(void)`
- `void al_shutdown_font_addon(void)`
- `ALLEGRO_FONT *al_load_font(char const *filename, int size, int flags)`
- `void al_destroy_font(ALLEGRO_FONT *f)`
- `int al_get_font_ascent(const ALLEGRO_FONT *f)`
- `int al_get_font_descent(const ALLEGRO_FONT *f)`
- `int al_get_text_width(const ALLEGRO_FONT *f, const char *str)`
- `int al_get_ustr_width(const ALLEGRO_FONT *f, ALLEGRO_USTR const *ustr)`
- `void al_draw_text(const ALLEGRO_FONT *font, ALLEGRO_COLOR color, float x, float y, int flags, char const *text)`
- `void al_draw_ustr(const ALLEGRO_FONT *font, ALLEGRO_COLOR color, float x, float y, int flags, const ALLEGRO_USTR *ustr)`
- `void al_draw_justified_text(const ALLEGRO_FONT *font, ALLEGRO_COLOR color, float x1, float x2, float y, float diff, int flags, const char *text)`
- `void al_draw_justified_ustr(const ALLEGRO_FONT *font, ALLEGRO_COLOR color, float x1, float x2, float y, float diff, int flags, const ALLEGRO_USTR *ustr)`

- `void al_get_text_dimensions(const ALLEGRO_FONT *f, char const *text, int *bbx, int *bby, int *bbw, int *bbh)`
- `void al_get_ustr_dimensions(const ALLEGRO_FONT *f, ALLEGRO_USTR const *ustr, int *bbx, int *bby, int *bbw, int *bbh)`
- `uint32_t al_get_allegro_font_version(void)`
- `ALLEGRO_FONT *al_grab_font_from_bitmap(ALLEGRO_BITMAP *bmp, int ranges_n, const int ranges[])`
- `ALLEGRO_FONT *al_load_bitmap_font(const char *fname)`
- `ALLEGRO_FONT *al_create_builtin_font(void)`
- `bool al_init_ttf_addon(void)`
- `void al_shutdown_ttf_addon(void)`
- `ALLEGRO_FONT *al_load_ttf_font(char const *filename, int size, int flags)`
- `ALLEGRO_FONT *al_load_ttf_font_f(ALLEGRO_FILE *file, char const *filename, int size, int flags)`
- `ALLEGRO_FONT *al_load_ttf_font_stretch(char const *filename, int w, int h, int flags)`
- `ALLEGRO_FONT *al_load_ttf_font_stretch_f(ALLEGRO_FILE *file, char const *filename, int w, int h, int flags)`
- `uint32_t al_get_allegro_ttf_version(void)`
- `bool al_init_image_addon(void)`
- `void al_shutdown_image_addon(void)`
- `uint32_t al_get_allegro_image_version(void)`
- `ALLEGRO_FILE *al_open_memfile(void *mem, int64_t size, const char *mode)`
- `uint32_t al_get_allegro_memfile_version(void)`
- `bool al_init_native_dialog_addon(void)`
- `void al_shutdown_native_dialog_addon(void)`
- `ALLEGRO_FILECHOOSER *al_create_native_file_dialog(char const *initial_path, char const *title, char const *patterns, int mode)`
- `bool al_show_native_file_dialog(ALLEGRO_DISPLAY *display, ALLEGRO_FILECHOOSER *dialog)`
- `int al_get_native_file_dialog_count(const ALLEGRO_FILECHOOSER *dialog)`
- `const char *al_get_native_file_dialog_path(const ALLEGRO_FILECHOOSER *dialog, size_t i)`
- `void al_destroy_native_file_dialog(ALLEGRO_FILECHOOSER *dialog)`
- `int al_show_native_message_box(ALLEGRO_DISPLAY *display, char const *title, char const *heading, char const *text, char const *buttons, int flags)`
- `ALLEGRO_TEXTLOG *al_open_native_text_log(char const *title, int flags)`
- `void al_close_native_text_log(ALLEGRO_TEXTLOG *textlog)`
- `uint32_t al_get_allegro_native_dialog_version(void)`
- `void al_set_physfs_file_interface(void)`
- `uint32_t al_get_allegro_physfs_version(void)`
- `uint32_t al_get_allegro_primitives_version(void)`
- `bool al_init_primitives_addon(void)`

- `void al_shutdown_primitives_addon(void)`
- `void al_draw_line(float x1, float y1, float x2, float y2, ALLEGRO_COLOR color, float thickness)`
- `void al_draw_triangle(float x1, float y1, float x2, float y2, float x3, float y3, ALLEGRO_COLOR color, float thickness)`
- `void al_draw_filled_triangle(float x1, float y1, float x2, float y2, float x3, float y3, ALLEGRO_COLOR color)`
- `void al_draw_rectangle(float x1, float y1, float x2, float y2, ALLEGRO_COLOR color, float thickness)`
- `void al_draw_filled_rectangle(float x1, float y1, float x2, float y2, ALLEGRO_COLOR color)`
- `void al_draw_rounded_rectangle(float x1, float y1, float x2, float y2, float rx, float ry, ALLEGRO_COLOR color, float thickness)`
- `void al_draw_filled_rounded_rectangle(float x1, float y1, float x2, float y2, float rx, float ry, ALLEGRO_COLOR color)`
- `void al_calculate_arc(float* dest, int stride, float cx, float cy, float rx, float ry, float start_theta, float delta_theta, float thickness, int num_points)`
- `void al_draw_pieslice(float cx, float cy, float r, float start_theta, float delta_theta, ALLEGRO_COLOR color, float thickness)`
- `void al_draw_filled_pieslice(float cx, float cy, float r, float start_theta, float delta_theta, ALLEGRO_COLOR color)`
- `void al_draw_ellipse(float cx, float cy, float rx, float ry, ALLEGRO_COLOR color, float thickness)`
- `void al_draw_filled_ellipse(float cx, float cy, float rx, float ry, ALLEGRO_COLOR color)`
- `void al_draw_circle(float cx, float cy, float r, ALLEGRO_COLOR color, float thickness)`
- `void al_draw_filled_circle(float cx, float cy, float r, ALLEGRO_COLOR color)`
- `void al_draw_arc(float cx, float cy, float r, float start_theta, float delta_theta, ALLEGRO_COLOR color, float thickness)`
- `void al_draw_elliptical_arc(float cx, float cy, float rx, float ry, float start_theta, float delta_theta, ALLEGRO_COLOR color, float thickness)`
- `void al_draw_ribbon(const float *points, int points_stride, ALLEGRO_COLOR color, float thickness, int num_segments)`
- `int al_draw_prim(const void* vtxs, const ALLEGRO_VERTEX_DECL* decl, ALLEGRO_BITMAP* texture, int start, int end, int type)`
- `int al_draw_indexed_prim(const void* vtxs, const ALLEGRO_VERTEX_DECL* decl, ALLEGRO_BITMAP* texture, const int* indices, int num_vtx, int type)`
- `ALLEGRO_VERTEX_DECL* al_create_vertex_decl(const ALLEGRO_VERTEX_ELEMENT* elements, int stride)`
- `void al_destroy_vertex_decl(ALLEGRO_VERTEX_DECL* decl)`

RINGLIBSDL FUNCTIONS REFERENCE

- void SDL_RenderCopy2(SDL_Renderer *,SDL_Texture *)
- void SDL_Delay(int)
- void SDL_Init(int)
- int SDL_InitSubSystem(Uint32 flags)
- void SDL_Quit(void)
- void SDL_QuitSubSystem(Uint32 flags)
- void SDL_SetMainReady(void)
- Uint32 SDL_WasInit(Uint32 flags)
- SDL_bool SDL_SetHint(const char *name,const char *value)
- SDL_bool SDL_SetHintWithPriority(const char *name,const char *value,SDL_HintPriority priority)
- void SDL_ClearError(void)
- const char *SDL_GetError(void)
- SDL_LogPriority SDL_LogGetPriority(int category)
- void SDL_LogResetPriorities(void)
- void SDL_LogSetAllPriority(SDL_LogPriority priority)
- SDL_AssertionHandler SDL_GetDefaultAssertionHandler(void)
- void SDL_ResetAssertionReport(void)
- void SDL_SetAssertionHandler(SDL_AssertionHandler handler,void *userdata)
- void SDL_TriggerBreakpoint(void)
- void SDL_assert(int)
- void SDL_assert_paranoid(int)
- void SDL_assert_release(int)
- const char * SDL_GetRevision(void)
- int SDL_GetRevisionNumber(void)
- void SDL_GetVersion(SDL_version *ver)
- SDL_Window *SDL_CreateWindow(const char * title,int x, int y,int w,int h,Uint32 flags)
- void SDL_DestroyWindow(SDL_Window *window)

- void SDL_DisableScreenSaver(void)
- void SDL_EnableScreenSaver(void)
- SDL_GLCtx SDL_GL_CreateContext(SDL_Window *window)
- void SDL_GL_DeleteContext(SDL_GLCtx ctx)
- SDL_bool SDL_GL_ExtensionSupported(const char *extension)
- int SDL_GL_GetAttribute(SDL_GLAttr attr,int *value)
- SDL_GLCtx SDL_GL_GetCurrentContext(void)
- SDL_Window *SDL_GL_GetCurrentWindow(void)
- void SDL_GL_GetDrawableSize(SDL_Window *window,int *w,int *h)
- void *SDL_GL_GetProcAddress(const char *proc)
- int SDL_GL_GetSwapInterval(void)
- int SDL_GL_LoadLibrary(const char *path)
- int SDL_GL_MakeCurrent(SDL_Window *window,SDL_GLCtx ctx)
- void SDL_GL_ResetAttributes(void)
- int SDL_GL_SetAttribute(SDL_GLAttr attr,int value)
- int SDL_GL_SetSwapInterval(int interval)
- void SDL_GL_SwapWindow(SDL_Window *window)
- void SDL_GL_UnloadLibrary(void)
- SDL_DisplayMode *SDL_GetClosestDisplayMode(int displayIndex,SDL_DisplayMode *mode,SDL_DisplayMode *closest)
- int SDL_GetCurrentDisplayMode(int displayIndex,SDL_DisplayMode *mode)
- const char *SDL_GetCurrentVideoDriver(void)
- int SDL_GetDesktopDisplayMode(int displayIndex,SDL_DisplayMode *mode)
- int SDL_GetDisplayBounds(int displayIndex,SDL_Rect *rect)
- int SDL_GetNumVideoDisplays(void)
- int SDL_GetNumVideoDrivers(void)
- const char * SDL_GetVideoDriver(int index)
- void *SDL_GetWindowData(SDL_Window *window,const char *name)
- int SDL_GetWindowDisplayIndex(SDL_Window *window)
- int SDL_GetWindowDisplayMode(SDL_Window *window,SDL_DisplayMode *mode)
- Uint32 SDL_GetWindowFlags(SDL_Window *window)
- SDL_Window *SDL_GetWindowFromID(Uint32 id)
- int SDL_GetWindowGammaRamp(SDL_Window *window,Uint16 *red,Uint16 *green,Uint16 *blue)
- SDL_bool SDL_GetWindowGrab(SDL_Window *window)
- Uint32 SDL_GetWindowID(SDL_Window* window)
- void SDL_GetWindowMaximumSize(SDL_Window *window,int *w,int *h)

- void SDL_GetWindowMinimumSize(SDL_Window *window,int *w,int *h)
- void SDL_GetWindowPosition(SDL_Window *window,int *x,int *y)
- void SDL_GetWindowSize(SDL_Window *window,int *w,int *h)
- SDL_Surface *SDL_GetWindowSurface(SDL_Window *window)
- const char *SDL_GetWindowTitle(SDL_Window *window)
- SDL_bool SDL_IsScreenSaverEnabled(void)
- void SDL_MaximizeWindow(SDL_Window *window)
- void SDL_MinimizeWindow(SDL_Window *window)
- void SDL_RaiseWindow(SDL_Window *window)
- void SDL_RestoreWindow(SDL_Window *window)
- void SDL_SetWindowBordered(SDL_Window *window,SDL_bool bordered)
- int SDL_SetWindowBrightness(SDL_Window *window,float brightness)
- void *SDL_SetWindowData(SDL_Window *window,const char *name,void *userdata)
- int SDL_SetWindowDisplayMode(SDL_Window *window,const SDL_DisplayMode *mode)
- int SDL_SetWindowFullscreen(SDL_Window *window,Uint32 flags)
- int SDL_SetWindowGammaRamp(SDL_Window window,const Uint16 *red,const Uint16 *green,const Uint16 blue)
- void SDL_SetWindowGrab(SDL_Window *window,SDL_bool grabbed)
- void SDL_SetWindowMinimumSize(SDL_Window* window,int min_w,int min_h)
- void SDL_SetWindowSize(SDL_Window *window,int w,int h)
- void SDL_SetWindowTitle(SDL_Window *window,const char *title)
- int SDL_ShowMessageBox(const SDL_MessageBoxData *messageboxdata,int *buttonid)
- int SDL_ShowSimpleMessageBox(Uint32 flags,const char *title,const char *message,SDL_Window *window)
- void SDL_ShowWindow(SDL_Window *window)
- int SDL_UpdateWindowSurface(SDL_Window *window)
- int SDL_UpdateWindowSurfaceRects(SDL_Window *window,const SDL_Rect *rects,int numrects)
- int SDL_VideoInit(const char *driver_name)
- void SDL_VideoQuit(void)
- SDL_Renderer *SDL_CreateRenderer(SDL_Window *window,int index,Uint32 flags)
- SDL_Renderer *SDL_CreateSoftwareRenderer(SDL_Surface *surface)
- SDL_Texture *SDL_CreateTexture(SDL_Renderer *renderer,Uint32 format,int access,int w,int h)
- SDL_Texture *SDL_CreateTextureFromSurface(SDL_Renderer *renderer,SDL_Surface *surface)
- void SDL_DestroyTexture(SDL_Texture *texture)
- int SDL_GL_BindTexture(SDL_Texture *texture,float *texw,float *texh)
- int SDL_GL_UnbindTexture(SDL_Texture *texture)
- int SDL_GetNumRenderDrivers(void)

- int SDL_GetRenderDrawBlendMode(SDL_Renderer *renderer, SDL_BlendMode *blendMode)
- int SDL_GetRenderDrawColor(SDL_Renderer *renderer, Uint8 *r, Uint8 *g, Uint8 *b, Uint8 *a)
- int SDL_GetRenderDriverInfo(int index, SDL_RendererInfo *info)
- SDL_Texture *SDL_GetRenderTarget(SDL_Renderer *renderer)
- SDL_Renderer *SDL_GetRenderer(SDL_Window *window)
- int SDL_GetRendererInfo(SDL_Renderer *renderer, SDL_RendererInfo *info)
- int SDL_GetRendererOutputSize(SDL_Renderer *renderer, int *w, int *h)
- int SDL_GetTextureAlphaMod(SDL_Texture *texture, Uint8 *alpha)
- int SDL_GetTextureBlendMode(SDL_Texture *texture, SDL_BlendMode *blendMode)
- int SDL_GetTextureColorMod(SDL_Texture *texture, Uint8 *r, Uint8 *g, Uint8 *b)
- int SDL_LockTexture(SDL_Texture *texture, const SDL_Rect *rect, void **pixels, int *pitch)
- int SDL_QueryTexture(SDL_Texture *texture, int *format, int *access, int *w, int *h)
- int SDL_RenderClear(SDL_Renderer *renderer)
- int SDL_RenderCopy(SDL_Renderer *renderer, SDL_Texture *texture, const SDL_Rect *srcrect, const SDL_Rect *dstrect)
- int SDL_RenderCopyEx(SDL_Renderer *renderer, SDL_Texture *texture, const SDL_Rect *srcrect, const SDL_Rect *dstrect, const double angle, const SDL_Point *center, const SDL_RendererFlip flip)
- int SDL_RenderDrawLine(SDL_Renderer *renderer, int x1, int y1, int x2, int y2)
- int SDL_RenderDrawLines(SDL_Renderer *renderer, const SDL_Point *points, int count)
- int SDL_RenderDrawPoint(SDL_Renderer *renderer, int x, int y)
- int SDL_RenderDrawPoints(SDL_Renderer *renderer, const SDL_Point *points, int count)
- int SDL_RenderDrawRect(SDL_Renderer *renderer, const SDL_Rect *rect)
- int SDL_RenderDrawRects(SDL_Renderer *renderer, const SDL_Rect *rects, int count)
- int SDL_RenderFillRect(SDL_Renderer *renderer, const SDL_Rect *rect)
- int SDL_RenderFillRects(SDL_Renderer *renderer, const SDL_Rect *rects, int count)
- void SDL_RenderGetClipRect(SDL_Renderer *renderer, SDL_Rect *rect)
- void SDL_RenderGetScale(SDL_Renderer *renderer, float *scaleX, float *scaleY)
- void SDL_RenderGetViewport(SDL_Renderer *renderer, SDL_Rect *rect)
- int SDL_RenderReadPixels(SDL_Renderer *renderer, const SDL_Rect *rect, Uint32 format, void *pixels, int pitch)
- int SDL_RenderSetClipRect(SDL_Renderer *renderer, const SDL_Rect *rect)
- int SDL_RenderSetScale(SDL_Renderer *renderer, float scaleX, float scaleY)
- int SDL_RenderSetViewport(SDL_Renderer *renderer, const SDL_Rect *rect)
- SDL_bool SDL_RenderTargetSupported(SDL_Renderer *renderer)
- int SDL_SetRenderDrawBlendMode(SDL_Renderer *renderer, SDL_BlendMode blendMode)
- int SDL_SetRenderDrawColor(SDL_Renderer *renderer, Uint8 r, Uint8 g, Uint8 b, Uint8 a)
- int SDL_SetRenderTarget(SDL_Renderer *renderer, SDL_Texture *texture)

- `int SDL_SetTextureAlphaMod(SDL_Texture *texture, Uint8 alpha)`
- `int SDL_SetTextureBlendMode(SDL_Texture *texture, SDL_BlendMode blendMode)`
- `int SDL_SetTextureColorMod(SDL_Texture *texture, Uint8 r, Uint8 g, Uint8 b)`
- `void SDL_UnlockTexture(SDL_Texture *texture)`
- `int SDL_UpdateTexture(SDL_Texture *texture, const SDL_Rect *rect, const void *pixels, int pitch)`
- `int SDL_UpdateYUVTexture(SDL_Texture *texture, const SDL_Rect *rect, const Uint8 *Yplane, int Ypitch, const Uint8 *Uplane, int Upitch, const Uint8 *Vplane, int Vpitch)`
- `SDL_PixelFormat *SDL_AllocFormat(Uint32 pixel_format)`
- `SDL_Palette *SDL_AllocPalette(int ncolors)`
- `void SDL_CalculateGammaRamp(float gamma, Uint16 *ramp)`
- `void SDL_FreeFormat(SDL_PixelFormat *format)`
- `void SDL_FreePalette(SDL_Palette *palette)`
- `const char *SDL_GetPixelFormatName(Uint32 format)`
- `void SDL_GetRGB(Uint32 pixel, const SDL_PixelFormat *format, Uint8 *r, Uint8 *g, Uint8 *b)`
- `void SDL_GetRGBA(Uint32 pixel, const SDL_PixelFormat *format, Uint8 *r, Uint8 *g, Uint8 *b, Uint8 *a)`
- `Uint32 SDL_MapRGB(const SDL_PixelFormat *format, Uint8 r, Uint8 g, Uint8 b)`
- `Uint32 SDL_MapRGBA(const SDL_PixelFormat *format, Uint8 r, Uint8 g, Uint8 b, Uint8 a)`
- `Uint32 SDL_MasksToPixelFormatEnum(int bpp, Uint32 Rmask, Uint32 Gmask, Uint32 Bmask, Uint32 Amask)`
- `SDL_bool SDL_PixelFormatEnumToMasks(Uint32 format, int *bpp, Uint32 *Rmask, Uint32 *Gmask, Uint32 *Bmask, Uint32 *Amask)`
- `int SDL_SetPaletteColors(SDL_Palette *palette, const SDL_Color *colors, int firstcolor, int ncolors)`
- `int SDL_SetPixelFormatPalette(SDL_PixelFormat *format, SDL_Palette *palette)`
- `SDL_bool SDL_EnclosePoints(const SDL_Point *points, int count, const SDL_Rect *clip, SDL_Rect *result)`
- `SDL_bool SDL_HasIntersection(const SDL_Rect *A, const SDL_Rect *B)`
- `SDL_bool SDL_IntersectRect(const SDL_Rect *A, const SDL_Rect *B, SDL_Rect *result)`
- `SDL_bool SDL_IntersectRectAndLine(const SDL_Rect *rect, int *X1, int *Y1, int *X2, int *Y2)`
- `SDL_bool SDL_RectEquals(const SDL_Rect *a, const SDL_Rect *b)`
- `void SDL_UnionRect(const SDL_Rect *A, const SDL_Rect *B, SDL_Rect *result)`
- `int SDL_BlittScaled(SDL_Surface *src, const SDL_Rect *srcrect, SDL_Surface *dst, SDL_Rect *dstrect)`
- `int SDL_BlittSurface(SDL_Surface src, const SDL_Rect srcrect, SDL_Surface *dst, SDL_Rect *dstrect)`
- `int SDL_ConvertPixels(int width, int height, Uint32 src_format, const void *src, int src_pitch, Uint32 dst_format, void *dst, int dst_pitch)`
- `SDL_Surface *SDL_ConvertSurface(SDL_Surface *src, const SDL_PixelFormat *fmt, Uint32 flags)`
- `SDL_Surface *SDL_ConvertSurfaceFormat(SDL_Surface *src, Uint32 pixel_format, Uint32 flags)`
- `SDL_Surface *SDL_CreateRGBSurface(Uint32 flags, int width, int height, int depth, Uint32 Rmask, Uint32 Gmask, Uint32 Bmask, Uint32 Amask)`
- `SDL_Surface *SDL_CreateRGBSurfaceFrom(void *pixels, int width, int height, int depth, int pitch, Uint32 Rmask, Uint32 Gmask, Uint32 Bmask, Uint32 Amask)`

- `int SDL_FillRect(SDL_Surface *dst, const SDL_Rect *rect, Uint32 color)`
- `int SDL_FillRects(SDL_Surface *dst, const SDL_Rect *rects, int count, Uint32 color)`
- `void SDL_FreeSurface(SDL_Surface *surface)`
- `void SDL_GetClipRect(SDL_Surface *surface, SDL_Rect *rect)`
- `int SDL_GetColorKey(SDL_Surface *surface, Uint32 *key)`
- `int SDL_GetSurfaceAlphaMod(SDL_Surface *surface, Uint8 *alpha)`
- `int SDL_GetSurfaceBlendMode(SDL_Surface *surface, SDL_BlendMode *blendMode)`
- `int SDL_GetSurfaceColorMod(SDL_Surface *surface, Uint8 *r, Uint8 *g, Uint8 *b)`
- `SDL_Surface *SDL_LoadBMP(const char *file)`
- `SDL_Surface *SDL_LoadBMP_RW(SDL_RWops *src, int freesrc)`
- `int SDL_LockSurface(SDL_Surface *surface)`
- `int SDL_LowerBlit(SDL_Surface *src, SDL_Rect *srcRect, SDL_Surface *dst, SDL_Rect *dstRect)`
- `int SDL_LowerBlitScaled(SDL_Surface *src, SDL_Rect *srcRect, SDL_Surface *dst, SDL_Rect *dstRect)`
- `SDL_bool SDL_MUSTLOCK(SDL_Surface *surface)`
- `int SDL_SaveBMP(SDL_Surface *surface, const char *file)`
- `int SDL_SaveBMP_RW(SDL_Surface *surface, SDL_RWops *dst, int freedst)`
- `SDL_bool SDL_SetClipRect(SDL_Surface *surface, const SDL_Rect *rect)`
- `int SDL_SetColorKey(SDL_Surface *surface, int flag, Uint32 key)`
- `int SDL_SetSurfaceAlphaMod(SDL_Surface *surface, Uint8 alpha)`
- `int SDL_SetSurfaceBlendMode(SDL_Surface *surface, SDL_BlendMode blendMode)`
- `int SDL_SetSurfaceColorMod(SDL_Surface *surface, Uint8 r, Uint8 g, Uint8 b)`
- `int SDL_SetSurfacePalette(SDL_Surface *surface, SDL_Palette *palette)`
- `int SDL_SetSurfaceRLE(SDL_Surface *surface, int flag)`
- `void SDL_UnlockSurface(SDL_Surface * surface)`
- `SDL_bool SDL_GetWindowWMInfo(SDL_Window *window, SDL_SysWMInfo *info)`
- `char *SDL_GetClipboardText(void)`
- `SDL_bool SDL_HasClipboardText(void)`
- `int SDL_SetClipboardText(const char *text)`
- `void SDL_AddEventWatch(SDL_EventFilter filter, void *userdata)`
- `void SDL_DelEventWatch(SDL_EventFilter filter, void *userdata)`
- `Uint8 SDL_EventState(Uint32 type, int state)`
- `void SDL_FilterEvents(SDL_EventFilter filter, void *userdata)`
- `void SDL_FlushEvent(Uint32 type)`
- `void SDL_FlushEvents(Uint32 minType, Uint32 maxType)`
- `SDL_bool SDL_GetEventFilter(SDL_EventFilter *filter, void **userdata)`
- `Uint8 SDL_GetEventState(Uint32 type)`

- `int SDL_GetNumTouchDevices(void)`
- `int SDL_GetNumTouchFingers(SDL_TouchID touchID)`
- `SDL_TouchID SDL_GetTouchDevice(int index)`
- `SDL_Finger* SDL_GetTouchFinger(SDL_TouchID touchID,int index)`
- `SDL_bool SDL_HasEvent(Uint32 type)`
- `SDL_bool SDL_HasEvents(Uint32 minType,Uint32 maxType)`
- `int SDL_LoadDollarTemplates(SDL_TouchID touchId,SDL_RWops *src)`
- `int SDL_PeepEvents(SDL_Event *events,int numevents,SDL_eventaction action,Uint32 minType,Uint32 maxType)`
- `int SDL_PollEvent(SDL_Event *event)`
- `void SDL_PumpEvents(void)`
- `int SDL_PushEvent(SDL_Event *event)`
- `SDL_bool SDL_QuitRequested(void)`
- `int SDL_RecordGesture(SDL_TouchID touchId)`
- `Uint32 SDL_RegisterEvents(int numevents)`
- `int SDL_SaveAllDollarTemplates(SDL_RWops *dst)`
- `int SDL_SaveDollarTemplate(SDL_GestureID gestureId,SDL_RWops *dst)`
- `void SDL_SetEventFilter(SDL_EventFilter filter,void *userdata)`
- `int SDL_WaitEvent(SDL_Event *event)`
- `int SDL_WaitEventTimeout(SDL_Event *event,int timeout)`
- `SDL_Keycode SDL_GetKeyFromName(const char * name)`
- `SDL_Keycode SDL_GetKeyFromScancode(SDL_Scancode scancode)`
- `const char * SDL_GetKeyName(SDL_Keycode key)`
- `SDL_Window* SDL_GetKeyboardFocus(void)`
- `const Uint8* SDL_GetKeyboardState(int* numkeys)`
- `SDL_Keymod SDL_GetModState(void)`
- `SDL_Scancode SDL_GetScancodeFromKey(SDL_Keycode key)`
- `SDL_Scancode SDL_GetScancodeFromName(const char * name)`
- `const char * SDL_GetScancodeName(SDL_Scancode scancode)`
- `SDL_bool SDL_HasScreenKeyboardSupport(void)`
- `SDL_bool SDL_IsScreenKeyboardShown(SDL_Window* window)`
- `SDL_bool SDL_IsTextInputActive(void)`
- `void SDL_SetModState(SDL_Keymod modstate)`
- `void SDL_SetTextInputRect(SDL_Rect* rect)`
- `void SDL_StartTextInput(void)`
- `void SDL_StopTextInput(void)`

- `SDL_Cursor *SDL_CreateCursor(const Uint8 *data, const Uint8 *mask, int w, int h, int hot_x, int hot_y)`
- `void SDL_FreeCursor(SDL_Cursor *cursor)`
- `SDL_Cursor *SDL_GetCursor(void)`
- `SDL_Cursor *SDL_GetDefaultCursor(void)`
- `Uint32 SDL_GetMouseState(int *x, int *y)`
- `SDL_bool SDL_GetRelativeMouseMode(void)`
- `Uint32 SDL_GetRelativeMouseState(int *x, int *y)`
- `void SDL_SetCursor(SDL_Cursor *cursor)`
- `int SDL_SetRelativeMouseMode(SDL_bool enabled)`
- `int SDL_ShowCursor(int toggle)`
- `void SDL_JoystickClose(SDL_Joystick *joystick)`
- `SDL_bool SDL_JoystickGetAttached(SDL_Joystick *joystick)`
- `Sint16 SDL_JoystickGetAxis(SDL_Joystick *joystick, int axis)`
- `int SDL_JoystickGetBall(SDL_Joystick *joystick, int ball, int *dx, int *dy)`
- `Uint8 SDL_JoystickGetButton(SDL_Joystick *joystick, int button)`
- `SDL_JoystickGUID SDL_JoystickGetDeviceGUID(int device_index)`
- `SDL_JoystickGUID SDL_JoystickGetGUID(SDL_Joystick *joystick)`
- `SDL_JoystickGUID SDL_JoystickGetGUIDFromString(const char *pchGUID)`
- `void SDL_JoystickGetGUIDString(SDL_JoystickGUID guid, char *pszGUID, int cbGUID)`
- `Uint8 SDL_JoystickGetHat(SDL_Joystick *joystick, int hat)`
- `SDL_JoystickID SDL_JoystickInstanceID(SDL_Joystick *joystick)`
- `const char *SDL_JoystickName(SDL_Joystick *joystick)`
- `const char *SDL_JoystickNameForIndex(int device_index)`
- `int SDL_JoystickNumAxes(SDL_Joystick *joystick)`
- `int SDL_JoystickNumBalls(SDL_Joystick *joystick)`
- `int SDL_JoystickNumButtons(SDL_Joystick *joystick)`
- `int SDL_JoystickNumHats(SDL_Joystick *joystick)`
- `SDL_Joystick *SDL_JoystickOpen(int device_index)`
- `void SDL_JoystickUpdate(void)`
- `int SDL_NumJoysticks(void)`
- `int SDL_GameControllerAddMapping(const char *mappingString)`
- `int SDL_GameControllerAddMappingsFromFile(const char *filename)`
- `int SDL_GameControllerAddMappingsFromRW(SDL_RWops *rw, int freerw)`
- `void SDL_GameControllerClose(SDL_GameController *gamecontroller)`
- `int SDL_GameControllerEventState(int state)`
- `Sint16 SDL_GameControllerGetAxis(SDL_GameController *gamecontroller, SDL_GameControllerAxis axis)`

- `SDL_GameControllerAxis SDL_GameControllerGetAxisFromString(const char *pchString)`
- `SDL_GameControllerButtonBind SDL_GameControllerGetBindForAxis(SDL_GameController *gamecontroller, SDL_GameControllerAxis axis)`
- `SDL_GameControllerButtonBind SDL_GameControllerGetBindForButton(SDL_GameController *gamecontroller, SDL_GameControllerButton button)`
- `Uint8 SDL_GameControllerGetButton(SDL_GameController *gamecontroller, SDL_GameControllerButton button)`
- `SDL_GameControllerButton SDL_GameControllerGetButtonFromString(const char *pchString)`
- `SDL_Joystick *SDL_GameControllerGetJoystick(SDL_GameController *gamecontroller)`
- `const char *SDL_GameControllerGetStringForAxis(SDL_GameControllerAxis axis)`
- `const char *SDL_GameControllerGetStringForButton(SDL_GameControllerButton button)`
- `char *SDL_GameControllerMapping(SDL_GameController *gamecontroller)`
- `char *SDL_GameControllerMappingForGUID(SDL_JoystickGUID guid)`
- `const char *SDL_GameControllerName(SDL_GameController *gamecontroller)`
- `const char *SDL_GameControllerNameForIndex(int joystick_index)`
- `SDL_GameController* SDL_GameControllerOpen(int joystick_index)`
- `void SDL_GameControllerUpdate(void)`
- `SDL_bool SDL_IsGameController(int joystick_index)`
- `void SDL_HapticClose(SDL_Haptic* haptic)`
- `void SDL_HapticDestroyEffect(SDL_Haptic *haptic, int effect)`
- `int SDL_HapticEffectSupported(SDL_Haptic *haptic, SDL_HapticEffect *effect)`
- `int SDL_HapticGetEffectStatus(SDL_Haptic *haptic, int effect)`
- `int SDL_HapticIndex(SDL_Haptic *haptic)`
- `const char *SDL_HapticName(int device_index)`
- `int SDL_HapticNewEffect(SDL_Haptic *haptic, SDL_HapticEffect *effect)`
- `int SDL_HapticNumAxes(SDL_Haptic *haptic)`
- `int SDL_HapticNumEffects(SDL_Haptic *haptic)`
- `int SDL_HapticNumEffectsPlaying(SDL_Haptic *haptic)`
- `SDL_Haptic *SDL_HapticOpen(int device_index)`
- `SDL_Haptic *SDL_HapticOpenFromJoystick(SDL_Joystick *joystick)`
- `SDL_Haptic *SDL_HapticOpenFromMouse(void)`
- `int SDL_HapticOpened(int device_index)`
- `int SDL_HapticPause(SDL_Haptic *haptic)`
- `unsigned int SDL_HapticQuery(SDL_Haptic *haptic)`
- `int SDL_HapticRumbleInit(SDL_Haptic *haptic)`
- `int SDL_HapticRumblePlay(SDL_Haptic *haptic, float strength, Uint32 length)`
- `int SDL_HapticRumbleStop(SDL_Haptic *haptic)`

- `int SDL_HapticRumbleSupported(SDL_Haptic *haptic)`
- `int SDL_HapticRunEffect(SDL_Haptic *haptic,int effect,Uint32 iterations)`
- `int SDL_HapticSetAutocenter(SDL_Haptic *haptic,int autocenter)`
- `int SDL_HapticSetGain(SDL_Haptic *haptic,int gain)`
- `int SDL_HapticStopAll(SDL_Haptic *haptic)`
- `int SDL_HapticStopEffect(SDL_Haptic *haptic,int effect)`
- `int SDL_HapticUnpause(SDL_Haptic *haptic)`
- `int SDL_HapticUpdateEffect(SDL_Haptic *haptic,int effect,SDL_HapticEffect *data)`
- `int SDL_JoystickIsHaptic(SDL_Joystick *joystick)`
- `int SDL_MouseIsHaptic(void)`
- `int SDL_NumHaptics(void)`
- `int SDL_AudioInit(const char * driver_name)`
- `void SDL_AudioQuit(void)`
- `int SDL_BuildAudioCVT(SDL_AudioCVT *cvt,SDL_AudioFormat src_format,Uint8 src_channels,int src_rate,SDL_AudioFormat dst_format,Uint8 dst_channels,int dst_rate)`
- `void SDL_CloseAudioDevice(SDL_AudioDeviceID dev)`
- `int SDL_ConvertAudio(SDL_AudioCVT *cvt)`
- `void SDL_FreeWAV(Uint8 *audio_buf)`
- `const char * SDL_GetAudioDeviceName(int index,int iscapture)`
- `SDL_AudioStatus SDL_GetAudioDeviceStatus(SDL_AudioDeviceID dev)`
- `const char * SDL_GetAudioDriver(int index)`
- `SDL_AudioStatus SDL_GetAudioStatus(void)`
- `const char * SDL_GetCurrentAudioDriver(void)`
- `int SDL_GetNumAudioDevices(int iscapture)`
- `int SDL_GetNumAudioDrivers(void)`
- `SDL_AudioSpec *SDL_LoadWAV_RW(SDL_RWops *src,int freesrc,SDL_AudioSpec *spec,Uint8 **audio_buf,Uint32 *audio_len)`
- `void SDL_LockAudio(void)`
- `void SDL_LockAudioDevice(SDL_AudioDeviceID dev)`
- `void SDL_MixAudio(Uint8 dst,const Uint8 src,Uint32 len,int volume)`
- `void SDL_MixAudioFormat(Uint8 *dst,const Uint8 *src,SDL_AudioFormat format,Uint32 len,int volume)`
- `int SDL_OpenAudio(SDL_AudioSpec *desired,SDL_AudioSpec *obtained)`
- `SDL_AudioDeviceID SDL_OpenAudioDevice(const char *device,int iscapture,const SDL_AudioSpec *desired,SDL_AudioSpec *obtained,int allowed_changes)`
- `void SDL_PauseAudio(int pause_on)`
- `void SDL_PauseAudioDevice(SDL_AudioDeviceID dev,int pause_on)`
- `void SDL_UnlockAudio(void)`

- void SDL_UnlockAudioDevice(SDL_AudioDeviceID dev)
- char *SDL_GetBasePath(void)
- char *SDL_GetPrefPath(const char *org,const char *app)
- SDL_RWops *SDL_AllocRW(void)
- void SDL_FreeRW(SDL_RWops *area)
- SDL_RWops *SDL_RWFromConstMem(const void mem,int size)*
- SDL_RWops *SDL_RWFromFP(void *fp,SDL_bool autoclose)
- SDL_RWops *SDL_RWFromFile(const char *file,const char *mode)
- SDL_RWops *SDL_RWFromMem(void *mem,int size)
- int SDL_RWclose(struct SDL_RWops *context)
- size_t SDL_RWread(struct SDL_RWops *context,void *ptr,size_t size,size_t maxnum)
- Sint64 SDL_RWseek(SDL_RWops *context,Sint64 offset,int whence)
- Sint64 SDL_RWsize(SDL_RWops *context)
- Sint64 SDL_RWtell(struct SDL_RWops *context)
- size_t SDL_RWwrite(struct SDL_RWops *context,const void *ptr,size_t size,size_t num)
- Uint16 SDL_ReadBE16(SDL_RWops *src)
- Uint32 SDL_ReadBE32(SDL_RWops *src)
- Uint64 SDL_ReadBE64(SDL_RWops *src)
- Uint16 SDL_ReadLE16(SDL_RWops *src)
- Uint32 SDL_ReadLE32(SDL_RWops *src)
- Uint64 SDL_ReadLE64(SDL_RWops *src)
- Uint8 SDL_ReadU8(SDL_RWops *src)
- size_t SDL_WriteBE16(SDL_RWops *dst,Uint16 value)
- size_t SDL_WriteBE32(SDL_RWops *dst,Uint32 value)
- size_t SDL_WriteBE64(SDL_RWops *dst,Uint64 value)
- size_t SDL_WriteLE16(SDL_RWops *dst,Uint16 value)
- size_t SDL_WriteLE32(SDL_RWops *dst,Uint32 value)
- size_t SDL_WriteLE64(SDL_RWops *dst,Uint64 value)
- size_t SDL_WriteU8(SDL_RWops *dst,Uint8 value)
- void *SDL_LoadFunction(void *handle,const char *name)
- void *SDL_LoadObject(const char *sofile)
- void SDL_UnloadObject(void *handle)
- const char *SDL_GetPlatform(void)
- int SDL_GetCPUCacheLineSize(void)
- int SDL_GetCPUCount(void)
- int SDL_GetSystemRAM(void)

- `SDL_bool SDL_Has3DNow(void)`
- `SDL_bool SDL_HasAVX(void)`
- `SDL_bool SDL_HasMMX(void)`
- `SDL_bool SDL_HasRDTSC(void)`
- `SDL_bool SDL_HasSSE(void)`
- `SDL_bool SDL_HasSSE2(void)`
- `SDL_bool SDL_HasSSE3(void)`
- `SDL_bool SDL_HasSSE41(void)`
- `SDL_bool SDL_HasSSE42(void)`
- `SDL_PowerState SDL_GetPowerInfo(int *secs,int *pct)`
- `double SDL_acos(double x)`
- `int IMG_Init(int flags)`
- `void IMG_Quit(void)`
- `SDL_Surface *IMG_Load(const char *file)`
- `SDL_Surface *IMG_Load_RW(SDL_RWops *src, int freesrc)`
- `SDL_Surface *IMG_LoadTyped_RW(SDL_RWops *src, int freesrc, char *type)`
- `SDL_Surface *IMG_LoadCUR_RW(SDL_RWops *src)`
- `SDL_Surface *IMG_LoadBMP_RW(SDL_RWops *src)`
- `SDL_Surface *IMG_LoadPNM_RW(SDL_RWops *src)`
- `SDL_Surface *IMG_LoadXPM_RW(SDL_RWops *src)`
- `SDL_Surface *IMG_LoadXCF_RW(SDL_RWops *src)`
- `SDL_Surface *IMG_LoadPCX_RW(SDL_RWops *src)`
- `SDL_Surface *IMG_LoadGIF_RW(SDL_RWops *src)`
- `SDL_Surface *IMG_LoadJPG_RW(SDL_RWops *src)`
- `SDL_Surface *IMG_LoadTIF_RW(SDL_RWops *src)`
- `SDL_Surface *IMG_LoadPNG_RW(SDL_RWops *src)`
- `SDL_Surface *IMG_LoadTGA_RW(SDL_RWops *src)`
- `SDL_Surface *IMG_LoadLBM_RW(SDL_RWops *src)`
- `SDL_Surface *IMG_LoadXV_RW(SDL_RWops *src)`
- `SDL_Surface *IMG_ReadXPMFromArray(char **xpm)`
- `int IMG_isCUR(SDL_RWops *src)`
- `int IMG_isICO(SDL_RWops *src)`
- `int IMG_isBMP(SDL_RWops *src)`
- `int IMG_isPNM(SDL_RWops *src)`
- `int IMG_isXPM(SDL_RWops *src)`
- `int IMG_isXCF(SDL_RWops *src)`

- int IMG_isPCX(SDL_RWops *src)
- int IMG_isGIF(SDL_RWops *src)
- int IMG_isJPG(SDL_RWops *src)
- int IMG_isTIF(SDL_RWops *src)
- int IMG_isPNG(SDL_RWops *src)
- int IMG_isLBM(SDL_RWops *src)
- int IMG_isXV(SDL_RWops *src)
- int TTF_Init(void)
- int TTF_WasInit(void)
- void TTF_Quit(void)
- TTF_Font *TTF_OpenFont(const char *file, int psize)
- TTF_Font *TTF_OpenFontRW(SDL_RWops *src, int freesrc, int psize)
- TTF_Font *TTF_OpenFontIndex(const char *file, int psize, long index)
- TTF_Font *TTF_OpenFontIndexRW(SDL_RWops *src, int freesrc, int psize, long index)
- void TTF_CloseFont(TTF_Font *font)
- void TTF_ByteSwappedUNICODE(int swapped)
- int TTF_GetFontStyle(TTF_Font *font)
- void TTF_SetFontStyle(TTF_Font *font, int style)
- int TTF_GetFontOutline(TTF_Font *font)
- void TTF_SetFontOutline(TTF_Font *font, int outline)
- int TTF_GetFontHinting(TTF_Font *font)
- void TTF_SetFontHinting(TTF_Font *font, int hinting)
- int TTF_GetFontKerning(TTF_Font *font)
- void TTF_SetFontKerning(TTF_Font *font, int allowed)
- int TTF_FontHeight(const TTF_Font *font)
- int TTF_FontAscent(const TTF_Font *font)
- int TTF_FontDescent(const TTF_Font *font)
- int TTF_FontLineSkip(const TTF_Font *font)
- long TTF_FontFaces(const TTF_Font *font)
- int TTF_FontFaceIsFixedWidth(const TTF_Font *font)
- char *TTF_FontFaceFamilyName(const TTF_Font *font)
- char *TTF_FontFaceStyleName(const TTF_Font *font)
- int TTF_GlyphIsProvided(const TTF_Font *font, Uint16 ch)
- int TTF_GlyphMetrics(TTF_Font *font, Uint16 ch, int *minx, int *maxx, int *miny, int *maxy, int *advance)
- int TTF_SizeText(TTF_Font *font, const char *text, int *w, int *h)
- int TTF_SizeUTF8(TTF_Font *font, const char *text, int *w, int *h)

- `int TTF_SizeUNICODE(TTF_Font *font, const Uint16 *text, int *w, int *h)`
- `SDL_Surface *TTF_RenderText_Solid(TTF_Font *font, const char *text, SDL_Color fg)`
- `SDL_Surface *TTF_RenderUTF8_Solid(TTF_Font *font, const char *text,SDL_Color fg)`
- `SDL_Surface *TTF_RenderUNICODE_Solid(TTF_Font *font, const Uint16 *text,SDL_Color fg)`
- `SDL_Surface *TTF_RenderGlyph_Solid(TTF_Font *font, Uint16 ch, SDL_Color fg)`
- `SDL_Surface *TTF_RenderText_Shaded(TTF_Font *font, const char *text,SDL_Color fg, SDL_Color bg)`
- `SDL_Surface *TTF_RenderUTF8_Shaded(TTF_Font *font, const char *text,SDL_Color fg, SDL_Color bg)`
- `SDL_Surface *TTF_RenderUNICODE_Shaded(TTF_Font *font, const Uint16 *text,SDL_Color fg, SDL_Color bg)`
- `SDL_Surface *TTF_RenderGlyph_Shaded(TTF_Font *font, Uint16 ch, SDL_Color fg,SDL_Color bg)`
- `SDL_Surface *TTF_RenderText_Blended(TTF_Font *font, const char *text,SDL_Color fg)`
- `SDL_Surface *TTF_RenderUTF8_Blended(TTF_Font *font, const char *text,SDL_Color fg)`
- `SDL_Surface *TTF_RenderUNICODE_Blended(TTF_Font *font, const Uint16 *text,SDL_Color fg)`
- `SDL_Surface *TTF_RenderGlyph_Blended(TTF_Font *font, Uint16 ch, SDL_Color fg)`
- `int Mix_Init(int flags)`
- `void Mix_Quit(void)`
- `int Mix_OpenAudio(int frequency, Uint16 format, int channels, int chunksize)`
- `void Mix_CloseAudio(void)`
- `int Mix_QuerySpec(int *frequency, Uint16 *format, int *channels)`
- `int Mix_GetNumChunkDecoders(void)`
- `const char *Mix_GetChunkDecoder(int index)`
- `Mix_Chunk *Mix_LoadWAV(char *file)`
- `Mix_Chunk *Mix_LoadWAV_RW(SDL_RWops *src, int freesrc)`
- `Mix_Chunk *Mix_QuickLoad_WAV(Uint8 *mem)`
- `void Mix_FreeChunk(Mix_Chunk *chunk)`
- `int Mix_AllocateChannels(int numchans)`
- `int Mix_Volume(int channel, int volume)`
- `int Mix_PlayChannel(int channel, Mix_Chunk *chunk, int loops)`
- `int Mix_PlayChannelTimed(int channel, Mix_Chunk *chunk, int loops, int ticks)`
- `int Mix_FadeInChannel(int channel, Mix_Chunk *chunk, int loops, int ms)`
- `int Mix_FadeInChannelTimed(int channel, Mix_Chunk *chunk,int loops, int ms, int ticks)`
- `void Mix_Pause(int channel)`
- `void Mix_Resume(int channel)`
- `int Mix_HaltChannel(int channel)`
- `int Mix_ExpireChannel(int channel, int ticks)`
- `int Mix_FadeOutChannel(int channel, int ms)`

- int Mix_Paused(int channel)
- Mix_Fading Mix_FadingChannel(int which)
- Mix_Chunk *Mix_GetChunk(int channel)
- int Mix_ReserveChannels(int num)
- int Mix_GroupChannel(int which, int tag)
- int Mix_GroupChannels(int from, int to, int tag)
- int Mix_GroupCount(int tag)
- int Mix_GroupAvailable(int tag)
- int Mix_GroupOldest(int tag)
- int Mix_GroupNewer(int tag)
- int Mix_FadeOutGroup(int tag, int ms)
- int Mix_HaltGroup(int tag)
- int Mix_GetNumMusicDecoders(void)
- const char *Mix_GetMusicDecoder(int index)
- Mix_Music *Mix_LoadMUS(const char *file)
- void Mix_FreeMusic(Mix_Music *music)
- int Mix_PlayMusic(Mix_Music *music, int loops)
- int Mix_FadeInMusic(Mix_Music *music, int loops, int ms)
- int Mix_FadeInMusicPos(Mix_Music *music, int loops, int ms, double position)
- int Mix_PlayingMusic(void)
- int Mix_PausedMusic(void)
- Mix_Fading Mix_FadingMusic(void)
- void *Mix_GetMusicHookData(void)
- int Mix_RegisterEffect(int chan, Mix_EffectFunc_t f, Mix_EffectDone_t d, void *arg)
- int Mix_UnregisterEffect(int channel, Mix_EffectFunc_t f)
- int Mix_UnregisterAllEffects(int channel)
- int Mix_SetDistance(int channel, Uint8 distance)
- int Mix_SetPosition(int channel, Sint16 angle, Uint8 distance)
- int Mix_SetReverseStereo(int channel, int flip)
- int SDLNet_Init(void)
- void SDLNet_Quit(void)
- char *SDLNet_GetError(void)
- void SDLNet_Write16(Uint16 value, void *area)
- void SDLNet_Write32(Uint32 value, void *area)
- Uint16 SDLNet_Read16(void *area)
- Uint32 SDLNet_Read32(void *area)

- `int SDLNet_ResolveHost(IPAddress *address, const char *host, Uint16 port)`
- `const char *SDLNet_ResolveIP(IPAddress *address)`
- `TCPsocket SDLNet_TCP_Open(IPAddress *ip)`
- `void SDLNet_TCP_Close(TCPsocket sock)`
- `TCPsocket SDLNet_TCP_Accept(TCPsocket server)`
- `IPAddress *SDLNet_TCP_GetPeerAddress(TCPsocket sock)`
- `int SDLNet_TCP_Send(TCPsocket sock, const void *data, int len)`
- `int SDLNet_TCP_Recv(TCPsocket sock, void *data, int maxlen)`
- `UDPsocket SDLNet_UDP_Open(Uint16 port)`
- `void SDLNet_UDP_Close(UDPsocket sock)`
- `int SDLNet_UDP_Bind(UDPsocket sock, int channel, IPAddress *address)`
- `void SDLNet_UDP_Unbind(UDPsocket sock, int channel)`
- `IPAddress *SDLNet_UDP_GetPeerAddress(UDPsocket sock, int channel)`
- `int SDLNet_UDP_Send(UDPsocket sock, int channel, UDPpacket *packet)`
- `int SDLNet_UDP_Recv(UDPsocket sock, UDPpacket *packet)`
- `int SDLNet_UDP_SendV(UDPsocket sock, UDPpacket **packetV, int npackets)`
- `int SDLNet_UDP_RecvV(UDPsocket sock, UDPpacket **packetV)`
- `UDPpacket *SDLNet_AllocPacket(int size)`
- `int SDLNet_ResizePacket(UDPpacket *packet, int size)`
- `void SDLNet_FreePacket(UDPpacket *packet)`
- `UDPpacket **SDLNet_AllocPacketV(int howmany, int size)`
- `void SDLNet_FreePacketV(UDPpacket **packetV)`
- `SDLNet_SocketSet SDLNet_AllocSocketSet(int maxsockets)`
- `void SDLNet_FreeSocketSet(SDLNet_SocketSet set)`
- `int SDLNet_AddSocket(SDLNet_SocketSet set, SDLNet_GenericSocket sock)`
- `int SDLNet_TCP_AddSocket(SDLNet_SocketSet set, TCPsocket sock)`
- `int SDLNet_UDP_AddSocket(SDLNet_SocketSet set, UDPsocket sock)`
- `int SDLNet_DelSocket(SDLNet_SocketSet set, SDLNet_GenericSocket sock)`
- `int SDLNet_TCP_DelSocket(SDLNet_SocketSet set, TCPsocket sock)`
- `int SDLNet_UDP_DelSocket(SDLNet_SocketSet set, UDPsocket sock)`
- `int SDLNet_CheckSockets(SDLNet_SocketSet set, Uint32 timeout)`
- `int SDLNet_SocketReady(TCPsocket sock)`
- `int circleRGBA(SDL_Renderer *renderer, Sint16 x, Sint16 y, Sint16 rad, Uint8 r, Uint8 g, Uint8 b, Uint8 a)`

RINGFREEGLUT FUNCTIONS REFERENCE

- GLUT_RGB
- GLUT_RGBA
- GLUT_INDEX
- GLUT_SINGLE
- GLUT_DOUBLE
- GLUT_ACCUM
- GLUT_ALPHA
- GLUT_DEPTH
- GLUT_STENCIL
- GLUT_MULTISAMPLE
- GLUT_STEREO
- GLUT_LUMINANCE
- GLUT_KEY_F1
- GLUT_KEY_F2
- GLUT_KEY_F3
- GLUT_KEY_F4
- GLUT_KEY_F5
- GLUT_KEY_F6
- GLUT_KEY_F7
- GLUT_KEY_F8
- GLUT_KEY_F9
- GLUT_KEY_F10
- GLUT_KEY_F11
- GLUT_KEY_F12
- GLUT_KEY_LEFT
- GLUT_KEY_UP
- GLUT_KEY_RIGHT

- GLUT_KEY_DOWN
- GLUT_KEY_PAGE_UP
- GLUT_KEY_PAGE_DOWN
- GLUT_KEY_HOME
- GLUT_KEY_END
- GLUT_KEY_INSERT
- GLUT_LEFT_BUTTON
- GLUT_MIDDLE_BUTTON
- GLUT_RIGHT_BUTTON
- GLUT_DOWN
- GLUT_UP
- GLUT_LEFT
- GLUT_ENTERED
- GLUT_MENU_NOT_IN_USE
- GLUT_MENU_IN_USE
- GLUT_NOT_VISIBLE
- GLUT_VISIBLE
- GLUT_HIDDEN
- GLUT_FULLY_RETAINED
- GLUT_PARTIALLY_RETAINED
- GLUT_FULLY_COVERED
- GLUT_WINDOW_X
- GLUT_WINDOW_Y
- GLUT_WINDOW_WIDTH
- GLUT_WINDOW_HEIGHT
- GLUT_WINDOW_BUFFER_SIZE
- GLUT_WINDOW_STENCIL_SIZE
- GLUT_WINDOW_DEPTH_SIZE
- GLUT_WINDOW_RED_SIZE
- GLUT_WINDOW_GREEN_SIZE
- GLUT_WINDOW_BLUE_SIZE
- GLUT_WINDOW_ALPHA_SIZE
- GLUT_WINDOW_ACCUM_RED_SIZE
- GLUT_WINDOW_ACCUM_GREEN_SIZE
- GLUT_WINDOW_ACCUM_BLUE_SIZE
- GLUT_WINDOW_ACCUM_ALPHA_SIZE

- GLUT_WINDOW_DOUBLEBUFFER
- GLUT_WINDOW_RGBA
- GLUT_WINDOW_PARENT
- GLUT_WINDOW_NUM_CHILDREN
- GLUT_WINDOW_COLORMAP_SIZE
- GLUT_WINDOW_NUM_SAMPLES
- GLUT_WINDOW_STEREO
- GLUT_WINDOW_CURSOR
- GLUT_SCREEN_WIDTH
- GLUT_SCREEN_HEIGHT
- GLUT_SCREEN_WIDTH_MM
- GLUT_SCREEN_HEIGHT_MM
- GLUT_MENU_NUM_ITEMS
- GLUT_DISPLAY_MODE_POSSIBLE
- GLUT_INIT_WINDOW_X
- GLUT_INIT_WINDOW_Y
- GLUT_INIT_WINDOW_WIDTH
- GLUT_INIT_WINDOW_HEIGHT
- GLUT_INIT_DISPLAY_MODE
- GLUT_ELAPSED_TIME
- GLUT_WINDOW_FORMAT_ID
- GLUT_HAS_KEYBOARD
- GLUT_HAS_MOUSE
- GLUT_HAS_SPACEBALL
- GLUT_HAS_DIAL_AND_BUTTON_BOX
- GLUT_HAS_TABLET
- GLUT_NUM_MOUSE_BUTTONS
- GLUT_NUM_SPACEBALL_BUTTONS
- GLUT_NUM_BUTTON_BOX_BUTTONS
- GLUT_NUM_DIALS
- GLUT_NUM_TABLET_BUTTONS
- GLUT_DEVICE_IGNORE_KEY_REPEAT
- GLUT_DEVICE_KEY_REPEAT
- GLUT_HAS_JOYSTICK
- GLUT_OWNS_JOYSTICK
- GLUT_JOYSTICK_BUTTONS

- GLUT_JOYSTICK_AXES
- GLUT_JOYSTICK_POLL_RATE
- GLUT_OVERLAY_POSSIBLE
- GLUT_LAYER_IN_USE
- GLUT_HAS_OVERLAY
- GLUT_TRANSPARENT_INDEX
- GLUT_NORMAL_DAMAGED
- GLUT_OVERLAY_DAMAGED
- GLUT_VIDEO_RESIZE_POSSIBLE
- GLUT_VIDEO_RESIZE_IN_USE
- GLUT_VIDEO_RESIZE_X_DELTA
- GLUT_VIDEO_RESIZE_Y_DELTA
- GLUT_VIDEO_RESIZE_WIDTH_DELTA
- GLUT_VIDEO_RESIZE_HEIGHT_DELTA
- GLUT_VIDEO_RESIZE_X
- GLUT_VIDEO_RESIZE_Y
- GLUT_VIDEO_RESIZE_WIDTH
- GLUT_VIDEO_RESIZE_HEIGHT
- GLUT_NORMAL
- GLUT_OVERLAY
- GLUT_ACTIVE_SHIFT
- GLUT_ACTIVE_CTRL
- GLUT_ACTIVE_ALT
- GLUT_CURSOR_RIGHT_ARROW
- GLUT_CURSOR_LEFT_ARROW
- GLUT_CURSOR_INFO
- GLUT_CURSOR_DESTROY
- GLUT_CURSOR_HELP
- GLUT_CURSOR_CYCLE
- GLUT_CURSOR_SPRAY
- GLUT_CURSOR_WAIT
- GLUT_CURSOR_TEXT
- GLUT_CURSOR_CROSSHAIR
- GLUT_CURSOR_UP_DOWN
- GLUT_CURSOR_LEFT_RIGHT
- GLUT_CURSOR_TOP_SIDE

- GLUT_CURSOR_BOTTOM_SIDE
- GLUT_CURSOR_LEFT_SIDE
- GLUT_CURSOR_RIGHT_SIDE
- GLUT_CURSOR_TOP_LEFT_CORNER
- GLUT_CURSOR_TOP_RIGHT_CORNER
- GLUT_CURSOR_BOTTOM_RIGHT_CORNER
- GLUT_CURSOR_BOTTOM_LEFT_CORNER
- GLUT_CURSOR_INHERIT
- GLUT_CURSOR_NONE
- GLUT_CURSOR_FULL_CROSSHAIR
- GLUT_RED
- GLUT_GREEN
- GLUT_BLUE
- GLUT_KEY_REPEAT_OFF
- GLUT_KEY_REPEAT_ON
- GLUT_KEY_REPEAT_DEFAULT
- GLUT_JOYSTICK_BUTTON_A
- GLUT_JOYSTICK_BUTTON_B
- GLUT_JOYSTICK_BUTTON_C
- GLUT_JOYSTICK_BUTTON_D
- GLUT_GAME_MODE_ACTIVE
- GLUT_GAME_MODE_POSSIBLE
- GLUT_GAME_MODE_WIDTH
- GLUT_GAME_MODE_HEIGHT
- GLUT_GAME_MODE_PIXEL_DEPTH
- GLUT_GAME_MODE_REFRESH_RATE
- GLUT_GAME_MODE_DISPLAY_CHANGED
- GLUT_STROKE_ROMAN
- GLUT_STROKE_MONO_ROMAN
- GLUT_BITMAP_9_BY_15
- GLUT_BITMAP_8_BY_13
- GLUT_BITMAP_TIMES_ROMAN_10
- GLUT_BITMAP_TIMES_ROMAN_24
- GLUT_BITMAP_HELVETICA_10
- GLUT_BITMAP_HELVETICA_12
- GLUT_BITMAP_HELVETICA_18

- void glutInit(void)
- void glutDisplayFunc(const char *)
- void glutReshapeFunc(const char *)
- int glutEventWidth(void)
- int glutEventHeight(void)
- void glutIdleFunc(const char *)
- void glutKeyboardFunc(const char *)
- void glutSpecialFunc(const char *)
- void glutSpecialUpFunc(const char *)
- void glutMouseFunc(const char *)
- void glutMotionFunc(const char *)
- int glutCreateMenu(const char *)
- void glutMenuStatusFunc(const char *)
- int glutEventKey(void)
- int glutEventX(void)
- int glutEventY(void)
- int glutEventButton(void)
- int glutEventState(void)
- int glutEventValue(void)
- int glutEventStatus(void)
- void test_draw(void)
- void glutInitWindowPosition(int x, int y)
- void glutInitWindowSize(int width, int height)
- void glutInitDisplayMode(unsigned displayMode)
- void glutInitDisplayString(const char * displayMode)
- int glutCreateWindow(const char * title)
- int glutCreateSubWindow(int window, int x, int y, int width, int height)
- void glutDestroyWindow(int window)
- void glutSetWindow(int window)
- int glutGetWindow(void)
- void glutSetWindowTitle(const char * title)
- void glutSetIconTitle(const char * title)
- void glutReshapeWindow(int width, int height)
- void glutPositionWindow(int x, int y)
- void glutShowWindow(void)
- void glutHideWindow(void)

- void glutIconifyWindow(void)
- void glutPushWindow(void)
- void glutPopWindow(void)
- void glutFullScreen(void)
- void glutPostWindowRedisplay(int window)
- void glutPostRedisplay(void)
- void glutSwapBuffers(void)
- void glutWarpPointer(int x, int y)
- void glutSetCursor(int cursor)
- void glutEstablishOverlay(void)
- void glutRemoveOverlay(void)
- void glutUseLayer(GLenum layer)
- void glutPostOverlayRedisplay(void)
- void glutPostWindowOverlayRedisplay(int window)
- void glutShowOverlay(void)
- void glutHideOverlay(void)
- void glutDestroyMenu(int menu)
- int glutGetMenu(void)
- void glutSetMenu(int menu)
- void glutAddMenuEntry(const char * label, int value)
- void glutAddSubMenu(const char * label, int subMenu)
- void glutChangeToMenuEntry(int item, const char * label, int value)
- void glutChangeToSubMenu(int item, const char * label, int value)
- void glutRemoveMenuItem(int item)
- void glutAttachMenu(int button)
- void glutDetachMenu(int button)
- int glutGet(GLenum query)
- int glutDeviceGet(GLenum query)
- int glutGetModifiers(void)
- int glutLayerGet(GLenum query)
- void glutBitmapCharacter(void *font, int character)
- int glutBitmapWidth(void *font, int character)
- void glutStrokeCharacter(void *font, int character)
- int glutStrokeWidth(void *font, int character)
- GLfloat glutStrokeWidthf(void *font, int character)
- int glutBitmapLength(void *font, char * string)

- `int glutStrokeLength(void *font, char * string)`
- `GLfloat glutStrokeLengthf(void *font, char *string)`
- `void glutWireCube(double size)`
- `void glutSolidCube(double size)`
- `void glutWireSphere(double radius, GLint slices, GLint stacks)`
- `void glutSolidSphere(double radius, GLint slices, GLint stacks)`
- `void glutWireCone(double base, double height, GLint slices, GLint stacks)`
- `void glutSolidCone(double base, double height, GLint slices, GLint stacks)`
- `void glutWireTorus(double innerRadius, double outerRadius, GLint sides, GLint rings)`
- `void glutSolidTorus(double innerRadius, double outerRadius, GLint sides, GLint rings)`
- `void glutWireDodecahedron(void)`
- `void glutSolidDodecahedron(void)`
- `void glutWireOctahedron(void)`
- `void glutSolidOctahedron(void)`
- `void glutWireTetrahedron(void)`
- `void glutSolidTetrahedron(void)`
- `void glutWireIcosahedron(void)`
- `void glutSolidIcosahedron(void)`
- `void glutWireTeapot(double size)`
- `void glutSolidTeapot(double size)`
- `void glutGameModeString(const char * string)`
- `int glutEnterGameMode(void)`
- `void glutLeaveGameMode(void)`
- `int glutGameModeGet(GLenum query)`
- `int glutVideoResizeGet(GLenum query)`
- `void glutSetupVideoResizing(void)`
- `void glutStopVideoResizing(void)`
- `void glutVideoResize(int x, int y, int width, int height)`
- `void glutVideoPan(int x, int y, int width, int height)`
- `void glutSetColor(int color, GLfloat red, GLfloat green, GLfloat blue)`
- `GLfloat glutGetColor(int color, int component)`
- `void glutCopyColormap(int window)`
- `void glutIgnoreKeyRepeat(int ignore)`
- `void glutSetKeyRepeat(int repeatMode)`
- `void glutForceJoystickFunc(void)`
- `int glutExtensionSupported(const char * extension)`

- `void glutReportErrors(void)`
- `void glutMainLoop(void)`

RINGOPENGL (OPENGL 1.1) FUNCTIONS REFERENCE

- GL_ZERO
- GL_FALSE
- GL_LOGIC_OP
- GL_NONE
- GL_TEXTURE_COMPONENTS
- GL_NO_ERROR
- GL_POINTS
- GL_CURRENT_BIT
- GL_TRUE
- GL_ONE
- GL_CLIENT_PIXEL_STORE_BIT
- GL_LINES
- GL_LINE_LOOP
- GL_POINT_BIT
- GL_CLIENT_VERTEX_ARRAY_BIT
- GL_LINE_STRIP
- GL_LINE_BIT
- GL_TRIANGLES
- GL_TRIANGLE_STRIP
- GL_TRIANGLE_FAN
- GL_QUADS
- GL_QUAD_STRIP
- GL_POLYGON_BIT
- GL_POLYGON
- GL_POLYGON_STIPPLE_BIT
- GL_PIXEL_MODE_BIT
- GL_LIGHTING_BIT

- GL_FOG_BIT
- GL_DEPTH_BUFFER_BIT
- GL_ACCUM
- GL_LOAD
- GL_RETURN
- GL_MULT
- GL_ADD
- GL_NEVER
- GL_ACCUM_BUFFER_BIT
- GL_LESS
- GL_EQUAL
- GL_LEQUAL
- GL_GREATER
- GL_NOTEQUAL
- GL_GEQUAL
- GL_ALWAYS
- GL_SRC_COLOR
- GL_ONE_MINUS_SRC_COLOR
- GL_SRC_ALPHA
- GL_ONE_MINUS_SRC_ALPHA
- GL_DST_ALPHA
- GL_ONE_MINUS_DST_ALPHA
- GL_DST_COLOR
- GL_ONE_MINUS_DST_COLOR
- GL_SRC_ALPHA_SATURATE
- GL_STENCIL_BUFFER_BIT
- GL_FRONT_LEFT
- GL_FRONT_RIGHT
- GL_BACK_LEFT
- GL_BACK_RIGHT
- GL_FRONT
- GL_BACK
- GL_LEFT
- GL_RIGHT
- GL_FRONT_AND_BACK
- GL_AUX0

- GL_AUX1
- GL_AUX2
- GL_AUX3
- GL_INVALID_ENUM
- GL_INVALID_VALUE
- GL_INVALID_OPERATION
- GL_STACK_OVERFLOW
- GL_STACK_UNDERFLOW
- GL_OUT_OF_MEMORY
- GL_2D
- GL_3D
- GL_3D_COLOR
- GL_3D_COLOR_TEXTURE
- GL_4D_COLOR_TEXTURE
- GL_PASS_THROUGH_TOKEN
- GL_POINT_TOKEN
- GL_LINE_TOKEN
- GL_POLYGON_TOKEN
- GL_BITMAP_TOKEN
- GL_DRAW_PIXEL_TOKEN
- GL_COPY_PIXEL_TOKEN
- GL_LINE_RESET_TOKEN
- GL_EXP
- GL_VIEWPORT_BIT
- GL_EXP2
- GL_CW
- GL_CCW
- GL_COEFF
- GL_ORDER
- GL_DOMAIN
- GL_CURRENT_COLOR
- GL_CURRENT_INDEX
- GL_CURRENT_NORMAL
- GL_CURRENT_TEXTURE_COORDS
- GL_CURRENT_RASTER_COLOR
- GL_CURRENT_RASTER_INDEX

- GL_CURRENT_RASTER_TEXTURE_COORDS
- GL_CURRENT_RASTER_POSITION
- GL_CURRENT_RASTER_POSITION_VALID
- GL_CURRENT_RASTER_DISTANCE
- GL_POINT_SMOOTH
- GL_POINT_SIZE
- GL_POINT_SIZE_RANGE
- GL_POINT_SIZE_GRANULARITY
- GL_LINE_SMOOTH
- GL_LINE_WIDTH
- GL_LINE_WIDTH_RANGE
- GL_LINE_WIDTH_GRANULARITY
- GL_LINE_STIPPLE
- GL_LINE_STIPPLE_PATTERN
- GL_LINE_STIPPLE_REPEAT
- GL_LIST_MODE
- GL_MAX_LIST_NESTING
- GL_LIST_BASE
- GL_LIST_INDEX
- GL_POLYGON_MODE
- GL_POLYGON_SMOOTH
- GL_POLYGON_STIPPLE
- GL_EDGE_FLAG
- GL_CULL_FACE
- GL_CULL_FACE_MODE
- GL_FRONT_FACE
- GL_LIGHTING
- GL_LIGHT_MODEL_LOCAL_VIEWER
- GL_LIGHT_MODEL_TWO_SIDE
- GL_LIGHT_MODEL_AMBIENT
- GL_SHADE_MODEL
- GL_COLOR_MATERIAL_FACE
- GL_COLOR_MATERIAL_PARAMETER
- GL_COLOR_MATERIAL
- GL_FOG
- GL_FOG_INDEX

- GL_FOG_DENSITY
- GL_FOG_START
- GL_FOG_END
- GL_FOG_MODE
- GL_FOG_COLOR
- GL_DEPTH_RANGE
- GL_DEPTH_TEST
- GL_DEPTH_WRITEMASK
- GL_DEPTH_CLEAR_VALUE
- GL_DEPTH_FUNC
- GL_ACCUM_CLEAR_VALUE
- GL_STENCIL_TEST
- GL_STENCIL_CLEAR_VALUE
- GL_STENCIL_FUNC
- GL_STENCIL_VALUE_MASK
- GL_STENCIL_FAIL
- GL_STENCIL_PASS_DEPTH_FAIL
- GL_STENCIL_PASS_DEPTH_PASS
- GL_STENCIL_REF
- GL_STENCIL_WRITEMASK
- GL_MATRIX_MODE
- GL_NORMALIZE
- GL_VIEWPORT
- GL_MODELVIEW_STACK_DEPTH
- GL_PROJECTION_STACK_DEPTH
- GL_TEXTURE_STACK_DEPTH
- GL_MODELVIEW_MATRIX
- GL_PROJECTION_MATRIX
- GL_TEXTURE_MATRIX
- GL_ATTRIB_STACK_DEPTH
- GL_CLIENT_ATTRIB_STACK_DEPTH
- GL_ALPHA_TEST
- GL_ALPHA_TEST_FUNC
- GL_ALPHA_TEST_REF
- GL_DITHER
- GL_BLEND_DST

- GL_BLEND_SRC
- GL_BLEND
- GL_LOGIC_OP_MODE
- GL_INDEX_LOGIC_OP
- GL_COLOR_LOGIC_OP
- GL_AUX_BUFFERS
- GL_DRAW_BUFFER
- GL_READ_BUFFER
- GL_SCISSOR_BOX
- GL_SCISSOR_TEST
- GL_INDEX_CLEAR_VALUE
- GL_INDEX_WRITEMASK
- GL_COLOR_CLEAR_VALUE
- GL_COLOR_WRITEMASK
- GL_INDEX_MODE
- GL_RGBA_MODE
- GL_DOUBLEBUFFER
- GL_STEREO
- GL_RENDER_MODE
- GL_PERSPECTIVE_CORRECTION_HINT
- GL_POINT_SMOOTH_HINT
- GL_LINE_SMOOTH_HINT
- GL_POLYGON_SMOOTH_HINT
- GL_FOG_HINT
- GL_TEXTURE_GEN_S
- GL_TEXTURE_GEN_T
- GL_TEXTURE_GEN_R
- GL_TEXTURE_GEN_Q
- GL_PIXEL_MAP_I_TO_I
- GL_PIXEL_MAP_S_TO_S
- GL_PIXEL_MAP_I_TO_R
- GL_PIXEL_MAP_I_TO_G
- GL_PIXEL_MAP_I_TO_B
- GL_PIXEL_MAP_I_TO_A
- GL_PIXEL_MAP_R_TO_R
- GL_PIXEL_MAP_G_TO_G

- GL_PIXEL_MAP_B_TO_B
- GL_PIXEL_MAP_A_TO_A
- GL_PIXEL_MAP_I_TO_I_SIZE
- GL_PIXEL_MAP_S_TO_S_SIZE
- GL_PIXEL_MAP_I_TO_R_SIZE
- GL_PIXEL_MAP_I_TO_G_SIZE
- GL_PIXEL_MAP_I_TO_B_SIZE
- GL_PIXEL_MAP_I_TO_A_SIZE
- GL_PIXEL_MAP_R_TO_R_SIZE
- GL_PIXEL_MAP_G_TO_G_SIZE
- GL_PIXEL_MAP_B_TO_B_SIZE
- GL_PIXEL_MAP_A_TO_A_SIZE
- GL_UNPACK_SWAP_BYTES
- GL_UNPACK_LSB_FIRST
- GL_UNPACK_ROW_LENGTH
- GL_UNPACK_SKIP_ROWS
- GL_UNPACK_SKIP_PIXELS
- GL_UNPACK_ALIGNMENT
- GL_PACK_SWAP_BYTES
- GL_PACK_LSB_FIRST
- GL_PACK_ROW_LENGTH
- GL_PACK_SKIP_ROWS
- GL_PACK_SKIP_PIXELS
- GL_PACK_ALIGNMENT
- GL_MAP_COLOR
- GL_MAP_STENCIL
- GL_INDEX_SHIFT
- GL_INDEX_OFFSET
- GL_RED_SCALE
- GL_RED_BIAS
- GL_ZOOM_X
- GL_ZOOM_Y
- GL_GREEN_SCALE
- GL_GREEN_BIAS
- GL_BLUE_SCALE
- GL_BLUE_BIAS

- GL_ALPHA_SCALE
- GL_ALPHA_BIAS
- GL_DEPTH_SCALE
- GL_DEPTH_BIAS
- GL_MAX_EVAL_ORDER
- GL_MAX_LIGHTS
- GL_MAX_CLIP_PLANES
- GL_MAX_TEXTURE_SIZE
- GL_MAX_PIXEL_MAP_TABLE
- GL_MAX_ATTRIB_STACK_DEPTH
- GL_MAX_MODELVIEW_STACK_DEPTH
- GL_MAX_NAME_STACK_DEPTH
- GL_MAX_PROJECTION_STACK_DEPTH
- GL_MAX_TEXTURE_STACK_DEPTH
- GL_MAX_VIEWPORT_DIMS
- GL_MAX_CLIENT_ATTRIB_STACK_DEPTH
- GL_SUBPIXEL_BITS
- GL_INDEX_BITS
- GL_RED_BITS
- GL_GREEN_BITS
- GL_BLUE_BITS
- GL_ALPHA_BITS
- GL_DEPTH_BITS
- GL_STENCIL_BITS
- GL_ACCUM_RED_BITS
- GL_ACCUM_GREEN_BITS
- GL_ACCUM_BLUE_BITS
- GL_ACCUM_ALPHA_BITS
- GL_NAME_STACK_DEPTH
- GL_AUTO_NORMAL
- GL_MAP1_COLOR_4
- GL_MAP1_INDEX
- GL_MAP1_NORMAL
- GL_MAP1_TEXTURE_COORD_1
- GL_MAP1_TEXTURE_COORD_2
- GL_MAP1_TEXTURE_COORD_3

- GL_MAP1_TEXTURE_COORD_4
- GL_MAP1_VERTEX_3
- GL_MAP1_VERTEX_4
- GL_MAP2_COLOR_4
- GL_MAP2_INDEX
- GL_MAP2_NORMAL
- GL_MAP2_TEXTURE_COORD_1
- GL_MAP2_TEXTURE_COORD_2
- GL_MAP2_TEXTURE_COORD_3
- GL_MAP2_TEXTURE_COORD_4
- GL_MAP2_VERTEX_3
- GL_MAP2_VERTEX_4
- GL_MAP1_GRID_DOMAIN
- GL_MAP1_GRID_SEGMENTS
- GL_MAP2_GRID_DOMAIN
- GL_MAP2_GRID_SEGMENTS
- GL_TEXTURE_1D
- GL_TEXTURE_2D
- GL_FEEDBACK_BUFFER_POINTER
- GL_FEEDBACK_BUFFER_SIZE
- GL_FEEDBACK_BUFFER_TYPE
- GL_SELECTION_BUFFER_POINTER
- GL_SELECTION_BUFFER_SIZE
- GL_TEXTURE_WIDTH
- GL_TRANSFORM_BIT
- GL_TEXTURE_HEIGHT
- GL_TEXTURE_INTERNAL_FORMAT
- GL_TEXTURE_BORDER_COLOR
- GL_TEXTURE_BORDER
- GL_DONT_CARE
- GL_FASTEST
- GL_NICEST
- GL_AMBIENT
- GL_DIFFUSE
- GL_SPECULAR
- GL_POSITION

- GL_SPOT_DIRECTION
- GL_SPOT_EXPONENT
- GL_SPOT_CUTOFF
- GL_CONSTANT_ATTENUATION
- GL_LINEAR_ATTENUATION
- GL_QUADRATIC_ATTENUATION
- GL_COMPILE
- GL_COMPILE_AND_EXECUTE
- GL_BYTE
- GL_UNSIGNED_BYTE
- GL_SHORT
- GL_UNSIGNED_SHORT
- GL_INT
- GL_UNSIGNED_INT
- GL_FLOAT
- GL_2_BYTES
- GL_3_BYTES
- GL_4_BYTES
- GL_DOUBLE
- GL_CLEAR
- GL_AND
- GL_AND_REVERSE
- GL_COPY
- GL_AND_INVERTED
- GL_NOOP
- GL_XOR
- GL_OR
- GL_NOR
- GL_EQUIV
- GL_INVERT
- GL_OR_REVERSE
- GL_COPY_INVERTED
- GL_OR_INVERTED
- GL_NAND
- GL_SET
- GL_EMISSION

- GL_SHININESS
- GL_AMBIENT_AND_DIFFUSE
- GL_COLOR_INDEXES
- GL_MODELVIEW
- GL_PROJECTION
- GL_TEXTURE
- GL_COLOR
- GL_DEPTH
- GL_STENCIL
- GL_COLOR_INDEX
- GL_STENCIL_INDEX
- GL_DEPTH_COMPONENT
- GL_RED
- GL_GREEN
- GL_BLUE
- GL_ALPHA
- GL_RGB
- GL_RGBA
- GL_LUMINANCE
- GL_LUMINANCE_ALPHA
- GL_BITMAP
- GL_POINT
- GL_LINE
- GL_FILL
- GL_RENDER
- GL_FEEDBACK
- GL_SELECT
- GL_FLAT
- GL_SMOOTH
- GL_KEEP
- GL_REPLACE
- GL_INCR
- GL_DECR
- GL_VENDOR
- GL_RENDERER
- GL_VERSION

- GL_EXTENSIONS
- GL_S
- GL_ENABLE_BIT
- GL_T
- GL_R
- GL_Q
- GL_MODULATE
- GL_DECAL
- GL_TEXTURE_ENV_MODE
- GL_TEXTURE_ENV_COLOR
- GL_TEXTURE_ENV
- GL_EYE_LINEAR
- GL_OBJECT_LINEAR
- GL_SPHERE_MAP
- GL_TEXTURE_GEN_MODE
- GL_OBJECT_PLANE
- GL_EYE_PLANE
- GL_NEAREST
- GL_LINEAR
- GL_NEAREST_MIPMAP_NEAREST
- GL_LINEAR_MIPMAP_NEAREST
- GL_NEAREST_MIPMAP_LINEAR
- GL_LINEAR_MIPMAP_LINEAR
- GL_TEXTURE_MAG_FILTER
- GL_TEXTURE_MIN_FILTER
- GL_TEXTURE_WRAP_S
- GL_TEXTURE_WRAP_T
- GL_CLAMP
- GL_REPEAT
- GL_POLYGON_OFFSET_UNITS
- GL_POLYGON_OFFSET_POINT
- GL_POLYGON_OFFSET_LINE
- GL_R3_G3_B2
- GL_V2F
- GL_V3F
- GL_C4UB_V2F

- GL_C4UB_V3F
- GL_C3F_V3F
- GL_N3F_V3F
- GL_C4F_N3F_V3F
- GL_T2F_V3F
- GL_T4F_V4F
- GL_T2F_C4UB_V3F
- GL_T2F_C3F_V3F
- GL_T2F_N3F_V3F
- GL_T2F_C4F_N3F_V3F
- GL_T4F_C4F_N3F_V4F
- GL_CLIP_PLANE0
- GL_CLIP_PLANE1
- GL_CLIP_PLANE2
- GL_CLIP_PLANE3
- GL_CLIP_PLANE4
- GL_CLIP_PLANE5
- GL_LIGHT0
- GL_COLOR_BUFFER_BIT
- GL_LIGHT1
- GL_LIGHT2
- GL_LIGHT3
- GL_LIGHT4
- GL_LIGHT5
- GL_LIGHT6
- GL_LIGHT7
- GL_HINT_BIT
- GL_POLYGON_OFFSET_FILL
- GL_POLYGON_OFFSET_FACTOR
- GL_ALPHA4
- GL_ALPHA8
- GL_ALPHA12
- GL_ALPHA16
- GL_LUMINANCE4
- GL_LUMINANCE8
- GL_LUMINANCE12

- GL_LUMINANCE16
- GL_LUMINANCE4_ALPHA4
- GL_LUMINANCE6_ALPHA2
- GL_LUMINANCE8_ALPHA8
- GL_LUMINANCE12_ALPHA4
- GL_LUMINANCE12_ALPHA12
- GL_LUMINANCE16_ALPHA16
- GL_INTENSITY
- GL_INTENSITY4
- GL_INTENSITY8
- GL_INTENSITY12
- GL_INTENSITY16
- GL_RGB4
- GL_RGB5
- GL_RGB8
- GL_RGB10
- GL_RGB12
- GL_RGB16
- GL_RGBA2
- GL_RGBA4
- GL_RGB5_A1
- GL_RGBA8
- GL_RGB10_A2
- GL_RGBA12
- GL_RGBA16
- GL_TEXTURE_RED_SIZE
- GL_TEXTURE_GREEN_SIZE
- GL_TEXTURE_BLUE_SIZE
- GL_TEXTURE_ALPHA_SIZE
- GL_TEXTURE_LUMINANCE_SIZE
- GL_TEXTURE_INTENSITY_SIZE
- GL_PROXY_TEXTURE_1D
- GL_PROXY_TEXTURE_2D
- GL_TEXTURE_PRIORITY
- GL_TEXTURE_RESIDENT
- GL_TEXTURE_BINDING_1D

- GL_TEXTURE_BINDING_2D
- GL_VERTEX_ARRAY
- GL_NORMAL_ARRAY
- GL_COLOR_ARRAY
- GL_INDEX_ARRAY
- GL_TEXTURE_COORD_ARRAY
- GL_EDGE_FLAG_ARRAY
- GL_VERTEX_ARRAY_SIZE
- GL_VERTEX_ARRAY_TYPE
- GL_VERTEX_ARRAY_STRIDE
- GL_NORMAL_ARRAY_TYPE
- GL_NORMAL_ARRAY_STRIDE
- GL_COLOR_ARRAY_SIZE
- GL_COLOR_ARRAY_TYPE
- GL_COLOR_ARRAY_STRIDE
- GL_INDEX_ARRAY_TYPE
- GL_INDEX_ARRAY_STRIDE
- GL_TEXTURE_COORD_ARRAY_SIZE
- GL_TEXTURE_COORD_ARRAY_TYPE
- GL_TEXTURE_COORD_ARRAY_STRIDE
- GL_EDGE_FLAG_ARRAY_STRIDE
- GL_VERTEX_ARRAY_POINTER
- GL_NORMAL_ARRAY_POINTER
- GL_COLOR_ARRAY_POINTER
- GL_INDEX_ARRAY_POINTER
- GL_TEXTURE_COORD_ARRAY_POINTER
- GL_EDGE_FLAG_ARRAY_POINTER
- GL_COLOR_INDEX1_EXT
- GL_COLOR_INDEX2_EXT
- GL_COLOR_INDEX4_EXT
- GL_COLOR_INDEX8_EXT
- GL_COLOR_INDEX12_EXT
- GL_COLOR_INDEX16_EXT
- GL_EVAL_BIT
- GL_LIST_BIT
- GL_TEXTURE_BIT

- GL_SCISSOR_BIT
- GL_ALL_ATTRIB_BITS
- GL_CLIENT_ALL_ATTRIB_BITS
- void glAccum(GLenum op, GLfloat value)
- void glAlphaFunc(GLenum func, GLclampf ref)
- GLboolean glAreTexturesResident(GLsizei n, const GLuint *textures, GLboolean *residences)
- void glArrayElement(GLint i)
- void glBegin(GLenum mode)
- void glBindTexture(GLenum target, GLuint texture)
- void glBitmap(GLsizei width, GLsizei height, GLfloat xorig, GLfloat yorig, GLfloat xmove, GLfloat ymove, const GLubyte *bitmap)
- void glBlendFunc(GLenum sfactor, GLenum dfactor)
- void glCallList(GLuint list)
- void glCallLists(GLsizei n, GLenum type, const void *lists)
- void glClear(GLbitfield mask)
- void glClearAccum(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glClearColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)
- void glClearDepth(GLclampd depth)
- void glClearIndex(GLfloat c)
- void glClearStencil(GLint s)
- void glClipPlane(GLenum plane, const GLdouble *equation)
- void glColor3b(GLbyte red, GLbyte green, GLbyte blue)
- void glColor3bv(const GLbyte *v)
- void glColor3d(GLdouble red, GLdouble green, GLdouble blue)
- void glColor3dv(const GLdouble *v)
- void glColor3f(GLfloat red, GLfloat green, GLfloat blue)
- void glColor3fv(const GLfloat *v)
- void glColor3i(GLint red, GLint green, GLint blue)
- void glColor3iv(const GLint *v)
- void glColor3s(GLshort red, GLshort green, GLshort blue)
- void glColor3sv(const GLshort *v)
- void glColor3ub(GLubyte red, GLubyte green, GLubyte blue)
- void glColor3ubv(const GLubyte *v)
- void glColor3ui(GLuint red, GLuint green, GLuint blue)
- void glColor3uiv(const GLuint *v)
- void glColor3us(GLushort red, GLushort green, GLushort blue)

- `void glColor3usv(const GLushort *v)`
- `void glColor4b(GLbyte red, GLbyte green, GLbyte blue, GLbyte alpha)`
- `void glColor4bv(const GLbyte *v)`
- `void glColor4d(GLdouble red, GLdouble green, GLdouble blue, GLdouble alpha)`
- `void glColor4dv(const GLdouble *v)`
- `void glColor4f(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)`
- `void glColor4fv(const GLfloat *v)`
- `void glColor4i(GLint red, GLint green, GLint blue, GLint alpha)`
- `void glColor4iv(const GLint *v)`
- `void glColor4s(GLshort red, GLshort green, GLshort blue, GLshort alpha)`
- `void glColor4sv(const GLshort *v)`
- `void glColor4ub(GLubyte red, GLubyte green, GLubyte blue, GLubyte alpha)`
- `void glColor4ubv(const GLubyte *v)`
- `void glColor4ui(GLuint red, GLuint green, GLuint blue, GLuint alpha)`
- `void glColor4uiv(const GLuint *v)`
- `void glColor4us(GLushort red, GLushort green, GLushort blue, GLushort alpha)`
- `void glColor4usv(const GLushort *v)`
- `void glColorMask(GLboolean red, GLboolean green, GLboolean blue, GLboolean alpha)`
- `void glColorMaterial(GLenum face, GLenum mode)`
- `void glColorPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)`
- `void glCopyPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum type)`
- `void glCopyTexImage1D(GLenum target, GLint level, GLenum internalFormat, GLint x, GLint y, GLsizei width, GLint border)`
- `void glCopyTexImage2D(GLenum target, GLint level, GLenum internalFormat, GLint x, GLint y, GLsizei width, GLsizei height, GLint border)`
- `void glCopyTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLint x, GLint y, GLsizei width)`
- `void glCopyTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint x, GLint y, GLsizei width, GLsizei height)`
- `void glCullFace(GLenum mode)`
- `void glDeleteLists(GLuint list, GLsizei range)`
- `void glDeleteTextures(GLsizei n, const GLuint *textures)`
- `void glDepthFunc(GLenum func)`
- `void glDepthMask(GLboolean flag)`
- `void glDepthRange(GLclampd zNear, GLclampd zFar)`
- `void glDisable(GLenum cap)`
- `void glDisableClientState(GLenum array)`
- `void glDrawArrays(GLenum mode, GLint first, GLsizei count)`

- void glDrawBuffer(GLenum mode)
- void glDrawElements(GLenum mode, GLsizei count, GLenum type, const void *indices)
- void glDrawPixels(GLsizei width, GLsizei height, GLenum format, GLenum type, const void *pixels)
- void glEdgeFlag(GLboolean flag)
- void glEdgeFlagPointer(GLsizei stride, const void *pointer)
- void glEdgeFlagv(const GLboolean *flag)
- void glEnable(GLenum cap)
- void glEnableClientState(GLenum array)
- void glEnd(void)
- void glEndList(void)
- void glEvalCoord1d(GLdouble u)
- void glEvalCoord1dv(const GLdouble *u)
- void glEvalCoord1f(GLfloat u)
- void glEvalCoord1fv(const GLfloat *u)
- void glEvalCoord2d(GLdouble u, GLdouble v)
- void glEvalCoord2dv(const GLdouble *u)
- void glEvalCoord2f(GLfloat u, GLfloat v)
- void glEvalCoord2fv(const GLfloat *u)
- void glEvalMesh1(GLenum mode, GLint i1, GLint i2)
- void glEvalMesh2(GLenum mode, GLint i1, GLint i2, GLint j1, GLint j2)
- void glEvalPoint1(GLint i)
- void glEvalPoint2(GLint i, GLint j)
- void glFeedbackBuffer(GLsizei size, GLenum type, GLfloat *buffer)
- void glFinish(void)
- void glFlush(void)
- void glFogf(GLenum pname, GLfloat param)
- void glFogfv(GLenum pname, const GLfloat *params)
- void glFogi(GLenum pname, GLint param)
- void glFogiv(GLenum pname, const GLint *params)
- void glFrontFace(GLenum mode)
- void glFrustum(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble zNear, GLdouble zFar)
- GLuint glGenLists(GLsizei range)
- void glGenTextures(GLsizei n, GLuint *textures)
- void glGetBooleanv(GLenum pname, GLboolean *params)
- void glGetClipPlane(GLenum plane, GLdouble *equation)

- void glGetDoublev(GLenum pname, GLdouble *params)
- GLenum glGetError(void)
- void glGetFloatv(GLenum pname, GLfloat *params)
- void glGetIntegerv(GLenum pname, GLint *params)
- void glGetLightfv(GLenum light, GLenum pname, GLfloat *params)
- void glGetLightiv(GLenum light, GLenum pname, GLint *params)
- void glGetMapdv(GLenum target, GLenum query, GLdouble *v)
- void glGetMapfv(GLenum target, GLenum query, GLfloat *v)
- void glGetMapiv(GLenum target, GLenum query, GLint *v)
- void glGetMaterialfv(GLenum face, GLenum pname, GLfloat *params)
- void glGetMaterialiv(GLenum face, GLenum pname, GLint *params)
- void glGetPixelMapfv(GLenum map, GLfloat *values)
- void glGetPixelMapuiv(GLenum map, GLuint *values)
- void glGetPixelMapusv(GLenum map, GLushort *values)
- void glGetPointerv(GLenum pname, void* *params)
- void glGetPolygonStipple(GLubyte *mask)
- GLubyte * getString(GLenum name)
- void glGetTexEnvfv(GLenum target, GLenum pname, GLfloat *params)
- void glGetTexEnviv(GLenum target, GLenum pname, GLint *params)
- void glGetTexGendv(GLenum coord, GLenum pname, GLdouble *params)
- void glGetTexGenfv(GLenum coord, GLenum pname, GLfloat *params)
- void glGetTexGeniv(GLenum coord, GLenum pname, GLint *params)
- void glGetTexImage(GLenum target, GLint level, GLenum format, GLenum type, void *pixels)
- void glGetTexLevelParameterfv(GLenum target, GLint level, GLenum pname, GLfloat *params)
- void glGetTexLevelParameteriv(GLenum target, GLint level, GLenum pname, GLint *params)
- void glGetTexParameterfv(GLenum target, GLenum pname, GLfloat *params)
- void glGetTexParameteriv(GLenum target, GLenum pname, GLint *params)
- void glHint(GLenum target, GLenum mode)
- void glIndexMask(GLuint mask)
- void glIndexPointer(GLenum type, GLsizei stride, const void *pointer)
- void glIndexd(GLdouble c)
- void glIndexdv(const GLdouble *c)
- void glIndexf(GLfloat c)
- void glIndexfv(const GLfloat *c)
- void glIndexi(GLint c)
- void glIndexiv(const GLint *c)

- void glIndexs(GLshort c)
- void glIndexsv(const GLshort *c)
- void glIndexub(GLubyte c)
- void glIndexubv(const GLubyte *c)
- void glInitNames(void)
- void glInterleavedArrays(GLenum format, GLsizei stride, const void *pointer)
- GLboolean glIsEnabled(GLenum cap)
- GLboolean glIsList(GLuint list)
- GLboolean glIsTexture(GLuint texture)
- void glLightModelf(GLenum pname, GLfloat param)
- void glLightModelfv(GLenum pname, const GLfloat *params)
- void glLightModeli(GLenum pname, GLint param)
- void glLightModeliv(GLenum pname, const GLint *params)
- void glLightf(GLenum light, GLenum pname, GLfloat param)
- void glLightfv(GLenum light, GLenum pname, const GLfloat *params)
- void glLighti(GLenum light, GLenum pname, GLint param)
- void glLightiv(GLenum light, GLenum pname, const GLint *params)
- void glLineStipple(GLint factor, GLushort pattern)
- void glLineWidth(GLfloat width)
- void glListBase(GLuint base)
- void glLoadIdentity(void)
- void glLoadMatrixd(const GLdouble *m)
- void glLoadMatrixf(const GLfloat *m)
- void glLoadName(GLuint name)
- void glLogicOp(GLenum opcode)
- void glMap1d(GLenum target, GLdouble u1, GLdouble u2, GLint stride, GLint order, const GLdouble *points)
- void glMap1f(GLenum target, GLfloat u1, GLfloat u2, GLint stride, GLint order, const GLfloat *points)
- void glMap2d(GLenum target, GLdouble u1, GLdouble u2, GLint ustride, GLint uorder, GLdouble v1, GLdouble v2, GLint vstride, GLint vorder, const GLdouble *points)
- void glMap2f(GLenum target, GLfloat u1, GLfloat u2, GLint ustride, GLint uorder, GLfloat v1, GLfloat v2, GLint vstride, GLint vorder, const GLfloat *points)
- void glMapGrid1d(GLint un, GLdouble u1, GLdouble u2)
- void glMapGrid1f(GLint un, GLfloat u1, GLfloat u2)
- void glMapGrid2d(GLint un, GLdouble u1, GLdouble u2, GLint vn, GLdouble v1, GLdouble v2)
- void glMapGrid2f(GLint un, GLfloat u1, GLfloat u2, GLint vn, GLfloat v1, GLfloat v2)
- void glMaterialf(GLenum face, GLenum pname, GLfloat param)
- void glMaterialfv(GLenum face, GLenum pname, const GLfloat *params)

- void glMateriali(GLenum face, GLenum pname, GLint param)
- void glMaterialiv(GLenum face, GLenum pname, const GLint *params)
- void glMatrixMode(GLenum mode)
- void glMultMatrixd(const GLdouble *m)
- void glMultMatrixf(const GLfloat *m)
- void glNewList(GLuint list, GLenum mode)
- void glNormal3b(GLbyte nx, GLbyte ny, GLbyte nz)
- void glNormal3bv(const GLbyte *v)
- void glNormal3d(GLdouble nx, GLdouble ny, GLdouble nz)
- void glNormal3dv(const GLdouble *v)
- void glNormal3f(GLfloat nx, GLfloat ny, GLfloat nz)
- void glNormal3fv(const GLfloat *v)
- void glNormal3i(GLint nx, GLint ny, GLint nz)
- void glNormal3iv(const GLint *v)
- void glNormal3s(GLshort nx, GLshort ny, GLshort nz)
- void glNormal3sv(const GLshort *v)
- void glNormalPointer(GLenum type, GLsizei stride, const void *pointer)
- void glOrtho(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble zNear, GLdouble zFar)
- void glPassThrough(GLfloat token)
- void glPixelMapfv(GLenum map, GLsizei mapsize, const GLfloat *values)
- void glPixelMapuiv(GLenum map, GLsizei mapsize, const GLuint *values)
- void glPixelMapusv(GLenum map, GLsizei mapsize, const GLushort *values)
- void glPixelStoref(GLenum pname, GLfloat param)
- void glPixelStorei(GLenum pname, GLint param)
- void glPixelTransferf(GLenum pname, GLfloat param)
- void glPixelTransferi(GLenum pname, GLint param)
- void glPixelZoom(GLfloat xfactor, GLfloat yfactor)
- void glPointSize(GLfloat size)
- void glPolygonMode(GLenum face, GLenum mode)
- void glPolygonOffset(GLfloat factor, GLfloat units)
- void glPolygonStipple(const GLubyte *mask)
- void glPopAttrib(void)
- void glPopClientAttrib(void)
- void glPopMatrix(void)
- void glPopName(void)

- void glPrioritizeTextures(GLsizei n, const GLuint *textures, const GLclampf *priorities)
- void glPushAttrib(GLbitfield mask)
- void glPushClientAttrib(GLbitfield mask)
- void glPushMatrix(void)
- void glPushName(GLuint name)
- void glRasterPos2d(GLdouble x, GLdouble y)
- void glRasterPos2dv(const GLdouble *v)
- void glRasterPos2f(GLfloat x, GLfloat y)
- void glRasterPos2fv(const GLfloat *v)
- void glRasterPos2i(GLint x, GLint y)
- void glRasterPos2iv(const GLint *v)
- void glRasterPos2s(GLshort x, GLshort y)
- void glRasterPos2sv(const GLshort *v)
- void glRasterPos3d(GLdouble x, GLdouble y, GLdouble z)
- void glRasterPos3dv(const GLdouble *v)
- void glRasterPos3f(GLfloat x, GLfloat y, GLfloat z)
- void glRasterPos3fv(const GLfloat *v)
- void glRasterPos3i(GLint x, GLint y, GLint z)
- void glRasterPos3iv(const GLint *v)
- void glRasterPos3s(GLshort x, GLshort y, GLshort z)
- void glRasterPos3sv(const GLshort *v)
- void glRasterPos4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)
- void glRasterPos4dv(const GLdouble *v)
- void glRasterPos4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- void glRasterPos4fv(const GLfloat *v)
- void glRasterPos4i(GLint x, GLint y, GLint z, GLint w)
- void glRasterPos4iv(const GLint *v)
- void glRasterPos4s(GLshort x, GLshort y, GLshort z, GLshort w)
- void glRasterPos4sv(const GLshort *v)
- void glReadBuffer(GLenum mode)
- void glReadPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum format, GLenum type, void *pixels)
- void glRectd(GLdouble x1, GLdouble y1, GLdouble x2, GLdouble y2)
- void glRectdv(const GLdouble *v1, const GLdouble *v2)
- void glRectf(GLfloat x1, GLfloat y1, GLfloat x2, GLfloat y2)
- void glRectfv(const GLfloat *v1, const GLfloat *v2)

- void glRecti(GLint x1, GLint y1, GLint x2, GLint y2)
- void glRectiv(const GLint *v1, const GLint *v2)
- void glRects(GLshort x1, GLshort y1, GLshort x2, GLshort y2)
- void glRectsv(const GLshort *v1, const GLshort *v2)
- GLint glRenderMode(GLenum mode)
- void glRotated(GLdouble angle, GLdouble x, GLdouble y, GLdouble z)
- void glRotatef(GLfloat angle, GLfloat x, GLfloat y, GLfloat z)
- void glScaled(GLdouble x, GLdouble y, GLdouble z)
- void glScalef(GLfloat x, GLfloat y, GLfloat z)
- void glScissor(GLint x, GLint y, GLsizei width, GLsizei height)
- void glSelectBuffer(GLsizei size, GLuint *buffer)
- void glShadeModel(GLenum mode)
- void glStencilFunc(GLenum func, GLint ref, GLuint mask)
- void glStencilMask(GLuint mask)
- void glStencilOp(GLenum fail, GLenum zfail, GLenum zpass)
- void glTexCoord1d(GLdouble s)
- void glTexCoord1dv(const GLdouble *v)
- void glTexCoord1f(GLfloat s)
- void glTexCoord1fv(const GLfloat *v)
- void glTexCoord1i(GLint s)
- void glTexCoord1iv(const GLint *v)
- void glTexCoord1s(GLshort s)
- void glTexCoord1sv(const GLshort *v)
- void glTexCoord2d(GLdouble s, GLdouble t)
- void glTexCoord2dv(const GLdouble *v)
- void glTexCoord2f(GLfloat s, GLfloat t)
- void glTexCoord2fv(const GLfloat *v)
- void glTexCoord2i(GLint s, GLint t)
- void glTexCoord2iv(const GLint *v)
- void glTexCoord2s(GLshort s, GLshort t)
- void glTexCoord2sv(const GLshort *v)
- void glTexCoord3d(GLdouble s, GLdouble t, GLdouble r)
- void glTexCoord3dv(const GLdouble *v)
- void glTexCoord3f(GLfloat s, GLfloat t, GLfloat r)
- void glTexCoord3fv(const GLfloat *v)
- void glTexCoord3i(GLint s, GLint t, GLint r)

- void glTexCoord3iv(const GLint *v)
- void glTexCoord3s(GLshort s, GLshort t, GLshort r)
- void glTexCoord3sv(const GLshort *v)
- void glTexCoord4d(GLdouble s, GLdouble t, GLdouble r, GLdouble q)
- void glTexCoord4dv(const GLdouble *v)
- void glTexCoord4f(GLfloat s, GLfloat t, GLfloat r, GLfloat q)
- void glTexCoord4fv(const GLfloat *v)
- void glTexCoord4i(GLint s, GLint t, GLint r, GLint q)
- void glTexCoord4iv(const GLint *v)
- void glTexCoord4s(GLshort s, GLshort t, GLshort r, GLshort q)
- void glTexCoord4sv(const GLshort *v)
- void glTexCoordPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)
- void glTexEnvf(GLenum target, GLenum pname, GLfloat param)
- void glTexEnvfv(GLenum target, GLenum pname, const GLfloat *params)
- void glTexEnvi(GLenum target, GLenum pname, GLint param)
- void glTexEnviv(GLenum target, GLenum pname, const GLint *params)
- void glTexGend(GLenum coord, GLenum pname, GLdouble param)
- void glTexGendv(GLenum coord, GLenum pname, const GLdouble *params)
- void glTexGenf(GLenum coord, GLenum pname, GLfloat param)
- void glTexGenfv(GLenum coord, GLenum pname, const GLfloat *params)
- void glTexGeni(GLenum coord, GLenum pname, GLint param)
- void glTexGeniv(GLenum coord, GLenum pname, const GLint *params)
- void glTexImage1D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLint border, GLenum format, GLenum type, const void *pixels)
- void glTexImage2D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, const void *pixels)
- void glTexParameterf(GLenum target, GLenum pname, GLfloat param)
- void glTexParameterfv(GLenum target, GLenum pname, const GLfloat *params)
- void glTexParameteri(GLenum target, GLenum pname, GLint param)
- void glTexParameteriv(GLenum target, GLenum pname, const GLint *params)
- void glTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLenum type, const void *pixels)
- void glTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLenum type, const void *pixels)
- void glTranslated(GLdouble x, GLdouble y, GLdouble z)
- void glTranslatef(GLfloat x, GLfloat y, GLfloat z)
- void glVertex2d(GLdouble x, GLdouble y)

- `void glVertex2dv(const GLdouble *v)`
- `void glVertex2f(GLfloat x, GLfloat y)`
- `void glVertex2fv(const GLfloat *v)`
- `void glVertex2i(GLint x, GLint y)`
- `void glVertex2iv(const GLint *v)`
- `void glVertex2s(GLshort x, GLshort y)`
- `void glVertex2sv(const GLshort *v)`
- `void glVertex3d(GLdouble x, GLdouble y, GLdouble z)`
- `void glVertex3dv(const GLdouble *v)`
- `void glVertex3f(GLfloat x, GLfloat y, GLfloat z)`
- `void glVertex3fv(const GLfloat *v)`
- `void glVertex3i(GLint x, GLint y, GLint z)`
- `void glVertex3iv(const GLint *v)`
- `void glVertex3s(GLshort x, GLshort y, GLshort z)`
- `void glVertex3sv(const GLshort *v)`
- `void glVertex4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)`
- `void glVertex4dv(const GLdouble *v)`
- `void glVertex4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)`
- `void glVertex4fv(const GLfloat *v)`
- `void glVertex4i(GLint x, GLint y, GLint z, GLint w)`
- `void glVertex4iv(const GLint *v)`
- `void glVertex4s(GLshort x, GLshort y, GLshort z, GLshort w)`
- `void glVertex4sv(const GLshort *v)`
- `void glVertexPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)`
- `void glViewport(GLint x, GLint y, GLsizei width, GLsizei height)`

RINGOPENGL (OPENGL 1.2) FUNCTIONS REFERENCE

- GL_ZERO
- GL_FALSE
- GL_LOGIC_OP
- GL_NONE
- GL_TEXTURE_COMPONENTS
- GL_NO_ERROR
- GL_POINTS
- GL_CURRENT_BIT
- GL_TRUE
- GL_ONE
- GL_CLIENT_PIXEL_STORE_BIT
- GL_LINES
- GL_LINE_LOOP
- GL_POINT_BIT
- GL_CLIENT_VERTEX_ARRAY_BIT
- GL_LINE_STRIP
- GL_LINE_BIT
- GL_TRIANGLES
- GL_TRIANGLE_STRIP
- GL_TRIANGLE_FAN
- GL_QUADS
- GL_QUAD_STRIP
- GL_POLYGON_BIT
- GL_POLYGON
- GL_POLYGON_STIPPLE_BIT
- GL_PIXEL_MODE_BIT
- GL_LIGHTING_BIT

- GL_FOG_BIT
- GL_DEPTH_BUFFER_BIT
- GL_ACCUM
- GL_LOAD
- GL_RETURN
- GL_MULT
- GL_ADD
- GL_NEVER
- GL_ACCUM_BUFFER_BIT
- GL_LESS
- GL_EQUAL
- GL_LEQUAL
- GL_GREATER
- GL_NOTEQUAL
- GL_GEQUAL
- GL_ALWAYS
- GL_SRC_COLOR
- GL_ONE_MINUS_SRC_COLOR
- GL_SRC_ALPHA
- GL_ONE_MINUS_SRC_ALPHA
- GL_DST_ALPHA
- GL_ONE_MINUS_DST_ALPHA
- GL_DST_COLOR
- GL_ONE_MINUS_DST_COLOR
- GL_SRC_ALPHA_SATURATE
- GL_STENCIL_BUFFER_BIT
- GL_FRONT_LEFT
- GL_FRONT_RIGHT
- GL_BACK_LEFT
- GL_BACK_RIGHT
- GL_FRONT
- GL_BACK
- GL_LEFT
- GL_RIGHT
- GL_FRONT_AND_BACK
- GL_AUX0

- GL_AUX1
- GL_AUX2
- GL_AUX3
- GL_INVALID_ENUM
- GL_INVALID_VALUE
- GL_INVALID_OPERATION
- GL_STACK_OVERFLOW
- GL_STACK_UNDERFLOW
- GL_OUT_OF_MEMORY
- GL_2D
- GL_3D
- GL_3D_COLOR
- GL_3D_COLOR_TEXTURE
- GL_4D_COLOR_TEXTURE
- GL_PASS_THROUGH_TOKEN
- GL_POINT_TOKEN
- GL_LINE_TOKEN
- GL_POLYGON_TOKEN
- GL_BITMAP_TOKEN
- GL_DRAW_PIXEL_TOKEN
- GL_COPY_PIXEL_TOKEN
- GL_LINE_RESET_TOKEN
- GL_EXP
- GL_VIEWPORT_BIT
- GL_EXP2
- GL_CW
- GL_CCW
- GL_COEFF
- GL_ORDER
- GL_DOMAIN
- GL_CURRENT_COLOR
- GL_CURRENT_INDEX
- GL_CURRENT_NORMAL
- GL_CURRENT_TEXTURE_COORDS
- GL_CURRENT_RASTER_COLOR
- GL_CURRENT_RASTER_INDEX

- GL_CURRENT_RASTER_TEXTURE_COORDS
- GL_CURRENT_RASTER_POSITION
- GL_CURRENT_RASTER_POSITION_VALID
- GL_CURRENT_RASTER_DISTANCE
- GL_POINT_SMOOTH
- GL_POINT_SIZE
- GL_POINT_SIZE_RANGE
- GL_POINT_SIZE_GRANULARITY
- GL_LINE_SMOOTH
- GL_LINE_WIDTH
- GL_LINE_WIDTH_RANGE
- GL_LINE_WIDTH_GRANULARITY
- GL_LINE_STIPPLE
- GL_LINE_STIPPLE_PATTERN
- GL_LINE_STIPPLE_REPEAT
- GL_LIST_MODE
- GL_MAX_LIST_NESTING
- GL_LIST_BASE
- GL_LIST_INDEX
- GL_POLYGON_MODE
- GL_POLYGON_SMOOTH
- GL_POLYGON_STIPPLE
- GL_EDGE_FLAG
- GL_CULL_FACE
- GL_CULL_FACE_MODE
- GL_FRONT_FACE
- GL_LIGHTING
- GL_LIGHT_MODEL_LOCAL_VIEWER
- GL_LIGHT_MODEL_TWO_SIDE
- GL_LIGHT_MODEL_AMBIENT
- GL_SHADE_MODEL
- GL_COLOR_MATERIAL_FACE
- GL_COLOR_MATERIAL_PARAMETER
- GL_COLOR_MATERIAL
- GL_FOG
- GL_FOG_INDEX

- GL_FOG_DENSITY
- GL_FOG_START
- GL_FOG_END
- GL_FOG_MODE
- GL_FOG_COLOR
- GL_DEPTH_RANGE
- GL_DEPTH_TEST
- GL_DEPTH_WRITEMASK
- GL_DEPTH_CLEAR_VALUE
- GL_DEPTH_FUNC
- GL_ACCUM_CLEAR_VALUE
- GL_STENCIL_TEST
- GL_STENCIL_CLEAR_VALUE
- GL_STENCIL_FUNC
- GL_STENCIL_VALUE_MASK
- GL_STENCIL_FAIL
- GL_STENCIL_PASS_DEPTH_FAIL
- GL_STENCIL_PASS_DEPTH_PASS
- GL_STENCIL_REF
- GL_STENCIL_WRITEMASK
- GL_MATRIX_MODE
- GL_NORMALIZE
- GL_VIEWPORT
- GL_MODELVIEW_STACK_DEPTH
- GL_PROJECTION_STACK_DEPTH
- GL_TEXTURE_STACK_DEPTH
- GL_MODELVIEW_MATRIX
- GL_PROJECTION_MATRIX
- GL_TEXTURE_MATRIX
- GL_ATTRIB_STACK_DEPTH
- GL_CLIENT_ATTRIB_STACK_DEPTH
- GL_ALPHA_TEST
- GL_ALPHA_TEST_FUNC
- GL_ALPHA_TEST_REF
- GL_DITHER
- GL_BLEND_DST

- GL_BLEND_SRC
- GL_BLEND
- GL_LOGIC_OP_MODE
- GL_INDEX_LOGIC_OP
- GL_COLOR_LOGIC_OP
- GL_AUX_BUFFERS
- GL_DRAW_BUFFER
- GL_READ_BUFFER
- GL_SCISSOR_BOX
- GL_SCISSOR_TEST
- GL_INDEX_CLEAR_VALUE
- GL_INDEX_WRITEMASK
- GL_COLOR_CLEAR_VALUE
- GL_COLOR_WRITEMASK
- GL_INDEX_MODE
- GL_RGBA_MODE
- GL_DOUBLEBUFFER
- GL_STEREO
- GL_RENDER_MODE
- GL_PERSPECTIVE_CORRECTION_HINT
- GL_POINT_SMOOTH_HINT
- GL_LINE_SMOOTH_HINT
- GL_POLYGON_SMOOTH_HINT
- GL_FOG_HINT
- GL_TEXTURE_GEN_S
- GL_TEXTURE_GEN_T
- GL_TEXTURE_GEN_R
- GL_TEXTURE_GEN_Q
- GL_PIXEL_MAP_I_TO_I
- GL_PIXEL_MAP_S_TO_S
- GL_PIXEL_MAP_I_TO_R
- GL_PIXEL_MAP_I_TO_G
- GL_PIXEL_MAP_I_TO_B
- GL_PIXEL_MAP_I_TO_A
- GL_PIXEL_MAP_R_TO_R
- GL_PIXEL_MAP_G_TO_G

- GL_PIXEL_MAP_B_TO_B
- GL_PIXEL_MAP_A_TO_A
- GL_PIXEL_MAP_I_TO_I_SIZE
- GL_PIXEL_MAP_S_TO_S_SIZE
- GL_PIXEL_MAP_I_TO_R_SIZE
- GL_PIXEL_MAP_I_TO_G_SIZE
- GL_PIXEL_MAP_I_TO_B_SIZE
- GL_PIXEL_MAP_I_TO_A_SIZE
- GL_PIXEL_MAP_R_TO_R_SIZE
- GL_PIXEL_MAP_G_TO_G_SIZE
- GL_PIXEL_MAP_B_TO_B_SIZE
- GL_PIXEL_MAP_A_TO_A_SIZE
- GL_UNPACK_SWAP_BYTES
- GL_UNPACK_LSB_FIRST
- GL_UNPACK_ROW_LENGTH
- GL_UNPACK_SKIP_ROWS
- GL_UNPACK_SKIP_PIXELS
- GL_UNPACK_ALIGNMENT
- GL_PACK_SWAP_BYTES
- GL_PACK_LSB_FIRST
- GL_PACK_ROW_LENGTH
- GL_PACK_SKIP_ROWS
- GL_PACK_SKIP_PIXELS
- GL_PACK_ALIGNMENT
- GL_MAP_COLOR
- GL_MAP_STENCIL
- GL_INDEX_SHIFT
- GL_INDEX_OFFSET
- GL_RED_SCALE
- GL_RED_BIAS
- GL_ZOOM_X
- GL_ZOOM_Y
- GL_GREEN_SCALE
- GL_GREEN_BIAS
- GL_BLUE_SCALE
- GL_BLUE_BIAS

- GL_ALPHA_SCALE
- GL_ALPHA_BIAS
- GL_DEPTH_SCALE
- GL_DEPTH_BIAS
- GL_MAX_EVAL_ORDER
- GL_MAX_LIGHTS
- GL_MAX_CLIP_PLANES
- GL_MAX_TEXTURE_SIZE
- GL_MAX_PIXEL_MAP_TABLE
- GL_MAX_ATTRIB_STACK_DEPTH
- GL_MAX_MODELVIEW_STACK_DEPTH
- GL_MAX_NAME_STACK_DEPTH
- GL_MAX_PROJECTION_STACK_DEPTH
- GL_MAX_TEXTURE_STACK_DEPTH
- GL_MAX_VIEWPORT_DIMS
- GL_MAX_CLIENT_ATTRIB_STACK_DEPTH
- GL_SUBPIXEL_BITS
- GL_INDEX_BITS
- GL_RED_BITS
- GL_GREEN_BITS
- GL_BLUE_BITS
- GL_ALPHA_BITS
- GL_DEPTH_BITS
- GL_STENCIL_BITS
- GL_ACCUM_RED_BITS
- GL_ACCUM_GREEN_BITS
- GL_ACCUM_BLUE_BITS
- GL_ACCUM_ALPHA_BITS
- GL_NAME_STACK_DEPTH
- GL_AUTO_NORMAL
- GL_MAP1_COLOR_4
- GL_MAP1_INDEX
- GL_MAP1_NORMAL
- GL_MAP1_TEXTURE_COORD_1
- GL_MAP1_TEXTURE_COORD_2
- GL_MAP1_TEXTURE_COORD_3

- GL_MAP1_TEXTURE_COORD_4
- GL_MAP1_VERTEX_3
- GL_MAP1_VERTEX_4
- GL_MAP2_COLOR_4
- GL_MAP2_INDEX
- GL_MAP2_NORMAL
- GL_MAP2_TEXTURE_COORD_1
- GL_MAP2_TEXTURE_COORD_2
- GL_MAP2_TEXTURE_COORD_3
- GL_MAP2_TEXTURE_COORD_4
- GL_MAP2_VERTEX_3
- GL_MAP2_VERTEX_4
- GL_MAP1_GRID_DOMAIN
- GL_MAP1_GRID_SEGMENTS
- GL_MAP2_GRID_DOMAIN
- GL_MAP2_GRID_SEGMENTS
- GL_TEXTURE_1D
- GL_TEXTURE_2D
- GL_FEEDBACK_BUFFER_POINTER
- GL_FEEDBACK_BUFFER_SIZE
- GL_FEEDBACK_BUFFER_TYPE
- GL_SELECTION_BUFFER_POINTER
- GL_SELECTION_BUFFER_SIZE
- GL_TEXTURE_WIDTH
- GL_TRANSFORM_BIT
- GL_TEXTURE_HEIGHT
- GL_TEXTURE_INTERNAL_FORMAT
- GL_TEXTURE_BORDER_COLOR
- GL_TEXTURE_BORDER
- GL_DONT_CARE
- GL_FASTEST
- GL_NICEST
- GL_AMBIENT
- GL_DIFFUSE
- GL_SPECULAR
- GL_POSITION

- GL_SPOT_DIRECTION
- GL_SPOT_EXPONENT
- GL_SPOT_CUTOFF
- GL_CONSTANT_ATTENUATION
- GL_LINEAR_ATTENUATION
- GL_QUADRATIC_ATTENUATION
- GL_COMPILE
- GL_COMPILE_AND_EXECUTE
- GL_BYTE
- GL_UNSIGNED_BYTE
- GL_SHORT
- GL_UNSIGNED_SHORT
- GL_INT
- GL_UNSIGNED_INT
- GL_FLOAT
- GL_2_BYTES
- GL_3_BYTES
- GL_4_BYTES
- GL_DOUBLE
- GL_CLEAR
- GL_AND
- GL_AND_REVERSE
- GL_COPY
- GL_AND_INVERTED
- GL_NOOP
- GL_XOR
- GL_OR
- GL_NOR
- GL_EQUIV
- GL_INVERT
- GL_OR_REVERSE
- GL_COPY_INVERTED
- GL_OR_INVERTED
- GL_NAND
- GL_SET
- GL_EMISSION

- GL_SHININESS
- GL_AMBIENT_AND_DIFFUSE
- GL_COLOR_INDEXES
- GL_MODELVIEW
- GL_PROJECTION
- GL_TEXTURE
- GL_COLOR
- GL_DEPTH
- GL_STENCIL
- GL_COLOR_INDEX
- GL_STENCIL_INDEX
- GL_DEPTH_COMPONENT
- GL_RED
- GL_GREEN
- GL_BLUE
- GL_ALPHA
- GL_RGB
- GL_RGBA
- GL_LUMINANCE
- GL_LUMINANCE_ALPHA
- GL_BITMAP
- GL_POINT
- GL_LINE
- GL_FILL
- GL_RENDER
- GL_FEEDBACK
- GL_SELECT
- GL_FLAT
- GL_SMOOTH
- GL_KEEP
- GL_REPLACE
- GL_INCR
- GL_DECR
- GL_VENDOR
- GL_RENDERER
- GL_VERSION

- GL_EXTENSIONS
- GL_S
- GL_ENABLE_BIT
- GL_T
- GL_R
- GL_Q
- GL_MODULATE
- GL_DECAL
- GL_TEXTURE_ENV_MODE
- GL_TEXTURE_ENV_COLOR
- GL_TEXTURE_ENV
- GL_EYE_LINEAR
- GL_OBJECT_LINEAR
- GL_SPHERE_MAP
- GL_TEXTURE_GEN_MODE
- GL_OBJECT_PLANE
- GL_EYE_PLANE
- GL_NEAREST
- GL_LINEAR
- GL_NEAREST_MIPMAP_NEAREST
- GL_LINEAR_MIPMAP_NEAREST
- GL_NEAREST_MIPMAP_LINEAR
- GL_LINEAR_MIPMAP_LINEAR
- GL_TEXTURE_MAG_FILTER
- GL_TEXTURE_MIN_FILTER
- GL_TEXTURE_WRAP_S
- GL_TEXTURE_WRAP_T
- GL_CLAMP
- GL_REPEAT
- GL_POLYGON_OFFSET_UNITS
- GL_POLYGON_OFFSET_POINT
- GL_POLYGON_OFFSET_LINE
- GL_R3_G3_B2
- GL_V2F
- GL_V3F
- GL_C4UB_V2F

- GL_C4UB_V3F
- GL_C3F_V3F
- GL_N3F_V3F
- GL_C4F_N3F_V3F
- GL_T2F_V3F
- GL_T4F_V4F
- GL_T2F_C4UB_V3F
- GL_T2F_C3F_V3F
- GL_T2F_N3F_V3F
- GL_T2F_C4F_N3F_V3F
- GL_T4F_C4F_N3F_V4F
- GL_CLIP_PLANE0
- GL_CLIP_PLANE1
- GL_CLIP_PLANE2
- GL_CLIP_PLANE3
- GL_CLIP_PLANE4
- GL_CLIP_PLANE5
- GL_LIGHT0
- GL_COLOR_BUFFER_BIT
- GL_LIGHT1
- GL_LIGHT2
- GL_LIGHT3
- GL_LIGHT4
- GL_LIGHT5
- GL_LIGHT6
- GL_LIGHT7
- GL_HINT_BIT
- GL_POLYGON_OFFSET_FILL
- GL_POLYGON_OFFSET_FACTOR
- GL_ALPHA4
- GL_ALPHA8
- GL_ALPHA12
- GL_ALPHA16
- GL_LUMINANCE4
- GL_LUMINANCE8
- GL_LUMINANCE12

- GL_LUMINANCE16
- GL_LUMINANCE4_ALPHA4
- GL_LUMINANCE6_ALPHA2
- GL_LUMINANCE8_ALPHA8
- GL_LUMINANCE12_ALPHA4
- GL_LUMINANCE12_ALPHA12
- GL_LUMINANCE16_ALPHA16
- GL_INTENSITY
- GL_INTENSITY4
- GL_INTENSITY8
- GL_INTENSITY12
- GL_INTENSITY16
- GL_RGB4
- GL_RGB5
- GL_RGB8
- GL_RGB10
- GL_RGB12
- GL_RGB16
- GL_RGBA2
- GL_RGBA4
- GL_RGB5_A1
- GL_RGBA8
- GL_RGB10_A2
- GL_RGBA12
- GL_RGBA16
- GL_TEXTURE_RED_SIZE
- GL_TEXTURE_GREEN_SIZE
- GL_TEXTURE_BLUE_SIZE
- GL_TEXTURE_ALPHA_SIZE
- GL_TEXTURE_LUMINANCE_SIZE
- GL_TEXTURE_INTENSITY_SIZE
- GL_PROXY_TEXTURE_1D
- GL_PROXY_TEXTURE_2D
- GL_TEXTURE_PRIORITY
- GL_TEXTURE_RESIDENT
- GL_TEXTURE_BINDING_1D

- GL_TEXTURE_BINDING_2D
- GL_VERTEX_ARRAY
- GL_NORMAL_ARRAY
- GL_COLOR_ARRAY
- GL_INDEX_ARRAY
- GL_TEXTURE_COORD_ARRAY
- GL_EDGE_FLAG_ARRAY
- GL_VERTEX_ARRAY_SIZE
- GL_VERTEX_ARRAY_TYPE
- GL_VERTEX_ARRAY_STRIDE
- GL_NORMAL_ARRAY_TYPE
- GL_NORMAL_ARRAY_STRIDE
- GL_COLOR_ARRAY_SIZE
- GL_COLOR_ARRAY_TYPE
- GL_COLOR_ARRAY_STRIDE
- GL_INDEX_ARRAY_TYPE
- GL_INDEX_ARRAY_STRIDE
- GL_TEXTURE_COORD_ARRAY_SIZE
- GL_TEXTURE_COORD_ARRAY_TYPE
- GL_TEXTURE_COORD_ARRAY_STRIDE
- GL_EDGE_FLAG_ARRAY_STRIDE
- GL_VERTEX_ARRAY_POINTER
- GL_NORMAL_ARRAY_POINTER
- GL_COLOR_ARRAY_POINTER
- GL_INDEX_ARRAY_POINTER
- GL_TEXTURE_COORD_ARRAY_POINTER
- GL_EDGE_FLAG_ARRAY_POINTER
- GL_COLOR_INDEX1_EXT
- GL_COLOR_INDEX2_EXT
- GL_COLOR_INDEX4_EXT
- GL_COLOR_INDEX8_EXT
- GL_COLOR_INDEX12_EXT
- GL_COLOR_INDEX16_EXT
- GL_EVAL_BIT
- GL_LIST_BIT
- GL_TEXTURE_BIT

- GL_SCISSOR_BIT
- GL_ALL_ATTRIB_BITS
- GL_CLIENT_ALL_ATTRIB_BITS
- GL_SMOOTH_POINT_SIZE_RANGE
- GL_SMOOTH_POINT_SIZE_GRANULARITY
- GL_SMOOTH_LINE_WIDTH_RANGE
- GL_SMOOTH_LINE_WIDTH_GRANULARITY
- GL_UNSIGNED_BYTE_3_3_2
- GL_UNSIGNED_SHORT_4_4_4_4
- GL_UNSIGNED_SHORT_5_5_5_1
- GL_UNSIGNED_INT_8_8_8_8
- GL_UNSIGNED_INT_10_10_10_2
- GL_RESCALE_NORMAL
- GL_TEXTURE_BINDING_3D
- GL_PACK_SKIP_IMAGES
- GL_PACK_IMAGE_HEIGHT
- GL_UNPACK_SKIP_IMAGES
- GL_UNPACK_IMAGE_HEIGHT
- GL_TEXTURE_3D
- GL_PROXY_TEXTURE_3D
- GL_TEXTURE_DEPTH
- GL_TEXTURE_WRAP_R
- GL_MAX_3D_TEXTURE_SIZE
- GL_BGR
- GL_BGRA
- GL_MAX_ELEMENTS_VERTICES
- GL_MAX_ELEMENTS_INDICES
- GL_CLAMP_TO_EDGE
- GL_TEXTURE_MIN_LOD
- GL_TEXTURE_MAX_LOD
- GL_TEXTURE_BASE_LEVEL
- GL_TEXTURE_MAX_LEVEL
- GL_LIGHT_MODEL_COLOR_CONTROL
- GL_SINGLE_COLOR
- GL_SEPARATE_SPECULAR_COLOR
- GL_UNSIGNED_BYTE_2_3_3_REV

- GL_UNSIGNED_SHORT_5_6_5
- GL_UNSIGNED_SHORT_5_6_5_REV
- GL_UNSIGNED_SHORT_4_4_4_REV
- GL_UNSIGNED_SHORT_1_5_5_5_REV
- GL_UNSIGNED_INT_8_8_8_8_REV
- GL_ALIASED_POINT_SIZE_RANGE
- GL_ALIASED_LINE_WIDTH_RANGE
- void glAccum(GLenum op, GLfloat value)
- void glAlphaFunc(GLenum func, GLclampf ref)
- GLboolean glAreTexturesResident(GLsizei n, const GLuint *textures, GLboolean *residences)
- void glArrayElement(GLint i)
- void glBegin(GLenum mode)
- void glBindTexture(GLenum target, GLuint texture)
- void glBitmap(GLsizei width, GLsizei height, GLfloat xorig, GLfloat yorig, GLfloat xmove, GLfloat ymove, const GLubyte *bitmap)
- void glBlendFunc(GLenum sfactor, GLenum dfactor)
- void glCallList(GLuint list)
- void glCallLists(GLsizei n, GLenum type, const void *lists)
- void glClear(GLbitfield mask)
- void glClearAccum(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glClearColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)
- void glClearDepth(GLclampd depth)
- void glClearIndex(GLfloat c)
- void glClearStencil(GLint s)
- void glClipPlane(GLenum plane, const GLdouble *equation)
- void glColor3b(GLbyte red, GLbyte green, GLbyte blue)
- void glColor3bv(const GLbyte *v)
- void glColor3d(GLdouble red, GLdouble green, GLdouble blue)
- void glColor3dv(const GLdouble *v)
- void glColor3f(GLfloat red, GLfloat green, GLfloat blue)
- void glColor3fv(const GLfloat *v)
- void glColor3i(GLint red, GLint green, GLint blue)
- void glColor3iv(const GLint *v)
- void glColor3s(GLshort red, GLshort green, GLshort blue)
- void glColor3sv(const GLshort *v)
- void glColor3ub(GLubyte red, GLubyte green, GLubyte blue)

- void glColor3ubv(const GLubyte *v)
- void glColor3ui(GLuint red, GLuint green, GLuint blue)
- void glColor3uiv(const GLuint *v)
- void glColor3us(GLushort red, GLushort green, GLushort blue)
- void glColor3usv(const GLushort *v)
- void glColor4b(GLbyte red, GLbyte green, GLbyte blue, GLbyte alpha)
- void glColor4bv(const GLbyte *v)
- void glColor4d(GLdouble red, GLdouble green, GLdouble blue, GLdouble alpha)
- void glColor4dv(const GLdouble *v)
- void glColor4f(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glColor4fv(const GLfloat *v)
- void glColor4i(GLint red, GLint green, GLint blue, GLint alpha)
- void glColor4iv(const GLint *v)
- void glColor4s(GLshort red, GLshort green, GLshort blue, GLshort alpha)
- void glColor4sv(const GLshort *v)
- void glColor4ub(GLubyte red, GLubyte green, GLubyte blue, GLubyte alpha)
- void glColor4ubv(const GLubyte *v)
- void glColor4ui(GLuint red, GLuint green, GLuint blue, GLuint alpha)
- void glColor4uiv(const GLuint *v)
- void glColor4us(GLushort red, GLushort green, GLushort blue, GLushort alpha)
- void glColor4usv(const GLushort *v)
- void glColorMask(GLboolean red, GLboolean green, GLboolean blue, GLboolean alpha)
- void glColorMaterial(GLenum face, GLenum mode)
- void glColorPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)
- void glCopyPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum type)
- void glCopyTexImage1D(GLenum target, GLint level, GLenum internalFormat, GLint x, GLint y, GLsizei width, GLint border)
- void glCopyTexImage2D(GLenum target, GLint level, GLenum internalFormat, GLint x, GLint y, GLsizei width, GLsizei height, GLint border)
- void glCopyTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLint x, GLint y, GLsizei width)
- void glCopyTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint x, GLint y, GLsizei width, GLsizei height)
- void glCullFace(GLenum mode)
- void glDeleteLists(GLuint list, GLsizei range)
- void glDeleteTextures(GLsizei n, const GLuint *textures)
- void glDepthFunc(GLenum func)
- void glDepthMask(GLboolean flag)

- void glDepthRange(GLclampd zNear, GLclampd zFar)
- void glDisable(GLenum cap)
- void glDisableClientState(GLenum array)
- void glDrawArrays(GLenum mode, GLint first, GLsizei count)
- void glDrawBuffer(GLenum mode)
- void glDrawElements(GLenum mode, GLsizei count, GLenum type, const void *indices)
- void glDrawPixels(GLsizei width, GLsizei height, GLenum format, GLenum type, const void *pixels)
- void glEdgeFlag(GLboolean flag)
- void glEdgeFlagPointer(GLsizei stride, const void *pointer)
- void glEdgeFlagv(const GLboolean *flag)
- void glEnable(GLenum cap)
- void glEnableClientState(GLenum array)
- void glEnd(void)
- void glEndList(void)
- void glEvalCoord1d(GLdouble u)
- void glEvalCoord1dv(const GLdouble *u)
- void glEvalCoord1f(GLfloat u)
- void glEvalCoord1fv(const GLfloat *u)
- void glEvalCoord2d(GLdouble u, GLdouble v)
- void glEvalCoord2dv(const GLdouble *u)
- void glEvalCoord2f(GLfloat u, GLfloat v)
- void glEvalCoord2fv(const GLfloat *u)
- void glEvalMesh1(GLenum mode, GLint i1, GLint i2)
- void glEvalMesh2(GLenum mode, GLint i1, GLint i2, GLint j1, GLint j2)
- void glEvalPoint1(GLint i)
- void glEvalPoint2(GLint i, GLint j)
- void glFeedbackBuffer(GLsizei size, GLenum type, GLfloat *buffer)
- void glFinish(void)
- void glFlush(void)
- void glFogf(GLenum pname, GLfloat param)
- void glFogfv(GLenum pname, const GLfloat *params)
- void glFogi(GLenum pname, GLint param)
- void glFogiv(GLenum pname, const GLint *params)
- void glFrontFace(GLenum mode)
- void glFrustum(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble zNear, GLdouble zFar)

- GLuint glGenLists(GLsizei range)
- void glGenTextures(GLsizei n, GLuint *textures)
- void glGetBooleanv(GLenum pname, GLboolean *params)
- void glGetClipPlane(GLenum plane, GLdouble *equation)
- void glGetDoublev(GLenum pname, GLdouble *params)
- GLenum glGetError(void)
- void glGetFloatv(GLenum pname, GLfloat *params)
- void glGetIntegerv(GLenum pname, GLint *params)
- void glGetLightfv(GLenum light, GLenum pname, GLfloat *params)
- void glGetLightiv(GLenum light, GLenum pname, GLint *params)
- void glGetMapdv(GLenum target, GLenum query, GLdouble *v)
- void glGetMapfv(GLenum target, GLenum query, GLfloat *v)
- void glGetMapiv(GLenum target, GLenum query, GLint *v)
- void glGetMaterialfv(GLenum face, GLenum pname, GLfloat *params)
- void glGetMaterialiv(GLenum face, GLenum pname, GLint *params)
- void glGetPixelMapfv(GLenum map, GLfloat *values)
- void glGetPixelMapuiv(GLenum map, GLuint *values)
- void glGetPixelMapusv(GLenum map, GLushort *values)
- void glGetPointerv(GLenum pname, void* *params)
- void glGetPolygonStipple(GLubyte *mask)
- GLubyte * getString(GLenum name)
- void glGetTexEnvfv(GLenum target, GLenum pname, GLfloat *params)
- void glGetTexEnviv(GLenum target, GLenum pname, GLint *params)
- void glGetTexGendv(GLenum coord, GLenum pname, GLdouble *params)
- void glGetTexGenfv(GLenum coord, GLenum pname, GLfloat *params)
- void glGetTexGeniv(GLenum coord, GLenum pname, GLint *params)
- void glGetTexImage(GLenum target, GLint level, GLenum format, GLenum type, void *pixels)
- void glGetTexLevelParameterfv(GLenum target, GLint level, GLenum pname, GLfloat *params)
- void glGetTexLevelParameteriv(GLenum target, GLint level, GLenum pname, GLint *params)
- void glGetTexParameterfv(GLenum target, GLenum pname, GLfloat *params)
- void glGetTexParameteriv(GLenum target, GLenum pname, GLint *params)
- void glHint(GLenum target, GLenum mode)
- void glIndexMask(GLuint mask)
- void glIndexPointer(GLenum type, GLsizei stride, const void *pointer)
- void glIndexd(GLdouble c)
- void glIndexdv(const GLdouble *c)

- void glIndexf(GLfloat c)
- void glIndexfv(const GLfloat *c)
- void glIndexi(GLint c)
- void glIndexiv(const GLint *c)
- void glIndexs(GLshort c)
- void glIndexsv(const GLshort *c)
- void glIndexub(GLubyte c)
- void glIndexubv(const GLubyte *c)
- void glInitNames(void)
- void glInterleavedArrays(GLenum format, GLsizei stride, const void *pointer)
- GLboolean glIsEnabled(GLenum cap)
- GLboolean glIsList(GLuint list)
- GLboolean glIsTexture(GLuint texture)
- void glLightModelf(GLenum pname, GLfloat param)
- void glLightModelfv(GLenum pname, const GLfloat *params)
- void glLightModeli(GLenum pname, GLint param)
- void glLightModeliv(GLenum pname, const GLint *params)
- void glLightf(GLenum light, GLenum pname, GLfloat param)
- void glLightfv(GLenum light, GLenum pname, const GLfloat *params)
- void glLighti(GLenum light, GLenum pname, GLint param)
- void glLightiv(GLenum light, GLenum pname, const GLint *params)
- void glLineStipple(GLint factor, GLushort pattern)
- void glLineWidth(GLfloat width)
- void glListBase(GLuint base)
- void glLoadIdentity(void)
- void glLoadMatrixd(const GLdouble *m)
- void glLoadMatrixf(const GLfloat *m)
- void glLoadName(GLuint name)
- void glLogicOp(GLenum opcode)
- void glMap1d(GLenum target, GLdouble u1, GLdouble u2, GLint stride, GLint order, const GLdouble *points)
- void glMap1f(GLenum target, GLfloat u1, GLfloat u2, GLint stride, GLint order, const GLfloat *points)
- void glMap2d(GLenum target, GLdouble u1, GLdouble u2, GLint ustride, GLint uorder, GLdouble v1, GLdouble v2, GLint vstride, GLint vorder, const GLdouble *points)
- void glMap2f(GLenum target, GLfloat u1, GLfloat u2, GLint ustride, GLint uorder, GLfloat v1, GLfloat v2, GLint vstride, GLint vorder, const GLfloat *points)
- void glMapGrid1d(GLint un, GLdouble u1, GLdouble u2)
- void glMapGrid1f(GLint un, GLfloat u1, GLfloat u2)

- void glMapGrid2d(GLint un, GLdouble u1, GLdouble u2, GLint vn, GLdouble v1, GLdouble v2)
- void glMapGrid2f(GLint un, GLfloat u1, GLfloat u2, GLint vn, GLfloat v1, GLfloat v2)
- void glMaterialf(GLenum face, GLenum pname, GLfloat param)
- void glMaterialfv(GLenum face, GLenum pname, const GLfloat *params)
- void glMateriali(GLenum face, GLenum pname, GLint param)
- void glMaterialiv(GLenum face, GLenum pname, const GLint *params)
- void glMatrixMode(GLenum mode)
- void glMultMatrixd(const GLdouble *m)
- void glMultMatrixf(const GLfloat *m)
- void glNewList(GLuint list, GLenum mode)
- void glNormal3b(GLbyte nx, GLbyte ny, GLbyte nz)
- void glNormal3bv(const GLbyte *v)
- void glNormal3d(GLdouble nx, GLdouble ny, GLdouble nz)
- void glNormal3dv(const GLdouble *v)
- void glNormal3f(GLfloat nx, GLfloat ny, GLfloat nz)
- void glNormal3fv(const GLfloat *v)
- void glNormal3i(GLint nx, GLint ny, GLint nz)
- void glNormal3iv(const GLint *v)
- void glNormal3s(GLshort nx, GLshort ny, GLshort nz)
- void glNormal3sv(const GLshort *v)
- void glNormalPointer(GLenum type, GLsizei stride, const void *pointer)
- void glOrtho(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble zNear, GLdouble zFar)
- void glPassThrough(GLfloat token)
- void glPixelMapfv(GLenum map, GLsizei mapsize, const GLfloat *values)
- void glPixelMapuiv(GLenum map, GLsizei mapsize, const GLuint *values)
- void glPixelMapusv(GLenum map, GLsizei mapsize, const GLushort *values)
- void glPixelStoref(GLenum pname, GLfloat param)
- void glPixelStorei(GLenum pname, GLint param)
- void glPixelTransferf(GLenum pname, GLfloat param)
- void glPixelTransferi(GLenum pname, GLint param)
- void glPixelZoom(GLfloat xfactor, GLfloat yfactor)
- void glPointSize(GLfloat size)
- void glPolygonMode(GLenum face, GLenum mode)
- void glPolygonOffset(GLfloat factor, GLfloat units)
- void glPolygonStipple(const GLubyte *mask)

- void glPopAttrib(void)
- void glPopClientAttrib(void)
- void glPopMatrix(void)
- void glPopName(void)
- void glPrioritizeTextures(GLsizei n, const GLuint *textures, const GLclampf *priorities)
- void glPushAttrib(GLbitfield mask)
- void glPushClientAttrib(GLbitfield mask)
- void glPushMatrix(void)
- void glPushName(GLuint name)
- void glRasterPos2d(GLdouble x, GLdouble y)
- void glRasterPos2dv(const GLdouble *v)
- void glRasterPos2f(GLfloat x, GLfloat y)
- void glRasterPos2fv(const GLfloat *v)
- void glRasterPos2i(GLint x, GLint y)
- void glRasterPos2iv(const GLint *v)
- void glRasterPos2s(GLshort x, GLshort y)
- void glRasterPos2sv(const GLshort *v)
- void glRasterPos3d(GLdouble x, GLdouble y, GLdouble z)
- void glRasterPos3dv(const GLdouble *v)
- void glRasterPos3f(GLfloat x, GLfloat y, GLfloat z)
- void glRasterPos3fv(const GLfloat *v)
- void glRasterPos3i(GLint x, GLint y, GLint z)
- void glRasterPos3iv(const GLint *v)
- void glRasterPos3s(GLshort x, GLshort y, GLshort z)
- void glRasterPos3sv(const GLshort *v)
- void glRasterPos4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)
- void glRasterPos4dv(const GLdouble *v)
- void glRasterPos4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- void glRasterPos4fv(const GLfloat *v)
- void glRasterPos4i(GLint x, GLint y, GLint z, GLint w)
- void glRasterPos4iv(const GLint *v)
- void glRasterPos4s(GLshort x, GLshort y, GLshort z, GLshort w)
- void glRasterPos4sv(const GLshort *v)
- void glReadBuffer(GLenum mode)
- void glReadPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum format, GLenum type, void *pixels)

- void glRectd(GLdouble x1, GLdouble y1, GLdouble x2, GLdouble y2)
- void glRectdv(const GLdouble *v1, const GLdouble *v2)
- void glRectf(GLfloat x1, GLfloat y1, GLfloat x2, GLfloat y2)
- void glRectfv(const GLfloat *v1, const GLfloat *v2)
- void glRecti(GLint x1, GLint y1, GLint x2, GLint y2)
- void glRectiv(const GLint *v1, const GLint *v2)
- void glRects(GLshort x1, GLshort y1, GLshort x2, GLshort y2)
- void glRectsv(const GLshort *v1, const GLshort *v2)
- GLint glRenderMode(GLenum mode)
- void glRotated(GLdouble angle, GLdouble x, GLdouble y, GLdouble z)
- void glRotatef(GLfloat angle, GLfloat x, GLfloat y, GLfloat z)
- void glScaled(GLdouble x, GLdouble y, GLdouble z)
- void glScalef(GLfloat x, GLfloat y, GLfloat z)
- void glScissor(GLint x, GLint y, GLsizei width, GLsizei height)
- void glSelectBuffer(GLsizei size, GLuint *buffer)
- void glShadeModel(GLenum mode)
- void glStencilFunc(GLenum func, GLint ref, GLuint mask)
- void glStencilMask(GLuint mask)
- void glStencilOp(GLenum fail, GLenum zfail, GLenum zpass)
- void glTexCoord1d(GLdouble s)
- void glTexCoord1dv(const GLdouble *v)
- void glTexCoord1f(GLfloat s)
- void glTexCoord1fv(const GLfloat *v)
- void glTexCoord1i(GLint s)
- void glTexCoord1iv(const GLint *v)
- void glTexCoord1s(GLshort s)
- void glTexCoord1sv(const GLshort *v)
- void glTexCoord2d(GLdouble s, GLdouble t)
- void glTexCoord2dv(const GLdouble *v)
- void glTexCoord2f(GLfloat s, GLfloat t)
- void glTexCoord2fv(const GLfloat *v)
- void glTexCoord2i(GLint s, GLint t)
- void glTexCoord2iv(const GLint *v)
- void glTexCoord2s(GLshort s, GLshort t)
- void glTexCoord2sv(const GLshort *v)
- void glTexCoord3d(GLdouble s, GLdouble t, GLdouble r)

- `void glTexCoord3dv(const GLdouble *v)`
- `void glTexCoord3f(GLfloat s, GLfloat t, GLfloat r)`
- `void glTexCoord3fv(const GLfloat *v)`
- `void glTexCoord3i(GLint s, GLint t, GLint r)`
- `void glTexCoord3iv(const GLint *v)`
- `void glTexCoord3s(GLshort s, GLshort t, GLshort r)`
- `void glTexCoord3sv(const GLshort *v)`
- `void glTexCoord4d(GLdouble s, GLdouble t, GLdouble r, GLdouble q)`
- `void glTexCoord4dv(const GLdouble *v)`
- `void glTexCoord4f(GLfloat s, GLfloat t, GLfloat r, GLfloat q)`
- `void glTexCoord4fv(const GLfloat *v)`
- `void glTexCoord4i(GLint s, GLint t, GLint r, GLint q)`
- `void glTexCoord4iv(const GLint *v)`
- `void glTexCoord4s(GLshort s, GLshort t, GLshort r, GLshort q)`
- `void glTexCoord4sv(const GLshort *v)`
- `void glTexCoordPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)`
- `void glTexEnvf(GLenum target, GLenum pname, GLfloat param)`
- `void glTexEnvfv(GLenum target, GLenum pname, const GLfloat *params)`
- `void glTexEnvf(GLenum target, GLenum pname, GLint param)`
- `void glTexEnviv(GLenum target, GLenum pname, const GLint *params)`
- `void glTexGend(GLenum coord, GLenum pname, GLdouble param)`
- `void glTexGendv(GLenum coord, GLenum pname, const GLdouble *params)`
- `void glTexGenf(GLenum coord, GLenum pname, GLfloat param)`
- `void glTexGenfv(GLenum coord, GLenum pname, const GLfloat *params)`
- `void glTexGeni(GLenum coord, GLenum pname, GLint param)`
- `void glTexGeniv(GLenum coord, GLenum pname, const GLint *params)`
- `void glTexImage1D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLint border, GLenum format, GLenum type, const void *pixels)`
- `void glTexImage2D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, const void *pixels)`
- `void glTexParameterf(GLenum target, GLenum pname, GLfloat param)`
- `void glTexParameterfv(GLenum target, GLenum pname, const GLfloat *params)`
- `void glTexParameteri(GLenum target, GLenum pname, GLint param)`
- `void glTexParameteriv(GLenum target, GLenum pname, const GLint *params)`
- `void glTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLenum type, const void *pixels)`

- void glTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLenum type, const void *pixels)
- void glTranslated(GLdouble x, GLdouble y, GLdouble z)
- void glTranslatef(GLfloat x, GLfloat y, GLfloat z)
- void glVertex2d(GLdouble x, GLdouble y)
- void glVertex2dv(const GLdouble *v)
- void glVertex2f(GLfloat x, GLfloat y)
- void glVertex2fv(const GLfloat *v)
- void glVertex2i(GLint x, GLint y)
- void glVertex2iv(const GLint *v)
- void glVertex2s(GLshort x, GLshort y)
- void glVertex2sv(const GLshort *v)
- void glVertex3d(GLdouble x, GLdouble y, GLdouble z)
- void glVertex3dv(const GLdouble *v)
- void glVertex3f(GLfloat x, GLfloat y, GLfloat z)
- void glVertex3fv(const GLfloat *v)
- void glVertex3i(GLint x, GLint y, GLint z)
- void glVertex3iv(const GLint *v)
- void glVertex3s(GLshort x, GLshort y, GLshort z)
- void glVertex3sv(const GLshort *v)
- void glVertex4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)
- void glVertex4dv(const GLdouble *v)
- void glVertex4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- void glVertex4fv(const GLfloat *v)
- void glVertex4i(GLint x, GLint y, GLint z, GLint w)
- void glVertex4iv(const GLint *v)
- void glVertex4s(GLshort x, GLshort y, GLshort z, GLshort w)
- void glVertex4sv(const GLshort *v)
- void glVertexPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)
- void glViewport(GLint x, GLint y, GLsizei width, GLsizei height)

RINGOPENGL (OPENGL 1.3) FUNCTIONS REFERENCE

- GL_ZERO
- GL_FALSE
- GL_LOGIC_OP
- GL_NONE
- GL_TEXTURE_COMPONENTS
- GL_NO_ERROR
- GL_POINTS
- GL_CURRENT_BIT
- GL_TRUE
- GL_ONE
- GL_CLIENT_PIXEL_STORE_BIT
- GL_LINES
- GL_LINE_LOOP
- GL_POINT_BIT
- GL_CLIENT_VERTEX_ARRAY_BIT
- GL_LINE_STRIP
- GL_LINE_BIT
- GL_TRIANGLES
- GL_TRIANGLE_STRIP
- GL_TRIANGLE_FAN
- GL_QUADS
- GL_QUAD_STRIP
- GL_POLYGON_BIT
- GL_POLYGON
- GL_POLYGON_STIPPLE_BIT
- GL_PIXEL_MODE_BIT
- GL_LIGHTING_BIT

- GL_FOG_BIT
- GL_DEPTH_BUFFER_BIT
- GL_ACCUM
- GL_LOAD
- GL_RETURN
- GL_MULT
- GL_ADD
- GL_NEVER
- GL_ACCUM_BUFFER_BIT
- GL_LESS
- GL_EQUAL
- GL_LEQUAL
- GL_GREATER
- GL_NOTEQUAL
- GL_GEQUAL
- GL_ALWAYS
- GL_SRC_COLOR
- GL_ONE_MINUS_SRC_COLOR
- GL_SRC_ALPHA
- GL_ONE_MINUS_SRC_ALPHA
- GL_DST_ALPHA
- GL_ONE_MINUS_DST_ALPHA
- GL_DST_COLOR
- GL_ONE_MINUS_DST_COLOR
- GL_SRC_ALPHA_SATURATE
- GL_STENCIL_BUFFER_BIT
- GL_FRONT_LEFT
- GL_FRONT_RIGHT
- GL_BACK_LEFT
- GL_BACK_RIGHT
- GL_FRONT
- GL_BACK
- GL_LEFT
- GL_RIGHT
- GL_FRONT_AND_BACK
- GL_AUX0

- GL_AUX1
- GL_AUX2
- GL_AUX3
- GL_INVALID_ENUM
- GL_INVALID_VALUE
- GL_INVALID_OPERATION
- GL_STACK_OVERFLOW
- GL_STACK_UNDERFLOW
- GL_OUT_OF_MEMORY
- GL_2D
- GL_3D
- GL_3D_COLOR
- GL_3D_COLOR_TEXTURE
- GL_4D_COLOR_TEXTURE
- GL_PASS_THROUGH_TOKEN
- GL_POINT_TOKEN
- GL_LINE_TOKEN
- GL_POLYGON_TOKEN
- GL_BITMAP_TOKEN
- GL_DRAW_PIXEL_TOKEN
- GL_COPY_PIXEL_TOKEN
- GL_LINE_RESET_TOKEN
- GL_EXP
- GL_VIEWPORT_BIT
- GL_EXP2
- GL_CW
- GL_CCW
- GL_COEFF
- GL_ORDER
- GL_DOMAIN
- GL_CURRENT_COLOR
- GL_CURRENT_INDEX
- GL_CURRENT_NORMAL
- GL_CURRENT_TEXTURE_COORDS
- GL_CURRENT_RASTER_COLOR
- GL_CURRENT_RASTER_INDEX

- GL_CURRENT_RASTER_TEXTURE_COORDS
- GL_CURRENT_RASTER_POSITION
- GL_CURRENT_RASTER_POSITION_VALID
- GL_CURRENT_RASTER_DISTANCE
- GL_POINT_SMOOTH
- GL_POINT_SIZE
- GL_POINT_SIZE_RANGE
- GL_POINT_SIZE_GRANULARITY
- GL_LINE_SMOOTH
- GL_LINE_WIDTH
- GL_LINE_WIDTH_RANGE
- GL_LINE_WIDTH_GRANULARITY
- GL_LINE_STIPPLE
- GL_LINE_STIPPLE_PATTERN
- GL_LINE_STIPPLE_REPEAT
- GL_LIST_MODE
- GL_MAX_LIST_NESTING
- GL_LIST_BASE
- GL_LIST_INDEX
- GL_POLYGON_MODE
- GL_POLYGON_SMOOTH
- GL_POLYGON_STIPPLE
- GL_EDGE_FLAG
- GL_CULL_FACE
- GL_CULL_FACE_MODE
- GL_FRONT_FACE
- GL_LIGHTING
- GL_LIGHT_MODEL_LOCAL_VIEWER
- GL_LIGHT_MODEL_TWO_SIDE
- GL_LIGHT_MODEL_AMBIENT
- GL_SHADE_MODEL
- GL_COLOR_MATERIAL_FACE
- GL_COLOR_MATERIAL_PARAMETER
- GL_COLOR_MATERIAL
- GL_FOG
- GL_FOG_INDEX

- GL_FOG_DENSITY
- GL_FOG_START
- GL_FOG_END
- GL_FOG_MODE
- GL_FOG_COLOR
- GL_DEPTH_RANGE
- GL_DEPTH_TEST
- GL_DEPTH_WRITEMASK
- GL_DEPTH_CLEAR_VALUE
- GL_DEPTH_FUNC
- GL_ACCUM_CLEAR_VALUE
- GL_STENCIL_TEST
- GL_STENCIL_CLEAR_VALUE
- GL_STENCIL_FUNC
- GL_STENCIL_VALUE_MASK
- GL_STENCIL_FAIL
- GL_STENCIL_PASS_DEPTH_FAIL
- GL_STENCIL_PASS_DEPTH_PASS
- GL_STENCIL_REF
- GL_STENCIL_WRITEMASK
- GL_MATRIX_MODE
- GL_NORMALIZE
- GL_VIEWPORT
- GL_MODELVIEW_STACK_DEPTH
- GL_PROJECTION_STACK_DEPTH
- GL_TEXTURE_STACK_DEPTH
- GL_MODELVIEW_MATRIX
- GL_PROJECTION_MATRIX
- GL_TEXTURE_MATRIX
- GL_ATTRIB_STACK_DEPTH
- GL_CLIENT_ATTRIB_STACK_DEPTH
- GL_ALPHA_TEST
- GL_ALPHA_TEST_FUNC
- GL_ALPHA_TEST_REF
- GL_DITHER
- GL_BLEND_DST

- GL_BLEND_SRC
- GL_BLEND
- GL_LOGIC_OP_MODE
- GL_INDEX_LOGIC_OP
- GL_COLOR_LOGIC_OP
- GL_AUX_BUFFERS
- GL_DRAW_BUFFER
- GL_READ_BUFFER
- GL_SCISSOR_BOX
- GL_SCISSOR_TEST
- GL_INDEX_CLEAR_VALUE
- GL_INDEX_WRITEMASK
- GL_COLOR_CLEAR_VALUE
- GL_COLOR_WRITEMASK
- GL_INDEX_MODE
- GL_RGBA_MODE
- GL_DOUBLEBUFFER
- GL_STEREO
- GL_RENDER_MODE
- GL_PERSPECTIVE_CORRECTION_HINT
- GL_POINT_SMOOTH_HINT
- GL_LINE_SMOOTH_HINT
- GL_POLYGON_SMOOTH_HINT
- GL_FOG_HINT
- GL_TEXTURE_GEN_S
- GL_TEXTURE_GEN_T
- GL_TEXTURE_GEN_R
- GL_TEXTURE_GEN_Q
- GL_PIXEL_MAP_I_TO_I
- GL_PIXEL_MAP_S_TO_S
- GL_PIXEL_MAP_I_TO_R
- GL_PIXEL_MAP_I_TO_G
- GL_PIXEL_MAP_I_TO_B
- GL_PIXEL_MAP_I_TO_A
- GL_PIXEL_MAP_R_TO_R
- GL_PIXEL_MAP_G_TO_G

- GL_PIXEL_MAP_B_TO_B
- GL_PIXEL_MAP_A_TO_A
- GL_PIXEL_MAP_I_TO_I_SIZE
- GL_PIXEL_MAP_S_TO_S_SIZE
- GL_PIXEL_MAP_I_TO_R_SIZE
- GL_PIXEL_MAP_I_TO_G_SIZE
- GL_PIXEL_MAP_I_TO_B_SIZE
- GL_PIXEL_MAP_I_TO_A_SIZE
- GL_PIXEL_MAP_R_TO_R_SIZE
- GL_PIXEL_MAP_G_TO_G_SIZE
- GL_PIXEL_MAP_B_TO_B_SIZE
- GL_PIXEL_MAP_A_TO_A_SIZE
- GL_UNPACK_SWAP_BYTES
- GL_UNPACK_LSB_FIRST
- GL_UNPACK_ROW_LENGTH
- GL_UNPACK_SKIP_ROWS
- GL_UNPACK_SKIP_PIXELS
- GL_UNPACK_ALIGNMENT
- GL_PACK_SWAP_BYTES
- GL_PACK_LSB_FIRST
- GL_PACK_ROW_LENGTH
- GL_PACK_SKIP_ROWS
- GL_PACK_SKIP_PIXELS
- GL_PACK_ALIGNMENT
- GL_MAP_COLOR
- GL_MAP_STENCIL
- GL_INDEX_SHIFT
- GL_INDEX_OFFSET
- GL_RED_SCALE
- GL_RED_BIAS
- GL_ZOOM_X
- GL_ZOOM_Y
- GL_GREEN_SCALE
- GL_GREEN_BIAS
- GL_BLUE_SCALE
- GL_BLUE_BIAS

- GL_ALPHA_SCALE
- GL_ALPHA_BIAS
- GL_DEPTH_SCALE
- GL_DEPTH_BIAS
- GL_MAX_EVAL_ORDER
- GL_MAX_LIGHTS
- GL_MAX_CLIP_PLANES
- GL_MAX_TEXTURE_SIZE
- GL_MAX_PIXEL_MAP_TABLE
- GL_MAX_ATTRIB_STACK_DEPTH
- GL_MAX_MODELVIEW_STACK_DEPTH
- GL_MAX_NAME_STACK_DEPTH
- GL_MAX_PROJECTION_STACK_DEPTH
- GL_MAX_TEXTURE_STACK_DEPTH
- GL_MAX_VIEWPORT_DIMS
- GL_MAX_CLIENT_ATTRIB_STACK_DEPTH
- GL_SUBPIXEL_BITS
- GL_INDEX_BITS
- GL_RED_BITS
- GL_GREEN_BITS
- GL_BLUE_BITS
- GL_ALPHA_BITS
- GL_DEPTH_BITS
- GL_STENCIL_BITS
- GL_ACCUM_RED_BITS
- GL_ACCUM_GREEN_BITS
- GL_ACCUM_BLUE_BITS
- GL_ACCUM_ALPHA_BITS
- GL_NAME_STACK_DEPTH
- GL_AUTO_NORMAL
- GL_MAP1_COLOR_4
- GL_MAP1_INDEX
- GL_MAP1_NORMAL
- GL_MAP1_TEXTURE_COORD_1
- GL_MAP1_TEXTURE_COORD_2
- GL_MAP1_TEXTURE_COORD_3

- GL_MAP1_TEXTURE_COORD_4
- GL_MAP1_VERTEX_3
- GL_MAP1_VERTEX_4
- GL_MAP2_COLOR_4
- GL_MAP2_INDEX
- GL_MAP2_NORMAL
- GL_MAP2_TEXTURE_COORD_1
- GL_MAP2_TEXTURE_COORD_2
- GL_MAP2_TEXTURE_COORD_3
- GL_MAP2_TEXTURE_COORD_4
- GL_MAP2_VERTEX_3
- GL_MAP2_VERTEX_4
- GL_MAP1_GRID_DOMAIN
- GL_MAP1_GRID_SEGMENTS
- GL_MAP2_GRID_DOMAIN
- GL_MAP2_GRID_SEGMENTS
- GL_TEXTURE_1D
- GL_TEXTURE_2D
- GL_FEEDBACK_BUFFER_POINTER
- GL_FEEDBACK_BUFFER_SIZE
- GL_FEEDBACK_BUFFER_TYPE
- GL_SELECTION_BUFFER_POINTER
- GL_SELECTION_BUFFER_SIZE
- GL_TEXTURE_WIDTH
- GL_TRANSFORM_BIT
- GL_TEXTURE_HEIGHT
- GL_TEXTURE_INTERNAL_FORMAT
- GL_TEXTURE_BORDER_COLOR
- GL_TEXTURE_BORDER
- GL_DONT_CARE
- GL_FASTEST
- GL_NICEST
- GL_AMBIENT
- GL_DIFFUSE
- GL_SPECULAR
- GL_POSITION

- GL_SPOT_DIRECTION
- GL_SPOT_EXPONENT
- GL_SPOT_CUTOFF
- GL_CONSTANT_ATTENUATION
- GL_LINEAR_ATTENUATION
- GL_QUADRATIC_ATTENUATION
- GL_COMPILE
- GL_COMPILE_AND_EXECUTE
- GL_BYTE
- GL_UNSIGNED_BYTE
- GL_SHORT
- GL_UNSIGNED_SHORT
- GL_INT
- GL_UNSIGNED_INT
- GL_FLOAT
- GL_2_BYTES
- GL_3_BYTES
- GL_4_BYTES
- GL_DOUBLE
- GL_CLEAR
- GL_AND
- GL_AND_REVERSE
- GL_COPY
- GL_AND_INVERTED
- GL_NOOP
- GL_XOR
- GL_OR
- GL_NOR
- GL_EQUIV
- GL_INVERT
- GL_OR_REVERSE
- GL_COPY_INVERTED
- GL_OR_INVERTED
- GL_NAND
- GL_SET
- GL_EMISSION

- GL_SHININESS
- GL_AMBIENT_AND_DIFFUSE
- GL_COLOR_INDEXES
- GL_MODELVIEW
- GL_PROJECTION
- GL_TEXTURE
- GL_COLOR
- GL_DEPTH
- GL_STENCIL
- GL_COLOR_INDEX
- GL_STENCIL_INDEX
- GL_DEPTH_COMPONENT
- GL_RED
- GL_GREEN
- GL_BLUE
- GL_ALPHA
- GL_RGB
- GL_RGBA
- GL_LUMINANCE
- GL_LUMINANCE_ALPHA
- GL_BITMAP
- GL_POINT
- GL_LINE
- GL_FILL
- GL_RENDER
- GL_FEEDBACK
- GL_SELECT
- GL_FLAT
- GL_SMOOTH
- GL_KEEP
- GL_REPLACE
- GL_INCR
- GL_DECR
- GL_VENDOR
- GL_RENDERER
- GL_VERSION

- GL_EXTENSIONS
- GL_S
- GL_ENABLE_BIT
- GL_T
- GL_R
- GL_Q
- GL_MODULATE
- GL_DECAL
- GL_TEXTURE_ENV_MODE
- GL_TEXTURE_ENV_COLOR
- GL_TEXTURE_ENV
- GL_EYE_LINEAR
- GL_OBJECT_LINEAR
- GL_SPHERE_MAP
- GL_TEXTURE_GEN_MODE
- GL_OBJECT_PLANE
- GL_EYE_PLANE
- GL_NEAREST
- GL_LINEAR
- GL_NEAREST_MIPMAP_NEAREST
- GL_LINEAR_MIPMAP_NEAREST
- GL_NEAREST_MIPMAP_LINEAR
- GL_LINEAR_MIPMAP_LINEAR
- GL_TEXTURE_MAG_FILTER
- GL_TEXTURE_MIN_FILTER
- GL_TEXTURE_WRAP_S
- GL_TEXTURE_WRAP_T
- GL_CLAMP
- GL_REPEAT
- GL_POLYGON_OFFSET_UNITS
- GL_POLYGON_OFFSET_POINT
- GL_POLYGON_OFFSET_LINE
- GL_R3_G3_B2
- GL_V2F
- GL_V3F
- GL_C4UB_V2F

- GL_C4UB_V3F
- GL_C3F_V3F
- GL_N3F_V3F
- GL_C4F_N3F_V3F
- GL_T2F_V3F
- GL_T4F_V4F
- GL_T2F_C4UB_V3F
- GL_T2F_C3F_V3F
- GL_T2F_N3F_V3F
- GL_T2F_C4F_N3F_V3F
- GL_T4F_C4F_N3F_V4F
- GL_CLIP_PLANE0
- GL_CLIP_PLANE1
- GL_CLIP_PLANE2
- GL_CLIP_PLANE3
- GL_CLIP_PLANE4
- GL_CLIP_PLANE5
- GL_LIGHT0
- GL_COLOR_BUFFER_BIT
- GL_LIGHT1
- GL_LIGHT2
- GL_LIGHT3
- GL_LIGHT4
- GL_LIGHT5
- GL_LIGHT6
- GL_LIGHT7
- GL_HINT_BIT
- GL_POLYGON_OFFSET_FILL
- GL_POLYGON_OFFSET_FACTOR
- GL_ALPHA4
- GL_ALPHA8
- GL_ALPHA12
- GL_ALPHA16
- GL_LUMINANCE4
- GL_LUMINANCE8
- GL_LUMINANCE12

- GL_LUMINANCE16
- GL_LUMINANCE4_ALPHA4
- GL_LUMINANCE6_ALPHA2
- GL_LUMINANCE8_ALPHA8
- GL_LUMINANCE12_ALPHA4
- GL_LUMINANCE12_ALPHA12
- GL_LUMINANCE16_ALPHA16
- GL_INTENSITY
- GL_INTENSITY4
- GL_INTENSITY8
- GL_INTENSITY12
- GL_INTENSITY16
- GL_RGB4
- GL_RGB5
- GL_RGB8
- GL_RGB10
- GL_RGB12
- GL_RGB16
- GL_RGBA2
- GL_RGBA4
- GL_RGB5_A1
- GL_RGBA8
- GL_RGB10_A2
- GL_RGBA12
- GL_RGBA16
- GL_TEXTURE_RED_SIZE
- GL_TEXTURE_GREEN_SIZE
- GL_TEXTURE_BLUE_SIZE
- GL_TEXTURE_ALPHA_SIZE
- GL_TEXTURE_LUMINANCE_SIZE
- GL_TEXTURE_INTENSITY_SIZE
- GL_PROXY_TEXTURE_1D
- GL_PROXY_TEXTURE_2D
- GL_TEXTURE_PRIORITY
- GL_TEXTURE_RESIDENT
- GL_TEXTURE_BINDING_1D

- GL_TEXTURE_BINDING_2D
- GL_VERTEX_ARRAY
- GL_NORMAL_ARRAY
- GL_COLOR_ARRAY
- GL_INDEX_ARRAY
- GL_TEXTURE_COORD_ARRAY
- GL_EDGE_FLAG_ARRAY
- GL_VERTEX_ARRAY_SIZE
- GL_VERTEX_ARRAY_TYPE
- GL_VERTEX_ARRAY_STRIDE
- GL_NORMAL_ARRAY_TYPE
- GL_NORMAL_ARRAY_STRIDE
- GL_COLOR_ARRAY_SIZE
- GL_COLOR_ARRAY_TYPE
- GL_COLOR_ARRAY_STRIDE
- GL_INDEX_ARRAY_TYPE
- GL_INDEX_ARRAY_STRIDE
- GL_TEXTURE_COORD_ARRAY_SIZE
- GL_TEXTURE_COORD_ARRAY_TYPE
- GL_TEXTURE_COORD_ARRAY_STRIDE
- GL_EDGE_FLAG_ARRAY_STRIDE
- GL_VERTEX_ARRAY_POINTER
- GL_NORMAL_ARRAY_POINTER
- GL_COLOR_ARRAY_POINTER
- GL_INDEX_ARRAY_POINTER
- GL_TEXTURE_COORD_ARRAY_POINTER
- GL_EDGE_FLAG_ARRAY_POINTER
- GL_COLOR_INDEX1_EXT
- GL_COLOR_INDEX2_EXT
- GL_COLOR_INDEX4_EXT
- GL_COLOR_INDEX8_EXT
- GL_COLOR_INDEX12_EXT
- GL_COLOR_INDEX16_EXT
- GL_EVAL_BIT
- GL_LIST_BIT
- GL_TEXTURE_BIT

- GL_SCISSOR_BIT
- GL_ALL_ATTRIB_BITS
- GL_CLIENT_ALL_ATTRIB_BITS
- GL_SMOOTH_POINT_SIZE_RANGE
- GL_SMOOTH_POINT_SIZE_GRANULARITY
- GL_SMOOTH_LINE_WIDTH_RANGE
- GL_SMOOTH_LINE_WIDTH_GRANULARITY
- GL_UNSIGNED_BYTE_3_3_2
- GL_UNSIGNED_SHORT_4_4_4_4
- GL_UNSIGNED_SHORT_5_5_5_1
- GL_UNSIGNED_INT_8_8_8_8
- GL_UNSIGNED_INT_10_10_10_2
- GL_RESCALE_NORMAL
- GL_TEXTURE_BINDING_3D
- GL_PACK_SKIP_IMAGES
- GL_PACK_IMAGE_HEIGHT
- GL_UNPACK_SKIP_IMAGES
- GL_UNPACK_IMAGE_HEIGHT
- GL_TEXTURE_3D
- GL_PROXY_TEXTURE_3D
- GL_TEXTURE_DEPTH
- GL_TEXTURE_WRAP_R
- GL_MAX_3D_TEXTURE_SIZE
- GL_BGR
- GL_BGRA
- GL_MAX_ELEMENTS_VERTICES
- GL_MAX_ELEMENTS_INDICES
- GL_CLAMP_TO_EDGE
- GL_TEXTURE_MIN_LOD
- GL_TEXTURE_MAX_LOD
- GL_TEXTURE_BASE_LEVEL
- GL_TEXTURE_MAX_LEVEL
- GL_LIGHT_MODEL_COLOR_CONTROL
- GL_SINGLE_COLOR
- GL_SEPARATE_SPECULAR_COLOR
- GL_UNSIGNED_BYTE_2_3_3_REV

- GL_UNSIGNED_SHORT_5_6_5
- GL_UNSIGNED_SHORT_5_6_5_REV
- GL_UNSIGNED_SHORT_4_4_4_4_REV
- GL_UNSIGNED_SHORT_1_5_5_5_REV
- GL_UNSIGNED_INT_8_8_8_8_REV
- GL_ALIASED_POINT_SIZE_RANGE
- GL_ALIASED_LINE_WIDTH_RANGE
- GL_MULTISAMPLE
- GL_SAMPLE_ALPHA_TO_COVERAGE
- GL_SAMPLE_ALPHA_TO_ONE
- GL_SAMPLE_COVERAGE
- GL_SAMPLE_BUFFERS
- GL_SAMPLES
- GL_SAMPLE_COVERAGE_VALUE
- GL_SAMPLE_COVERAGE_INVERT
- GL_CLAMP_TO_BORDER
- GL_TEXTURE0
- GL_TEXTURE1
- GL_TEXTURE2
- GL_TEXTURE3
- GL_TEXTURE4
- GL_TEXTURE5
- GL_TEXTURE6
- GL_TEXTURE7
- GL_TEXTURE8
- GL_TEXTURE9
- GL_TEXTURE10
- GL_TEXTURE11
- GL_TEXTURE12
- GL_TEXTURE13
- GL_TEXTURE14
- GL_TEXTURE15
- GL_TEXTURE16
- GL_TEXTURE17
- GL_TEXTURE18
- GL_TEXTURE19

- GL_TEXTURE20
- GL_TEXTURE21
- GL_TEXTURE22
- GL_TEXTURE23
- GL_TEXTURE24
- GL_TEXTURE25
- GL_TEXTURE26
- GL_TEXTURE27
- GL_TEXTURE28
- GL_TEXTURE29
- GL_TEXTURE30
- GL_TEXTURE31
- GL_ACTIVE_TEXTURE
- GL_CLIENT_ACTIVE_TEXTURE
- GL_MAX_TEXTURE_UNITS
- GL_TRANSPOSE_MODELVIEW_MATRIX
- GL_TRANSPOSE_PROJECTION_MATRIX
- GL_TRANSPOSE_TEXTURE_MATRIX
- GL_TRANSPOSE_COLOR_MATRIX
- GL_SUBTRACT
- GL_COMPRESSED_ALPHA
- GL_COMPRESSED_LUMINANCE
- GL_COMPRESSED_LUMINANCE_ALPHA
- GL_COMPRESSED_INTENSITY
- GL_COMPRESSED_RGB
- GL_COMPRESSED_RGBA
- GL_TEXTURE_COMPRESSION_HINT
- GL_NORMAL_MAP
- GL_REFLECTION_MAP
- GL_TEXTURE_CUBE_MAP
- GL_TEXTURE_BINDING_CUBE_MAP
- GL_TEXTURE_CUBE_MAP_POSITIVE_X
- GL_TEXTURE_CUBE_MAP_NEGATIVE_X
- GL_TEXTURE_CUBE_MAP_POSITIVE_Y
- GL_TEXTURE_CUBE_MAP_NEGATIVE_Y
- GL_TEXTURE_CUBE_MAP_POSITIVE_Z

- GL_TEXTURE_CUBE_MAP_NEGATIVE_Z
- GL_PROXY_TEXTURE_CUBE_MAP
- GL_MAX_CUBE_MAP_TEXTURE_SIZE
- GL_COMBINE
- GL_COMBINE_RGB
- GL_COMBINE_ALPHA
- GL_RGB_SCALE
- GL_ADD_SIGNED
- GL_INTERPOLATE
- GL_CONSTANT
- GL_PRIMARY_COLOR
- GL_PREVIOUS
- GL_SOURCE0_RGB
- GL_SOURCE1_RGB
- GL_SOURCE2_RGB
- GL_SOURCE0_ALPHA
- GL_SOURCE1_ALPHA
- GL_SOURCE2_ALPHA
- GL_OPERAND0_RGB
- GL_OPERAND1_RGB
- GL_OPERAND2_RGB
- GL_OPERAND0_ALPHA
- GL_OPERAND1_ALPHA
- GL_OPERAND2_ALPHA
- GL_TEXTURE_COMPRESSED_IMAGE_SIZE
- GL_TEXTURE_COMPRESSED
- GL_NUM_COMPRESSED_TEXTURE_FORMATS
- GL_COMPRESSED_TEXTURE_FORMATS
- GL_DOT3_RGB
- GL_DOT3_RGBA
- GL_MULTISAMPLE_BIT
- void glAccum(GLenum op, GLfloat value)
- void glAlphaFunc(GLenum func, GLclampf ref)
- GLboolean glAreTexturesResident(GLsizei n, const GLuint *textures, GLboolean *residences)
- void glArrayElement(GLint i)
- void glBegin(GLenum mode)

- void glBindTexture(GLenum target, GLuint texture)
- void glBitmap(GLsizei width, GLsizei height, GLfloat xorig, GLfloat yorig, GLfloat xmove, GLfloat ymove, const GLubyte *bitmap)
- void glBlendFunc(GLenum sfactor, GLenum dfactor)
- void glCallList(GLuint list)
- void glCallLists(GLsizei n, GLenum type, const void *lists)
- void glClear(GLbitfield mask)
- void glClearAccum(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glClearColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)
- void glClearDepth(GLclampd depth)
- void glClearIndex(GLfloat c)
- void glClearStencil(GLint s)
- void glClipPlane(GLenum plane, const GLdouble *equation)
- void glColor3b(GLbyte red, GLbyte green, GLbyte blue)
- void glColor3bv(const GLbyte *v)
- void glColor3d(GLdouble red, GLdouble green, GLdouble blue)
- void glColor3dv(const GLdouble *v)
- void glColor3f(GLfloat red, GLfloat green, GLfloat blue)
- void glColor3fv(const GLfloat *v)
- void glColor3i(GLint red, GLint green, GLint blue)
- void glColor3iv(const GLint *v)
- void glColor3s(GLshort red, GLshort green, GLshort blue)
- void glColor3sv(const GLshort *v)
- void glColor3ub(GLubyte red, GLubyte green, GLubyte blue)
- void glColor3ubv(const GLubyte *v)
- void glColor3ui(GLuint red, GLuint green, GLuint blue)
- void glColor3uiv(const GLuint *v)
- void glColor3us(GLushort red, GLushort green, GLushort blue)
- void glColor3usv(const GLushort *v)
- void glColor4b(GLbyte red, GLbyte green, GLbyte blue, GLbyte alpha)
- void glColor4bv(const GLbyte *v)
- void glColor4d(GLdouble red, GLdouble green, GLdouble blue, GLdouble alpha)
- void glColor4dv(const GLdouble *v)
- void glColor4f(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glColor4fv(const GLfloat *v)
- void glColor4i(GLint red, GLint green, GLint blue, GLint alpha)

- `void glColor4iv(const GLint *v)`
- `void glColor4s(GLshort red, GLshort green, GLshort blue, GLshort alpha)`
- `void glColor4sv(const GLshort *v)`
- `void glColor4ub(GLubyte red, GLubyte green, GLubyte blue, GLubyte alpha)`
- `void glColor4ubv(const GLubyte *v)`
- `void glColor4ui(GLuint red, GLuint green, GLuint blue, GLuint alpha)`
- `void glColor4uiv(const GLuint *v)`
- `void glColor4us(GLushort red, GLushort green, GLushort blue, GLushort alpha)`
- `void glColor4usv(const GLushort *v)`
- `void glColorMask(GLboolean red, GLboolean green, GLboolean blue, GLboolean alpha)`
- `void glColorMaterial(GLenum face, GLenum mode)`
- `void glColorPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)`
- `void glCopyPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum type)`
- `void glCopyTexImage1D(GLenum target, GLint level, GLenum internalFormat, GLint x, GLint y, GLsizei width, GLint border)`
- `void glCopyTexImage2D(GLenum target, GLint level, GLenum internalFormat, GLint x, GLint y, GLsizei width, GLsizei height, GLint border)`
- `void glCopyTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLint x, GLint y, GLsizei width)`
- `void glCopyTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint x, GLint y, GLsizei width, GLsizei height)`
- `void glCullFace(GLenum mode)`
- `void glDeleteLists(GLuint list, GLsizei range)`
- `void glDeleteTextures(GLsizei n, const GLuint *textures)`
- `void glDepthFunc(GLenum func)`
- `void glDepthMask(GLboolean flag)`
- `void glDepthRange(GLclampd zNear, GLclampd zFar)`
- `void glDisable(GLenum cap)`
- `void glDisableClientState(GLenum array)`
- `void glDrawArrays(GLenum mode, GLint first, GLsizei count)`
- `void glDrawBuffer(GLenum mode)`
- `void glDrawElements(GLenum mode, GLsizei count, GLenum type, const void *indices)`
- `void glDrawPixels(GLsizei width, GLsizei height, GLenum format, GLenum type, const void *pixels)`
- `void glEdgeFlag(GLboolean flag)`
- `void glEdgeFlagPointer(GLsizei stride, const void *pointer)`
- `void glEdgeFlagv(const GLboolean *flag)`
- `void glEnable(GLenum cap)`
- `void glEnableClientState(GLenum array)`

- void glEnd(void)
- void glEndList(void)
- void glEvalCoord1d(GLdouble u)
- void glEvalCoord1dv(const GLdouble *u)
- void glEvalCoord1f(GLfloat u)
- void glEvalCoord1fv(const GLfloat *u)
- void glEvalCoord2d(GLdouble u, GLdouble v)
- void glEvalCoord2dv(const GLdouble *u)
- void glEvalCoord2f(GLfloat u, GLfloat v)
- void glEvalCoord2fv(const GLfloat *u)
- void glEvalMesh1(GLenum mode, GLint i1, GLint i2)
- void glEvalMesh2(GLenum mode, GLint i1, GLint i2, GLint j1, GLint j2)
- void glEvalPoint1(GLint i)
- void glEvalPoint2(GLint i, GLint j)
- void glFeedbackBuffer(GLsizei size, GLenum type, GLfloat *buffer)
- void glFinish(void)
- void glFlush(void)
- void glFogf(GLenum pname, GLfloat param)
- void glFogfv(GLenum pname, const GLfloat *params)
- void glFogi(GLenum pname, GLint param)
- void glFogiv(GLenum pname, const GLint *params)
- void glFrontFace(GLenum mode)
- void glFrustum(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble zNear, GLdouble zFar)
- GLuint glGenLists(GLsizei range)
- void glGenTextures(GLsizei n, GLuint *textures)
- void glGetBooleanv(GLenum pname, GLboolean *params)
- void glGetClipPlane(GLenum plane, GLdouble *equation)
- void glGetDoublev(GLenum pname, GLdouble *params)
- GLenum glGetError(void)
- void glGetFloatv(GLenum pname, GLfloat *params)
- void glGetIntegerv(GLenum pname, GLint *params)
- void glGetLightfv(GLenum light, GLenum pname, GLfloat *params)
- void glGetLightiv(GLenum light, GLenum pname, GLint *params)
- void glGetMapdv(GLenum target, GLenum query, GLdouble *v)
- void glGetMapfv(GLenum target, GLenum query, GLfloat *v)

- void glGetMapiv(GLenum target, GLenum query, GLint *v)
- void glGetMaterialfv(GLenum face, GLenum pname, GLfloat *params)
- void glGetMaterialiv(GLenum face, GLenum pname, GLint *params)
- void glGetPixelMapfv(GLenum map, GLfloat *values)
- void glGetPixelMapuiv(GLenum map, GLuint *values)
- void glGetPixelMapusv(GLenum map, GLushort *values)
- void glGetPointerv(GLenum pname, void* *params)
- void glGetPolygonStipple(GLubyte *mask)
- GLubyte * getString(GLenum name)
- void glGetTexEnvfv(GLenum target, GLenum pname, GLfloat *params)
- void glGetTexEnviv(GLenum target, GLenum pname, GLint *params)
- void glGetTexGendv(GLenum coord, GLenum pname, GLdouble *params)
- void glGetTexGenfv(GLenum coord, GLenum pname, GLfloat *params)
- void glGetTexGeniv(GLenum coord, GLenum pname, GLint *params)
- void glGetTexImage(GLenum target, GLint level, GLenum format, GLenum type, void *pixels)
- void glGetTexLevelParameterfv(GLenum target, GLint level, GLenum pname, GLfloat *params)
- void glGetTexLevelParameteriv(GLenum target, GLint level, GLenum pname, GLint *params)
- void glGetTexParameterfv(GLenum target, GLenum pname, GLfloat *params)
- void glGetTexParameteriv(GLenum target, GLenum pname, GLint *params)
- void glHint(GLenum target, GLenum mode)
- void glIndexMask(GLuint mask)
- void glIndexPointer(GLenum type, GLsizei stride, const void *pointer)
- void glIndxd(GLdouble c)
- void glIndexedv(const GLdouble *c)
- void glIndexf(GLfloat c)
- void glIndexfv(const GLfloat *c)
- void glIndexi(GLint c)
- void glIndexiv(const GLint *c)
- void glIndexs(GLshort c)
- void glIndexsv(const GLshort *c)
- void glIndexub(GLubyte c)
- void glIndexubv(const GLubyte *c)
- void glInitNames(void)
- void glInterleavedArrays(GLenum format, GLsizei stride, const void *pointer)
- GLboolean glIsEnabled(GLenum cap)
- GLboolean glIsList(GLuint list)

- GLboolean glIsTexture(GLuint texture)
- void glLightModelf(GLenum pname, GLfloat param)
- void glLightModelfv(GLenum pname, const GLfloat *params)
- void glLightModeli(GLenum pname, GLint param)
- void glLightModeliv(GLenum pname, const GLint *params)
- void glLightf(GLenum light, GLenum pname, GLfloat param)
- void glLightfv(GLenum light, GLenum pname, const GLfloat *params)
- void glLighti(GLenum light, GLenum pname, GLint param)
- void glLightiv(GLenum light, GLenum pname, const GLint *params)
- void glLineStipple(GLint factor, GLushort pattern)
- void glLineWidth(GLfloat width)
- void glListBase(GLuint base)
- void glLoadIdentity(void)
- void glLoadMatrixd(const GLdouble *m)
- void glLoadMatrixf(const GLfloat *m)
- void glLoadName(GLuint name)
- void glLogicOp(GLenum opcode)
- void glMap1d(GLenum target, GLdouble u1, GLdouble u2, GLint stride, GLint order, const GLdouble *points)
- void glMap1f(GLenum target, GLfloat u1, GLfloat u2, GLint stride, GLint order, const GLfloat *points)
- void glMap2d(GLenum target, GLdouble u1, GLdouble u2, GLint ustride, GLint uorder, GLdouble v1, GLdouble v2, GLint vstride, GLint vorder, const GLdouble *points)
- void glMap2f(GLenum target, GLfloat u1, GLfloat u2, GLint ustride, GLint uorder, GLfloat v1, GLfloat v2, GLint vstride, GLint vorder, const GLfloat *points)
- void glMapGrid1d(GLint un, GLdouble u1, GLdouble u2)
- void glMapGrid1f(GLint un, GLfloat u1, GLfloat u2)
- void glMapGrid2d(GLint un, GLdouble u1, GLdouble u2, GLint vn, GLdouble v1, GLdouble v2)
- void glMapGrid2f(GLint un, GLfloat u1, GLfloat u2, GLint vn, GLfloat v1, GLfloat v2)
- void glMaterialf(GLenum face, GLenum pname, GLfloat param)
- void glMaterialfv(GLenum face, GLenum pname, const GLfloat *params)
- void glMateriali(GLenum face, GLenum pname, GLint param)
- void glMaterialiv(GLenum face, GLenum pname, const GLint *params)
- void glMatrixMode(GLenum mode)
- void glMultMatrixd(const GLdouble *m)
- void glMultMatrixf(const GLfloat *m)
- void glNewList(GLuint list, GLenum mode)
- void glNormal3b(GLbyte nx, GLbyte ny, GLbyte nz)
- void glNormal3bv(const GLbyte *v)

- void glNormal3d(GLdouble nx, GLdouble ny, GLdouble nz)
- void glNormal3dv(const GLdouble *v)
- void glNormal3f(GLfloat nx, GLfloat ny, GLfloat nz)
- void glNormal3fv(const GLfloat *v)
- void glNormal3i(GLint nx, GLint ny, GLint nz)
- void glNormal3iv(const GLint *v)
- void glNormal3s(GLshort nx, GLshort ny, GLshort nz)
- void glNormal3sv(const GLshort *v)
- void glNormalPointer(GLenum type, GLsizei stride, const void *pointer)
- void glOrtho(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble zNear, GLdouble zFar)
- void glPassThrough(GLfloat token)
- void glPixelMapfv(GLenum map, GLsizei mapsize, const GLfloat *values)
- void glPixelMapuiv(GLenum map, GLsizei mapsize, const GLuint *values)
- void glPixelMapusv(GLenum map, GLsizei mapsize, const GLushort *values)
- void glPixelStoref(GLenum pname, GLfloat param)
- void glPixelStorei(GLenum pname, GLint param)
- void glPixelTransferf(GLenum pname, GLfloat param)
- void glPixelTransferi(GLenum pname, GLint param)
- void glPixelZoom(GLfloat xfactor, GLfloat yfactor)
- void glPointSize(GLfloat size)
- void glPolygonMode(GLenum face, GLenum mode)
- void glPolygonOffset(GLfloat factor, GLfloat units)
- void glPolygonStipple(const GLubyte *mask)
- void glPopAttrib(void)
- void glPopClientAttrib(void)
- void glPopMatrix(void)
- void glPopName(void)
- void glPrioritizeTextures(GLsizei n, const GLuint *textures, const GLclampf *priorities)
- void glPushAttrib(GLbitfield mask)
- void glPushClientAttrib(GLbitfield mask)
- void glPushMatrix(void)
- void glPushName(GLuint name)
- void glRasterPos2d(GLdouble x, GLdouble y)
- void glRasterPos2dv(const GLdouble *v)
- void glRasterPos2f(GLfloat x, GLfloat y)

- void glRasterPos2fv(const GLfloat *v)
- void glRasterPos2i(GLint x, GLint y)
- void glRasterPos2iv(const GLint *v)
- void glRasterPos2s(GLshort x, GLshort y)
- void glRasterPos2sv(const GLshort *v)
- void glRasterPos3d(GLdouble x, GLdouble y, GLdouble z)
- void glRasterPos3dv(const GLdouble *v)
- void glRasterPos3f(GLfloat x, GLfloat y, GLfloat z)
- void glRasterPos3fv(const GLfloat *v)
- void glRasterPos3i(GLint x, GLint y, GLint z)
- void glRasterPos3iv(const GLint *v)
- void glRasterPos3s(GLshort x, GLshort y, GLshort z)
- void glRasterPos3sv(const GLshort *v)
- void glRasterPos4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)
- void glRasterPos4dv(const GLdouble *v)
- void glRasterPos4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- void glRasterPos4fv(const GLfloat *v)
- void glRasterPos4i(GLint x, GLint y, GLint z, GLint w)
- void glRasterPos4iv(const GLint *v)
- void glRasterPos4s(GLshort x, GLshort y, GLshort z, GLshort w)
- void glRasterPos4sv(const GLshort *v)
- void glReadBuffer(GLenum mode)
- void glReadPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum format, GLenum type, void *pixels)
- void glRectd(GLdouble x1, GLdouble y1, GLdouble x2, GLdouble y2)
- void glRectdv(const GLdouble *v1, const GLdouble *v2)
- void glRectf(GLfloat x1, GLfloat y1, GLfloat x2, GLfloat y2)
- void glRectfv(const GLfloat *v1, const GLfloat *v2)
- void glRecti(GLint x1, GLint y1, GLint x2, GLint y2)
- void glRectiv(const GLint *v1, const GLint *v2)
- void glRects(GLshort x1, GLshort y1, GLshort x2, GLshort y2)
- void glRectsv(const GLshort *v1, const GLshort *v2)
- GLint glRenderMode(GLenum mode)
- void glRotated(GLdouble angle, GLdouble x, GLdouble y, GLdouble z)
- void glRotatef(GLfloat angle, GLfloat x, GLfloat y, GLfloat z)
- void glScaled(GLdouble x, GLdouble y, GLdouble z)

- void glScalef(GLfloat x, GLfloat y, GLfloat z)
- void glScissor(GLint x, GLint y, GLsizei width, GLsizei height)
- void glSelectBuffer(GLsizei size, GLuint *buffer)
- void glShadeModel(GLenum mode)
- void glStencilFunc(GLenum func, GLint ref, GLuint mask)
- void glStencilMask(GLuint mask)
- void glStencilOp(GLenum fail, GLenum zfail, GLenum zpass)
- void glTexCoord1d(GLdouble s)
- void glTexCoord1dv(const GLdouble *v)
- void glTexCoord1f(GLfloat s)
- void glTexCoord1fv(const GLfloat *v)
- void glTexCoord1i(GLint s)
- void glTexCoord1iv(const GLint *v)
- void glTexCoord1s(GLshort s)
- void glTexCoord1sv(const GLshort *v)
- void glTexCoord2d(GLdouble s, GLdouble t)
- void glTexCoord2dv(const GLdouble *v)
- void glTexCoord2f(GLfloat s, GLfloat t)
- void glTexCoord2fv(const GLfloat *v)
- void glTexCoord2i(GLint s, GLint t)
- void glTexCoord2iv(const GLint *v)
- void glTexCoord2s(GLshort s, GLshort t)
- void glTexCoord2sv(const GLshort *v)
- void glTexCoord3d(GLdouble s, GLdouble t, GLdouble r)
- void glTexCoord3dv(const GLdouble *v)
- void glTexCoord3f(GLfloat s, GLfloat t, GLfloat r)
- void glTexCoord3fv(const GLfloat *v)
- void glTexCoord3i(GLint s, GLint t, GLint r)
- void glTexCoord3iv(const GLint *v)
- void glTexCoord3s(GLshort s, GLshort t, GLshort r)
- void glTexCoord3sv(const GLshort *v)
- void glTexCoord4d(GLdouble s, GLdouble t, GLdouble r, GLdouble q)
- void glTexCoord4dv(const GLdouble *v)
- void glTexCoord4f(GLfloat s, GLfloat t, GLfloat r, GLfloat q)
- void glTexCoord4fv(const GLfloat *v)
- void glTexCoord4i(GLint s, GLint t, GLint r, GLint q)

- void glTexCoord4iv(const GLint *v)
- void glTexCoord4s(GLshort s, GLshort t, GLshort r, GLshort q)
- void glTexCoord4sv(const GLshort *v)
- void glTexCoordPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)
- void glTexEnvf(GLenum target, GLenum pname, GLfloat param)
- void glTexEnvfv(GLenum target, GLenum pname, const GLfloat *params)
- void glTexEnvf(GLenum target, GLenum pname, GLint param)
- void glTexEnviv(GLenum target, GLenum pname, const GLint *params)
- void glTexGend(GLenum coord, GLenum pname, GLdouble param)
- void glTexGendv(GLenum coord, GLenum pname, const GLdouble *params)
- void glTexGenf(GLenum coord, GLenum pname, GLfloat param)
- void glTexGenfv(GLenum coord, GLenum pname, const GLfloat *params)
- void glTexGeni(GLenum coord, GLenum pname, GLint param)
- void glTexGeniv(GLenum coord, GLenum pname, const GLint *params)
- void glTexImage1D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLint border, GLenum format, GLenum type, const void *pixels)
- void glTexImage2D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, const void *pixels)
- void glTexParameterf(GLenum target, GLenum pname, GLfloat param)
- void glTexParameterfv(GLenum target, GLenum pname, const GLfloat *params)
- void glTexParameteri(GLenum target, GLenum pname, GLint param)
- void glTexParameteriv(GLenum target, GLenum pname, const GLint *params)
- void glTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLenum type, const void *pixels)
- void glTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLenum type, const void *pixels)
- void glTranslated(GLdouble x, GLdouble y, GLdouble z)
- void glTranslatef(GLfloat x, GLfloat y, GLfloat z)
- void glVertex2d(GLdouble x, GLdouble y)
- void glVertex2dv(const GLdouble *v)
- void glVertex2f(GLfloat x, GLfloat y)
- void glVertex2fv(const GLfloat *v)
- void glVertex2i(GLint x, GLint y)
- void glVertex2iv(const GLint *v)
- void glVertex2s(GLshort x, GLshort y)
- void glVertex2sv(const GLshort *v)
- void glVertex3d(GLdouble x, GLdouble y, GLdouble z)

- `void glVertex3dv(const GLdouble *v)`
- `void glVertex3f(GLfloat x, GLfloat y, GLfloat z)`
- `void glVertex3fv(const GLfloat *v)`
- `void glVertex3i(GLint x, GLint y, GLint z)`
- `void glVertex3iv(const GLint *v)`
- `void glVertex3s(GLshort x, GLshort y, GLshort z)`
- `void glVertex3sv(const GLshort *v)`
- `void glVertex4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)`
- `void glVertex4dv(const GLdouble *v)`
- `void glVertex4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)`
- `void glVertex4fv(const GLfloat *v)`
- `void glVertex4i(GLint x, GLint y, GLint z, GLint w)`
- `void glVertex4iv(const GLint *v)`
- `void glVertex4s(GLshort x, GLshort y, GLshort z, GLshort w)`
- `void glVertex4sv(const GLshort *v)`
- `void glVertexPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)`
- `void glViewport(GLint x, GLint y, GLsizei width, GLsizei height)`

RINGOPENGL (OPENGL 1.4) FUNCTIONS REFERENCE

- GL_ZERO
- GL_FALSE
- GL_LOGIC_OP
- GL_NONE
- GL_TEXTURE_COMPONENTS
- GL_NO_ERROR
- GL_POINTS
- GL_CURRENT_BIT
- GL_TRUE
- GL_ONE
- GL_CLIENT_PIXEL_STORE_BIT
- GL_LINES
- GL_LINE_LOOP
- GL_POINT_BIT
- GL_CLIENT_VERTEX_ARRAY_BIT
- GL_LINE_STRIP
- GL_LINE_BIT
- GL_TRIANGLES
- GL_TRIANGLE_STRIP
- GL_TRIANGLE_FAN
- GL_QUADS
- GL_QUAD_STRIP
- GL_POLYGON_BIT
- GL_POLYGON
- GL_POLYGON_STIPPLE_BIT
- GL_PIXEL_MODE_BIT
- GL_LIGHTING_BIT

- GL_FOG_BIT
- GL_DEPTH_BUFFER_BIT
- GL_ACCUM
- GL_LOAD
- GL_RETURN
- GL_MULT
- GL_ADD
- GL_NEVER
- GL_ACCUM_BUFFER_BIT
- GL_LESS
- GL_EQUAL
- GL_LEQUAL
- GL_GREATER
- GL_NOTEQUAL
- GL_GEQUAL
- GL_ALWAYS
- GL_SRC_COLOR
- GL_ONE_MINUS_SRC_COLOR
- GL_SRC_ALPHA
- GL_ONE_MINUS_SRC_ALPHA
- GL_DST_ALPHA
- GL_ONE_MINUS_DST_ALPHA
- GL_DST_COLOR
- GL_ONE_MINUS_DST_COLOR
- GL_SRC_ALPHA_SATURATE
- GL_STENCIL_BUFFER_BIT
- GL_FRONT_LEFT
- GL_FRONT_RIGHT
- GL_BACK_LEFT
- GL_BACK_RIGHT
- GL_FRONT
- GL_BACK
- GL_LEFT
- GL_RIGHT
- GL_FRONT_AND_BACK
- GL_AUX0

- GL_AUX1
- GL_AUX2
- GL_AUX3
- GL_INVALID_ENUM
- GL_INVALID_VALUE
- GL_INVALID_OPERATION
- GL_STACK_OVERFLOW
- GL_STACK_UNDERFLOW
- GL_OUT_OF_MEMORY
- GL_2D
- GL_3D
- GL_3D_COLOR
- GL_3D_COLOR_TEXTURE
- GL_4D_COLOR_TEXTURE
- GL_PASS_THROUGH_TOKEN
- GL_POINT_TOKEN
- GL_LINE_TOKEN
- GL_POLYGON_TOKEN
- GL_BITMAP_TOKEN
- GL_DRAW_PIXEL_TOKEN
- GL_COPY_PIXEL_TOKEN
- GL_LINE_RESET_TOKEN
- GL_EXP
- GL_VIEWPORT_BIT
- GL_EXP2
- GL_CW
- GL_CCW
- GL_COEFF
- GL_ORDER
- GL_DOMAIN
- GL_CURRENT_COLOR
- GL_CURRENT_INDEX
- GL_CURRENT_NORMAL
- GL_CURRENT_TEXTURE_COORDS
- GL_CURRENT_RASTER_COLOR
- GL_CURRENT_RASTER_INDEX

- GL_CURRENT_RASTER_TEXTURE_COORDS
- GL_CURRENT_RASTER_POSITION
- GL_CURRENT_RASTER_POSITION_VALID
- GL_CURRENT_RASTER_DISTANCE
- GL_POINT_SMOOTH
- GL_POINT_SIZE
- GL_POINT_SIZE_RANGE
- GL_POINT_SIZE_GRANULARITY
- GL_LINE_SMOOTH
- GL_LINE_WIDTH
- GL_LINE_WIDTH_RANGE
- GL_LINE_WIDTH_GRANULARITY
- GL_LINE_STIPPLE
- GL_LINE_STIPPLE_PATTERN
- GL_LINE_STIPPLE_REPEAT
- GL_LIST_MODE
- GL_MAX_LIST_NESTING
- GL_LIST_BASE
- GL_LIST_INDEX
- GL_POLYGON_MODE
- GL_POLYGON_SMOOTH
- GL_POLYGON_STIPPLE
- GL_EDGE_FLAG
- GL_CULL_FACE
- GL_CULL_FACE_MODE
- GL_FRONT_FACE
- GL_LIGHTING
- GL_LIGHT_MODEL_LOCAL_VIEWER
- GL_LIGHT_MODEL_TWO_SIDE
- GL_LIGHT_MODEL_AMBIENT
- GL_SHADE_MODEL
- GL_COLOR_MATERIAL_FACE
- GL_COLOR_MATERIAL_PARAMETER
- GL_COLOR_MATERIAL
- GL_FOG
- GL_FOG_INDEX

- GL_FOG_DENSITY
- GL_FOG_START
- GL_FOG_END
- GL_FOG_MODE
- GL_FOG_COLOR
- GL_DEPTH_RANGE
- GL_DEPTH_TEST
- GL_DEPTH_WRITEMASK
- GL_DEPTH_CLEAR_VALUE
- GL_DEPTH_FUNC
- GL_ACCUM_CLEAR_VALUE
- GL_STENCIL_TEST
- GL_STENCIL_CLEAR_VALUE
- GL_STENCIL_FUNC
- GL_STENCIL_VALUE_MASK
- GL_STENCIL_FAIL
- GL_STENCIL_PASS_DEPTH_FAIL
- GL_STENCIL_PASS_DEPTH_PASS
- GL_STENCIL_REF
- GL_STENCIL_WRITEMASK
- GL_MATRIX_MODE
- GL_NORMALIZE
- GL_VIEWPORT
- GL_MODELVIEW_STACK_DEPTH
- GL_PROJECTION_STACK_DEPTH
- GL_TEXTURE_STACK_DEPTH
- GL_MODELVIEW_MATRIX
- GL_PROJECTION_MATRIX
- GL_TEXTURE_MATRIX
- GL_ATTRIB_STACK_DEPTH
- GL_CLIENT_ATTRIB_STACK_DEPTH
- GL_ALPHA_TEST
- GL_ALPHA_TEST_FUNC
- GL_ALPHA_TEST_REF
- GL_DITHER
- GL_BLEND_DST

- GL_BLEND_SRC
- GL_BLEND
- GL_LOGIC_OP_MODE
- GL_INDEX_LOGIC_OP
- GL_COLOR_LOGIC_OP
- GL_AUX_BUFFERS
- GL_DRAW_BUFFER
- GL_READ_BUFFER
- GL_SCISSOR_BOX
- GL_SCISSOR_TEST
- GL_INDEX_CLEAR_VALUE
- GL_INDEX_WRITEMASK
- GL_COLOR_CLEAR_VALUE
- GL_COLOR_WRITEMASK
- GL_INDEX_MODE
- GL_RGBA_MODE
- GL_DOUBLEBUFFER
- GL_STEREO
- GL_RENDER_MODE
- GL_PERSPECTIVE_CORRECTION_HINT
- GL_POINT_SMOOTH_HINT
- GL_LINE_SMOOTH_HINT
- GL_POLYGON_SMOOTH_HINT
- GL_FOG_HINT
- GL_TEXTURE_GEN_S
- GL_TEXTURE_GEN_T
- GL_TEXTURE_GEN_R
- GL_TEXTURE_GEN_Q
- GL_PIXEL_MAP_I_TO_I
- GL_PIXEL_MAP_S_TO_S
- GL_PIXEL_MAP_I_TO_R
- GL_PIXEL_MAP_I_TO_G
- GL_PIXEL_MAP_I_TO_B
- GL_PIXEL_MAP_I_TO_A
- GL_PIXEL_MAP_R_TO_R
- GL_PIXEL_MAP_G_TO_G

- GL_PIXEL_MAP_B_TO_B
- GL_PIXEL_MAP_A_TO_A
- GL_PIXEL_MAP_I_TO_I_SIZE
- GL_PIXEL_MAP_S_TO_S_SIZE
- GL_PIXEL_MAP_I_TO_R_SIZE
- GL_PIXEL_MAP_I_TO_G_SIZE
- GL_PIXEL_MAP_I_TO_B_SIZE
- GL_PIXEL_MAP_I_TO_A_SIZE
- GL_PIXEL_MAP_R_TO_R_SIZE
- GL_PIXEL_MAP_G_TO_G_SIZE
- GL_PIXEL_MAP_B_TO_B_SIZE
- GL_PIXEL_MAP_A_TO_A_SIZE
- GL_UNPACK_SWAP_BYTES
- GL_UNPACK_LSB_FIRST
- GL_UNPACK_ROW_LENGTH
- GL_UNPACK_SKIP_ROWS
- GL_UNPACK_SKIP_PIXELS
- GL_UNPACK_ALIGNMENT
- GL_PACK_SWAP_BYTES
- GL_PACK_LSB_FIRST
- GL_PACK_ROW_LENGTH
- GL_PACK_SKIP_ROWS
- GL_PACK_SKIP_PIXELS
- GL_PACK_ALIGNMENT
- GL_MAP_COLOR
- GL_MAP_STENCIL
- GL_INDEX_SHIFT
- GL_INDEX_OFFSET
- GL_RED_SCALE
- GL_RED_BIAS
- GL_ZOOM_X
- GL_ZOOM_Y
- GL_GREEN_SCALE
- GL_GREEN_BIAS
- GL_BLUE_SCALE
- GL_BLUE_BIAS

- GL_ALPHA_SCALE
- GL_ALPHA_BIAS
- GL_DEPTH_SCALE
- GL_DEPTH_BIAS
- GL_MAX_EVAL_ORDER
- GL_MAX_LIGHTS
- GL_MAX_CLIP_PLANES
- GL_MAX_TEXTURE_SIZE
- GL_MAX_PIXEL_MAP_TABLE
- GL_MAX_ATTRIB_STACK_DEPTH
- GL_MAX_MODELVIEW_STACK_DEPTH
- GL_MAX_NAME_STACK_DEPTH
- GL_MAX_PROJECTION_STACK_DEPTH
- GL_MAX_TEXTURE_STACK_DEPTH
- GL_MAX_VIEWPORT_DIMS
- GL_MAX_CLIENT_ATTRIB_STACK_DEPTH
- GL_SUBPIXEL_BITS
- GL_INDEX_BITS
- GL_RED_BITS
- GL_GREEN_BITS
- GL_BLUE_BITS
- GL_ALPHA_BITS
- GL_DEPTH_BITS
- GL_STENCIL_BITS
- GL_ACCUM_RED_BITS
- GL_ACCUM_GREEN_BITS
- GL_ACCUM_BLUE_BITS
- GL_ACCUM_ALPHA_BITS
- GL_NAME_STACK_DEPTH
- GL_AUTO_NORMAL
- GL_MAP1_COLOR_4
- GL_MAP1_INDEX
- GL_MAP1_NORMAL
- GL_MAP1_TEXTURE_COORD_1
- GL_MAP1_TEXTURE_COORD_2
- GL_MAP1_TEXTURE_COORD_3

- GL_MAP1_TEXTURE_COORD_4
- GL_MAP1_VERTEX_3
- GL_MAP1_VERTEX_4
- GL_MAP2_COLOR_4
- GL_MAP2_INDEX
- GL_MAP2_NORMAL
- GL_MAP2_TEXTURE_COORD_1
- GL_MAP2_TEXTURE_COORD_2
- GL_MAP2_TEXTURE_COORD_3
- GL_MAP2_TEXTURE_COORD_4
- GL_MAP2_VERTEX_3
- GL_MAP2_VERTEX_4
- GL_MAP1_GRID_DOMAIN
- GL_MAP1_GRID_SEGMENTS
- GL_MAP2_GRID_DOMAIN
- GL_MAP2_GRID_SEGMENTS
- GL_TEXTURE_1D
- GL_TEXTURE_2D
- GL_FEEDBACK_BUFFER_POINTER
- GL_FEEDBACK_BUFFER_SIZE
- GL_FEEDBACK_BUFFER_TYPE
- GL_SELECTION_BUFFER_POINTER
- GL_SELECTION_BUFFER_SIZE
- GL_TEXTURE_WIDTH
- GL_TRANSFORM_BIT
- GL_TEXTURE_HEIGHT
- GL_TEXTURE_INTERNAL_FORMAT
- GL_TEXTURE_BORDER_COLOR
- GL_TEXTURE_BORDER
- GL_DONT_CARE
- GL_FASTEST
- GL_NICEST
- GL_AMBIENT
- GL_DIFFUSE
- GL_SPECULAR
- GL_POSITION

- GL_SPOT_DIRECTION
- GL_SPOT_EXPONENT
- GL_SPOT_CUTOFF
- GL_CONSTANT_ATTENUATION
- GL_LINEAR_ATTENUATION
- GL_QUADRATIC_ATTENUATION
- GL_COMPILE
- GL_COMPILE_AND_EXECUTE
- GL_BYTE
- GL_UNSIGNED_BYTE
- GL_SHORT
- GL_UNSIGNED_SHORT
- GL_INT
- GL_UNSIGNED_INT
- GL_FLOAT
- GL_2_BYTES
- GL_3_BYTES
- GL_4_BYTES
- GL_DOUBLE
- GL_CLEAR
- GL_AND
- GL_AND_REVERSE
- GL_COPY
- GL_AND_INVERTED
- GL_NOOP
- GL_XOR
- GL_OR
- GL_NOR
- GL_EQUIV
- GL_INVERT
- GL_OR_REVERSE
- GL_COPY_INVERTED
- GL_OR_INVERTED
- GL_NAND
- GL_SET
- GL_EMISSION

- GL_SHININESS
- GL_AMBIENT_AND_DIFFUSE
- GL_COLOR_INDEXES
- GL_MODELVIEW
- GL_PROJECTION
- GL_TEXTURE
- GL_COLOR
- GL_DEPTH
- GL_STENCIL
- GL_COLOR_INDEX
- GL_STENCIL_INDEX
- GL_DEPTH_COMPONENT
- GL_RED
- GL_GREEN
- GL_BLUE
- GL_ALPHA
- GL_RGB
- GL_RGBA
- GL_LUMINANCE
- GL_LUMINANCE_ALPHA
- GL_BITMAP
- GL_POINT
- GL_LINE
- GL_FILL
- GL_RENDER
- GL_FEEDBACK
- GL_SELECT
- GL_FLAT
- GL_SMOOTH
- GL_KEEP
- GL_REPLACE
- GL_INCR
- GL_DECR
- GL_VENDOR
- GL_RENDERER
- GL_VERSION

- GL_EXTENSIONS
- GL_S
- GL_ENABLE_BIT
- GL_T
- GL_R
- GL_Q
- GL_MODULATE
- GL_DECAL
- GL_TEXTURE_ENV_MODE
- GL_TEXTURE_ENV_COLOR
- GL_TEXTURE_ENV
- GL_EYE_LINEAR
- GL_OBJECT_LINEAR
- GL_SPHERE_MAP
- GL_TEXTURE_GEN_MODE
- GL_OBJECT_PLANE
- GL_EYE_PLANE
- GL_NEAREST
- GL_LINEAR
- GL_NEAREST_MIPMAP_NEAREST
- GL_LINEAR_MIPMAP_NEAREST
- GL_NEAREST_MIPMAP_LINEAR
- GL_LINEAR_MIPMAP_LINEAR
- GL_TEXTURE_MAG_FILTER
- GL_TEXTURE_MIN_FILTER
- GL_TEXTURE_WRAP_S
- GL_TEXTURE_WRAP_T
- GL_CLAMP
- GL_REPEAT
- GL_POLYGON_OFFSET_UNITS
- GL_POLYGON_OFFSET_POINT
- GL_POLYGON_OFFSET_LINE
- GL_R3_G3_B2
- GL_V2F
- GL_V3F
- GL_C4UB_V2F

- GL_C4UB_V3F
- GL_C3F_V3F
- GL_N3F_V3F
- GL_C4F_N3F_V3F
- GL_T2F_V3F
- GL_T4F_V4F
- GL_T2F_C4UB_V3F
- GL_T2F_C3F_V3F
- GL_T2F_N3F_V3F
- GL_T2F_C4F_N3F_V3F
- GL_T4F_C4F_N3F_V4F
- GL_CLIP_PLANE0
- GL_CLIP_PLANE1
- GL_CLIP_PLANE2
- GL_CLIP_PLANE3
- GL_CLIP_PLANE4
- GL_CLIP_PLANE5
- GL_LIGHT0
- GL_COLOR_BUFFER_BIT
- GL_LIGHT1
- GL_LIGHT2
- GL_LIGHT3
- GL_LIGHT4
- GL_LIGHT5
- GL_LIGHT6
- GL_LIGHT7
- GL_HINT_BIT
- GL_POLYGON_OFFSET_FILL
- GL_POLYGON_OFFSET_FACTOR
- GL_ALPHA4
- GL_ALPHA8
- GL_ALPHA12
- GL_ALPHA16
- GL_LUMINANCE4
- GL_LUMINANCE8
- GL_LUMINANCE12

- GL_LUMINANCE16
- GL_LUMINANCE4_ALPHA4
- GL_LUMINANCE6_ALPHA2
- GL_LUMINANCE8_ALPHA8
- GL_LUMINANCE12_ALPHA4
- GL_LUMINANCE12_ALPHA12
- GL_LUMINANCE16_ALPHA16
- GL_INTENSITY
- GL_INTENSITY4
- GL_INTENSITY8
- GL_INTENSITY12
- GL_INTENSITY16
- GL_RGB4
- GL_RGB5
- GL_RGB8
- GL_RGB10
- GL_RGB12
- GL_RGB16
- GL_RGBA2
- GL_RGBA4
- GL_RGB5_A1
- GL_RGBA8
- GL_RGB10_A2
- GL_RGBA12
- GL_RGBA16
- GL_TEXTURE_RED_SIZE
- GL_TEXTURE_GREEN_SIZE
- GL_TEXTURE_BLUE_SIZE
- GL_TEXTURE_ALPHA_SIZE
- GL_TEXTURE_LUMINANCE_SIZE
- GL_TEXTURE_INTENSITY_SIZE
- GL_PROXY_TEXTURE_1D
- GL_PROXY_TEXTURE_2D
- GL_TEXTURE_PRIORITY
- GL_TEXTURE_RESIDENT
- GL_TEXTURE_BINDING_1D

- GL_TEXTURE_BINDING_2D
- GL_VERTEX_ARRAY
- GL_NORMAL_ARRAY
- GL_COLOR_ARRAY
- GL_INDEX_ARRAY
- GL_TEXTURE_COORD_ARRAY
- GL_EDGE_FLAG_ARRAY
- GL_VERTEX_ARRAY_SIZE
- GL_VERTEX_ARRAY_TYPE
- GL_VERTEX_ARRAY_STRIDE
- GL_NORMAL_ARRAY_TYPE
- GL_NORMAL_ARRAY_STRIDE
- GL_COLOR_ARRAY_SIZE
- GL_COLOR_ARRAY_TYPE
- GL_COLOR_ARRAY_STRIDE
- GL_INDEX_ARRAY_TYPE
- GL_INDEX_ARRAY_STRIDE
- GL_TEXTURE_COORD_ARRAY_SIZE
- GL_TEXTURE_COORD_ARRAY_TYPE
- GL_TEXTURE_COORD_ARRAY_STRIDE
- GL_EDGE_FLAG_ARRAY_STRIDE
- GL_VERTEX_ARRAY_POINTER
- GL_NORMAL_ARRAY_POINTER
- GL_COLOR_ARRAY_POINTER
- GL_INDEX_ARRAY_POINTER
- GL_TEXTURE_COORD_ARRAY_POINTER
- GL_EDGE_FLAG_ARRAY_POINTER
- GL_COLOR_INDEX1_EXT
- GL_COLOR_INDEX2_EXT
- GL_COLOR_INDEX4_EXT
- GL_COLOR_INDEX8_EXT
- GL_COLOR_INDEX12_EXT
- GL_COLOR_INDEX16_EXT
- GL_EVAL_BIT
- GL_LIST_BIT
- GL_TEXTURE_BIT

- GL_SCISSOR_BIT
- GL_ALL_ATTRIB_BITS
- GL_CLIENT_ALL_ATTRIB_BITS
- GL_SMOOTH_POINT_SIZE_RANGE
- GL_SMOOTH_POINT_SIZE_GRANULARITY
- GL_SMOOTH_LINE_WIDTH_RANGE
- GL_SMOOTH_LINE_WIDTH_GRANULARITY
- GL_UNSIGNED_BYTE_3_3_2
- GL_UNSIGNED_SHORT_4_4_4_4
- GL_UNSIGNED_SHORT_5_5_5_1
- GL_UNSIGNED_INT_8_8_8_8
- GL_UNSIGNED_INT_10_10_10_2
- GL_RESCALE_NORMAL
- GL_TEXTURE_BINDING_3D
- GL_PACK_SKIP_IMAGES
- GL_PACK_IMAGE_HEIGHT
- GL_UNPACK_SKIP_IMAGES
- GL_UNPACK_IMAGE_HEIGHT
- GL_TEXTURE_3D
- GL_PROXY_TEXTURE_3D
- GL_TEXTURE_DEPTH
- GL_TEXTURE_WRAP_R
- GL_MAX_3D_TEXTURE_SIZE
- GL_BGR
- GL_BGRA
- GL_MAX_ELEMENTS_VERTICES
- GL_MAX_ELEMENTS_INDICES
- GL_CLAMP_TO_EDGE
- GL_TEXTURE_MIN_LOD
- GL_TEXTURE_MAX_LOD
- GL_TEXTURE_BASE_LEVEL
- GL_TEXTURE_MAX_LEVEL
- GL_LIGHT_MODEL_COLOR_CONTROL
- GL_SINGLE_COLOR
- GL_SEPARATE_SPECULAR_COLOR
- GL_UNSIGNED_BYTE_2_3_3_REV

- GL_UNSIGNED_SHORT_5_6_5
- GL_UNSIGNED_SHORT_5_6_5_REV
- GL_UNSIGNED_SHORT_4_4_4_REV
- GL_UNSIGNED_SHORT_1_5_5_5_REV
- GL_UNSIGNED_INT_8_8_8_8_REV
- GL_ALIASED_POINT_SIZE_RANGE
- GL_ALIASED_LINE_WIDTH_RANGE
- GL_MULTISAMPLE
- GL_SAMPLE_ALPHA_TO_COVERAGE
- GL_SAMPLE_ALPHA_TO_ONE
- GL_SAMPLE_COVERAGE
- GL_SAMPLE_BUFFERS
- GL_SAMPLES
- GL_SAMPLE_COVERAGE_VALUE
- GL_SAMPLE_COVERAGE_INVERT
- GL_CLAMP_TO_BORDER
- GL_TEXTURE0
- GL_TEXTURE1
- GL_TEXTURE2
- GL_TEXTURE3
- GL_TEXTURE4
- GL_TEXTURE5
- GL_TEXTURE6
- GL_TEXTURE7
- GL_TEXTURE8
- GL_TEXTURE9
- GL_TEXTURE10
- GL_TEXTURE11
- GL_TEXTURE12
- GL_TEXTURE13
- GL_TEXTURE14
- GL_TEXTURE15
- GL_TEXTURE16
- GL_TEXTURE17
- GL_TEXTURE18
- GL_TEXTURE19

- GL_TEXTURE20
- GL_TEXTURE21
- GL_TEXTURE22
- GL_TEXTURE23
- GL_TEXTURE24
- GL_TEXTURE25
- GL_TEXTURE26
- GL_TEXTURE27
- GL_TEXTURE28
- GL_TEXTURE29
- GL_TEXTURE30
- GL_TEXTURE31
- GL_ACTIVE_TEXTURE
- GL_CLIENT_ACTIVE_TEXTURE
- GL_MAX_TEXTURE_UNITS
- GL_TRANSPOSE_MODELVIEW_MATRIX
- GL_TRANSPOSE_PROJECTION_MATRIX
- GL_TRANSPOSE_TEXTURE_MATRIX
- GL_TRANSPOSE_COLOR_MATRIX
- GL_SUBTRACT
- GL_COMPRESSED_ALPHA
- GL_COMPRESSED_LUMINANCE
- GL_COMPRESSED_LUMINANCE_ALPHA
- GL_COMPRESSED_INTENSITY
- GL_COMPRESSED_RGB
- GL_COMPRESSED_RGBA
- GL_TEXTURE_COMPRESSION_HINT
- GL_NORMAL_MAP
- GL_REFLECTION_MAP
- GL_TEXTURE_CUBE_MAP
- GL_TEXTURE_BINDING_CUBE_MAP
- GL_TEXTURE_CUBE_MAP_POSITIVE_X
- GL_TEXTURE_CUBE_MAP_NEGATIVE_X
- GL_TEXTURE_CUBE_MAP_POSITIVE_Y
- GL_TEXTURE_CUBE_MAP_NEGATIVE_Y
- GL_TEXTURE_CUBE_MAP_POSITIVE_Z

- GL_TEXTURE_CUBE_MAP_NEGATIVE_Z
- GL_PROXY_TEXTURE_CUBE_MAP
- GL_MAX_CUBE_MAP_TEXTURE_SIZE
- GL_COMBINE
- GL_COMBINE_RGB
- GL_COMBINE_ALPHA
- GL_RGB_SCALE
- GL_ADD_SIGNED
- GL_INTERPOLATE
- GL_CONSTANT
- GL_PRIMARY_COLOR
- GL_PREVIOUS
- GL_SOURCE0_RGB
- GL_SOURCE1_RGB
- GL_SOURCE2_RGB
- GL_SOURCE0_ALPHA
- GL_SOURCE1_ALPHA
- GL_SOURCE2_ALPHA
- GL_OPERAND0_RGB
- GL_OPERAND1_RGB
- GL_OPERAND2_RGB
- GL_OPERAND0_ALPHA
- GL_OPERAND1_ALPHA
- GL_OPERAND2_ALPHA
- GL_TEXTURE_COMPRESSED_IMAGE_SIZE
- GL_TEXTURE_COMPRESSED
- GL_NUM_COMPRESSED_TEXTURE_FORMATS
- GL_COMPRESSED_TEXTURE_FORMATS
- GL_DOT3_RGB
- GL_DOT3_RGBA
- GL_MULTISAMPLE_BIT
- GL_BLEND_DST_RGB
- GL_BLEND_SRC_RGB
- GL_BLEND_DST_ALPHA
- GL_BLEND_SRC_ALPHA
- GL_POINT_SIZE_MIN

- GL_POINT_SIZE_MAX
- GL_POINT_FADE_THRESHOLD_SIZE
- GL_POINT_DISTANCE_ATTENUATION
- GL_GENERATE_MIPMAP
- GL_GENERATE_MIPMAP_HINT
- GL_DEPTH_COMPONENT16
- GL_DEPTH_COMPONENT24
- GL_DEPTH_COMPONENT32
- GL_MIRRORED_REPEAT
- GL_FOG_COORDINATE_SOURCE
- GL_FOG_COORDINATE
- GL_FRAGMENT_DEPTH
- GL_CURRENT_FOG_COORDINATE
- GL_FOG_COORDINATE_ARRAY_TYPE
- GL_FOG_COORDINATE_ARRAY_STRIDE
- GL_FOG_COORDINATE_ARRAY_POINTER
- GL_FOG_COORDINATE_ARRAY
- GL_COLOR_SUM
- GL_CURRENT_SECONDARY_COLOR
- GL_SECONDARY_COLOR_ARRAY_SIZE
- GL_SECONDARY_COLOR_ARRAY_TYPE
- GL_SECONDARY_COLOR_ARRAY_STRIDE
- GL_SECONDARY_COLOR_ARRAY_POINTER
- GL_SECONDARY_COLOR_ARRAY
- GL_MAX_TEXTURE_LOD_BIAS
- GL_TEXTURE_FILTER_CONTROL
- GL_TEXTURE_LOD_BIAS
- GL_INCR_WRAP
- GL_DECR_WRAP
- GL_TEXTURE_DEPTH_SIZE
- GL_DEPTH_TEXTURE_MODE
- GL_TEXTURE_COMPARE_MODE
- GL_TEXTURE_COMPARE_FUNC
- GL_COMPARE_R_TO_TEXTURE
- void glAccum(GLenum op, GLfloat value)
- void glAlphaFunc(GLenum func, GLclampf ref)

- GLboolean glAreTexturesResident(GLsizei n, const GLuint *textures, GLboolean *residences)
- void glArrayElement(GLint i)
- void glBegin(GLenum mode)
- void glBindTexture(GLenum target, GLuint texture)
- void glBitmap(GLsizei width, GLsizei height, GLfloat xorig, GLfloat yorig, GLfloat xmove, GLfloat ymove, const GLubyte *bitmap)
- void glBlendFunc(GLenum sfactor, GLenum dfactor)
- void glCallList(GLuint list)
- void glCallLists(GLsizei n, GLenum type, const void *lists)
- void glClear(GLbitfield mask)
- void glClearAccum(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glClearColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)
- void glClearDepth(GLclampd depth)
- void glClearIndex(GLfloat c)
- void glClearStencil(GLint s)
- void glClipPlane(GLenum plane, const GLdouble *equation)
- void glColor3b(GLbyte red, GLbyte green, GLbyte blue)
- void glColor3bv(const GLbyte *v)
- void glColor3d(GLdouble red, GLdouble green, GLdouble blue)
- void glColor3dv(const GLdouble *v)
- void glColor3f(GLfloat red, GLfloat green, GLfloat blue)
- void glColor3fv(const GLfloat *v)
- void glColor3i(GLint red, GLint green, GLint blue)
- void glColor3iv(const GLint *v)
- void glColor3s(GLshort red, GLshort green, GLshort blue)
- void glColor3sv(const GLshort *v)
- void glColor3ub(GLubyte red, GLubyte green, GLubyte blue)
- void glColor3ubv(const GLubyte *v)
- void glColor3ui(GLuint red, GLuint green, GLuint blue)
- void glColor3uiv(const GLuint *v)
- void glColor3us(GLushort red, GLushort green, GLushort blue)
- void glColor3usv(const GLushort *v)
- void glColor4b(GLbyte red, GLbyte green, GLbyte blue, GLbyte alpha)
- void glColor4bv(const GLbyte *v)
- void glColor4d(GLdouble red, GLdouble green, GLdouble blue, GLdouble alpha)
- void glColor4dv(const GLdouble *v)

- void glColor4f(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glColor4fv(const GLfloat *v)
- void glColor4i(GLint red, GLint green, GLint blue, GLint alpha)
- void glColor4iv(const GLint *v)
- void glColor4s(GLshort red, GLshort green, GLshort blue, GLshort alpha)
- void glColor4sv(const GLshort *v)
- void glColor4ub(GLubyte red, GLubyte green, GLubyte blue, GLubyte alpha)
- void glColor4ubv(const GLubyte *v)
- void glColor4ui(GLuint red, GLuint green, GLuint blue, GLuint alpha)
- void glColor4uiv(const GLuint *v)
- void glColor4us(GLushort red, GLushort green, GLushort blue, GLushort alpha)
- void glColor4usv(const GLushort *v)
- void glColorMask(GLboolean red, GLboolean green, GLboolean blue, GLboolean alpha)
- void glColorMaterial(GLenum face, GLenum mode)
- void glColorPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)
- void glCopyPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum type)
- void glCopyTexImage1D(GLenum target, GLint level, GLenum internalFormat, GLint x, GLint y, GLsizei width, GLint border)
- void glCopyTexImage2D(GLenum target, GLint level, GLenum internalFormat, GLint x, GLint y, GLsizei width, GLsizei height, GLint border)
- void glCopyTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLint x, GLint y, GLsizei width)
- void glCopyTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint x, GLint y, GLsizei width, GLsizei height)
- void glCullFace(GLenum mode)
- void glDeleteLists(GLuint list, GLsizei range)
- void glDeleteTextures(GLsizei n, const GLuint *textures)
- void glDepthFunc(GLenum func)
- void glDepthMask(GLboolean flag)
- void glDepthRange(GLclampd zNear, GLclampd zFar)
- void glDisable(GLenum cap)
- void glDisableClientState(GLenum array)
- void glDrawArrays(GLenum mode, GLint first, GLsizei count)
- void glDrawBuffer(GLenum mode)
- void glDrawElements(GLenum mode, GLsizei count, GLenum type, const void *indices)
- void glDrawPixels(GLsizei width, GLsizei height, GLenum format, GLenum type, const void *pixels)
- void glEdgeFlag(GLboolean flag)
- void glEdgeFlagPointer(GLsizei stride, const void *pointer)

- void glEdgeFlagv(const GLboolean *flag)
- void glEnable(GLenum cap)
- void glEnableClientState(GLenum array)
- void glEnd(void)
- void glEndList(void)
- void glEvalCoord1d(GLdouble u)
- void glEvalCoord1dv(const GLdouble *u)
- void glEvalCoord1f(GLfloat u)
- void glEvalCoord1fv(const GLfloat *u)
- void glEvalCoord2d(GLdouble u, GLdouble v)
- void glEvalCoord2dv(const GLdouble *u)
- void glEvalCoord2f(GLfloat u, GLfloat v)
- void glEvalCoord2fv(const GLfloat *u)
- void glEvalMesh1(GLenum mode, GLint i1, GLint i2)
- void glEvalMesh2(GLenum mode, GLint i1, GLint i2, GLint j1, GLint j2)
- void glEvalPoint1(GLint i)
- void glEvalPoint2(GLint i, GLint j)
- void glFeedbackBuffer(GLsizei size, GLenum type, GLfloat *buffer)
- void glFinish(void)
- void glFlush(void)
- void glFogf(GLenum pname, GLfloat param)
- void glFogfv(GLenum pname, const GLfloat *params)
- void glFogi(GLenum pname, GLint param)
- void glFogiv(GLenum pname, const GLint *params)
- void glFrontFace(GLenum mode)
- void glFrustum(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble zNear, GLdouble zFar)
- GLuint glGenLists(GLsizei range)
- void glGenTextures(GLsizei n, GLuint *textures)
- void glGetBooleanv(GLenum pname, GLboolean *params)
- void glGetClipPlane(GLenum plane, GLdouble *equation)
- void glGetDoublev(GLenum pname, GLdouble *params)
- GLenum glGetError(void)
- void glGetFloatv(GLenum pname, GLfloat *params)
- void glGetIntegerv(GLenum pname, GLint *params)
- void glGetLightfv(GLenum light, GLenum pname, GLfloat *params)

- void glGetLightiv(GLenum light, GLenum pname, GLint *params)
- void glGetMapdv(GLenum target, GLenum query, GLdouble *v)
- void glGetMapfv(GLenum target, GLenum query, GLfloat *v)
- void glGetMapiv(GLenum target, GLenum query, GLint *v)
- void glGetMaterialfv(GLenum face, GLenum pname, GLfloat *params)
- void glGetMaterialiv(GLenum face, GLenum pname, GLint *params)
- void glGetPixelMapfv(GLenum map, GLfloat *values)
- void glGetPixelMapuiv(GLenum map, GLuint *values)
- void glGetPixelMapusv(GLenum map, GLushort *values)
- void glGetPointerv(GLenum pname, void* *params)
- void glGetPolygonStipple(GLubyte *mask)
- GLubyte * glGetString(GLenum name)
- void glGetTexEnvfv(GLenum target, GLenum pname, GLfloat *params)
- void glGetTexEnviv(GLenum target, GLenum pname, GLint *params)
- void glGetTexGendv(GLenum coord, GLenum pname, GLdouble *params)
- void glGetTexGenfv(GLenum coord, GLenum pname, GLfloat *params)
- void glGetTexGeniv(GLenum coord, GLenum pname, GLint *params)
- void glGetTexImage(GLenum target, GLint level, GLenum format, GLenum type, void *pixels)
- void glGetTexLevelParameterfv(GLenum target, GLint level, GLenum pname, GLfloat *params)
- void glGetTexLevelParameteriv(GLenum target, GLint level, GLenum pname, GLint *params)
- void glGetTexParameterfv(GLenum target, GLenum pname, GLfloat *params)
- void glGetTexParameteriv(GLenum target, GLenum pname, GLint *params)
- void glHint(GLenum target, GLenum mode)
- void glIndexMask(GLuint mask)
- void glIndexPointer(GLenum type, GLsizei stride, const void *pointer)
- void glIndexd(GLdouble c)
- void glIndexdv(const GLdouble *c)
- void glIndexf(GLfloat c)
- void glIndexfv(const GLfloat *c)
- void glIndexi(GLint c)
- void glIndexiv(const GLint *c)
- void glIndexs(GLshort c)
- void glIndexsv(const GLshort *c)
- void glIndexub(GLubyte c)
- void glIndexubv(const GLubyte *c)
- void glInitNames(void)

- void glInterleavedArrays(GLenum format, GLsizei stride, const void *pointer)
- GLboolean glIsEnabled(GLenum cap)
- GLboolean glIsList(GLuint list)
- GLboolean glIsTexture(GLuint texture)
- void glLightModelf(GLenum pname, GLfloat param)
- void glLightModelfv(GLenum pname, const GLfloat *params)
- void glLightModeli(GLenum pname, GLint param)
- void glLightModeliv(GLenum pname, const GLint *params)
- void glLightf(GLenum light, GLenum pname, GLfloat param)
- void glLightfv(GLenum light, GLenum pname, const GLfloat *params)
- void glLighti(GLenum light, GLenum pname, GLint param)
- void glLightiv(GLenum light, GLenum pname, const GLint *params)
- void glLineStipple(GLint factor, GLushort pattern)
- void glLineWidth(GLfloat width)
- void glListBase(GLuint base)
- void glLoadIdentity(void)
- void glLoadMatrixd(const GLdouble *m)
- void glLoadMatrixf(const GLfloat *m)
- void glLoadName(GLuint name)
- void glLogicOp(GLenum opcode)
- void glMap1d(GLenum target, GLdouble u1, GLdouble u2, GLint stride, GLint order, const GLdouble *points)
- void glMap1f(GLenum target, GLfloat u1, GLfloat u2, GLint stride, GLint order, const GLfloat *points)
- void glMap2d(GLenum target, GLdouble u1, GLdouble u2, GLint ustride, GLint uorder, GLdouble v1, GLdouble v2, GLint vstride, GLint vorder, const GLdouble *points)
- void glMap2f(GLenum target, GLfloat u1, GLfloat u2, GLint ustride, GLint uorder, GLfloat v1, GLfloat v2, GLint vstride, GLint vorder, const GLfloat *points)
- void glMapGrid1d(GLint un, GLdouble u1, GLdouble u2)
- void glMapGrid1f(GLint un, GLfloat u1, GLfloat u2)
- void glMapGrid2d(GLint un, GLdouble u1, GLdouble u2, GLint vn, GLdouble v1, GLdouble v2)
- void glMapGrid2f(GLint un, GLfloat u1, GLfloat u2, GLint vn, GLfloat v1, GLfloat v2)
- void glMaterialf(GLenum face, GLenum pname, GLfloat param)
- void glMaterialfv(GLenum face, GLenum pname, const GLfloat *params)
- void glMateriali(GLenum face, GLenum pname, GLint param)
- void glMaterialiv(GLenum face, GLenum pname, const GLint *params)
- void glMatrixMode(GLenum mode)
- void glMultMatrixd(const GLdouble *m)
- void glMultMatrixf(const GLfloat *m)

- void glNewList(GLuint list, GLenum mode)
- void glNormal3b(GLbyte nx, GLbyte ny, GLbyte nz)
- void glNormal3bv(const GLbyte *v)
- void glNormal3d(GLdouble nx, GLdouble ny, GLdouble nz)
- void glNormal3dv(const GLdouble *v)
- void glNormal3f(GLfloat nx, GLfloat ny, GLfloat nz)
- void glNormal3fv(const GLfloat *v)
- void glNormal3i(GLint nx, GLint ny, GLint nz)
- void glNormal3iv(const GLint *v)
- void glNormal3s(GLshort nx, GLshort ny, GLshort nz)
- void glNormal3sv(const GLshort *v)
- void glNormalPointer(GLenum type, GLsizei stride, const void *pointer)
- void glOrtho(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble zNear, GLdouble zFar)
- void glPassThrough(GLfloat token)
- void glPixelMapfv(GLenum map, GLsizei mapsize, const GLfloat *values)
- void glPixelMapuiv(GLenum map, GLsizei mapsize, const GLuint *values)
- void glPixelMapusv(GLenum map, GLsizei mapsize, const GLushort *values)
- void glPixelStoref(GLenum pname, GLfloat param)
- void glPixelStorei(GLenum pname, GLint param)
- void glPixelTransferf(GLenum pname, GLfloat param)
- void glPixelTransferi(GLenum pname, GLint param)
- void glPixelZoom(GLfloat xfactor, GLfloat yfactor)
- void glPointSize(GLfloat size)
- void glPolygonMode(GLenum face, GLenum mode)
- void glPolygonOffset(GLfloat factor, GLfloat units)
- void glPolygonStipple(const GLubyte *mask)
- void glPopAttrib(void)
- void glPopClientAttrib(void)
- void glPopMatrix(void)
- void glPopName(void)
- void glPrioritizeTextures(GLsizei n, const GLuint *textures, const GLclampf *priorities)
- void glPushAttrib(GLbitfield mask)
- void glPushClientAttrib(GLbitfield mask)
- void glPushMatrix(void)
- void glPushName(GLuint name)

- `void glRasterPos2d(GLdouble x, GLdouble y)`
- `void glRasterPos2dv(const GLdouble *v)`
- `void glRasterPos2f(GLfloat x, GLfloat y)`
- `void glRasterPos2fv(const GLfloat *v)`
- `void glRasterPos2i(GLint x, GLint y)`
- `void glRasterPos2iv(const GLint *v)`
- `void glRasterPos2s(GLshort x, GLshort y)`
- `void glRasterPos2sv(const GLshort *v)`
- `void glRasterPos3d(GLdouble x, GLdouble y, GLdouble z)`
- `void glRasterPos3dv(const GLdouble *v)`
- `void glRasterPos3f(GLfloat x, GLfloat y, GLfloat z)`
- `void glRasterPos3fv(const GLfloat *v)`
- `void glRasterPos3i(GLint x, GLint y, GLint z)`
- `void glRasterPos3iv(const GLint *v)`
- `void glRasterPos3s(GLshort x, GLshort y, GLshort z)`
- `void glRasterPos3sv(const GLshort *v)`
- `void glRasterPos4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)`
- `void glRasterPos4dv(const GLdouble *v)`
- `void glRasterPos4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)`
- `void glRasterPos4fv(const GLfloat *v)`
- `void glRasterPos4i(GLint x, GLint y, GLint z, GLint w)`
- `void glRasterPos4iv(const GLint *v)`
- `void glRasterPos4s(GLshort x, GLshort y, GLshort z, GLshort w)`
- `void glRasterPos4sv(const GLshort *v)`
- `void glReadBuffer(GLenum mode)`
- `void glReadPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum format, GLenum type, void *pixels)`
- `void glRectd(GLdouble x1, GLdouble y1, GLdouble x2, GLdouble y2)`
- `void glRectdv(const GLdouble *v1, const GLdouble *v2)`
- `void glRectf(GLfloat x1, GLfloat y1, GLfloat x2, GLfloat y2)`
- `void glRectfv(const GLfloat *v1, const GLfloat *v2)`
- `void glRecti(GLint x1, GLint y1, GLint x2, GLint y2)`
- `void glRectiv(const GLint *v1, const GLint *v2)`
- `void glRects(GLshort x1, GLshort y1, GLshort x2, GLshort y2)`
- `void glRectsv(const GLshort *v1, const GLshort *v2)`
- `GLint glRenderMode(GLenum mode)`

- void glRotated(GLdouble angle, GLdouble x, GLdouble y, GLdouble z)
- void glRotatef(GLfloat angle, GLfloat x, GLfloat y, GLfloat z)
- void glScaled(GLdouble x, GLdouble y, GLdouble z)
- void glScalef(GLfloat x, GLfloat y, GLfloat z)
- void glScissor(GLint x, GLint y, GLsizei width, GLsizei height)
- void glSelectBuffer(GLsizei size, GLuint *buffer)
- void glShadeModel(GLenum mode)
- void glStencilFunc(GLenum func, GLint ref, GLuint mask)
- void glStencilMask(GLuint mask)
- void glStencilOp(GLenum fail, GLenum zfail, GLenum zpass)
- void glTexCoord1d(GLdouble s)
- void glTexCoord1dv(const GLdouble *v)
- void glTexCoord1f(GLfloat s)
- void glTexCoord1fv(const GLfloat *v)
- void glTexCoord1i(GLint s)
- void glTexCoord1iv(const GLint *v)
- void glTexCoord1s(GLshort s)
- void glTexCoord1sv(const GLshort *v)
- void glTexCoord2d(GLdouble s, GLdouble t)
- void glTexCoord2dv(const GLdouble *v)
- void glTexCoord2f(GLfloat s, GLfloat t)
- void glTexCoord2fv(const GLfloat *v)
- void glTexCoord2i(GLint s, GLint t)
- void glTexCoord2iv(const GLint *v)
- void glTexCoord2s(GLshort s, GLshort t)
- void glTexCoord2sv(const GLshort *v)
- void glTexCoord3d(GLdouble s, GLdouble t, GLdouble r)
- void glTexCoord3dv(const GLdouble *v)
- void glTexCoord3f(GLfloat s, GLfloat t, GLfloat r)
- void glTexCoord3fv(const GLfloat *v)
- void glTexCoord3i(GLint s, GLint t, GLint r)
- void glTexCoord3iv(const GLint *v)
- void glTexCoord3s(GLshort s, GLshort t, GLshort r)
- void glTexCoord3sv(const GLshort *v)
- void glTexCoord4d(GLdouble s, GLdouble t, GLdouble r, GLdouble q)
- void glTexCoord4dv(const GLdouble *v)

- `void glTexCoord4f(GLfloat s, GLfloat t, GLfloat r, GLfloat q)`
- `void glTexCoord4fv(const GLfloat *v)`
- `void glTexCoord4i(GLint s, GLint t, GLint r, GLint q)`
- `void glTexCoord4iv(const GLint *v)`
- `void glTexCoord4s(GLshort s, GLshort t, GLshort r, GLshort q)`
- `void glTexCoord4sv(const GLshort *v)`
- `void glTexCoordPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)`
- `void glTexEnvf(GLenum target, GLenum pname, GLfloat param)`
- `void glTexEnvfv(GLenum target, GLenum pname, const GLfloat *params)`
- `void glTexEnvi(GLenum target, GLenum pname, GLint param)`
- `void glTexEnviv(GLenum target, GLenum pname, const GLint *params)`
- `void glTexGend(GLenum coord, GLenum pname, GLdouble param)`
- `void glTexGendv(GLenum coord, GLenum pname, const GLdouble *params)`
- `void glTexGenf(GLenum coord, GLenum pname, GLfloat param)`
- `void glTexGenfv(GLenum coord, GLenum pname, const GLfloat *params)`
- `void glTexGeni(GLenum coord, GLenum pname, GLint param)`
- `void glTexGeniv(GLenum coord, GLenum pname, const GLint *params)`
- `void glTexImage1D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLint border, GLenum format, GLenum type, const void *pixels)`
- `void glTexImage2D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, const void *pixels)`
- `void glTexParameterf(GLenum target, GLenum pname, GLfloat param)`
- `void glTexParameterfv(GLenum target, GLenum pname, const GLfloat *params)`
- `void glTexParameteri(GLenum target, GLenum pname, GLint param)`
- `void glTexParameteriv(GLenum target, GLenum pname, const GLint *params)`
- `void glTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLenum type, const void *pixels)`
- `void glTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLenum type, const void *pixels)`
- `void glTranslated(GLdouble x, GLdouble y, GLdouble z)`
- `void glTranslatef(GLfloat x, GLfloat y, GLfloat z)`
- `void glVertex2d(GLdouble x, GLdouble y)`
- `void glVertex2dv(const GLdouble *v)`
- `void glVertex2f(GLfloat x, GLfloat y)`
- `void glVertex2fv(const GLfloat *v)`
- `void glVertex2i(GLint x, GLint y)`
- `void glVertex2iv(const GLint *v)`

- `void glVertex2s(GLshort x, GLshort y)`
- `void glVertex2sv(const GLshort *v)`
- `void glVertex3d(GLdouble x, GLdouble y, GLdouble z)`
- `void glVertex3dv(const GLdouble *v)`
- `void glVertex3f(GLfloat x, GLfloat y, GLfloat z)`
- `void glVertex3fv(const GLfloat *v)`
- `void glVertex3i(GLint x, GLint y, GLint z)`
- `void glVertex3iv(const GLint *v)`
- `void glVertex3s(GLshort x, GLshort y, GLshort z)`
- `void glVertex3sv(const GLshort *v)`
- `void glVertex4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)`
- `void glVertex4dv(const GLdouble *v)`
- `void glVertex4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)`
- `void glVertex4fv(const GLfloat *v)`
- `void glVertex4i(GLint x, GLint y, GLint z, GLint w)`
- `void glVertex4iv(const GLint *v)`
- `void glVertex4s(GLshort x, GLshort y, GLshort z, GLshort w)`
- `void glVertex4sv(const GLshort *v)`
- `void glVertexPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)`
- `void glViewport(GLint x, GLint y, GLsizei width, GLsizei height)`

RINGOPENGL (OPENGL 1.5) FUNCTIONS REFERENCE

- GL_ZERO
- GL_FALSE
- GL_LOGIC_OP
- GL_NONE
- GL_TEXTURE_COMPONENTS
- GL_NO_ERROR
- GL_POINTS
- GL_CURRENT_BIT
- GL_TRUE
- GL_ONE
- GL_CLIENT_PIXEL_STORE_BIT
- GL_LINES
- GL_LINE_LOOP
- GL_POINT_BIT
- GL_CLIENT_VERTEX_ARRAY_BIT
- GL_LINE_STRIP
- GL_LINE_BIT
- GL_TRIANGLES
- GL_TRIANGLE_STRIP
- GL_TRIANGLE_FAN
- GL_QUADS
- GL_QUAD_STRIP
- GL_POLYGON_BIT
- GL_POLYGON
- GL_POLYGON_STIPPLE_BIT
- GL_PIXEL_MODE_BIT
- GL_LIGHTING_BIT

- GL_FOG_BIT
- GL_DEPTH_BUFFER_BIT
- GL_ACCUM
- GL_LOAD
- GL_RETURN
- GL_MULT
- GL_ADD
- GL_NEVER
- GL_ACCUM_BUFFER_BIT
- GL_LESS
- GL_EQUAL
- GL_LEQUAL
- GL_GREATER
- GL_NOTEQUAL
- GL_GEQUAL
- GL_ALWAYS
- GL_SRC_COLOR
- GL_ONE_MINUS_SRC_COLOR
- GL_SRC_ALPHA
- GL_ONE_MINUS_SRC_ALPHA
- GL_DST_ALPHA
- GL_ONE_MINUS_DST_ALPHA
- GL_DST_COLOR
- GL_ONE_MINUS_DST_COLOR
- GL_SRC_ALPHA_SATURATE
- GL_STENCIL_BUFFER_BIT
- GL_FRONT_LEFT
- GL_FRONT_RIGHT
- GL_BACK_LEFT
- GL_BACK_RIGHT
- GL_FRONT
- GL_BACK
- GL_LEFT
- GL_RIGHT
- GL_FRONT_AND_BACK
- GL_AUX0

- GL_AUX1
- GL_AUX2
- GL_AUX3
- GL_INVALID_ENUM
- GL_INVALID_VALUE
- GL_INVALID_OPERATION
- GL_STACK_OVERFLOW
- GL_STACK_UNDERFLOW
- GL_OUT_OF_MEMORY
- GL_2D
- GL_3D
- GL_3D_COLOR
- GL_3D_COLOR_TEXTURE
- GL_4D_COLOR_TEXTURE
- GL_PASS_THROUGH_TOKEN
- GL_POINT_TOKEN
- GL_LINE_TOKEN
- GL_POLYGON_TOKEN
- GL_BITMAP_TOKEN
- GL_DRAW_PIXEL_TOKEN
- GL_COPY_PIXEL_TOKEN
- GL_LINE_RESET_TOKEN
- GL_EXP
- GL_VIEWPORT_BIT
- GL_EXP2
- GL_CW
- GL_CCW
- GL_COEFF
- GL_ORDER
- GL_DOMAIN
- GL_CURRENT_COLOR
- GL_CURRENT_INDEX
- GL_CURRENT_NORMAL
- GL_CURRENT_TEXTURE_COORDS
- GL_CURRENT_RASTER_COLOR
- GL_CURRENT_RASTER_INDEX

- GL_CURRENT_RASTER_TEXTURE_COORDS
- GL_CURRENT_RASTER_POSITION
- GL_CURRENT_RASTER_POSITION_VALID
- GL_CURRENT_RASTER_DISTANCE
- GL_POINT_SMOOTH
- GL_POINT_SIZE
- GL_POINT_SIZE_RANGE
- GL_POINT_SIZE_GRANULARITY
- GL_LINE_SMOOTH
- GL_LINE_WIDTH
- GL_LINE_WIDTH_RANGE
- GL_LINE_WIDTH_GRANULARITY
- GL_LINE_STIPPLE
- GL_LINE_STIPPLE_PATTERN
- GL_LINE_STIPPLE_REPEAT
- GL_LIST_MODE
- GL_MAX_LIST_NESTING
- GL_LIST_BASE
- GL_LIST_INDEX
- GL_POLYGON_MODE
- GL_POLYGON_SMOOTH
- GL_POLYGON_STIPPLE
- GL_EDGE_FLAG
- GL_CULL_FACE
- GL_CULL_FACE_MODE
- GL_FRONT_FACE
- GL_LIGHTING
- GL_LIGHT_MODEL_LOCAL_VIEWER
- GL_LIGHT_MODEL_TWO_SIDE
- GL_LIGHT_MODEL_AMBIENT
- GL_SHADE_MODEL
- GL_COLOR_MATERIAL_FACE
- GL_COLOR_MATERIAL_PARAMETER
- GL_COLOR_MATERIAL
- GL_FOG
- GL_FOG_INDEX

- GL_FOG_DENSITY
- GL_FOG_START
- GL_FOG_END
- GL_FOG_MODE
- GL_FOG_COLOR
- GL_DEPTH_RANGE
- GL_DEPTH_TEST
- GL_DEPTH_WRITEMASK
- GL_DEPTH_CLEAR_VALUE
- GL_DEPTH_FUNC
- GL_ACCUM_CLEAR_VALUE
- GL_STENCIL_TEST
- GL_STENCIL_CLEAR_VALUE
- GL_STENCIL_FUNC
- GL_STENCIL_VALUE_MASK
- GL_STENCIL_FAIL
- GL_STENCIL_PASS_DEPTH_FAIL
- GL_STENCIL_PASS_DEPTH_PASS
- GL_STENCIL_REF
- GL_STENCIL_WRITEMASK
- GL_MATRIX_MODE
- GL_NORMALIZE
- GL_VIEWPORT
- GL_MODELVIEW_STACK_DEPTH
- GL_PROJECTION_STACK_DEPTH
- GL_TEXTURE_STACK_DEPTH
- GL_MODELVIEW_MATRIX
- GL_PROJECTION_MATRIX
- GL_TEXTURE_MATRIX
- GL_ATTRIB_STACK_DEPTH
- GL_CLIENT_ATTRIB_STACK_DEPTH
- GL_ALPHA_TEST
- GL_ALPHA_TEST_FUNC
- GL_ALPHA_TEST_REF
- GL_DITHER
- GL_BLEND_DST

- GL_BLEND_SRC
- GL_BLEND
- GL_LOGIC_OP_MODE
- GL_INDEX_LOGIC_OP
- GL_COLOR_LOGIC_OP
- GL_AUX_BUFFERS
- GL_DRAW_BUFFER
- GL_READ_BUFFER
- GL_SCISSOR_BOX
- GL_SCISSOR_TEST
- GL_INDEX_CLEAR_VALUE
- GL_INDEX_WRITEMASK
- GL_COLOR_CLEAR_VALUE
- GL_COLOR_WRITEMASK
- GL_INDEX_MODE
- GL_RGBA_MODE
- GL_DOUBLEBUFFER
- GL_STEREO
- GL_RENDER_MODE
- GL_PERSPECTIVE_CORRECTION_HINT
- GL_POINT_SMOOTH_HINT
- GL_LINE_SMOOTH_HINT
- GL_POLYGON_SMOOTH_HINT
- GL_FOG_HINT
- GL_TEXTURE_GEN_S
- GL_TEXTURE_GEN_T
- GL_TEXTURE_GEN_R
- GL_TEXTURE_GEN_Q
- GL_PIXEL_MAP_I_TO_I
- GL_PIXEL_MAP_S_TO_S
- GL_PIXEL_MAP_I_TO_R
- GL_PIXEL_MAP_I_TO_G
- GL_PIXEL_MAP_I_TO_B
- GL_PIXEL_MAP_I_TO_A
- GL_PIXEL_MAP_R_TO_R
- GL_PIXEL_MAP_G_TO_G

- GL_PIXEL_MAP_B_TO_B
- GL_PIXEL_MAP_A_TO_A
- GL_PIXEL_MAP_I_TO_I_SIZE
- GL_PIXEL_MAP_S_TO_S_SIZE
- GL_PIXEL_MAP_I_TO_R_SIZE
- GL_PIXEL_MAP_I_TO_G_SIZE
- GL_PIXEL_MAP_I_TO_B_SIZE
- GL_PIXEL_MAP_I_TO_A_SIZE
- GL_PIXEL_MAP_R_TO_R_SIZE
- GL_PIXEL_MAP_G_TO_G_SIZE
- GL_PIXEL_MAP_B_TO_B_SIZE
- GL_PIXEL_MAP_A_TO_A_SIZE
- GL_UNPACK_SWAP_BYTES
- GL_UNPACK_LSB_FIRST
- GL_UNPACK_ROW_LENGTH
- GL_UNPACK_SKIP_ROWS
- GL_UNPACK_SKIP_PIXELS
- GL_UNPACK_ALIGNMENT
- GL_PACK_SWAP_BYTES
- GL_PACK_LSB_FIRST
- GL_PACK_ROW_LENGTH
- GL_PACK_SKIP_ROWS
- GL_PACK_SKIP_PIXELS
- GL_PACK_ALIGNMENT
- GL_MAP_COLOR
- GL_MAP_STENCIL
- GL_INDEX_SHIFT
- GL_INDEX_OFFSET
- GL_RED_SCALE
- GL_RED_BIAS
- GL_ZOOM_X
- GL_ZOOM_Y
- GL_GREEN_SCALE
- GL_GREEN_BIAS
- GL_BLUE_SCALE
- GL_BLUE_BIAS

- GL_ALPHA_SCALE
- GL_ALPHA_BIAS
- GL_DEPTH_SCALE
- GL_DEPTH_BIAS
- GL_MAX_EVAL_ORDER
- GL_MAX_LIGHTS
- GL_MAX_CLIP_PLANES
- GL_MAX_TEXTURE_SIZE
- GL_MAX_PIXEL_MAP_TABLE
- GL_MAX_ATTRIB_STACK_DEPTH
- GL_MAX_MODELVIEW_STACK_DEPTH
- GL_MAX_NAME_STACK_DEPTH
- GL_MAX_PROJECTION_STACK_DEPTH
- GL_MAX_TEXTURE_STACK_DEPTH
- GL_MAX_VIEWPORT_DIMS
- GL_MAX_CLIENT_ATTRIB_STACK_DEPTH
- GL_SUBPIXEL_BITS
- GL_INDEX_BITS
- GL_RED_BITS
- GL_GREEN_BITS
- GL_BLUE_BITS
- GL_ALPHA_BITS
- GL_DEPTH_BITS
- GL_STENCIL_BITS
- GL_ACCUM_RED_BITS
- GL_ACCUM_GREEN_BITS
- GL_ACCUM_BLUE_BITS
- GL_ACCUM_ALPHA_BITS
- GL_NAME_STACK_DEPTH
- GL_AUTO_NORMAL
- GL_MAP1_COLOR_4
- GL_MAP1_INDEX
- GL_MAP1_NORMAL
- GL_MAP1_TEXTURE_COORD_1
- GL_MAP1_TEXTURE_COORD_2
- GL_MAP1_TEXTURE_COORD_3

- GL_MAP1_TEXTURE_COORD_4
- GL_MAP1_VERTEX_3
- GL_MAP1_VERTEX_4
- GL_MAP2_COLOR_4
- GL_MAP2_INDEX
- GL_MAP2_NORMAL
- GL_MAP2_TEXTURE_COORD_1
- GL_MAP2_TEXTURE_COORD_2
- GL_MAP2_TEXTURE_COORD_3
- GL_MAP2_TEXTURE_COORD_4
- GL_MAP2_VERTEX_3
- GL_MAP2_VERTEX_4
- GL_MAP1_GRID_DOMAIN
- GL_MAP1_GRID_SEGMENTS
- GL_MAP2_GRID_DOMAIN
- GL_MAP2_GRID_SEGMENTS
- GL_TEXTURE_1D
- GL_TEXTURE_2D
- GL_FEEDBACK_BUFFER_POINTER
- GL_FEEDBACK_BUFFER_SIZE
- GL_FEEDBACK_BUFFER_TYPE
- GL_SELECTION_BUFFER_POINTER
- GL_SELECTION_BUFFER_SIZE
- GL_TEXTURE_WIDTH
- GL_TRANSFORM_BIT
- GL_TEXTURE_HEIGHT
- GL_TEXTURE_INTERNAL_FORMAT
- GL_TEXTURE_BORDER_COLOR
- GL_TEXTURE_BORDER
- GL_DONT_CARE
- GL_FASTEST
- GL_NICEST
- GL_AMBIENT
- GL_DIFFUSE
- GL_SPECULAR
- GL_POSITION

- GL_SPOT_DIRECTION
- GL_SPOT_EXPONENT
- GL_SPOT_CUTOFF
- GL_CONSTANT_ATTENUATION
- GL_LINEAR_ATTENUATION
- GL_QUADRATIC_ATTENUATION
- GL_COMPILE
- GL_COMPILE_AND_EXECUTE
- GL_BYTE
- GL_UNSIGNED_BYTE
- GL_SHORT
- GL_UNSIGNED_SHORT
- GL_INT
- GL_UNSIGNED_INT
- GL_FLOAT
- GL_2_BYTES
- GL_3_BYTES
- GL_4_BYTES
- GL_DOUBLE
- GL_CLEAR
- GL_AND
- GL_AND_REVERSE
- GL_COPY
- GL_AND_INVERTED
- GL_NOOP
- GL_XOR
- GL_OR
- GL_NOR
- GL_EQUIV
- GL_INVERT
- GL_OR_REVERSE
- GL_COPY_INVERTED
- GL_OR_INVERTED
- GL_NAND
- GL_SET
- GL_EMISSION

- GL_SHININESS
- GL_AMBIENT_AND_DIFFUSE
- GL_COLOR_INDEXES
- GL_MODELVIEW
- GL_PROJECTION
- GL_TEXTURE
- GL_COLOR
- GL_DEPTH
- GL_STENCIL
- GL_COLOR_INDEX
- GL_STENCIL_INDEX
- GL_DEPTH_COMPONENT
- GL_RED
- GL_GREEN
- GL_BLUE
- GL_ALPHA
- GL_RGB
- GL_RGBA
- GL_LUMINANCE
- GL_LUMINANCE_ALPHA
- GL_BITMAP
- GL_POINT
- GL_LINE
- GL_FILL
- GL_RENDER
- GL_FEEDBACK
- GL_SELECT
- GL_FLAT
- GL_SMOOTH
- GL_KEEP
- GL_REPLACE
- GL_INCR
- GL_DECR
- GL_VENDOR
- GL_RENDERER
- GL_VERSION

- GL_EXTENSIONS
- GL_S
- GL_ENABLE_BIT
- GL_T
- GL_R
- GL_Q
- GL_MODULATE
- GL_DECAL
- GL_TEXTURE_ENV_MODE
- GL_TEXTURE_ENV_COLOR
- GL_TEXTURE_ENV
- GL_EYE_LINEAR
- GL_OBJECT_LINEAR
- GL_SPHERE_MAP
- GL_TEXTURE_GEN_MODE
- GL_OBJECT_PLANE
- GL_EYE_PLANE
- GL_NEAREST
- GL_LINEAR
- GL_NEAREST_MIPMAP_NEAREST
- GL_LINEAR_MIPMAP_NEAREST
- GL_NEAREST_MIPMAP_LINEAR
- GL_LINEAR_MIPMAP_LINEAR
- GL_TEXTURE_MAG_FILTER
- GL_TEXTURE_MIN_FILTER
- GL_TEXTURE_WRAP_S
- GL_TEXTURE_WRAP_T
- GL_CLAMP
- GL_REPEAT
- GL_POLYGON_OFFSET_UNITS
- GL_POLYGON_OFFSET_POINT
- GL_POLYGON_OFFSET_LINE
- GL_R3_G3_B2
- GL_V2F
- GL_V3F
- GL_C4UB_V2F

- GL_C4UB_V3F
- GL_C3F_V3F
- GL_N3F_V3F
- GL_C4F_N3F_V3F
- GL_T2F_V3F
- GL_T4F_V4F
- GL_T2F_C4UB_V3F
- GL_T2F_C3F_V3F
- GL_T2F_N3F_V3F
- GL_T2F_C4F_N3F_V3F
- GL_T4F_C4F_N3F_V4F
- GL_CLIP_PLANE0
- GL_CLIP_PLANE1
- GL_CLIP_PLANE2
- GL_CLIP_PLANE3
- GL_CLIP_PLANE4
- GL_CLIP_PLANE5
- GL_LIGHT0
- GL_COLOR_BUFFER_BIT
- GL_LIGHT1
- GL_LIGHT2
- GL_LIGHT3
- GL_LIGHT4
- GL_LIGHT5
- GL_LIGHT6
- GL_LIGHT7
- GL_HINT_BIT
- GL_POLYGON_OFFSET_FILL
- GL_POLYGON_OFFSET_FACTOR
- GL_ALPHA4
- GL_ALPHA8
- GL_ALPHA12
- GL_ALPHA16
- GL_LUMINANCE4
- GL_LUMINANCE8
- GL_LUMINANCE12

- GL_LUMINANCE16
- GL_LUMINANCE4_ALPHA4
- GL_LUMINANCE6_ALPHA2
- GL_LUMINANCE8_ALPHA8
- GL_LUMINANCE12_ALPHA4
- GL_LUMINANCE12_ALPHA12
- GL_LUMINANCE16_ALPHA16
- GL_INTENSITY
- GL_INTENSITY4
- GL_INTENSITY8
- GL_INTENSITY12
- GL_INTENSITY16
- GL_RGB4
- GL_RGB5
- GL_RGB8
- GL_RGB10
- GL_RGB12
- GL_RGB16
- GL_RGBA2
- GL_RGBA4
- GL_RGB5_A1
- GL_RGBA8
- GL_RGB10_A2
- GL_RGBA12
- GL_RGBA16
- GL_TEXTURE_RED_SIZE
- GL_TEXTURE_GREEN_SIZE
- GL_TEXTURE_BLUE_SIZE
- GL_TEXTURE_ALPHA_SIZE
- GL_TEXTURE_LUMINANCE_SIZE
- GL_TEXTURE_INTENSITY_SIZE
- GL_PROXY_TEXTURE_1D
- GL_PROXY_TEXTURE_2D
- GL_TEXTURE_PRIORITY
- GL_TEXTURE_RESIDENT
- GL_TEXTURE_BINDING_1D

- GL_TEXTURE_BINDING_2D
- GL_VERTEX_ARRAY
- GL_NORMAL_ARRAY
- GL_COLOR_ARRAY
- GL_INDEX_ARRAY
- GL_TEXTURE_COORD_ARRAY
- GL_EDGE_FLAG_ARRAY
- GL_VERTEX_ARRAY_SIZE
- GL_VERTEX_ARRAY_TYPE
- GL_VERTEX_ARRAY_STRIDE
- GL_NORMAL_ARRAY_TYPE
- GL_NORMAL_ARRAY_STRIDE
- GL_COLOR_ARRAY_SIZE
- GL_COLOR_ARRAY_TYPE
- GL_COLOR_ARRAY_STRIDE
- GL_INDEX_ARRAY_TYPE
- GL_INDEX_ARRAY_STRIDE
- GL_TEXTURE_COORD_ARRAY_SIZE
- GL_TEXTURE_COORD_ARRAY_TYPE
- GL_TEXTURE_COORD_ARRAY_STRIDE
- GL_EDGE_FLAG_ARRAY_STRIDE
- GL_VERTEX_ARRAY_POINTER
- GL_NORMAL_ARRAY_POINTER
- GL_COLOR_ARRAY_POINTER
- GL_INDEX_ARRAY_POINTER
- GL_TEXTURE_COORD_ARRAY_POINTER
- GL_EDGE_FLAG_ARRAY_POINTER
- GL_COLOR_INDEX1_EXT
- GL_COLOR_INDEX2_EXT
- GL_COLOR_INDEX4_EXT
- GL_COLOR_INDEX8_EXT
- GL_COLOR_INDEX12_EXT
- GL_COLOR_INDEX16_EXT
- GL_EVAL_BIT
- GL_LIST_BIT
- GL_TEXTURE_BIT

- GL_SCISSOR_BIT
- GL_ALL_ATTRIB_BITS
- GL_CLIENT_ALL_ATTRIB_BITS
- GL_SMOOTH_POINT_SIZE_RANGE
- GL_SMOOTH_POINT_SIZE_GRANULARITY
- GL_SMOOTH_LINE_WIDTH_RANGE
- GL_SMOOTH_LINE_WIDTH_GRANULARITY
- GL_UNSIGNED_BYTE_3_3_2
- GL_UNSIGNED_SHORT_4_4_4_4
- GL_UNSIGNED_SHORT_5_5_5_1
- GL_UNSIGNED_INT_8_8_8_8
- GL_UNSIGNED_INT_10_10_10_2
- GL_RESCALE_NORMAL
- GL_TEXTURE_BINDING_3D
- GL_PACK_SKIP_IMAGES
- GL_PACK_IMAGE_HEIGHT
- GL_UNPACK_SKIP_IMAGES
- GL_UNPACK_IMAGE_HEIGHT
- GL_TEXTURE_3D
- GL_PROXY_TEXTURE_3D
- GL_TEXTURE_DEPTH
- GL_TEXTURE_WRAP_R
- GL_MAX_3D_TEXTURE_SIZE
- GL_BGR
- GL_BGRA
- GL_MAX_ELEMENTS_VERTICES
- GL_MAX_ELEMENTS_INDICES
- GL_CLAMP_TO_EDGE
- GL_TEXTURE_MIN_LOD
- GL_TEXTURE_MAX_LOD
- GL_TEXTURE_BASE_LEVEL
- GL_TEXTURE_MAX_LEVEL
- GL_LIGHT_MODEL_COLOR_CONTROL
- GL_SINGLE_COLOR
- GL_SEPARATE_SPECULAR_COLOR
- GL_UNSIGNED_BYTE_2_3_3_REV

- GL_UNSIGNED_SHORT_5_6_5
- GL_UNSIGNED_SHORT_5_6_5_REV
- GL_UNSIGNED_SHORT_4_4_4_REV
- GL_UNSIGNED_SHORT_1_5_5_5_REV
- GL_UNSIGNED_INT_8_8_8_8_REV
- GL_ALIASED_POINT_SIZE_RANGE
- GL_ALIASED_LINE_WIDTH_RANGE
- GL_MULTISAMPLE
- GL_SAMPLE_ALPHA_TO_COVERAGE
- GL_SAMPLE_ALPHA_TO_ONE
- GL_SAMPLE_COVERAGE
- GL_SAMPLE_BUFFERS
- GL_SAMPLES
- GL_SAMPLE_COVERAGE_VALUE
- GL_SAMPLE_COVERAGE_INVERT
- GL_CLAMP_TO_BORDER
- GL_TEXTURE0
- GL_TEXTURE1
- GL_TEXTURE2
- GL_TEXTURE3
- GL_TEXTURE4
- GL_TEXTURE5
- GL_TEXTURE6
- GL_TEXTURE7
- GL_TEXTURE8
- GL_TEXTURE9
- GL_TEXTURE10
- GL_TEXTURE11
- GL_TEXTURE12
- GL_TEXTURE13
- GL_TEXTURE14
- GL_TEXTURE15
- GL_TEXTURE16
- GL_TEXTURE17
- GL_TEXTURE18
- GL_TEXTURE19

- GL_TEXTURE20
- GL_TEXTURE21
- GL_TEXTURE22
- GL_TEXTURE23
- GL_TEXTURE24
- GL_TEXTURE25
- GL_TEXTURE26
- GL_TEXTURE27
- GL_TEXTURE28
- GL_TEXTURE29
- GL_TEXTURE30
- GL_TEXTURE31
- GL_ACTIVE_TEXTURE
- GL_CLIENT_ACTIVE_TEXTURE
- GL_MAX_TEXTURE_UNITS
- GL_TRANSPOSE_MODELVIEW_MATRIX
- GL_TRANSPOSE_PROJECTION_MATRIX
- GL_TRANSPOSE_TEXTURE_MATRIX
- GL_TRANSPOSE_COLOR_MATRIX
- GL_SUBTRACT
- GL_COMPRESSED_ALPHA
- GL_COMPRESSED_LUMINANCE
- GL_COMPRESSED_LUMINANCE_ALPHA
- GL_COMPRESSED_INTENSITY
- GL_COMPRESSED_RGB
- GL_COMPRESSED_RGBA
- GL_TEXTURE_COMPRESSION_HINT
- GL_NORMAL_MAP
- GL_REFLECTION_MAP
- GL_TEXTURE_CUBE_MAP
- GL_TEXTURE_BINDING_CUBE_MAP
- GL_TEXTURE_CUBE_MAP_POSITIVE_X
- GL_TEXTURE_CUBE_MAP_NEGATIVE_X
- GL_TEXTURE_CUBE_MAP_POSITIVE_Y
- GL_TEXTURE_CUBE_MAP_NEGATIVE_Y
- GL_TEXTURE_CUBE_MAP_POSITIVE_Z

- GL_TEXTURE_CUBE_MAP_NEGATIVE_Z
- GL_PROXY_TEXTURE_CUBE_MAP
- GL_MAX_CUBE_MAP_TEXTURE_SIZE
- GL_COMBINE
- GL_COMBINE_RGB
- GL_COMBINE_ALPHA
- GL_RGB_SCALE
- GL_ADD_SIGNED
- GL_INTERPOLATE
- GL_CONSTANT
- GL_PRIMARY_COLOR
- GL_PREVIOUS
- GL_SOURCE0_RGB
- GL_SOURCE1_RGB
- GL_SOURCE2_RGB
- GL_SOURCE0_ALPHA
- GL_SOURCE1_ALPHA
- GL_SOURCE2_ALPHA
- GL_OPERAND0_RGB
- GL_OPERAND1_RGB
- GL_OPERAND2_RGB
- GL_OPERAND0_ALPHA
- GL_OPERAND1_ALPHA
- GL_OPERAND2_ALPHA
- GL_TEXTURE_COMPRESSED_IMAGE_SIZE
- GL_TEXTURE_COMPRESSED
- GL_NUM_COMPRESSED_TEXTURE_FORMATS
- GL_COMPRESSED_TEXTURE_FORMATS
- GL_DOT3_RGB
- GL_DOT3_RGBA
- GL_MULTISAMPLE_BIT
- GL_BLEND_DST_RGB
- GL_BLEND_SRC_RGB
- GL_BLEND_DST_ALPHA
- GL_BLEND_SRC_ALPHA
- GL_POINT_SIZE_MIN

- GL_POINT_SIZE_MAX
- GL_POINT_FADE_THRESHOLD_SIZE
- GL_POINT_DISTANCE_ATTENUATION
- GL_GENERATE_MIPMAP
- GL_GENERATE_MIPMAP_HINT
- GL_DEPTH_COMPONENT16
- GL_DEPTH_COMPONENT24
- GL_DEPTH_COMPONENT32
- GL_MIRRORED_REPEAT
- GL_FOG_COORDINATE_SOURCE
- GL_FOG_COORDINATE
- GL_FRAGMENT_DEPTH
- GL_CURRENT_FOG_COORDINATE
- GL_FOG_COORDINATE_ARRAY_TYPE
- GL_FOG_COORDINATE_ARRAY_STRIDE
- GL_FOG_COORDINATE_ARRAY_POINTER
- GL_FOG_COORDINATE_ARRAY
- GL_COLOR_SUM
- GL_CURRENT_SECONDARY_COLOR
- GL_SECONDARY_COLOR_ARRAY_SIZE
- GL_SECONDARY_COLOR_ARRAY_TYPE
- GL_SECONDARY_COLOR_ARRAY_STRIDE
- GL_SECONDARY_COLOR_ARRAY_POINTER
- GL_SECONDARY_COLOR_ARRAY
- GL_MAX_TEXTURE_LOD_BIAS
- GL_TEXTURE_FILTER_CONTROL
- GL_TEXTURE_LOD_BIAS
- GL_INCR_WRAP
- GL_DECR_WRAP
- GL_TEXTURE_DEPTH_SIZE
- GL_DEPTH_TEXTURE_MODE
- GL_TEXTURE_COMPARE_MODE
- GL_TEXTURE_COMPARE_FUNC
- GL_COMPARE_R_TO_TEXTURE
- GL_CURRENT_FOG_COORD
- GL_FOG_COORD

- GL_FOG_COORD_ARRAY
- GL_FOG_COORD_ARRAY_BUFFER_BINDING
- GL_FOG_COORD_ARRAY_POINTER
- GL_FOG_COORD_ARRAY_STRIDE
- GL_FOG_COORD_ARRAY_TYPE
- GL_FOG_COORD_SRC
- GL_SRC0_ALPHA
- GL_SRC0_RGB
- GL_SRC1_ALPHA
- GL_SRC1_RGB
- GL_SRC2_ALPHA
- GL_SRC2_RGB
- GL_BUFFER_SIZE
- GL_BUFFER_USAGE
- GL_QUERY_COUNTER_BITS
- GL_CURRENT_QUERY
- GL_QUERY_RESULT
- GL_QUERY_RESULT_AVAILABLE
- GL_ARRAY_BUFFER
- GL_ELEMENT_ARRAY_BUFFER
- GL_ARRAY_BUFFER_BINDING
- GL_ELEMENT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ARRAY_BUFFER_BINDING
- GL_NORMAL_ARRAY_BUFFER_BINDING
- GL_COLOR_ARRAY_BUFFER_BINDING
- GL_INDEX_ARRAY_BUFFER_BINDING
- GL_TEXTURE_COORD_ARRAY_BUFFER_BINDING
- GL_EDGE_FLAG_ARRAY_BUFFER_BINDING
- GL_SECONDARY_COLOR_ARRAY_BUFFER_BINDING
- GL_FOG_COORDINATE_ARRAY_BUFFER_BINDING
- GL_WEIGHT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ATTRIB_ARRAY_BUFFER_BINDING
- GL_READ_ONLY
- GL_WRITE_ONLY
- GL_READ_WRITE
- GL_BUFFER_ACCESS

- GL_BUFFER_MAPPED
- GL_BUFFER_MAP_POINTER
- GL_STREAM_DRAW
- GL_STREAM_READ
- GL_STREAM_COPY
- GL_STATIC_DRAW
- GL_STATIC_READ
- GL_STATIC_COPY
- GL_DYNAMIC_DRAW
- GL_DYNAMIC_READ
- GL_DYNAMIC_COPY
- GL_SAMPLES_PASSED
- void glAccum(GLenum op, GLfloat value)
- void glAlphaFunc(GLenum func, GLclampf ref)
- GLboolean glAreTexturesResident(GLsizei n, const GLuint *textures, GLboolean *residences)
- void glArrayElement(GLint i)
- void glBegin(GLenum mode)
- void glBindTexture(GLenum target, GLuint texture)
- void glBitmap(GLsizei width, GLsizei height, GLfloat xorig, GLfloat yorig, GLfloat xmove, GLfloat ymove, const GLubyte *bitmap)
- void glBlendFunc(GLenum sfactor, GLenum dfactor)
- void glCallList(GLuint list)
- void glCallLists(GLsizei n, GLenum type, const void *lists)
- void glClear(GLbitfield mask)
- void glClearAccum(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glClearColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)
- void glClearDepth(GLclampd depth)
- void glClearIndex(GLfloat c)
- void glClearStencil(GLint s)
- void glClipPlane(GLenum plane, const GLdouble *equation)
- void glColor3b(GLbyte red, GLbyte green, GLbyte blue)
- void glColor3bv(const GLbyte *v)
- void glColor3d(GLdouble red, GLdouble green, GLdouble blue)
- void glColor3dv(const GLdouble *v)
- void glColor3f(GLfloat red, GLfloat green, GLfloat blue)
- void glColor3fv(const GLfloat *v)

- void glColor3i(GLint red, GLint green, GLint blue)
- void glColor3iv(const GLint *v)
- void glColor3s(GLshort red, GLshort green, GLshort blue)
- void glColor3sv(const GLshort *v)
- void glColor3ub(GLubyte red, GLubyte green, GLubyte blue)
- void glColor3ubv(const GLubyte *v)
- void glColor3ui(GLuint red, GLuint green, GLuint blue)
- void glColor3uiv(const GLuint *v)
- void glColor3us(GLushort red, GLushort green, GLushort blue)
- void glColor3usv(const GLushort *v)
- void glColor4b(GLbyte red, GLbyte green, GLbyte blue, GLbyte alpha)
- void glColor4bv(const GLbyte *v)
- void glColor4d(GLdouble red, GLdouble green, GLdouble blue, GLdouble alpha)
- void glColor4dv(const GLdouble *v)
- void glColor4f(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glColor4fv(const GLfloat *v)
- void glColor4i(GLint red, GLint green, GLint blue, GLint alpha)
- void glColor4iv(const GLint *v)
- void glColor4s(GLshort red, GLshort green, GLshort blue, GLshort alpha)
- void glColor4sv(const GLshort *v)
- void glColor4ub(GLubyte red, GLubyte green, GLubyte blue, GLubyte alpha)
- void glColor4ubv(const GLubyte *v)
- void glColor4ui(GLuint red, GLuint green, GLuint blue, GLuint alpha)
- void glColor4uiv(const GLuint *v)
- void glColor4us(GLushort red, GLushort green, GLushort blue, GLushort alpha)
- void glColor4usv(const GLushort *v)
- void glColorMask(GLboolean red, GLboolean green, GLboolean blue, GLboolean alpha)
- void glColorMaterial(GLenum face, GLenum mode)
- void glColorPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)
- void glCopyPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum type)
- void glCopyTexImage1D(GLenum target, GLint level, GLenum internalFormat, GLint x, GLint y, GLsizei width, GLint border)
- void glCopyTexImage2D(GLenum target, GLint level, GLenum internalFormat, GLint x, GLint y, GLsizei width, GLsizei height, GLint border)
- void glCopyTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLint x, GLint y, GLsizei width)
- void glCopyTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint x, GLint y, GLsizei width, GLsizei height)

- void glCullFace(GLenum mode)
- void glDeleteLists(GLuint list, GLsizei range)
- void glDeleteTextures(GLsizei n, const GLuint *textures)
- void glDepthFunc(GLenum func)
- void glDepthMask(GLboolean flag)
- void glDepthRange(GLclampd zNear, GLclampd zFar)
- void glDisable(GLenum cap)
- void glDisableClientState(GLenum array)
- void glDrawArrays(GLenum mode, GLint first, GLsizei count)
- void glDrawBuffer(GLenum mode)
- void glDrawElements(GLenum mode, GLsizei count, GLenum type, const void *indices)
- void glDrawPixels(GLsizei width, GLsizei height, GLenum format, GLenum type, const void *pixels)
- void glEdgeFlag(GLboolean flag)
- void glEdgeFlagPointer(GLsizei stride, const void *pointer)
- void glEdgeFlagv(const GLboolean *flag)
- void glEnable(GLenum cap)
- void glEnableClientState(GLenum array)
- void glEnd(void)
- void glEndList(void)
- void glEvalCoord1d(GLdouble u)
- void glEvalCoord1dv(const GLdouble *u)
- void glEvalCoord1f(GLfloat u)
- void glEvalCoord1fv(const GLfloat *u)
- void glEvalCoord2d(GLdouble u, GLdouble v)
- void glEvalCoord2dv(const GLdouble *u)
- void glEvalCoord2f(GLfloat u, GLfloat v)
- void glEvalCoord2fv(const GLfloat *u)
- void glEvalMesh1(GLenum mode, GLint i1, GLint i2)
- void glEvalMesh2(GLenum mode, GLint i1, GLint i2, GLint j1, GLint j2)
- void glEvalPoint1(GLint i)
- void glEvalPoint2(GLint i, GLint j)
- void glFeedbackBuffer(GLsizei size, GLenum type, GLfloat *buffer)
- void glFinish(void)
- void glFlush(void)
- void glFogf(GLenum pname, GLfloat param)
- void glFogfv(GLenum pname, const GLfloat *params)

- void glFogi(GLenum pname, GLint param)
- void glFogiv(GLenum pname, const GLint *params)
- void glFrontFace(GLenum mode)
- void glFrustum(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble zNear, GLdouble zFar)
- GLuint glGenLists(GLsizei range)
- void glGenTextures(GLsizei n, GLuint *textures)
- void glGetBooleanv(GLenum pname, GLboolean *params)
- void glGetClipPlane(GLenum plane, GLdouble *equation)
- void glGetDoublev(GLenum pname, GLdouble *params)
- GLenum glGetError(void)
- void glGetFloatv(GLenum pname, GLfloat *params)
- void glGetIntegerv(GLenum pname, GLint *params)
- void glGetLightfv(GLenum light, GLenum pname, GLfloat *params)
- void glGetLightiv(GLenum light, GLenum pname, GLint *params)
- void glGetMapdv(GLenum target, GLenum query, GLdouble *v)
- void glGetMapfv(GLenum target, GLenum query, GLfloat *v)
- void glGetMapiv(GLenum target, GLenum query, GLint *v)
- void glGetMaterialfv(GLenum face, GLenum pname, GLfloat *params)
- void glGetMaterialiv(GLenum face, GLenum pname, GLint *params)
- void glGetPixelMapfv(GLenum map, GLfloat *values)
- void glGetPixelMapuiv(GLenum map, GLuint *values)
- void glGetPixelMapusv(GLenum map, GLushort *values)
- void glGetPointerv(GLenum pname, void* *params)
- void glGetPolygonStipple(GLubyte *mask)
- GLubyte * glGetString(GLenum name)
- void glGetTexEnvfv(GLenum target, GLenum pname, GLfloat *params)
- void glGetTexEnviv(GLenum target, GLenum pname, GLint *params)
- void glGetTexGendv(GLenum coord, GLenum pname, GLdouble *params)
- void glGetTexGenfv(GLenum coord, GLenum pname, GLfloat *params)
- void glGetTexGeniv(GLenum coord, GLenum pname, GLint *params)
- void glGetTexImage(GLenum target, GLint level, GLenum format, GLenum type, void *pixels)
- void glGetTexLevelParameterfv(GLenum target, GLint level, GLenum pname, GLfloat *params)
- void glGetTexLevelParameteriv(GLenum target, GLint level, GLenum pname, GLint *params)
- void glGetTexParameterfv(GLenum target, GLenum pname, GLfloat *params)
- void glGetTexParameteriv(GLenum target, GLenum pname, GLint *params)

- void glHint(GLenum target, GLenum mode)
- void glIndexMask(GLuint mask)
- void glIndexPointer(GLenum type, GLsizei stride, const void *pointer)
- void glIndexd(GLdouble c)
- void glIndexdv(const GLdouble *c)
- void glIndexf(GLfloat c)
- void glIndexfv(const GLfloat *c)
- void glIndexi(GLint c)
- void glIndexiv(const GLint *c)
- void glIndexs(GLshort c)
- void glIndexsv(const GLshort *c)
- void glIndexub(GLubyte c)
- void glIndexubv(const GLubyte *c)
- void glInitNames(void)
- void glInterleavedArrays(GLenum format, GLsizei stride, const void *pointer)
- GLboolean glIsEnabled(GLenum cap)
- GLboolean glIsList(GLuint list)
- GLboolean glIsTexture(GLuint texture)
- void glLightModelf(GLenum pname, GLfloat param)
- void glLightModelfv(GLenum pname, const GLfloat *params)
- void glLightModeli(GLenum pname, GLint param)
- void glLightModeliv(GLenum pname, const GLint *params)
- void glLightf(GLenum light, GLenum pname, GLfloat param)
- void glLightfv(GLenum light, GLenum pname, const GLfloat *params)
- void glLighti(GLenum light, GLenum pname, GLint param)
- void glLightiv(GLenum light, GLenum pname, const GLint *params)
- void glLineStipple(GLint factor, GLushort pattern)
- void glLineWidth(GLfloat width)
- void glListBase(GLuint base)
- void glLoadIdentity(void)
- void glLoadMatrixd(const GLdouble *m)
- void glLoadMatrixf(const GLfloat *m)
- void glLoadName(GLuint name)
- void glLogicOp(GLenum opcode)
- void glMap1d(GLenum target, GLdouble u1, GLdouble u2, GLint stride, GLint order, const GLdouble *points)
- void glMap1f(GLenum target, GLfloat u1, GLfloat u2, GLint stride, GLint order, const GLfloat *points)

- void glMap2d(GLenum target, GLdouble u1, GLdouble u2, GLint ustride, GLint uorder, GLdouble v1, GLdouble v2, GLint vstride, GLint vorder, const GLdouble *points)
- void glMap2f(GLenum target, GLfloat u1, GLfloat u2, GLint ustride, GLint uorder, GLfloat v1, GLfloat v2, GLint vstride, GLint vorder, const GLfloat *points)
- void glMapGrid1d(GLint un, GLdouble u1, GLdouble u2)
- void glMapGrid1f(GLint un, GLfloat u1, GLfloat u2)
- void glMapGrid2d(GLint un, GLdouble u1, GLdouble u2, GLint vn, GLdouble v1, GLdouble v2)
- void glMapGrid2f(GLint un, GLfloat u1, GLfloat u2, GLint vn, GLfloat v1, GLfloat v2)
- void glMaterialf(GLenum face, GLenum pname, GLfloat param)
- void glMaterialfv(GLenum face, GLenum pname, const GLfloat *params)
- void glMateriali(GLenum face, GLenum pname, GLint param)
- void glMaterialiv(GLenum face, GLenum pname, const GLint *params)
- void glMatrixMode(GLenum mode)
- void glMultMatrixd(const GLdouble *m)
- void glMultMatrixf(const GLfloat *m)
- void glNewList(GLuint list, GLenum mode)
- void glNormal3b(GLbyte nx, GLbyte ny, GLbyte nz)
- void glNormal3bv(const GLbyte *v)
- void glNormal3d(GLdouble nx, GLdouble ny, GLdouble nz)
- void glNormal3dv(const GLdouble *v)
- void glNormal3f(GLfloat nx, GLfloat ny, GLfloat nz)
- void glNormal3fv(const GLfloat *v)
- void glNormal3i(GLint nx, GLint ny, GLint nz)
- void glNormal3iv(const GLint *v)
- void glNormal3s(GLshort nx, GLshort ny, GLshort nz)
- void glNormal3sv(const GLshort *v)
- void glNormalPointer(GLenum type, GLsizei stride, const void *pointer)
- void glOrtho(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble zNear, GLdouble zFar)
- void glPassThrough(GLfloat token)
- void glPixelMapfv(GLenum map, GLsizei mapsize, const GLfloat *values)
- void glPixelMapuiv(GLenum map, GLsizei mapsize, const GLuint *values)
- void glPixelMapusv(GLenum map, GLsizei mapsize, const GLushort *values)
- void glPixelStoref(GLenum pname, GLfloat param)
- void glPixelStorei(GLenum pname, GLint param)
- void glPixelTransferf(GLenum pname, GLfloat param)
- void glPixelTransferi(GLenum pname, GLint param)

- void glPixelZoom(GLfloat xfactor, GLfloat yfactor)
- void glPointSize(GLfloat size)
- void glPolygonMode(GLenum face, GLenum mode)
- void glPolygonOffset(GLfloat factor, GLfloat units)
- void glPolygonStipple(const GLubyte *mask)
- void glPopAttrib(void)
- void glPopClientAttrib(void)
- void glPopMatrix(void)
- void glPopName(void)
- void glPrioritizeTextures(GLsizei n, const GLuint *textures, const GLclampf *priorities)
- void glPushAttrib(GLbitfield mask)
- void glPushClientAttrib(GLbitfield mask)
- void glPushMatrix(void)
- void glPushName(GLuint name)
- void glRasterPos2d(GLdouble x, GLdouble y)
- void glRasterPos2dv(const GLdouble *v)
- void glRasterPos2f(GLfloat x, GLfloat y)
- void glRasterPos2fv(const GLfloat *v)
- void glRasterPos2i(GLint x, GLint y)
- void glRasterPos2iv(const GLint *v)
- void glRasterPos2s(GLshort x, GLshort y)
- void glRasterPos2sv(const GLshort *v)
- void glRasterPos3d(GLdouble x, GLdouble y, GLdouble z)
- void glRasterPos3dv(const GLdouble *v)
- void glRasterPos3f(GLfloat x, GLfloat y, GLfloat z)
- void glRasterPos3fv(const GLfloat *v)
- void glRasterPos3i(GLint x, GLint y, GLint z)
- void glRasterPos3iv(const GLint *v)
- void glRasterPos3s(GLshort x, GLshort y, GLshort z)
- void glRasterPos3sv(const GLshort *v)
- void glRasterPos4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)
- void glRasterPos4dv(const GLdouble *v)
- void glRasterPos4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- void glRasterPos4fv(const GLfloat *v)
- void glRasterPos4i(GLint x, GLint y, GLint z, GLint w)
- void glRasterPos4iv(const GLint *v)

- void glRasterPos4s(GLshort x, GLshort y, GLshort z, GLshort w)
- void glRasterPos4sv(const GLshort *v)
- void glReadBuffer(GLenum mode)
- void glReadPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum format, GLenum type, void *pixels)
- void glRectd(GLdouble x1, GLdouble y1, GLdouble x2, GLdouble y2)
- void glRectdv(const GLdouble *v1, const GLdouble *v2)
- void glRectf(GLfloat x1, GLfloat y1, GLfloat x2, GLfloat y2)
- void glRectfv(const GLfloat *v1, const GLfloat *v2)
- void glRecti(GLint x1, GLint y1, GLint x2, GLint y2)
- void glRectiv(const GLint *v1, const GLint *v2)
- void glRects(GLshort x1, GLshort y1, GLshort x2, GLshort y2)
- void glRectsv(const GLshort *v1, const GLshort *v2)
- GLint glRenderMode(GLenum mode)
- void glRotated(GLdouble angle, GLdouble x, GLdouble y, GLdouble z)
- void glRotatef(GLfloat angle, GLfloat x, GLfloat y, GLfloat z)
- void glScaled(GLdouble x, GLdouble y, GLdouble z)
- void glScalef(GLfloat x, GLfloat y, GLfloat z)
- void glScissor(GLint x, GLint y, GLsizei width, GLsizei height)
- void glSelectBuffer(GLsizei size, GLuint *buffer)
- void glShadeModel(GLenum mode)
- void glStencilFunc(GLenum func, GLint ref, GLuint mask)
- void glStencilMask(GLuint mask)
- void glStencilOp(GLenum fail, GLenum zfail, GLenum zpass)
- void glTexCoord1d(GLdouble s)
- void glTexCoord1dv(const GLdouble *v)
- void glTexCoord1f(GLfloat s)
- void glTexCoord1fv(const GLfloat *v)
- void glTexCoord1i(GLint s)
- void glTexCoord1iv(const GLint *v)
- void glTexCoord1s(GLshort s)
- void glTexCoord1sv(const GLshort *v)
- void glTexCoord2d(GLdouble s, GLdouble t)
- void glTexCoord2dv(const GLdouble *v)
- void glTexCoord2f(GLfloat s, GLfloat t)
- void glTexCoord2fv(const GLfloat *v)

- `void glTexCoord2i(GLint s, GLint t)`
- `void glTexCoord2iv(const GLint *v)`
- `void glTexCoord2s(GLshort s, GLshort t)`
- `void glTexCoord2sv(const GLshort *v)`
- `void glTexCoord3d(GLdouble s, GLdouble t, GLdouble r)`
- `void glTexCoord3dv(const GLdouble *v)`
- `void glTexCoord3f(GLfloat s, GLfloat t, GLfloat r)`
- `void glTexCoord3fv(const GLfloat *v)`
- `void glTexCoord3i(GLint s, GLint t, GLint r)`
- `void glTexCoord3iv(const GLint *v)`
- `void glTexCoord3s(GLshort s, GLshort t, GLshort r)`
- `void glTexCoord3sv(const GLshort *v)`
- `void glTexCoord4d(GLdouble s, GLdouble t, GLdouble r, GLdouble q)`
- `void glTexCoord4dv(const GLdouble *v)`
- `void glTexCoord4f(GLfloat s, GLfloat t, GLfloat r, GLfloat q)`
- `void glTexCoord4fv(const GLfloat *v)`
- `void glTexCoord4i(GLint s, GLint t, GLint r, GLint q)`
- `void glTexCoord4iv(const GLint *v)`
- `void glTexCoord4s(GLshort s, GLshort t, GLshort r, GLshort q)`
- `void glTexCoord4sv(const GLshort *v)`
- `void glTexCoordPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)`
- `void glTexEnvf(GLenum target, GLenum pname, GLfloat param)`
- `void glTexEnvfv(GLenum target, GLenum pname, const GLfloat *params)`
- `void glTexEnvf(GLenum target, GLenum pname, GLint param)`
- `void glTexEnviv(GLenum target, GLenum pname, const GLint *params)`
- `void glTexGend(GLenum coord, GLenum pname, GLdouble param)`
- `void glTexGendv(GLenum coord, GLenum pname, const GLdouble *params)`
- `void glTexGenf(GLenum coord, GLenum pname, GLfloat param)`
- `void glTexGenfv(GLenum coord, GLenum pname, const GLfloat *params)`
- `void glTexGeni(GLenum coord, GLenum pname, GLint param)`
- `void glTexGeniv(GLenum coord, GLenum pname, const GLint *params)`
- `void glTexImage1D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLint border, GLenum format, GLenum type, const void *pixels)`
- `void glTexImage2D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, const void *pixels)`
- `void glTexParameterf(GLenum target, GLenum pname, GLfloat param)`
- `void glTexParameterfv(GLenum target, GLenum pname, const GLfloat *params)`

- void glTexParameteri(GLenum target, GLenum pname, GLint param)
- void glTexParameteriv(GLenum target, GLenum pname, const GLint *params)
- void glTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLenum type, const void *pixels)
- void glTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLenum type, const void *pixels)
- void glTranslated(GLdouble x, GLdouble y, GLdouble z)
- void glTranslatef(GLfloat x, GLfloat y, GLfloat z)
- void glVertex2d(GLdouble x, GLdouble y)
- void glVertex2dv(const GLdouble *v)
- void glVertex2f(GLfloat x, GLfloat y)
- void glVertex2fv(const GLfloat *v)
- void glVertex2i(GLint x, GLint y)
- void glVertex2iv(const GLint *v)
- void glVertex2s(GLshort x, GLshort y)
- void glVertex2sv(const GLshort *v)
- void glVertex3d(GLdouble x, GLdouble y, GLdouble z)
- void glVertex3dv(const GLdouble *v)
- void glVertex3f(GLfloat x, GLfloat y, GLfloat z)
- void glVertex3fv(const GLfloat *v)
- void glVertex3i(GLint x, GLint y, GLint z)
- void glVertex3iv(const GLint *v)
- void glVertex3s(GLshort x, GLshort y, GLshort z)
- void glVertex3sv(const GLshort *v)
- void glVertex4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)
- void glVertex4dv(const GLdouble *v)
- void glVertex4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- void glVertex4fv(const GLfloat *v)
- void glVertex4i(GLint x, GLint y, GLint z, GLint w)
- void glVertex4iv(const GLint *v)
- void glVertex4s(GLshort x, GLshort y, GLshort z, GLshort w)
- void glVertex4sv(const GLshort *v)
- void glVertexPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)
- void glViewport(GLint x, GLint y, GLsizei width, GLsizei height)

RINGOPENGL (OPENGL 2.0) FUNCTIONS REFERENCE

- GL_ZERO
- GL_FALSE
- GL_LOGIC_OP
- GL_NONE
- GL_TEXTURE_COMPONENTS
- GL_NO_ERROR
- GL_POINTS
- GL_CURRENT_BIT
- GL_TRUE
- GL_ONE
- GL_CLIENT_PIXEL_STORE_BIT
- GL_LINES
- GL_LINE_LOOP
- GL_POINT_BIT
- GL_CLIENT_VERTEX_ARRAY_BIT
- GL_LINE_STRIP
- GL_LINE_BIT
- GL_TRIANGLES
- GL_TRIANGLE_STRIP
- GL_TRIANGLE_FAN
- GL_QUADS
- GL_QUAD_STRIP
- GL_POLYGON_BIT
- GL_POLYGON
- GL_POLYGON_STIPPLE_BIT
- GL_PIXEL_MODE_BIT
- GL_LIGHTING_BIT

- GL_FOG_BIT
- GL_DEPTH_BUFFER_BIT
- GL_ACCUM
- GL_LOAD
- GL_RETURN
- GL_MULT
- GL_ADD
- GL_NEVER
- GL_ACCUM_BUFFER_BIT
- GL_LESS
- GL_EQUAL
- GL_LEQUAL
- GL_GREATER
- GL_NOTEQUAL
- GL_GEQUAL
- GL_ALWAYS
- GL_SRC_COLOR
- GL_ONE_MINUS_SRC_COLOR
- GL_SRC_ALPHA
- GL_ONE_MINUS_SRC_ALPHA
- GL_DST_ALPHA
- GL_ONE_MINUS_DST_ALPHA
- GL_DST_COLOR
- GL_ONE_MINUS_DST_COLOR
- GL_SRC_ALPHA_SATURATE
- GL_STENCIL_BUFFER_BIT
- GL_FRONT_LEFT
- GL_FRONT_RIGHT
- GL_BACK_LEFT
- GL_BACK_RIGHT
- GL_FRONT
- GL_BACK
- GL_LEFT
- GL_RIGHT
- GL_FRONT_AND_BACK
- GL_AUX0

- GL_AUX1
- GL_AUX2
- GL_AUX3
- GL_INVALID_ENUM
- GL_INVALID_VALUE
- GL_INVALID_OPERATION
- GL_STACK_OVERFLOW
- GL_STACK_UNDERFLOW
- GL_OUT_OF_MEMORY
- GL_2D
- GL_3D
- GL_3D_COLOR
- GL_3D_COLOR_TEXTURE
- GL_4D_COLOR_TEXTURE
- GL_PASS_THROUGH_TOKEN
- GL_POINT_TOKEN
- GL_LINE_TOKEN
- GL_POLYGON_TOKEN
- GL_BITMAP_TOKEN
- GL_DRAW_PIXEL_TOKEN
- GL_COPY_PIXEL_TOKEN
- GL_LINE_RESET_TOKEN
- GL_EXP
- GL_VIEWPORT_BIT
- GL_EXP2
- GL_CW
- GL_CCW
- GL_COEFF
- GL_ORDER
- GL_DOMAIN
- GL_CURRENT_COLOR
- GL_CURRENT_INDEX
- GL_CURRENT_NORMAL
- GL_CURRENT_TEXTURE_COORDS
- GL_CURRENT_RASTER_COLOR
- GL_CURRENT_RASTER_INDEX

- GL_CURRENT_RASTER_TEXTURE_COORDS
- GL_CURRENT_RASTER_POSITION
- GL_CURRENT_RASTER_POSITION_VALID
- GL_CURRENT_RASTER_DISTANCE
- GL_POINT_SMOOTH
- GL_POINT_SIZE
- GL_POINT_SIZE_RANGE
- GL_POINT_SIZE_GRANULARITY
- GL_LINE_SMOOTH
- GL_LINE_WIDTH
- GL_LINE_WIDTH_RANGE
- GL_LINE_WIDTH_GRANULARITY
- GL_LINE_STIPPLE
- GL_LINE_STIPPLE_PATTERN
- GL_LINE_STIPPLE_REPEAT
- GL_LIST_MODE
- GL_MAX_LIST_NESTING
- GL_LIST_BASE
- GL_LIST_INDEX
- GL_POLYGON_MODE
- GL_POLYGON_SMOOTH
- GL_POLYGON_STIPPLE
- GL_EDGE_FLAG
- GL_CULL_FACE
- GL_CULL_FACE_MODE
- GL_FRONT_FACE
- GL_LIGHTING
- GL_LIGHT_MODEL_LOCAL_VIEWER
- GL_LIGHT_MODEL_TWO_SIDE
- GL_LIGHT_MODEL_AMBIENT
- GL_SHADE_MODEL
- GL_COLOR_MATERIAL_FACE
- GL_COLOR_MATERIAL_PARAMETER
- GL_COLOR_MATERIAL
- GL_FOG
- GL_FOG_INDEX

- GL_FOG_DENSITY
- GL_FOG_START
- GL_FOG_END
- GL_FOG_MODE
- GL_FOG_COLOR
- GL_DEPTH_RANGE
- GL_DEPTH_TEST
- GL_DEPTH_WRITEMASK
- GL_DEPTH_CLEAR_VALUE
- GL_DEPTH_FUNC
- GL_ACCUM_CLEAR_VALUE
- GL_STENCIL_TEST
- GL_STENCIL_CLEAR_VALUE
- GL_STENCIL_FUNC
- GL_STENCIL_VALUE_MASK
- GL_STENCIL_FAIL
- GL_STENCIL_PASS_DEPTH_FAIL
- GL_STENCIL_PASS_DEPTH_PASS
- GL_STENCIL_REF
- GL_STENCIL_WRITEMASK
- GL_MATRIX_MODE
- GL_NORMALIZE
- GL_VIEWPORT
- GL_MODELVIEW_STACK_DEPTH
- GL_PROJECTION_STACK_DEPTH
- GL_TEXTURE_STACK_DEPTH
- GL_MODELVIEW_MATRIX
- GL_PROJECTION_MATRIX
- GL_TEXTURE_MATRIX
- GL_ATTRIB_STACK_DEPTH
- GL_CLIENT_ATTRIB_STACK_DEPTH
- GL_ALPHA_TEST
- GL_ALPHA_TEST_FUNC
- GL_ALPHA_TEST_REF
- GL_DITHER
- GL_BLEND_DST

- GL_BLEND_SRC
- GL_BLEND
- GL_LOGIC_OP_MODE
- GL_INDEX_LOGIC_OP
- GL_COLOR_LOGIC_OP
- GL_AUX_BUFFERS
- GL_DRAW_BUFFER
- GL_READ_BUFFER
- GL_SCISSOR_BOX
- GL_SCISSOR_TEST
- GL_INDEX_CLEAR_VALUE
- GL_INDEX_WRITEMASK
- GL_COLOR_CLEAR_VALUE
- GL_COLOR_WRITEMASK
- GL_INDEX_MODE
- GL_RGBA_MODE
- GL_DOUBLEBUFFER
- GL_STEREO
- GL_RENDER_MODE
- GL_PERSPECTIVE_CORRECTION_HINT
- GL_POINT_SMOOTH_HINT
- GL_LINE_SMOOTH_HINT
- GL_POLYGON_SMOOTH_HINT
- GL_FOG_HINT
- GL_TEXTURE_GEN_S
- GL_TEXTURE_GEN_T
- GL_TEXTURE_GEN_R
- GL_TEXTURE_GEN_Q
- GL_PIXEL_MAP_I_TO_I
- GL_PIXEL_MAP_S_TO_S
- GL_PIXEL_MAP_I_TO_R
- GL_PIXEL_MAP_I_TO_G
- GL_PIXEL_MAP_I_TO_B
- GL_PIXEL_MAP_I_TO_A
- GL_PIXEL_MAP_R_TO_R
- GL_PIXEL_MAP_G_TO_G

- GL_PIXEL_MAP_B_TO_B
- GL_PIXEL_MAP_A_TO_A
- GL_PIXEL_MAP_I_TO_I_SIZE
- GL_PIXEL_MAP_S_TO_S_SIZE
- GL_PIXEL_MAP_I_TO_R_SIZE
- GL_PIXEL_MAP_I_TO_G_SIZE
- GL_PIXEL_MAP_I_TO_B_SIZE
- GL_PIXEL_MAP_I_TO_A_SIZE
- GL_PIXEL_MAP_R_TO_R_SIZE
- GL_PIXEL_MAP_G_TO_G_SIZE
- GL_PIXEL_MAP_B_TO_B_SIZE
- GL_PIXEL_MAP_A_TO_A_SIZE
- GL_UNPACK_SWAP_BYTES
- GL_UNPACK_LSB_FIRST
- GL_UNPACK_ROW_LENGTH
- GL_UNPACK_SKIP_ROWS
- GL_UNPACK_SKIP_PIXELS
- GL_UNPACK_ALIGNMENT
- GL_PACK_SWAP_BYTES
- GL_PACK_LSB_FIRST
- GL_PACK_ROW_LENGTH
- GL_PACK_SKIP_ROWS
- GL_PACK_SKIP_PIXELS
- GL_PACK_ALIGNMENT
- GL_MAP_COLOR
- GL_MAP_STENCIL
- GL_INDEX_SHIFT
- GL_INDEX_OFFSET
- GL_RED_SCALE
- GL_RED_BIAS
- GL_ZOOM_X
- GL_ZOOM_Y
- GL_GREEN_SCALE
- GL_GREEN_BIAS
- GL_BLUE_SCALE
- GL_BLUE_BIAS

- GL_ALPHA_SCALE
- GL_ALPHA_BIAS
- GL_DEPTH_SCALE
- GL_DEPTH_BIAS
- GL_MAX_EVAL_ORDER
- GL_MAX_LIGHTS
- GL_MAX_CLIP_PLANES
- GL_MAX_TEXTURE_SIZE
- GL_MAX_PIXEL_MAP_TABLE
- GL_MAX_ATTRIB_STACK_DEPTH
- GL_MAX_MODELVIEW_STACK_DEPTH
- GL_MAX_NAME_STACK_DEPTH
- GL_MAX_PROJECTION_STACK_DEPTH
- GL_MAX_TEXTURE_STACK_DEPTH
- GL_MAX_VIEWPORT_DIMS
- GL_MAX_CLIENT_ATTRIB_STACK_DEPTH
- GL_SUBPIXEL_BITS
- GL_INDEX_BITS
- GL_RED_BITS
- GL_GREEN_BITS
- GL_BLUE_BITS
- GL_ALPHA_BITS
- GL_DEPTH_BITS
- GL_STENCIL_BITS
- GL_ACCUM_RED_BITS
- GL_ACCUM_GREEN_BITS
- GL_ACCUM_BLUE_BITS
- GL_ACCUM_ALPHA_BITS
- GL_NAME_STACK_DEPTH
- GL_AUTO_NORMAL
- GL_MAP1_COLOR_4
- GL_MAP1_INDEX
- GL_MAP1_NORMAL
- GL_MAP1_TEXTURE_COORD_1
- GL_MAP1_TEXTURE_COORD_2
- GL_MAP1_TEXTURE_COORD_3

- GL_MAP1_TEXTURE_COORD_4
- GL_MAP1_VERTEX_3
- GL_MAP1_VERTEX_4
- GL_MAP2_COLOR_4
- GL_MAP2_INDEX
- GL_MAP2_NORMAL
- GL_MAP2_TEXTURE_COORD_1
- GL_MAP2_TEXTURE_COORD_2
- GL_MAP2_TEXTURE_COORD_3
- GL_MAP2_TEXTURE_COORD_4
- GL_MAP2_VERTEX_3
- GL_MAP2_VERTEX_4
- GL_MAP1_GRID_DOMAIN
- GL_MAP1_GRID_SEGMENTS
- GL_MAP2_GRID_DOMAIN
- GL_MAP2_GRID_SEGMENTS
- GL_TEXTURE_1D
- GL_TEXTURE_2D
- GL_FEEDBACK_BUFFER_POINTER
- GL_FEEDBACK_BUFFER_SIZE
- GL_FEEDBACK_BUFFER_TYPE
- GL_SELECTION_BUFFER_POINTER
- GL_SELECTION_BUFFER_SIZE
- GL_TEXTURE_WIDTH
- GL_TRANSFORM_BIT
- GL_TEXTURE_HEIGHT
- GL_TEXTURE_INTERNAL_FORMAT
- GL_TEXTURE_BORDER_COLOR
- GL_TEXTURE_BORDER
- GL_DONT_CARE
- GL_FASTEST
- GL_NICEST
- GL_AMBIENT
- GL_DIFFUSE
- GL_SPECULAR
- GL_POSITION

- GL_SPOT_DIRECTION
- GL_SPOT_EXPONENT
- GL_SPOT_CUTOFF
- GL_CONSTANT_ATTENUATION
- GL_LINEAR_ATTENUATION
- GL_QUADRATIC_ATTENUATION
- GL_COMPILE
- GL_COMPILE_AND_EXECUTE
- GL_BYTE
- GL_UNSIGNED_BYTE
- GL_SHORT
- GL_UNSIGNED_SHORT
- GL_INT
- GL_UNSIGNED_INT
- GL_FLOAT
- GL_2_BYTES
- GL_3_BYTES
- GL_4_BYTES
- GL_DOUBLE
- GL_CLEAR
- GL_AND
- GL_AND_REVERSE
- GL_COPY
- GL_AND_INVERTED
- GL_NOOP
- GL_XOR
- GL_OR
- GL_NOR
- GL_EQUIV
- GL_INVERT
- GL_OR_REVERSE
- GL_COPY_INVERTED
- GL_OR_INVERTED
- GL_NAND
- GL_SET
- GL_EMISSION

- GL_SHININESS
- GL_AMBIENT_AND_DIFFUSE
- GL_COLOR_INDEXES
- GL_MODELVIEW
- GL_PROJECTION
- GL_TEXTURE
- GL_COLOR
- GL_DEPTH
- GL_STENCIL
- GL_COLOR_INDEX
- GL_STENCIL_INDEX
- GL_DEPTH_COMPONENT
- GL_RED
- GL_GREEN
- GL_BLUE
- GL_ALPHA
- GL_RGB
- GL_RGBA
- GL_LUMINANCE
- GL_LUMINANCE_ALPHA
- GL_BITMAP
- GL_POINT
- GL_LINE
- GL_FILL
- GL_RENDER
- GL_FEEDBACK
- GL_SELECT
- GL_FLAT
- GL_SMOOTH
- GL_KEEP
- GL_REPLACE
- GL_INCR
- GL_DECR
- GL_VENDOR
- GL_RENDERER
- GL_VERSION

- GL_EXTENSIONS
- GL_S
- GL_ENABLE_BIT
- GL_T
- GL_R
- GL_Q
- GL_MODULATE
- GL_DECAL
- GL_TEXTURE_ENV_MODE
- GL_TEXTURE_ENV_COLOR
- GL_TEXTURE_ENV
- GL_EYE_LINEAR
- GL_OBJECT_LINEAR
- GL_SPHERE_MAP
- GL_TEXTURE_GEN_MODE
- GL_OBJECT_PLANE
- GL_EYE_PLANE
- GL_NEAREST
- GL_LINEAR
- GL_NEAREST_MIPMAP_NEAREST
- GL_LINEAR_MIPMAP_NEAREST
- GL_NEAREST_MIPMAP_LINEAR
- GL_LINEAR_MIPMAP_LINEAR
- GL_TEXTURE_MAG_FILTER
- GL_TEXTURE_MIN_FILTER
- GL_TEXTURE_WRAP_S
- GL_TEXTURE_WRAP_T
- GL_CLAMP
- GL_REPEAT
- GL_POLYGON_OFFSET_UNITS
- GL_POLYGON_OFFSET_POINT
- GL_POLYGON_OFFSET_LINE
- GL_R3_G3_B2
- GL_V2F
- GL_V3F
- GL_C4UB_V2F

- GL_C4UB_V3F
- GL_C3F_V3F
- GL_N3F_V3F
- GL_C4F_N3F_V3F
- GL_T2F_V3F
- GL_T4F_V4F
- GL_T2F_C4UB_V3F
- GL_T2F_C3F_V3F
- GL_T2F_N3F_V3F
- GL_T2F_C4F_N3F_V3F
- GL_T4F_C4F_N3F_V4F
- GL_CLIP_PLANE0
- GL_CLIP_PLANE1
- GL_CLIP_PLANE2
- GL_CLIP_PLANE3
- GL_CLIP_PLANE4
- GL_CLIP_PLANE5
- GL_LIGHT0
- GL_COLOR_BUFFER_BIT
- GL_LIGHT1
- GL_LIGHT2
- GL_LIGHT3
- GL_LIGHT4
- GL_LIGHT5
- GL_LIGHT6
- GL_LIGHT7
- GL_HINT_BIT
- GL_POLYGON_OFFSET_FILL
- GL_POLYGON_OFFSET_FACTOR
- GL_ALPHA4
- GL_ALPHA8
- GL_ALPHA12
- GL_ALPHA16
- GL_LUMINANCE4
- GL_LUMINANCE8
- GL_LUMINANCE12

- GL_LUMINANCE16
- GL_LUMINANCE4_ALPHA4
- GL_LUMINANCE6_ALPHA2
- GL_LUMINANCE8_ALPHA8
- GL_LUMINANCE12_ALPHA4
- GL_LUMINANCE12_ALPHA12
- GL_LUMINANCE16_ALPHA16
- GL_INTENSITY
- GL_INTENSITY4
- GL_INTENSITY8
- GL_INTENSITY12
- GL_INTENSITY16
- GL_RGB4
- GL_RGB5
- GL_RGB8
- GL_RGB10
- GL_RGB12
- GL_RGB16
- GL_RGBA2
- GL_RGBA4
- GL_RGB5_A1
- GL_RGBA8
- GL_RGB10_A2
- GL_RGBA12
- GL_RGBA16
- GL_TEXTURE_RED_SIZE
- GL_TEXTURE_GREEN_SIZE
- GL_TEXTURE_BLUE_SIZE
- GL_TEXTURE_ALPHA_SIZE
- GL_TEXTURE_LUMINANCE_SIZE
- GL_TEXTURE_INTENSITY_SIZE
- GL_PROXY_TEXTURE_1D
- GL_PROXY_TEXTURE_2D
- GL_TEXTURE_PRIORITY
- GL_TEXTURE_RESIDENT
- GL_TEXTURE_BINDING_1D

- GL_TEXTURE_BINDING_2D
- GL_VERTEX_ARRAY
- GL_NORMAL_ARRAY
- GL_COLOR_ARRAY
- GL_INDEX_ARRAY
- GL_TEXTURE_COORD_ARRAY
- GL_EDGE_FLAG_ARRAY
- GL_VERTEX_ARRAY_SIZE
- GL_VERTEX_ARRAY_TYPE
- GL_VERTEX_ARRAY_STRIDE
- GL_NORMAL_ARRAY_TYPE
- GL_NORMAL_ARRAY_STRIDE
- GL_COLOR_ARRAY_SIZE
- GL_COLOR_ARRAY_TYPE
- GL_COLOR_ARRAY_STRIDE
- GL_INDEX_ARRAY_TYPE
- GL_INDEX_ARRAY_STRIDE
- GL_TEXTURE_COORD_ARRAY_SIZE
- GL_TEXTURE_COORD_ARRAY_TYPE
- GL_TEXTURE_COORD_ARRAY_STRIDE
- GL_EDGE_FLAG_ARRAY_STRIDE
- GL_VERTEX_ARRAY_POINTER
- GL_NORMAL_ARRAY_POINTER
- GL_COLOR_ARRAY_POINTER
- GL_INDEX_ARRAY_POINTER
- GL_TEXTURE_COORD_ARRAY_POINTER
- GL_EDGE_FLAG_ARRAY_POINTER
- GL_COLOR_INDEX1_EXT
- GL_COLOR_INDEX2_EXT
- GL_COLOR_INDEX4_EXT
- GL_COLOR_INDEX8_EXT
- GL_COLOR_INDEX12_EXT
- GL_COLOR_INDEX16_EXT
- GL_EVAL_BIT
- GL_LIST_BIT
- GL_TEXTURE_BIT

- GL_SCISSOR_BIT
- GL_ALL_ATTRIB_BITS
- GL_CLIENT_ALL_ATTRIB_BITS
- GL_SMOOTH_POINT_SIZE_RANGE
- GL_SMOOTH_POINT_SIZE_GRANULARITY
- GL_SMOOTH_LINE_WIDTH_RANGE
- GL_SMOOTH_LINE_WIDTH_GRANULARITY
- GL_UNSIGNED_BYTE_3_3_2
- GL_UNSIGNED_SHORT_4_4_4_4
- GL_UNSIGNED_SHORT_5_5_5_1
- GL_UNSIGNED_INT_8_8_8_8
- GL_UNSIGNED_INT_10_10_10_2
- GL_RESCALE_NORMAL
- GL_TEXTURE_BINDING_3D
- GL_PACK_SKIP_IMAGES
- GL_PACK_IMAGE_HEIGHT
- GL_UNPACK_SKIP_IMAGES
- GL_UNPACK_IMAGE_HEIGHT
- GL_TEXTURE_3D
- GL_PROXY_TEXTURE_3D
- GL_TEXTURE_DEPTH
- GL_TEXTURE_WRAP_R
- GL_MAX_3D_TEXTURE_SIZE
- GL_BGR
- GL_BGRA
- GL_MAX_ELEMENTS_VERTICES
- GL_MAX_ELEMENTS_INDICES
- GL_CLAMP_TO_EDGE
- GL_TEXTURE_MIN_LOD
- GL_TEXTURE_MAX_LOD
- GL_TEXTURE_BASE_LEVEL
- GL_TEXTURE_MAX_LEVEL
- GL_LIGHT_MODEL_COLOR_CONTROL
- GL_SINGLE_COLOR
- GL_SEPARATE_SPECULAR_COLOR
- GL_UNSIGNED_BYTE_2_3_3_REV

- GL_UNSIGNED_SHORT_5_6_5
- GL_UNSIGNED_SHORT_5_6_5_REV
- GL_UNSIGNED_SHORT_4_4_4_REV
- GL_UNSIGNED_SHORT_1_5_5_5_REV
- GL_UNSIGNED_INT_8_8_8_8_REV
- GL_ALIASED_POINT_SIZE_RANGE
- GL_ALIASED_LINE_WIDTH_RANGE
- GL_MULTISAMPLE
- GL_SAMPLE_ALPHA_TO_COVERAGE
- GL_SAMPLE_ALPHA_TO_ONE
- GL_SAMPLE_COVERAGE
- GL_SAMPLE_BUFFERS
- GL_SAMPLES
- GL_SAMPLE_COVERAGE_VALUE
- GL_SAMPLE_COVERAGE_INVERT
- GL_CLAMP_TO_BORDER
- GL_TEXTURE0
- GL_TEXTURE1
- GL_TEXTURE2
- GL_TEXTURE3
- GL_TEXTURE4
- GL_TEXTURE5
- GL_TEXTURE6
- GL_TEXTURE7
- GL_TEXTURE8
- GL_TEXTURE9
- GL_TEXTURE10
- GL_TEXTURE11
- GL_TEXTURE12
- GL_TEXTURE13
- GL_TEXTURE14
- GL_TEXTURE15
- GL_TEXTURE16
- GL_TEXTURE17
- GL_TEXTURE18
- GL_TEXTURE19

- GL_TEXTURE20
- GL_TEXTURE21
- GL_TEXTURE22
- GL_TEXTURE23
- GL_TEXTURE24
- GL_TEXTURE25
- GL_TEXTURE26
- GL_TEXTURE27
- GL_TEXTURE28
- GL_TEXTURE29
- GL_TEXTURE30
- GL_TEXTURE31
- GL_ACTIVE_TEXTURE
- GL_CLIENT_ACTIVE_TEXTURE
- GL_MAX_TEXTURE_UNITS
- GL_TRANSPOSE_MODELVIEW_MATRIX
- GL_TRANSPOSE_PROJECTION_MATRIX
- GL_TRANSPOSE_TEXTURE_MATRIX
- GL_TRANSPOSE_COLOR_MATRIX
- GL_SUBTRACT
- GL_COMPRESSED_ALPHA
- GL_COMPRESSED_LUMINANCE
- GL_COMPRESSED_LUMINANCE_ALPHA
- GL_COMPRESSED_INTENSITY
- GL_COMPRESSED_RGB
- GL_COMPRESSED_RGBA
- GL_TEXTURE_COMPRESSION_HINT
- GL_NORMAL_MAP
- GL_REFLECTION_MAP
- GL_TEXTURE_CUBE_MAP
- GL_TEXTURE_BINDING_CUBE_MAP
- GL_TEXTURE_CUBE_MAP_POSITIVE_X
- GL_TEXTURE_CUBE_MAP_NEGATIVE_X
- GL_TEXTURE_CUBE_MAP_POSITIVE_Y
- GL_TEXTURE_CUBE_MAP_NEGATIVE_Y
- GL_TEXTURE_CUBE_MAP_POSITIVE_Z

- GL_TEXTURE_CUBE_MAP_NEGATIVE_Z
- GL_PROXY_TEXTURE_CUBE_MAP
- GL_MAX_CUBE_MAP_TEXTURE_SIZE
- GL_COMBINE
- GL_COMBINE_RGB
- GL_COMBINE_ALPHA
- GL_RGB_SCALE
- GL_ADD_SIGNED
- GL_INTERPOLATE
- GL_CONSTANT
- GL_PRIMARY_COLOR
- GL_PREVIOUS
- GL_SOURCE0_RGB
- GL_SOURCE1_RGB
- GL_SOURCE2_RGB
- GL_SOURCE0_ALPHA
- GL_SOURCE1_ALPHA
- GL_SOURCE2_ALPHA
- GL_OPERAND0_RGB
- GL_OPERAND1_RGB
- GL_OPERAND2_RGB
- GL_OPERAND0_ALPHA
- GL_OPERAND1_ALPHA
- GL_OPERAND2_ALPHA
- GL_TEXTURE_COMPRESSED_IMAGE_SIZE
- GL_TEXTURE_COMPRESSED
- GL_NUM_COMPRESSED_TEXTURE_FORMATS
- GL_COMPRESSED_TEXTURE_FORMATS
- GL_DOT3_RGB
- GL_DOT3_RGBA
- GL_MULTISAMPLE_BIT
- GL_BLEND_DST_RGB
- GL_BLEND_SRC_RGB
- GL_BLEND_DST_ALPHA
- GL_BLEND_SRC_ALPHA
- GL_POINT_SIZE_MIN

- GL_POINT_SIZE_MAX
- GL_POINT_FADE_THRESHOLD_SIZE
- GL_POINT_DISTANCE_ATTENUATION
- GL_GENERATE_MIPMAP
- GL_GENERATE_MIPMAP_HINT
- GL_DEPTH_COMPONENT16
- GL_DEPTH_COMPONENT24
- GL_DEPTH_COMPONENT32
- GL_MIRRORED_REPEAT
- GL_FOG_COORDINATE_SOURCE
- GL_FOG_COORDINATE
- GL_FRAGMENT_DEPTH
- GL_CURRENT_FOG_COORDINATE
- GL_FOG_COORDINATE_ARRAY_TYPE
- GL_FOG_COORDINATE_ARRAY_STRIDE
- GL_FOG_COORDINATE_ARRAY_POINTER
- GL_FOG_COORDINATE_ARRAY
- GL_COLOR_SUM
- GL_CURRENT_SECONDARY_COLOR
- GL_SECONDARY_COLOR_ARRAY_SIZE
- GL_SECONDARY_COLOR_ARRAY_TYPE
- GL_SECONDARY_COLOR_ARRAY_STRIDE
- GL_SECONDARY_COLOR_ARRAY_POINTER
- GL_SECONDARY_COLOR_ARRAY
- GL_MAX_TEXTURE_LOD_BIAS
- GL_TEXTURE_FILTER_CONTROL
- GL_TEXTURE_LOD_BIAS
- GL_INCR_WRAP
- GL_DECR_WRAP
- GL_TEXTURE_DEPTH_SIZE
- GL_DEPTH_TEXTURE_MODE
- GL_TEXTURE_COMPARE_MODE
- GL_TEXTURE_COMPARE_FUNC
- GL_COMPARE_R_TO_TEXTURE
- GL_CURRENT_FOG_COORD
- GL_FOG_COORD

- GL_FOG_COORD_ARRAY
- GL_FOG_COORD_ARRAY_BUFFER_BINDING
- GL_FOG_COORD_ARRAY_POINTER
- GL_FOG_COORD_ARRAY_STRIDE
- GL_FOG_COORD_ARRAY_TYPE
- GL_FOG_COORD_SRC
- GL_SRC0_ALPHA
- GL_SRC0_RGB
- GL_SRC1_ALPHA
- GL_SRC1_RGB
- GL_SRC2_ALPHA
- GL_SRC2_RGB
- GL_BUFFER_SIZE
- GL_BUFFER_USAGE
- GL_QUERY_COUNTER_BITS
- GL_CURRENT_QUERY
- GL_QUERY_RESULT
- GL_QUERY_RESULT_AVAILABLE
- GL_ARRAY_BUFFER
- GL_ELEMENT_ARRAY_BUFFER
- GL_ARRAY_BUFFER_BINDING
- GL_ELEMENT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ARRAY_BUFFER_BINDING
- GL_NORMAL_ARRAY_BUFFER_BINDING
- GL_COLOR_ARRAY_BUFFER_BINDING
- GL_INDEX_ARRAY_BUFFER_BINDING
- GL_TEXTURE_COORD_ARRAY_BUFFER_BINDING
- GL_EDGE_FLAG_ARRAY_BUFFER_BINDING
- GL_SECONDARY_COLOR_ARRAY_BUFFER_BINDING
- GL_FOG_COORDINATE_ARRAY_BUFFER_BINDING
- GL_WEIGHT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ATTRIB_ARRAY_BUFFER_BINDING
- GL_READ_ONLY
- GL_WRITE_ONLY
- GL_READ_WRITE
- GL_BUFFER_ACCESS

- GL_BUFFER_MAPPED
- GL_BUFFER_MAP_POINTER
- GL_STREAM_DRAW
- GL_STREAM_READ
- GL_STREAM_COPY
- GL_STATIC_DRAW
- GL_STATIC_READ
- GL_STATIC_COPY
- GL_DYNAMIC_DRAW
- GL_DYNAMIC_READ
- GL_DYNAMIC_COPY
- GL_SAMPLES_PASSED
- GL_BLEND_EQUATION_RGB
- GL_VERTEX_ATTRIB_ARRAY_ENABLED
- GL_VERTEX_ATTRIB_ARRAY_SIZE
- GL_VERTEX_ATTRIB_ARRAY_STRIDE
- GL_VERTEX_ATTRIB_ARRAY_TYPE
- GL_CURRENT_VERTEX_ATTRIB
- GL_VERTEX_PROGRAM_POINT_SIZE
- GL_VERTEX_PROGRAM_TWO_SIDE
- GL_VERTEX_ATTRIB_ARRAY_POINTER
- GL_STENCIL_BACK_FUNC
- GL_STENCIL_BACK_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_PASS
- GL_MAX_DRAW_BUFFERS
- GL_DRAW_BUFFER0
- GL_DRAW_BUFFER1
- GL_DRAW_BUFFER2
- GL_DRAW_BUFFER3
- GL_DRAW_BUFFER4
- GL_DRAW_BUFFER5
- GL_DRAW_BUFFER6
- GL_DRAW_BUFFER7
- GL_DRAW_BUFFER8
- GL_DRAW_BUFFER9

- GL_DRAW_BUFFER10
- GL_DRAW_BUFFER11
- GL_DRAW_BUFFER12
- GL_DRAW_BUFFER13
- GL_DRAW_BUFFER14
- GL_DRAW_BUFFER15
- GL_BLEND_EQUATION_ALPHA
- GL_POINT_SPRITE
- GL_COORD_REPLACE
- GL_MAX_VERTEX_ATTRIBS
- GL_VERTEX_ATTRIB_ARRAY_NORMALIZED
- GL_MAX_TEXTURE_COORDS
- GL_MAX_TEXTURE_IMAGE_UNITS
- GL_FRAGMENT_SHADER
- GL_VERTEX_SHADER
- GL_MAX_FRAGMENT_UNIFORM_COMPONENTS
- GL_MAX_VERTEX_UNIFORM_COMPONENTS
- GL_MAX_VARYING_FLOATS
- GL_MAX_VERTEX_TEXTURE_IMAGE_UNITS
- GL_MAX_COMBINED_TEXTURE_IMAGE_UNITS
- GL_SHADER_TYPE
- GL_FLOAT_VEC2
- GL_FLOAT_VEC3
- GL_FLOAT_VEC4
- GL_INT_VEC2
- GL_INT_VEC3
- GL_INT_VEC4
- GL_BOOL
- GL_BOOL_VEC2
- GL_BOOL_VEC3
- GL_BOOL_VEC4
- GL_FLOAT_MAT2
- GL_FLOAT_MAT3
- GL_FLOAT_MAT4
- GL_SAMPLER_1D
- GL_SAMPLER_2D

- GL_SAMPLER_3D
- GL_SAMPLER_CUBE
- GL_SAMPLER_1D_SHADOW
- GL_SAMPLER_2D_SHADOW
- GL_DELETE_STATUS
- GL_COMPILE_STATUS
- GL_LINK_STATUS
- GL_VALIDATE_STATUS
- GL_INFO_LOG_LENGTH
- GL_ATTACHED_SHADERS
- GL_ACTIVE_UNIFORMS
- GL_ACTIVE_UNIFORM_MAX_LENGTH
- GL_SHADER_SOURCE_LENGTH
- GL_ACTIVE_ATTRIBUTES
- GL_ACTIVE_ATTRIBUTE_MAX_LENGTH
- GL_FRAGMENT_SHADER_DERIVATIVE_HINT
- GL_SHADING_LANGUAGE_VERSION
- GL_CURRENT_PROGRAM
- GL_POINT_SPRITE_COORD_ORIGIN
- GL_LOWER_LEFT
- GL_UPPER_LEFT
- GL_STENCIL_BACK_REF
- GL_STENCIL_BACK_VALUE_MASK
- GL_STENCIL_BACK_WRITEMASK
- void glAccum(GLenum op, GLfloat value)
- void glAlphaFunc(GLenum func, GLclampf ref)
- GLboolean glAreTexturesResident(GLsizei n, const GLuint *textures, GLboolean *residences)
- void glArrayElement(GLint i)
- void glBegin(GLenum mode)
- void glBindTexture(GLenum target, GLuint texture)
- void glBitmap(GLsizei width, GLsizei height, GLfloat xorig, GLfloat yorig, GLfloat xmove, GLfloat ymove, const GLubyte *bitmap)
- void glBlendFunc(GLenum sfactor, GLenum dfactor)
- void glCallList(GLuint list)
- void glCallLists(GLsizei n, GLenum type, const void *lists)
- void glClear(GLbitfield mask)

- void glClearAccum(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glClearColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)
- void glClearDepth(GLclampd depth)
- void glClearIndex(GLfloat c)
- void glClearStencil(GLint s)
- void glClipPlane(GLenum plane, const GLdouble *equation)
- void glColor3b(GLbyte red, GLbyte green, GLbyte blue)
- void glColor3bv(const GLbyte *v)
- void glColor3d(GLdouble red, GLdouble green, GLdouble blue)
- void glColor3dv(const GLdouble *v)
- void glColor3f(GLfloat red, GLfloat green, GLfloat blue)
- void glColor3fv(const GLfloat *v)
- void glColor3i(GLint red, GLint green, GLint blue)
- void glColor3iv(const GLint *v)
- void glColor3s(GLshort red, GLshort green, GLshort blue)
- void glColor3sv(const GLshort *v)
- void glColor3ub(GLubyte red, GLubyte green, GLubyte blue)
- void glColor3ubv(const GLubyte *v)
- void glColor3ui(GLuint red, GLuint green, GLuint blue)
- void glColor3uiv(const GLuint *v)
- void glColor3us(GLushort red, GLushort green, GLushort blue)
- void glColor3usv(const GLushort *v)
- void glColor4b(GLbyte red, GLbyte green, GLbyte blue, GLbyte alpha)
- void glColor4bv(const GLbyte *v)
- void glColor4d(GLdouble red, GLdouble green, GLdouble blue, GLdouble alpha)
- void glColor4dv(const GLdouble *v)
- void glColor4f(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glColor4fv(const GLfloat *v)
- void glColor4i(GLint red, GLint green, GLint blue, GLint alpha)
- void glColor4iv(const GLint *v)
- void glColor4s(GLshort red, GLshort green, GLshort blue, GLshort alpha)
- void glColor4sv(const GLshort *v)
- void glColor4ub(GLubyte red, GLubyte green, GLubyte blue, GLubyte alpha)
- void glColor4ubv(const GLubyte *v)
- void glColor4ui(GLuint red, GLuint green, GLuint blue, GLuint alpha)
- void glColor4uiv(const GLuint *v)

- void glColor4us(GLushort red, GLushort green, GLushort blue, GLushort alpha)
- void glColor4usv(const GLushort *v)
- void glColorMask(GLboolean red, GLboolean green, GLboolean blue, GLboolean alpha)
- void glColorMaterial(GLenum face, GLenum mode)
- void glColorPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)
- void glCopyPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum type)
- void glCopyTexImage1D(GLenum target, GLint level, GLenum internalFormat, GLint x, GLint y, GLsizei width, GLint border)
- void glCopyTexImage2D(GLenum target, GLint level, GLenum internalFormat, GLint x, GLint y, GLsizei width, GLsizei height, GLint border)
- void glCopyTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLint x, GLint y, GLsizei width)
- void glCopyTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint x, GLint y, GLsizei width, GLsizei height)
- void glCullFace(GLenum mode)
- void glDeleteLists(GLuint list, GLsizei range)
- void glDeleteTextures(GLsizei n, const GLuint *textures)
- void glDepthFunc(GLenum func)
- void glDepthMask(GLboolean flag)
- void glDepthRange(GLclampd zNear, GLclampd zFar)
- void glDisable(GLenum cap)
- void glDisableClientState(GLenum array)
- void glDrawArrays(GLenum mode, GLint first, GLsizei count)
- void glDrawBuffer(GLenum mode)
- void glDrawElements(GLenum mode, GLsizei count, GLenum type, const void *indices)
- void glDrawPixels(GLsizei width, GLsizei height, GLenum format, GLenum type, const void *pixels)
- void glEdgeFlag(GLboolean flag)
- void glEdgeFlagPointer(GLsizei stride, const void *pointer)
- void glEdgeFlagv(const GLboolean *flag)
- void glEnable(GLenum cap)
- void glEnableClientState(GLenum array)
- void glEnd(void)
- void glEndList(void)
- void glEvalCoord1d(GLdouble u)
- void glEvalCoord1dv(const GLdouble *u)
- void glEvalCoord1f(GLfloat u)
- void glEvalCoord1fv(const GLfloat *u)
- void glEvalCoord2d(GLdouble u, GLdouble v)

- void glEvalCoord2dv(const GLdouble *u)
- void glEvalCoord2f(GLfloat u, GLfloat v)
- void glEvalCoord2fv(const GLfloat *u)
- void glEvalMesh1(GLenum mode, GLint i1, GLint i2)
- void glEvalMesh2(GLenum mode, GLint i1, GLint i2, GLint j1, GLint j2)
- void glEvalPoint1(GLint i)
- void glEvalPoint2(GLint i, GLint j)
- void glFeedbackBuffer(GLsizei size, GLenum type, GLfloat *buffer)
- void glFinish(void)
- void glFlush(void)
- void glFogf(GLenum pname, GLfloat param)
- void glFogfv(GLenum pname, const GLfloat *params)
- void glFogi(GLenum pname, GLint param)
- void glFogiv(GLenum pname, const GLint *params)
- void glFrontFace(GLenum mode)
- void glFrustum(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble zNear, GLdouble zFar)
- GLuint glGenLists(GLsizei range)
- void glGenTextures(GLsizei n, GLuint *textures)
- void glGetBooleanv(GLenum pname, GLboolean *params)
- void glGetClipPlane(GLenum plane, GLdouble *equation)
- void glGetDoublev(GLenum pname, GLdouble *params)
- GLenum glGetError(void)
- void glGetFloatv(GLenum pname, GLfloat *params)
- void glGetIntegerv(GLenum pname, GLint *params)
- void glGetLightfv(GLenum light, GLenum pname, GLfloat *params)
- void glGetLightiv(GLenum light, GLenum pname, GLint *params)
- void glGetMapdv(GLenum target, GLenum query, GLdouble *v)
- void glGetMapfv(GLenum target, GLenum query, GLfloat *v)
- void glGetMapiv(GLenum target, GLenum query, GLint *v)
- void glGetMaterialfv(GLenum face, GLenum pname, GLfloat *params)
- void glGetMaterialiv(GLenum face, GLenum pname, GLint *params)
- void glGetPixelMapfv(GLenum map, GLfloat *values)
- void glGetPixelMapuiv(GLenum map, GLuint *values)
- void glGetPixelMapusv(GLenum map, GLushort *values)
- void glGetPointerv(GLenum pname, void* *params)

- void glGetPolygonStipple(GLubyte *mask)
- GLubyte * glGetString(GLenum name)
- void glGetTexEnvfv(GLenum target, GLenum pname, GLfloat *params)
- void glGetTexEnviv(GLenum target, GLenum pname, GLint *params)
- void glGetTexGendv(GLenum coord, GLenum pname, GLdouble *params)
- void glGetTexGenfv(GLenum coord, GLenum pname, GLfloat *params)
- void glGetTexGeniv(GLenum coord, GLenum pname, GLint *params)
- void glGetTexImage(GLenum target, GLint level, GLenum format, GLenum type, void *pixels)
- void glGetTexLevelParameterfv(GLenum target, GLint level, GLenum pname, GLfloat *params)
- void glGetTexLevelParameteriv(GLenum target, GLint level, GLenum pname, GLint *params)
- void glGetTexParameterfv(GLenum target, GLenum pname, GLfloat *params)
- void glGetTexParameteriv(GLenum target, GLenum pname, GLint *params)
- void glHint(GLenum target, GLenum mode)
- void glIndexMask(GLuint mask)
- void glIndexPointer(GLenum type, GLsizei stride, const void *pointer)
- void glIndexd(GLdouble c)
- void glIndexdv(const GLdouble *c)
- void glIndexf(GLfloat c)
- void glIndexfv(const GLfloat *c)
- void glIndexi(GLint c)
- void glIndexiv(const GLint *c)
- void glIndexs(GLshort c)
- void glIndexsv(const GLshort *c)
- void glIndexub(GLubyte c)
- void glIndexubv(const GLubyte *c)
- void glInitNames(void)
- void glInterleavedArrays(GLenum format, GLsizei stride, const void *pointer)
- GLboolean glIsEnabled(GLenum cap)
- GLboolean glIsList(GLuint list)
- GLboolean glIsTexture(GLuint texture)
- void glLightModelf(GLenum pname, GLfloat param)
- void glLightModelfv(GLenum pname, const GLfloat *params)
- void glLightModeli(GLenum pname, GLint param)
- void glLightModeliv(GLenum pname, const GLint *params)
- void glLightf(GLenum light, GLenum pname, GLfloat param)
- void glLightfv(GLenum light, GLenum pname, const GLfloat *params)

- void glLighti(GLenum light, GLenum pname, GLint param)
- void glLightiv(GLenum light, GLenum pname, const GLint *params)
- void glLineStipple(GLint factor, GLushort pattern)
- void glLineWidth(GLfloat width)
- void glListBase(GLuint base)
- void glLoadIdentity(void)
- void glLoadMatrixd(const GLdouble *m)
- void glLoadMatrixf(const GLfloat *m)
- void glLoadName(GLuint name)
- void glLogicOp(GLenum opcode)
- void glMap1d(GLenum target, GLdouble u1, GLdouble u2, GLint stride, GLint order, const GLdouble *points)
- void glMap1f(GLenum target, GLfloat u1, GLfloat u2, GLint stride, GLint order, const GLfloat *points)
- void glMap2d(GLenum target, GLdouble u1, GLdouble u2, GLint ustride, GLint uorder, GLdouble v1, GLdouble v2, GLint vstride, GLint vorder, const GLdouble *points)
- void glMap2f(GLenum target, GLfloat u1, GLfloat u2, GLint ustride, GLint uorder, GLfloat v1, GLfloat v2, GLint vstride, GLint vorder, const GLfloat *points)
- void glMapGrid1d(GLint un, GLdouble u1, GLdouble u2)
- void glMapGrid1f(GLint un, GLfloat u1, GLfloat u2)
- void glMapGrid2d(GLint un, GLdouble u1, GLdouble u2, GLint vn, GLdouble v1, GLdouble v2)
- void glMapGrid2f(GLint un, GLfloat u1, GLfloat u2, GLint vn, GLfloat v1, GLfloat v2)
- void glMaterialf(GLenum face, GLenum pname, GLfloat param)
- void glMaterialfv(GLenum face, GLenum pname, const GLfloat *params)
- void glMateriali(GLenum face, GLenum pname, GLint param)
- void glMaterialiv(GLenum face, GLenum pname, const GLint *params)
- void glMatrixMode(GLenum mode)
- void glMultMatrixd(const GLdouble *m)
- void glMultMatrixf(const GLfloat *m)
- void glNewList(GLuint list, GLenum mode)
- void glNormal3b(GLbyte nx, GLbyte ny, GLbyte nz)
- void glNormal3bv(const GLbyte *v)
- void glNormal3d(GLdouble nx, GLdouble ny, GLdouble nz)
- void glNormal3dv(const GLdouble *v)
- void glNormal3f(GLfloat nx, GLfloat ny, GLfloat nz)
- void glNormal3fv(const GLfloat *v)
- void glNormal3i(GLint nx, GLint ny, GLint nz)
- void glNormal3iv(const GLint *v)
- void glNormal3s(GLshort nx, GLshort ny, GLshort nz)

- void glNormal3sv(const GLshort *v)
- void glNormalPointer(GLenum type, GLsizei stride, const void *pointer)
- void glOrtho(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble zNear, GLdouble zFar)
- void glPassThrough(GLfloat token)
- void glPixelMapfv(GLenum map, GLsizei mapsize, const GLfloat *values)
- void glPixelMapuiv(GLenum map, GLsizei mapsize, const GLuint *values)
- void glPixelMapusv(GLenum map, GLsizei mapsize, const GLushort *values)
- void glPixelStoref(GLenum pname, GLfloat param)
- void glPixelStorei(GLenum pname, GLint param)
- void glPixelTransferf(GLenum pname, GLfloat param)
- void glPixelTransferi(GLenum pname, GLint param)
- void glPixelZoom(GLfloat xfactor, GLfloat yfactor)
- void glPointSize(GLfloat size)
- void glPolygonMode(GLenum face, GLenum mode)
- void glPolygonOffset(GLfloat factor, GLfloat units)
- void glPolygonStipple(const GLubyte *mask)
- void glPopAttrib(void)
- void glPopClientAttrib(void)
- void glPopMatrix(void)
- void glPopName(void)
- void glPrioritizeTextures(GLsizei n, const GLuint *textures, const GLclampf *priorities)
- void glPushAttrib(GLbitfield mask)
- void glPushClientAttrib(GLbitfield mask)
- void glPushMatrix(void)
- void glPushName(GLuint name)
- void glRasterPos2d(GLdouble x, GLdouble y)
- void glRasterPos2dv(const GLdouble *v)
- void glRasterPos2f(GLfloat x, GLfloat y)
- void glRasterPos2fv(const GLfloat *v)
- void glRasterPos2i(GLint x, GLint y)
- void glRasterPos2iv(const GLint *v)
- void glRasterPos2s(GLshort x, GLshort y)
- void glRasterPos2sv(const GLshort *v)
- void glRasterPos3d(GLdouble x, GLdouble y, GLdouble z)
- void glRasterPos3dv(const GLdouble *v)

- void glRasterPos3f(GLfloat x, GLfloat y, GLfloat z)
- void glRasterPos3fv(const GLfloat *v)
- void glRasterPos3i(GLint x, GLint y, GLint z)
- void glRasterPos3iv(const GLint *v)
- void glRasterPos3s(GLshort x, GLshort y, GLshort z)
- void glRasterPos3sv(const GLshort *v)
- void glRasterPos4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)
- void glRasterPos4dv(const GLdouble *v)
- void glRasterPos4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- void glRasterPos4fv(const GLfloat *v)
- void glRasterPos4i(GLint x, GLint y, GLint z, GLint w)
- void glRasterPos4iv(const GLint *v)
- void glRasterPos4s(GLshort x, GLshort y, GLshort z, GLshort w)
- void glRasterPos4sv(const GLshort *v)
- void glReadBuffer(GLenum mode)
- void glReadPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum format, GLenum type, void *pixels)
- void glRectd(GLdouble x1, GLdouble y1, GLdouble x2, GLdouble y2)
- void glRectdv(const GLdouble *v1, const GLdouble *v2)
- void glRectf(GLfloat x1, GLfloat y1, GLfloat x2, GLfloat y2)
- void glRectfv(const GLfloat *v1, const GLfloat *v2)
- void glRecti(GLint x1, GLint y1, GLint x2, GLint y2)
- void glRectiv(const GLint *v1, const GLint *v2)
- void glRects(GLshort x1, GLshort y1, GLshort x2, GLshort y2)
- void glRectsv(const GLshort *v1, const GLshort *v2)
- GLint glRenderMode(GLenum mode)
- void glRotated(GLdouble angle, GLdouble x, GLdouble y, GLdouble z)
- void glRotatef(GLfloat angle, GLfloat x, GLfloat y, GLfloat z)
- void glScaled(GLdouble x, GLdouble y, GLdouble z)
- void glScalef(GLfloat x, GLfloat y, GLfloat z)
- void glScissor(GLint x, GLint y, GLsizei width, GLsizei height)
- void glSelectBuffer(GLsizei size, GLuint *buffer)
- void glShadeModel(GLenum mode)
- void glStencilFunc(GLenum func, GLint ref, GLuint mask)
- void glStencilMask(GLuint mask)
- void glStencilOp(GLenum fail, GLenum zfail, GLenum zpass)

- `void glTexCoord1d(GLdouble s)`
- `void glTexCoord1dv(const GLdouble *v)`
- `void glTexCoord1f(GLfloat s)`
- `void glTexCoord1fv(const GLfloat *v)`
- `void glTexCoord1i(GLint s)`
- `void glTexCoord1iv(const GLint *v)`
- `void glTexCoord1s(GLshort s)`
- `void glTexCoord1sv(const GLshort *v)`
- `void glTexCoord2d(GLdouble s, GLdouble t)`
- `void glTexCoord2dv(const GLdouble *v)`
- `void glTexCoord2f(GLfloat s, GLfloat t)`
- `void glTexCoord2fv(const GLfloat *v)`
- `void glTexCoord2i(GLint s, GLint t)`
- `void glTexCoord2iv(const GLint *v)`
- `void glTexCoord2s(GLshort s, GLshort t)`
- `void glTexCoord2sv(const GLshort *v)`
- `void glTexCoord3d(GLdouble s, GLdouble t, GLdouble r)`
- `void glTexCoord3dv(const GLdouble *v)`
- `void glTexCoord3f(GLfloat s, GLfloat t, GLfloat r)`
- `void glTexCoord3fv(const GLfloat *v)`
- `void glTexCoord3i(GLint s, GLint t, GLint r)`
- `void glTexCoord3iv(const GLint *v)`
- `void glTexCoord3s(GLshort s, GLshort t, GLshort r)`
- `void glTexCoord3sv(const GLshort *v)`
- `void glTexCoord4d(GLdouble s, GLdouble t, GLdouble r, GLdouble q)`
- `void glTexCoord4dv(const GLdouble *v)`
- `void glTexCoord4f(GLfloat s, GLfloat t, GLfloat r, GLfloat q)`
- `void glTexCoord4fv(const GLfloat *v)`
- `void glTexCoord4i(GLint s, GLint t, GLint r, GLint q)`
- `void glTexCoord4iv(const GLint *v)`
- `void glTexCoord4s(GLshort s, GLshort t, GLshort r, GLshort q)`
- `void glTexCoord4sv(const GLshort *v)`
- `void glTexCoordPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)`
- `void glTexEnvf(GLenum target, GLenum pname, GLfloat param)`
- `void glTexEnvfv(GLenum target, GLenum pname, const GLfloat *params)`
- `void glTexEnvf(GLenum target, GLenum pname, GLint param)`

- void glTexEnviv(GLenum target, GLenum pname, const GLint *params)
- void glTexGend(GLenum coord, GLenum pname, GLdouble param)
- void glTexGendv(GLenum coord, GLenum pname, const GLdouble *params)
- void glTexGenf(GLenum coord, GLenum pname, GLfloat param)
- void glTexGenfv(GLenum coord, GLenum pname, const GLfloat *params)
- void glTexGeni(GLenum coord, GLenum pname, GLint param)
- void glTexGeniv(GLenum coord, GLenum pname, const GLint *params)
- void glTexImage1D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLint border, GLenum format, GLenum type, const void *pixels)
- void glTexImage2D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, const void *pixels)
- void glTexParameterf(GLenum target, GLenum pname, GLfloat param)
- void glTexParameterfv(GLenum target, GLenum pname, const GLfloat *params)
- void glTexParameteri(GLenum target, GLenum pname, GLint param)
- void glTexParameteriv(GLenum target, GLenum pname, const GLint *params)
- void glTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLenum type, const void *pixels)
- void glTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLenum type, const void *pixels)
- void glTranslated(GLdouble x, GLdouble y, GLdouble z)
- void glTranslatef(GLfloat x, GLfloat y, GLfloat z)
- void glVertex2d(GLdouble x, GLdouble y)
- void glVertex2dv(const GLdouble *v)
- void glVertex2f(GLfloat x, GLfloat y)
- void glVertex2fv(const GLfloat *v)
- void glVertex2i(GLint x, GLint y)
- void glVertex2iv(const GLint *v)
- void glVertex2s(GLshort x, GLshort y)
- void glVertex2sv(const GLshort *v)
- void glVertex3d(GLdouble x, GLdouble y, GLdouble z)
- void glVertex3dv(const GLdouble *v)
- void glVertex3f(GLfloat x, GLfloat y, GLfloat z)
- void glVertex3fv(const GLfloat *v)
- void glVertex3i(GLint x, GLint y, GLint z)
- void glVertex3iv(const GLint *v)
- void glVertex3s(GLshort x, GLshort y, GLshort z)
- void glVertex3sv(const GLshort *v)

- `void glVertex4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)`
- `void glVertex4dv(const GLdouble *v)`
- `void glVertex4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)`
- `void glVertex4fv(const GLfloat *v)`
- `void glVertex4i(GLint x, GLint y, GLint z, GLint w)`
- `void glVertex4iv(const GLint *v)`
- `void glVertex4s(GLshort x, GLshort y, GLshort z, GLshort w)`
- `void glVertex4sv(const GLshort *v)`
- `void glVertexPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)`
- `void glViewport(GLint x, GLint y, GLsizei width, GLsizei height)`

RINGOPENGL (OPENGL 2.1) FUNCTIONS REFERENCE

- GL_ZERO
- GL_FALSE
- GL_LOGIC_OP
- GL_NONE
- GL_TEXTURE_COMPONENTS
- GL_NO_ERROR
- GL_POINTS
- GL_CURRENT_BIT
- GL_TRUE
- GL_ONE
- GL_CLIENT_PIXEL_STORE_BIT
- GL_LINES
- GL_LINE_LOOP
- GL_POINT_BIT
- GL_CLIENT_VERTEX_ARRAY_BIT
- GL_LINE_STRIP
- GL_LINE_BIT
- GL_TRIANGLES
- GL_TRIANGLE_STRIP
- GL_TRIANGLE_FAN
- GL_QUADS
- GL_QUAD_STRIP
- GL_POLYGON_BIT
- GL_POLYGON
- GL_POLYGON_STIPPLE_BIT
- GL_PIXEL_MODE_BIT
- GL_LIGHTING_BIT

- GL_FOG_BIT
- GL_DEPTH_BUFFER_BIT
- GL_ACCUM
- GL_LOAD
- GL_RETURN
- GL_MULT
- GL_ADD
- GL_NEVER
- GL_ACCUM_BUFFER_BIT
- GL_LESS
- GL_EQUAL
- GL_LEQUAL
- GL_GREATER
- GL_NOTEQUAL
- GL_GEQUAL
- GL_ALWAYS
- GL_SRC_COLOR
- GL_ONE_MINUS_SRC_COLOR
- GL_SRC_ALPHA
- GL_ONE_MINUS_SRC_ALPHA
- GL_DST_ALPHA
- GL_ONE_MINUS_DST_ALPHA
- GL_DST_COLOR
- GL_ONE_MINUS_DST_COLOR
- GL_SRC_ALPHA_SATURATE
- GL_STENCIL_BUFFER_BIT
- GL_FRONT_LEFT
- GL_FRONT_RIGHT
- GL_BACK_LEFT
- GL_BACK_RIGHT
- GL_FRONT
- GL_BACK
- GL_LEFT
- GL_RIGHT
- GL_FRONT_AND_BACK
- GL_AUX0

- GL_AUX1
- GL_AUX2
- GL_AUX3
- GL_INVALID_ENUM
- GL_INVALID_VALUE
- GL_INVALID_OPERATION
- GL_STACK_OVERFLOW
- GL_STACK_UNDERFLOW
- GL_OUT_OF_MEMORY
- GL_2D
- GL_3D
- GL_3D_COLOR
- GL_3D_COLOR_TEXTURE
- GL_4D_COLOR_TEXTURE
- GL_PASS_THROUGH_TOKEN
- GL_POINT_TOKEN
- GL_LINE_TOKEN
- GL_POLYGON_TOKEN
- GL_BITMAP_TOKEN
- GL_DRAW_PIXEL_TOKEN
- GL_COPY_PIXEL_TOKEN
- GL_LINE_RESET_TOKEN
- GL_EXP
- GL_VIEWPORT_BIT
- GL_EXP2
- GL_CW
- GL_CCW
- GL_COEFF
- GL_ORDER
- GL_DOMAIN
- GL_CURRENT_COLOR
- GL_CURRENT_INDEX
- GL_CURRENT_NORMAL
- GL_CURRENT_TEXTURE_COORDS
- GL_CURRENT_RASTER_COLOR
- GL_CURRENT_RASTER_INDEX

- GL_CURRENT_RASTER_TEXTURE_COORDS
- GL_CURRENT_RASTER_POSITION
- GL_CURRENT_RASTER_POSITION_VALID
- GL_CURRENT_RASTER_DISTANCE
- GL_POINT_SMOOTH
- GL_POINT_SIZE
- GL_POINT_SIZE_RANGE
- GL_POINT_SIZE_GRANULARITY
- GL_LINE_SMOOTH
- GL_LINE_WIDTH
- GL_LINE_WIDTH_RANGE
- GL_LINE_WIDTH_GRANULARITY
- GL_LINE_STIPPLE
- GL_LINE_STIPPLE_PATTERN
- GL_LINE_STIPPLE_REPEAT
- GL_LIST_MODE
- GL_MAX_LIST_NESTING
- GL_LIST_BASE
- GL_LIST_INDEX
- GL_POLYGON_MODE
- GL_POLYGON_SMOOTH
- GL_POLYGON_STIPPLE
- GL_EDGE_FLAG
- GL_CULL_FACE
- GL_CULL_FACE_MODE
- GL_FRONT_FACE
- GL_LIGHTING
- GL_LIGHT_MODEL_LOCAL_VIEWER
- GL_LIGHT_MODEL_TWO_SIDE
- GL_LIGHT_MODEL_AMBIENT
- GL_SHADE_MODEL
- GL_COLOR_MATERIAL_FACE
- GL_COLOR_MATERIAL_PARAMETER
- GL_COLOR_MATERIAL
- GL_FOG
- GL_FOG_INDEX

- GL_FOG_DENSITY
- GL_FOG_START
- GL_FOG_END
- GL_FOG_MODE
- GL_FOG_COLOR
- GL_DEPTH_RANGE
- GL_DEPTH_TEST
- GL_DEPTH_WRITEMASK
- GL_DEPTH_CLEAR_VALUE
- GL_DEPTH_FUNC
- GL_ACCUM_CLEAR_VALUE
- GL_STENCIL_TEST
- GL_STENCIL_CLEAR_VALUE
- GL_STENCIL_FUNC
- GL_STENCIL_VALUE_MASK
- GL_STENCIL_FAIL
- GL_STENCIL_PASS_DEPTH_FAIL
- GL_STENCIL_PASS_DEPTH_PASS
- GL_STENCIL_REF
- GL_STENCIL_WRITEMASK
- GL_MATRIX_MODE
- GL_NORMALIZE
- GL_VIEWPORT
- GL_MODELVIEW_STACK_DEPTH
- GL_PROJECTION_STACK_DEPTH
- GL_TEXTURE_STACK_DEPTH
- GL_MODELVIEW_MATRIX
- GL_PROJECTION_MATRIX
- GL_TEXTURE_MATRIX
- GL_ATTRIB_STACK_DEPTH
- GL_CLIENT_ATTRIB_STACK_DEPTH
- GL_ALPHA_TEST
- GL_ALPHA_TEST_FUNC
- GL_ALPHA_TEST_REF
- GL_DITHER
- GL_BLEND_DST

- GL_BLEND_SRC
- GL_BLEND
- GL_LOGIC_OP_MODE
- GL_INDEX_LOGIC_OP
- GL_COLOR_LOGIC_OP
- GL_AUX_BUFFERS
- GL_DRAW_BUFFER
- GL_READ_BUFFER
- GL_SCISSOR_BOX
- GL_SCISSOR_TEST
- GL_INDEX_CLEAR_VALUE
- GL_INDEX_WRITEMASK
- GL_COLOR_CLEAR_VALUE
- GL_COLOR_WRITEMASK
- GL_INDEX_MODE
- GL_RGBA_MODE
- GL_DOUBLEBUFFER
- GL_STEREO
- GL_RENDER_MODE
- GL_PERSPECTIVE_CORRECTION_HINT
- GL_POINT_SMOOTH_HINT
- GL_LINE_SMOOTH_HINT
- GL_POLYGON_SMOOTH_HINT
- GL_FOG_HINT
- GL_TEXTURE_GEN_S
- GL_TEXTURE_GEN_T
- GL_TEXTURE_GEN_R
- GL_TEXTURE_GEN_Q
- GL_PIXEL_MAP_I_TO_I
- GL_PIXEL_MAP_S_TO_S
- GL_PIXEL_MAP_I_TO_R
- GL_PIXEL_MAP_I_TO_G
- GL_PIXEL_MAP_I_TO_B
- GL_PIXEL_MAP_I_TO_A
- GL_PIXEL_MAP_R_TO_R
- GL_PIXEL_MAP_G_TO_G

- GL_PIXEL_MAP_B_TO_B
- GL_PIXEL_MAP_A_TO_A
- GL_PIXEL_MAP_I_TO_I_SIZE
- GL_PIXEL_MAP_S_TO_S_SIZE
- GL_PIXEL_MAP_I_TO_R_SIZE
- GL_PIXEL_MAP_I_TO_G_SIZE
- GL_PIXEL_MAP_I_TO_B_SIZE
- GL_PIXEL_MAP_I_TO_A_SIZE
- GL_PIXEL_MAP_R_TO_R_SIZE
- GL_PIXEL_MAP_G_TO_G_SIZE
- GL_PIXEL_MAP_B_TO_B_SIZE
- GL_PIXEL_MAP_A_TO_A_SIZE
- GL_UNPACK_SWAP_BYTES
- GL_UNPACK_LSB_FIRST
- GL_UNPACK_ROW_LENGTH
- GL_UNPACK_SKIP_ROWS
- GL_UNPACK_SKIP_PIXELS
- GL_UNPACK_ALIGNMENT
- GL_PACK_SWAP_BYTES
- GL_PACK_LSB_FIRST
- GL_PACK_ROW_LENGTH
- GL_PACK_SKIP_ROWS
- GL_PACK_SKIP_PIXELS
- GL_PACK_ALIGNMENT
- GL_MAP_COLOR
- GL_MAP_STENCIL
- GL_INDEX_SHIFT
- GL_INDEX_OFFSET
- GL_RED_SCALE
- GL_RED_BIAS
- GL_ZOOM_X
- GL_ZOOM_Y
- GL_GREEN_SCALE
- GL_GREEN_BIAS
- GL_BLUE_SCALE
- GL_BLUE_BIAS

- GL_ALPHA_SCALE
- GL_ALPHA_BIAS
- GL_DEPTH_SCALE
- GL_DEPTH_BIAS
- GL_MAX_EVAL_ORDER
- GL_MAX_LIGHTS
- GL_MAX_CLIP_PLANES
- GL_MAX_TEXTURE_SIZE
- GL_MAX_PIXEL_MAP_TABLE
- GL_MAX_ATTRIB_STACK_DEPTH
- GL_MAX_MODELVIEW_STACK_DEPTH
- GL_MAX_NAME_STACK_DEPTH
- GL_MAX_PROJECTION_STACK_DEPTH
- GL_MAX_TEXTURE_STACK_DEPTH
- GL_MAX_VIEWPORT_DIMS
- GL_MAX_CLIENT_ATTRIB_STACK_DEPTH
- GL_SUBPIXEL_BITS
- GL_INDEX_BITS
- GL_RED_BITS
- GL_GREEN_BITS
- GL_BLUE_BITS
- GL_ALPHA_BITS
- GL_DEPTH_BITS
- GL_STENCIL_BITS
- GL_ACCUM_RED_BITS
- GL_ACCUM_GREEN_BITS
- GL_ACCUM_BLUE_BITS
- GL_ACCUM_ALPHA_BITS
- GL_NAME_STACK_DEPTH
- GL_AUTO_NORMAL
- GL_MAP1_COLOR_4
- GL_MAP1_INDEX
- GL_MAP1_NORMAL
- GL_MAP1_TEXTURE_COORD_1
- GL_MAP1_TEXTURE_COORD_2
- GL_MAP1_TEXTURE_COORD_3

- GL_MAP1_TEXTURE_COORD_4
- GL_MAP1_VERTEX_3
- GL_MAP1_VERTEX_4
- GL_MAP2_COLOR_4
- GL_MAP2_INDEX
- GL_MAP2_NORMAL
- GL_MAP2_TEXTURE_COORD_1
- GL_MAP2_TEXTURE_COORD_2
- GL_MAP2_TEXTURE_COORD_3
- GL_MAP2_TEXTURE_COORD_4
- GL_MAP2_VERTEX_3
- GL_MAP2_VERTEX_4
- GL_MAP1_GRID_DOMAIN
- GL_MAP1_GRID_SEGMENTS
- GL_MAP2_GRID_DOMAIN
- GL_MAP2_GRID_SEGMENTS
- GL_TEXTURE_1D
- GL_TEXTURE_2D
- GL_FEEDBACK_BUFFER_POINTER
- GL_FEEDBACK_BUFFER_SIZE
- GL_FEEDBACK_BUFFER_TYPE
- GL_SELECTION_BUFFER_POINTER
- GL_SELECTION_BUFFER_SIZE
- GL_TEXTURE_WIDTH
- GL_TRANSFORM_BIT
- GL_TEXTURE_HEIGHT
- GL_TEXTURE_INTERNAL_FORMAT
- GL_TEXTURE_BORDER_COLOR
- GL_TEXTURE_BORDER
- GL_DONT_CARE
- GL_FASTEST
- GL_NICEST
- GL_AMBIENT
- GL_DIFFUSE
- GL_SPECULAR
- GL_POSITION

- GL_SPOT_DIRECTION
- GL_SPOT_EXPONENT
- GL_SPOT_CUTOFF
- GL_CONSTANT_ATTENUATION
- GL_LINEAR_ATTENUATION
- GL_QUADRATIC_ATTENUATION
- GL_COMPILE
- GL_COMPILE_AND_EXECUTE
- GL_BYTE
- GL_UNSIGNED_BYTE
- GL_SHORT
- GL_UNSIGNED_SHORT
- GL_INT
- GL_UNSIGNED_INT
- GL_FLOAT
- GL_2_BYTES
- GL_3_BYTES
- GL_4_BYTES
- GL_DOUBLE
- GL_CLEAR
- GL_AND
- GL_AND_REVERSE
- GL_COPY
- GL_AND_INVERTED
- GL_NOOP
- GL_XOR
- GL_OR
- GL_NOR
- GL_EQUIV
- GL_INVERT
- GL_OR_REVERSE
- GL_COPY_INVERTED
- GL_OR_INVERTED
- GL_NAND
- GL_SET
- GL_EMISSION

- GL_SHININESS
- GL_AMBIENT_AND_DIFFUSE
- GL_COLOR_INDEXES
- GL_MODELVIEW
- GL_PROJECTION
- GL_TEXTURE
- GL_COLOR
- GL_DEPTH
- GL_STENCIL
- GL_COLOR_INDEX
- GL_STENCIL_INDEX
- GL_DEPTH_COMPONENT
- GL_RED
- GL_GREEN
- GL_BLUE
- GL_ALPHA
- GL_RGB
- GL_RGBA
- GL_LUMINANCE
- GL_LUMINANCE_ALPHA
- GL_BITMAP
- GL_POINT
- GL_LINE
- GL_FILL
- GL_RENDER
- GL_FEEDBACK
- GL_SELECT
- GL_FLAT
- GL_SMOOTH
- GL_KEEP
- GL_REPLACE
- GL_INCR
- GL_DECR
- GL_VENDOR
- GL_RENDERER
- GL_VERSION

- GL_EXTENSIONS
- GL_S
- GL_ENABLE_BIT
- GL_T
- GL_R
- GL_Q
- GL_MODULATE
- GL_DECAL
- GL_TEXTURE_ENV_MODE
- GL_TEXTURE_ENV_COLOR
- GL_TEXTURE_ENV
- GL_EYE_LINEAR
- GL_OBJECT_LINEAR
- GL_SPHERE_MAP
- GL_TEXTURE_GEN_MODE
- GL_OBJECT_PLANE
- GL_EYE_PLANE
- GL_NEAREST
- GL_LINEAR
- GL_NEAREST_MIPMAP_NEAREST
- GL_LINEAR_MIPMAP_NEAREST
- GL_NEAREST_MIPMAP_LINEAR
- GL_LINEAR_MIPMAP_LINEAR
- GL_TEXTURE_MAG_FILTER
- GL_TEXTURE_MIN_FILTER
- GL_TEXTURE_WRAP_S
- GL_TEXTURE_WRAP_T
- GL_CLAMP
- GL_REPEAT
- GL_POLYGON_OFFSET_UNITS
- GL_POLYGON_OFFSET_POINT
- GL_POLYGON_OFFSET_LINE
- GL_R3_G3_B2
- GL_V2F
- GL_V3F
- GL_C4UB_V2F

- GL_C4UB_V3F
- GL_C3F_V3F
- GL_N3F_V3F
- GL_C4F_N3F_V3F
- GL_T2F_V3F
- GL_T4F_V4F
- GL_T2F_C4UB_V3F
- GL_T2F_C3F_V3F
- GL_T2F_N3F_V3F
- GL_T2F_C4F_N3F_V3F
- GL_T4F_C4F_N3F_V4F
- GL_CLIP_PLANE0
- GL_CLIP_PLANE1
- GL_CLIP_PLANE2
- GL_CLIP_PLANE3
- GL_CLIP_PLANE4
- GL_CLIP_PLANE5
- GL_LIGHT0
- GL_COLOR_BUFFER_BIT
- GL_LIGHT1
- GL_LIGHT2
- GL_LIGHT3
- GL_LIGHT4
- GL_LIGHT5
- GL_LIGHT6
- GL_LIGHT7
- GL_HINT_BIT
- GL_POLYGON_OFFSET_FILL
- GL_POLYGON_OFFSET_FACTOR
- GL_ALPHA4
- GL_ALPHA8
- GL_ALPHA12
- GL_ALPHA16
- GL_LUMINANCE4
- GL_LUMINANCE8
- GL_LUMINANCE12

- GL_LUMINANCE16
- GL_LUMINANCE4_ALPHA4
- GL_LUMINANCE6_ALPHA2
- GL_LUMINANCE8_ALPHA8
- GL_LUMINANCE12_ALPHA4
- GL_LUMINANCE12_ALPHA12
- GL_LUMINANCE16_ALPHA16
- GL_INTENSITY
- GL_INTENSITY4
- GL_INTENSITY8
- GL_INTENSITY12
- GL_INTENSITY16
- GL_RGB4
- GL_RGB5
- GL_RGB8
- GL_RGB10
- GL_RGB12
- GL_RGB16
- GL_RGBA2
- GL_RGBA4
- GL_RGB5_A1
- GL_RGBA8
- GL_RGB10_A2
- GL_RGBA12
- GL_RGBA16
- GL_TEXTURE_RED_SIZE
- GL_TEXTURE_GREEN_SIZE
- GL_TEXTURE_BLUE_SIZE
- GL_TEXTURE_ALPHA_SIZE
- GL_TEXTURE_LUMINANCE_SIZE
- GL_TEXTURE_INTENSITY_SIZE
- GL_PROXY_TEXTURE_1D
- GL_PROXY_TEXTURE_2D
- GL_TEXTURE_PRIORITY
- GL_TEXTURE_RESIDENT
- GL_TEXTURE_BINDING_1D

- GL_TEXTURE_BINDING_2D
- GL_VERTEX_ARRAY
- GL_NORMAL_ARRAY
- GL_COLOR_ARRAY
- GL_INDEX_ARRAY
- GL_TEXTURE_COORD_ARRAY
- GL_EDGE_FLAG_ARRAY
- GL_VERTEX_ARRAY_SIZE
- GL_VERTEX_ARRAY_TYPE
- GL_VERTEX_ARRAY_STRIDE
- GL_NORMAL_ARRAY_TYPE
- GL_NORMAL_ARRAY_STRIDE
- GL_COLOR_ARRAY_SIZE
- GL_COLOR_ARRAY_TYPE
- GL_COLOR_ARRAY_STRIDE
- GL_INDEX_ARRAY_TYPE
- GL_INDEX_ARRAY_STRIDE
- GL_TEXTURE_COORD_ARRAY_SIZE
- GL_TEXTURE_COORD_ARRAY_TYPE
- GL_TEXTURE_COORD_ARRAY_STRIDE
- GL_EDGE_FLAG_ARRAY_STRIDE
- GL_VERTEX_ARRAY_POINTER
- GL_NORMAL_ARRAY_POINTER
- GL_COLOR_ARRAY_POINTER
- GL_INDEX_ARRAY_POINTER
- GL_TEXTURE_COORD_ARRAY_POINTER
- GL_EDGE_FLAG_ARRAY_POINTER
- GL_COLOR_INDEX1_EXT
- GL_COLOR_INDEX2_EXT
- GL_COLOR_INDEX4_EXT
- GL_COLOR_INDEX8_EXT
- GL_COLOR_INDEX12_EXT
- GL_COLOR_INDEX16_EXT
- GL_EVAL_BIT
- GL_LIST_BIT
- GL_TEXTURE_BIT

- GL_SCISSOR_BIT
- GL_ALL_ATTRIB_BITS
- GL_CLIENT_ALL_ATTRIB_BITS
- GL_SMOOTH_POINT_SIZE_RANGE
- GL_SMOOTH_POINT_SIZE_GRANULARITY
- GL_SMOOTH_LINE_WIDTH_RANGE
- GL_SMOOTH_LINE_WIDTH_GRANULARITY
- GL_UNSIGNED_BYTE_3_3_2
- GL_UNSIGNED_SHORT_4_4_4_4
- GL_UNSIGNED_SHORT_5_5_5_1
- GL_UNSIGNED_INT_8_8_8_8
- GL_UNSIGNED_INT_10_10_10_2
- GL_RESCALE_NORMAL
- GL_TEXTURE_BINDING_3D
- GL_PACK_SKIP_IMAGES
- GL_PACK_IMAGE_HEIGHT
- GL_UNPACK_SKIP_IMAGES
- GL_UNPACK_IMAGE_HEIGHT
- GL_TEXTURE_3D
- GL_PROXY_TEXTURE_3D
- GL_TEXTURE_DEPTH
- GL_TEXTURE_WRAP_R
- GL_MAX_3D_TEXTURE_SIZE
- GL_BGR
- GL_BGRA
- GL_MAX_ELEMENTS_VERTICES
- GL_MAX_ELEMENTS_INDICES
- GL_CLAMP_TO_EDGE
- GL_TEXTURE_MIN_LOD
- GL_TEXTURE_MAX_LOD
- GL_TEXTURE_BASE_LEVEL
- GL_TEXTURE_MAX_LEVEL
- GL_LIGHT_MODEL_COLOR_CONTROL
- GL_SINGLE_COLOR
- GL_SEPARATE_SPECULAR_COLOR
- GL_UNSIGNED_BYTE_2_3_3_REV

- GL_UNSIGNED_SHORT_5_6_5
- GL_UNSIGNED_SHORT_5_6_5_REV
- GL_UNSIGNED_SHORT_4_4_4_4_REV
- GL_UNSIGNED_SHORT_1_5_5_5_REV
- GL_UNSIGNED_INT_8_8_8_8_REV
- GL_ALIASED_POINT_SIZE_RANGE
- GL_ALIASED_LINE_WIDTH_RANGE
- GL_MULTISAMPLE
- GL_SAMPLE_ALPHA_TO_COVERAGE
- GL_SAMPLE_ALPHA_TO_ONE
- GL_SAMPLE_COVERAGE
- GL_SAMPLE_BUFFERS
- GL_SAMPLES
- GL_SAMPLE_COVERAGE_VALUE
- GL_SAMPLE_COVERAGE_INVERT
- GL_CLAMP_TO_BORDER
- GL_TEXTURE0
- GL_TEXTURE1
- GL_TEXTURE2
- GL_TEXTURE3
- GL_TEXTURE4
- GL_TEXTURE5
- GL_TEXTURE6
- GL_TEXTURE7
- GL_TEXTURE8
- GL_TEXTURE9
- GL_TEXTURE10
- GL_TEXTURE11
- GL_TEXTURE12
- GL_TEXTURE13
- GL_TEXTURE14
- GL_TEXTURE15
- GL_TEXTURE16
- GL_TEXTURE17
- GL_TEXTURE18
- GL_TEXTURE19

- GL_TEXTURE20
- GL_TEXTURE21
- GL_TEXTURE22
- GL_TEXTURE23
- GL_TEXTURE24
- GL_TEXTURE25
- GL_TEXTURE26
- GL_TEXTURE27
- GL_TEXTURE28
- GL_TEXTURE29
- GL_TEXTURE30
- GL_TEXTURE31
- GL_ACTIVE_TEXTURE
- GL_CLIENT_ACTIVE_TEXTURE
- GL_MAX_TEXTURE_UNITS
- GL_TRANSPOSE_MODELVIEW_MATRIX
- GL_TRANSPOSE_PROJECTION_MATRIX
- GL_TRANSPOSE_TEXTURE_MATRIX
- GL_TRANSPOSE_COLOR_MATRIX
- GL_SUBTRACT
- GL_COMPRESSED_ALPHA
- GL_COMPRESSED_LUMINANCE
- GL_COMPRESSED_LUMINANCE_ALPHA
- GL_COMPRESSED_INTENSITY
- GL_COMPRESSED_RGB
- GL_COMPRESSED_RGBA
- GL_TEXTURE_COMPRESSION_HINT
- GL_NORMAL_MAP
- GL_REFLECTION_MAP
- GL_TEXTURE_CUBE_MAP
- GL_TEXTURE_BINDING_CUBE_MAP
- GL_TEXTURE_CUBE_MAP_POSITIVE_X
- GL_TEXTURE_CUBE_MAP_NEGATIVE_X
- GL_TEXTURE_CUBE_MAP_POSITIVE_Y
- GL_TEXTURE_CUBE_MAP_NEGATIVE_Y
- GL_TEXTURE_CUBE_MAP_POSITIVE_Z

- GL_TEXTURE_CUBE_MAP_NEGATIVE_Z
- GL_PROXY_TEXTURE_CUBE_MAP
- GL_MAX_CUBE_MAP_TEXTURE_SIZE
- GL_COMBINE
- GL_COMBINE_RGB
- GL_COMBINE_ALPHA
- GL_RGB_SCALE
- GL_ADD_SIGNED
- GL_INTERPOLATE
- GL_CONSTANT
- GL_PRIMARY_COLOR
- GL_PREVIOUS
- GL_SOURCE0_RGB
- GL_SOURCE1_RGB
- GL_SOURCE2_RGB
- GL_SOURCE0_ALPHA
- GL_SOURCE1_ALPHA
- GL_SOURCE2_ALPHA
- GL_OPERAND0_RGB
- GL_OPERAND1_RGB
- GL_OPERAND2_RGB
- GL_OPERAND0_ALPHA
- GL_OPERAND1_ALPHA
- GL_OPERAND2_ALPHA
- GL_TEXTURE_COMPRESSED_IMAGE_SIZE
- GL_TEXTURE_COMPRESSED
- GL_NUM_COMPRESSED_TEXTURE_FORMATS
- GL_COMPRESSED_TEXTURE_FORMATS
- GL_DOT3_RGB
- GL_DOT3_RGBA
- GL_MULTISAMPLE_BIT
- GL_BLEND_DST_RGB
- GL_BLEND_SRC_RGB
- GL_BLEND_DST_ALPHA
- GL_BLEND_SRC_ALPHA
- GL_POINT_SIZE_MIN

- GL_POINT_SIZE_MAX
- GL_POINT_FADE_THRESHOLD_SIZE
- GL_POINT_DISTANCE_ATTENUATION
- GL_GENERATE_MIPMAP
- GL_GENERATE_MIPMAP_HINT
- GL_DEPTH_COMPONENT16
- GL_DEPTH_COMPONENT24
- GL_DEPTH_COMPONENT32
- GL_MIRRORED_REPEAT
- GL_FOG_COORDINATE_SOURCE
- GL_FOG_COORDINATE
- GL_FRAGMENT_DEPTH
- GL_CURRENT_FOG_COORDINATE
- GL_FOG_COORDINATE_ARRAY_TYPE
- GL_FOG_COORDINATE_ARRAY_STRIDE
- GL_FOG_COORDINATE_ARRAY_POINTER
- GL_FOG_COORDINATE_ARRAY
- GL_COLOR_SUM
- GL_CURRENT_SECONDARY_COLOR
- GL_SECONDARY_COLOR_ARRAY_SIZE
- GL_SECONDARY_COLOR_ARRAY_TYPE
- GL_SECONDARY_COLOR_ARRAY_STRIDE
- GL_SECONDARY_COLOR_ARRAY_POINTER
- GL_SECONDARY_COLOR_ARRAY
- GL_MAX_TEXTURE_LOD_BIAS
- GL_TEXTURE_FILTER_CONTROL
- GL_TEXTURE_LOD_BIAS
- GL_INCR_WRAP
- GL_DECR_WRAP
- GL_TEXTURE_DEPTH_SIZE
- GL_DEPTH_TEXTURE_MODE
- GL_TEXTURE_COMPARE_MODE
- GL_TEXTURE_COMPARE_FUNC
- GL_COMPARE_R_TO_TEXTURE
- GL_CURRENT_FOG_COORD
- GL_FOG_COORD

- GL_FOG_COORD_ARRAY
- GL_FOG_COORD_ARRAY_BUFFER_BINDING
- GL_FOG_COORD_ARRAY_POINTER
- GL_FOG_COORD_ARRAY_STRIDE
- GL_FOG_COORD_ARRAY_TYPE
- GL_FOG_COORD_SRC
- GL_SRC0_ALPHA
- GL_SRC0_RGB
- GL_SRC1_ALPHA
- GL_SRC1_RGB
- GL_SRC2_ALPHA
- GL_SRC2_RGB
- GL_BUFFER_SIZE
- GL_BUFFER_USAGE
- GL_QUERY_COUNTER_BITS
- GL_CURRENT_QUERY
- GL_QUERY_RESULT
- GL_QUERY_RESULT_AVAILABLE
- GL_ARRAY_BUFFER
- GL_ELEMENT_ARRAY_BUFFER
- GL_ARRAY_BUFFER_BINDING
- GL_ELEMENT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ARRAY_BUFFER_BINDING
- GL_NORMAL_ARRAY_BUFFER_BINDING
- GL_COLOR_ARRAY_BUFFER_BINDING
- GL_INDEX_ARRAY_BUFFER_BINDING
- GL_TEXTURE_COORD_ARRAY_BUFFER_BINDING
- GL_EDGE_FLAG_ARRAY_BUFFER_BINDING
- GL_SECONDARY_COLOR_ARRAY_BUFFER_BINDING
- GL_FOG_COORDINATE_ARRAY_BUFFER_BINDING
- GL_WEIGHT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ATTRIB_ARRAY_BUFFER_BINDING
- GL_READ_ONLY
- GL_WRITE_ONLY
- GL_READ_WRITE
- GL_BUFFER_ACCESS

- GL_BUFFER_MAPPED
- GL_BUFFER_MAP_POINTER
- GL_STREAM_DRAW
- GL_STREAM_READ
- GL_STREAM_COPY
- GL_STATIC_DRAW
- GL_STATIC_READ
- GL_STATIC_COPY
- GL_DYNAMIC_DRAW
- GL_DYNAMIC_READ
- GL_DYNAMIC_COPY
- GL_SAMPLES_PASSED
- GL_BLEND_EQUATION_RGB
- GL_VERTEX_ATTRIB_ARRAY_ENABLED
- GL_VERTEX_ATTRIB_ARRAY_SIZE
- GL_VERTEX_ATTRIB_ARRAY_STRIDE
- GL_VERTEX_ATTRIB_ARRAY_TYPE
- GL_CURRENT_VERTEX_ATTRIB
- GL_VERTEX_PROGRAM_POINT_SIZE
- GL_VERTEX_PROGRAM_TWO_SIDE
- GL_VERTEX_ATTRIB_ARRAY_POINTER
- GL_STENCIL_BACK_FUNC
- GL_STENCIL_BACK_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_PASS
- GL_MAX_DRAW_BUFFERS
- GL_DRAW_BUFFER0
- GL_DRAW_BUFFER1
- GL_DRAW_BUFFER2
- GL_DRAW_BUFFER3
- GL_DRAW_BUFFER4
- GL_DRAW_BUFFER5
- GL_DRAW_BUFFER6
- GL_DRAW_BUFFER7
- GL_DRAW_BUFFER8
- GL_DRAW_BUFFER9

- GL_DRAW_BUFFER10
- GL_DRAW_BUFFER11
- GL_DRAW_BUFFER12
- GL_DRAW_BUFFER13
- GL_DRAW_BUFFER14
- GL_DRAW_BUFFER15
- GL_BLEND_EQUATION_ALPHA
- GL_POINT_SPRITE
- GL_COORD_REPLACE
- GL_MAX_VERTEX_ATTRIBS
- GL_VERTEX_ATTRIB_ARRAY_NORMALIZED
- GL_MAX_TEXTURE_COORDS
- GL_MAX_TEXTURE_IMAGE_UNITS
- GL_FRAGMENT_SHADER
- GL_VERTEX_SHADER
- GL_MAX_FRAGMENT_UNIFORM_COMPONENTS
- GL_MAX_VERTEX_UNIFORM_COMPONENTS
- GL_MAX_VARYING_FLOATS
- GL_MAX_VERTEX_TEXTURE_IMAGE_UNITS
- GL_MAX_COMBINED_TEXTURE_IMAGE_UNITS
- GL_SHADER_TYPE
- GL_FLOAT_VEC2
- GL_FLOAT_VEC3
- GL_FLOAT_VEC4
- GL_INT_VEC2
- GL_INT_VEC3
- GL_INT_VEC4
- GL_BOOL
- GL_BOOL_VEC2
- GL_BOOL_VEC3
- GL_BOOL_VEC4
- GL_FLOAT_MAT2
- GL_FLOAT_MAT3
- GL_FLOAT_MAT4
- GL_SAMPLER_1D
- GL_SAMPLER_2D

- GL_SAMPLER_3D
- GL_SAMPLER_CUBE
- GL_SAMPLER_1D_SHADOW
- GL_SAMPLER_2D_SHADOW
- GL_DELETE_STATUS
- GL_COMPILE_STATUS
- GL_LINK_STATUS
- GL_VALIDATE_STATUS
- GL_INFO_LOG_LENGTH
- GL_ATTACHED_SHADERS
- GL_ACTIVE_UNIFORMS
- GL_ACTIVE_UNIFORM_MAX_LENGTH
- GL_SHADER_SOURCE_LENGTH
- GL_ACTIVE_ATTRIBUTES
- GL_ACTIVE_ATTRIBUTE_MAX_LENGTH
- GL_FRAGMENT_SHADER_DERIVATIVE_HINT
- GL_SHADING_LANGUAGE_VERSION
- GL_CURRENT_PROGRAM
- GL_POINT_SPRITE_COORD_ORIGIN
- GL_LOWER_LEFT
- GL_UPPER_LEFT
- GL_STENCIL_BACK_REF
- GL_STENCIL_BACK_VALUE_MASK
- GL_STENCIL_BACK_WRITEMASK
- GL_CURRENT_RASTER_SECONDARY_COLOR
- GL_PIXEL_PACK_BUFFER
- GL_PIXEL_UNPACK_BUFFER
- GL_PIXEL_PACK_BUFFER_BINDING
- GL_PIXEL_UNPACK_BUFFER_BINDING
- GL_FLOAT_MAT2x3
- GL_FLOAT_MAT2x4
- GL_FLOAT_MAT3x2
- GL_FLOAT_MAT3x4
- GL_FLOAT_MAT4x2
- GL_FLOAT_MAT4x3
- GL_SRGB

- GL_SRGB8
- GL_SRGB_ALPHA
- GL_SRGB8_ALPHA8
- GL_SLUMINANCE_ALPHA
- GL_SLUMINANCE8_ALPHA8
- GL_SLUMINANCE
- GL_SLUMINANCE8
- GL_COMPRESSED_SRGB
- GL_COMPRESSED_SRGB_ALPHA
- GL_COMPRESSED_SLUMINANCE
- GL_COMPRESSED_SLUMINANCE_ALPHA
- void glAccum(GLenum op, GLfloat value)
- void glActiveTexture(GLenum texture)
- void glAlphaFunc(GLenum func, GLclampf ref)
- GLboolean glAreTexturesResident(GLsizei n, const GLuint * textures, GLboolean * residences)
- void glArrayElement(GLint i)
- void glAttachShader(GLuint program, GLuint shader)
- void glBegin(GLenum mode)
- void glBeginQuery(GLenum target, GLuint id)
- void glBindAttribLocation(GLuint program, GLuint index, const GLchar *name)
- void glBindBuffer(GLenum target, GLuint buffer)
- void glBindTexture(GLenum target, GLuint texture)
- void glBitmap(GLsizei width, GLsizei height, GLfloat xorig, GLfloat yorig, GLfloat xmove, GLfloat ymove, const GLubyte * bitmap)
- void glBlendColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)
- void glBlendEquation(GLenum mode)
- void glBlendEquationSeparate(GLenum modeRGB, GLenum modeAlpha)
- void glBlendFunc(GLenum sfactor, GLenum dfactor)
- void glBlendFuncSeparate(GLenum srcRGB, GLenum dstRGB, GLenum srcAlpha, GLenum dstAlpha)
- void glBufferData(GLenum target, GLsizeiptr size, const GLvoid * data, GLenum usage)
- void glBufferSubData(GLenum target, GLintptr offset, GLsizeiptr size, const GLvoid * data)
- void glCallList(GLuint list)
- void glCallLists(GLsizei n, GLenum type, const GLvoid * lists)
- void glClear(GLbitfield mask)
- void glClearAccum(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glClearColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)

- `void glClearDepth(GLclampd depth)`
- `void glClearIndex(GLfloat c)`
- `void glClearStencil(GLint s)`
- `void glClientActiveTexture(GLenum texture)`
- `void glClipPlane(GLenum plane,const GLdouble * equation)`
- `void glColor3b(GLbyte red,GLbyte green,GLbyte blue)`
- `void glColor3s(GLshort red,GLshort green,GLshort blue)`
- `void glColor3i(GLint red,GLint green,GLint blue)`
- `void glColor3f(GLfloat red,GLfloat green,GLfloat blue)`
- `void glColor3d(GLdouble red,GLdouble green,GLdouble blue)`
- `void glColor3ub(GLubyte red,GLubyte green,GLubyte blue)`
- `void glColor3us(GLushort red,GLushort green,GLushort blue)`
- `void glColor3ui(GLuint red,GLuint green,GLuint blue)`
- `void glColor4b(GLbyte red,GLbyte green,GLbyte blue,GLbyte alpha)`
- `void glColor4s(GLshort red,GLshort green,GLshort blue,GLshort alpha)`
- `void glColor4i(GLint red,GLint green,GLint blue,GLint alpha)`
- `void glColor4f(GLfloat red,GLfloat green,GLfloat blue,GLfloat alpha)`
- `void glColor4d(GLdouble red,GLdouble green,GLdouble blue,GLdouble alpha)`
- `void glColor4ub(GLubyte red,GLubyte green,GLubyte blue,GLubyte alpha)`
- `void glColor4us(GLushort red,GLushort green,GLushort blue,GLushort alpha)`
- `void glColor4ui(GLuint red,GLuint green,GLuint blue,GLuint alpha)`
- `void glColor3bv(const GLbyte * v)`
- `void glColor3sv(const GLshort * v)`
- `void glColor3iv(const GLint * v)`
- `void glColor3fv(const GLfloat * v)`
- `void glColor3dv(const GLdouble * v)`
- `void glColor3ubv(const GLubyte * v)`
- `void glColor3usv(const GLushort * v)`
- `void glColor3uiv(const GLuint * v)`
- `void glColor4bv(const GLbyte * v)`
- `void glColor4sv(const GLshort * v)`
- `void glColor4iv(const GLint * v)`
- `void glColor4fv(const GLfloat * v)`
- `void glColor4dv(const GLdouble * v)`
- `void glColor4ubv(const GLubyte * v)`
- `void glColor4usv(const GLushort * v)`

- `void glColor4uiv(const GLuint * v)`
- `void glColorMask(GLboolean red, GLboolean green, GLboolean blue, GLboolean alpha)`
- `void glColorMaterial(GLenum face, GLenum mode)`
- `void glColorPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)`
- `void glColorSubTable(GLenum target, GLsizei start, GLsizei count, GLenum format, GLenum type, const GLvoid * data)`
- `void glColorTable(GLenum target, GLenum internalformat, GLsizei width, GLenum format, GLenum type, const GLvoid * data)`
- `void glColorTableParameterfv(GLenum target, GLenum pname, const GLfloat * params)`
- `void glColorTableParameteriv(GLenum target, GLenum pname, const GLint * params)`
- `void glCompileShader(GLuint shader)`
- `void glCompressedTexImage1D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLint border, GLsizei imageSize, const GLvoid * data)`
- `void glCompressedTexImage2D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLint border, GLsizei imageSize, const GLvoid * data)`
- `void glCompressedTexImage3D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLsizei depth, GLint border, GLsizei imageSize, const GLvoid * data)`
- `void glCompressedTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLsizei imageSize, const GLvoid * data)`
- `void glCompressedTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLsizei imageSize, const GLvoid * data)`
- `void glCompressedTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLsizei imageSize, const GLvoid * data)`
- `void glConvolutionFilter1D(GLenum target, GLenum internalformat, GLsizei width, GLenum format, GLenum type, const GLvoid * data)`
- `void glConvolutionFilter2D(GLenum target, GLenum internalformat, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)`
- `void glConvolutionParameterf(GLenum target, GLenum pname, GLfloat params)`
- `void glConvolutionParameteri(GLenum target, GLenum pname, GLint params)`
- `void glConvolutionParameterfv(GLenum target, GLenum pname, const GLfloat * params)`
- `void glConvolutionParameteriv(GLenum target, GLenum pname, const GLint * params)`
- `void glCopyColorSubTable(GLenum target, GLsizei start, GLint x, GLint y, GLsizei width)`
- `void glCopyColorTable(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width)`
- `void glCopyConvolutionFilter1D(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width)`
- `void glCopyConvolutionFilter2D(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width, GLsizei height)`
- `void glCopyPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum type)`
- `void glCopyTexImage1D(GLenum target, GLint level, GLenum internalformat, GLint x, GLint y, GLsizei width, GLint border)`

- void glCopyTexImage2D(GLenum target,GLint level,GLenum internalformat,GLint x,GLint y,GLsizei width,GLsizei height,GLint border)
- void glCopyTexSubImage1D(GLenum target,GLint level,GLint xoffset,GLint x,GLint y,GLsizei width)
- void glCopyTexSubImage2D(GLenum target,GLint level,GLint xoffset,GLint yoffset,GLint x,GLint y,GLsizei width,GLsizei height)
- void glCopyTexSubImage3D(GLenum target,GLint level,GLint xoffset,GLint yoffset,GLint zoffset,GLint x,GLint y,GLsizei width,GLsizei height)
- GLuint glCreateProgram(void)
- GLuint glCreateShader(GLenum shaderType)
- void glCullFace(GLenum mode)
- void glDeleteBuffers(GLsizei n,const GLuint * buffers)
- void glDeleteLists(GLuint list,GLsizei range)
- void glDeleteProgram(GLuint program)
- void glDeleteQueries(GLsizei n,const GLuint * ids)
- void glDeleteShader(GLuint shader)
- void glDeleteTextures(GLsizei n,const GLuint * textures)
- void glDepthFunc(GLenum func)
- void glDepthMask(GLboolean flag)
- void glDepthRange(GLclampd nearVal,GLclampd farVal)
- void glDetachShader(GLuint program,GLuint shader)
- void glEnable(GLenum cap)
- void glEnableClientState(GLenum cap)
- void glEnableVertexAttribArray(GLuint index)
- void glDisableVertexAttribArray(GLuint index)
- void glDrawArrays(GLenum mode,GLint first,GLsizei count)
- void glDrawBuffer(GLenum mode)
- void glDrawBuffers(GLsizei n,const GLenum *bufs)
- void glDrawElements(GLenum mode,GLsizei count,GLenum type,const GLvoid * indices)
- void glDrawPixels(GLsizei width,GLsizei height,GLenum format,GLenum type,const GLvoid * data)
- void glDrawRangeElements(GLenum mode,GLuint start,GLuint end,GLsizei count,GLenum type,const GLvoid * indices)
- void glEdgeFlag(GLboolean flag)
- void glEdgeFlagPointer(GLsizei stride,const GLvoid * pointer)
- void glEnd(void)
- void glEndList(void)
- void glEndQuery(GLenum target)
- void glEvalCoord1f(GLfloat u)

- void glEvalCoord1d(GLdouble u)
- void glEvalCoord2f(GLfloat u,GLfloat v)
- void glEvalCoord2d(GLdouble u,GLdouble v)
- void glEvalMesh1(GLenum mode,GLint i1,GLint i2)
- void glEvalPoint1(GLint i)
- void glEvalPoint2(GLint i,GLint j)
- void glFeedbackBuffer(GLsizei size,GLenum type,GLfloat * buffer)
- void glFinish(void)
- void glFlush(void)
- void glFogf(GLenum pname,GLfloat param)
- void glFogi(GLenum pname,GLint param)
- void glFogfv(GLenum pname,const GLfloat * params)
- void glFogiv(GLenum pname,const GLint * params)
- void glFogCoordd(GLdouble coord)
- void glFogCoordf(GLfloat coord)
- void glFogCoorddv(GLdouble * coord)
- void glFogCoordfv(GLfloat * coord)
- void glFogCoordPointer(GLenum type,GLsizei stride,GLvoid * pointer)
- void glFrontFace(GLenum mode)
- void glFrustum(GLdouble left,GLdouble right,GLdouble bottom,GLdouble top,GLdouble nearVal,GLdouble farVal)
- void glGenBuffers(GLsizei n,GLuint * buffers)
- GLuint glGenLists(GLsizei range)
- void glGenQueries(GLsizei n,GLuint * ids)
- void glGenTextures(GLsizei n,GLuint * textures)
- void glGetBooleanv(GLenum pname,GLboolean * params)
- void glGetDoublev(GLenum pname,GLdouble * params)
- void glGetFloatv(GLenum pname,GLfloat * params)
- void glGetIntegerv(GLenum pname,GLint * params)
- void glGetActiveAttrib(GLuint program,GLuint index,GLsizei bufSize,GLsizei *length,GLint *size,GLenum *type,GLchar *name)
- void glGetActiveUniform(GLuint program,GLuint index,GLsizei bufSize,GLsizei *length,GLint *size,GLenum *type,GLchar *name)
- void glGetAttachedShaders(GLuint program,GLsizei maxCount,GLsizei *count,GLuint *shaders)
- GLint glGetAttribLocation(GLuint program,const GLchar *name)
- void glGetBufferParameteriv(GLenum target,GLenum value,GLint * data)
- void glGetBufferPointerv(GLenum target,GLenum pname,GLvoid ** params)

- void glGetBufferSubData(GLenum target, GLintptr offset, GLsizeiptr size, GLvoid * data)
- void glGetClipPlane(GLenum plane, GLdouble * equation)
- void glGetColorTable(GLenum target, GLenum format, GLenum type, GLvoid * table)
- void glGetColorTableParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetColorTableParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetCompressedTexImage(GLenum target, GLint lod, GLvoid * img)
- void glGetConvolutionFilter(GLenum target, GLenum format, GLenum type, GLvoid * image)
- void glGetConvolutionParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetConvolutionParameteriv(GLenum target, GLenum pname, GLint * params)
- GLenum glGetError(void)
- void glGetHistogram(GLenum target, GLboolean reset, GLenum format, GLenum type, GLvoid * values)
- void glGetHistogramParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetHistogramParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetLightfv(GLenum light, GLenum pname, GLfloat * params)
- void glGetLightiv(GLenum light, GLenum pname, GLint * params)
- void glGetMapdv(GLenum target, GLenum query, GLdouble * v)
- void glGetMapfv(GLenum target, GLenum query, GLfloat * v)
- void glGetMapiv(GLenum target, GLenum query, GLint * v)
- void glGetMaterialfv(GLenum face, GLenum pname, GLfloat * params)
- void glGetMaterialiv(GLenum face, GLenum pname, GLint * params)
- void glGetMinmax(GLenum target, GLboolean reset, GLenum format, GLenum types, GLvoid * values)
- void glGetMinmaxParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetMinmaxParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetPixelMapfv(GLenum map, GLfloat * data)
- void glGetPixelMapuiv(GLenum map, GLuint * data)
- void glGetPixelMapusv(GLenum map, GLushort * data)
- void glGetPointerv(GLenum pname, GLvoid ** params)
- void glGetPolygonStipple(GLubyte * pattern)
- void glGetProgramiv(GLuint program, GLenum pname, GLint *params)
- void glGetProgramInfoLog(GLuint program, GLsizei maxLength, GLsizei *length, GLchar *infoLog)
- void glGetQueryObjectiv(GLuint id, GLenum pname, GLint * params)
- void glGetQueryObjectuiv(GLuint id, GLenum pname, GLuint * params)
- void glGetQueryiv(GLenum target, GLenum pname, GLint * params)
- void glGetSeparableFilter(GLenum target, GLenum format, GLenum type, GLvoid * row, GLvoid * column, GLvoid * span)
- void glGetShaderiv(GLuint shader, GLenum pname, GLint *params)

- void glGetShaderInfoLog(GLuint shader, GLsizei maxLength, GLsizei *length, GLchar *infoLog)
- void glGetShaderSource(GLuint shader, GLsizei bufSize, GLsizei *length, GLchar *source)
- const GLubyte* glGetString(GLenum name)
- void glGetTexEnvfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetTexEnviv(GLenum target, GLenum pname, GLint * params)
- void glGetTexGendv(GLenum coord, GLenum pname, GLdouble * params)
- void glGetTexGenfv(GLenum coord, GLenum pname, GLfloat * params)
- void glGetTexGeniv(GLenum coord, GLenum pname, GLint * params)
- void glGetTexImage(GLenum target, GLint level, GLenum format, GLenum type, GLvoid * img)
- void glGetTexLevelParameterfv(GLenum target, GLint level, GLenum pname, GLfloat * params)
- void glGetTexLevelParameteriv(GLenum target, GLint level, GLenum pname, GLint * params)
- void glGetTexParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetTexParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetUniformfv(GLuint program, GLint location, GLfloat *params)
- void glGetUniformiv(GLuint program, GLint location, GLint *params)
- GLint glGetUniformLocation(GLuint program, const GLchar *name)
- void glGetVertexAttribdv(GLuint index, GLenum pname, GLdouble *params)
- void glGetVertexAttribfv(GLuint index, GLenum pname, GLfloat *params)
- void glGetVertexAttribiv(GLuint index, GLenum pname, GLint *params)
- void glGetVertexAttribPointerv(GLuint index, GLenum pname, GLvoid **pointer)
- void glHint(GLenum target, GLenum mode)
- void glHistogram(GLenum target, GLsizei width, GLenum internalformat, GLboolean sink)
- void glIndexs(GLshort c)
- void glIndexi(GLint c)
- void glIndexf(GLfloat c)
- void glIndexd(GLdouble c)
- void glIndexub(GLubyte c)
- void glIndexsv(const GLshort * c)
- void glIndexiv(const GLint * c)
- void glIndexfv(const GLfloat * c)
- void glIndexdv(const GLdouble * c)
- void glIndexubv(const GLubyte * c)
- void glIndexMask(GLuint mask)
- void glIndexPointer(GLenum type, GLsizei stride, const GLvoid * pointer)
- void glInitNames(void)
- void glInterleavedArrays(GLenum format, GLsizei stride, const GLvoid * pointer)

- GLboolean glIsBuffer(GLuint buffer)
- GLboolean glIsEnabled(GLenum cap)
- GLboolean glIsList(GLuint list)
- GLboolean glIsProgram(GLuint program)
- GLboolean glIsQuery(GLuint id)
- GLboolean glIsShader(GLuint shader)
- GLboolean glIsTexture(GLuint texture)
- void glLightf(GLenum light, GLenum pname, GLfloat param)
- void glLighti(GLenum light, GLenum pname, GLint param)
- void glLightfv(GLenum light, GLenum pname, const GLfloat * params)
- void glLightiv(GLenum light, GLenum pname, const GLint * params)
- void glLightModelf(GLenum pname, GLfloat param)
- void glLightModeli(GLenum pname, GLint param)
- void glLightModelfv(GLenum pname, const GLfloat * params)
- void glLightModeliv(GLenum pname, const GLint * params)
- void glLineStipple(GLint factor, GLushort pattern)
- void glLineWidth(GLfloat width)
- void glLinkProgram(GLuint program)
- void glListBase(GLuint base)
- void glLoadIdentity(void)
- void glLoadMatrixd(const GLdouble * m)
- void glLoadMatrixf(const GLfloat * m)
- void glLoadName(GLuint name)
- void glLoadTransposeMatrixd(const GLdouble * m)
- void glLoadTransposeMatrixf(const GLfloat * m)
- void glLogicOp(GLenum opcode)
- void glMap1f(GLenum target, GLfloat u1, GLfloat u2, GLint stride, GLint order, const GLfloat * points)
- void glMap1d(GLenum target, GLdouble u1, GLdouble u2, GLint stride, GLint order, const GLdouble * points)
- void glMap2f(GLenum target, GLfloat u1, GLfloat u2, GLint ustride, GLint uorder, GLfloat v1, GLfloat v2, GLint vstride, GLint vorder, const GLfloat * points)
- void glMap2d(GLenum target, GLdouble u1, GLdouble u2, GLint ustride, GLint uorder, GLdouble v1, GLdouble v2, GLint vstride, GLint vorder, const GLdouble * points)
- void * glMapBuffer(GLenum target, GLenum access)
- void glMapGrid1d(GLint un, GLdouble u1, GLdouble u2)
- void glMapGrid1f(GLint un, GLfloat u1, GLfloat u2)
- void glMapGrid2d(GLint un, GLdouble u1, GLdouble u2, GLint vn, GLdouble v1, GLdouble v2)
- void glMapGrid2f(GLint un, GLfloat u1, GLfloat u2, GLint vn, GLfloat v1, GLfloat v2)

- `void glMaterialf(GLenum face, GLenum pname, GLfloat param)`
- `void glMateriali(GLenum face, GLenum pname, GLint param)`
- `void glMatrixMode(GLenum mode)`
- `void glMinmax(GLenum target, GLenum internalformat, GLboolean sink)`
- `void glMultMatrixd(const GLdouble * m)`
- `void glMultMatrixf(const GLfloat * m)`
- `void glMultTransposeMatrixd(const GLdouble * m)`
- `void glMultTransposeMatrixf(const GLfloat * m)`
- `void glMultiDrawArrays(GLenum mode, GLint * first, GLsizei * count, GLsizei primcount)`
- `void glMultiDrawElements(GLenum mode, const GLsizei * count, GLenum type, const GLvoid ** indices, GLsizei primcount)`
- `void glMultiTexCoord1s(GLenum target, GLshort s)`
- `void glMultiTexCoord1i(GLenum target, GLint s)`
- `void glMultiTexCoord1f(GLenum target, GLfloat s)`
- `void glMultiTexCoord1d(GLenum target, GLdouble s)`
- `void glMultiTexCoord2s(GLenum target, GLshort s, GLshort t)`
- `void glMultiTexCoord2i(GLenum target, GLint s, GLint t)`
- `void glMultiTexCoord2f(GLenum target, GLfloat s, GLfloat t)`
- `void glMultiTexCoord2d(GLenum target, GLdouble s, GLdouble t)`
- `void glMultiTexCoord3s(GLenum target, GLshort s, GLshort t, GLshort r)`
- `void glMultiTexCoord3i(GLenum target, GLint s, GLint t, GLint r)`
- `void glMultiTexCoord3f(GLenum target, GLfloat s, GLfloat t, GLfloat r)`
- `void glMultiTexCoord3d(GLenum target, GLdouble s, GLdouble t, GLdouble r)`
- `void glMultiTexCoord4s(GLenum target, GLshort s, GLshort t, GLshort r, GLshort q)`
- `void glMultiTexCoord4i(GLenum target, GLint s, GLint t, GLint r, GLint q)`
- `void glMultiTexCoord4f(GLenum target, GLfloat s, GLfloat t, GLfloat r, GLfloat q)`
- `void glMultiTexCoord4d(GLenum target, GLdouble s, GLdouble t, GLdouble r, GLdouble q)`
- `void glMultiTexCoord1sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord1iv(GLenum target, const GLint * v)`
- `void glMultiTexCoord1fv(GLenum target, const GLfloat * v)`
- `void glMultiTexCoord1dv(GLenum target, const GLdouble * v)`
- `void glMultiTexCoord2sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord2iv(GLenum target, const GLint * v)`
- `void glMultiTexCoord2fv(GLenum target, const GLfloat * v)`
- `void glMultiTexCoord2dv(GLenum target, const GLdouble * v)`
- `void glMultiTexCoord3sv(GLenum target, const GLshort * v)`

- void glMultiTexCoord3iv(GLenum target,const GLint * v)
- void glMultiTexCoord3fv(GLenum target,const GLfloat * v)
- void glMultiTexCoord3dv(GLenum target,const GLdouble * v)
- void glMultiTexCoord4sv(GLenum target,const GLshort * v)
- void glMultiTexCoord4iv(GLenum target,const GLint * v)
- void glMultiTexCoord4fv(GLenum target,const GLfloat * v)
- void glMultiTexCoord4dv(GLenum target,const GLdouble * v)
- void glNewList(GLuint list,GLenum mode)
- void glNormal3b(GLbyte nx,GLbyte ny,GLbyte nz)
- void glNormal3d(GLdouble nx,GLdouble ny,GLdouble nz)
- void glNormal3f(GLfloat nx,GLfloat ny,GLfloat nz)
- void glNormal3i(GLint nx,GLint ny,GLint nz)
- void glNormal3s(GLshort nx,GLshort ny,GLshort nz)
- void glNormal3bv(const GLbyte * v)
- void glNormal3dv(const GLdouble * v)
- void glNormal3fv(const GLfloat * v)
- void glNormal3iv(const GLint * v)
- void glNormal3sv(const GLshort * v)
- void glNormalPointer(GLenum type,GLsizei stride,const GLvoid * pointer)
- void glOrtho(GLdouble left,GLdouble right,GLdouble bottom,GLdouble top,GLdouble nearVal,GLdouble farVal)
- void glPassThrough(GLfloat token)
- void glPixelMapfv(GLenum map,GLsizei mapsize,const GLfloat * values)
- void glPixelMapuiv(GLenum map,GLsizei mapsize,const GLuint * values)
- void glPixelMapusv(GLenum map,GLsizei mapsize,const GLushort * values)
- void glPixelStoref(GLenum pname,GLfloat param)
- void glPixelStorei(GLenum pname,GLint param)
- void glPixelTransferf(GLenum pname,GLfloat param)
- void glPixelTransferi(GLenum pname,GLint param)
- void glPixelZoom(GLfloat xfactor,GLfloat yfactor)
- void glPointParameterf(GLenum pname,GLfloat param)
- void glPointParameteri(GLenum pname,GLint param)
- void glPointSize(GLfloat size)
- void glPolygonMode(GLenum face,GLenum mode)
- void glPolygonOffset(GLfloat factor,GLfloat units)
- void glPolygonStipple(const GLubyte * pattern)

- void glPushAttrib(GLbitfield mask)
- void glPushClientAttrib(GLbitfield mask)
- void glPushMatrix(void)
- void glPushName(GLuint name)
- void glPrioritizeTextures(GLsizei n,const GLuint * textures,const GLclampf * priorities)
- void glPopMatrix(void)
- void glRasterPos2s(GLshort x,GLshort y)
- void glRasterPos2i(GLint x,GLint y)
- void glRasterPos2f(GLfloat x,GLfloat y)
- void glRasterPos2d(GLdouble x,GLdouble y)
- void glRasterPos3s(GLshort x,GLshort y,GLshort z)
- void glRasterPos3i(GLint x,GLint y,GLint z)
- void glRasterPos3f(GLfloat x,GLfloat y,GLfloat z)
- void glRasterPos3d(GLdouble x,GLdouble y,GLdouble z)
- void glRasterPos4s(GLshort x,GLshort y,GLshort z,GLshort w)
- void glRasterPos4i(GLint x,GLint y,GLint z,GLint w)
- void glRasterPos4f(GLfloat x,GLfloat y,GLfloat z,GLfloat w)
- void glRasterPos4d(GLdouble x,GLdouble y,GLdouble z,GLdouble w)
- void glReadBuffer(GLenum mode)
- void glReadPixels(GLint x,GLint y,GLsizei width,GLsizei height,GLenum format,GLenum type,GLvoid * data)
- void glRectd(GLdouble x1,GLdouble y1,GLdouble x2,GLdouble y2)
- void glRectf(GLfloat x1,GLfloat y1,GLfloat x2,GLfloat y2)
- void glRecti(GLint x1,GLint y1,GLint x2,GLint y2)
- void glRects(GLshort x1,GLshort y1,GLshort x2,GLshort y2)
- void glRectdv(const GLdouble * v1,const GLdouble * v2)
- void glRectfv(const GLfloat * v1,const GLfloat * v2)
- void glRectiv(const GLint * v1,const GLint * v2)
- void glRectsv(const GLshort * v1,const GLshort * v2)
- GLint glRenderMode(GLenum mode)
- void glResetHistogram(GLenum target)
- void glRotated(GLdouble angle,GLdouble x,GLdouble y,GLdouble z)
- void glRotatef(GLfloat angle,GLfloat x,GLfloat y,GLfloat z)
- void glSampleCoverage(GLclampf value,GLboolean invert)
- void glScaled(GLdouble x,GLdouble y,GLdouble z)
- void glScalef(GLfloat x,GLfloat y,GLfloat z)
- void glScissor(GLint x,GLint y,GLsizei width,GLsizei height)

- void glSecondaryColor3b(GLbyte red, GLbyte green, GLbyte blue)
- void glSecondaryColor3s(GLshort red, GLshort green, GLshort blue)
- void glSecondaryColor3i(GLint red, GLint green, GLint blue)
- void glSecondaryColor3f(GLfloat red, GLfloat green, GLfloat blue)
- void glSecondaryColor3d(GLdouble red, GLdouble green, GLdouble blue)
- void glSecondaryColor3ub(GLubyte red, GLubyte green, GLubyte blue)
- void glSecondaryColor3us(GLushort red, GLushort green, GLushort blue)
- void glSecondaryColor3ui(GLuint red, GLuint green, GLuint blue)
- void glSecondaryColor3bv(const GLbyte * v)
- void glSecondaryColor3sv(const GLshort * v)
- void glSecondaryColor3iv(const GLint * v)
- void glSecondaryColor3fv(const GLfloat * v)
- void glSecondaryColor3dv(const GLdouble * v)
- void glSecondaryColor3ubv(const GLubyte * v)
- void glSecondaryColor3usv(const GLushort * v)
- void glSecondaryColor3uiv(const GLuint * v)
- void glSecondaryColorPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)
- void glSelectBuffer(GLsizei size, GLuint * buffer)
- void glSeparableFilter2D(GLenum target, GLenum internalformat, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * row, const GLvoid * column)
- void glShadeModel(GLenum mode)
- void glShaderSource(GLuint shader, GLsizei count, const GLchar **string, const GLint *length)
- void glStencilFunc(GLenum func, GLint ref, GLuint mask)
- void glStencilFuncSeparate(GLenum face, GLenum func, GLint ref, GLuint mask)
- void glStencilMask(GLuint mask)
- void glStencilMaskSeparate(GLenum face, GLuint mask)
- void glStencilOp(GLenum sfail, GLenum dpfail, GLenum dppass)
- void glStencilOpSeparate(GLenum face, GLenum sfail, GLenum dpfail, GLenum dppass)
- void glTexCoord1s(GLshort s)
- void glTexCoord1i(GLint s)
- void glTexCoord1f(GLfloat s)
- void glTexCoord1d(GLdouble s)
- void glTexCoord2s(GLshort s, GLshort t)
- void glTexCoord2i(GLint s, GLint t)
- void glTexCoord2f(GLfloat s, GLfloat t)
- void glTexCoord2d(GLdouble s, GLdouble t)

- `void glTexCoord3s(GLshort s, GLshort t, GLshort r)`
- `void glTexCoord3i(GLint s, GLint t, GLint r)`
- `void glTexCoord3f(GLfloat s, GLfloat t, GLfloat r)`
- `void glTexCoord3d(GLdouble s, GLdouble t, GLdouble r)`
- `void glTexCoord4s(GLshort s, GLshort t, GLshort r, GLshort q)`
- `void glTexCoord4i(GLint s, GLint t, GLint r, GLint q)`
- `void glTexCoord4f(GLfloat s, GLfloat t, GLfloat r, GLfloat q)`
- `void glTexCoord4d(GLdouble s, GLdouble t, GLdouble r, GLdouble q)`
- `void glTexCoord1sv(const GLshort * v)`
- `void glTexCoord1iv(const GLint * v)`
- `void glTexCoord1fv(const GLfloat * v)`
- `void glTexCoord1dv(const GLdouble * v)`
- `void glTexCoord2sv(const GLshort * v)`
- `void glTexCoord2iv(const GLint * v)`
- `void glTexCoord2fv(const GLfloat * v)`
- `void glTexCoord2dv(const GLdouble * v)`
- `void glTexCoord3sv(const GLshort * v)`
- `void glTexCoord3iv(const GLint * v)`
- `void glTexCoord3fv(const GLfloat * v)`
- `void glTexCoord3dv(const GLdouble * v)`
- `void glTexCoord4sv(const GLshort * v)`
- `void glTexCoord4iv(const GLint * v)`
- `void glTexCoord4fv(const GLfloat * v)`
- `void glTexCoord4dv(const GLdouble * v)`
- `void glTexCoordPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)`
- `void glTexEnvf(GLenum target, GLenum pname, GLfloat param)`
- `void glTexEnvf(GLenum target, GLenum pname, GLint param)`
- `void glTexGeni(GLenum coord, GLenum pname, GLint param)`
- `void glTexGenf(GLenum coord, GLenum pname, GLfloat param)`
- `void glTexGend(GLenum coord, GLenum pname, GLdouble param)`
- `void glTexGeniv(GLenum coord, GLenum pname, const GLint * params)`
- `void glTexGenfv(GLenum coord, GLenum pname, const GLfloat * params)`
- `void glTexGendv(GLenum coord, GLenum pname, const GLdouble * params)`
- `void glTexImage1D(GLenum target, GLint level, GLint internalFormat, GLsizei width, GLint border, GLenum format, GLenum type, const GLvoid * data)`
- `void glTexImage2D(GLenum target, GLint level, GLint internalFormat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, const GLvoid * data)`

- void glTexImage3D(GLenum target,GLint level,GLint internalFormat,GLsizei width,GLsizei height,GLsizei depth,GLint border,GLenum format,GLenum type,const GLvoid * data)
- void glTexParameterf(GLenum target,GLenum pname,GLfloat param)
- void glTexParameteri(GLenum target,GLenum pname,GLint param)
- void glTexParameterfv(GLenum target,GLenum pname,const GLfloat * params)
- void glTexParameteriv(GLenum target,GLenum pname,const GLint * params)
- void glTexSubImage1D(GLenum target,GLint level,GLint xoffset,GLsizei width,GLenum format,GLenum type,const GLvoid * data)
- void glTexSubImage2D(GLenum target,GLint level,GLint xoffset,GLint yoffset,GLsizei width,GLsizei height,GLenum format,GLenum type,const GLvoid * data)
- void glTexSubImage3D(GLenum target,GLint level,GLint xoffset,GLint yoffset,GLint zoffset,GLsizei width,GLsizei height,GLsizei depth,GLenum format,GLenum type,const GLvoid * data)
- void glTranslated(GLdouble x,GLdouble y,GLdouble z)
- void glTranslatef(GLfloat x,GLfloat y,GLfloat z)
- void glUniform1f(GLint location,GLfloat v0)
- void glUniform2f(GLint location,GLfloat v0,GLfloat v1)
- void glUniform3f(GLint location,GLfloat v0,GLfloat v1,GLfloat v2)
- void glUniform4f(GLint location,GLfloat v0,GLfloat v1,GLfloat v2,GLfloat v3)
- void glUniform1i(GLint location,GLint v0)
- void glUniform2i(GLint location,GLint v0,GLint v1)
- void glUniform3i(GLint location,GLint v0,GLint v1,GLint v2)
- void glUniform4i(GLint location,GLint v0,GLint v1,GLint v2,GLint v3)
- void glUniform1fv(GLint location,GLsizei count,const GLfloat *value)
- void glUniform2fv(GLint location,GLsizei count,const GLfloat *value)
- void glUniform3fv(GLint location,GLsizei count,const GLfloat *value)
- void glUniform4fv(GLint location,GLsizei count,const GLfloat *value)
- void glUniform1iv(GLint location,GLsizei count,const GLint *value)
- void glUniform2iv(GLint location,GLsizei count,const GLint *value)
- void glUniform3iv(GLint location,GLsizei count,const GLint *value)
- void glUniform4iv(GLint location,GLsizei count,const GLint *value)
- void glUniformMatrix2fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix3fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix4fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix2x3fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix3x2fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix2x4fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix4x2fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)

- void glUniformMatrix3x4fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix4x3fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUseProgram(GLuint program)
- void glValidateProgram(GLuint program)
- void glVertex2s(GLshort x,GLshort y)
- void glVertex2i(GLint x,GLint y)
- void glVertex2f(GLfloat x,GLfloat y)
- void glVertex2d(GLdouble x,GLdouble y)
- void glVertex3s(GLshort x,GLshort y,GLshort z)
- void glVertex3i(GLint x,GLint y,GLint z)
- void glVertex3f(GLfloat x,GLfloat y,GLfloat z)
- void glVertex3d(GLdouble x,GLdouble y,GLdouble z)
- void glVertex4s(GLshort x,GLshort y,GLshort z,GLshort w)
- void glVertex4i(GLint x,GLint y,GLint z,GLint w)
- void glVertex4f(GLfloat x,GLfloat y,GLfloat z,GLfloat w)
- void glVertex4d(GLdouble x,GLdouble y,GLdouble z,GLdouble w)
- void glVertex2sv(const GLshort * v)
- void glVertex2iv(const GLint * v)
- void glVertex2fv(const GLfloat * v)
- void glVertex2dv(const GLdouble * v)
- void glVertex3sv(const GLshort * v)
- void glVertex3iv(const GLint * v)
- void glVertex3fv(const GLfloat * v)
- void glVertex3dv(const GLdouble * v)
- void glVertex4sv(const GLshort * v)
- void glVertex4iv(const GLint * v)
- void glVertex4fv(const GLfloat * v)
- void glVertex4dv(const GLdouble * v)
- void glVertexAttrib1f(GLuint index,GLfloat v0)
- void glVertexAttrib1s(GLuint index,GLshort v0)
- void glVertexAttrib1d(GLuint index,GLdouble v0)
- void glVertexAttrib2f(GLuint index,GLfloat v0,GLfloat v1)
- void glVertexAttrib2s(GLuint index,GLshort v0,GLshort v1)
- void glVertexAttrib2d(GLuint index,GLdouble v0,GLdouble v1)
- void glVertexAttrib3f(GLuint index,GLfloat v0,GLfloat v1,GLfloat v2)
- void glVertexAttrib3s(GLuint index,GLshort v0,GLshort v1,GLshort v2)

- void glVertexAttrib3d(GLuint index, GLdouble v0, GLdouble v1, GLdouble v2)
- void glVertexAttrib4f(GLuint index, GLfloat v0, GLfloat v1, GLfloat v2, GLfloat v3)
- void glVertexAttrib4s(GLuint index, GLshort v0, GLshort v1, GLshort v2, GLshort v3)
- void glVertexAttrib4d(GLuint index, GLdouble v0, GLdouble v1, GLdouble v2, GLdouble v3)
- void glVertexAttrib4Nub(GLuint index, GLubyte v0, GLubyte v1, GLubyte v2, GLubyte v3)
- void glVertexAttrib1fv(GLuint index, const GLfloat *v)
- void glVertexAttrib1sv(GLuint index, const GLshort *v)
- void glVertexAttrib1dv(GLuint index, const GLdouble *v)
- void glVertexAttrib2fv(GLuint index, const GLfloat *v)
- void glVertexAttrib2sv(GLuint index, const GLshort *v)
- void glVertexAttrib2dv(GLuint index, const GLdouble *v)
- void glVertexAttrib3fv(GLuint index, const GLfloat *v)
- void glVertexAttrib3sv(GLuint index, const GLshort *v)
- void glVertexAttrib3dv(GLuint index, const GLdouble *v)
- void glVertexAttrib4fv(GLuint index, const GLfloat *v)
- void glVertexAttrib4sv(GLuint index, const GLshort *v)
- void glVertexAttrib4dv(GLuint index, const GLdouble *v)
- void glVertexAttrib4iv(GLuint index, const GLint *v)
- void glVertexAttrib4bv(GLuint index, const GLbyte *v)
- void glVertexAttrib4ubv(GLuint index, const GLubyte *v)
- void glVertexAttrib4usv(GLuint index, const GLushort *v)
- void glVertexAttrib4uiv(GLuint index, const GLuint *v)
- void glVertexAttribPointer(GLuint index, GLint size, GLenum type, GLboolean normalized, GLsizei stride, const GLvoid * pointer)
- void glVertexPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)
- void glViewport(GLint x, GLint y, GLsizei width, GLsizei height)
- void glWindowPos2s(GLshort x, GLshort y)
- void glWindowPos2i(GLint x, GLint y)
- void glWindowPos2f(GLfloat x, GLfloat y)
- void glWindowPos2d(GLdouble x, GLdouble y)
- void glWindowPos3s(GLshort x, GLshort y, GLshort z)
- void glWindowPos3i(GLint x, GLint y, GLint z)
- void glWindowPos3f(GLfloat x, GLfloat y, GLfloat z)
- void glWindowPos3d(GLdouble x, GLdouble y, GLdouble z)
- void glWindowPos2sv(const GLshort * v)
- void glWindowPos2iv(const GLint * v)

- void glWindowPos2fv(const GLfloat * v)
- void glWindowPos2dv(const GLdouble * v)
- void glWindowPos3sv(const GLshort * v)
- void glWindowPos3iv(const GLint * v)
- void glWindowPos3fv(const GLfloat * v)
- void glWindowPos3dv(const GLdouble * v)
- void gluBeginCurve(GLUnurbs* nurb)
- void gluBeginPolygon(GLUtesselator* tess)
- void gluBeginSurface(GLUnurbs* nurb)
- void gluBeginTrim(GLUnurbs* nurb)
- void gluCylinder(GLUquadric* quad, GLdouble base, GLdouble top, GLdouble height, GLint slices, GLint stacks)
- void gluDeleteNurbsRenderer(GLUnurbs* nurb)
- void gluDeleteQuadric(GLUquadric* quad)
- void gluDeleteTess(GLUtesselator* tess)
- void gluDisk(GLUquadric* quad, GLdouble inner, GLdouble outer, GLint slices, GLint loops)
- void gluEndCurve(GLUnurbs* nurb)
- void gluEndPolygon(GLUtesselator* tess)
- void gluEndSurface(GLUnurbs* nurb)
- void gluEndTrim(GLUnurbs* nurb)
- const GLubyte * gluErrorString(GLenum error)
- void gluGetNurbsProperty(GLUnurbs* nurb, GLenum property, GLfloat* data)
- const GLubyte * gluGetString(GLenum name)
- void gluGetTessProperty(GLUtesselator* tess, GLenum which, GLdouble* data)
- void gluLoadSamplingMatrices(GLUnurbs* nurb, const GLfloat * model, const GLfloat * perspective, const GLint * view)
- void gluLookAt(GLdouble eyeX, GLdouble eyeY, GLdouble eyeZ, GLdouble centerX, GLdouble centerY, GLdouble centerZ, GLdouble upX, GLdouble upY, GLdouble upZ)
- GLUnurbs *gluNewNurbsRenderer(void)
- GLUquadric *gluNewQuadric(void)
- GLUtesselator* gluNewTess(void)
- void gluNextContour(GLUtesselator* tess, GLenum type)
- void gluNurbsCurve(GLUnurbs* nurb, GLint knotCount, GLfloat * knots, GLint stride, GLfloat * control, GLint order, GLenum type)
- void gluNurbsProperty(GLUnurbs* nurb, GLenum property, GLfloat value)
- void gluNurbsSurface(GLUnurbs* nurb, GLint sKnotCount, GLfloat* sKnots, GLint tKnotCount, GLfloat* tKnots, GLint sStride, GLint tStride, GLfloat* control, GLint sOrder, GLint tOrder, GLenum type)
- void gluOrtho2D(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top)

- void gluPartialDisk(GLUquadric* quad, GLdouble inner, GLdouble outer, GLint slices, GLint loops, GLdouble start, GLdouble sweep)
- void gluPerspective(GLdouble fovy, GLdouble aspect, GLdouble zNear, GLdouble zFar)
- void gluPickMatrix(GLdouble x, GLdouble y, GLdouble delX, GLdouble delY, GLint * viewport)
- GLint gluProject(GLdouble objX, GLdouble objY, GLdouble objZ, const GLdouble * model, const GLdouble * proj, const GLint * view, GLdouble* winX, GLdouble* winY, GLdouble* winZ)
- void gluPwlCurve(GLUnurbs* nurb, GLint count, GLfloat* data, GLint stride, GLenum type)
- void gluQuadricDrawStyle(GLUquadric* quad, GLenum draw)
- void gluQuadricNormals(GLUquadric* quad, GLenum normal)
- void gluQuadricOrientation(GLUquadric* quad, GLenum orientation)
- void gluQuadricTexture(GLUquadric* quad, GLboolean texture)
- GLint gluScaleImage(GLenum format, GLsizei wIn, GLsizei hIn, GLenum typeIn, const void * dataIn, GLsizei wOut, GLsizei hOut, GLenum typeOut, GLvoid* dataOut)
- void gluSphere(GLUquadric* quad, GLdouble radius, GLint slices, GLint stacks)
- void gluTessBeginContour(GLUtesselator* tess)
- void gluTessBeginPolygon(GLUtesselator* tess, GLvoid* data)
- void gluTessEndContour(GLUtesselator* tess)
- void gluTessEndPolygon(GLUtesselator* tess)
- void gluTessNormal(GLUtesselator* tess, GLdouble valueX, GLdouble valueY, GLdouble valueZ)
- void gluTessProperty(GLUtesselator* tess, GLenum which, GLdouble data)
- void gluTessVertex(GLUtesselator* tess, GLdouble * location, GLvoid* data)
- GLint gluUnProject(GLdouble winX, GLdouble winY, GLdouble winZ, const GLdouble * model, const GLdouble * proj, const GLint * view, GLdouble* objX, GLdouble* objY, GLdouble* objZ)
- void glDisable(GLenum cap)

RINGOPENGL (OPENGL 3.0) FUNCTIONS REFERENCE

- GL_ZERO
- GL_FALSE
- GL_LOGIC_OP
- GL_NONE
- GL_TEXTURE_COMPONENTS
- GL_NO_ERROR
- GL_POINTS
- GL_CURRENT_BIT
- GL_TRUE
- GL_ONE
- GL_CLIENT_PIXEL_STORE_BIT
- GL_LINES
- GL_LINE_LOOP
- GL_POINT_BIT
- GL_CLIENT_VERTEX_ARRAY_BIT
- GL_LINE_STRIP
- GL_LINE_BIT
- GL_TRIANGLES
- GL_TRIANGLE_STRIP
- GL_TRIANGLE_FAN
- GL_QUADS
- GL_QUAD_STRIP
- GL_POLYGON_BIT
- GL_POLYGON
- GL_POLYGON_STIPPLE_BIT
- GL_PIXEL_MODE_BIT
- GL_LIGHTING_BIT

- GL_FOG_BIT
- GL_DEPTH_BUFFER_BIT
- GL_ACCUM
- GL_LOAD
- GL_RETURN
- GL_MULT
- GL_ADD
- GL_NEVER
- GL_ACCUM_BUFFER_BIT
- GL_LESS
- GL_EQUAL
- GL_LEQUAL
- GL_GREATER
- GL_NOTEQUAL
- GL_GEQUAL
- GL_ALWAYS
- GL_SRC_COLOR
- GL_ONE_MINUS_SRC_COLOR
- GL_SRC_ALPHA
- GL_ONE_MINUS_SRC_ALPHA
- GL_DST_ALPHA
- GL_ONE_MINUS_DST_ALPHA
- GL_DST_COLOR
- GL_ONE_MINUS_DST_COLOR
- GL_SRC_ALPHA_SATURATE
- GL_STENCIL_BUFFER_BIT
- GL_FRONT_LEFT
- GL_FRONT_RIGHT
- GL_BACK_LEFT
- GL_BACK_RIGHT
- GL_FRONT
- GL_BACK
- GL_LEFT
- GL_RIGHT
- GL_FRONT_AND_BACK
- GL_AUX0

- GL_AUX1
- GL_AUX2
- GL_AUX3
- GL_INVALID_ENUM
- GL_INVALID_VALUE
- GL_INVALID_OPERATION
- GL_STACK_OVERFLOW
- GL_STACK_UNDERFLOW
- GL_OUT_OF_MEMORY
- GL_2D
- GL_3D
- GL_3D_COLOR
- GL_3D_COLOR_TEXTURE
- GL_4D_COLOR_TEXTURE
- GL_PASS_THROUGH_TOKEN
- GL_POINT_TOKEN
- GL_LINE_TOKEN
- GL_POLYGON_TOKEN
- GL_BITMAP_TOKEN
- GL_DRAW_PIXEL_TOKEN
- GL_COPY_PIXEL_TOKEN
- GL_LINE_RESET_TOKEN
- GL_EXP
- GL_VIEWPORT_BIT
- GL_EXP2
- GL_CW
- GL_CCW
- GL_COEFF
- GL_ORDER
- GL_DOMAIN
- GL_CURRENT_COLOR
- GL_CURRENT_INDEX
- GL_CURRENT_NORMAL
- GL_CURRENT_TEXTURE_COORDS
- GL_CURRENT_RASTER_COLOR
- GL_CURRENT_RASTER_INDEX

- GL_CURRENT_RASTER_TEXTURE_COORDS
- GL_CURRENT_RASTER_POSITION
- GL_CURRENT_RASTER_POSITION_VALID
- GL_CURRENT_RASTER_DISTANCE
- GL_POINT_SMOOTH
- GL_POINT_SIZE
- GL_POINT_SIZE_RANGE
- GL_POINT_SIZE_GRANULARITY
- GL_LINE_SMOOTH
- GL_LINE_WIDTH
- GL_LINE_WIDTH_RANGE
- GL_LINE_WIDTH_GRANULARITY
- GL_LINE_STIPPLE
- GL_LINE_STIPPLE_PATTERN
- GL_LINE_STIPPLE_REPEAT
- GL_LIST_MODE
- GL_MAX_LIST_NESTING
- GL_LIST_BASE
- GL_LIST_INDEX
- GL_POLYGON_MODE
- GL_POLYGON_SMOOTH
- GL_POLYGON_STIPPLE
- GL_EDGE_FLAG
- GL_CULL_FACE
- GL_CULL_FACE_MODE
- GL_FRONT_FACE
- GL_LIGHTING
- GL_LIGHT_MODEL_LOCAL_VIEWER
- GL_LIGHT_MODEL_TWO_SIDE
- GL_LIGHT_MODEL_AMBIENT
- GL_SHADE_MODEL
- GL_COLOR_MATERIAL_FACE
- GL_COLOR_MATERIAL_PARAMETER
- GL_COLOR_MATERIAL
- GL_FOG
- GL_FOG_INDEX

- GL_FOG_DENSITY
- GL_FOG_START
- GL_FOG_END
- GL_FOG_MODE
- GL_FOG_COLOR
- GL_DEPTH_RANGE
- GL_DEPTH_TEST
- GL_DEPTH_WRITEMASK
- GL_DEPTH_CLEAR_VALUE
- GL_DEPTH_FUNC
- GL_ACCUM_CLEAR_VALUE
- GL_STENCIL_TEST
- GL_STENCIL_CLEAR_VALUE
- GL_STENCIL_FUNC
- GL_STENCIL_VALUE_MASK
- GL_STENCIL_FAIL
- GL_STENCIL_PASS_DEPTH_FAIL
- GL_STENCIL_PASS_DEPTH_PASS
- GL_STENCIL_REF
- GL_STENCIL_WRITEMASK
- GL_MATRIX_MODE
- GL_NORMALIZE
- GL_VIEWPORT
- GL_MODELVIEW_STACK_DEPTH
- GL_PROJECTION_STACK_DEPTH
- GL_TEXTURE_STACK_DEPTH
- GL_MODELVIEW_MATRIX
- GL_PROJECTION_MATRIX
- GL_TEXTURE_MATRIX
- GL_ATTRIB_STACK_DEPTH
- GL_CLIENT_ATTRIB_STACK_DEPTH
- GL_ALPHA_TEST
- GL_ALPHA_TEST_FUNC
- GL_ALPHA_TEST_REF
- GL_DITHER
- GL_BLEND_DST

- GL_BLEND_SRC
- GL_BLEND
- GL_LOGIC_OP_MODE
- GL_INDEX_LOGIC_OP
- GL_COLOR_LOGIC_OP
- GL_AUX_BUFFERS
- GL_DRAW_BUFFER
- GL_READ_BUFFER
- GL_SCISSOR_BOX
- GL_SCISSOR_TEST
- GL_INDEX_CLEAR_VALUE
- GL_INDEX_WRITEMASK
- GL_COLOR_CLEAR_VALUE
- GL_COLOR_WRITEMASK
- GL_INDEX_MODE
- GL_RGBA_MODE
- GL_DOUBLEBUFFER
- GL_STEREO
- GL_RENDER_MODE
- GL_PERSPECTIVE_CORRECTION_HINT
- GL_POINT_SMOOTH_HINT
- GL_LINE_SMOOTH_HINT
- GL_POLYGON_SMOOTH_HINT
- GL_FOG_HINT
- GL_TEXTURE_GEN_S
- GL_TEXTURE_GEN_T
- GL_TEXTURE_GEN_R
- GL_TEXTURE_GEN_Q
- GL_PIXEL_MAP_I_TO_I
- GL_PIXEL_MAP_S_TO_S
- GL_PIXEL_MAP_I_TO_R
- GL_PIXEL_MAP_I_TO_G
- GL_PIXEL_MAP_I_TO_B
- GL_PIXEL_MAP_I_TO_A
- GL_PIXEL_MAP_R_TO_R
- GL_PIXEL_MAP_G_TO_G

- GL_PIXEL_MAP_B_TO_B
- GL_PIXEL_MAP_A_TO_A
- GL_PIXEL_MAP_I_TO_I_SIZE
- GL_PIXEL_MAP_S_TO_S_SIZE
- GL_PIXEL_MAP_I_TO_R_SIZE
- GL_PIXEL_MAP_I_TO_G_SIZE
- GL_PIXEL_MAP_I_TO_B_SIZE
- GL_PIXEL_MAP_I_TO_A_SIZE
- GL_PIXEL_MAP_R_TO_R_SIZE
- GL_PIXEL_MAP_G_TO_G_SIZE
- GL_PIXEL_MAP_B_TO_B_SIZE
- GL_PIXEL_MAP_A_TO_A_SIZE
- GL_UNPACK_SWAP_BYTES
- GL_UNPACK_LSB_FIRST
- GL_UNPACK_ROW_LENGTH
- GL_UNPACK_SKIP_ROWS
- GL_UNPACK_SKIP_PIXELS
- GL_UNPACK_ALIGNMENT
- GL_PACK_SWAP_BYTES
- GL_PACK_LSB_FIRST
- GL_PACK_ROW_LENGTH
- GL_PACK_SKIP_ROWS
- GL_PACK_SKIP_PIXELS
- GL_PACK_ALIGNMENT
- GL_MAP_COLOR
- GL_MAP_STENCIL
- GL_INDEX_SHIFT
- GL_INDEX_OFFSET
- GL_RED_SCALE
- GL_RED_BIAS
- GL_ZOOM_X
- GL_ZOOM_Y
- GL_GREEN_SCALE
- GL_GREEN_BIAS
- GL_BLUE_SCALE
- GL_BLUE_BIAS

- GL_ALPHA_SCALE
- GL_ALPHA_BIAS
- GL_DEPTH_SCALE
- GL_DEPTH_BIAS
- GL_MAX_EVAL_ORDER
- GL_MAX_LIGHTS
- GL_MAX_CLIP_PLANES
- GL_MAX_TEXTURE_SIZE
- GL_MAX_PIXEL_MAP_TABLE
- GL_MAX_ATTRIB_STACK_DEPTH
- GL_MAX_MODELVIEW_STACK_DEPTH
- GL_MAX_NAME_STACK_DEPTH
- GL_MAX_PROJECTION_STACK_DEPTH
- GL_MAX_TEXTURE_STACK_DEPTH
- GL_MAX_VIEWPORT_DIMS
- GL_MAX_CLIENT_ATTRIB_STACK_DEPTH
- GL_SUBPIXEL_BITS
- GL_INDEX_BITS
- GL_RED_BITS
- GL_GREEN_BITS
- GL_BLUE_BITS
- GL_ALPHA_BITS
- GL_DEPTH_BITS
- GL_STENCIL_BITS
- GL_ACCUM_RED_BITS
- GL_ACCUM_GREEN_BITS
- GL_ACCUM_BLUE_BITS
- GL_ACCUM_ALPHA_BITS
- GL_NAME_STACK_DEPTH
- GL_AUTO_NORMAL
- GL_MAP1_COLOR_4
- GL_MAP1_INDEX
- GL_MAP1_NORMAL
- GL_MAP1_TEXTURE_COORD_1
- GL_MAP1_TEXTURE_COORD_2
- GL_MAP1_TEXTURE_COORD_3

- GL_MAP1_TEXTURE_COORD_4
- GL_MAP1_VERTEX_3
- GL_MAP1_VERTEX_4
- GL_MAP2_COLOR_4
- GL_MAP2_INDEX
- GL_MAP2_NORMAL
- GL_MAP2_TEXTURE_COORD_1
- GL_MAP2_TEXTURE_COORD_2
- GL_MAP2_TEXTURE_COORD_3
- GL_MAP2_TEXTURE_COORD_4
- GL_MAP2_VERTEX_3
- GL_MAP2_VERTEX_4
- GL_MAP1_GRID_DOMAIN
- GL_MAP1_GRID_SEGMENTS
- GL_MAP2_GRID_DOMAIN
- GL_MAP2_GRID_SEGMENTS
- GL_TEXTURE_1D
- GL_TEXTURE_2D
- GL_FEEDBACK_BUFFER_POINTER
- GL_FEEDBACK_BUFFER_SIZE
- GL_FEEDBACK_BUFFER_TYPE
- GL_SELECTION_BUFFER_POINTER
- GL_SELECTION_BUFFER_SIZE
- GL_TEXTURE_WIDTH
- GL_TRANSFORM_BIT
- GL_TEXTURE_HEIGHT
- GL_TEXTURE_INTERNAL_FORMAT
- GL_TEXTURE_BORDER_COLOR
- GL_TEXTURE_BORDER
- GL_DONT_CARE
- GL_FASTEST
- GL_NICEST
- GL_AMBIENT
- GL_DIFFUSE
- GL_SPECULAR
- GL_POSITION

- GL_SPOT_DIRECTION
- GL_SPOT_EXPONENT
- GL_SPOT_CUTOFF
- GL_CONSTANT_ATTENUATION
- GL_LINEAR_ATTENUATION
- GL_QUADRATIC_ATTENUATION
- GL_COMPILE
- GL_COMPILE_AND_EXECUTE
- GL_BYTE
- GL_UNSIGNED_BYTE
- GL_SHORT
- GL_UNSIGNED_SHORT
- GL_INT
- GL_UNSIGNED_INT
- GL_FLOAT
- GL_2_BYTES
- GL_3_BYTES
- GL_4_BYTES
- GL_DOUBLE
- GL_CLEAR
- GL_AND
- GL_AND_REVERSE
- GL_COPY
- GL_AND_INVERTED
- GL_NOOP
- GL_XOR
- GL_OR
- GL_NOR
- GL_EQUIV
- GL_INVERT
- GL_OR_REVERSE
- GL_COPY_INVERTED
- GL_OR_INVERTED
- GL_NAND
- GL_SET
- GL_EMISSION

- GL_SHININESS
- GL_AMBIENT_AND_DIFFUSE
- GL_COLOR_INDEXES
- GL_MODELVIEW
- GL_PROJECTION
- GL_TEXTURE
- GL_COLOR
- GL_DEPTH
- GL_STENCIL
- GL_COLOR_INDEX
- GL_STENCIL_INDEX
- GL_DEPTH_COMPONENT
- GL_RED
- GL_GREEN
- GL_BLUE
- GL_ALPHA
- GL_RGB
- GL_RGBA
- GL_LUMINANCE
- GL_LUMINANCE_ALPHA
- GL_BITMAP
- GL_POINT
- GL_LINE
- GL_FILL
- GL_RENDER
- GL_FEEDBACK
- GL_SELECT
- GL_FLAT
- GL_SMOOTH
- GL_KEEP
- GL_REPLACE
- GL_INCR
- GL_DECR
- GL_VENDOR
- GL_RENDERER
- GL_VERSION

- GL_EXTENSIONS
- GL_S
- GL_ENABLE_BIT
- GL_T
- GL_R
- GL_Q
- GL_MODULATE
- GL_DECAL
- GL_TEXTURE_ENV_MODE
- GL_TEXTURE_ENV_COLOR
- GL_TEXTURE_ENV
- GL_EYE_LINEAR
- GL_OBJECT_LINEAR
- GL_SPHERE_MAP
- GL_TEXTURE_GEN_MODE
- GL_OBJECT_PLANE
- GL_EYE_PLANE
- GL_NEAREST
- GL_LINEAR
- GL_NEAREST_MIPMAP_NEAREST
- GL_LINEAR_MIPMAP_NEAREST
- GL_NEAREST_MIPMAP_LINEAR
- GL_LINEAR_MIPMAP_LINEAR
- GL_TEXTURE_MAG_FILTER
- GL_TEXTURE_MIN_FILTER
- GL_TEXTURE_WRAP_S
- GL_TEXTURE_WRAP_T
- GL_CLAMP
- GL_REPEAT
- GL_POLYGON_OFFSET_UNITS
- GL_POLYGON_OFFSET_POINT
- GL_POLYGON_OFFSET_LINE
- GL_R3_G3_B2
- GL_V2F
- GL_V3F
- GL_C4UB_V2F

- GL_C4UB_V3F
- GL_C3F_V3F
- GL_N3F_V3F
- GL_C4F_N3F_V3F
- GL_T2F_V3F
- GL_T4F_V4F
- GL_T2F_C4UB_V3F
- GL_T2F_C3F_V3F
- GL_T2F_N3F_V3F
- GL_T2F_C4F_N3F_V3F
- GL_T4F_C4F_N3F_V4F
- GL_CLIP_PLANE0
- GL_CLIP_PLANE1
- GL_CLIP_PLANE2
- GL_CLIP_PLANE3
- GL_CLIP_PLANE4
- GL_CLIP_PLANE5
- GL_LIGHT0
- GL_COLOR_BUFFER_BIT
- GL_LIGHT1
- GL_LIGHT2
- GL_LIGHT3
- GL_LIGHT4
- GL_LIGHT5
- GL_LIGHT6
- GL_LIGHT7
- GL_HINT_BIT
- GL_POLYGON_OFFSET_FILL
- GL_POLYGON_OFFSET_FACTOR
- GL_ALPHA4
- GL_ALPHA8
- GL_ALPHA12
- GL_ALPHA16
- GL_LUMINANCE4
- GL_LUMINANCE8
- GL_LUMINANCE12

- GL_LUMINANCE16
- GL_LUMINANCE4_ALPHA4
- GL_LUMINANCE6_ALPHA2
- GL_LUMINANCE8_ALPHA8
- GL_LUMINANCE12_ALPHA4
- GL_LUMINANCE12_ALPHA12
- GL_LUMINANCE16_ALPHA16
- GL_INTENSITY
- GL_INTENSITY4
- GL_INTENSITY8
- GL_INTENSITY12
- GL_INTENSITY16
- GL_RGB4
- GL_RGB5
- GL_RGB8
- GL_RGB10
- GL_RGB12
- GL_RGB16
- GL_RGBA2
- GL_RGBA4
- GL_RGB5_A1
- GL_RGBA8
- GL_RGB10_A2
- GL_RGBA12
- GL_RGBA16
- GL_TEXTURE_RED_SIZE
- GL_TEXTURE_GREEN_SIZE
- GL_TEXTURE_BLUE_SIZE
- GL_TEXTURE_ALPHA_SIZE
- GL_TEXTURE_LUMINANCE_SIZE
- GL_TEXTURE_INTENSITY_SIZE
- GL_PROXY_TEXTURE_1D
- GL_PROXY_TEXTURE_2D
- GL_TEXTURE_PRIORITY
- GL_TEXTURE_RESIDENT
- GL_TEXTURE_BINDING_1D

- GL_TEXTURE_BINDING_2D
- GL_VERTEX_ARRAY
- GL_NORMAL_ARRAY
- GL_COLOR_ARRAY
- GL_INDEX_ARRAY
- GL_TEXTURE_COORD_ARRAY
- GL_EDGE_FLAG_ARRAY
- GL_VERTEX_ARRAY_SIZE
- GL_VERTEX_ARRAY_TYPE
- GL_VERTEX_ARRAY_STRIDE
- GL_NORMAL_ARRAY_TYPE
- GL_NORMAL_ARRAY_STRIDE
- GL_COLOR_ARRAY_SIZE
- GL_COLOR_ARRAY_TYPE
- GL_COLOR_ARRAY_STRIDE
- GL_INDEX_ARRAY_TYPE
- GL_INDEX_ARRAY_STRIDE
- GL_TEXTURE_COORD_ARRAY_SIZE
- GL_TEXTURE_COORD_ARRAY_TYPE
- GL_TEXTURE_COORD_ARRAY_STRIDE
- GL_EDGE_FLAG_ARRAY_STRIDE
- GL_VERTEX_ARRAY_POINTER
- GL_NORMAL_ARRAY_POINTER
- GL_COLOR_ARRAY_POINTER
- GL_INDEX_ARRAY_POINTER
- GL_TEXTURE_COORD_ARRAY_POINTER
- GL_EDGE_FLAG_ARRAY_POINTER
- GL_COLOR_INDEX1_EXT
- GL_COLOR_INDEX2_EXT
- GL_COLOR_INDEX4_EXT
- GL_COLOR_INDEX8_EXT
- GL_COLOR_INDEX12_EXT
- GL_COLOR_INDEX16_EXT
- GL_EVAL_BIT
- GL_LIST_BIT
- GL_TEXTURE_BIT

- GL_SCISSOR_BIT
- GL_ALL_ATTRIB_BITS
- GL_CLIENT_ALL_ATTRIB_BITS
- GL_SMOOTH_POINT_SIZE_RANGE
- GL_SMOOTH_POINT_SIZE_GRANULARITY
- GL_SMOOTH_LINE_WIDTH_RANGE
- GL_SMOOTH_LINE_WIDTH_GRANULARITY
- GL_UNSIGNED_BYTE_3_3_2
- GL_UNSIGNED_SHORT_4_4_4_4
- GL_UNSIGNED_SHORT_5_5_5_1
- GL_UNSIGNED_INT_8_8_8_8
- GL_UNSIGNED_INT_10_10_10_2
- GL_RESCALE_NORMAL
- GL_TEXTURE_BINDING_3D
- GL_PACK_SKIP_IMAGES
- GL_PACK_IMAGE_HEIGHT
- GL_UNPACK_SKIP_IMAGES
- GL_UNPACK_IMAGE_HEIGHT
- GL_TEXTURE_3D
- GL_PROXY_TEXTURE_3D
- GL_TEXTURE_DEPTH
- GL_TEXTURE_WRAP_R
- GL_MAX_3D_TEXTURE_SIZE
- GL_BGR
- GL_BGRA
- GL_MAX_ELEMENTS_VERTICES
- GL_MAX_ELEMENTS_INDICES
- GL_CLAMP_TO_EDGE
- GL_TEXTURE_MIN_LOD
- GL_TEXTURE_MAX_LOD
- GL_TEXTURE_BASE_LEVEL
- GL_TEXTURE_MAX_LEVEL
- GL_LIGHT_MODEL_COLOR_CONTROL
- GL_SINGLE_COLOR
- GL_SEPARATE_SPECULAR_COLOR
- GL_UNSIGNED_BYTE_2_3_3_REV

- GL_UNSIGNED_SHORT_5_6_5
- GL_UNSIGNED_SHORT_5_6_5_REV
- GL_UNSIGNED_SHORT_4_4_4_REV
- GL_UNSIGNED_SHORT_1_5_5_5_REV
- GL_UNSIGNED_INT_8_8_8_8_REV
- GL_ALIASED_POINT_SIZE_RANGE
- GL_ALIASED_LINE_WIDTH_RANGE
- GL_MULTISAMPLE
- GL_SAMPLE_ALPHA_TO_COVERAGE
- GL_SAMPLE_ALPHA_TO_ONE
- GL_SAMPLE_COVERAGE
- GL_SAMPLE_BUFFERS
- GL_SAMPLES
- GL_SAMPLE_COVERAGE_VALUE
- GL_SAMPLE_COVERAGE_INVERT
- GL_CLAMP_TO_BORDER
- GL_TEXTURE0
- GL_TEXTURE1
- GL_TEXTURE2
- GL_TEXTURE3
- GL_TEXTURE4
- GL_TEXTURE5
- GL_TEXTURE6
- GL_TEXTURE7
- GL_TEXTURE8
- GL_TEXTURE9
- GL_TEXTURE10
- GL_TEXTURE11
- GL_TEXTURE12
- GL_TEXTURE13
- GL_TEXTURE14
- GL_TEXTURE15
- GL_TEXTURE16
- GL_TEXTURE17
- GL_TEXTURE18
- GL_TEXTURE19

- GL_TEXTURE20
- GL_TEXTURE21
- GL_TEXTURE22
- GL_TEXTURE23
- GL_TEXTURE24
- GL_TEXTURE25
- GL_TEXTURE26
- GL_TEXTURE27
- GL_TEXTURE28
- GL_TEXTURE29
- GL_TEXTURE30
- GL_TEXTURE31
- GL_ACTIVE_TEXTURE
- GL_CLIENT_ACTIVE_TEXTURE
- GL_MAX_TEXTURE_UNITS
- GL_TRANSPOSE_MODELVIEW_MATRIX
- GL_TRANSPOSE_PROJECTION_MATRIX
- GL_TRANSPOSE_TEXTURE_MATRIX
- GL_TRANSPOSE_COLOR_MATRIX
- GL_SUBTRACT
- GL_COMPRESSED_ALPHA
- GL_COMPRESSED_LUMINANCE
- GL_COMPRESSED_LUMINANCE_ALPHA
- GL_COMPRESSED_INTENSITY
- GL_COMPRESSED_RGB
- GL_COMPRESSED_RGBA
- GL_TEXTURE_COMPRESSION_HINT
- GL_NORMAL_MAP
- GL_REFLECTION_MAP
- GL_TEXTURE_CUBE_MAP
- GL_TEXTURE_BINDING_CUBE_MAP
- GL_TEXTURE_CUBE_MAP_POSITIVE_X
- GL_TEXTURE_CUBE_MAP_NEGATIVE_X
- GL_TEXTURE_CUBE_MAP_POSITIVE_Y
- GL_TEXTURE_CUBE_MAP_NEGATIVE_Y
- GL_TEXTURE_CUBE_MAP_POSITIVE_Z

- GL_TEXTURE_CUBE_MAP_NEGATIVE_Z
- GL_PROXY_TEXTURE_CUBE_MAP
- GL_MAX_CUBE_MAP_TEXTURE_SIZE
- GL_COMBINE
- GL_COMBINE_RGB
- GL_COMBINE_ALPHA
- GL_RGB_SCALE
- GL_ADD_SIGNED
- GL_INTERPOLATE
- GL_CONSTANT
- GL_PRIMARY_COLOR
- GL_PREVIOUS
- GL_SOURCE0_RGB
- GL_SOURCE1_RGB
- GL_SOURCE2_RGB
- GL_SOURCE0_ALPHA
- GL_SOURCE1_ALPHA
- GL_SOURCE2_ALPHA
- GL_OPERAND0_RGB
- GL_OPERAND1_RGB
- GL_OPERAND2_RGB
- GL_OPERAND0_ALPHA
- GL_OPERAND1_ALPHA
- GL_OPERAND2_ALPHA
- GL_TEXTURE_COMPRESSED_IMAGE_SIZE
- GL_TEXTURE_COMPRESSED
- GL_NUM_COMPRESSED_TEXTURE_FORMATS
- GL_COMPRESSED_TEXTURE_FORMATS
- GL_DOT3_RGB
- GL_DOT3_RGBA
- GL_MULTISAMPLE_BIT
- GL_BLEND_DST_RGB
- GL_BLEND_SRC_RGB
- GL_BLEND_DST_ALPHA
- GL_BLEND_SRC_ALPHA
- GL_POINT_SIZE_MIN

- GL_POINT_SIZE_MAX
- GL_POINT_FADE_THRESHOLD_SIZE
- GL_POINT_DISTANCE_ATTENUATION
- GL_GENERATE_MIPMAP
- GL_GENERATE_MIPMAP_HINT
- GL_DEPTH_COMPONENT16
- GL_DEPTH_COMPONENT24
- GL_DEPTH_COMPONENT32
- GL_MIRRORED_REPEAT
- GL_FOG_COORDINATE_SOURCE
- GL_FOG_COORDINATE
- GL_FRAGMENT_DEPTH
- GL_CURRENT_FOG_COORDINATE
- GL_FOG_COORDINATE_ARRAY_TYPE
- GL_FOG_COORDINATE_ARRAY_STRIDE
- GL_FOG_COORDINATE_ARRAY_POINTER
- GL_FOG_COORDINATE_ARRAY
- GL_COLOR_SUM
- GL_CURRENT_SECONDARY_COLOR
- GL_SECONDARY_COLOR_ARRAY_SIZE
- GL_SECONDARY_COLOR_ARRAY_TYPE
- GL_SECONDARY_COLOR_ARRAY_STRIDE
- GL_SECONDARY_COLOR_ARRAY_POINTER
- GL_SECONDARY_COLOR_ARRAY
- GL_MAX_TEXTURE_LOD_BIAS
- GL_TEXTURE_FILTER_CONTROL
- GL_TEXTURE_LOD_BIAS
- GL_INCR_WRAP
- GL_DECR_WRAP
- GL_TEXTURE_DEPTH_SIZE
- GL_DEPTH_TEXTURE_MODE
- GL_TEXTURE_COMPARE_MODE
- GL_TEXTURE_COMPARE_FUNC
- GL_COMPARE_R_TO_TEXTURE
- GL_CURRENT_FOG_COORD
- GL_FOG_COORD

- GL_FOG_COORD_ARRAY
- GL_FOG_COORD_ARRAY_BUFFER_BINDING
- GL_FOG_COORD_ARRAY_POINTER
- GL_FOG_COORD_ARRAY_STRIDE
- GL_FOG_COORD_ARRAY_TYPE
- GL_FOG_COORD_SRC
- GL_SRC0_ALPHA
- GL_SRC0_RGB
- GL_SRC1_ALPHA
- GL_SRC1_RGB
- GL_SRC2_ALPHA
- GL_SRC2_RGB
- GL_BUFFER_SIZE
- GL_BUFFER_USAGE
- GL_QUERY_COUNTER_BITS
- GL_CURRENT_QUERY
- GL_QUERY_RESULT
- GL_QUERY_RESULT_AVAILABLE
- GL_ARRAY_BUFFER
- GL_ELEMENT_ARRAY_BUFFER
- GL_ARRAY_BUFFER_BINDING
- GL_ELEMENT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ARRAY_BUFFER_BINDING
- GL_NORMAL_ARRAY_BUFFER_BINDING
- GL_COLOR_ARRAY_BUFFER_BINDING
- GL_INDEX_ARRAY_BUFFER_BINDING
- GL_TEXTURE_COORD_ARRAY_BUFFER_BINDING
- GL_EDGE_FLAG_ARRAY_BUFFER_BINDING
- GL_SECONDARY_COLOR_ARRAY_BUFFER_BINDING
- GL_FOG_COORDINATE_ARRAY_BUFFER_BINDING
- GL_WEIGHT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ATTRIB_ARRAY_BUFFER_BINDING
- GL_READ_ONLY
- GL_WRITE_ONLY
- GL_READ_WRITE
- GL_BUFFER_ACCESS

- GL_BUFFER_MAPPED
- GL_BUFFER_MAP_POINTER
- GL_STREAM_DRAW
- GL_STREAM_READ
- GL_STREAM_COPY
- GL_STATIC_DRAW
- GL_STATIC_READ
- GL_STATIC_COPY
- GL_DYNAMIC_DRAW
- GL_DYNAMIC_READ
- GL_DYNAMIC_COPY
- GL_SAMPLES_PASSED
- GL_BLEND_EQUATION_RGB
- GL_VERTEX_ATTRIB_ARRAY_ENABLED
- GL_VERTEX_ATTRIB_ARRAY_SIZE
- GL_VERTEX_ATTRIB_ARRAY_STRIDE
- GL_VERTEX_ATTRIB_ARRAY_TYPE
- GL_CURRENT_VERTEX_ATTRIB
- GL_VERTEX_PROGRAM_POINT_SIZE
- GL_VERTEX_PROGRAM_TWO_SIDE
- GL_VERTEX_ATTRIB_ARRAY_POINTER
- GL_STENCIL_BACK_FUNC
- GL_STENCIL_BACK_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_PASS
- GL_MAX_DRAW_BUFFERS
- GL_DRAW_BUFFER0
- GL_DRAW_BUFFER1
- GL_DRAW_BUFFER2
- GL_DRAW_BUFFER3
- GL_DRAW_BUFFER4
- GL_DRAW_BUFFER5
- GL_DRAW_BUFFER6
- GL_DRAW_BUFFER7
- GL_DRAW_BUFFER8
- GL_DRAW_BUFFER9

- GL_DRAW_BUFFER10
- GL_DRAW_BUFFER11
- GL_DRAW_BUFFER12
- GL_DRAW_BUFFER13
- GL_DRAW_BUFFER14
- GL_DRAW_BUFFER15
- GL_BLEND_EQUATION_ALPHA
- GL_POINT_SPRITE
- GL_COORD_REPLACE
- GL_MAX_VERTEX_ATTRIBS
- GL_VERTEX_ATTRIB_ARRAY_NORMALIZED
- GL_MAX_TEXTURE_COORDS
- GL_MAX_TEXTURE_IMAGE_UNITS
- GL_FRAGMENT_SHADER
- GL_VERTEX_SHADER
- GL_MAX_FRAGMENT_UNIFORM_COMPONENTS
- GL_MAX_VERTEX_UNIFORM_COMPONENTS
- GL_MAX_VARYING_FLOATS
- GL_MAX_VERTEX_TEXTURE_IMAGE_UNITS
- GL_MAX_COMBINED_TEXTURE_IMAGE_UNITS
- GL_SHADER_TYPE
- GL_FLOAT_VEC2
- GL_FLOAT_VEC3
- GL_FLOAT_VEC4
- GL_INT_VEC2
- GL_INT_VEC3
- GL_INT_VEC4
- GL_BOOL
- GL_BOOL_VEC2
- GL_BOOL_VEC3
- GL_BOOL_VEC4
- GL_FLOAT_MAT2
- GL_FLOAT_MAT3
- GL_FLOAT_MAT4
- GL_SAMPLER_1D
- GL_SAMPLER_2D

- GL_SAMPLER_3D
- GL_SAMPLER_CUBE
- GL_SAMPLER_1D_SHADOW
- GL_SAMPLER_2D_SHADOW
- GL_DELETE_STATUS
- GL_COMPILE_STATUS
- GL_LINK_STATUS
- GL_VALIDATE_STATUS
- GL_INFO_LOG_LENGTH
- GL_ATTACHED_SHADERS
- GL_ACTIVE_UNIFORMS
- GL_ACTIVE_UNIFORM_MAX_LENGTH
- GL_SHADER_SOURCE_LENGTH
- GL_ACTIVE_ATTRIBUTES
- GL_ACTIVE_ATTRIBUTE_MAX_LENGTH
- GL_FRAGMENT_SHADER_DERIVATIVE_HINT
- GL_SHADING_LANGUAGE_VERSION
- GL_CURRENT_PROGRAM
- GL_POINT_SPRITE_COORD_ORIGIN
- GL_LOWER_LEFT
- GL_UPPER_LEFT
- GL_STENCIL_BACK_REF
- GL_STENCIL_BACK_VALUE_MASK
- GL_STENCIL_BACK_WRITEMASK
- GL_CURRENT_RASTER_SECONDARY_COLOR
- GL_PIXEL_PACK_BUFFER
- GL_PIXEL_UNPACK_BUFFER
- GL_PIXEL_PACK_BUFFER_BINDING
- GL_PIXEL_UNPACK_BUFFER_BINDING
- GL_FLOAT_MAT2x3
- GL_FLOAT_MAT2x4
- GL_FLOAT_MAT3x2
- GL_FLOAT_MAT3x4
- GL_FLOAT_MAT4x2
- GL_FLOAT_MAT4x3
- GL_SRGB

- GL_SRGB8
- GL_SRGB_ALPHA
- GL_SRGB8_ALPHA8
- GL_SLUMINANCE_ALPHA
- GL_SLUMINANCE8_ALPHA8
- GL_SLUMINANCE
- GL_SLUMINANCE8
- GL_COMPRESSED_SRGB
- GL_COMPRESSED_SRGB_ALPHA
- GL_COMPRESSED_SLUMINANCE
- GL_COMPRESSED_SLUMINANCE_ALPHA
- GL_CLIP_DISTANCE0
- GL_CLIP_DISTANCE1
- GL_CLIP_DISTANCE2
- GL_CLIP_DISTANCE3
- GL_CLIP_DISTANCE4
- GL_CLIP_DISTANCE5
- GL_COMPARE_REF_TO_TEXTURE
- GL_MAX_CLIP_DISTANCES
- GL_MAX_VARYING_COMPONENTS
- GL_CONTEXT_FLAG_FORWARD_COMPATIBLE_BIT
- GL_MAJOR_VERSION
- GL_MINOR_VERSION
- GL_NUM_EXTENSIONS
- GL_CONTEXT_FLAGS
- GL_DEPTH_BUFFER
- GL_STENCIL_BUFFER
- GL_RGBA32F
- GL_RGB32F
- GL_RGBA16F
- GL_RGB16F
- GL_VERTEX_ATTRIB_ARRAY_INTEGER
- GL_MAX_ARRAY_TEXTURE_LAYERS
- GL_MIN_PROGRAM_TEXEL_OFFSET
- GL_MAX_PROGRAM_TEXEL_OFFSET
- GL_CLAMP_VERTEX_COLOR

- GL_CLAMP_FRAGMENT_COLOR
- GL_CLAMP_READ_COLOR
- GL_FIXED_ONLY
- GL_TEXTURE_RED_TYPE
- GL_TEXTURE_GREEN_TYPE
- GL_TEXTURE_BLUE_TYPE
- GL_TEXTURE_ALPHA_TYPE
- GL_TEXTURE_LUMINANCE_TYPE
- GL_TEXTURE_INTENSITY_TYPE
- GL_TEXTURE_DEPTH_TYPE
- GL_TEXTURE_1D_ARRAY
- GL_PROXY_TEXTURE_1D_ARRAY
- GL_TEXTURE_2D_ARRAY
- GL_PROXY_TEXTURE_2D_ARRAY
- GL_TEXTURE_BINDING_1D_ARRAY
- GL_TEXTURE_BINDING_2D_ARRAY
- GL_R11F_G11F_B10F
- GL_UNSIGNED_INT_10F_11F_11F_REV
- GL_RGB9_E5
- GL_UNSIGNED_INT_5_9_9_9_REV
- GL_TEXTURE_SHARED_SIZE
- GL_TRANSFORM_FEEDBACK_VARYING_MAX_LENGTH
- GL_TRANSFORM_FEEDBACK_BUFFER_MODE
- GL_MAX_TRANSFORM_FEEDBACK_SEPARATE_COMPONENTS
- GL_TRANSFORM_FEEDBACK_VARYINGS
- GL_TRANSFORM_FEEDBACK_BUFFER_START
- GL_TRANSFORM_FEEDBACK_BUFFER_SIZE
- GL_PRIMITIVES_GENERATED
- GL_TRANSFORM_FEEDBACK_PRIMITIVES_WRITTEN
- GL_RASTERIZER_DISCARD
- GL_MAX_TRANSFORM_FEEDBACK_INTERLEAVED_COMPONENTS
- GL_MAX_TRANSFORM_FEEDBACK_SEPARATE_ATTRIBS
- GL_INTERLEAVED_ATTRIBS
- GL_SEPARATE_ATTRIBS
- GL_TRANSFORM_FEEDBACK_BUFFER
- GL_TRANSFORM_FEEDBACK_BUFFER_BINDING

- GL_RGBA32UI
- GL_RGB32UI
- GL_RGBA16UI
- GL_RGB16UI
- GL_RGBA8UI
- GL_RGB8UI
- GL_RGBA32I
- GL_RGB32I
- GL_RGBA16I
- GL_RGB16I
- GL_RGBA8I
- GL_RGB8I
- GL_RED_INTEGER
- GL_GREEN_INTEGER
- GL_BLUE_INTEGER
- GL_ALPHA_INTEGER
- GL_RGB_INTEGER
- GL_RGBA_INTEGER
- GL_BGR_INTEGER
- GL_BGRA_INTEGER
- GL_SAMPLER_1D_ARRAY
- GL_SAMPLER_2D_ARRAY
- GL_SAMPLER_1D_ARRAY_SHADOW
- GL_SAMPLER_2D_ARRAY_SHADOW
- GL_SAMPLER_CUBE_SHADOW
- GL_UNSIGNED_INT_VEC2
- GL_UNSIGNED_INT_VEC3
- GL_UNSIGNED_INT_VEC4
- GL_INT_SAMPLER_1D
- GL_INT_SAMPLER_2D
- GL_INT_SAMPLER_3D
- GL_INT_SAMPLER_CUBE
- GL_INT_SAMPLER_1D_ARRAY
- GL_INT_SAMPLER_2D_ARRAY
- GL_UNSIGNED_INT_SAMPLER_1D
- GL_UNSIGNED_INT_SAMPLER_2D

- GL_UNSIGNED_INT_SAMPLER_3D
- GL_UNSIGNED_INT_SAMPLER_CUBE
- GL_UNSIGNED_INT_SAMPLER_1D_ARRAY
- GL_UNSIGNED_INT_SAMPLER_2D_ARRAY
- GL_QUERY_WAIT
- GL_QUERY_NO_WAIT
- GL_QUERY_BY_REGION_WAIT
- GL_QUERY_BY_REGION_NO_WAIT
- void glAccum(GLenum op, GLfloat value)
- void glActiveTexture(GLenum texture)
- void glAlphaFunc(GLenum func, GLclampf ref)
- GLboolean glAreTexturesResident(GLsizei n, const GLuint * textures, GLboolean * residences)
- void glArrayElement(GLint i)
- void glAttachShader(GLuint program, GLuint shader)
- void glBegin(GLenum mode)
- void glBeginQuery(GLenum target, GLuint id)
- void glBindAttribLocation(GLuint program, GLuint index, const GLchar *name)
- void glBindBuffer(GLenum target, GLuint buffer)
- void glBindTexture(GLenum target, GLuint texture)
- void glBitmap(GLsizei width, GLsizei height, GLfloat xorig, GLfloat yorig, GLfloat xmove, GLfloat ymove, const GLubyte * bitmap)
- void glBlendColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)
- void glBlendEquation(GLenum mode)
- void glBlendEquationSeparate(GLenum modeRGB, GLenum modeAlpha)
- void glBlendFunc(GLenum sfactor, GLenum dfactor)
- void glBlendFuncSeparate(GLenum srcRGB, GLenum dstRGB, GLenum srcAlpha, GLenum dstAlpha)
- void glBufferData(GLenum target, GLsizeiptr size, const GLvoid * data, GLenum usage)
- void glBufferSubData(GLenum target, GLintptr offset, GLsizeiptr size, const GLvoid * data)
- void glCallList(GLuint list)
- void glCallLists(GLsizei n, GLenum type, const GLvoid * lists)
- void glClear(GLbitfield mask)
- void glClearAccum(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glClearColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)
- void glClearDepth(GLclampd depth)
- void glClearIndex(GLfloat c)
- void glClearStencil(GLint s)

- `void glClientActiveTexture(GLenum texture)`
- `void glClipPlane(GLenum plane,const GLdouble * equation)`
- `void glColor3b(GLbyte red,GLbyte green,GLbyte blue)`
- `void glColor3s(GLshort red,GLshort green,GLshort blue)`
- `void glColor3i(GLint red,GLint green,GLint blue)`
- `void glColor3f(GLfloat red,GLfloat green,GLfloat blue)`
- `void glColor3d(GLdouble red,GLdouble green,GLdouble blue)`
- `void glColor3ub(GLubyte red,GLubyte green,GLubyte blue)`
- `void glColor3us(GLushort red,GLushort green,GLushort blue)`
- `void glColor3ui(GLuint red,GLuint green,GLuint blue)`
- `void glColor4b(GLbyte red,GLbyte green,GLbyte blue,GLbyte alpha)`
- `void glColor4s(GLshort red,GLshort green,GLshort blue,GLshort alpha)`
- `void glColor4i(GLint red,GLint green,GLint blue,GLint alpha)`
- `void glColor4f(GLfloat red,GLfloat green,GLfloat blue,GLfloat alpha)`
- `void glColor4d(GLdouble red,GLdouble green,GLdouble blue,GLdouble alpha)`
- `void glColor4ub(GLubyte red,GLubyte green,GLubyte blue,GLubyte alpha)`
- `void glColor4us(GLushort red,GLushort green,GLushort blue,GLushort alpha)`
- `void glColor4ui(GLuint red,GLuint green,GLuint blue,GLuint alpha)`
- `void glColor3bv(const GLbyte * v)`
- `void glColor3sv(const GLshort * v)`
- `void glColor3iv(const GLint * v)`
- `void glColor3fv(const GLfloat * v)`
- `void glColor3dv(const GLdouble * v)`
- `void glColor3ubv(const GLubyte * v)`
- `void glColor3usv(const GLushort * v)`
- `void glColor3uiv(const GLuint * v)`
- `void glColor4bv(const GLbyte * v)`
- `void glColor4sv(const GLshort * v)`
- `void glColor4iv(const GLint * v)`
- `void glColor4fv(const GLfloat * v)`
- `void glColor4dv(const GLdouble * v)`
- `void glColor4ubv(const GLubyte * v)`
- `void glColor4usv(const GLushort * v)`
- `void glColor4uiv(const GLuint * v)`
- `void glColorMask(GLboolean red,GLboolean green,GLboolean blue,GLboolean alpha)`
- `void glColorMaterial(GLenum face,GLenum mode)`

- void glColorPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)
- void glColorSubTable(GLenum target, GLsizei start, GLsizei count, GLenum format, GLenum type, const GLvoid * data)
- void glColorTable(GLenum target, GLenum internalformat, GLsizei width, GLenum format, GLenum type, const GLvoid * data)
- void glColorTableParameterfv(GLenum target, GLenum pname, const GLfloat * params)
- void glColorTableParameteriv(GLenum target, GLenum pname, const GLint * params)
- void glCompileShader(GLuint shader)
- void glCompressedTexImage1D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLint border, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexImage2D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLint border, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexImage3D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLsizei depth, GLint border, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLsizei imageSize, const GLvoid * data)
- void glConvolutionFilter1D(GLenum target, GLenum internalformat, GLsizei width, GLenum format, GLenum type, const GLvoid * data)
- void glConvolutionFilter2D(GLenum target, GLenum internalformat, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)
- void glConvolutionParameterf(GLenum target, GLenum pname, GLfloat params)
- void glConvolutionParameteri(GLenum target, GLenum pname, GLint params)
- void glConvolutionParameterfv(GLenum target, GLenum pname, const GLfloat * params)
- void glConvolutionParameteriv(GLenum target, GLenum pname, const GLint * params)
- void glCopyColorSubTable(GLenum target, GLsizei start, GLint x, GLint y, GLsizei width)
- void glCopyColorTable(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width)
- void glCopyConvolutionFilter1D(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width)
- void glCopyConvolutionFilter2D(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width, GLsizei height)
- void glCopyPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum type)
- void glCopyTexImage1D(GLenum target, GLint level, GLenum internalformat, GLint x, GLint y, GLsizei width, GLint border)
- void glCopyTexImage2D(GLenum target, GLint level, GLenum internalformat, GLint x, GLint y, GLsizei width, GLsizei height, GLint border)
- void glCopyTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLint x, GLint y, GLsizei width)
- void glCopyTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint x, GLint y, GLsizei width, GLsizei height)

- `void glCopyTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLint x, GLint y, GLsizei width, GLsizei height)`
- `GLuint glCreateProgram(void)`
- `GLuint glCreateShader(GLenum shaderType)`
- `void glCullFace(GLenum mode)`
- `void glDeleteBuffers(GLsizei n, const GLuint * buffers)`
- `void glDeleteLists(GLuint list, GLsizei range)`
- `void glDeleteProgram(GLuint program)`
- `void glDeleteQueries(GLsizei n, const GLuint * ids)`
- `void glDeleteShader(GLuint shader)`
- `void glDeleteTextures(GLsizei n, const GLuint * textures)`
- `void glDepthFunc(GLenum func)`
- `void glDepthMask(GLboolean flag)`
- `void glDepthRange(GLclampd nearVal, GLclampd farVal)`
- `void glDetachShader(GLuint program, GLuint shader)`
- `void glEnable(GLenum cap)`
- `void glEnableClientState(GLenum cap)`
- `void glEnableVertexAttribArray(GLuint index)`
- `void glDisableVertexAttribArray(GLuint index)`
- `void glDrawArrays(GLenum mode, GLint first, GLsizei count)`
- `void glDrawBuffer(GLenum mode)`
- `void glDrawBuffers(GLsizei n, const GLenum *bufs)`
- `void glDrawElements(GLenum mode, GLsizei count, GLenum type, const GLvoid * indices)`
- `void glDrawPixels(GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)`
- `void glDrawRangeElements(GLenum mode, GLuint start, GLuint end, GLsizei count, GLenum type, const GLvoid * indices)`
- `void glEdgeFlag(GLboolean flag)`
- `void glEdgeFlagPointer(GLsizei stride, const GLvoid * pointer)`
- `void glEnd(void)`
- `void glEndList(void)`
- `void glEndQuery(GLenum target)`
- `void glEvalCoord1f(GLfloat u)`
- `void glEvalCoord1d(GLdouble u)`
- `void glEvalCoord2f(GLfloat u, GLfloat v)`
- `void glEvalCoord2d(GLdouble u, GLdouble v)`
- `void glEvalMesh1(GLenum mode, GLint i1, GLint i2)`
- `void glEvalPoint1(GLint i)`

- void glEvalPoint2(GLint i, GLint j)
- void glFeedbackBuffer(GLsizei size, GLenum type, GLfloat * buffer)
- void glFinish(void)
- void glFlush(void)
- void glFogf(GLenum pname, GLfloat param)
- void glFogi(GLenum pname, GLint param)
- void glFogfv(GLenum pname, const GLfloat * params)
- void glFogiv(GLenum pname, const GLint * params)
- void glFogCoordd(GLdouble coord)
- void glFogCoordf(GLfloat coord)
- void glFogCoorddv(GLdouble * coord)
- void glFogCoordfv(GLfloat * coord)
- void glFogCoordPointer(GLenum type, GLsizei stride, GLvoid * pointer)
- void glFrontFace(GLenum mode)
- void glFrustum(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble nearVal, GLdouble farVal)
- void glGenBuffers(GLsizei n, GLuint * buffers)
- GLuint glGenLists(GLsizei range)
- void glGenQueries(GLsizei n, GLuint * ids)
- void glGenTextures(GLsizei n, GLuint * textures)
- void glGetBooleany(GLenum pname, GLboolean * params)
- void glGetDoublev(GLenum pname, GLdouble * params)
- void glGetFloatv(GLenum pname, GLfloat * params)
- void glGetIntegerv(GLenum pname, GLint * params)
- void glGetActiveAttrib(GLuint program, GLuint index, GLsizei bufSize, GLsizei *length, GLint *size, GLenum *type, GLchar *name)
- void glGetActiveUniform(GLuint program, GLuint index, GLsizei bufSize, GLsizei *length, GLint *size, GLenum *type, GLchar *name)
- void glGetAttachedShaders(GLuint program, GLsizei maxCount, GLsizei *count, GLuint *shaders)
- GLint glGetAttribLocation(GLuint program, const GLchar *name)
- void glGetBufferParameteriv(GLenum target, GLenum value, GLint * data)
- void glGetBufferPointerv(GLenum target, GLenum pname, GLvoid ** params)
- void glGetBufferSubData(GLenum target, GLintptr offset, GLsizeiptr size, GLvoid * data)
- void glGetClipPlane(GLenum plane, GLdouble * equation)
- void glGetColorTable(GLenum target, GLenum format, GLenum type, GLvoid * table)
- void glGetColorTableParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetColorTableParameteriv(GLenum target, GLenum pname, GLint * params)

- void glGetCompressedTexImage(GLenum target, GLint lod, GLvoid * img)
- void glGetConvolutionFilter(GLenum target, GLenum format, GLenum type, GLvoid * image)
- void glGetConvolutionParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetConvolutionParameteriv(GLenum target, GLenum pname, GLint * params)
- GLenum glGetError(void)
- void glGetHistogram(GLenum target, GLboolean reset, GLenum format, GLenum type, GLvoid * values)
- void glGetHistogramParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetHistogramParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetLightfv(GLenum light, GLenum pname, GLfloat * params)
- void glGetLightiv(GLenum light, GLenum pname, GLint * params)
- void glGetMapdv(GLenum target, GLenum query, GLdouble * v)
- void glGetMapfv(GLenum target, GLenum query, GLfloat * v)
- void glGetMapiv(GLenum target, GLenum query, GLint * v)
- void glGetMaterialfv(GLenum face, GLenum pname, GLfloat * params)
- void glGetMaterialiv(GLenum face, GLenum pname, GLint * params)
- void glGetMinmax(GLenum target, GLboolean reset, GLenum format, GLenum types, GLvoid * values)
- void glGetMinmaxParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetMinmaxParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetPixelMapfv(GLenum map, GLfloat * data)
- void glGetPixelMapuiv(GLenum map, GLuint * data)
- void glGetPixelMapusv(GLenum map, GLushort * data)
- void glGetPointerv(GLenum pname, GLvoid ** params)
- void glGetPolygonStipple(GLubyte * pattern)
- void glGetProgramiv(GLuint program, GLenum pname, GLint *params)
- void glGetProgramInfoLog(GLuint program, GLsizei maxLength, GLsizei *length, GLchar *infoLog)
- void glGetQueryObjectiv(GLuint id, GLenum pname, GLint * params)
- void glGetQueryObjectuiv(GLuint id, GLenum pname, GLuint * params)
- void glGetQueryiv(GLenum target, GLenum pname, GLint * params)
- void glGetSeparableFilter(GLenum target, GLenum format, GLenum type, GLvoid * row, GLvoid * column, GLvoid * span)
- void glGetShaderiv(GLuint shader, GLenum pname, GLint *params)
- void glGetShaderInfoLog(GLuint shader, GLsizei maxLength, GLsizei *length, GLchar *infoLog)
- void glGetShaderSource(GLuint shader, GLsizei bufSize, GLsizei *length, GLchar *source)
- const GLubyte* getString(GLenum name)
- void glGetTexEnvfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetTexEnviv(GLenum target, GLenum pname, GLint * params)

- void glGetTexGendv(GLenum coord, GLenum pname, GLdouble * params)
- void glGetTexGenfv(GLenum coord, GLenum pname, GLfloat * params)
- void glGetTexGeniv(GLenum coord, GLenum pname, GLint * params)
- void glGetTexImage(GLenum target, GLint level, GLenum format, GLenum type, GLvoid * img)
- void glGetTexLevelParameterfv(GLenum target, GLint level, GLenum pname, GLfloat * params)
- void glGetTexLevelParameteriv(GLenum target, GLint level, GLenum pname, GLint * params)
- void glGetTexParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetTexParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetUniformfv(GLuint program, GLint location, GLfloat *params)
- void glGetUniformiv(GLuint program, GLint location, GLint *params)
- GLint glGetUniformLocation(GLuint program, const GLchar *name)
- void glGetVertexAttribdv(GLuint index, GLenum pname, GLdouble *params)
- void glGetVertexAttribfv(GLuint index, GLenum pname, GLfloat *params)
- void glGetVertexAttribiv(GLuint index, GLenum pname, GLint *params)
- void glGetVertexAttribPointerv(GLuint index, GLenum pname, GLvoid **pointer)
- void glHint(GLenum target, GLenum mode)
- void glHistogram(GLenum target, GLsizei width, GLenum internalformat, GLboolean sink)
- void glIndexs(GLshort c)
- void glIndexi(GLint c)
- void glIndexf(GLfloat c)
- void glIndexd(GLdouble c)
- void glIndexub(GLubyte c)
- void glIndexsv(const GLshort * c)
- void glIndexiv(const GLint * c)
- void glIndexfv(const GLfloat * c)
- void glIndexdv(const GLdouble * c)
- void glIndexubv(const GLubyte * c)
- void glIndexMask(GLuint mask)
- void glIndexPointer(GLenum type, GLsizei stride, const GLvoid * pointer)
- void glInitNames(void)
- void glInterleavedArrays(GLenum format, GLsizei stride, const GLvoid * pointer)
- GLboolean glIsBuffer(GLuint buffer)
- GLboolean glIsEnabled(GLenum cap)
- GLboolean glIsList(GLuint list)
- GLboolean glIsProgram(GLuint program)
- GLboolean glIsQuery(GLuint id)

- GLboolean glIsShader(GLuint shader)
- GLboolean glIsTexture(GLuint texture)
- void glLightf(GLenum light, GLenum pname, GLfloat param)
- void glLighti(GLenum light, GLenum pname, GLint param)
- void glLightfv(GLenum light, GLenum pname, const GLfloat * params)
- void glLightiv(GLenum light, GLenum pname, const GLint * params)
- void glLightModelf(GLenum pname, GLfloat param)
- void glLightModeli(GLenum pname, GLint param)
- void glLightModelfv(GLenum pname, const GLfloat * params)
- void glLightModeliv(GLenum pname, const GLint * params)
- void glLineStipple(GLint factor, GLushort pattern)
- void glLineWidth(GLfloat width)
- void glLinkProgram(GLuint program)
- void glListBase(GLuint base)
- void glLoadIdentity(void)
- void glLoadMatrixd(const GLdouble * m)
- void glLoadMatrixf(const GLfloat * m)
- void glLoadName(GLuint name)
- void glLoadTransposeMatrixd(const GLdouble * m)
- void glLoadTransposeMatrixf(const GLfloat * m)
- void glLogicOp(GLenum opcode)
- void glMap1f(GLenum target, GLfloat u1, GLfloat u2, GLint stride, GLint order, const GLfloat * points)
- void glMap1d(GLenum target, GLdouble u1, GLdouble u2, GLint stride, GLint order, const GLdouble * points)
- void glMap2f(GLenum target, GLfloat u1, GLfloat u2, GLint ustride, GLint uorder, GLfloat v1, GLfloat v2, GLint vstride, GLint vorder, const GLfloat * points)
- void glMap2d(GLenum target, GLdouble u1, GLdouble u2, GLint ustride, GLint uorder, GLdouble v1, GLdouble v2, GLint vstride, GLint vorder, const GLdouble * points)
- void * glMapBuffer(GLenum target, GLenum access)
- void glMapGrid1d(GLint un, GLdouble u1, GLdouble u2)
- void glMapGrid1f(GLint un, GLfloat u1, GLfloat u2)
- void glMapGrid2d(GLint un, GLdouble u1, GLdouble u2, GLint vn, GLdouble v1, GLdouble v2)
- void glMapGrid2f(GLint un, GLfloat u1, GLfloat u2, GLint vn, GLfloat v1, GLfloat v2)
- void glMaterialf(GLenum face, GLenum pname, GLfloat param)
- void glMateriali(GLenum face, GLenum pname, GLint param)
- void glMatrixMode(GLenum mode)
- void glMinmax(GLenum target, GLenum internalformat, GLboolean sink)
- void glMultMatrixd(const GLdouble * m)

- `void glMultMatrixf(const GLfloat * m)`
- `void glMultTransposeMatrixd(const GLdouble * m)`
- `void glMultTransposeMatrixf(const GLfloat * m)`
- `void glMultiDrawArrays(GLenum mode, GLint * first, GLsizei * count, GLsizei primcount)`
- `void glMultiDrawElements(GLenum mode, const GLsizei * count, GLenum type, const GLvoid ** indices, GLsizei primcount)`
- `void glMultiTexCoord1s(GLenum target, GLshort s)`
- `void glMultiTexCoord1i(GLenum target, GLint s)`
- `void glMultiTexCoord1f(GLenum target, GLfloat s)`
- `void glMultiTexCoord1d(GLenum target, GLdouble s)`
- `void glMultiTexCoord2s(GLenum target, GLshort s, GLshort t)`
- `void glMultiTexCoord2i(GLenum target, GLint s, GLint t)`
- `void glMultiTexCoord2f(GLenum target, GLfloat s, GLfloat t)`
- `void glMultiTexCoord2d(GLenum target, GLdouble s, GLdouble t)`
- `void glMultiTexCoord3s(GLenum target, GLshort s, GLshort t, GLshort r)`
- `void glMultiTexCoord3i(GLenum target, GLint s, GLint t, GLint r)`
- `void glMultiTexCoord3f(GLenum target, GLfloat s, GLfloat t, GLfloat r)`
- `void glMultiTexCoord3d(GLenum target, GLdouble s, GLdouble t, GLdouble r)`
- `void glMultiTexCoord4s(GLenum target, GLshort s, GLshort t, GLshort r, GLshort q)`
- `void glMultiTexCoord4i(GLenum target, GLint s, GLint t, GLint r, GLint q)`
- `void glMultiTexCoord4f(GLenum target, GLfloat s, GLfloat t, GLfloat r, GLfloat q)`
- `void glMultiTexCoord4d(GLenum target, GLdouble s, GLdouble t, GLdouble r, GLdouble q)`
- `void glMultiTexCoord1sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord1iv(GLenum target, const GLint * v)`
- `void glMultiTexCoord1fv(GLenum target, const GLfloat * v)`
- `void glMultiTexCoord1dv(GLenum target, const GLdouble * v)`
- `void glMultiTexCoord2sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord2iv(GLenum target, const GLint * v)`
- `void glMultiTexCoord2fv(GLenum target, const GLfloat * v)`
- `void glMultiTexCoord2dv(GLenum target, const GLdouble * v)`
- `void glMultiTexCoord3sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord3iv(GLenum target, const GLint * v)`
- `void glMultiTexCoord3fv(GLenum target, const GLfloat * v)`
- `void glMultiTexCoord3dv(GLenum target, const GLdouble * v)`
- `void glMultiTexCoord4sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord4iv(GLenum target, const GLint * v)`

- `void glMultiTexCoord4fv(GLenum target,const GLfloat * v)`
- `void glMultiTexCoord4dv(GLenum target,const GLdouble * v)`
- `void glNewList(GLuint list,GLenum mode)`
- `void glNormal3b(GLbyte nx,GLbyte ny,GLbyte nz)`
- `void glNormal3d(GLdouble nx,GLdouble ny,GLdouble nz)`
- `void glNormal3f(GLfloat nx,GLfloat ny,GLfloat nz)`
- `void glNormal3i(GLint nx,GLint ny,GLint nz)`
- `void glNormal3s(GLshort nx,GLshort ny,GLshort nz)`
- `void glNormal3bv(const GLbyte * v)`
- `void glNormal3dv(const GLdouble * v)`
- `void glNormal3fv(const GLfloat * v)`
- `void glNormal3iv(const GLint * v)`
- `void glNormal3sv(const GLshort * v)`
- `void glNormalPointer(GLenum type,GLsizei stride,const GLvoid * pointer)`
- `void glOrtho(GLdouble left,GLdouble right,GLdouble bottom,GLdouble top,GLdouble nearVal,GLdouble farVal)`
- `void glPassThrough(GLfloat token)`
- `void glPixelMapfv(GLenum map,GLsizei mapsize,const GLfloat * values)`
- `void glPixelMapuiv(GLenum map,GLsizei mapsize,const GLuint * values)`
- `void glPixelMapusv(GLenum map,GLsizei mapsize,const GLushort * values)`
- `void glPixelStoref(GLenum pname,GLfloat param)`
- `void glPixelStorei(GLenum pname,GLint param)`
- `void glPixelTransferf(GLenum pname,GLfloat param)`
- `void glPixelTransferi(GLenum pname,GLint param)`
- `void glPixelZoom(GLfloat xfactor,GLfloat yfactor)`
- `void glPointParameterf(GLenum pname,GLfloat param)`
- `void glPointParameteri(GLenum pname,GLint param)`
- `void glPointSize(GLfloat size)`
- `void glPolygonMode(GLenum face,GLenum mode)`
- `void glPolygonOffset(GLfloat factor,GLfloat units)`
- `void glPolygonStipple(const GLubyte * pattern)`
- `void glPushAttrib(GLbitfield mask)`
- `void glPushClientAttrib(GLbitfield mask)`
- `void glPushMatrix(void)`
- `void glPushName(GLuint name)`
- `void glPrioritizeTextures(GLsizei n,const GLuint * textures,const GLclampf * priorities)`

- `void glPopMatrix(void)`
- `void glRasterPos2s(GLshort x, GLshort y)`
- `void glRasterPos2i(GLint x, GLint y)`
- `void glRasterPos2f(GLfloat x, GLfloat y)`
- `void glRasterPos2d(GLdouble x, GLdouble y)`
- `void glRasterPos3s(GLshort x, GLshort y, GLshort z)`
- `void glRasterPos3i(GLint x, GLint y, GLint z)`
- `void glRasterPos3f(GLfloat x, GLfloat y, GLfloat z)`
- `void glRasterPos3d(GLdouble x, GLdouble y, GLdouble z)`
- `void glRasterPos4s(GLshort x, GLshort y, GLshort z, GLshort w)`
- `void glRasterPos4i(GLint x, GLint y, GLint z, GLint w)`
- `void glRasterPos4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)`
- `void glRasterPos4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)`
- `void glReadBuffer(GLenum mode)`
- `void glReadPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum format, GLenum type, GLvoid * data)`
- `void glRectd(GLdouble x1, GLdouble y1, GLdouble x2, GLdouble y2)`
- `void glRectf(GLfloat x1, GLfloat y1, GLfloat x2, GLfloat y2)`
- `void glRecti(GLint x1, GLint y1, GLint x2, GLint y2)`
- `void glRects(GLshort x1, GLshort y1, GLshort x2, GLshort y2)`
- `void glRectdv(const GLdouble * v1, const GLdouble * v2)`
- `void glRectfv(const GLfloat * v1, const GLfloat * v2)`
- `void glRectiv(const GLint * v1, const GLint * v2)`
- `void glRectsv(const GLshort * v1, const GLshort * v2)`
- `GLint glRenderMode(GLenum mode)`
- `void glResetHistogram(GLenum target)`
- `void glRotated(GLdouble angle, GLdouble x, GLdouble y, GLdouble z)`
- `void glRotatef(GLfloat angle, GLfloat x, GLfloat y, GLfloat z)`
- `void glSampleCoverage(GLclampf value, GLboolean invert)`
- `void glScaled(GLdouble x, GLdouble y, GLdouble z)`
- `void glScalef(GLfloat x, GLfloat y, GLfloat z)`
- `void glScissor(GLint x, GLint y, GLsizei width, GLsizei height)`
- `void glSecondaryColor3b(GLbyte red, GLbyte green, GLbyte blue)`
- `void glSecondaryColor3s(GLshort red, GLshort green, GLshort blue)`
- `void glSecondaryColor3i(GLint red, GLint green, GLint blue)`
- `void glSecondaryColor3f(GLfloat red, GLfloat green, GLfloat blue)`
- `void glSecondaryColor3d(GLdouble red, GLdouble green, GLdouble blue)`

- void glSecondaryColor3ub(GLubyte red,GLubyte green,GLubyte blue)
- void glSecondaryColor3us(GLushort red,GLushort green,GLushort blue)
- void glSecondaryColor3ui(GLuint red,GLuint green,GLuint blue)
- void glSecondaryColor3bv(const GLbyte * v)
- void glSecondaryColor3sv(const GLshort * v)
- void glSecondaryColor3iv(const GLint * v)
- void glSecondaryColor3fv(const GLfloat * v)
- void glSecondaryColor3dv(const GLdouble * v)
- void glSecondaryColor3ubv(const GLubyte * v)
- void glSecondaryColor3usv(const GLushort * v)
- void glSecondaryColor3uiv(const GLuint * v)
- void glSecondaryColorPointer(GLint size,GLenum type,GLsizei stride,const GLvoid * pointer)
- void glSelectBuffer(GLsizei size,GLuint * buffer)
- void glSeparableFilter2D(GLenum target,GLenum internalformat,GLsizei width,GLsizei height,GLenum format,GLenum type,const GLvoid * row,const GLvoid * column)
- void glShadeModel(GLenum mode)
- void glShaderSource(GLuint shader,GLsizei count,const GLchar **string,const GLint *length)
- void glStencilFunc(GLenum func,GLint ref,GLuint mask)
- void glStencilFuncSeparate(GLenum face,GLenum func,GLint ref,GLuint mask)
- void glStencilMask(GLuint mask)
- void glStencilMaskSeparate(GLenum face,GLuint mask)
- void glStencilOp(GLenum sfail,GLenum dpfail,GLenum dppass)
- void glStencilOpSeparate(GLenum face,GLenum sfail,GLenum dpfail,GLenum dppass)
- void glTexCoord1s(GLshort s)
- void glTexCoord1i(GLint s)
- void glTexCoord1f(GLfloat s)
- void glTexCoord1d(GLdouble s)
- void glTexCoord2s(GLshort s,GLshort t)
- void glTexCoord2i(GLint s,GLint t)
- void glTexCoord2f(GLfloat s,GLfloat t)
- void glTexCoord2d(GLdouble s,GLdouble t)
- void glTexCoord3s(GLshort s,GLshort t,GLshort r)
- void glTexCoord3i(GLint s,GLint t,GLint r)
- void glTexCoord3f(GLfloat s,GLfloat t,GLfloat r)
- void glTexCoord3d(GLdouble s,GLdouble t,GLdouble r)
- void glTexCoord4s(GLshort s,GLshort t,GLshort r,GLshort q)

- `void glTexCoord4i(GLint s,GLint t,GLint r,GLint q)`
- `void glTexCoord4f(GLfloat s,GLfloat t,GLfloat r,GLfloat q)`
- `void glTexCoord4d(GLdouble s,GLdouble t,GLdouble r,GLdouble q)`
- `void glTexCoord1sv(const GLshort * v)`
- `void glTexCoord1iv(const GLint * v)`
- `void glTexCoord1fv(const GLfloat * v)`
- `void glTexCoord1dv(const GLdouble * v)`
- `void glTexCoord2sv(const GLshort * v)`
- `void glTexCoord2iv(const GLint * v)`
- `void glTexCoord2fv(const GLfloat * v)`
- `void glTexCoord2dv(const GLdouble * v)`
- `void glTexCoord3sv(const GLshort * v)`
- `void glTexCoord3iv(const GLint * v)`
- `void glTexCoord3fv(const GLfloat * v)`
- `void glTexCoord3dv(const GLdouble * v)`
- `void glTexCoord4sv(const GLshort * v)`
- `void glTexCoord4iv(const GLint * v)`
- `void glTexCoord4fv(const GLfloat * v)`
- `void glTexCoord4dv(const GLdouble * v)`
- `void glTexCoordPointer(GLint size,GLenum type,GLsizei stride,const GLvoid * pointer)`
- `void glTexEnvf(GLenum target,GLenum pname,GLfloat param)`
- `void glTexEnvi(GLenum target,GLenum pname,GLint param)`
- `void glTexGeni(GLenum coord,GLenum pname,GLint param)`
- `void glTexGenf(GLenum coord,GLenum pname,GLfloat param)`
- `void glTexGend(GLenum coord,GLenum pname,GLdouble param)`
- `void glTexGeniv(GLenum coord,GLenum pname,const GLint * params)`
- `void glTexGenfv(GLenum coord,GLenum pname,const GLfloat * params)`
- `void glTexGendv(GLenum coord,GLenum pname,const GLdouble * params)`
- `void glTexImage1D(GLenum target,GLint level,GLint internalFormat,GLsizei width,GLint border,GLenum format,GLenum type,const GLvoid * data)`
- `void glTexImage2D(GLenum target,GLint level,GLint internalFormat,GLsizei width,GLsizei height,GLint border,GLenum format,GLenum type,const GLvoid * data)`
- `void glTexImage3D(GLenum target,GLint level,GLint internalFormat,GLsizei width,GLsizei height,GLsizei depth,GLint border,GLenum format,GLenum type,const GLvoid * data)`
- `void glTexParameterf(GLenum target,GLenum pname,GLfloat param)`
- `void glTexParameteri(GLenum target,GLenum pname,GLint param)`
- `void glTexParameterfv(GLenum target,GLenum pname,const GLfloat * params)`

- `void glTexParameteriv(GLenum target, GLenum pname, const GLint * params)`
- `void glTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLenum type, const GLvoid * data)`
- `void glTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)`
- `void glTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLenum type, const GLvoid * data)`
- `void glTranslated(GLdouble x, GLdouble y, GLdouble z)`
- `void glTranslatef(GLfloat x, GLfloat y, GLfloat z)`
- `void glUniform1f(GLint location, GLfloat v0)`
- `void glUniform2f(GLint location, GLfloat v0, GLfloat v1)`
- `void glUniform3f(GLint location, GLfloat v0, GLfloat v1, GLfloat v2)`
- `void glUniform4f(GLint location, GLfloat v0, GLfloat v1, GLfloat v2, GLfloat v3)`
- `void glUniform1i(GLint location, GLint v0)`
- `void glUniform2i(GLint location, GLint v0, GLint v1)`
- `void glUniform3i(GLint location, GLint v0, GLint v1, GLint v2)`
- `void glUniform4i(GLint location, GLint v0, GLint v1, GLint v2, GLint v3)`
- `void glUniform1fv(GLint location, GLsizei count, const GLfloat *value)`
- `void glUniform2fv(GLint location, GLsizei count, const GLfloat *value)`
- `void glUniform3fv(GLint location, GLsizei count, const GLfloat *value)`
- `void glUniform4fv(GLint location, GLsizei count, const GLfloat *value)`
- `void glUniform1iv(GLint location, GLsizei count, const GLint *value)`
- `void glUniform2iv(GLint location, GLsizei count, const GLint *value)`
- `void glUniform3iv(GLint location, GLsizei count, const GLint *value)`
- `void glUniform4iv(GLint location, GLsizei count, const GLint *value)`
- `void glUniformMatrix2fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix3fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix4fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix2x3fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix3x2fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix2x4fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix4x2fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix3x4fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix4x3fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUseProgram(GLuint program)`
- `void glValidateProgram(GLuint program)`
- `void glVertex2s(GLshort x, GLshort y)`

- `void glVertex2i(GLint x, GLint y)`
- `void glVertex2f(GLfloat x, GLfloat y)`
- `void glVertex2d(GLdouble x, GLdouble y)`
- `void glVertex3s(GLshort x, GLshort y, GLshort z)`
- `void glVertex3i(GLint x, GLint y, GLint z)`
- `void glVertex3f(GLfloat x, GLfloat y, GLfloat z)`
- `void glVertex3d(GLdouble x, GLdouble y, GLdouble z)`
- `void glVertex4s(GLshort x, GLshort y, GLshort z, GLshort w)`
- `void glVertex4i(GLint x, GLint y, GLint z, GLint w)`
- `void glVertex4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)`
- `void glVertex4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)`
- `void glVertex2sv(const GLshort * v)`
- `void glVertex2iv(const GLint * v)`
- `void glVertex2fv(const GLfloat * v)`
- `void glVertex2dv(const GLdouble * v)`
- `void glVertex3sv(const GLshort * v)`
- `void glVertex3iv(const GLint * v)`
- `void glVertex3fv(const GLfloat * v)`
- `void glVertex3dv(const GLdouble * v)`
- `void glVertex4sv(const GLshort * v)`
- `void glVertex4iv(const GLint * v)`
- `void glVertex4fv(const GLfloat * v)`
- `void glVertex4dv(const GLdouble * v)`
- `void glVertexAttrib1f(GLuint index, GLfloat v0)`
- `void glVertexAttrib1s(GLuint index, GLshort v0)`
- `void glVertexAttrib1d(GLuint index, GLdouble v0)`
- `void glVertexAttrib2f(GLuint index, GLfloat v0, GLfloat v1)`
- `void glVertexAttrib2s(GLuint index, GLshort v0, GLshort v1)`
- `void glVertexAttrib2d(GLuint index, GLdouble v0, GLdouble v1)`
- `void glVertexAttrib3f(GLuint index, GLfloat v0, GLfloat v1, GLfloat v2)`
- `void glVertexAttrib3s(GLuint index, GLshort v0, GLshort v1, GLshort v2)`
- `void glVertexAttrib3d(GLuint index, GLdouble v0, GLdouble v1, GLdouble v2)`
- `void glVertexAttrib4f(GLuint index, GLfloat v0, GLfloat v1, GLfloat v2, GLfloat v3)`
- `void glVertexAttrib4s(GLuint index, GLshort v0, GLshort v1, GLshort v2, GLshort v3)`
- `void glVertexAttrib4d(GLuint index, GLdouble v0, GLdouble v1, GLdouble v2, GLdouble v3)`
- `void glVertexAttrib4Nub(GLuint index, GLubyte v0, GLubyte v1, GLubyte v2, GLubyte v3)`

- void glVertexAttrib1fv(GLuint index,const GLfloat *v)
- void glVertexAttrib1sv(GLuint index,const GLshort *v)
- void glVertexAttrib1dv(GLuint index,const GLdouble *v)
- void glVertexAttrib2fv(GLuint index,const GLfloat *v)
- void glVertexAttrib2sv(GLuint index,const GLshort *v)
- void glVertexAttrib2dv(GLuint index,const GLdouble *v)
- void glVertexAttrib3fv(GLuint index,const GLfloat *v)
- void glVertexAttrib3sv(GLuint index,const GLshort *v)
- void glVertexAttrib3dv(GLuint index,const GLdouble *v)
- void glVertexAttrib4fv(GLuint index,const GLfloat *v)
- void glVertexAttrib4sv(GLuint index,const GLshort *v)
- void glVertexAttrib4dv(GLuint index,const GLdouble *v)
- void glVertexAttrib4iv(GLuint index,const GLint *v)
- void glVertexAttrib4bv(GLuint index,const GLbyte *v)
- void glVertexAttrib4ubv(GLuint index,const GLubyte *v)
- void glVertexAttrib4usv(GLuint index,const GLushort *v)
- void glVertexAttrib4uiv(GLuint index,const GLuint *v)
- void glVertexAttribPointer(GLuint index,GLint size,GLenum type,GLboolean normalized,GLsizei stride,const GLvoid * pointer)
- void glVertexPointer(GLint size,GLenum type,GLsizei stride,const GLvoid * pointer)
- void glViewport(GLint x,GLint y,GLsizei width,GLsizei height)
- void glWindowPos2s(GLshort x,GLshort y)
- void glWindowPos2i(GLint x,GLint y)
- void glWindowPos2f(GLfloat x,GLfloat y)
- void glWindowPos2d(GLdouble x,GLdouble y)
- void glWindowPos3s(GLshort x,GLshort y,GLshort z)
- void glWindowPos3i(GLint x,GLint y,GLint z)
- void glWindowPos3f(GLfloat x,GLfloat y,GLfloat z)
- void glWindowPos3d(GLdouble x,GLdouble y,GLdouble z)
- void glWindowPos2sv(const GLshort * v)
- void glWindowPos2iv(const GLint * v)
- void glWindowPos2fv(const GLfloat * v)
- void glWindowPos2dv(const GLdouble * v)
- void glWindowPos3sv(const GLshort * v)
- void glWindowPos3iv(const GLint * v)
- void glWindowPos3fv(const GLfloat * v)

- void glWindowPos3dv(const GLdouble * v)
- void gluBeginCurve(GLUnurbs* nurb)
- void gluBeginPolygon(GLUtesselator* tess)
- void gluBeginSurface(GLUnurbs* nurb)
- void gluBeginTrim(GLUnurbs* nurb)
- void gluCylinder(GLUquadric* quad, GLdouble base, GLdouble top, GLdouble height, GLint slices, GLint stacks)
- void gluDeleteNurbsRenderer(GLUnurbs* nurb)
- void gluDeleteQuadric(GLUquadric* quad)
- void gluDeleteTess(GLUtesselator* tess)
- void gluDisk(GLUquadric* quad, GLdouble inner, GLdouble outer, GLint slices, GLint loops)
- void gluEndCurve(GLUnurbs* nurb)
- void gluEndPolygon(GLUtesselator* tess)
- void gluEndSurface(GLUnurbs* nurb)
- void gluEndTrim(GLUnurbs* nurb)
- const GLubyte * gluErrorString(GLenum error)
- void gluGetNurbsProperty(GLUnurbs* nurb, GLenum property, GLfloat* data)
- const GLubyte * gluGetString(GLenum name)
- void gluGetTessProperty(GLUtesselator* tess, GLenum which, GLdouble* data)
- void gluLoadSamplingMatrices(GLUnurbs* nurb, const GLfloat * model, const GLfloat * perspective, const GLint * view)
- void gluLookAt(GLdouble eyeX, GLdouble eyeY, GLdouble eyeZ, GLdouble centerX, GLdouble centerY, GLdouble centerZ, GLdouble upX, GLdouble upY, GLdouble upZ)
- GLUnurbs *gluNewNurbsRenderer(void)
- GLUquadric *gluNewQuadric(void)
- GLUtesselator* gluNewTess(void)
- void gluNextContour(GLUtesselator* tess, GLenum type)
- void gluNurbsCurve(GLUnurbs* nurb, GLint knotCount, GLfloat * knots, GLint stride, GLfloat * control, GLint order, GLenum type)
- void gluNurbsProperty(GLUnurbs* nurb, GLenum property, GLfloat value)
- void gluNurbsSurface(GLUnurbs* nurb, GLint sKnotCount, GLfloat* sKnots, GLint tKnotCount, GLfloat* tKnots, GLint sStride, GLint tStride, GLfloat* control, GLint sOrder, GLint tOrder, GLenum type)
- void gluOrtho2D(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top)
- void gluPartialDisk(GLUquadric* quad, GLdouble inner, GLdouble outer, GLint slices, GLint loops, GLdouble start, GLdouble sweep)
- void gluPerspective(GLdouble fovy, GLdouble aspect, GLdouble zNear, GLdouble zFar)
- void gluPickMatrix(GLdouble x, GLdouble y, GLdouble delX, GLdouble delY, GLint * viewport)
- GLint gluProject(GLdouble objX, GLdouble objY, GLdouble objZ, const GLdouble * model, const GLdouble * proj, const GLint * view, GLdouble* winX, GLdouble* winY, GLdouble* winZ)

- void gluPwlCurve(GLUnurbs* nurb,GLint count,GLfloat* data,GLint stride,GLenum type)
- void gluQuadricDrawStyle(GLUquadric* quad,GLenum draw)
- void gluQuadricNormals(GLUquadric* quad,GLenum normal)
- void gluQuadricOrientation(GLUquadric* quad,GLenum orientation)
- void gluQuadricTexture(GLUquadric* quad,GLboolean texture)
- GLint gluScaleImage(GLenum format,GLsizei wIn,GLsizei hIn,GLenum typeIn,const void * dataIn,GLsizei wOut,GLsizei hOut,GLenum typeOut,GLvoid* dataOut)
- void gluSphere(GLUquadric* quad,GLdouble radius,GLint slices,GLint stacks)
- void gluTessBeginContour(GLUtesselator* tess)
- void gluTessBeginPolygon(GLUtesselator* tess,GLvoid* data)
- void gluTessEndContour(GLUtesselator* tess)
- void gluTessEndPolygon(GLUtesselator* tess)
- void gluTessNormal(GLUtesselator* tess,GLdouble valueX,GLdouble valueY,GLdouble valueZ)
- void gluTessProperty(GLUtesselator* tess,GLenum which,GLdouble data)
- void gluTessVertex(GLUtesselator* tess,GLdouble * location,GLvoid* data)
- GLint gluUnProject(GLdouble winX,GLdouble winY,GLdouble winZ,const GLdouble * model,const GLdouble * proj,const GLint * view,GLdouble* objX,GLdouble* objY,GLdouble* objZ)
- void glDisable(GLenum cap)

RINGOPENGL (OPENGL 3.1) FUNCTIONS REFERENCE

- GL_ZERO
- GL_FALSE
- GL_LOGIC_OP
- GL_NONE
- GL_TEXTURE_COMPONENTS
- GL_NO_ERROR
- GL_POINTS
- GL_CURRENT_BIT
- GL_TRUE
- GL_ONE
- GL_CLIENT_PIXEL_STORE_BIT
- GL_LINES
- GL_LINE_LOOP
- GL_POINT_BIT
- GL_CLIENT_VERTEX_ARRAY_BIT
- GL_LINE_STRIP
- GL_LINE_BIT
- GL_TRIANGLES
- GL_TRIANGLE_STRIP
- GL_TRIANGLE_FAN
- GL_QUADS
- GL_QUAD_STRIP
- GL_POLYGON_BIT
- GL_POLYGON
- GL_POLYGON_STIPPLE_BIT
- GL_PIXEL_MODE_BIT
- GL_LIGHTING_BIT

- GL_FOG_BIT
- GL_DEPTH_BUFFER_BIT
- GL_ACCUM
- GL_LOAD
- GL_RETURN
- GL_MULT
- GL_ADD
- GL_NEVER
- GL_ACCUM_BUFFER_BIT
- GL_LESS
- GL_EQUAL
- GL_LEQUAL
- GL_GREATER
- GL_NOTEQUAL
- GL_GEQUAL
- GL_ALWAYS
- GL_SRC_COLOR
- GL_ONE_MINUS_SRC_COLOR
- GL_SRC_ALPHA
- GL_ONE_MINUS_SRC_ALPHA
- GL_DST_ALPHA
- GL_ONE_MINUS_DST_ALPHA
- GL_DST_COLOR
- GL_ONE_MINUS_DST_COLOR
- GL_SRC_ALPHA_SATURATE
- GL_STENCIL_BUFFER_BIT
- GL_FRONT_LEFT
- GL_FRONT_RIGHT
- GL_BACK_LEFT
- GL_BACK_RIGHT
- GL_FRONT
- GL_BACK
- GL_LEFT
- GL_RIGHT
- GL_FRONT_AND_BACK
- GL_AUX0

- GL_AUX1
- GL_AUX2
- GL_AUX3
- GL_INVALID_ENUM
- GL_INVALID_VALUE
- GL_INVALID_OPERATION
- GL_STACK_OVERFLOW
- GL_STACK_UNDERFLOW
- GL_OUT_OF_MEMORY
- GL_2D
- GL_3D
- GL_3D_COLOR
- GL_3D_COLOR_TEXTURE
- GL_4D_COLOR_TEXTURE
- GL_PASS_THROUGH_TOKEN
- GL_POINT_TOKEN
- GL_LINE_TOKEN
- GL_POLYGON_TOKEN
- GL_BITMAP_TOKEN
- GL_DRAW_PIXEL_TOKEN
- GL_COPY_PIXEL_TOKEN
- GL_LINE_RESET_TOKEN
- GL_EXP
- GL_VIEWPORT_BIT
- GL_EXP2
- GL_CW
- GL_CCW
- GL_COEFF
- GL_ORDER
- GL_DOMAIN
- GL_CURRENT_COLOR
- GL_CURRENT_INDEX
- GL_CURRENT_NORMAL
- GL_CURRENT_TEXTURE_COORDS
- GL_CURRENT_RASTER_COLOR
- GL_CURRENT_RASTER_INDEX

- GL_CURRENT_RASTER_TEXTURE_COORDS
- GL_CURRENT_RASTER_POSITION
- GL_CURRENT_RASTER_POSITION_VALID
- GL_CURRENT_RASTER_DISTANCE
- GL_POINT_SMOOTH
- GL_POINT_SIZE
- GL_POINT_SIZE_RANGE
- GL_POINT_SIZE_GRANULARITY
- GL_LINE_SMOOTH
- GL_LINE_WIDTH
- GL_LINE_WIDTH_RANGE
- GL_LINE_WIDTH_GRANULARITY
- GL_LINE_STIPPLE
- GL_LINE_STIPPLE_PATTERN
- GL_LINE_STIPPLE_REPEAT
- GL_LIST_MODE
- GL_MAX_LIST_NESTING
- GL_LIST_BASE
- GL_LIST_INDEX
- GL_POLYGON_MODE
- GL_POLYGON_SMOOTH
- GL_POLYGON_STIPPLE
- GL_EDGE_FLAG
- GL_CULL_FACE
- GL_CULL_FACE_MODE
- GL_FRONT_FACE
- GL_LIGHTING
- GL_LIGHT_MODEL_LOCAL_VIEWER
- GL_LIGHT_MODEL_TWO_SIDE
- GL_LIGHT_MODEL_AMBIENT
- GL_SHADE_MODEL
- GL_COLOR_MATERIAL_FACE
- GL_COLOR_MATERIAL_PARAMETER
- GL_COLOR_MATERIAL
- GL_FOG
- GL_FOG_INDEX

- GL_FOG_DENSITY
- GL_FOG_START
- GL_FOG_END
- GL_FOG_MODE
- GL_FOG_COLOR
- GL_DEPTH_RANGE
- GL_DEPTH_TEST
- GL_DEPTH_WRITEMASK
- GL_DEPTH_CLEAR_VALUE
- GL_DEPTH_FUNC
- GL_ACCUM_CLEAR_VALUE
- GL_STENCIL_TEST
- GL_STENCIL_CLEAR_VALUE
- GL_STENCIL_FUNC
- GL_STENCIL_VALUE_MASK
- GL_STENCIL_FAIL
- GL_STENCIL_PASS_DEPTH_FAIL
- GL_STENCIL_PASS_DEPTH_PASS
- GL_STENCIL_REF
- GL_STENCIL_WRITEMASK
- GL_MATRIX_MODE
- GL_NORMALIZE
- GL_VIEWPORT
- GL_MODELVIEW_STACK_DEPTH
- GL_PROJECTION_STACK_DEPTH
- GL_TEXTURE_STACK_DEPTH
- GL_MODELVIEW_MATRIX
- GL_PROJECTION_MATRIX
- GL_TEXTURE_MATRIX
- GL_ATTRIB_STACK_DEPTH
- GL_CLIENT_ATTRIB_STACK_DEPTH
- GL_ALPHA_TEST
- GL_ALPHA_TEST_FUNC
- GL_ALPHA_TEST_REF
- GL_DITHER
- GL_BLEND_DST

- GL_BLEND_SRC
- GL_BLEND
- GL_LOGIC_OP_MODE
- GL_INDEX_LOGIC_OP
- GL_COLOR_LOGIC_OP
- GL_AUX_BUFFERS
- GL_DRAW_BUFFER
- GL_READ_BUFFER
- GL_SCISSOR_BOX
- GL_SCISSOR_TEST
- GL_INDEX_CLEAR_VALUE
- GL_INDEX_WRITEMASK
- GL_COLOR_CLEAR_VALUE
- GL_COLOR_WRITEMASK
- GL_INDEX_MODE
- GL_RGBA_MODE
- GL_DOUBLEBUFFER
- GL_STEREO
- GL_RENDER_MODE
- GL_PERSPECTIVE_CORRECTION_HINT
- GL_POINT_SMOOTH_HINT
- GL_LINE_SMOOTH_HINT
- GL_POLYGON_SMOOTH_HINT
- GL_FOG_HINT
- GL_TEXTURE_GEN_S
- GL_TEXTURE_GEN_T
- GL_TEXTURE_GEN_R
- GL_TEXTURE_GEN_Q
- GL_PIXEL_MAP_I_TO_I
- GL_PIXEL_MAP_S_TO_S
- GL_PIXEL_MAP_I_TO_R
- GL_PIXEL_MAP_I_TO_G
- GL_PIXEL_MAP_I_TO_B
- GL_PIXEL_MAP_I_TO_A
- GL_PIXEL_MAP_R_TO_R
- GL_PIXEL_MAP_G_TO_G

- GL_PIXEL_MAP_B_TO_B
- GL_PIXEL_MAP_A_TO_A
- GL_PIXEL_MAP_I_TO_I_SIZE
- GL_PIXEL_MAP_S_TO_S_SIZE
- GL_PIXEL_MAP_I_TO_R_SIZE
- GL_PIXEL_MAP_I_TO_G_SIZE
- GL_PIXEL_MAP_I_TO_B_SIZE
- GL_PIXEL_MAP_I_TO_A_SIZE
- GL_PIXEL_MAP_R_TO_R_SIZE
- GL_PIXEL_MAP_G_TO_G_SIZE
- GL_PIXEL_MAP_B_TO_B_SIZE
- GL_PIXEL_MAP_A_TO_A_SIZE
- GL_UNPACK_SWAP_BYTES
- GL_UNPACK_LSB_FIRST
- GL_UNPACK_ROW_LENGTH
- GL_UNPACK_SKIP_ROWS
- GL_UNPACK_SKIP_PIXELS
- GL_UNPACK_ALIGNMENT
- GL_PACK_SWAP_BYTES
- GL_PACK_LSB_FIRST
- GL_PACK_ROW_LENGTH
- GL_PACK_SKIP_ROWS
- GL_PACK_SKIP_PIXELS
- GL_PACK_ALIGNMENT
- GL_MAP_COLOR
- GL_MAP_STENCIL
- GL_INDEX_SHIFT
- GL_INDEX_OFFSET
- GL_RED_SCALE
- GL_RED_BIAS
- GL_ZOOM_X
- GL_ZOOM_Y
- GL_GREEN_SCALE
- GL_GREEN_BIAS
- GL_BLUE_SCALE
- GL_BLUE_BIAS

- GL_ALPHA_SCALE
- GL_ALPHA_BIAS
- GL_DEPTH_SCALE
- GL_DEPTH_BIAS
- GL_MAX_EVAL_ORDER
- GL_MAX_LIGHTS
- GL_MAX_CLIP_PLANES
- GL_MAX_TEXTURE_SIZE
- GL_MAX_PIXEL_MAP_TABLE
- GL_MAX_ATTRIB_STACK_DEPTH
- GL_MAX_MODELVIEW_STACK_DEPTH
- GL_MAX_NAME_STACK_DEPTH
- GL_MAX_PROJECTION_STACK_DEPTH
- GL_MAX_TEXTURE_STACK_DEPTH
- GL_MAX_VIEWPORT_DIMS
- GL_MAX_CLIENT_ATTRIB_STACK_DEPTH
- GL_SUBPIXEL_BITS
- GL_INDEX_BITS
- GL_RED_BITS
- GL_GREEN_BITS
- GL_BLUE_BITS
- GL_ALPHA_BITS
- GL_DEPTH_BITS
- GL_STENCIL_BITS
- GL_ACCUM_RED_BITS
- GL_ACCUM_GREEN_BITS
- GL_ACCUM_BLUE_BITS
- GL_ACCUM_ALPHA_BITS
- GL_NAME_STACK_DEPTH
- GL_AUTO_NORMAL
- GL_MAP1_COLOR_4
- GL_MAP1_INDEX
- GL_MAP1_NORMAL
- GL_MAP1_TEXTURE_COORD_1
- GL_MAP1_TEXTURE_COORD_2
- GL_MAP1_TEXTURE_COORD_3

- GL_MAP1_TEXTURE_COORD_4
- GL_MAP1_VERTEX_3
- GL_MAP1_VERTEX_4
- GL_MAP2_COLOR_4
- GL_MAP2_INDEX
- GL_MAP2_NORMAL
- GL_MAP2_TEXTURE_COORD_1
- GL_MAP2_TEXTURE_COORD_2
- GL_MAP2_TEXTURE_COORD_3
- GL_MAP2_TEXTURE_COORD_4
- GL_MAP2_VERTEX_3
- GL_MAP2_VERTEX_4
- GL_MAP1_GRID_DOMAIN
- GL_MAP1_GRID_SEGMENTS
- GL_MAP2_GRID_DOMAIN
- GL_MAP2_GRID_SEGMENTS
- GL_TEXTURE_1D
- GL_TEXTURE_2D
- GL_FEEDBACK_BUFFER_POINTER
- GL_FEEDBACK_BUFFER_SIZE
- GL_FEEDBACK_BUFFER_TYPE
- GL_SELECTION_BUFFER_POINTER
- GL_SELECTION_BUFFER_SIZE
- GL_TEXTURE_WIDTH
- GL_TRANSFORM_BIT
- GL_TEXTURE_HEIGHT
- GL_TEXTURE_INTERNAL_FORMAT
- GL_TEXTURE_BORDER_COLOR
- GL_TEXTURE_BORDER
- GL_DONT_CARE
- GL_FASTEST
- GL_NICEST
- GL_AMBIENT
- GL_DIFFUSE
- GL_SPECULAR
- GL_POSITION

- GL_SPOT_DIRECTION
- GL_SPOT_EXPONENT
- GL_SPOT_CUTOFF
- GL_CONSTANT_ATTENUATION
- GL_LINEAR_ATTENUATION
- GL_QUADRATIC_ATTENUATION
- GL_COMPILE
- GL_COMPILE_AND_EXECUTE
- GL_BYTE
- GL_UNSIGNED_BYTE
- GL_SHORT
- GL_UNSIGNED_SHORT
- GL_INT
- GL_UNSIGNED_INT
- GL_FLOAT
- GL_2_BYTES
- GL_3_BYTES
- GL_4_BYTES
- GL_DOUBLE
- GL_CLEAR
- GL_AND
- GL_AND_REVERSE
- GL_COPY
- GL_AND_INVERTED
- GL_NOOP
- GL_XOR
- GL_OR
- GL_NOR
- GL_EQUIV
- GL_INVERT
- GL_OR_REVERSE
- GL_COPY_INVERTED
- GL_OR_INVERTED
- GL_NAND
- GL_SET
- GL_EMISSION

- GL_SHININESS
- GL_AMBIENT_AND_DIFFUSE
- GL_COLOR_INDEXES
- GL_MODELVIEW
- GL_PROJECTION
- GL_TEXTURE
- GL_COLOR
- GL_DEPTH
- GL_STENCIL
- GL_COLOR_INDEX
- GL_STENCIL_INDEX
- GL_DEPTH_COMPONENT
- GL_RED
- GL_GREEN
- GL_BLUE
- GL_ALPHA
- GL_RGB
- GL_RGBA
- GL_LUMINANCE
- GL_LUMINANCE_ALPHA
- GL_BITMAP
- GL_POINT
- GL_LINE
- GL_FILL
- GL_RENDER
- GL_FEEDBACK
- GL_SELECT
- GL_FLAT
- GL_SMOOTH
- GL_KEEP
- GL_REPLACE
- GL_INCR
- GL_DECR
- GL_VENDOR
- GL_RENDERER
- GL_VERSION

- GL_EXTENSIONS
- GL_S
- GL_ENABLE_BIT
- GL_T
- GL_R
- GL_Q
- GL_MODULATE
- GL_DECAL
- GL_TEXTURE_ENV_MODE
- GL_TEXTURE_ENV_COLOR
- GL_TEXTURE_ENV
- GL_EYE_LINEAR
- GL_OBJECT_LINEAR
- GL_SPHERE_MAP
- GL_TEXTURE_GEN_MODE
- GL_OBJECT_PLANE
- GL_EYE_PLANE
- GL_NEAREST
- GL_LINEAR
- GL_NEAREST_MIPMAP_NEAREST
- GL_LINEAR_MIPMAP_NEAREST
- GL_NEAREST_MIPMAP_LINEAR
- GL_LINEAR_MIPMAP_LINEAR
- GL_TEXTURE_MAG_FILTER
- GL_TEXTURE_MIN_FILTER
- GL_TEXTURE_WRAP_S
- GL_TEXTURE_WRAP_T
- GL_CLAMP
- GL_REPEAT
- GL_POLYGON_OFFSET_UNITS
- GL_POLYGON_OFFSET_POINT
- GL_POLYGON_OFFSET_LINE
- GL_R3_G3_B2
- GL_V2F
- GL_V3F
- GL_C4UB_V2F

- GL_C4UB_V3F
- GL_C3F_V3F
- GL_N3F_V3F
- GL_C4F_N3F_V3F
- GL_T2F_V3F
- GL_T4F_V4F
- GL_T2F_C4UB_V3F
- GL_T2F_C3F_V3F
- GL_T2F_N3F_V3F
- GL_T2F_C4F_N3F_V3F
- GL_T4F_C4F_N3F_V4F
- GL_CLIP_PLANE0
- GL_CLIP_PLANE1
- GL_CLIP_PLANE2
- GL_CLIP_PLANE3
- GL_CLIP_PLANE4
- GL_CLIP_PLANE5
- GL_LIGHT0
- GL_COLOR_BUFFER_BIT
- GL_LIGHT1
- GL_LIGHT2
- GL_LIGHT3
- GL_LIGHT4
- GL_LIGHT5
- GL_LIGHT6
- GL_LIGHT7
- GL_HINT_BIT
- GL_POLYGON_OFFSET_FILL
- GL_POLYGON_OFFSET_FACTOR
- GL_ALPHA4
- GL_ALPHA8
- GL_ALPHA12
- GL_ALPHA16
- GL_LUMINANCE4
- GL_LUMINANCE8
- GL_LUMINANCE12

- GL_LUMINANCE16
- GL_LUMINANCE4_ALPHA4
- GL_LUMINANCE6_ALPHA2
- GL_LUMINANCE8_ALPHA8
- GL_LUMINANCE12_ALPHA4
- GL_LUMINANCE12_ALPHA12
- GL_LUMINANCE16_ALPHA16
- GL_INTENSITY
- GL_INTENSITY4
- GL_INTENSITY8
- GL_INTENSITY12
- GL_INTENSITY16
- GL_RGB4
- GL_RGB5
- GL_RGB8
- GL_RGB10
- GL_RGB12
- GL_RGB16
- GL_RGBA2
- GL_RGBA4
- GL_RGB5_A1
- GL_RGBA8
- GL_RGB10_A2
- GL_RGBA12
- GL_RGBA16
- GL_TEXTURE_RED_SIZE
- GL_TEXTURE_GREEN_SIZE
- GL_TEXTURE_BLUE_SIZE
- GL_TEXTURE_ALPHA_SIZE
- GL_TEXTURE_LUMINANCE_SIZE
- GL_TEXTURE_INTENSITY_SIZE
- GL_PROXY_TEXTURE_1D
- GL_PROXY_TEXTURE_2D
- GL_TEXTURE_PRIORITY
- GL_TEXTURE_RESIDENT
- GL_TEXTURE_BINDING_1D

- GL_TEXTURE_BINDING_2D
- GL_VERTEX_ARRAY
- GL_NORMAL_ARRAY
- GL_COLOR_ARRAY
- GL_INDEX_ARRAY
- GL_TEXTURE_COORD_ARRAY
- GL_EDGE_FLAG_ARRAY
- GL_VERTEX_ARRAY_SIZE
- GL_VERTEX_ARRAY_TYPE
- GL_VERTEX_ARRAY_STRIDE
- GL_NORMAL_ARRAY_TYPE
- GL_NORMAL_ARRAY_STRIDE
- GL_COLOR_ARRAY_SIZE
- GL_COLOR_ARRAY_TYPE
- GL_COLOR_ARRAY_STRIDE
- GL_INDEX_ARRAY_TYPE
- GL_INDEX_ARRAY_STRIDE
- GL_TEXTURE_COORD_ARRAY_SIZE
- GL_TEXTURE_COORD_ARRAY_TYPE
- GL_TEXTURE_COORD_ARRAY_STRIDE
- GL_EDGE_FLAG_ARRAY_STRIDE
- GL_VERTEX_ARRAY_POINTER
- GL_NORMAL_ARRAY_POINTER
- GL_COLOR_ARRAY_POINTER
- GL_INDEX_ARRAY_POINTER
- GL_TEXTURE_COORD_ARRAY_POINTER
- GL_EDGE_FLAG_ARRAY_POINTER
- GL_COLOR_INDEX1_EXT
- GL_COLOR_INDEX2_EXT
- GL_COLOR_INDEX4_EXT
- GL_COLOR_INDEX8_EXT
- GL_COLOR_INDEX12_EXT
- GL_COLOR_INDEX16_EXT
- GL_EVAL_BIT
- GL_LIST_BIT
- GL_TEXTURE_BIT

- GL_SCISSOR_BIT
- GL_ALL_ATTRIB_BITS
- GL_CLIENT_ALL_ATTRIB_BITS
- GL_SMOOTH_POINT_SIZE_RANGE
- GL_SMOOTH_POINT_SIZE_GRANULARITY
- GL_SMOOTH_LINE_WIDTH_RANGE
- GL_SMOOTH_LINE_WIDTH_GRANULARITY
- GL_UNSIGNED_BYTE_3_3_2
- GL_UNSIGNED_SHORT_4_4_4_4
- GL_UNSIGNED_SHORT_5_5_5_1
- GL_UNSIGNED_INT_8_8_8_8
- GL_UNSIGNED_INT_10_10_10_2
- GL_RESCALE_NORMAL
- GL_TEXTURE_BINDING_3D
- GL_PACK_SKIP_IMAGES
- GL_PACK_IMAGE_HEIGHT
- GL_UNPACK_SKIP_IMAGES
- GL_UNPACK_IMAGE_HEIGHT
- GL_TEXTURE_3D
- GL_PROXY_TEXTURE_3D
- GL_TEXTURE_DEPTH
- GL_TEXTURE_WRAP_R
- GL_MAX_3D_TEXTURE_SIZE
- GL_BGR
- GL_BGRA
- GL_MAX_ELEMENTS_VERTICES
- GL_MAX_ELEMENTS_INDICES
- GL_CLAMP_TO_EDGE
- GL_TEXTURE_MIN_LOD
- GL_TEXTURE_MAX_LOD
- GL_TEXTURE_BASE_LEVEL
- GL_TEXTURE_MAX_LEVEL
- GL_LIGHT_MODEL_COLOR_CONTROL
- GL_SINGLE_COLOR
- GL_SEPARATE_SPECULAR_COLOR
- GL_UNSIGNED_BYTE_2_3_3_REV

- GL_UNSIGNED_SHORT_5_6_5
- GL_UNSIGNED_SHORT_5_6_5_REV
- GL_UNSIGNED_SHORT_4_4_4_4_REV
- GL_UNSIGNED_SHORT_1_5_5_5_REV
- GL_UNSIGNED_INT_8_8_8_8_REV
- GL_ALIASED_POINT_SIZE_RANGE
- GL_ALIASED_LINE_WIDTH_RANGE
- GL_MULTISAMPLE
- GL_SAMPLE_ALPHA_TO_COVERAGE
- GL_SAMPLE_ALPHA_TO_ONE
- GL_SAMPLE_COVERAGE
- GL_SAMPLE_BUFFERS
- GL_SAMPLES
- GL_SAMPLE_COVERAGE_VALUE
- GL_SAMPLE_COVERAGE_INVERT
- GL_CLAMP_TO_BORDER
- GL_TEXTURE0
- GL_TEXTURE1
- GL_TEXTURE2
- GL_TEXTURE3
- GL_TEXTURE4
- GL_TEXTURE5
- GL_TEXTURE6
- GL_TEXTURE7
- GL_TEXTURE8
- GL_TEXTURE9
- GL_TEXTURE10
- GL_TEXTURE11
- GL_TEXTURE12
- GL_TEXTURE13
- GL_TEXTURE14
- GL_TEXTURE15
- GL_TEXTURE16
- GL_TEXTURE17
- GL_TEXTURE18
- GL_TEXTURE19

- GL_TEXTURE20
- GL_TEXTURE21
- GL_TEXTURE22
- GL_TEXTURE23
- GL_TEXTURE24
- GL_TEXTURE25
- GL_TEXTURE26
- GL_TEXTURE27
- GL_TEXTURE28
- GL_TEXTURE29
- GL_TEXTURE30
- GL_TEXTURE31
- GL_ACTIVE_TEXTURE
- GL_CLIENT_ACTIVE_TEXTURE
- GL_MAX_TEXTURE_UNITS
- GL_TRANSPOSE_MODELVIEW_MATRIX
- GL_TRANSPOSE_PROJECTION_MATRIX
- GL_TRANSPOSE_TEXTURE_MATRIX
- GL_TRANSPOSE_COLOR_MATRIX
- GL_SUBTRACT
- GL_COMPRESSED_ALPHA
- GL_COMPRESSED_LUMINANCE
- GL_COMPRESSED_LUMINANCE_ALPHA
- GL_COMPRESSED_INTENSITY
- GL_COMPRESSED_RGB
- GL_COMPRESSED_RGBA
- GL_TEXTURE_COMPRESSION_HINT
- GL_NORMAL_MAP
- GL_REFLECTION_MAP
- GL_TEXTURE_CUBE_MAP
- GL_TEXTURE_BINDING_CUBE_MAP
- GL_TEXTURE_CUBE_MAP_POSITIVE_X
- GL_TEXTURE_CUBE_MAP_NEGATIVE_X
- GL_TEXTURE_CUBE_MAP_POSITIVE_Y
- GL_TEXTURE_CUBE_MAP_NEGATIVE_Y
- GL_TEXTURE_CUBE_MAP_POSITIVE_Z

- GL_TEXTURE_CUBE_MAP_NEGATIVE_Z
- GL_PROXY_TEXTURE_CUBE_MAP
- GL_MAX_CUBE_MAP_TEXTURE_SIZE
- GL_COMBINE
- GL_COMBINE_RGB
- GL_COMBINE_ALPHA
- GL_RGB_SCALE
- GL_ADD_SIGNED
- GL_INTERPOLATE
- GL_CONSTANT
- GL_PRIMARY_COLOR
- GL_PREVIOUS
- GL_SOURCE0_RGB
- GL_SOURCE1_RGB
- GL_SOURCE2_RGB
- GL_SOURCE0_ALPHA
- GL_SOURCE1_ALPHA
- GL_SOURCE2_ALPHA
- GL_OPERAND0_RGB
- GL_OPERAND1_RGB
- GL_OPERAND2_RGB
- GL_OPERAND0_ALPHA
- GL_OPERAND1_ALPHA
- GL_OPERAND2_ALPHA
- GL_TEXTURE_COMPRESSED_IMAGE_SIZE
- GL_TEXTURE_COMPRESSED
- GL_NUM_COMPRESSED_TEXTURE_FORMATS
- GL_COMPRESSED_TEXTURE_FORMATS
- GL_DOT3_RGB
- GL_DOT3_RGBA
- GL_MULTISAMPLE_BIT
- GL_BLEND_DST_RGB
- GL_BLEND_SRC_RGB
- GL_BLEND_DST_ALPHA
- GL_BLEND_SRC_ALPHA
- GL_POINT_SIZE_MIN

- GL_POINT_SIZE_MAX
- GL_POINT_FADE_THRESHOLD_SIZE
- GL_POINT_DISTANCE_ATTENUATION
- GL_GENERATE_MIPMAP
- GL_GENERATE_MIPMAP_HINT
- GL_DEPTH_COMPONENT16
- GL_DEPTH_COMPONENT24
- GL_DEPTH_COMPONENT32
- GL_MIRRORED_REPEAT
- GL_FOG_COORDINATE_SOURCE
- GL_FOG_COORDINATE
- GL_FRAGMENT_DEPTH
- GL_CURRENT_FOG_COORDINATE
- GL_FOG_COORDINATE_ARRAY_TYPE
- GL_FOG_COORDINATE_ARRAY_STRIDE
- GL_FOG_COORDINATE_ARRAY_POINTER
- GL_FOG_COORDINATE_ARRAY
- GL_COLOR_SUM
- GL_CURRENT_SECONDARY_COLOR
- GL_SECONDARY_COLOR_ARRAY_SIZE
- GL_SECONDARY_COLOR_ARRAY_TYPE
- GL_SECONDARY_COLOR_ARRAY_STRIDE
- GL_SECONDARY_COLOR_ARRAY_POINTER
- GL_SECONDARY_COLOR_ARRAY
- GL_MAX_TEXTURE_LOD_BIAS
- GL_TEXTURE_FILTER_CONTROL
- GL_TEXTURE_LOD_BIAS
- GL_INCR_WRAP
- GL_DECR_WRAP
- GL_TEXTURE_DEPTH_SIZE
- GL_DEPTH_TEXTURE_MODE
- GL_TEXTURE_COMPARE_MODE
- GL_TEXTURE_COMPARE_FUNC
- GL_COMPARE_R_TO_TEXTURE
- GL_CURRENT_FOG_COORD
- GL_FOG_COORD

- GL_FOG_COORD_ARRAY
- GL_FOG_COORD_ARRAY_BUFFER_BINDING
- GL_FOG_COORD_ARRAY_POINTER
- GL_FOG_COORD_ARRAY_STRIDE
- GL_FOG_COORD_ARRAY_TYPE
- GL_FOG_COORD_SRC
- GL_SRC0_ALPHA
- GL_SRC0_RGB
- GL_SRC1_ALPHA
- GL_SRC1_RGB
- GL_SRC2_ALPHA
- GL_SRC2_RGB
- GL_BUFFER_SIZE
- GL_BUFFER_USAGE
- GL_QUERY_COUNTER_BITS
- GL_CURRENT_QUERY
- GL_QUERY_RESULT
- GL_QUERY_RESULT_AVAILABLE
- GL_ARRAY_BUFFER
- GL_ELEMENT_ARRAY_BUFFER
- GL_ARRAY_BUFFER_BINDING
- GL_ELEMENT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ARRAY_BUFFER_BINDING
- GL_NORMAL_ARRAY_BUFFER_BINDING
- GL_COLOR_ARRAY_BUFFER_BINDING
- GL_INDEX_ARRAY_BUFFER_BINDING
- GL_TEXTURE_COORD_ARRAY_BUFFER_BINDING
- GL_EDGE_FLAG_ARRAY_BUFFER_BINDING
- GL_SECONDARY_COLOR_ARRAY_BUFFER_BINDING
- GL_FOG_COORDINATE_ARRAY_BUFFER_BINDING
- GL_WEIGHT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ATTRIB_ARRAY_BUFFER_BINDING
- GL_READ_ONLY
- GL_WRITE_ONLY
- GL_READ_WRITE
- GL_BUFFER_ACCESS

- GL_BUFFER_MAPPED
- GL_BUFFER_MAP_POINTER
- GL_STREAM_DRAW
- GL_STREAM_READ
- GL_STREAM_COPY
- GL_STATIC_DRAW
- GL_STATIC_READ
- GL_STATIC_COPY
- GL_DYNAMIC_DRAW
- GL_DYNAMIC_READ
- GL_DYNAMIC_COPY
- GL_SAMPLES_PASSED
- GL_BLEND_EQUATION_RGB
- GL_VERTEX_ATTRIB_ARRAY_ENABLED
- GL_VERTEX_ATTRIB_ARRAY_SIZE
- GL_VERTEX_ATTRIB_ARRAY_STRIDE
- GL_VERTEX_ATTRIB_ARRAY_TYPE
- GL_CURRENT_VERTEX_ATTRIB
- GL_VERTEX_PROGRAM_POINT_SIZE
- GL_VERTEX_PROGRAM_TWO_SIDE
- GL_VERTEX_ATTRIB_ARRAY_POINTER
- GL_STENCIL_BACK_FUNC
- GL_STENCIL_BACK_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_PASS
- GL_MAX_DRAW_BUFFERS
- GL_DRAW_BUFFER0
- GL_DRAW_BUFFER1
- GL_DRAW_BUFFER2
- GL_DRAW_BUFFER3
- GL_DRAW_BUFFER4
- GL_DRAW_BUFFER5
- GL_DRAW_BUFFER6
- GL_DRAW_BUFFER7
- GL_DRAW_BUFFER8
- GL_DRAW_BUFFER9

- GL_DRAW_BUFFER10
- GL_DRAW_BUFFER11
- GL_DRAW_BUFFER12
- GL_DRAW_BUFFER13
- GL_DRAW_BUFFER14
- GL_DRAW_BUFFER15
- GL_BLEND_EQUATION_ALPHA
- GL_POINT_SPRITE
- GL_COORD_REPLACE
- GL_MAX_VERTEX_ATTRIBS
- GL_VERTEX_ATTRIB_ARRAY_NORMALIZED
- GL_MAX_TEXTURE_COORDS
- GL_MAX_TEXTURE_IMAGE_UNITS
- GL_FRAGMENT_SHADER
- GL_VERTEX_SHADER
- GL_MAX_FRAGMENT_UNIFORM_COMPONENTS
- GL_MAX_VERTEX_UNIFORM_COMPONENTS
- GL_MAX_VARYING_FLOATS
- GL_MAX_VERTEX_TEXTURE_IMAGE_UNITS
- GL_MAX_COMBINED_TEXTURE_IMAGE_UNITS
- GL_SHADER_TYPE
- GL_FLOAT_VEC2
- GL_FLOAT_VEC3
- GL_FLOAT_VEC4
- GL_INT_VEC2
- GL_INT_VEC3
- GL_INT_VEC4
- GL_BOOL
- GL_BOOL_VEC2
- GL_BOOL_VEC3
- GL_BOOL_VEC4
- GL_FLOAT_MAT2
- GL_FLOAT_MAT3
- GL_FLOAT_MAT4
- GL_SAMPLER_1D
- GL_SAMPLER_2D

- GL_SAMPLER_3D
- GL_SAMPLER_CUBE
- GL_SAMPLER_1D_SHADOW
- GL_SAMPLER_2D_SHADOW
- GL_DELETE_STATUS
- GL_COMPILE_STATUS
- GL_LINK_STATUS
- GL_VALIDATE_STATUS
- GL_INFO_LOG_LENGTH
- GL_ATTACHED_SHADERS
- GL_ACTIVE_UNIFORMS
- GL_ACTIVE_UNIFORM_MAX_LENGTH
- GL_SHADER_SOURCE_LENGTH
- GL_ACTIVE_ATTRIBUTES
- GL_ACTIVE_ATTRIBUTE_MAX_LENGTH
- GL_FRAGMENT_SHADER_DERIVATIVE_HINT
- GL_SHADING_LANGUAGE_VERSION
- GL_CURRENT_PROGRAM
- GL_POINT_SPRITE_COORD_ORIGIN
- GL_LOWER_LEFT
- GL_UPPER_LEFT
- GL_STENCIL_BACK_REF
- GL_STENCIL_BACK_VALUE_MASK
- GL_STENCIL_BACK_WRITEMASK
- GL_CURRENT_RASTER_SECONDARY_COLOR
- GL_PIXEL_PACK_BUFFER
- GL_PIXEL_UNPACK_BUFFER
- GL_PIXEL_PACK_BUFFER_BINDING
- GL_PIXEL_UNPACK_BUFFER_BINDING
- GL_FLOAT_MAT2x3
- GL_FLOAT_MAT2x4
- GL_FLOAT_MAT3x2
- GL_FLOAT_MAT3x4
- GL_FLOAT_MAT4x2
- GL_FLOAT_MAT4x3
- GL_SRGB

- GL_SRGB8
- GL_SRGB_ALPHA
- GL_SRGB8_ALPHA8
- GL_SLUMINANCE_ALPHA
- GL_SLUMINANCE8_ALPHA8
- GL_SLUMINANCE
- GL_SLUMINANCE8
- GL_COMPRESSED_SRGB
- GL_COMPRESSED_SRGB_ALPHA
- GL_COMPRESSED_SLUMINANCE
- GL_COMPRESSED_SLUMINANCE_ALPHA
- GL_CLIP_DISTANCE0
- GL_CLIP_DISTANCE1
- GL_CLIP_DISTANCE2
- GL_CLIP_DISTANCE3
- GL_CLIP_DISTANCE4
- GL_CLIP_DISTANCE5
- GL_COMPARE_REF_TO_TEXTURE
- GL_MAX_CLIP_DISTANCES
- GL_MAX_VARYING_COMPONENTS
- GL_CONTEXT_FLAG_FORWARD_COMPATIBLE_BIT
- GL_MAJOR_VERSION
- GL_MINOR_VERSION
- GL_NUM_EXTENSIONS
- GL_CONTEXT_FLAGS
- GL_DEPTH_BUFFER
- GL_STENCIL_BUFFER
- GL_RGBA32F
- GL_RGB32F
- GL_RGBA16F
- GL_RGB16F
- GL_VERTEX_ATTRIB_ARRAY_INTEGER
- GL_MAX_ARRAY_TEXTURE_LAYERS
- GL_MIN_PROGRAM_TEXEL_OFFSET
- GL_MAX_PROGRAM_TEXEL_OFFSET
- GL_CLAMP_VERTEX_COLOR

- GL_CLAMP_FRAGMENT_COLOR
- GL_CLAMP_READ_COLOR
- GL_FIXED_ONLY
- GL_TEXTURE_RED_TYPE
- GL_TEXTURE_GREEN_TYPE
- GL_TEXTURE_BLUE_TYPE
- GL_TEXTURE_ALPHA_TYPE
- GL_TEXTURE_LUMINANCE_TYPE
- GL_TEXTURE_INTENSITY_TYPE
- GL_TEXTURE_DEPTH_TYPE
- GL_TEXTURE_1D_ARRAY
- GL_PROXY_TEXTURE_1D_ARRAY
- GL_TEXTURE_2D_ARRAY
- GL_PROXY_TEXTURE_2D_ARRAY
- GL_TEXTURE_BINDING_1D_ARRAY
- GL_TEXTURE_BINDING_2D_ARRAY
- GL_R11F_G11F_B10F
- GL_UNSIGNED_INT_10F_11F_11F_REV
- GL_RGB9_E5
- GL_UNSIGNED_INT_5_9_9_9_REV
- GL_TEXTURE_SHARED_SIZE
- GL_TRANSFORM_FEEDBACK_VARYING_MAX_LENGTH
- GL_TRANSFORM_FEEDBACK_BUFFER_MODE
- GL_MAX_TRANSFORM_FEEDBACK_SEPARATE_COMPONENTS
- GL_TRANSFORM_FEEDBACK_VARYINGS
- GL_TRANSFORM_FEEDBACK_BUFFER_START
- GL_TRANSFORM_FEEDBACK_BUFFER_SIZE
- GL_PRIMITIVES_GENERATED
- GL_TRANSFORM_FEEDBACK_PRIMITIVES_WRITTEN
- GL_RASTERIZER_DISCARD
- GL_MAX_TRANSFORM_FEEDBACK_INTERLEAVED_COMPONENTS
- GL_MAX_TRANSFORM_FEEDBACK_SEPARATE_ATTRIBS
- GL_INTERLEAVED_ATTRIBS
- GL_SEPARATE_ATTRIBS
- GL_TRANSFORM_FEEDBACK_BUFFER
- GL_TRANSFORM_FEEDBACK_BUFFER_BINDING

- GL_RGBA32UI
- GL_RGB32UI
- GL_RGBA16UI
- GL_RGB16UI
- GL_RGBA8UI
- GL_RGB8UI
- GL_RGBA32I
- GL_RGB32I
- GL_RGBA16I
- GL_RGB16I
- GL_RGBA8I
- GL_RGB8I
- GL_RED_INTEGER
- GL_GREEN_INTEGER
- GL_BLUE_INTEGER
- GL_ALPHA_INTEGER
- GL_RGB_INTEGER
- GL_RGBA_INTEGER
- GL_BGR_INTEGER
- GL_BGRA_INTEGER
- GL_SAMPLER_1D_ARRAY
- GL_SAMPLER_2D_ARRAY
- GL_SAMPLER_1D_ARRAY_SHADOW
- GL_SAMPLER_2D_ARRAY_SHADOW
- GL_SAMPLER_CUBE_SHADOW
- GL_UNSIGNED_INT_VEC2
- GL_UNSIGNED_INT_VEC3
- GL_UNSIGNED_INT_VEC4
- GL_INT_SAMPLER_1D
- GL_INT_SAMPLER_2D
- GL_INT_SAMPLER_3D
- GL_INT_SAMPLER_CUBE
- GL_INT_SAMPLER_1D_ARRAY
- GL_INT_SAMPLER_2D_ARRAY
- GL_UNSIGNED_INT_SAMPLER_1D
- GL_UNSIGNED_INT_SAMPLER_2D

- GL_UNSIGNED_INT_SAMPLER_3D
- GL_UNSIGNED_INT_SAMPLER_CUBE
- GL_UNSIGNED_INT_SAMPLER_1D_ARRAY
- GL_UNSIGNED_INT_SAMPLER_2D_ARRAY
- GL_QUERY_WAIT
- GL_QUERY_NO_WAIT
- GL_QUERY_BY_REGION_WAIT
- GL_QUERY_BY_REGION_NO_WAIT
- GL_TEXTURE_RECTANGLE
- GL_TEXTURE_BINDING_RECTANGLE
- GL_PROXY_TEXTURE_RECTANGLE
- GL_MAX_RECTANGLE_TEXTURE_SIZE
- GL_SAMPLER_2D_RECT
- GL_SAMPLER_2D_RECT_SHADOW
- GL_TEXTURE_BUFFER
- GL_MAX_TEXTURE_BUFFER_SIZE
- GL_TEXTURE_BINDING_BUFFER
- GL_TEXTURE_BUFFER_DATA_STORE_BINDING
- GL_TEXTURE_BUFFER_FORMAT
- GL_SAMPLER_BUFFER
- GL_INT_SAMPLER_2D_RECT
- GL_INT_SAMPLER_BUFFER
- GL_UNSIGNED_INT_SAMPLER_2D_RECT
- GL_UNSIGNED_INT_SAMPLER_BUFFER
- GL_RED_SNORM
- GL_RG_SNORM
- GL_RGB_SNORM
- GL_RGBA_SNORM
- GL_R8_SNORM
- GL_RG8_SNORM
- GL_RGB8_SNORM
- GL_RGBA8_SNORM
- GL_R16_SNORM
- GL_RG16_SNORM
- GL_RGB16_SNORM
- GL_RGBA16_SNORM

- GL_SIGNED_NORMALIZED
- GL_PRIMITIVE_RESTART
- GL_PRIMITIVE_RESTART_INDEX
- GL_BUFFER_ACCESS_FLAGS
- GL_BUFFER_MAP_LENGTH
- GL_BUFFER_MAP_OFFSET
- void glAccum(GLenum op, GLfloat value)
- void glActiveTexture(GLenum texture)
- void glAlphaFunc(GLenum func, GLclampf ref)
- GLboolean glAreTexturesResident(GLsizei n, const GLuint * textures, GLboolean * residences)
- void glArrayElement(GLint i)
- void glAttachShader(GLuint program, GLuint shader)
- void glBegin(GLenum mode)
- void glBeginQuery(GLenum target, GLuint id)
- void glBindAttribLocation(GLuint program, GLuint index, const GLchar * name)
- void glBindBuffer(GLenum target, GLuint buffer)
- void glBindTexture(GLenum target, GLuint texture)
- void glBitmap(GLsizei width, GLsizei height, GLfloat xorig, GLfloat yorig, GLfloat xmove, GLfloat ymove, const GLubyte * bitmap)
- void glBlendColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)
- void glBlendEquation(GLenum mode)
- void glBlendEquationSeparate(GLenum modeRGB, GLenum modeAlpha)
- void glBlendFunc(GLenum sfactor, GLenum dfactor)
- void glBlendFuncSeparate(GLenum srcRGB, GLenum dstRGB, GLenum srcAlpha, GLenum dstAlpha)
- void glBufferData(GLenum target, GLsizeiptr size, const GLvoid * data, GLenum usage)
- void glBufferSubData(GLenum target, GLintptr offset, GLsizeiptr size, const GLvoid * data)
- void glCallList(GLuint list)
- void glCallLists(GLsizei n, GLenum type, const GLvoid * lists)
- void glClear(GLbitfield mask)
- void glClearAccum(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glClearColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)
- void glClearDepth(GLclampd depth)
- void glClearIndex(GLfloat c)
- void glClearStencil(GLint s)
- void glClientActiveTexture(GLenum texture)
- void glClipPlane(GLenum plane, const GLdouble * equation)

- `void glColor3b(GLbyte red, GLbyte green, GLbyte blue)`
- `void glColor3s(GLshort red, GLshort green, GLshort blue)`
- `void glColor3i(GLint red, GLint green, GLint blue)`
- `void glColor3f(GLfloat red, GLfloat green, GLfloat blue)`
- `void glColor3d(GLdouble red, GLdouble green, GLdouble blue)`
- `void glColor3ub(GLubyte red, GLubyte green, GLubyte blue)`
- `void glColor3us(GLushort red, GLushort green, GLushort blue)`
- `void glColor3ui(GLuint red, GLuint green, GLuint blue)`
- `void glColor4b(GLbyte red, GLbyte green, GLbyte blue, GLbyte alpha)`
- `void glColor4s(GLshort red, GLshort green, GLshort blue, GLshort alpha)`
- `void glColor4i(GLint red, GLint green, GLint blue, GLint alpha)`
- `void glColor4f(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)`
- `void glColor4d(GLdouble red, GLdouble green, GLdouble blue, GLdouble alpha)`
- `void glColor4ub(GLubyte red, GLubyte green, GLubyte blue, GLubyte alpha)`
- `void glColor4us(GLushort red, GLushort green, GLushort blue, GLushort alpha)`
- `void glColor4ui(GLuint red, GLuint green, GLuint blue, GLuint alpha)`
- `void glColor3bv(const GLbyte * v)`
- `void glColor3sv(const GLshort * v)`
- `void glColor3iv(const GLint * v)`
- `void glColor3fv(const GLfloat * v)`
- `void glColor3dv(const GLdouble * v)`
- `void glColor3ubv(const GLubyte * v)`
- `void glColor3usv(const GLushort * v)`
- `void glColor3uiv(const GLuint * v)`
- `void glColor4bv(const GLbyte * v)`
- `void glColor4sv(const GLshort * v)`
- `void glColor4iv(const GLint * v)`
- `void glColor4fv(const GLfloat * v)`
- `void glColor4dv(const GLdouble * v)`
- `void glColor4ubv(const GLubyte * v)`
- `void glColor4usv(const GLushort * v)`
- `void glColor4uiv(const GLuint * v)`
- `void glColorMask(GLboolean red, GLboolean green, GLboolean blue, GLboolean alpha)`
- `void glColorMaterial(GLenum face, GLenum mode)`
- `void glColorPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)`

- void glColorSubTable(GLenum target, GLsizei start, GLsizei count, GLenum format, GLenum type, const GLvoid * data)
- void glColorTable(GLenum target, GLenum internalformat, GLsizei width, GLenum format, GLenum type, const GLvoid * data)
- void glColorTableParameterfv(GLenum target, GLenum pname, const GLfloat * params)
- void glColorTableParameteriv(GLenum target, GLenum pname, const GLint * params)
- void glCompileShader(GLuint shader)
- void glCompressedTexImage1D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLint border, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexImage2D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLint border, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexImage3D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLsizei depth, GLint border, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLsizei imageSize, const GLvoid * data)
- void glConvolutionFilter1D(GLenum target, GLenum internalformat, GLsizei width, GLenum format, GLenum type, const GLvoid * data)
- void glConvolutionFilter2D(GLenum target, GLenum internalformat, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)
- void glConvolutionParameterf(GLenum target, GLenum pname, GLfloat params)
- void glConvolutionParameteri(GLenum target, GLenum pname, GLint params)
- void glConvolutionParameterfv(GLenum target, GLenum pname, const GLfloat * params)
- void glConvolutionParameteriv(GLenum target, GLenum pname, const GLint * params)
- void glCopyColorSubTable(GLenum target, GLsizei start, GLint x, GLint y, GLsizei width)
- void glCopyColorTable(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width)
- void glCopyConvolutionFilter1D(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width)
- void glCopyConvolutionFilter2D(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width, GLsizei height)
- void glCopyPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum type)
- void glCopyTexImage1D(GLenum target, GLint level, GLenum internalformat, GLint x, GLint y, GLsizei width, GLint border)
- void glCopyTexImage2D(GLenum target, GLint level, GLenum internalformat, GLint x, GLint y, GLsizei width, GLsizei height, GLint border)
- void glCopyTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLint x, GLint y, GLsizei width)
- void glCopyTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint x, GLint y, GLsizei width, GLsizei height)

- `void glCopyTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLint x, GLint y, GLsizei width, GLsizei height)`
- `GLuint glCreateProgram(void)`
- `GLuint glCreateShader(GLenum shaderType)`
- `void glCullFace(GLenum mode)`
- `void glDeleteBuffers(GLsizei n, const GLuint * buffers)`
- `void glDeleteLists(GLuint list, GLsizei range)`
- `void glDeleteProgram(GLuint program)`
- `void glDeleteQueries(GLsizei n, const GLuint * ids)`
- `void glDeleteShader(GLuint shader)`
- `void glDeleteTextures(GLsizei n, const GLuint * textures)`
- `void glDepthFunc(GLenum func)`
- `void glDepthMask(GLboolean flag)`
- `void glDepthRange(GLclampd nearVal, GLclampd farVal)`
- `void glDetachShader(GLuint program, GLuint shader)`
- `void glEnable(GLenum cap)`
- `void glEnableClientState(GLenum cap)`
- `void glEnableVertexAttribArray(GLuint index)`
- `void glDisableVertexAttribArray(GLuint index)`
- `void glDrawArrays(GLenum mode, GLint first, GLsizei count)`
- `void glDrawBuffer(GLenum mode)`
- `void glDrawBuffers(GLsizei n, const GLenum *bufs)`
- `void glDrawElements(GLenum mode, GLsizei count, GLenum type, const GLvoid * indices)`
- `void glDrawPixels(GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)`
- `void glDrawRangeElements(GLenum mode, GLuint start, GLuint end, GLsizei count, GLenum type, const GLvoid * indices)`
- `void glEdgeFlag(GLboolean flag)`
- `void glEdgeFlagPointer(GLsizei stride, const GLvoid * pointer)`
- `void glEnd(void)`
- `void glEndList(void)`
- `void glEndQuery(GLenum target)`
- `void glEvalCoord1f(GLfloat u)`
- `void glEvalCoord1d(GLdouble u)`
- `void glEvalCoord2f(GLfloat u, GLfloat v)`
- `void glEvalCoord2d(GLdouble u, GLdouble v)`
- `void glEvalMesh1(GLenum mode, GLint i1, GLint i2)`
- `void glEvalPoint1(GLint i)`

- void glEvalPoint2(GLint i,GLint j)
- void glFeedbackBuffer(GLsizei size,GLenum type,GLfloat * buffer)
- void glFinish(void)
- void glFlush(void)
- void glFogf(GLenum pname,GLfloat param)
- void glFogi(GLenum pname,GLint param)
- void glFogfv(GLenum pname,const GLfloat * params)
- void glFogiv(GLenum pname,const GLint * params)
- void glFogCoordd(GLdouble coord)
- void glFogCoordf(GLfloat coord)
- void glFogCoorddv(GLdouble * coord)
- void glFogCoordfv(GLfloat * coord)
- void glFogCoordPointer(GLenum type,GLsizei stride,GLvoid * pointer)
- void glFrontFace(GLenum mode)
- void glFrustum(GLdouble left,GLdouble right,GLdouble bottom,GLdouble top,GLdouble nearVal,GLdouble farVal)
- void glGenBuffers(GLsizei n,GLuint * buffers)
- GLuint glGenLists(GLsizei range)
- void glGenQueries(GLsizei n,GLuint * ids)
- void glGenTextures(GLsizei n,GLuint * textures)
- void glGetBooleany(GLenum pname,GLboolean * params)
- void glGetDoublev(GLenum pname,GLdouble * params)
- void glGetFloatv(GLenum pname,GLfloat * params)
- void glGetIntegerv(GLenum pname,GLint * params)
- void glGetActiveAttrib(GLuint program,GLuint index,GLsizei bufSize,GLsizei *length,GLint *size,GLenum *type,GLchar *name)
- void glGetActiveUniform(GLuint program,GLuint index,GLsizei bufSize,GLsizei *length,GLint *size,GLenum *type,GLchar *name)
- void glGetAttachedShaders(GLuint program,GLsizei maxCount,GLsizei *count,GLuint *shaders)
- GLint glGetAttribLocation(GLuint program,const GLchar *name)
- void glGetBufferParameteriv(GLenum target,GLenum value,GLint * data)
- void glGetBufferPointerv(GLenum target,GLenum pname,GLvoid ** params)
- void glGetBufferSubData(GLenum target,GLintptr offset,GLsizeiptr size,GLvoid * data)
- void glGetClipPlane(GLenum plane,GLdouble * equation)
- void glGetColorTable(GLenum target,GLenum format,GLenum type,GLvoid * table)
- void glGetColorTableParameterfv(GLenum target,GLenum pname,GLfloat * params)
- void glGetColorTableParameteriv(GLenum target,GLenum pname,GLint * params)

- void glGetCompressedTexImage(GLenum target, GLint lod, GLvoid * img)
- void glGetConvolutionFilter(GLenum target, GLenum format, GLenum type, GLvoid * image)
- void glGetConvolutionParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetConvolutionParameteriv(GLenum target, GLenum pname, GLint * params)
- GLenum glGetError(void)
- void glGetHistogram(GLenum target, GLboolean reset, GLenum format, GLenum type, GLvoid * values)
- void glGetHistogramParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetHistogramParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetLightfv(GLenum light, GLenum pname, GLfloat * params)
- void glGetLightiv(GLenum light, GLenum pname, GLint * params)
- void glGetMapdv(GLenum target, GLenum query, GLdouble * v)
- void glGetMapfv(GLenum target, GLenum query, GLfloat * v)
- void glGetMapiv(GLenum target, GLenum query, GLint * v)
- void glGetMaterialfv(GLenum face, GLenum pname, GLfloat * params)
- void glGetMaterialiv(GLenum face, GLenum pname, GLint * params)
- void glGetMinmax(GLenum target, GLboolean reset, GLenum format, GLenum types, GLvoid * values)
- void glGetMinmaxParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetMinmaxParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetPixelMapfv(GLenum map, GLfloat * data)
- void glGetPixelMapuiv(GLenum map, GLuint * data)
- void glGetPixelMapusv(GLenum map, GLushort * data)
- void glGetPointerv(GLenum pname, GLvoid ** params)
- void glGetPolygonStipple(GLubyte * pattern)
- void glGetProgramiv(GLuint program, GLenum pname, GLint *params)
- void glGetProgramInfoLog(GLuint program, GLsizei maxLength, GLsizei *length, GLchar *infoLog)
- void glGetQueryObjectiv(GLuint id, GLenum pname, GLint * params)
- void glGetQueryObjectuiv(GLuint id, GLenum pname, GLuint * params)
- void glGetQueryiv(GLenum target, GLenum pname, GLint * params)
- void glGetSeparableFilter(GLenum target, GLenum format, GLenum type, GLvoid * row, GLvoid * column, GLvoid * span)
- void glGetShaderiv(GLuint shader, GLenum pname, GLint *params)
- void glGetShaderInfoLog(GLuint shader, GLsizei maxLength, GLsizei *length, GLchar *infoLog)
- void glGetShaderSource(GLuint shader, GLsizei bufSize, GLsizei *length, GLchar *source)
- const GLubyte* getString(GLenum name)
- void glGetTexEnvfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetTexEnviv(GLenum target, GLenum pname, GLint * params)

- void glGetTexGendv(GLenum coord, GLenum pname, GLdouble * params)
- void glGetTexGenfv(GLenum coord, GLenum pname, GLfloat * params)
- void glGetTexGeniv(GLenum coord, GLenum pname, GLint * params)
- void glGetTexImage(GLenum target, GLint level, GLenum format, GLenum type, GLvoid * img)
- void glGetTexLevelParameterfv(GLenum target, GLint level, GLenum pname, GLfloat * params)
- void glGetTexLevelParameteriv(GLenum target, GLint level, GLenum pname, GLint * params)
- void glGetTexParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetTexParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetUniformfv(GLuint program, GLint location, GLfloat *params)
- void glGetUniformiv(GLuint program, GLint location, GLint *params)
- GLint glGetUniformLocation(GLuint program, const GLchar *name)
- void glGetVertexAttribdv(GLuint index, GLenum pname, GLdouble *params)
- void glGetVertexAttribfv(GLuint index, GLenum pname, GLfloat *params)
- void glGetVertexAttribiv(GLuint index, GLenum pname, GLint *params)
- void glGetVertexAttribPointerv(GLuint index, GLenum pname, GLvoid **pointer)
- void glHint(GLenum target, GLenum mode)
- void glHistogram(GLenum target, GLsizei width, GLenum internalformat, GLboolean sink)
- void glIndexs(GLshort c)
- void glIndexi(GLint c)
- void glIndexf(GLfloat c)
- void glIndexd(GLdouble c)
- void glIndexub(GLubyte c)
- void glIndexsv(const GLshort * c)
- void glIndexiv(const GLint * c)
- void glIndexfv(const GLfloat * c)
- void glIndexdv(const GLdouble * c)
- void glIndexubv(const GLubyte * c)
- void glIndexMask(GLuint mask)
- void glIndexPointer(GLenum type, GLsizei stride, const GLvoid * pointer)
- void glInitNames(void)
- void glInterleavedArrays(GLenum format, GLsizei stride, const GLvoid * pointer)
- GLboolean glIsBuffer(GLuint buffer)
- GLboolean glIsEnabled(GLenum cap)
- GLboolean glIsList(GLuint list)
- GLboolean glIsProgram(GLuint program)
- GLboolean glIsQuery(GLuint id)

- GLboolean glIsShader(GLuint shader)
- GLboolean glIsTexture(GLuint texture)
- void glLightf(GLenum light, GLenum pname, GLfloat param)
- void glLighti(GLenum light, GLenum pname, GLint param)
- void glLightfv(GLenum light, GLenum pname, const GLfloat * params)
- void glLightiv(GLenum light, GLenum pname, const GLint * params)
- void glLightModelf(GLenum pname, GLfloat param)
- void glLightModeli(GLenum pname, GLint param)
- void glLightModelfv(GLenum pname, const GLfloat * params)
- void glLightModeliv(GLenum pname, const GLint * params)
- void glLineStipple(GLint factor, GLushort pattern)
- void glLineWidth(GLfloat width)
- void glLinkProgram(GLuint program)
- void glListBase(GLuint base)
- void glLoadIdentity(void)
- void glLoadMatrixd(const GLdouble * m)
- void glLoadMatrixf(const GLfloat * m)
- void glLoadName(GLuint name)
- void glLoadTransposeMatrixd(const GLdouble * m)
- void glLoadTransposeMatrixf(const GLfloat * m)
- void glLogicOp(GLenum opcode)
- void glMap1f(GLenum target, GLfloat u1, GLfloat u2, GLint stride, GLint order, const GLfloat * points)
- void glMap1d(GLenum target, GLdouble u1, GLdouble u2, GLint stride, GLint order, const GLdouble * points)
- void glMap2f(GLenum target, GLfloat u1, GLfloat u2, GLint ustride, GLint uorder, GLfloat v1, GLfloat v2, GLint vstride, GLint vorder, const GLfloat * points)
- void glMap2d(GLenum target, GLdouble u1, GLdouble u2, GLint ustride, GLint uorder, GLdouble v1, GLdouble v2, GLint vstride, GLint vorder, const GLdouble * points)
- void * glMapBuffer(GLenum target, GLenum access)
- void glMapGrid1d(GLint un, GLdouble u1, GLdouble u2)
- void glMapGrid1f(GLint un, GLfloat u1, GLfloat u2)
- void glMapGrid2d(GLint un, GLdouble u1, GLdouble u2, GLint vn, GLdouble v1, GLdouble v2)
- void glMapGrid2f(GLint un, GLfloat u1, GLfloat u2, GLint vn, GLfloat v1, GLfloat v2)
- void glMaterialf(GLenum face, GLenum pname, GLfloat param)
- void glMateriali(GLenum face, GLenum pname, GLint param)
- void glMatrixMode(GLenum mode)
- void glMinmax(GLenum target, GLenum internalformat, GLboolean sink)
- void glMultMatrixd(const GLdouble * m)

- `void glMultMatrixf(const GLfloat * m)`
- `void glMultTransposeMatrixd(const GLdouble * m)`
- `void glMultTransposeMatrixf(const GLfloat * m)`
- `void glMultiDrawArrays(GLenum mode, GLint * first, GLsizei * count, GLsizei primcount)`
- `void glMultiDrawElements(GLenum mode, const GLsizei * count, GLenum type, const GLvoid ** indices, GLsizei primcount)`
- `void glMultiTexCoord1s(GLenum target, GLshort s)`
- `void glMultiTexCoord1i(GLenum target, GLint s)`
- `void glMultiTexCoord1f(GLenum target, GLfloat s)`
- `void glMultiTexCoord1d(GLenum target, GLdouble s)`
- `void glMultiTexCoord2s(GLenum target, GLshort s, GLshort t)`
- `void glMultiTexCoord2i(GLenum target, GLint s, GLint t)`
- `void glMultiTexCoord2f(GLenum target, GLfloat s, GLfloat t)`
- `void glMultiTexCoord2d(GLenum target, GLdouble s, GLdouble t)`
- `void glMultiTexCoord3s(GLenum target, GLshort s, GLshort t, GLshort r)`
- `void glMultiTexCoord3i(GLenum target, GLint s, GLint t, GLint r)`
- `void glMultiTexCoord3f(GLenum target, GLfloat s, GLfloat t, GLfloat r)`
- `void glMultiTexCoord3d(GLenum target, GLdouble s, GLdouble t, GLdouble r)`
- `void glMultiTexCoord4s(GLenum target, GLshort s, GLshort t, GLshort r, GLshort q)`
- `void glMultiTexCoord4i(GLenum target, GLint s, GLint t, GLint r, GLint q)`
- `void glMultiTexCoord4f(GLenum target, GLfloat s, GLfloat t, GLfloat r, GLfloat q)`
- `void glMultiTexCoord4d(GLenum target, GLdouble s, GLdouble t, GLdouble r, GLdouble q)`
- `void glMultiTexCoord1sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord1iv(GLenum target, const GLint * v)`
- `void glMultiTexCoord1fv(GLenum target, const GLfloat * v)`
- `void glMultiTexCoord1dv(GLenum target, const GLdouble * v)`
- `void glMultiTexCoord2sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord2iv(GLenum target, const GLint * v)`
- `void glMultiTexCoord2fv(GLenum target, const GLfloat * v)`
- `void glMultiTexCoord2dv(GLenum target, const GLdouble * v)`
- `void glMultiTexCoord3sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord3iv(GLenum target, const GLint * v)`
- `void glMultiTexCoord3fv(GLenum target, const GLfloat * v)`
- `void glMultiTexCoord3dv(GLenum target, const GLdouble * v)`
- `void glMultiTexCoord4sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord4iv(GLenum target, const GLint * v)`

- `void glMultiTexCoord4fv(GLenum target,const GLfloat * v)`
- `void glMultiTexCoord4dv(GLenum target,const GLdouble * v)`
- `void glNewList(GLuint list,GLenum mode)`
- `void glNormal3b(GLbyte nx,GLbyte ny,GLbyte nz)`
- `void glNormal3d(GLdouble nx,GLdouble ny,GLdouble nz)`
- `void glNormal3f(GLfloat nx,GLfloat ny,GLfloat nz)`
- `void glNormal3i(GLint nx,GLint ny,GLint nz)`
- `void glNormal3s(GLshort nx,GLshort ny,GLshort nz)`
- `void glNormal3bv(const GLbyte * v)`
- `void glNormal3dv(const GLdouble * v)`
- `void glNormal3fv(const GLfloat * v)`
- `void glNormal3iv(const GLint * v)`
- `void glNormal3sv(const GLshort * v)`
- `void glNormalPointer(GLenum type,GLsizei stride,const GLvoid * pointer)`
- `void glOrtho(GLdouble left,GLdouble right,GLdouble bottom,GLdouble top,GLdouble nearVal,GLdouble farVal)`
- `void glPassThrough(GLfloat token)`
- `void glPixelMapfv(GLenum map,GLsizei mapsize,const GLfloat * values)`
- `void glPixelMapuiv(GLenum map,GLsizei mapsize,const GLuint * values)`
- `void glPixelMapusv(GLenum map,GLsizei mapsize,const GLushort * values)`
- `void glPixelStoref(GLenum pname,GLfloat param)`
- `void glPixelStorei(GLenum pname,GLint param)`
- `void glPixelTransferf(GLenum pname,GLfloat param)`
- `void glPixelTransferi(GLenum pname,GLint param)`
- `void glPixelZoom(GLfloat xfactor,GLfloat yfactor)`
- `void glPointParameterf(GLenum pname,GLfloat param)`
- `void glPointParameteri(GLenum pname,GLint param)`
- `void glPointSize(GLfloat size)`
- `void glPolygonMode(GLenum face,GLenum mode)`
- `void glPolygonOffset(GLfloat factor,GLfloat units)`
- `void glPolygonStipple(const GLubyte * pattern)`
- `void glPushAttrib(GLbitfield mask)`
- `void glPushClientAttrib(GLbitfield mask)`
- `void glPushMatrix(void)`
- `void glPushName(GLuint name)`
- `void glPrioritizeTextures(GLsizei n,const GLuint * textures,const GLclampf * priorities)`

- `void glPopMatrix(void)`
- `void glRasterPos2s(GLshort x, GLshort y)`
- `void glRasterPos2i(GLint x, GLint y)`
- `void glRasterPos2f(GLfloat x, GLfloat y)`
- `void glRasterPos2d(GLdouble x, GLdouble y)`
- `void glRasterPos3s(GLshort x, GLshort y, GLshort z)`
- `void glRasterPos3i(GLint x, GLint y, GLint z)`
- `void glRasterPos3f(GLfloat x, GLfloat y, GLfloat z)`
- `void glRasterPos3d(GLdouble x, GLdouble y, GLdouble z)`
- `void glRasterPos4s(GLshort x, GLshort y, GLshort z, GLshort w)`
- `void glRasterPos4i(GLint x, GLint y, GLint z, GLint w)`
- `void glRasterPos4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)`
- `void glRasterPos4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)`
- `void glReadBuffer(GLenum mode)`
- `void glReadPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum format, GLenum type, GLvoid * data)`
- `void glRectd(GLdouble x1, GLdouble y1, GLdouble x2, GLdouble y2)`
- `void glRectf(GLfloat x1, GLfloat y1, GLfloat x2, GLfloat y2)`
- `void glRecti(GLint x1, GLint y1, GLint x2, GLint y2)`
- `void glRects(GLshort x1, GLshort y1, GLshort x2, GLshort y2)`
- `void glRectdv(const GLdouble * v1, const GLdouble * v2)`
- `void glRectfv(const GLfloat * v1, const GLfloat * v2)`
- `void glRectiv(const GLint * v1, const GLint * v2)`
- `void glRectsv(const GLshort * v1, const GLshort * v2)`
- `GLint glRenderMode(GLenum mode)`
- `void glResetHistogram(GLenum target)`
- `void glRotated(GLdouble angle, GLdouble x, GLdouble y, GLdouble z)`
- `void glRotatef(GLfloat angle, GLfloat x, GLfloat y, GLfloat z)`
- `void glSampleCoverage(GLclampf value, GLboolean invert)`
- `void glScaled(GLdouble x, GLdouble y, GLdouble z)`
- `void glScalef(GLfloat x, GLfloat y, GLfloat z)`
- `void glScissor(GLint x, GLint y, GLsizei width, GLsizei height)`
- `void glSecondaryColor3b(GLbyte red, GLbyte green, GLbyte blue)`
- `void glSecondaryColor3s(GLshort red, GLshort green, GLshort blue)`
- `void glSecondaryColor3i(GLint red, GLint green, GLint blue)`
- `void glSecondaryColor3f(GLfloat red, GLfloat green, GLfloat blue)`
- `void glSecondaryColor3d(GLdouble red, GLdouble green, GLdouble blue)`

- void glSecondaryColor3ub(GLubyte red, GLubyte green, GLubyte blue)
- void glSecondaryColor3us(GLushort red, GLushort green, GLushort blue)
- void glSecondaryColor3ui(GLuint red, GLuint green, GLuint blue)
- void glSecondaryColor3bv(const GLbyte * v)
- void glSecondaryColor3sv(const GLshort * v)
- void glSecondaryColor3iv(const GLint * v)
- void glSecondaryColor3fv(const GLfloat * v)
- void glSecondaryColor3dv(const GLdouble * v)
- void glSecondaryColor3ubv(const GLubyte * v)
- void glSecondaryColor3usv(const GLushort * v)
- void glSecondaryColor3uiv(const GLuint * v)
- void glSecondaryColorPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)
- void glSelectBuffer(GLsizei size, GLuint * buffer)
- void glSeparableFilter2D(GLenum target, GLenum internalformat, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * row, const GLvoid * column)
- void glShadeModel(GLenum mode)
- void glShaderSource(GLuint shader, GLsizei count, const GLchar **string, const GLint *length)
- void glStencilFunc(GLenum func, GLint ref, GLuint mask)
- void glStencilFuncSeparate(GLenum face, GLenum func, GLint ref, GLuint mask)
- void glStencilMask(GLuint mask)
- void glStencilMaskSeparate(GLenum face, GLuint mask)
- void glStencilOp(GLenum sfail, GLenum dpfail, GLenum dppass)
- void glStencilOpSeparate(GLenum face, GLenum sfail, GLenum dpfail, GLenum dppass)
- void glTexCoord1s(GLshort s)
- void glTexCoord1i(GLint s)
- void glTexCoord1f(GLfloat s)
- void glTexCoord1d(GLdouble s)
- void glTexCoord2s(GLshort s, GLshort t)
- void glTexCoord2i(GLint s, GLint t)
- void glTexCoord2f(GLfloat s, GLfloat t)
- void glTexCoord2d(GLdouble s, GLdouble t)
- void glTexCoord3s(GLshort s, GLshort t, GLshort r)
- void glTexCoord3i(GLint s, GLint t, GLint r)
- void glTexCoord3f(GLfloat s, GLfloat t, GLfloat r)
- void glTexCoord3d(GLdouble s, GLdouble t, GLdouble r)
- void glTexCoord4s(GLshort s, GLshort t, GLshort r, GLshort q)

- `void glTexCoord4i(GLint s,GLint t,GLint r,GLint q)`
- `void glTexCoord4f(GLfloat s,GLfloat t,GLfloat r,GLfloat q)`
- `void glTexCoord4d(GLdouble s,GLdouble t,GLdouble r,GLdouble q)`
- `void glTexCoord1sv(const GLshort * v)`
- `void glTexCoord1iv(const GLint * v)`
- `void glTexCoord1fv(const GLfloat * v)`
- `void glTexCoord1dv(const GLdouble * v)`
- `void glTexCoord2sv(const GLshort * v)`
- `void glTexCoord2iv(const GLint * v)`
- `void glTexCoord2fv(const GLfloat * v)`
- `void glTexCoord2dv(const GLdouble * v)`
- `void glTexCoord3sv(const GLshort * v)`
- `void glTexCoord3iv(const GLint * v)`
- `void glTexCoord3fv(const GLfloat * v)`
- `void glTexCoord3dv(const GLdouble * v)`
- `void glTexCoord4sv(const GLshort * v)`
- `void glTexCoord4iv(const GLint * v)`
- `void glTexCoord4fv(const GLfloat * v)`
- `void glTexCoord4dv(const GLdouble * v)`
- `void glTexCoordPointer(GLint size,GLenum type,GLsizei stride,const GLvoid * pointer)`
- `void glTexEnvf(GLenum target,GLenum pname,GLfloat param)`
- `void glTexEnvi(GLenum target,GLenum pname,GLint param)`
- `void glTexGeni(GLenum coord,GLenum pname,GLint param)`
- `void glTexGenf(GLenum coord,GLenum pname,GLfloat param)`
- `void glTexGend(GLenum coord,GLenum pname,GLdouble param)`
- `void glTexGeniv(GLenum coord,GLenum pname,const GLint * params)`
- `void glTexGenfv(GLenum coord,GLenum pname,const GLfloat * params)`
- `void glTexGendv(GLenum coord,GLenum pname,const GLdouble * params)`
- `void glTexImage1D(GLenum target,GLint level,GLint internalFormat,GLsizei width,GLint border,GLenum format,GLenum type,const GLvoid * data)`
- `void glTexImage2D(GLenum target,GLint level,GLint internalFormat,GLsizei width,GLsizei height,GLint border,GLenum format,GLenum type,const GLvoid * data)`
- `void glTexImage3D(GLenum target,GLint level,GLint internalFormat,GLsizei width,GLsizei height,GLsizei depth,GLint border,GLenum format,GLenum type,const GLvoid * data)`
- `void glTexParameterf(GLenum target,GLenum pname,GLfloat param)`
- `void glTexParameteri(GLenum target,GLenum pname,GLint param)`
- `void glTexParameterfv(GLenum target,GLenum pname,const GLfloat * params)`

- void glTexParameteriv(GLenum target, GLenum pname, const GLint * params)
- void glTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLenum type, const GLvoid * data)
- void glTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)
- void glTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLenum type, const GLvoid * data)
- void glTranslated(GLdouble x, GLdouble y, GLdouble z)
- void glTranslatef(GLfloat x, GLfloat y, GLfloat z)
- void glUniform1f(GLint location, GLfloat v0)
- void glUniform2f(GLint location, GLfloat v0, GLfloat v1)
- void glUniform3f(GLint location, GLfloat v0, GLfloat v1, GLfloat v2)
- void glUniform4f(GLint location, GLfloat v0, GLfloat v1, GLfloat v2, GLfloat v3)
- void glUniform1i(GLint location, GLint v0)
- void glUniform2i(GLint location, GLint v0, GLint v1)
- void glUniform3i(GLint location, GLint v0, GLint v1, GLint v2)
- void glUniform4i(GLint location, GLint v0, GLint v1, GLint v2, GLint v3)
- void glUniform1fv(GLint location, GLsizei count, const GLfloat *value)
- void glUniform2fv(GLint location, GLsizei count, const GLfloat *value)
- void glUniform3fv(GLint location, GLsizei count, const GLfloat *value)
- void glUniform4fv(GLint location, GLsizei count, const GLfloat *value)
- void glUniform1iv(GLint location, GLsizei count, const GLint *value)
- void glUniform2iv(GLint location, GLsizei count, const GLint *value)
- void glUniform3iv(GLint location, GLsizei count, const GLint *value)
- void glUniform4iv(GLint location, GLsizei count, const GLint *value)
- void glUniformMatrix2fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)
- void glUniformMatrix3fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)
- void glUniformMatrix4fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)
- void glUniformMatrix2x3fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)
- void glUniformMatrix3x2fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)
- void glUniformMatrix2x4fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)
- void glUniformMatrix4x2fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)
- void glUniformMatrix3x4fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)
- void glUniformMatrix4x3fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)
- void glUseProgram(GLuint program)
- void glValidateProgram(GLuint program)
- void glVertex2s(GLshort x, GLshort y)

- `void glVertex2i(GLint x, GLint y)`
- `void glVertex2f(GLfloat x, GLfloat y)`
- `void glVertex2d(GLdouble x, GLdouble y)`
- `void glVertex3s(GLshort x, GLshort y, GLshort z)`
- `void glVertex3i(GLint x, GLint y, GLint z)`
- `void glVertex3f(GLfloat x, GLfloat y, GLfloat z)`
- `void glVertex3d(GLdouble x, GLdouble y, GLdouble z)`
- `void glVertex4s(GLshort x, GLshort y, GLshort z, GLshort w)`
- `void glVertex4i(GLint x, GLint y, GLint z, GLint w)`
- `void glVertex4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)`
- `void glVertex4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)`
- `void glVertex2sv(const GLshort * v)`
- `void glVertex2iv(const GLint * v)`
- `void glVertex2fv(const GLfloat * v)`
- `void glVertex2dv(const GLdouble * v)`
- `void glVertex3sv(const GLshort * v)`
- `void glVertex3iv(const GLint * v)`
- `void glVertex3fv(const GLfloat * v)`
- `void glVertex3dv(const GLdouble * v)`
- `void glVertex4sv(const GLshort * v)`
- `void glVertex4iv(const GLint * v)`
- `void glVertex4fv(const GLfloat * v)`
- `void glVertex4dv(const GLdouble * v)`
- `void glVertexAttrib1f(GLuint index, GLfloat v0)`
- `void glVertexAttrib1s(GLuint index, GLshort v0)`
- `void glVertexAttrib1d(GLuint index, GLdouble v0)`
- `void glVertexAttrib2f(GLuint index, GLfloat v0, GLfloat v1)`
- `void glVertexAttrib2s(GLuint index, GLshort v0, GLshort v1)`
- `void glVertexAttrib2d(GLuint index, GLdouble v0, GLdouble v1)`
- `void glVertexAttrib3f(GLuint index, GLfloat v0, GLfloat v1, GLfloat v2)`
- `void glVertexAttrib3s(GLuint index, GLshort v0, GLshort v1, GLshort v2)`
- `void glVertexAttrib3d(GLuint index, GLdouble v0, GLdouble v1, GLdouble v2)`
- `void glVertexAttrib4f(GLuint index, GLfloat v0, GLfloat v1, GLfloat v2, GLfloat v3)`
- `void glVertexAttrib4s(GLuint index, GLshort v0, GLshort v1, GLshort v2, GLshort v3)`
- `void glVertexAttrib4d(GLuint index, GLdouble v0, GLdouble v1, GLdouble v2, GLdouble v3)`
- `void glVertexAttrib4Nub(GLuint index, GLubyte v0, GLubyte v1, GLubyte v2, GLubyte v3)`

- void glVertexAttrib1fv(GLuint index,const GLfloat *v)
- void glVertexAttrib1sv(GLuint index,const GLshort *v)
- void glVertexAttrib1dv(GLuint index,const GLdouble *v)
- void glVertexAttrib2fv(GLuint index,const GLfloat *v)
- void glVertexAttrib2sv(GLuint index,const GLshort *v)
- void glVertexAttrib2dv(GLuint index,const GLdouble *v)
- void glVertexAttrib3fv(GLuint index,const GLfloat *v)
- void glVertexAttrib3sv(GLuint index,const GLshort *v)
- void glVertexAttrib3dv(GLuint index,const GLdouble *v)
- void glVertexAttrib4fv(GLuint index,const GLfloat *v)
- void glVertexAttrib4sv(GLuint index,const GLshort *v)
- void glVertexAttrib4dv(GLuint index,const GLdouble *v)
- void glVertexAttrib4iv(GLuint index,const GLint *v)
- void glVertexAttrib4bv(GLuint index,const GLbyte *v)
- void glVertexAttrib4ubv(GLuint index,const GLubyte *v)
- void glVertexAttrib4usv(GLuint index,const GLushort *v)
- void glVertexAttrib4uiv(GLuint index,const GLuint *v)
- void glVertexAttribPointer(GLuint index,GLint size,GLenum type,GLboolean normalized,GLsizei stride,const GLvoid * pointer)
- void glVertexPointer(GLint size,GLenum type,GLsizei stride,const GLvoid * pointer)
- void glViewport(GLint x,GLint y,GLsizei width,GLsizei height)
- void glWindowPos2s(GLshort x,GLshort y)
- void glWindowPos2i(GLint x,GLint y)
- void glWindowPos2f(GLfloat x,GLfloat y)
- void glWindowPos2d(GLdouble x,GLdouble y)
- void glWindowPos3s(GLshort x,GLshort y,GLshort z)
- void glWindowPos3i(GLint x,GLint y,GLint z)
- void glWindowPos3f(GLfloat x,GLfloat y,GLfloat z)
- void glWindowPos3d(GLdouble x,GLdouble y,GLdouble z)
- void glWindowPos2sv(const GLshort * v)
- void glWindowPos2iv(const GLint * v)
- void glWindowPos2fv(const GLfloat * v)
- void glWindowPos2dv(const GLdouble * v)
- void glWindowPos3sv(const GLshort * v)
- void glWindowPos3iv(const GLint * v)
- void glWindowPos3fv(const GLfloat * v)

- void glWindowPos3dv(const GLdouble * v)
- void gluBeginCurve(GLUnurbs* nurb)
- void gluBeginPolygon(GLUtesselator* tess)
- void gluBeginSurface(GLUnurbs* nurb)
- void gluBeginTrim(GLUnurbs* nurb)
- void gluCylinder(GLUquadric* quad, GLdouble base, GLdouble top, GLdouble height, GLint slices, GLint stacks)
- void gluDeleteNurbsRenderer(GLUnurbs* nurb)
- void gluDeleteQuadric(GLUquadric* quad)
- void gluDeleteTess(GLUtesselator* tess)
- void gluDisk(GLUquadric* quad, GLdouble inner, GLdouble outer, GLint slices, GLint loops)
- void gluEndCurve(GLUnurbs* nurb)
- void gluEndPolygon(GLUtesselator* tess)
- void gluEndSurface(GLUnurbs* nurb)
- void gluEndTrim(GLUnurbs* nurb)
- const GLubyte * gluErrorString(GLenum error)
- void gluGetNurbsProperty(GLUnurbs* nurb, GLenum property, GLfloat* data)
- const GLubyte * gluGetString(GLenum name)
- void gluGetTessProperty(GLUtesselator* tess, GLenum which, GLdouble* data)
- void gluLoadSamplingMatrices(GLUnurbs* nurb, const GLfloat * model, const GLfloat * perspective, const GLint * view)
- void gluLookAt(GLdouble eyeX, GLdouble eyeY, GLdouble eyeZ, GLdouble centerX, GLdouble centerY, GLdouble centerZ, GLdouble upX, GLdouble upY, GLdouble upZ)
- GLUnurbs *gluNewNurbsRenderer(void)
- GLUquadric *gluNewQuadric(void)
- GLUtesselator* gluNewTess(void)
- void gluNextContour(GLUtesselator* tess, GLenum type)
- void gluNurbsCurve(GLUnurbs* nurb, GLint knotCount, GLfloat * knots, GLint stride, GLfloat * control, GLint order, GLenum type)
- void gluNurbsProperty(GLUnurbs* nurb, GLenum property, GLfloat value)
- void gluNurbsSurface(GLUnurbs* nurb, GLint sKnotCount, GLfloat* sKnots, GLint tKnotCount, GLfloat* tKnots, GLint sStride, GLint tStride, GLfloat* control, GLint sOrder, GLint tOrder, GLenum type)
- void gluOrtho2D(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top)
- void gluPartialDisk(GLUquadric* quad, GLdouble inner, GLdouble outer, GLint slices, GLint loops, GLdouble start, GLdouble sweep)
- void gluPerspective(GLdouble fovy, GLdouble aspect, GLdouble zNear, GLdouble zFar)
- void gluPickMatrix(GLdouble x, GLdouble y, GLdouble delX, GLdouble delY, GLint * viewport)
- GLint gluProject(GLdouble objX, GLdouble objY, GLdouble objZ, const GLdouble * model, const GLdouble * proj, const GLint * view, GLdouble* winX, GLdouble* winY, GLdouble* winZ)

- void gluPwlCurve(GLUnurbs* nurb,GLint count,GLfloat* data,GLint stride,GLenum type)
- void gluQuadricDrawStyle(GLUquadric* quad,GLenum draw)
- void gluQuadricNormals(GLUquadric* quad,GLenum normal)
- void gluQuadricOrientation(GLUquadric* quad,GLenum orientation)
- void gluQuadricTexture(GLUquadric* quad,GLboolean texture)
- GLint gluScaleImage(GLenum format,GLsizei wIn,GLsizei hIn,GLenum typeIn,const void * dataIn,GLsizei wOut,GLsizei hOut,GLenum typeOut,GLvoid* dataOut)
- void gluSphere(GLUquadric* quad,GLdouble radius,GLint slices,GLint stacks)
- void gluTessBeginContour(GLUtesselator* tess)
- void gluTessBeginPolygon(GLUtesselator* tess,GLvoid* data)
- void gluTessEndContour(GLUtesselator* tess)
- void gluTessEndPolygon(GLUtesselator* tess)
- void gluTessNormal(GLUtesselator* tess,GLdouble valueX,GLdouble valueY,GLdouble valueZ)
- void gluTessProperty(GLUtesselator* tess,GLenum which,GLdouble data)
- void gluTessVertex(GLUtesselator* tess,GLdouble * location,GLvoid* data)
- GLint gluUnProject(GLdouble winX,GLdouble winY,GLdouble winZ,const GLdouble * model,const GLdouble * proj,const GLint * view,GLdouble* objX,GLdouble* objY,GLdouble* objZ)
- void glDisable(GLenum cap)

RINGOPENGL (OPENGL 3.2) FUNCTIONS REFERENCE

- GL_ZERO
- GL_FALSE
- GL_LOGIC_OP
- GL_NONE
- GL_TEXTURE_COMPONENTS
- GL_NO_ERROR
- GL_POINTS
- GL_CURRENT_BIT
- GL_TRUE
- GL_ONE
- GL_CLIENT_PIXEL_STORE_BIT
- GL_LINES
- GL_LINE_LOOP
- GL_POINT_BIT
- GL_CLIENT_VERTEX_ARRAY_BIT
- GL_LINE_STRIP
- GL_LINE_BIT
- GL_TRIANGLES
- GL_TRIANGLE_STRIP
- GL_TRIANGLE_FAN
- GL_QUADS
- GL_QUAD_STRIP
- GL_POLYGON_BIT
- GL_POLYGON
- GL_POLYGON_STIPPLE_BIT
- GL_PIXEL_MODE_BIT
- GL_LIGHTING_BIT

- GL_FOG_BIT
- GL_DEPTH_BUFFER_BIT
- GL_ACCUM
- GL_LOAD
- GL_RETURN
- GL_MULT
- GL_ADD
- GL_NEVER
- GL_ACCUM_BUFFER_BIT
- GL_LESS
- GL_EQUAL
- GL_LEQUAL
- GL_GREATER
- GL_NOTEQUAL
- GL_GEQUAL
- GL_ALWAYS
- GL_SRC_COLOR
- GL_ONE_MINUS_SRC_COLOR
- GL_SRC_ALPHA
- GL_ONE_MINUS_SRC_ALPHA
- GL_DST_ALPHA
- GL_ONE_MINUS_DST_ALPHA
- GL_DST_COLOR
- GL_ONE_MINUS_DST_COLOR
- GL_SRC_ALPHA_SATURATE
- GL_STENCIL_BUFFER_BIT
- GL_FRONT_LEFT
- GL_FRONT_RIGHT
- GL_BACK_LEFT
- GL_BACK_RIGHT
- GL_FRONT
- GL_BACK
- GL_LEFT
- GL_RIGHT
- GL_FRONT_AND_BACK
- GL_AUX0

- GL_AUX1
- GL_AUX2
- GL_AUX3
- GL_INVALID_ENUM
- GL_INVALID_VALUE
- GL_INVALID_OPERATION
- GL_STACK_OVERFLOW
- GL_STACK_UNDERFLOW
- GL_OUT_OF_MEMORY
- GL_2D
- GL_3D
- GL_3D_COLOR
- GL_3D_COLOR_TEXTURE
- GL_4D_COLOR_TEXTURE
- GL_PASS_THROUGH_TOKEN
- GL_POINT_TOKEN
- GL_LINE_TOKEN
- GL_POLYGON_TOKEN
- GL_BITMAP_TOKEN
- GL_DRAW_PIXEL_TOKEN
- GL_COPY_PIXEL_TOKEN
- GL_LINE_RESET_TOKEN
- GL_EXP
- GL_VIEWPORT_BIT
- GL_EXP2
- GL_CW
- GL_CCW
- GL_COEFF
- GL_ORDER
- GL_DOMAIN
- GL_CURRENT_COLOR
- GL_CURRENT_INDEX
- GL_CURRENT_NORMAL
- GL_CURRENT_TEXTURE_COORDS
- GL_CURRENT_RASTER_COLOR
- GL_CURRENT_RASTER_INDEX

- GL_CURRENT_RASTER_TEXTURE_COORDS
- GL_CURRENT_RASTER_POSITION
- GL_CURRENT_RASTER_POSITION_VALID
- GL_CURRENT_RASTER_DISTANCE
- GL_POINT_SMOOTH
- GL_POINT_SIZE
- GL_POINT_SIZE_RANGE
- GL_POINT_SIZE_GRANULARITY
- GL_LINE_SMOOTH
- GL_LINE_WIDTH
- GL_LINE_WIDTH_RANGE
- GL_LINE_WIDTH_GRANULARITY
- GL_LINE_STIPPLE
- GL_LINE_STIPPLE_PATTERN
- GL_LINE_STIPPLE_REPEAT
- GL_LIST_MODE
- GL_MAX_LIST_NESTING
- GL_LIST_BASE
- GL_LIST_INDEX
- GL_POLYGON_MODE
- GL_POLYGON_SMOOTH
- GL_POLYGON_STIPPLE
- GL_EDGE_FLAG
- GL_CULL_FACE
- GL_CULL_FACE_MODE
- GL_FRONT_FACE
- GL_LIGHTING
- GL_LIGHT_MODEL_LOCAL_VIEWER
- GL_LIGHT_MODEL_TWO_SIDE
- GL_LIGHT_MODEL_AMBIENT
- GL_SHADE_MODEL
- GL_COLOR_MATERIAL_FACE
- GL_COLOR_MATERIAL_PARAMETER
- GL_COLOR_MATERIAL
- GL_FOG
- GL_FOG_INDEX

- GL_FOG_DENSITY
- GL_FOG_START
- GL_FOG_END
- GL_FOG_MODE
- GL_FOG_COLOR
- GL_DEPTH_RANGE
- GL_DEPTH_TEST
- GL_DEPTH_WRITEMASK
- GL_DEPTH_CLEAR_VALUE
- GL_DEPTH_FUNC
- GL_ACCUM_CLEAR_VALUE
- GL_STENCIL_TEST
- GL_STENCIL_CLEAR_VALUE
- GL_STENCIL_FUNC
- GL_STENCIL_VALUE_MASK
- GL_STENCIL_FAIL
- GL_STENCIL_PASS_DEPTH_FAIL
- GL_STENCIL_PASS_DEPTH_PASS
- GL_STENCIL_REF
- GL_STENCIL_WRITEMASK
- GL_MATRIX_MODE
- GL_NORMALIZE
- GL_VIEWPORT
- GL_MODELVIEW_STACK_DEPTH
- GL_PROJECTION_STACK_DEPTH
- GL_TEXTURE_STACK_DEPTH
- GL_MODELVIEW_MATRIX
- GL_PROJECTION_MATRIX
- GL_TEXTURE_MATRIX
- GL_ATTRIB_STACK_DEPTH
- GL_CLIENT_ATTRIB_STACK_DEPTH
- GL_ALPHA_TEST
- GL_ALPHA_TEST_FUNC
- GL_ALPHA_TEST_REF
- GL_DITHER
- GL_BLEND_DST

- GL_BLEND_SRC
- GL_BLEND
- GL_LOGIC_OP_MODE
- GL_INDEX_LOGIC_OP
- GL_COLOR_LOGIC_OP
- GL_AUX_BUFFERS
- GL_DRAW_BUFFER
- GL_READ_BUFFER
- GL_SCISSOR_BOX
- GL_SCISSOR_TEST
- GL_INDEX_CLEAR_VALUE
- GL_INDEX_WRITEMASK
- GL_COLOR_CLEAR_VALUE
- GL_COLOR_WRITEMASK
- GL_INDEX_MODE
- GL_RGBA_MODE
- GL_DOUBLEBUFFER
- GL_STEREO
- GL_RENDER_MODE
- GL_PERSPECTIVE_CORRECTION_HINT
- GL_POINT_SMOOTH_HINT
- GL_LINE_SMOOTH_HINT
- GL_POLYGON_SMOOTH_HINT
- GL_FOG_HINT
- GL_TEXTURE_GEN_S
- GL_TEXTURE_GEN_T
- GL_TEXTURE_GEN_R
- GL_TEXTURE_GEN_Q
- GL_PIXEL_MAP_I_TO_I
- GL_PIXEL_MAP_S_TO_S
- GL_PIXEL_MAP_I_TO_R
- GL_PIXEL_MAP_I_TO_G
- GL_PIXEL_MAP_I_TO_B
- GL_PIXEL_MAP_I_TO_A
- GL_PIXEL_MAP_R_TO_R
- GL_PIXEL_MAP_G_TO_G

- GL_PIXEL_MAP_B_TO_B
- GL_PIXEL_MAP_A_TO_A
- GL_PIXEL_MAP_I_TO_I_SIZE
- GL_PIXEL_MAP_S_TO_S_SIZE
- GL_PIXEL_MAP_I_TO_R_SIZE
- GL_PIXEL_MAP_I_TO_G_SIZE
- GL_PIXEL_MAP_I_TO_B_SIZE
- GL_PIXEL_MAP_I_TO_A_SIZE
- GL_PIXEL_MAP_R_TO_R_SIZE
- GL_PIXEL_MAP_G_TO_G_SIZE
- GL_PIXEL_MAP_B_TO_B_SIZE
- GL_PIXEL_MAP_A_TO_A_SIZE
- GL_UNPACK_SWAP_BYTES
- GL_UNPACK_LSB_FIRST
- GL_UNPACK_ROW_LENGTH
- GL_UNPACK_SKIP_ROWS
- GL_UNPACK_SKIP_PIXELS
- GL_UNPACK_ALIGNMENT
- GL_PACK_SWAP_BYTES
- GL_PACK_LSB_FIRST
- GL_PACK_ROW_LENGTH
- GL_PACK_SKIP_ROWS
- GL_PACK_SKIP_PIXELS
- GL_PACK_ALIGNMENT
- GL_MAP_COLOR
- GL_MAP_STENCIL
- GL_INDEX_SHIFT
- GL_INDEX_OFFSET
- GL_RED_SCALE
- GL_RED_BIAS
- GL_ZOOM_X
- GL_ZOOM_Y
- GL_GREEN_SCALE
- GL_GREEN_BIAS
- GL_BLUE_SCALE
- GL_BLUE_BIAS

- GL_ALPHA_SCALE
- GL_ALPHA_BIAS
- GL_DEPTH_SCALE
- GL_DEPTH_BIAS
- GL_MAX_EVAL_ORDER
- GL_MAX_LIGHTS
- GL_MAX_CLIP_PLANES
- GL_MAX_TEXTURE_SIZE
- GL_MAX_PIXEL_MAP_TABLE
- GL_MAX_ATTRIB_STACK_DEPTH
- GL_MAX_MODELVIEW_STACK_DEPTH
- GL_MAX_NAME_STACK_DEPTH
- GL_MAX_PROJECTION_STACK_DEPTH
- GL_MAX_TEXTURE_STACK_DEPTH
- GL_MAX_VIEWPORT_DIMS
- GL_MAX_CLIENT_ATTRIB_STACK_DEPTH
- GL_SUBPIXEL_BITS
- GL_INDEX_BITS
- GL_RED_BITS
- GL_GREEN_BITS
- GL_BLUE_BITS
- GL_ALPHA_BITS
- GL_DEPTH_BITS
- GL_STENCIL_BITS
- GL_ACCUM_RED_BITS
- GL_ACCUM_GREEN_BITS
- GL_ACCUM_BLUE_BITS
- GL_ACCUM_ALPHA_BITS
- GL_NAME_STACK_DEPTH
- GL_AUTO_NORMAL
- GL_MAP1_COLOR_4
- GL_MAP1_INDEX
- GL_MAP1_NORMAL
- GL_MAP1_TEXTURE_COORD_1
- GL_MAP1_TEXTURE_COORD_2
- GL_MAP1_TEXTURE_COORD_3

- GL_MAP1_TEXTURE_COORD_4
- GL_MAP1_VERTEX_3
- GL_MAP1_VERTEX_4
- GL_MAP2_COLOR_4
- GL_MAP2_INDEX
- GL_MAP2_NORMAL
- GL_MAP2_TEXTURE_COORD_1
- GL_MAP2_TEXTURE_COORD_2
- GL_MAP2_TEXTURE_COORD_3
- GL_MAP2_TEXTURE_COORD_4
- GL_MAP2_VERTEX_3
- GL_MAP2_VERTEX_4
- GL_MAP1_GRID_DOMAIN
- GL_MAP1_GRID_SEGMENTS
- GL_MAP2_GRID_DOMAIN
- GL_MAP2_GRID_SEGMENTS
- GL_TEXTURE_1D
- GL_TEXTURE_2D
- GL_FEEDBACK_BUFFER_POINTER
- GL_FEEDBACK_BUFFER_SIZE
- GL_FEEDBACK_BUFFER_TYPE
- GL_SELECTION_BUFFER_POINTER
- GL_SELECTION_BUFFER_SIZE
- GL_TEXTURE_WIDTH
- GL_TRANSFORM_BIT
- GL_TEXTURE_HEIGHT
- GL_TEXTURE_INTERNAL_FORMAT
- GL_TEXTURE_BORDER_COLOR
- GL_TEXTURE_BORDER
- GL_DONT_CARE
- GL_FASTEST
- GL_NICEST
- GL_AMBIENT
- GL_DIFFUSE
- GL_SPECULAR
- GL_POSITION

- GL_SPOT_DIRECTION
- GL_SPOT_EXPONENT
- GL_SPOT_CUTOFF
- GL_CONSTANT_ATTENUATION
- GL_LINEAR_ATTENUATION
- GL_QUADRATIC_ATTENUATION
- GL_COMPILE
- GL_COMPILE_AND_EXECUTE
- GL_BYTE
- GL_UNSIGNED_BYTE
- GL_SHORT
- GL_UNSIGNED_SHORT
- GL_INT
- GL_UNSIGNED_INT
- GL_FLOAT
- GL_2_BYTES
- GL_3_BYTES
- GL_4_BYTES
- GL_DOUBLE
- GL_CLEAR
- GL_AND
- GL_AND_REVERSE
- GL_COPY
- GL_AND_INVERTED
- GL_NOOP
- GL_XOR
- GL_OR
- GL_NOR
- GL_EQUIV
- GL_INVERT
- GL_OR_REVERSE
- GL_COPY_INVERTED
- GL_OR_INVERTED
- GL_NAND
- GL_SET
- GL_EMISSION

- GL_SHININESS
- GL_AMBIENT_AND_DIFFUSE
- GL_COLOR_INDEXES
- GL_MODELVIEW
- GL_PROJECTION
- GL_TEXTURE
- GL_COLOR
- GL_DEPTH
- GL_STENCIL
- GL_COLOR_INDEX
- GL_STENCIL_INDEX
- GL_DEPTH_COMPONENT
- GL_RED
- GL_GREEN
- GL_BLUE
- GL_ALPHA
- GL_RGB
- GL_RGBA
- GL_LUMINANCE
- GL_LUMINANCE_ALPHA
- GL_BITMAP
- GL_POINT
- GL_LINE
- GL_FILL
- GL_RENDER
- GL_FEEDBACK
- GL_SELECT
- GL_FLAT
- GL_SMOOTH
- GL_KEEP
- GL_REPLACE
- GL_INCR
- GL_DECR
- GL_VENDOR
- GL_RENDERER
- GL_VERSION

- GL_EXTENSIONS
- GL_S
- GL_ENABLE_BIT
- GL_T
- GL_R
- GL_Q
- GL_MODULATE
- GL_DECAL
- GL_TEXTURE_ENV_MODE
- GL_TEXTURE_ENV_COLOR
- GL_TEXTURE_ENV
- GL_EYE_LINEAR
- GL_OBJECT_LINEAR
- GL_SPHERE_MAP
- GL_TEXTURE_GEN_MODE
- GL_OBJECT_PLANE
- GL_EYE_PLANE
- GL_NEAREST
- GL_LINEAR
- GL_NEAREST_MIPMAP_NEAREST
- GL_LINEAR_MIPMAP_NEAREST
- GL_NEAREST_MIPMAP_LINEAR
- GL_LINEAR_MIPMAP_LINEAR
- GL_TEXTURE_MAG_FILTER
- GL_TEXTURE_MIN_FILTER
- GL_TEXTURE_WRAP_S
- GL_TEXTURE_WRAP_T
- GL_CLAMP
- GL_REPEAT
- GL_POLYGON_OFFSET_UNITS
- GL_POLYGON_OFFSET_POINT
- GL_POLYGON_OFFSET_LINE
- GL_R3_G3_B2
- GL_V2F
- GL_V3F
- GL_C4UB_V2F

- GL_C4UB_V3F
- GL_C3F_V3F
- GL_N3F_V3F
- GL_C4F_N3F_V3F
- GL_T2F_V3F
- GL_T4F_V4F
- GL_T2F_C4UB_V3F
- GL_T2F_C3F_V3F
- GL_T2F_N3F_V3F
- GL_T2F_C4F_N3F_V3F
- GL_T4F_C4F_N3F_V4F
- GL_CLIP_PLANE0
- GL_CLIP_PLANE1
- GL_CLIP_PLANE2
- GL_CLIP_PLANE3
- GL_CLIP_PLANE4
- GL_CLIP_PLANE5
- GL_LIGHT0
- GL_COLOR_BUFFER_BIT
- GL_LIGHT1
- GL_LIGHT2
- GL_LIGHT3
- GL_LIGHT4
- GL_LIGHT5
- GL_LIGHT6
- GL_LIGHT7
- GL_HINT_BIT
- GL_POLYGON_OFFSET_FILL
- GL_POLYGON_OFFSET_FACTOR
- GL_ALPHA4
- GL_ALPHA8
- GL_ALPHA12
- GL_ALPHA16
- GL_LUMINANCE4
- GL_LUMINANCE8
- GL_LUMINANCE12

- GL_LUMINANCE16
- GL_LUMINANCE4_ALPHA4
- GL_LUMINANCE6_ALPHA2
- GL_LUMINANCE8_ALPHA8
- GL_LUMINANCE12_ALPHA4
- GL_LUMINANCE12_ALPHA12
- GL_LUMINANCE16_ALPHA16
- GL_INTENSITY
- GL_INTENSITY4
- GL_INTENSITY8
- GL_INTENSITY12
- GL_INTENSITY16
- GL_RGB4
- GL_RGB5
- GL_RGB8
- GL_RGB10
- GL_RGB12
- GL_RGB16
- GL_RGBA2
- GL_RGBA4
- GL_RGB5_A1
- GL_RGBA8
- GL_RGB10_A2
- GL_RGBA12
- GL_RGBA16
- GL_TEXTURE_RED_SIZE
- GL_TEXTURE_GREEN_SIZE
- GL_TEXTURE_BLUE_SIZE
- GL_TEXTURE_ALPHA_SIZE
- GL_TEXTURE_LUMINANCE_SIZE
- GL_TEXTURE_INTENSITY_SIZE
- GL_PROXY_TEXTURE_1D
- GL_PROXY_TEXTURE_2D
- GL_TEXTURE_PRIORITY
- GL_TEXTURE_RESIDENT
- GL_TEXTURE_BINDING_1D

- GL_TEXTURE_BINDING_2D
- GL_VERTEX_ARRAY
- GL_NORMAL_ARRAY
- GL_COLOR_ARRAY
- GL_INDEX_ARRAY
- GL_TEXTURE_COORD_ARRAY
- GL_EDGE_FLAG_ARRAY
- GL_VERTEX_ARRAY_SIZE
- GL_VERTEX_ARRAY_TYPE
- GL_VERTEX_ARRAY_STRIDE
- GL_NORMAL_ARRAY_TYPE
- GL_NORMAL_ARRAY_STRIDE
- GL_COLOR_ARRAY_SIZE
- GL_COLOR_ARRAY_TYPE
- GL_COLOR_ARRAY_STRIDE
- GL_INDEX_ARRAY_TYPE
- GL_INDEX_ARRAY_STRIDE
- GL_TEXTURE_COORD_ARRAY_SIZE
- GL_TEXTURE_COORD_ARRAY_TYPE
- GL_TEXTURE_COORD_ARRAY_STRIDE
- GL_EDGE_FLAG_ARRAY_STRIDE
- GL_VERTEX_ARRAY_POINTER
- GL_NORMAL_ARRAY_POINTER
- GL_COLOR_ARRAY_POINTER
- GL_INDEX_ARRAY_POINTER
- GL_TEXTURE_COORD_ARRAY_POINTER
- GL_EDGE_FLAG_ARRAY_POINTER
- GL_COLOR_INDEX1_EXT
- GL_COLOR_INDEX2_EXT
- GL_COLOR_INDEX4_EXT
- GL_COLOR_INDEX8_EXT
- GL_COLOR_INDEX12_EXT
- GL_COLOR_INDEX16_EXT
- GL_EVAL_BIT
- GL_LIST_BIT
- GL_TEXTURE_BIT

- GL_SCISSOR_BIT
- GL_ALL_ATTRIB_BITS
- GL_CLIENT_ALL_ATTRIB_BITS
- GL_SMOOTH_POINT_SIZE_RANGE
- GL_SMOOTH_POINT_SIZE_GRANULARITY
- GL_SMOOTH_LINE_WIDTH_RANGE
- GL_SMOOTH_LINE_WIDTH_GRANULARITY
- GL_UNSIGNED_BYTE_3_3_2
- GL_UNSIGNED_SHORT_4_4_4_4
- GL_UNSIGNED_SHORT_5_5_5_1
- GL_UNSIGNED_INT_8_8_8_8
- GL_UNSIGNED_INT_10_10_10_2
- GL_RESCALE_NORMAL
- GL_TEXTURE_BINDING_3D
- GL_PACK_SKIP_IMAGES
- GL_PACK_IMAGE_HEIGHT
- GL_UNPACK_SKIP_IMAGES
- GL_UNPACK_IMAGE_HEIGHT
- GL_TEXTURE_3D
- GL_PROXY_TEXTURE_3D
- GL_TEXTURE_DEPTH
- GL_TEXTURE_WRAP_R
- GL_MAX_3D_TEXTURE_SIZE
- GL_BGR
- GL_BGRA
- GL_MAX_ELEMENTS_VERTICES
- GL_MAX_ELEMENTS_INDICES
- GL_CLAMP_TO_EDGE
- GL_TEXTURE_MIN_LOD
- GL_TEXTURE_MAX_LOD
- GL_TEXTURE_BASE_LEVEL
- GL_TEXTURE_MAX_LEVEL
- GL_LIGHT_MODEL_COLOR_CONTROL
- GL_SINGLE_COLOR
- GL_SEPARATE_SPECULAR_COLOR
- GL_UNSIGNED_BYTE_2_3_3_REV

- GL_UNSIGNED_SHORT_5_6_5
- GL_UNSIGNED_SHORT_5_6_5_REV
- GL_UNSIGNED_SHORT_4_4_4_4_REV
- GL_UNSIGNED_SHORT_1_5_5_5_REV
- GL_UNSIGNED_INT_8_8_8_8_REV
- GL_ALIASED_POINT_SIZE_RANGE
- GL_ALIASED_LINE_WIDTH_RANGE
- GL_MULTISAMPLE
- GL_SAMPLE_ALPHA_TO_COVERAGE
- GL_SAMPLE_ALPHA_TO_ONE
- GL_SAMPLE_COVERAGE
- GL_SAMPLE_BUFFERS
- GL_SAMPLES
- GL_SAMPLE_COVERAGE_VALUE
- GL_SAMPLE_COVERAGE_INVERT
- GL_CLAMP_TO_BORDER
- GL_TEXTURE0
- GL_TEXTURE1
- GL_TEXTURE2
- GL_TEXTURE3
- GL_TEXTURE4
- GL_TEXTURE5
- GL_TEXTURE6
- GL_TEXTURE7
- GL_TEXTURE8
- GL_TEXTURE9
- GL_TEXTURE10
- GL_TEXTURE11
- GL_TEXTURE12
- GL_TEXTURE13
- GL_TEXTURE14
- GL_TEXTURE15
- GL_TEXTURE16
- GL_TEXTURE17
- GL_TEXTURE18
- GL_TEXTURE19

- GL_TEXTURE20
- GL_TEXTURE21
- GL_TEXTURE22
- GL_TEXTURE23
- GL_TEXTURE24
- GL_TEXTURE25
- GL_TEXTURE26
- GL_TEXTURE27
- GL_TEXTURE28
- GL_TEXTURE29
- GL_TEXTURE30
- GL_TEXTURE31
- GL_ACTIVE_TEXTURE
- GL_CLIENT_ACTIVE_TEXTURE
- GL_MAX_TEXTURE_UNITS
- GL_TRANSPOSE_MODELVIEW_MATRIX
- GL_TRANSPOSE_PROJECTION_MATRIX
- GL_TRANSPOSE_TEXTURE_MATRIX
- GL_TRANSPOSE_COLOR_MATRIX
- GL_SUBTRACT
- GL_COMPRESSED_ALPHA
- GL_COMPRESSED_LUMINANCE
- GL_COMPRESSED_LUMINANCE_ALPHA
- GL_COMPRESSED_INTENSITY
- GL_COMPRESSED_RGB
- GL_COMPRESSED_RGBA
- GL_TEXTURE_COMPRESSION_HINT
- GL_NORMAL_MAP
- GL_REFLECTION_MAP
- GL_TEXTURE_CUBE_MAP
- GL_TEXTURE_BINDING_CUBE_MAP
- GL_TEXTURE_CUBE_MAP_POSITIVE_X
- GL_TEXTURE_CUBE_MAP_NEGATIVE_X
- GL_TEXTURE_CUBE_MAP_POSITIVE_Y
- GL_TEXTURE_CUBE_MAP_NEGATIVE_Y
- GL_TEXTURE_CUBE_MAP_POSITIVE_Z

- GL_TEXTURE_CUBE_MAP_NEGATIVE_Z
- GL_PROXY_TEXTURE_CUBE_MAP
- GL_MAX_CUBE_MAP_TEXTURE_SIZE
- GL_COMBINE
- GL_COMBINE_RGB
- GL_COMBINE_ALPHA
- GL_RGB_SCALE
- GL_ADD_SIGNED
- GL_INTERPOLATE
- GL_CONSTANT
- GL_PRIMARY_COLOR
- GL_PREVIOUS
- GL_SOURCE0_RGB
- GL_SOURCE1_RGB
- GL_SOURCE2_RGB
- GL_SOURCE0_ALPHA
- GL_SOURCE1_ALPHA
- GL_SOURCE2_ALPHA
- GL_OPERAND0_RGB
- GL_OPERAND1_RGB
- GL_OPERAND2_RGB
- GL_OPERAND0_ALPHA
- GL_OPERAND1_ALPHA
- GL_OPERAND2_ALPHA
- GL_TEXTURE_COMPRESSED_IMAGE_SIZE
- GL_TEXTURE_COMPRESSED
- GL_NUM_COMPRESSED_TEXTURE_FORMATS
- GL_COMPRESSED_TEXTURE_FORMATS
- GL_DOT3_RGB
- GL_DOT3_RGBA
- GL_MULTISAMPLE_BIT
- GL_BLEND_DST_RGB
- GL_BLEND_SRC_RGB
- GL_BLEND_DST_ALPHA
- GL_BLEND_SRC_ALPHA
- GL_POINT_SIZE_MIN

- GL_POINT_SIZE_MAX
- GL_POINT_FADE_THRESHOLD_SIZE
- GL_POINT_DISTANCE_ATTENUATION
- GL_GENERATE_MIPMAP
- GL_GENERATE_MIPMAP_HINT
- GL_DEPTH_COMPONENT16
- GL_DEPTH_COMPONENT24
- GL_DEPTH_COMPONENT32
- GL_MIRRORED_REPEAT
- GL_FOG_COORDINATE_SOURCE
- GL_FOG_COORDINATE
- GL_FRAGMENT_DEPTH
- GL_CURRENT_FOG_COORDINATE
- GL_FOG_COORDINATE_ARRAY_TYPE
- GL_FOG_COORDINATE_ARRAY_STRIDE
- GL_FOG_COORDINATE_ARRAY_POINTER
- GL_FOG_COORDINATE_ARRAY
- GL_COLOR_SUM
- GL_CURRENT_SECONDARY_COLOR
- GL_SECONDARY_COLOR_ARRAY_SIZE
- GL_SECONDARY_COLOR_ARRAY_TYPE
- GL_SECONDARY_COLOR_ARRAY_STRIDE
- GL_SECONDARY_COLOR_ARRAY_POINTER
- GL_SECONDARY_COLOR_ARRAY
- GL_MAX_TEXTURE_LOD_BIAS
- GL_TEXTURE_FILTER_CONTROL
- GL_TEXTURE_LOD_BIAS
- GL_INCR_WRAP
- GL_DECR_WRAP
- GL_TEXTURE_DEPTH_SIZE
- GL_DEPTH_TEXTURE_MODE
- GL_TEXTURE_COMPARE_MODE
- GL_TEXTURE_COMPARE_FUNC
- GL_COMPARE_R_TO_TEXTURE
- GL_CURRENT_FOG_COORD
- GL_FOG_COORD

- GL_FOG_COORD_ARRAY
- GL_FOG_COORD_ARRAY_BUFFER_BINDING
- GL_FOG_COORD_ARRAY_POINTER
- GL_FOG_COORD_ARRAY_STRIDE
- GL_FOG_COORD_ARRAY_TYPE
- GL_FOG_COORD_SRC
- GL_SRC0_ALPHA
- GL_SRC0_RGB
- GL_SRC1_ALPHA
- GL_SRC1_RGB
- GL_SRC2_ALPHA
- GL_SRC2_RGB
- GL_BUFFER_SIZE
- GL_BUFFER_USAGE
- GL_QUERY_COUNTER_BITS
- GL_CURRENT_QUERY
- GL_QUERY_RESULT
- GL_QUERY_RESULT_AVAILABLE
- GL_ARRAY_BUFFER
- GL_ELEMENT_ARRAY_BUFFER
- GL_ARRAY_BUFFER_BINDING
- GL_ELEMENT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ARRAY_BUFFER_BINDING
- GL_NORMAL_ARRAY_BUFFER_BINDING
- GL_COLOR_ARRAY_BUFFER_BINDING
- GL_INDEX_ARRAY_BUFFER_BINDING
- GL_TEXTURE_COORD_ARRAY_BUFFER_BINDING
- GL_EDGE_FLAG_ARRAY_BUFFER_BINDING
- GL_SECONDARY_COLOR_ARRAY_BUFFER_BINDING
- GL_FOG_COORDINATE_ARRAY_BUFFER_BINDING
- GL_WEIGHT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ATTRIB_ARRAY_BUFFER_BINDING
- GL_READ_ONLY
- GL_WRITE_ONLY
- GL_READ_WRITE
- GL_BUFFER_ACCESS

- GL_BUFFER_MAPPED
- GL_BUFFER_MAP_POINTER
- GL_STREAM_DRAW
- GL_STREAM_READ
- GL_STREAM_COPY
- GL_STATIC_DRAW
- GL_STATIC_READ
- GL_STATIC_COPY
- GL_DYNAMIC_DRAW
- GL_DYNAMIC_READ
- GL_DYNAMIC_COPY
- GL_SAMPLES_PASSED
- GL_BLEND_EQUATION_RGB
- GL_VERTEX_ATTRIB_ARRAY_ENABLED
- GL_VERTEX_ATTRIB_ARRAY_SIZE
- GL_VERTEX_ATTRIB_ARRAY_STRIDE
- GL_VERTEX_ATTRIB_ARRAY_TYPE
- GL_CURRENT_VERTEX_ATTRIB
- GL_VERTEX_PROGRAM_POINT_SIZE
- GL_VERTEX_PROGRAM_TWO_SIDE
- GL_VERTEX_ATTRIB_ARRAY_POINTER
- GL_STENCIL_BACK_FUNC
- GL_STENCIL_BACK_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_PASS
- GL_MAX_DRAW_BUFFERS
- GL_DRAW_BUFFER0
- GL_DRAW_BUFFER1
- GL_DRAW_BUFFER2
- GL_DRAW_BUFFER3
- GL_DRAW_BUFFER4
- GL_DRAW_BUFFER5
- GL_DRAW_BUFFER6
- GL_DRAW_BUFFER7
- GL_DRAW_BUFFER8
- GL_DRAW_BUFFER9

- GL_DRAW_BUFFER10
- GL_DRAW_BUFFER11
- GL_DRAW_BUFFER12
- GL_DRAW_BUFFER13
- GL_DRAW_BUFFER14
- GL_DRAW_BUFFER15
- GL_BLEND_EQUATION_ALPHA
- GL_POINT_SPRITE
- GL_COORD_REPLACE
- GL_MAX_VERTEX_ATTRIBS
- GL_VERTEX_ATTRIB_ARRAY_NORMALIZED
- GL_MAX_TEXTURE_COORDS
- GL_MAX_TEXTURE_IMAGE_UNITS
- GL_FRAGMENT_SHADER
- GL_VERTEX_SHADER
- GL_MAX_FRAGMENT_UNIFORM_COMPONENTS
- GL_MAX_VERTEX_UNIFORM_COMPONENTS
- GL_MAX_VARYING_FLOATS
- GL_MAX_VERTEX_TEXTURE_IMAGE_UNITS
- GL_MAX_COMBINED_TEXTURE_IMAGE_UNITS
- GL_SHADER_TYPE
- GL_FLOAT_VEC2
- GL_FLOAT_VEC3
- GL_FLOAT_VEC4
- GL_INT_VEC2
- GL_INT_VEC3
- GL_INT_VEC4
- GL_BOOL
- GL_BOOL_VEC2
- GL_BOOL_VEC3
- GL_BOOL_VEC4
- GL_FLOAT_MAT2
- GL_FLOAT_MAT3
- GL_FLOAT_MAT4
- GL_SAMPLER_1D
- GL_SAMPLER_2D

- GL_SAMPLER_3D
- GL_SAMPLER_CUBE
- GL_SAMPLER_1D_SHADOW
- GL_SAMPLER_2D_SHADOW
- GL_DELETE_STATUS
- GL_COMPILE_STATUS
- GL_LINK_STATUS
- GL_VALIDATE_STATUS
- GL_INFO_LOG_LENGTH
- GL_ATTACHED_SHADERS
- GL_ACTIVE_UNIFORMS
- GL_ACTIVE_UNIFORM_MAX_LENGTH
- GL_SHADER_SOURCE_LENGTH
- GL_ACTIVE_ATTRIBUTES
- GL_ACTIVE_ATTRIBUTE_MAX_LENGTH
- GL_FRAGMENT_SHADER_DERIVATIVE_HINT
- GL_SHADING_LANGUAGE_VERSION
- GL_CURRENT_PROGRAM
- GL_POINT_SPRITE_COORD_ORIGIN
- GL_LOWER_LEFT
- GL_UPPER_LEFT
- GL_STENCIL_BACK_REF
- GL_STENCIL_BACK_VALUE_MASK
- GL_STENCIL_BACK_WRITEMASK
- GL_CURRENT_RASTER_SECONDARY_COLOR
- GL_PIXEL_PACK_BUFFER
- GL_PIXEL_UNPACK_BUFFER
- GL_PIXEL_PACK_BUFFER_BINDING
- GL_PIXEL_UNPACK_BUFFER_BINDING
- GL_FLOAT_MAT2x3
- GL_FLOAT_MAT2x4
- GL_FLOAT_MAT3x2
- GL_FLOAT_MAT3x4
- GL_FLOAT_MAT4x2
- GL_FLOAT_MAT4x3
- GL_SRGB

- GL_SRGB8
- GL_SRGB_ALPHA
- GL_SRGB8_ALPHA8
- GL_SLUMINANCE_ALPHA
- GL_SLUMINANCE8_ALPHA8
- GL_SLUMINANCE
- GL_SLUMINANCE8
- GL_COMPRESSED_SRGB
- GL_COMPRESSED_SRGB_ALPHA
- GL_COMPRESSED_SLUMINANCE
- GL_COMPRESSED_SLUMINANCE_ALPHA
- GL_CLIP_DISTANCE0
- GL_CLIP_DISTANCE1
- GL_CLIP_DISTANCE2
- GL_CLIP_DISTANCE3
- GL_CLIP_DISTANCE4
- GL_CLIP_DISTANCE5
- GL_COMPARE_REF_TO_TEXTURE
- GL_MAX_CLIP_DISTANCES
- GL_MAX_VARYING_COMPONENTS
- GL_CONTEXT_FLAG_FORWARD_COMPATIBLE_BIT
- GL_MAJOR_VERSION
- GL_MINOR_VERSION
- GL_NUM_EXTENSIONS
- GL_CONTEXT_FLAGS
- GL_DEPTH_BUFFER
- GL_STENCIL_BUFFER
- GL_RGBA32F
- GL_RGB32F
- GL_RGBA16F
- GL_RGB16F
- GL_VERTEX_ATTRIB_ARRAY_INTEGER
- GL_MAX_ARRAY_TEXTURE_LAYERS
- GL_MIN_PROGRAM_TEXEL_OFFSET
- GL_MAX_PROGRAM_TEXEL_OFFSET
- GL_CLAMP_VERTEX_COLOR

- GL_CLAMP_FRAGMENT_COLOR
- GL_CLAMP_READ_COLOR
- GL_FIXED_ONLY
- GL_TEXTURE_RED_TYPE
- GL_TEXTURE_GREEN_TYPE
- GL_TEXTURE_BLUE_TYPE
- GL_TEXTURE_ALPHA_TYPE
- GL_TEXTURE_LUMINANCE_TYPE
- GL_TEXTURE_INTENSITY_TYPE
- GL_TEXTURE_DEPTH_TYPE
- GL_TEXTURE_1D_ARRAY
- GL_PROXY_TEXTURE_1D_ARRAY
- GL_TEXTURE_2D_ARRAY
- GL_PROXY_TEXTURE_2D_ARRAY
- GL_TEXTURE_BINDING_1D_ARRAY
- GL_TEXTURE_BINDING_2D_ARRAY
- GL_R11F_G11F_B10F
- GL_UNSIGNED_INT_10F_11F_11F_REV
- GL_RGB9_E5
- GL_UNSIGNED_INT_5_9_9_9_REV
- GL_TEXTURE_SHARED_SIZE
- GL_TRANSFORM_FEEDBACK_VARYING_MAX_LENGTH
- GL_TRANSFORM_FEEDBACK_BUFFER_MODE
- GL_MAX_TRANSFORM_FEEDBACK_SEPARATE_COMPONENTS
- GL_TRANSFORM_FEEDBACK_VARYINGS
- GL_TRANSFORM_FEEDBACK_BUFFER_START
- GL_TRANSFORM_FEEDBACK_BUFFER_SIZE
- GL_PRIMITIVES_GENERATED
- GL_TRANSFORM_FEEDBACK_PRIMITIVES_WRITTEN
- GL_RASTERIZER_DISCARD
- GL_MAX_TRANSFORM_FEEDBACK_INTERLEAVED_COMPONENTS
- GL_MAX_TRANSFORM_FEEDBACK_SEPARATE_ATTRIBS
- GL_INTERLEAVED_ATTRIBS
- GL_SEPARATE_ATTRIBS
- GL_TRANSFORM_FEEDBACK_BUFFER
- GL_TRANSFORM_FEEDBACK_BUFFER_BINDING

- GL_RGBA32UI
- GL_RGB32UI
- GL_RGBA16UI
- GL_RGB16UI
- GL_RGBA8UI
- GL_RGB8UI
- GL_RGBA32I
- GL_RGB32I
- GL_RGBA16I
- GL_RGB16I
- GL_RGBA8I
- GL_RGB8I
- GL_RED_INTEGER
- GL_GREEN_INTEGER
- GL_BLUE_INTEGER
- GL_ALPHA_INTEGER
- GL_RGB_INTEGER
- GL_RGBA_INTEGER
- GL_BGR_INTEGER
- GL_BGRA_INTEGER
- GL_SAMPLER_1D_ARRAY
- GL_SAMPLER_2D_ARRAY
- GL_SAMPLER_1D_ARRAY_SHADOW
- GL_SAMPLER_2D_ARRAY_SHADOW
- GL_SAMPLER_CUBE_SHADOW
- GL_UNSIGNED_INT_VEC2
- GL_UNSIGNED_INT_VEC3
- GL_UNSIGNED_INT_VEC4
- GL_INT_SAMPLER_1D
- GL_INT_SAMPLER_2D
- GL_INT_SAMPLER_3D
- GL_INT_SAMPLER_CUBE
- GL_INT_SAMPLER_1D_ARRAY
- GL_INT_SAMPLER_2D_ARRAY
- GL_UNSIGNED_INT_SAMPLER_1D
- GL_UNSIGNED_INT_SAMPLER_2D

- GL_UNSIGNED_INT_SAMPLER_3D
- GL_UNSIGNED_INT_SAMPLER_CUBE
- GL_UNSIGNED_INT_SAMPLER_1D_ARRAY
- GL_UNSIGNED_INT_SAMPLER_2D_ARRAY
- GL_QUERY_WAIT
- GL_QUERY_NO_WAIT
- GL_QUERY_BY_REGION_WAIT
- GL_QUERY_BY_REGION_NO_WAIT
- GL_TEXTURE_RECTANGLE
- GL_TEXTURE_BINDING_RECTANGLE
- GL_PROXY_TEXTURE_RECTANGLE
- GL_MAX_RECTANGLE_TEXTURE_SIZE
- GL_SAMPLER_2D_RECT
- GL_SAMPLER_2D_RECT_SHADOW
- GL_TEXTURE_BUFFER
- GL_MAX_TEXTURE_BUFFER_SIZE
- GL_TEXTURE_BINDING_BUFFER
- GL_TEXTURE_BUFFER_DATA_STORE_BINDING
- GL_TEXTURE_BUFFER_FORMAT
- GL_SAMPLER_BUFFER
- GL_INT_SAMPLER_2D_RECT
- GL_INT_SAMPLER_BUFFER
- GL_UNSIGNED_INT_SAMPLER_2D_RECT
- GL_UNSIGNED_INT_SAMPLER_BUFFER
- GL_RED_SNORM
- GL_RG_SNORM
- GL_RGB_SNORM
- GL_RGBA_SNORM
- GL_R8_SNORM
- GL_RG8_SNORM
- GL_RGB8_SNORM
- GL_RGBA8_SNORM
- GL_R16_SNORM
- GL_RG16_SNORM
- GL_RGB16_SNORM
- GL_RGBA16_SNORM

- GL_SIGNED_NORMALIZED
- GL_PRIMITIVE_RESTART
- GL_PRIMITIVE_RESTART_INDEX
- GL_BUFFER_ACCESS_FLAGS
- GL_BUFFER_MAP_LENGTH
- GL_BUFFER_MAP_OFFSET
- GL_CONTEXT_CORE_PROFILE_BIT
- GL_CONTEXT_COMPATIBILITY_PROFILE_BIT
- GL_LINES_ADJACENCY
- GL_LINE_STRIP_ADJACENCY
- GL_TRIANGLES_ADJACENCY
- GL_TRIANGLE_STRIP_ADJACENCY
- GL_PROGRAM_POINT_SIZE
- GL_GEOMETRY_VERTICES_OUT
- GL_GEOMETRY_INPUT_TYPE
- GL_GEOMETRY_OUTPUT_TYPE
- GL_MAX_GEOMETRY_TEXTURE_IMAGE_UNITS
- GL_FRAMEBUFFER_ATTACHMENT_LAYERED
- GL_FRAMEBUFFER_INCOMPLETE_LAYER_TARGETS
- GL_GEOMETRY_SHADER
- GL_MAX_GEOMETRY_UNIFORM_COMPONENTS
- GL_MAX_GEOMETRY_OUTPUT_VERTICES
- GL_MAX_GEOMETRY_TOTAL_OUTPUT_COMPONENTS
- GL_MAX_VERTEX_OUTPUT_COMPONENTS
- GL_MAX_GEOMETRY_INPUT_COMPONENTS
- GL_MAX_GEOMETRY_OUTPUT_COMPONENTS
- GL_MAX_FRAGMENT_INPUT_COMPONENTS
- GL_CONTEXT_PROFILE_MASK
- void glAccum(GLenum op, GLfloat value)
- void glActiveTexture(GLenum texture)
- void glAlphaFunc(GLenum func, GLclampf ref)
- GLboolean glAreTexturesResident(GLsizei n, const GLuint * textures, GLboolean * residences)
- void glArrayElement(GLint i)
- void glAttachShader(GLuint program, GLuint shader)
- void glBegin(GLenum mode)
- void glBeginQuery(GLenum target, GLuint id)

- void glBindAttribLocation(GLuint program,GLuint index,const GLchar *name)
- void glBindBuffer(GLenum target,GLuint buffer)
- void glBindTexture(GLenum target,GLuint texture)
- void glBitmap(GLsizei width,GLsizei height,GLfloat xorig,GLfloat yorig,GLfloat xmove,GLfloat ymove,const GLubyte * bitmap)
- void glBlendColor(GLclampf red,GLclampf green,GLclampf blue,GLclampf alpha)
- void glBlendEquation(GLenum mode)
- void glBlendEquationSeparate(GLenum modeRGB,GLenum modeAlpha)
- void glBlendFunc(GLenum sfactor,GLenum dfactor)
- void glBlendFuncSeparate(GLenum srcRGB,GLenum dstRGB,GLenum srcAlpha,GLenum dstAlpha)
- void glBufferData(GLenum target,GLsizeiptr size,const GLvoid * data,GLenum usage)
- void glBufferSubData(GLenum target,GLintptr offset,GLsizeiptr size,const GLvoid * data)
- void glCallList(GLuint list)
- void glCallLists(GLsizei n,GLenum type,const GLvoid * lists)
- void glClear(GLbitfield mask)
- void glClearAccum(GLfloat red,GLfloat green,GLfloat blue,GLfloat alpha)
- void glClearColor(GLclampf red,GLclampf green,GLclampf blue,GLclampf alpha)
- void glClearDepth(GLclampd depth)
- void glClearIndex(GLfloat c)
- void glClearStencil(GLint s)
- void glClientActiveTexture(GLenum texture)
- void glClipPlane(GLenum plane,const GLdouble * equation)
- void glColor3b(GLbyte red,GLbyte green,GLbyte blue)
- void glColor3s(GLshort red,GLshort green,GLshort blue)
- void glColor3i(GLint red,GLint green,GLint blue)
- void glColor3f(GLfloat red,GLfloat green,GLfloat blue)
- void glColor3d(GLdouble red,GLdouble green,GLdouble blue)
- void glColor3ub(GLubyte red,GLubyte green,GLubyte blue)
- void glColor3us(GLushort red,GLushort green,GLushort blue)
- void glColor3ui(GLuint red,GLuint green,GLuint blue)
- void glColor4b(GLbyte red,GLbyte green,GLbyte blue,GLbyte alpha)
- void glColor4s(GLshort red,GLshort green,GLshort blue,GLshort alpha)
- void glColor4i(GLint red,GLint green,GLint blue,GLint alpha)
- void glColor4f(GLfloat red,GLfloat green,GLfloat blue,GLfloat alpha)
- void glColor4d(GLdouble red,GLdouble green,GLdouble blue,GLdouble alpha)
- void glColor4ub(GLubyte red,GLubyte green,GLubyte blue,GLubyte alpha)

- void glColor4us(GLushort red, GLushort green, GLushort blue, GLushort alpha)
- void glColor4ui(GLuint red, GLuint green, GLuint blue, GLuint alpha)
- void glColor3bv(const GLbyte * v)
- void glColor3sv(const GLshort * v)
- void glColor3iv(const GLint * v)
- void glColor3fv(const GLfloat * v)
- void glColor3dv(const GLdouble * v)
- void glColor3ubv(const GLubyte * v)
- void glColor3usv(const GLushort * v)
- void glColor3uiv(const GLuint * v)
- void glColor4bv(const GLbyte * v)
- void glColor4sv(const GLshort * v)
- void glColor4iv(const GLint * v)
- void glColor4fv(const GLfloat * v)
- void glColor4dv(const GLdouble * v)
- void glColor4ubv(const GLubyte * v)
- void glColor4usv(const GLushort * v)
- void glColor4uiv(const GLuint * v)
- void glColorMask(GLboolean red, GLboolean green, GLboolean blue, GLboolean alpha)
- void glColorMaterial(GLenum face, GLenum mode)
- void glColorPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)
- void glColorSubTable(GLenum target, GLsizei start, GLsizei count, GLenum format, GLenum type, const GLvoid * data)
- void glColorTable(GLenum target, GLenum internalformat, GLsizei width, GLenum format, GLenum type, const GLvoid * data)
- void glColorTableParameterfv(GLenum target, GLenum pname, const GLfloat * params)
- void glColorTableParameteriv(GLenum target, GLenum pname, const GLint * params)
- void glCompileShader(GLuint shader)
- void glCompressedTexImage1D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLint border, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexImage2D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLint border, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexImage3D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLsizei depth, GLint border, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLsizei imageSize, const GLvoid * data)

- void glCompressedTexSubImage3D(GLenum target,GLint level,GLint xoffset,GLint yoffset,GLint zoffset,GLsizei width,GLsizei height,GLsizei depth,GLenum format,GLsizei imageSize,const GLvoid * data)
- void glConvolutionFilter1D(GLenum target,GLenum internalformat,GLsizei width,GLenum format,GLenum type,const GLvoid * data)
- void glConvolutionFilter2D(GLenum target,GLenum internalformat,GLsizei width,GLsizei height,GLenum format,GLenum type,const GLvoid * data)
- void glConvolutionParameterf(GLenum target,GLenum pname,GLfloat params)
- void glConvolutionParameteri(GLenum target,GLenum pname,GLint params)
- void glConvolutionParameterfv(GLenum target,GLenum pname,const GLfloat * params)
- void glConvolutionParameteriv(GLenum target,GLenum pname,const GLint * params)
- void glCopyColorSubTable(GLenum target,GLsizei start,GLint x,GLint y,GLsizei width)
- void glCopyColorTable(GLenum target,GLenum internalformat,GLint x,GLint y,GLsizei width)
- void glCopyConvolutionFilter1D(GLenum target,GLenum internalformat,GLint x,GLint y,GLsizei width)
- void glCopyConvolutionFilter2D(GLenum target,GLenum internalformat,GLint x,GLint y,GLsizei width,GLsizei height)
- void glCopyPixels(GLint x,GLint y,GLsizei width,GLsizei height,GLenum type)
- void glCopyTexImage1D(GLenum target,GLint level,GLenum internalformat,GLint x,GLint y,GLsizei width,GLint border)
- void glCopyTexImage2D(GLenum target,GLint level,GLenum internalformat,GLint x,GLint y,GLsizei width,GLsizei height,GLint border)
- void glCopyTexSubImage1D(GLenum target,GLint level,GLint xoffset,GLint x,GLint y,GLsizei width)
- void glCopyTexSubImage2D(GLenum target,GLint level,GLint xoffset,GLint yoffset,GLint x,GLint y,GLsizei width,GLsizei height)
- void glCopyTexSubImage3D(GLenum target,GLint level,GLint xoffset,GLint yoffset,GLint zoffset,GLint x,GLint y,GLsizei width,GLsizei height)
- GLuint glCreateProgram(void)
- GLuint glCreateShader(GLenum shaderType)
- void glCullFace(GLenum mode)
- void glDeleteBuffers(GLsizei n,const GLuint * buffers)
- void glDeleteLists(GLuint list,GLsizei range)
- void glDeleteProgram(GLuint program)
- void glDeleteQueries(GLsizei n,const GLuint * ids)
- void glDeleteShader(GLuint shader)
- void glDeleteTextures(GLsizei n,const GLuint * textures)
- void glDepthFunc(GLenum func)
- void glDepthMask(GLboolean flag)
- void glDepthRange(GLclampd nearVal,GLclampd farVal)
- void glDetachShader(GLuint program,GLuint shader)
- void glEnable(GLenum cap)

- `void glEnableClientState(GLenum cap)`
- `void glEnableVertexAttribArray(GLuint index)`
- `void glDisableVertexAttribArray(GLuint index)`
- `void glDrawArrays(GLenum mode, GLint first, GLsizei count)`
- `void glDrawBuffer(GLenum mode)`
- `void glDrawBuffers(GLsizei n, const GLenum *bufs)`
- `void glDrawElements(GLenum mode, GLsizei count, GLenum type, const GLvoid * indices)`
- `void glDrawPixels(GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)`
- `void glDrawRangeElements(GLenum mode, GLuint start, GLuint end, GLsizei count, GLenum type, const GLvoid * indices)`
- `void glEdgeFlag(GLboolean flag)`
- `void glEdgeFlagPointer(GLsizei stride, const GLvoid * pointer)`
- `void glEnd(void)`
- `void glEndList(void)`
- `void glEndQuery(GLenum target)`
- `void glEvalCoord1f(GLfloat u)`
- `void glEvalCoord1d(GLdouble u)`
- `void glEvalCoord2f(GLfloat u, GLfloat v)`
- `void glEvalCoord2d(GLdouble u, GLdouble v)`
- `void glEvalMesh1(GLenum mode, GLint i1, GLint i2)`
- `void glEvalPoint1(GLint i)`
- `void glEvalPoint2(GLint i, GLint j)`
- `void glFeedbackBuffer(GLsizei size, GLenum type, GLfloat * buffer)`
- `void glFinish(void)`
- `void glFlush(void)`
- `void glFogf(GLenum pname, GLfloat param)`
- `void glFogi(GLenum pname, GLint param)`
- `void glFogfv(GLenum pname, const GLfloat * params)`
- `void glFogiv(GLenum pname, const GLint * params)`
- `void glFogCoordd(GLdouble coord)`
- `void glFogCoordf(GLfloat coord)`
- `void glFogCoorddv(GLdouble * coord)`
- `void glFogCoordfv(GLfloat * coord)`
- `void glFogCoordPointer(GLenum type, GLsizei stride, GLvoid * pointer)`
- `void glFrontFace(GLenum mode)`
- `void glFrustum(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble nearVal, GLdouble farVal)`

- void glGenBuffers(GLsizei n, GLuint * buffers)
- GLuint glGenLists(GLsizei range)
- void glGenQueries(GLsizei n, GLuint * ids)
- void glGenTextures(GLsizei n, GLuint * textures)
- void glGetBooleanv(GLenum pname, GLboolean * params)
- void glGetDoublev(GLenum pname, GLdouble * params)
- void glGetFloatv(GLenum pname, GLfloat * params)
- void glGetIntegerv(GLenum pname, GLint * params)
- void glGetActiveAttrib(GLuint program, GLuint index, GLsizei bufSize, GLsizei *length, GLint *size, GLenum *type, GLchar *name)
- void glGetActiveUniform(GLuint program, GLuint index, GLsizei bufSize, GLsizei *length, GLint *size, GLenum *type, GLchar *name)
- void glGetAttachedShaders(GLuint program, GLsizei maxCount, GLsizei *count, GLuint *shaders)
- GLint glGetAttribLocation(GLuint program, const GLchar *name)
- void glGetBufferParameteriv(GLenum target, GLenum value, GLint * data)
- void glGetBufferPointerv(GLenum target, GLenum pname, GLvoid ** params)
- void glGetBufferSubData(GLenum target, GLintptr offset, GLsizeiptr size, GLvoid * data)
- void glGetClipPlane(GLenum plane, GLdouble * equation)
- void glGetColorTable(GLenum target, GLenum format, GLenum type, GLvoid * table)
- void glGetColorTableParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetColorTableParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetCompressedTexImage(GLenum target, GLint lod, GLvoid * img)
- void glGetConvolutionFilter(GLenum target, GLenum format, GLenum type, GLvoid * image)
- void glGetConvolutionParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetConvolutionParameteriv(GLenum target, GLenum pname, GLint * params)
- GLenum glGetError(void)
- void glGetHistogram(GLenum target, GLboolean reset, GLenum format, GLenum type, GLvoid * values)
- void glGetHistogramParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetHistogramParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetLightfv(GLenum light, GLenum pname, GLfloat * params)
- void glGetLightiv(GLenum light, GLenum pname, GLint * params)
- void glGetMapdv(GLenum target, GLenum query, GLdouble * v)
- void glGetMapfv(GLenum target, GLenum query, GLfloat * v)
- void glGetMapiv(GLenum target, GLenum query, GLint * v)
- void glGetMaterialfv(GLenum face, GLenum pname, GLfloat * params)
- void glGetMaterialiv(GLenum face, GLenum pname, GLint * params)
- void glGetMinmax(GLenum target, GLboolean reset, GLenum format, GLenum types, GLvoid * values)

- void glGetMinmaxParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetMinmaxParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetPixelMapfv(GLenum map, GLfloat * data)
- void glGetPixelMapuiv(GLenum map, GLuint * data)
- void glGetPixelMapusv(GLenum map, GLushort * data)
- void glGetPointerv(GLenum pname, GLvoid ** params)
- void glGetPolygonStipple(GLubyte * pattern)
- void glGetProgramiv(GLuint program, GLenum pname, GLint *params)
- void glGetProgramInfoLog(GLuint program, GLsizei maxLength, GLsizei *length, GLchar *infoLog)
- void glGetQueryObjectiv(GLuint id, GLenum pname, GLint * params)
- void glGetQueryObjectuiv(GLuint id, GLenum pname, GLuint * params)
- void glGetQueryiv(GLenum target, GLenum pname, GLint * params)
- void glGetSeparableFilter(GLenum target, GLenum format, GLenum type, GLvoid * row, GLvoid * column, GLvoid * span)
- void glGetShaderiv(GLuint shader, GLenum pname, GLint *params)
- void glGetShaderInfoLog(GLuint shader, GLsizei maxLength, GLsizei *length, GLchar *infoLog)
- void glGetShaderSource(GLuint shader, GLsizei bufSize, GLsizei *length, GLchar *source)
- const GLubyte* getString(GLenum name)
- void glGetTexEnvfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetTexEnviv(GLenum target, GLenum pname, GLint * params)
- void glGetTexGendv(GLenum coord, GLenum pname, GLdouble * params)
- void glGetTexGenfv(GLenum coord, GLenum pname, GLfloat * params)
- void glGetTexGeniv(GLenum coord, GLenum pname, GLint * params)
- void glGetTexImage(GLenum target, GLint level, GLenum format, GLenum type, GLvoid * img)
- void glGetTexLevelParameterfv(GLenum target, GLint level, GLenum pname, GLfloat * params)
- void glGetTexLevelParameteriv(GLenum target, GLint level, GLenum pname, GLint * params)
- void glGetTexParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetTexParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetUniformfv(GLuint program, GLint location, GLfloat *params)
- void glGetUniformiv(GLuint program, GLint location, GLint *params)
- GLint glGetUniformLocation(GLuint program, const GLchar *name)
- void glGetVertexAttribdv(GLuint index, GLenum pname, GLdouble *params)
- void glGetVertexAttribfv(GLuint index, GLenum pname, GLfloat *params)
- void glGetVertexAttribiv(GLuint index, GLenum pname, GLint *params)
- void glGetVertexAttribPointerv(GLuint index, GLenum pname, GLvoid **pointer)
- void glHint(GLenum target, GLenum mode)

- void glHistogram(GLenum target,GLsizei width,GLenum internalformat,GLboolean sink)
- void glIndexs(GLshort c)
- void glIndexi(GLint c)
- void glIndexf(GLfloat c)
- void glIndexd(GLdouble c)
- void glIndexub(GLubyte c)
- void glIndexsv(const GLshort * c)
- void glIndexiv(const GLint * c)
- void glIndexfv(const GLfloat * c)
- void glIndexdv(const GLdouble * c)
- void glIndexubv(const GLubyte * c)
- void glIndexMask(GLuint mask)
- void glIndexPointer(GLenum type,GLsizei stride,const GLvoid * pointer)
- void glInitNames(void)
- void glInterleavedArrays(GLenum format,GLsizei stride,const GLvoid * pointer)
- GLboolean glIsBuffer(GLuint buffer)
- GLboolean glIsEnabled(GLenum cap)
- GLboolean glIsList(GLuint list)
- GLboolean glIsProgram(GLuint program)
- GLboolean glIsQuery(GLuint id)
- GLboolean glIsShader(GLuint shader)
- GLboolean glIsTexture(GLuint texture)
- void glLightf(GLenum light,GLenum pname,GLfloat param)
- void glLighti(GLenum light,GLenum pname,GLint param)
- void glLightfv(GLenum light,GLenum pname,const GLfloat * params)
- void glLightiv(GLenum light,GLenum pname,const GLint * params)
- void glLightModelf(GLenum pname,GLfloat param)
- void glLightModeli(GLenum pname,GLint param)
- void glLightModelfv(GLenum pname,const GLfloat * params)
- void glLightModeliv(GLenum pname,const GLint * params)
- void glLineStipple(GLint factor,GLushort pattern)
- void glLineWidth(GLfloat width)
- void glLinkProgram(GLuint program)
- void glListBase(GLuint base)
- void glLoadIdentity(void)
- void glLoadMatrixd(const GLdouble * m)

- `void glLoadMatrixf(const GLfloat * m)`
- `void glLoadName(GLuint name)`
- `void glLoadTransposeMatrixd(const GLdouble * m)`
- `void glLoadTransposeMatrixf(const GLfloat * m)`
- `void glLogicOp(GLenum opcode)`
- `void glMap1f(GLenum target, GLfloat u1, GLfloat u2, GLint stride, GLint order, const GLfloat * points)`
- `void glMap1d(GLenum target, GLdouble u1, GLdouble u2, GLint stride, GLint order, const GLdouble * points)`
- `void glMap2f(GLenum target, GLfloat u1, GLfloat u2, GLint ustride, GLint uorder, GLfloat v1, GLfloat v2, GLint vstride, GLint vorder, const GLfloat * points)`
- `void glMap2d(GLenum target, GLdouble u1, GLdouble u2, GLint ustride, GLint uorder, GLdouble v1, GLdouble v2, GLint vstride, GLint vorder, const GLdouble * points)`
- `void * glMapBuffer(GLenum target, GLenum access)`
- `void glMapGrid1d(GLint un, GLdouble u1, GLdouble u2)`
- `void glMapGrid1f(GLint un, GLfloat u1, GLfloat u2)`
- `void glMapGrid2d(GLint un, GLdouble u1, GLdouble u2, GLint vn, GLdouble v1, GLdouble v2)`
- `void glMapGrid2f(GLint un, GLfloat u1, GLfloat u2, GLint vn, GLfloat v1, GLfloat v2)`
- `void glMaterialf(GLenum face, GLenum pname, GLfloat param)`
- `void glMateriali(GLenum face, GLenum pname, GLint param)`
- `void glMatrixMode(GLenum mode)`
- `void glMinmax(GLenum target, GLenum internalformat, GLboolean sink)`
- `void glMultMatrixd(const GLdouble * m)`
- `void glMultMatrixf(const GLfloat * m)`
- `void glMultTransposeMatrixd(const GLdouble * m)`
- `void glMultTransposeMatrixf(const GLfloat * m)`
- `void glMultiDrawArrays(GLenum mode, GLint * first, GLsizei * count, GLsizei primcount)`
- `void glMultiDrawElements(GLenum mode, const GLsizei * count, GLenum type, const GLvoid ** indices, GLsizei primcount)`
- `void glMultiTexCoord1s(GLenum target, GLshort s)`
- `void glMultiTexCoord1i(GLenum target, GLint s)`
- `void glMultiTexCoord1f(GLenum target, GLfloat s)`
- `void glMultiTexCoord1d(GLenum target, GLdouble s)`
- `void glMultiTexCoord2s(GLenum target, GLshort s, GLshort t)`
- `void glMultiTexCoord2i(GLenum target, GLint s, GLint t)`
- `void glMultiTexCoord2f(GLenum target, GLfloat s, GLfloat t)`
- `void glMultiTexCoord2d(GLenum target, GLdouble s, GLdouble t)`
- `void glMultiTexCoord3s(GLenum target, GLshort s, GLshort t, GLshort r)`
- `void glMultiTexCoord3i(GLenum target, GLint s, GLint t, GLint r)`

- `void glMultiTexCoord3f(GLenum target, GLfloat s, GLfloat t, GLfloat r)`
- `void glMultiTexCoord3d(GLenum target, GLdouble s, GLdouble t, GLdouble r)`
- `void glMultiTexCoord4s(GLenum target, GLshort s, GLshort t, GLshort r, GLshort q)`
- `void glMultiTexCoord4i(GLenum target, GLint s, GLint t, GLint r, GLint q)`
- `void glMultiTexCoord4f(GLenum target, GLfloat s, GLfloat t, GLfloat r, GLfloat q)`
- `void glMultiTexCoord4d(GLenum target, GLdouble s, GLdouble t, GLdouble r, GLdouble q)`
- `void glMultiTexCoord1sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord1iv(GLenum target, const GLint * v)`
- `void glMultiTexCoord1fv(GLenum target, const GLfloat * v)`
- `void glMultiTexCoord1dv(GLenum target, const GLdouble * v)`
- `void glMultiTexCoord2sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord2iv(GLenum target, const GLint * v)`
- `void glMultiTexCoord2fv(GLenum target, const GLfloat * v)`
- `void glMultiTexCoord2dv(GLenum target, const GLdouble * v)`
- `void glMultiTexCoord3sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord3iv(GLenum target, const GLint * v)`
- `void glMultiTexCoord3fv(GLenum target, const GLfloat * v)`
- `void glMultiTexCoord3dv(GLenum target, const GLdouble * v)`
- `void glMultiTexCoord4sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord4iv(GLenum target, const GLint * v)`
- `void glMultiTexCoord4fv(GLenum target, const GLfloat * v)`
- `void glMultiTexCoord4dv(GLenum target, const GLdouble * v)`
- `void glNewList(GLuint list, GLenum mode)`
- `void glNormal3b(GLbyte nx, GLbyte ny, GLbyte nz)`
- `void glNormal3d(GLdouble nx, GLdouble ny, GLdouble nz)`
- `void glNormal3f(GLfloat nx, GLfloat ny, GLfloat nz)`
- `void glNormal3i(GLint nx, GLint ny, GLint nz)`
- `void glNormal3s(GLshort nx, GLshort ny, GLshort nz)`
- `void glNormal3bv(const GLbyte * v)`
- `void glNormal3dv(const GLdouble * v)`
- `void glNormal3fv(const GLfloat * v)`
- `void glNormal3iv(const GLint * v)`
- `void glNormal3sv(const GLshort * v)`
- `void glNormalPointer(GLenum type, GLsizei stride, const GLvoid * pointer)`
- `void glOrtho(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble nearVal, GLdouble farVal)`

- void glPassThrough(GLfloat token)
- void glPixelMapfv(GLenum map,GLsizei mapsize,const GLfloat * values)
- void glPixelMapuiv(GLenum map,GLsizei mapsize,const GLuint * values)
- void glPixelMapusv(GLenum map,GLsizei mapsize,const GLushort * values)
- void glPixelStoref(GLenum pname,GLfloat param)
- void glPixelStorei(GLenum pname,GLint param)
- void glPixelTransferf(GLenum pname,GLfloat param)
- void glPixelTransferi(GLenum pname,GLint param)
- void glPixelZoom(GLfloat xfactor,GLfloat yfactor)
- void glPointParameterf(GLenum pname,GLfloat param)
- void glPointParameteri(GLenum pname,GLint param)
- void glPointSize(GLfloat size)
- void glPolygonMode(GLenum face,GLenum mode)
- void glPolygonOffset(GLfloat factor,GLfloat units)
- void glPolygonStipple(const GLubyte * pattern)
- void glPushAttrib(GLbitfield mask)
- void glPushClientAttrib(GLbitfield mask)
- void glPushMatrix(void)
- void glPushName(GLuint name)
- void glPrioritizeTextures(GLsizei n,const GLuint * textures,const GLclampf * priorities)
- void glPopMatrix(void)
- void glRasterPos2s(GLshort x,GLshort y)
- void glRasterPos2i(GLint x,GLint y)
- void glRasterPos2f(GLfloat x,GLfloat y)
- void glRasterPos2d(GLdouble x,GLdouble y)
- void glRasterPos3s(GLshort x,GLshort y,GLshort z)
- void glRasterPos3i(GLint x,GLint y,GLint z)
- void glRasterPos3f(GLfloat x,GLfloat y,GLfloat z)
- void glRasterPos3d(GLdouble x,GLdouble y,GLdouble z)
- void glRasterPos4s(GLshort x,GLshort y,GLshort z,GLshort w)
- void glRasterPos4i(GLint x,GLint y,GLint z,GLint w)
- void glRasterPos4f(GLfloat x,GLfloat y,GLfloat z,GLfloat w)
- void glRasterPos4d(GLdouble x,GLdouble y,GLdouble z,GLdouble w)
- void glReadBuffer(GLenum mode)
- void glReadPixels(GLint x,GLint y,GLsizei width,GLsizei height,GLenum format,GLenum type,GLvoid * data)
- void glRectd(GLdouble x1,GLdouble y1,GLdouble x2,GLdouble y2)

- `void glRectf(GLfloat x1,GLfloat y1,GLfloat x2,GLfloat y2)`
- `void glRecti(GLint x1,GLint y1,GLint x2,GLint y2)`
- `void glRects(GLshort x1,GLshort y1,GLshort x2,GLshort y2)`
- `void glRectdv(const GLdouble * v1,const GLdouble * v2)`
- `void glRectfv(const GLfloat * v1,const GLfloat * v2)`
- `void glRectiv(const GLint * v1,const GLint * v2)`
- `void glRectsv(const GLshort * v1,const GLshort * v2)`
- `GLint glRenderMode(GLenum mode)`
- `void glResetHistogram(GLenum target)`
- `void glRotated(GLdouble angle,GLdouble x,GLdouble y,GLdouble z)`
- `void glRotatef(GLfloat angle,GLfloat x,GLfloat y,GLfloat z)`
- `void glSampleCoverage(GLclampf value,GLboolean invert)`
- `void glScaled(GLdouble x,GLdouble y,GLdouble z)`
- `void glScalef(GLfloat x,GLfloat y,GLfloat z)`
- `void glScissor(GLint x,GLint y,GLsizei width,GLsizei height)`
- `void glSecondaryColor3b(GLbyte red,GLbyte green,GLbyte blue)`
- `void glSecondaryColor3s(GLshort red,GLshort green,GLshort blue)`
- `void glSecondaryColor3i(GLint red,GLint green,GLint blue)`
- `void glSecondaryColor3f(GLfloat red,GLfloat green,GLfloat blue)`
- `void glSecondaryColor3d(GLdouble red,GLdouble green,GLdouble blue)`
- `void glSecondaryColor3ub(GLubyte red,GLubyte green,GLubyte blue)`
- `void glSecondaryColor3us(GLushort red,GLushort green,GLushort blue)`
- `void glSecondaryColor3ui(GLuint red,GLuint green,GLuint blue)`
- `void glSecondaryColor3bv(const GLbyte * v)`
- `void glSecondaryColor3sv(const GLshort * v)`
- `void glSecondaryColor3iv(const GLint * v)`
- `void glSecondaryColor3fv(const GLfloat * v)`
- `void glSecondaryColor3dv(const GLdouble * v)`
- `void glSecondaryColor3ubv(const GLubyte * v)`
- `void glSecondaryColor3usv(const GLushort * v)`
- `void glSecondaryColor3uiv(const GLuint * v)`
- `void glSecondaryColorPointer(GLint size,GLenum type,GLsizei stride,const GLvoid * pointer)`
- `void glSelectBuffer(GLsizei size,GLuint * buffer)`
- `void glSeparableFilter2D(GLenum target,GLenum internalformat,GLsizei width,GLsizei height,GLenum format,GLenum type,const GLvoid * row,const GLvoid * column)`
- `void glShadeModel(GLenum mode)`

- void glShaderSource(GLuint shader,GLsizei count,const GLchar **string,const GLint *length)
- void glStencilFunc(GLenum func,GLint ref,GLuint mask)
- void glStencilFuncSeparate(GLenum face,GLenum func,GLint ref,GLuint mask)
- void glStencilMask(GLuint mask)
- void glStencilMaskSeparate(GLenum face,GLuint mask)
- void glStencilOp(GLenum sfail,GLenum dpfail,GLenum dppass)
- void glStencilOpSeparate(GLenum face,GLenum sfail,GLenum dpfail,GLenum dppass)
- void glTexCoord1s(GLshort s)
- void glTexCoord1i(GLint s)
- void glTexCoord1f(GLfloat s)
- void glTexCoord1d(GLdouble s)
- void glTexCoord2s(GLshort s,GLshort t)
- void glTexCoord2i(GLint s,GLint t)
- void glTexCoord2f(GLfloat s,GLfloat t)
- void glTexCoord2d(GLdouble s,GLdouble t)
- void glTexCoord3s(GLshort s,GLshort t,GLshort r)
- void glTexCoord3i(GLint s,GLint t,GLint r)
- void glTexCoord3f(GLfloat s,GLfloat t,GLfloat r)
- void glTexCoord3d(GLdouble s,GLdouble t,GLdouble r)
- void glTexCoord4s(GLshort s,GLshort t,GLshort r,GLshort q)
- void glTexCoord4i(GLint s,GLint t,GLint r,GLint q)
- void glTexCoord4f(GLfloat s,GLfloat t,GLfloat r,GLfloat q)
- void glTexCoord4d(GLdouble s,GLdouble t,GLdouble r,GLdouble q)
- void glTexCoord1sv(const GLshort * v)
- void glTexCoord1iv(const GLint * v)
- void glTexCoord1fv(const GLfloat * v)
- void glTexCoord1dv(const GLdouble * v)
- void glTexCoord2sv(const GLshort * v)
- void glTexCoord2iv(const GLint * v)
- void glTexCoord2fv(const GLfloat * v)
- void glTexCoord2dv(const GLdouble * v)
- void glTexCoord3sv(const GLshort * v)
- void glTexCoord3iv(const GLint * v)
- void glTexCoord3fv(const GLfloat * v)
- void glTexCoord3dv(const GLdouble * v)
- void glTexCoord4sv(const GLshort * v)

- `void glTexCoord4iv(const GLint * v)`
- `void glTexCoord4fv(const GLfloat * v)`
- `void glTexCoord4dv(const GLdouble * v)`
- `void glTexCoordPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)`
- `void glTexEnvf(GLenum target, GLenum pname, GLfloat param)`
- `void glTexEnvf(GLenum target, GLenum pname, GLint param)`
- `void glTexGenf(GLenum coord, GLenum pname, GLfloat param)`
- `void glTexGenf(GLenum coord, GLenum pname, GLdouble param)`
- `void glTexGeniv(GLenum coord, GLenum pname, const GLint * params)`
- `void glTexGenfv(GLenum coord, GLenum pname, const GLfloat * params)`
- `void glTexGendv(GLenum coord, GLenum pname, const GLdouble * params)`
- `void glTexImage1D(GLenum target, GLint level, GLint internalFormat, GLsizei width, GLint border, GLenum format, GLenum type, const GLvoid * data)`
- `void glTexImage2D(GLenum target, GLint level, GLint internalFormat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, const GLvoid * data)`
- `void glTexImage3D(GLenum target, GLint level, GLint internalFormat, GLsizei width, GLsizei height, GLsizei depth, GLint border, GLenum format, GLenum type, const GLvoid * data)`
- `void glTexParameterf(GLenum target, GLenum pname, GLfloat param)`
- `void glTexParameteri(GLenum target, GLenum pname, GLint param)`
- `void glTexParameterfv(GLenum target, GLenum pname, const GLfloat * params)`
- `void glTexParameteriv(GLenum target, GLenum pname, const GLint * params)`
- `void glTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLenum type, const GLvoid * data)`
- `void glTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)`
- `void glTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLenum type, const GLvoid * data)`
- `void glTranslated(GLdouble x, GLdouble y, GLdouble z)`
- `void glTranslatef(GLfloat x, GLfloat y, GLfloat z)`
- `void glUniform1f(GLint location, GLfloat v0)`
- `void glUniform2f(GLint location, GLfloat v0, GLfloat v1)`
- `void glUniform3f(GLint location, GLfloat v0, GLfloat v1, GLfloat v2)`
- `void glUniform4f(GLint location, GLfloat v0, GLfloat v1, GLfloat v2, GLfloat v3)`
- `void glUniform1i(GLint location, GLint v0)`
- `void glUniform2i(GLint location, GLint v0, GLint v1)`
- `void glUniform3i(GLint location, GLint v0, GLint v1, GLint v2)`
- `void glUniform4i(GLint location, GLint v0, GLint v1, GLint v2, GLint v3)`

- `void glUniform1fv(GLint location, GLsizei count, const GLfloat *value)`
- `void glUniform2fv(GLint location, GLsizei count, const GLfloat *value)`
- `void glUniform3fv(GLint location, GLsizei count, const GLfloat *value)`
- `void glUniform4fv(GLint location, GLsizei count, const GLfloat *value)`
- `void glUniform1iv(GLint location, GLsizei count, const GLint *value)`
- `void glUniform2iv(GLint location, GLsizei count, const GLint *value)`
- `void glUniform3iv(GLint location, GLsizei count, const GLint *value)`
- `void glUniform4iv(GLint location, GLsizei count, const GLint *value)`
- `void glUniformMatrix2fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix3fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix4fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix2x3fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix3x2fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix2x4fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix4x2fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix3x4fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix4x3fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUseProgram(GLuint program)`
- `void glValidateProgram(GLuint program)`
- `void glVertex2s(GLshort x, GLshort y)`
- `void glVertex2i(GLint x, GLint y)`
- `void glVertex2f(GLfloat x, GLfloat y)`
- `void glVertex2d(GLdouble x, GLdouble y)`
- `void glVertex3s(GLshort x, GLshort y, GLshort z)`
- `void glVertex3i(GLint x, GLint y, GLint z)`
- `void glVertex3f(GLfloat x, GLfloat y, GLfloat z)`
- `void glVertex3d(GLdouble x, GLdouble y, GLdouble z)`
- `void glVertex4s(GLshort x, GLshort y, GLshort z, GLshort w)`
- `void glVertex4i(GLint x, GLint y, GLint z, GLint w)`
- `void glVertex4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)`
- `void glVertex4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)`
- `void glVertex2sv(const GLshort * v)`
- `void glVertex2iv(const GLint * v)`
- `void glVertex2fv(const GLfloat * v)`
- `void glVertex2dv(const GLdouble * v)`
- `void glVertex3sv(const GLshort * v)`

- `void glVertex3iv(const GLint * v)`
- `void glVertex3fv(const GLfloat * v)`
- `void glVertex3dv(const GLdouble * v)`
- `void glVertex4sv(const GLshort * v)`
- `void glVertex4iv(const GLint * v)`
- `void glVertex4fv(const GLfloat * v)`
- `void glVertex4dv(const GLdouble * v)`
- `void glVertexAttrib1f(GLuint index, GLfloat v0)`
- `void glVertexAttrib1s(GLuint index, GLshort v0)`
- `void glVertexAttrib1d(GLuint index, GLdouble v0)`
- `void glVertexAttrib2f(GLuint index, GLfloat v0, GLfloat v1)`
- `void glVertexAttrib2s(GLuint index, GLshort v0, GLshort v1)`
- `void glVertexAttrib2d(GLuint index, GLdouble v0, GLdouble v1)`
- `void glVertexAttrib3f(GLuint index, GLfloat v0, GLfloat v1, GLfloat v2)`
- `void glVertexAttrib3s(GLuint index, GLshort v0, GLshort v1, GLshort v2)`
- `void glVertexAttrib3d(GLuint index, GLdouble v0, GLdouble v1, GLdouble v2)`
- `void glVertexAttrib4f(GLuint index, GLfloat v0, GLfloat v1, GLfloat v2, GLfloat v3)`
- `void glVertexAttrib4s(GLuint index, GLshort v0, GLshort v1, GLshort v2, GLshort v3)`
- `void glVertexAttrib4d(GLuint index, GLdouble v0, GLdouble v1, GLdouble v2, GLdouble v3)`
- `void glVertexAttrib4Nub(GLuint index, GLubyte v0, GLubyte v1, GLubyte v2, GLubyte v3)`
- `void glVertexAttrib1fv(GLuint index, const GLfloat *v)`
- `void glVertexAttrib1sv(GLuint index, const GLshort *v)`
- `void glVertexAttrib1dv(GLuint index, const GLdouble *v)`
- `void glVertexAttrib2fv(GLuint index, const GLfloat *v)`
- `void glVertexAttrib2sv(GLuint index, const GLshort *v)`
- `void glVertexAttrib2dv(GLuint index, const GLdouble *v)`
- `void glVertexAttrib3fv(GLuint index, const GLfloat *v)`
- `void glVertexAttrib3sv(GLuint index, const GLshort *v)`
- `void glVertexAttrib3dv(GLuint index, const GLdouble *v)`
- `void glVertexAttrib4fv(GLuint index, const GLfloat *v)`
- `void glVertexAttrib4sv(GLuint index, const GLshort *v)`
- `void glVertexAttrib4dv(GLuint index, const GLdouble *v)`
- `void glVertexAttrib4iv(GLuint index, const GLint *v)`
- `void glVertexAttrib4bv(GLuint index, const GLbyte *v)`
- `void glVertexAttrib4ubv(GLuint index, const GLubyte *v)`
- `void glVertexAttrib4usv(GLuint index, const GLushort *v)`

- void glVertexAttrib4uiv(GLuint index,const GLuint *v)
- void glVertexAttribPointer(GLuint index,GLint size,GLenum type,GLboolean normalized,GLsizei stride,const GLvoid * pointer)
- void glVertexPointer(GLint size,GLenum type,GLsizei stride,const GLvoid * pointer)
- void glViewport(GLint x,GLint y,GLsizei width,GLsizei height)
- void glWindowPos2s(GLshort x,GLshort y)
- void glWindowPos2i(GLint x,GLint y)
- void glWindowPos2f(GLfloat x,GLfloat y)
- void glWindowPos2d(GLdouble x,GLdouble y)
- void glWindowPos3s(GLshort x,GLshort y,GLshort z)
- void glWindowPos3i(GLint x,GLint y,GLint z)
- void glWindowPos3f(GLfloat x,GLfloat y,GLfloat z)
- void glWindowPos3d(GLdouble x,GLdouble y,GLdouble z)
- void glWindowPos2sv(const GLshort * v)
- void glWindowPos2iv(const GLint * v)
- void glWindowPos2fv(const GLfloat * v)
- void glWindowPos2dv(const GLdouble * v)
- void glWindowPos3sv(const GLshort * v)
- void glWindowPos3iv(const GLint * v)
- void glWindowPos3fv(const GLfloat * v)
- void glWindowPos3dv(const GLdouble * v)
- void gluBeginCurve(GLUnurbs* nurb)
- void gluBeginPolygon(GLUtesselator* tess)
- void gluBeginSurface(GLUnurbs* nurb)
- void gluBeginTrim(GLUnurbs* nurb)
- void gluCylinder(GLUquadric* quad,GLdouble base,GLdouble top,GLdouble height,GLint slices,GLint stacks)
- void gluDeleteNurbsRenderer(GLUnurbs* nurb)
- void gluDeleteQuadric(GLUquadric* quad)
- void gluDeleteTess(GLUtesselator* tess)
- void gluDisk(GLUquadric* quad,GLdouble inner,GLdouble outer,GLint slices,GLint loops)
- void gluEndCurve(GLUnurbs* nurb)
- void gluEndPolygon(GLUtesselator* tess)
- void gluEndSurface(GLUnurbs* nurb)
- void gluEndTrim(GLUnurbs* nurb)
- const GLubyte * gluErrorString(GLenum error)
- void gluGetNurbsProperty(GLUnurbs* nurb,GLenum property,GLfloat* data)

- `const GLubyte * gluGetString(GLenum name)`
- `void gluGetTessProperty(GLUtesselator* tess, GLenum which, GLdouble* data)`
- `void gluLoadSamplingMatrices(GLUnurbs* nurb, const GLfloat * model, const GLfloat * perspective, const GLint * view)`
- `void gluLookAt(GLdouble eyeX, GLdouble eyeY, GLdouble eyeZ, GLdouble centerX, GLdouble centerY, GLdouble centerZ, GLdouble upX, GLdouble upY, GLdouble upZ)`
- `GLUnurbs *gluNewNurbsRenderer(void)`
- `GLUquadric *gluNewQuadric(void)`
- `GLUtesselator* gluNewTess(void)`
- `void gluNextContour(GLUtesselator* tess, GLenum type)`
- `void gluNurbsCurve(GLUnurbs* nurb, GLint knotCount, GLfloat * knots, GLint stride, GLfloat * control, GLint order, GLenum type)`
- `void gluNurbsProperty(GLUnurbs* nurb, GLenum property, GLfloat value)`
- `void gluNurbsSurface(GLUnurbs* nurb, GLint sKnotCount, GLfloat* sKnots, GLint tKnotCount, GLfloat* tKnots, GLint sStride, GLint tStride, GLfloat* control, GLint sOrder, GLint tOrder, GLenum type)`
- `void gluOrtho2D(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top)`
- `void gluPartialDisk(GLUquadric* quad, GLdouble inner, GLdouble outer, GLint slices, GLint loops, GLdouble start, GLdouble sweep)`
- `void gluPerspective(GLdouble fovy, GLdouble aspect, GLdouble zNear, GLdouble zFar)`
- `void gluPickMatrix(GLdouble x, GLdouble y, GLdouble delX, GLdouble delY, GLint * viewport)`
- `GLint gluProject(GLdouble objX, GLdouble objY, GLdouble objZ, const GLdouble * model, const GLdouble * proj, const GLint * view, GLdouble* winX, GLdouble* winY, GLdouble* winZ)`
- `void gluPwlCurve(GLUnurbs* nurb, GLint count, GLfloat* data, GLint stride, GLenum type)`
- `void gluQuadricDrawStyle(GLUquadric* quad, GLenum draw)`
- `void gluQuadricNormals(GLUquadric* quad, GLenum normal)`
- `void gluQuadricOrientation(GLUquadric* quad, GLenum orientation)`
- `void gluQuadricTexture(GLUquadric* quad, GLboolean texture)`
- `GLint gluScaleImage(GLenum format, GLsizei wIn, GLsizei hIn, GLenum typeIn, const void * dataIn, GLsizei wOut, GLsizei hOut, GLenum typeOut, GLvoid* dataOut)`
- `void gluSphere(GLUquadric* quad, GLdouble radius, GLint slices, GLint stacks)`
- `void gluTessBeginContour(GLUtesselator* tess)`
- `void gluTessBeginPolygon(GLUtesselator* tess, GLvoid* data)`
- `void gluTessEndContour(GLUtesselator* tess)`
- `void gluTessEndPolygon(GLUtesselator* tess)`
- `void gluTessNormal(GLUtesselator* tess, GLdouble valueX, GLdouble valueY, GLdouble valueZ)`
- `void gluTessProperty(GLUtesselator* tess, GLenum which, GLdouble data)`
- `void gluTessVertex(GLUtesselator* tess, GLdouble * location, GLvoid* data)`
- `GLint gluUnProject(GLdouble winX, GLdouble winY, GLdouble winZ, const GLdouble * model, const GLdouble * proj, const GLint * view, GLdouble* objX, GLdouble* objY, GLdouble* objZ)`

- `void glDisable(GLenum cap)`

RINGOPENGL (OPENGL 3.3) FUNCTIONS REFERENCE

- GL_ZERO
- GL_FALSE
- GL_LOGIC_OP
- GL_NONE
- GL_TEXTURE_COMPONENTS
- GL_NO_ERROR
- GL_POINTS
- GL_CURRENT_BIT
- GL_TRUE
- GL_ONE
- GL_CLIENT_PIXEL_STORE_BIT
- GL_LINES
- GL_LINE_LOOP
- GL_POINT_BIT
- GL_CLIENT_VERTEX_ARRAY_BIT
- GL_LINE_STRIP
- GL_LINE_BIT
- GL_TRIANGLES
- GL_TRIANGLE_STRIP
- GL_TRIANGLE_FAN
- GL_QUADS
- GL_QUAD_STRIP
- GL_POLYGON_BIT
- GL_POLYGON
- GL_POLYGON_STIPPLE_BIT
- GL_PIXEL_MODE_BIT
- GL_LIGHTING_BIT

- GL_FOG_BIT
- GL_DEPTH_BUFFER_BIT
- GL_ACCUM
- GL_LOAD
- GL_RETURN
- GL_MULT
- GL_ADD
- GL_NEVER
- GL_ACCUM_BUFFER_BIT
- GL_LESS
- GL_EQUAL
- GL_LEQUAL
- GL_GREATER
- GL_NOTEQUAL
- GL_GEQUAL
- GL_ALWAYS
- GL_SRC_COLOR
- GL_ONE_MINUS_SRC_COLOR
- GL_SRC_ALPHA
- GL_ONE_MINUS_SRC_ALPHA
- GL_DST_ALPHA
- GL_ONE_MINUS_DST_ALPHA
- GL_DST_COLOR
- GL_ONE_MINUS_DST_COLOR
- GL_SRC_ALPHA_SATURATE
- GL_STENCIL_BUFFER_BIT
- GL_FRONT_LEFT
- GL_FRONT_RIGHT
- GL_BACK_LEFT
- GL_BACK_RIGHT
- GL_FRONT
- GL_BACK
- GL_LEFT
- GL_RIGHT
- GL_FRONT_AND_BACK
- GL_AUX0

- GL_AUX1
- GL_AUX2
- GL_AUX3
- GL_INVALID_ENUM
- GL_INVALID_VALUE
- GL_INVALID_OPERATION
- GL_STACK_OVERFLOW
- GL_STACK_UNDERFLOW
- GL_OUT_OF_MEMORY
- GL_2D
- GL_3D
- GL_3D_COLOR
- GL_3D_COLOR_TEXTURE
- GL_4D_COLOR_TEXTURE
- GL_PASS_THROUGH_TOKEN
- GL_POINT_TOKEN
- GL_LINE_TOKEN
- GL_POLYGON_TOKEN
- GL_BITMAP_TOKEN
- GL_DRAW_PIXEL_TOKEN
- GL_COPY_PIXEL_TOKEN
- GL_LINE_RESET_TOKEN
- GL_EXP
- GL_VIEWPORT_BIT
- GL_EXP2
- GL_CW
- GL_CCW
- GL_COEFF
- GL_ORDER
- GL_DOMAIN
- GL_CURRENT_COLOR
- GL_CURRENT_INDEX
- GL_CURRENT_NORMAL
- GL_CURRENT_TEXTURE_COORDS
- GL_CURRENT_RASTER_COLOR
- GL_CURRENT_RASTER_INDEX

- GL_CURRENT_RASTER_TEXTURE_COORDS
- GL_CURRENT_RASTER_POSITION
- GL_CURRENT_RASTER_POSITION_VALID
- GL_CURRENT_RASTER_DISTANCE
- GL_POINT_SMOOTH
- GL_POINT_SIZE
- GL_POINT_SIZE_RANGE
- GL_POINT_SIZE_GRANULARITY
- GL_LINE_SMOOTH
- GL_LINE_WIDTH
- GL_LINE_WIDTH_RANGE
- GL_LINE_WIDTH_GRANULARITY
- GL_LINE_STIPPLE
- GL_LINE_STIPPLE_PATTERN
- GL_LINE_STIPPLE_REPEAT
- GL_LIST_MODE
- GL_MAX_LIST_NESTING
- GL_LIST_BASE
- GL_LIST_INDEX
- GL_POLYGON_MODE
- GL_POLYGON_SMOOTH
- GL_POLYGON_STIPPLE
- GL_EDGE_FLAG
- GL_CULL_FACE
- GL_CULL_FACE_MODE
- GL_FRONT_FACE
- GL_LIGHTING
- GL_LIGHT_MODEL_LOCAL_VIEWER
- GL_LIGHT_MODEL_TWO_SIDE
- GL_LIGHT_MODEL_AMBIENT
- GL_SHADE_MODEL
- GL_COLOR_MATERIAL_FACE
- GL_COLOR_MATERIAL_PARAMETER
- GL_COLOR_MATERIAL
- GL_FOG
- GL_FOG_INDEX

- GL_FOG_DENSITY
- GL_FOG_START
- GL_FOG_END
- GL_FOG_MODE
- GL_FOG_COLOR
- GL_DEPTH_RANGE
- GL_DEPTH_TEST
- GL_DEPTH_WRITEMASK
- GL_DEPTH_CLEAR_VALUE
- GL_DEPTH_FUNC
- GL_ACCUM_CLEAR_VALUE
- GL_STENCIL_TEST
- GL_STENCIL_CLEAR_VALUE
- GL_STENCIL_FUNC
- GL_STENCIL_VALUE_MASK
- GL_STENCIL_FAIL
- GL_STENCIL_PASS_DEPTH_FAIL
- GL_STENCIL_PASS_DEPTH_PASS
- GL_STENCIL_REF
- GL_STENCIL_WRITEMASK
- GL_MATRIX_MODE
- GL_NORMALIZE
- GL_VIEWPORT
- GL_MODELVIEW_STACK_DEPTH
- GL_PROJECTION_STACK_DEPTH
- GL_TEXTURE_STACK_DEPTH
- GL_MODELVIEW_MATRIX
- GL_PROJECTION_MATRIX
- GL_TEXTURE_MATRIX
- GL_ATTRIB_STACK_DEPTH
- GL_CLIENT_ATTRIB_STACK_DEPTH
- GL_ALPHA_TEST
- GL_ALPHA_TEST_FUNC
- GL_ALPHA_TEST_REF
- GL_DITHER
- GL_BLEND_DST

- GL_BLEND_SRC
- GL_BLEND
- GL_LOGIC_OP_MODE
- GL_INDEX_LOGIC_OP
- GL_COLOR_LOGIC_OP
- GL_AUX_BUFFERS
- GL_DRAW_BUFFER
- GL_READ_BUFFER
- GL_SCISSOR_BOX
- GL_SCISSOR_TEST
- GL_INDEX_CLEAR_VALUE
- GL_INDEX_WRITEMASK
- GL_COLOR_CLEAR_VALUE
- GL_COLOR_WRITEMASK
- GL_INDEX_MODE
- GL_RGBA_MODE
- GL_DOUBLEBUFFER
- GL_STEREO
- GL_RENDER_MODE
- GL_PERSPECTIVE_CORRECTION_HINT
- GL_POINT_SMOOTH_HINT
- GL_LINE_SMOOTH_HINT
- GL_POLYGON_SMOOTH_HINT
- GL_FOG_HINT
- GL_TEXTURE_GEN_S
- GL_TEXTURE_GEN_T
- GL_TEXTURE_GEN_R
- GL_TEXTURE_GEN_Q
- GL_PIXEL_MAP_I_TO_I
- GL_PIXEL_MAP_S_TO_S
- GL_PIXEL_MAP_I_TO_R
- GL_PIXEL_MAP_I_TO_G
- GL_PIXEL_MAP_I_TO_B
- GL_PIXEL_MAP_I_TO_A
- GL_PIXEL_MAP_R_TO_R
- GL_PIXEL_MAP_G_TO_G

- GL_PIXEL_MAP_B_TO_B
- GL_PIXEL_MAP_A_TO_A
- GL_PIXEL_MAP_I_TO_I_SIZE
- GL_PIXEL_MAP_S_TO_S_SIZE
- GL_PIXEL_MAP_I_TO_R_SIZE
- GL_PIXEL_MAP_I_TO_G_SIZE
- GL_PIXEL_MAP_I_TO_B_SIZE
- GL_PIXEL_MAP_I_TO_A_SIZE
- GL_PIXEL_MAP_R_TO_R_SIZE
- GL_PIXEL_MAP_G_TO_G_SIZE
- GL_PIXEL_MAP_B_TO_B_SIZE
- GL_PIXEL_MAP_A_TO_A_SIZE
- GL_UNPACK_SWAP_BYTES
- GL_UNPACK_LSB_FIRST
- GL_UNPACK_ROW_LENGTH
- GL_UNPACK_SKIP_ROWS
- GL_UNPACK_SKIP_PIXELS
- GL_UNPACK_ALIGNMENT
- GL_PACK_SWAP_BYTES
- GL_PACK_LSB_FIRST
- GL_PACK_ROW_LENGTH
- GL_PACK_SKIP_ROWS
- GL_PACK_SKIP_PIXELS
- GL_PACK_ALIGNMENT
- GL_MAP_COLOR
- GL_MAP_STENCIL
- GL_INDEX_SHIFT
- GL_INDEX_OFFSET
- GL_RED_SCALE
- GL_RED_BIAS
- GL_ZOOM_X
- GL_ZOOM_Y
- GL_GREEN_SCALE
- GL_GREEN_BIAS
- GL_BLUE_SCALE
- GL_BLUE_BIAS

- GL_ALPHA_SCALE
- GL_ALPHA_BIAS
- GL_DEPTH_SCALE
- GL_DEPTH_BIAS
- GL_MAX_EVAL_ORDER
- GL_MAX_LIGHTS
- GL_MAX_CLIP_PLANES
- GL_MAX_TEXTURE_SIZE
- GL_MAX_PIXEL_MAP_TABLE
- GL_MAX_ATTRIB_STACK_DEPTH
- GL_MAX_MODELVIEW_STACK_DEPTH
- GL_MAX_NAME_STACK_DEPTH
- GL_MAX_PROJECTION_STACK_DEPTH
- GL_MAX_TEXTURE_STACK_DEPTH
- GL_MAX_VIEWPORT_DIMS
- GL_MAX_CLIENT_ATTRIB_STACK_DEPTH
- GL_SUBPIXEL_BITS
- GL_INDEX_BITS
- GL_RED_BITS
- GL_GREEN_BITS
- GL_BLUE_BITS
- GL_ALPHA_BITS
- GL_DEPTH_BITS
- GL_STENCIL_BITS
- GL_ACCUM_RED_BITS
- GL_ACCUM_GREEN_BITS
- GL_ACCUM_BLUE_BITS
- GL_ACCUM_ALPHA_BITS
- GL_NAME_STACK_DEPTH
- GL_AUTO_NORMAL
- GL_MAP1_COLOR_4
- GL_MAP1_INDEX
- GL_MAP1_NORMAL
- GL_MAP1_TEXTURE_COORD_1
- GL_MAP1_TEXTURE_COORD_2
- GL_MAP1_TEXTURE_COORD_3

- GL_MAP1_TEXTURE_COORD_4
- GL_MAP1_VERTEX_3
- GL_MAP1_VERTEX_4
- GL_MAP2_COLOR_4
- GL_MAP2_INDEX
- GL_MAP2_NORMAL
- GL_MAP2_TEXTURE_COORD_1
- GL_MAP2_TEXTURE_COORD_2
- GL_MAP2_TEXTURE_COORD_3
- GL_MAP2_TEXTURE_COORD_4
- GL_MAP2_VERTEX_3
- GL_MAP2_VERTEX_4
- GL_MAP1_GRID_DOMAIN
- GL_MAP1_GRID_SEGMENTS
- GL_MAP2_GRID_DOMAIN
- GL_MAP2_GRID_SEGMENTS
- GL_TEXTURE_1D
- GL_TEXTURE_2D
- GL_FEEDBACK_BUFFER_POINTER
- GL_FEEDBACK_BUFFER_SIZE
- GL_FEEDBACK_BUFFER_TYPE
- GL_SELECTION_BUFFER_POINTER
- GL_SELECTION_BUFFER_SIZE
- GL_TEXTURE_WIDTH
- GL_TRANSFORM_BIT
- GL_TEXTURE_HEIGHT
- GL_TEXTURE_INTERNAL_FORMAT
- GL_TEXTURE_BORDER_COLOR
- GL_TEXTURE_BORDER
- GL_DONT_CARE
- GL_FASTEST
- GL_NICEST
- GL_AMBIENT
- GL_DIFFUSE
- GL_SPECULAR
- GL_POSITION

- GL_SPOT_DIRECTION
- GL_SPOT_EXPONENT
- GL_SPOT_CUTOFF
- GL_CONSTANT_ATTENUATION
- GL_LINEAR_ATTENUATION
- GL_QUADRATIC_ATTENUATION
- GL_COMPILE
- GL_COMPILE_AND_EXECUTE
- GL_BYTE
- GL_UNSIGNED_BYTE
- GL_SHORT
- GL_UNSIGNED_SHORT
- GL_INT
- GL_UNSIGNED_INT
- GL_FLOAT
- GL_2_BYTES
- GL_3_BYTES
- GL_4_BYTES
- GL_DOUBLE
- GL_CLEAR
- GL_AND
- GL_AND_REVERSE
- GL_COPY
- GL_AND_INVERTED
- GL_NOOP
- GL_XOR
- GL_OR
- GL_NOR
- GL_EQUIV
- GL_INVERT
- GL_OR_REVERSE
- GL_COPY_INVERTED
- GL_OR_INVERTED
- GL_NAND
- GL_SET
- GL_EMISSION

- GL_SHININESS
- GL_AMBIENT_AND_DIFFUSE
- GL_COLOR_INDEXES
- GL_MODELVIEW
- GL_PROJECTION
- GL_TEXTURE
- GL_COLOR
- GL_DEPTH
- GL_STENCIL
- GL_COLOR_INDEX
- GL_STENCIL_INDEX
- GL_DEPTH_COMPONENT
- GL_RED
- GL_GREEN
- GL_BLUE
- GL_ALPHA
- GL_RGB
- GL_RGBA
- GL_LUMINANCE
- GL_LUMINANCE_ALPHA
- GL_BITMAP
- GL_POINT
- GL_LINE
- GL_FILL
- GL_RENDER
- GL_FEEDBACK
- GL_SELECT
- GL_FLAT
- GL_SMOOTH
- GL_KEEP
- GL_REPLACE
- GL_INCR
- GL_DECR
- GL_VENDOR
- GL_RENDERER
- GL_VERSION

- GL_EXTENSIONS
- GL_S
- GL_ENABLE_BIT
- GL_T
- GL_R
- GL_Q
- GL_MODULATE
- GL_DECAL
- GL_TEXTURE_ENV_MODE
- GL_TEXTURE_ENV_COLOR
- GL_TEXTURE_ENV
- GL_EYE_LINEAR
- GL_OBJECT_LINEAR
- GL_SPHERE_MAP
- GL_TEXTURE_GEN_MODE
- GL_OBJECT_PLANE
- GL_EYE_PLANE
- GL_NEAREST
- GL_LINEAR
- GL_NEAREST_MIPMAP_NEAREST
- GL_LINEAR_MIPMAP_NEAREST
- GL_NEAREST_MIPMAP_LINEAR
- GL_LINEAR_MIPMAP_LINEAR
- GL_TEXTURE_MAG_FILTER
- GL_TEXTURE_MIN_FILTER
- GL_TEXTURE_WRAP_S
- GL_TEXTURE_WRAP_T
- GL_CLAMP
- GL_REPEAT
- GL_POLYGON_OFFSET_UNITS
- GL_POLYGON_OFFSET_POINT
- GL_POLYGON_OFFSET_LINE
- GL_R3_G3_B2
- GL_V2F
- GL_V3F
- GL_C4UB_V2F

- GL_C4UB_V3F
- GL_C3F_V3F
- GL_N3F_V3F
- GL_C4F_N3F_V3F
- GL_T2F_V3F
- GL_T4F_V4F
- GL_T2F_C4UB_V3F
- GL_T2F_C3F_V3F
- GL_T2F_N3F_V3F
- GL_T2F_C4F_N3F_V3F
- GL_T4F_C4F_N3F_V4F
- GL_CLIP_PLANE0
- GL_CLIP_PLANE1
- GL_CLIP_PLANE2
- GL_CLIP_PLANE3
- GL_CLIP_PLANE4
- GL_CLIP_PLANE5
- GL_LIGHT0
- GL_COLOR_BUFFER_BIT
- GL_LIGHT1
- GL_LIGHT2
- GL_LIGHT3
- GL_LIGHT4
- GL_LIGHT5
- GL_LIGHT6
- GL_LIGHT7
- GL_HINT_BIT
- GL_POLYGON_OFFSET_FILL
- GL_POLYGON_OFFSET_FACTOR
- GL_ALPHA4
- GL_ALPHA8
- GL_ALPHA12
- GL_ALPHA16
- GL_LUMINANCE4
- GL_LUMINANCE8
- GL_LUMINANCE12

- GL_LUMINANCE16
- GL_LUMINANCE4_ALPHA4
- GL_LUMINANCE6_ALPHA2
- GL_LUMINANCE8_ALPHA8
- GL_LUMINANCE12_ALPHA4
- GL_LUMINANCE12_ALPHA12
- GL_LUMINANCE16_ALPHA16
- GL_INTENSITY
- GL_INTENSITY4
- GL_INTENSITY8
- GL_INTENSITY12
- GL_INTENSITY16
- GL_RGB4
- GL_RGB5
- GL_RGB8
- GL_RGB10
- GL_RGB12
- GL_RGB16
- GL_RGBA2
- GL_RGBA4
- GL_RGB5_A1
- GL_RGBA8
- GL_RGB10_A2
- GL_RGBA12
- GL_RGBA16
- GL_TEXTURE_RED_SIZE
- GL_TEXTURE_GREEN_SIZE
- GL_TEXTURE_BLUE_SIZE
- GL_TEXTURE_ALPHA_SIZE
- GL_TEXTURE_LUMINANCE_SIZE
- GL_TEXTURE_INTENSITY_SIZE
- GL_PROXY_TEXTURE_1D
- GL_PROXY_TEXTURE_2D
- GL_TEXTURE_PRIORITY
- GL_TEXTURE_RESIDENT
- GL_TEXTURE_BINDING_1D

- GL_TEXTURE_BINDING_2D
- GL_VERTEX_ARRAY
- GL_NORMAL_ARRAY
- GL_COLOR_ARRAY
- GL_INDEX_ARRAY
- GL_TEXTURE_COORD_ARRAY
- GL_EDGE_FLAG_ARRAY
- GL_VERTEX_ARRAY_SIZE
- GL_VERTEX_ARRAY_TYPE
- GL_VERTEX_ARRAY_STRIDE
- GL_NORMAL_ARRAY_TYPE
- GL_NORMAL_ARRAY_STRIDE
- GL_COLOR_ARRAY_SIZE
- GL_COLOR_ARRAY_TYPE
- GL_COLOR_ARRAY_STRIDE
- GL_INDEX_ARRAY_TYPE
- GL_INDEX_ARRAY_STRIDE
- GL_TEXTURE_COORD_ARRAY_SIZE
- GL_TEXTURE_COORD_ARRAY_TYPE
- GL_TEXTURE_COORD_ARRAY_STRIDE
- GL_EDGE_FLAG_ARRAY_STRIDE
- GL_VERTEX_ARRAY_POINTER
- GL_NORMAL_ARRAY_POINTER
- GL_COLOR_ARRAY_POINTER
- GL_INDEX_ARRAY_POINTER
- GL_TEXTURE_COORD_ARRAY_POINTER
- GL_EDGE_FLAG_ARRAY_POINTER
- GL_COLOR_INDEX1_EXT
- GL_COLOR_INDEX2_EXT
- GL_COLOR_INDEX4_EXT
- GL_COLOR_INDEX8_EXT
- GL_COLOR_INDEX12_EXT
- GL_COLOR_INDEX16_EXT
- GL_EVAL_BIT
- GL_LIST_BIT
- GL_TEXTURE_BIT

- GL_SCISSOR_BIT
- GL_ALL_ATTRIB_BITS
- GL_CLIENT_ALL_ATTRIB_BITS
- GL_SMOOTH_POINT_SIZE_RANGE
- GL_SMOOTH_POINT_SIZE_GRANULARITY
- GL_SMOOTH_LINE_WIDTH_RANGE
- GL_SMOOTH_LINE_WIDTH_GRANULARITY
- GL_UNSIGNED_BYTE_3_3_2
- GL_UNSIGNED_SHORT_4_4_4_4
- GL_UNSIGNED_SHORT_5_5_5_1
- GL_UNSIGNED_INT_8_8_8_8
- GL_UNSIGNED_INT_10_10_10_2
- GL_RESCALE_NORMAL
- GL_TEXTURE_BINDING_3D
- GL_PACK_SKIP_IMAGES
- GL_PACK_IMAGE_HEIGHT
- GL_UNPACK_SKIP_IMAGES
- GL_UNPACK_IMAGE_HEIGHT
- GL_TEXTURE_3D
- GL_PROXY_TEXTURE_3D
- GL_TEXTURE_DEPTH
- GL_TEXTURE_WRAP_R
- GL_MAX_3D_TEXTURE_SIZE
- GL_BGR
- GL_BGRA
- GL_MAX_ELEMENTS_VERTICES
- GL_MAX_ELEMENTS_INDICES
- GL_CLAMP_TO_EDGE
- GL_TEXTURE_MIN_LOD
- GL_TEXTURE_MAX_LOD
- GL_TEXTURE_BASE_LEVEL
- GL_TEXTURE_MAX_LEVEL
- GL_LIGHT_MODEL_COLOR_CONTROL
- GL_SINGLE_COLOR
- GL_SEPARATE_SPECULAR_COLOR
- GL_UNSIGNED_BYTE_2_3_3_REV

- GL_UNSIGNED_SHORT_5_6_5
- GL_UNSIGNED_SHORT_5_6_5_REV
- GL_UNSIGNED_SHORT_4_4_4_4_REV
- GL_UNSIGNED_SHORT_1_5_5_5_REV
- GL_UNSIGNED_INT_8_8_8_8_REV
- GL_ALIASED_POINT_SIZE_RANGE
- GL_ALIASED_LINE_WIDTH_RANGE
- GL_MULTISAMPLE
- GL_SAMPLE_ALPHA_TO_COVERAGE
- GL_SAMPLE_ALPHA_TO_ONE
- GL_SAMPLE_COVERAGE
- GL_SAMPLE_BUFFERS
- GL_SAMPLES
- GL_SAMPLE_COVERAGE_VALUE
- GL_SAMPLE_COVERAGE_INVERT
- GL_CLAMP_TO_BORDER
- GL_TEXTURE0
- GL_TEXTURE1
- GL_TEXTURE2
- GL_TEXTURE3
- GL_TEXTURE4
- GL_TEXTURE5
- GL_TEXTURE6
- GL_TEXTURE7
- GL_TEXTURE8
- GL_TEXTURE9
- GL_TEXTURE10
- GL_TEXTURE11
- GL_TEXTURE12
- GL_TEXTURE13
- GL_TEXTURE14
- GL_TEXTURE15
- GL_TEXTURE16
- GL_TEXTURE17
- GL_TEXTURE18
- GL_TEXTURE19

- GL_TEXTURE20
- GL_TEXTURE21
- GL_TEXTURE22
- GL_TEXTURE23
- GL_TEXTURE24
- GL_TEXTURE25
- GL_TEXTURE26
- GL_TEXTURE27
- GL_TEXTURE28
- GL_TEXTURE29
- GL_TEXTURE30
- GL_TEXTURE31
- GL_ACTIVE_TEXTURE
- GL_CLIENT_ACTIVE_TEXTURE
- GL_MAX_TEXTURE_UNITS
- GL_TRANSPOSE_MODELVIEW_MATRIX
- GL_TRANSPOSE_PROJECTION_MATRIX
- GL_TRANSPOSE_TEXTURE_MATRIX
- GL_TRANSPOSE_COLOR_MATRIX
- GL_SUBTRACT
- GL_COMPRESSED_ALPHA
- GL_COMPRESSED_LUMINANCE
- GL_COMPRESSED_LUMINANCE_ALPHA
- GL_COMPRESSED_INTENSITY
- GL_COMPRESSED_RGB
- GL_COMPRESSED_RGBA
- GL_TEXTURE_COMPRESSION_HINT
- GL_NORMAL_MAP
- GL_REFLECTION_MAP
- GL_TEXTURE_CUBE_MAP
- GL_TEXTURE_BINDING_CUBE_MAP
- GL_TEXTURE_CUBE_MAP_POSITIVE_X
- GL_TEXTURE_CUBE_MAP_NEGATIVE_X
- GL_TEXTURE_CUBE_MAP_POSITIVE_Y
- GL_TEXTURE_CUBE_MAP_NEGATIVE_Y
- GL_TEXTURE_CUBE_MAP_POSITIVE_Z

- GL_TEXTURE_CUBE_MAP_NEGATIVE_Z
- GL_PROXY_TEXTURE_CUBE_MAP
- GL_MAX_CUBE_MAP_TEXTURE_SIZE
- GL_COMBINE
- GL_COMBINE_RGB
- GL_COMBINE_ALPHA
- GL_RGB_SCALE
- GL_ADD_SIGNED
- GL_INTERPOLATE
- GL_CONSTANT
- GL_PRIMARY_COLOR
- GL_PREVIOUS
- GL_SOURCE0_RGB
- GL_SOURCE1_RGB
- GL_SOURCE2_RGB
- GL_SOURCE0_ALPHA
- GL_SOURCE1_ALPHA
- GL_SOURCE2_ALPHA
- GL_OPERAND0_RGB
- GL_OPERAND1_RGB
- GL_OPERAND2_RGB
- GL_OPERAND0_ALPHA
- GL_OPERAND1_ALPHA
- GL_OPERAND2_ALPHA
- GL_TEXTURE_COMPRESSED_IMAGE_SIZE
- GL_TEXTURE_COMPRESSED
- GL_NUM_COMPRESSED_TEXTURE_FORMATS
- GL_COMPRESSED_TEXTURE_FORMATS
- GL_DOT3_RGB
- GL_DOT3_RGBA
- GL_MULTISAMPLE_BIT
- GL_BLEND_DST_RGB
- GL_BLEND_SRC_RGB
- GL_BLEND_DST_ALPHA
- GL_BLEND_SRC_ALPHA
- GL_POINT_SIZE_MIN

- GL_POINT_SIZE_MAX
- GL_POINT_FADE_THRESHOLD_SIZE
- GL_POINT_DISTANCE_ATTENUATION
- GL_GENERATE_MIPMAP
- GL_GENERATE_MIPMAP_HINT
- GL_DEPTH_COMPONENT16
- GL_DEPTH_COMPONENT24
- GL_DEPTH_COMPONENT32
- GL_MIRRORED_REPEAT
- GL_FOG_COORDINATE_SOURCE
- GL_FOG_COORDINATE
- GL_FRAGMENT_DEPTH
- GL_CURRENT_FOG_COORDINATE
- GL_FOG_COORDINATE_ARRAY_TYPE
- GL_FOG_COORDINATE_ARRAY_STRIDE
- GL_FOG_COORDINATE_ARRAY_POINTER
- GL_FOG_COORDINATE_ARRAY
- GL_COLOR_SUM
- GL_CURRENT_SECONDARY_COLOR
- GL_SECONDARY_COLOR_ARRAY_SIZE
- GL_SECONDARY_COLOR_ARRAY_TYPE
- GL_SECONDARY_COLOR_ARRAY_STRIDE
- GL_SECONDARY_COLOR_ARRAY_POINTER
- GL_SECONDARY_COLOR_ARRAY
- GL_MAX_TEXTURE_LOD_BIAS
- GL_TEXTURE_FILTER_CONTROL
- GL_TEXTURE_LOD_BIAS
- GL_INCR_WRAP
- GL_DECR_WRAP
- GL_TEXTURE_DEPTH_SIZE
- GL_DEPTH_TEXTURE_MODE
- GL_TEXTURE_COMPARE_MODE
- GL_TEXTURE_COMPARE_FUNC
- GL_COMPARE_R_TO_TEXTURE
- GL_CURRENT_FOG_COORD
- GL_FOG_COORD

- GL_FOG_COORD_ARRAY
- GL_FOG_COORD_ARRAY_BUFFER_BINDING
- GL_FOG_COORD_ARRAY_POINTER
- GL_FOG_COORD_ARRAY_STRIDE
- GL_FOG_COORD_ARRAY_TYPE
- GL_FOG_COORD_SRC
- GL_SRC0_ALPHA
- GL_SRC0_RGB
- GL_SRC1_ALPHA
- GL_SRC1_RGB
- GL_SRC2_ALPHA
- GL_SRC2_RGB
- GL_BUFFER_SIZE
- GL_BUFFER_USAGE
- GL_QUERY_COUNTER_BITS
- GL_CURRENT_QUERY
- GL_QUERY_RESULT
- GL_QUERY_RESULT_AVAILABLE
- GL_ARRAY_BUFFER
- GL_ELEMENT_ARRAY_BUFFER
- GL_ARRAY_BUFFER_BINDING
- GL_ELEMENT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ARRAY_BUFFER_BINDING
- GL_NORMAL_ARRAY_BUFFER_BINDING
- GL_COLOR_ARRAY_BUFFER_BINDING
- GL_INDEX_ARRAY_BUFFER_BINDING
- GL_TEXTURE_COORD_ARRAY_BUFFER_BINDING
- GL_EDGE_FLAG_ARRAY_BUFFER_BINDING
- GL_SECONDARY_COLOR_ARRAY_BUFFER_BINDING
- GL_FOG_COORDINATE_ARRAY_BUFFER_BINDING
- GL_WEIGHT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ATTRIB_ARRAY_BUFFER_BINDING
- GL_READ_ONLY
- GL_WRITE_ONLY
- GL_READ_WRITE
- GL_BUFFER_ACCESS

- GL_BUFFER_MAPPED
- GL_BUFFER_MAP_POINTER
- GL_STREAM_DRAW
- GL_STREAM_READ
- GL_STREAM_COPY
- GL_STATIC_DRAW
- GL_STATIC_READ
- GL_STATIC_COPY
- GL_DYNAMIC_DRAW
- GL_DYNAMIC_READ
- GL_DYNAMIC_COPY
- GL_SAMPLES_PASSED
- GL_BLEND_EQUATION_RGB
- GL_VERTEX_ATTRIB_ARRAY_ENABLED
- GL_VERTEX_ATTRIB_ARRAY_SIZE
- GL_VERTEX_ATTRIB_ARRAY_STRIDE
- GL_VERTEX_ATTRIB_ARRAY_TYPE
- GL_CURRENT_VERTEX_ATTRIB
- GL_VERTEX_PROGRAM_POINT_SIZE
- GL_VERTEX_PROGRAM_TWO_SIDE
- GL_VERTEX_ATTRIB_ARRAY_POINTER
- GL_STENCIL_BACK_FUNC
- GL_STENCIL_BACK_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_PASS
- GL_MAX_DRAW_BUFFERS
- GL_DRAW_BUFFER0
- GL_DRAW_BUFFER1
- GL_DRAW_BUFFER2
- GL_DRAW_BUFFER3
- GL_DRAW_BUFFER4
- GL_DRAW_BUFFER5
- GL_DRAW_BUFFER6
- GL_DRAW_BUFFER7
- GL_DRAW_BUFFER8
- GL_DRAW_BUFFER9

- GL_DRAW_BUFFER10
- GL_DRAW_BUFFER11
- GL_DRAW_BUFFER12
- GL_DRAW_BUFFER13
- GL_DRAW_BUFFER14
- GL_DRAW_BUFFER15
- GL_BLEND_EQUATION_ALPHA
- GL_POINT_SPRITE
- GL_COORD_REPLACE
- GL_MAX_VERTEX_ATTRIBS
- GL_VERTEX_ATTRIB_ARRAY_NORMALIZED
- GL_MAX_TEXTURE_COORDS
- GL_MAX_TEXTURE_IMAGE_UNITS
- GL_FRAGMENT_SHADER
- GL_VERTEX_SHADER
- GL_MAX_FRAGMENT_UNIFORM_COMPONENTS
- GL_MAX_VERTEX_UNIFORM_COMPONENTS
- GL_MAX_VARYING_FLOATS
- GL_MAX_VERTEX_TEXTURE_IMAGE_UNITS
- GL_MAX_COMBINED_TEXTURE_IMAGE_UNITS
- GL_SHADER_TYPE
- GL_FLOAT_VEC2
- GL_FLOAT_VEC3
- GL_FLOAT_VEC4
- GL_INT_VEC2
- GL_INT_VEC3
- GL_INT_VEC4
- GL_BOOL
- GL_BOOL_VEC2
- GL_BOOL_VEC3
- GL_BOOL_VEC4
- GL_FLOAT_MAT2
- GL_FLOAT_MAT3
- GL_FLOAT_MAT4
- GL_SAMPLER_1D
- GL_SAMPLER_2D

- GL_SAMPLER_3D
- GL_SAMPLER_CUBE
- GL_SAMPLER_1D_SHADOW
- GL_SAMPLER_2D_SHADOW
- GL_DELETE_STATUS
- GL_COMPILE_STATUS
- GL_LINK_STATUS
- GL_VALIDATE_STATUS
- GL_INFO_LOG_LENGTH
- GL_ATTACHED_SHADERS
- GL_ACTIVE_UNIFORMS
- GL_ACTIVE_UNIFORM_MAX_LENGTH
- GL_SHADER_SOURCE_LENGTH
- GL_ACTIVE_ATTRIBUTES
- GL_ACTIVE_ATTRIBUTE_MAX_LENGTH
- GL_FRAGMENT_SHADER_DERIVATIVE_HINT
- GL_SHADING_LANGUAGE_VERSION
- GL_CURRENT_PROGRAM
- GL_POINT_SPRITE_COORD_ORIGIN
- GL_LOWER_LEFT
- GL_UPPER_LEFT
- GL_STENCIL_BACK_REF
- GL_STENCIL_BACK_VALUE_MASK
- GL_STENCIL_BACK_WRITEMASK
- GL_CURRENT_RASTER_SECONDARY_COLOR
- GL_PIXEL_PACK_BUFFER
- GL_PIXEL_UNPACK_BUFFER
- GL_PIXEL_PACK_BUFFER_BINDING
- GL_PIXEL_UNPACK_BUFFER_BINDING
- GL_FLOAT_MAT2x3
- GL_FLOAT_MAT2x4
- GL_FLOAT_MAT3x2
- GL_FLOAT_MAT3x4
- GL_FLOAT_MAT4x2
- GL_FLOAT_MAT4x3
- GL_SRGB

- GL_SRGB8
- GL_SRGB_ALPHA
- GL_SRGB8_ALPHA8
- GL_SLUMINANCE_ALPHA
- GL_SLUMINANCE8_ALPHA8
- GL_SLUMINANCE
- GL_SLUMINANCE8
- GL_COMPRESSED_SRGB
- GL_COMPRESSED_SRGB_ALPHA
- GL_COMPRESSED_SLUMINANCE
- GL_COMPRESSED_SLUMINANCE_ALPHA
- GL_CLIP_DISTANCE0
- GL_CLIP_DISTANCE1
- GL_CLIP_DISTANCE2
- GL_CLIP_DISTANCE3
- GL_CLIP_DISTANCE4
- GL_CLIP_DISTANCE5
- GL_COMPARE_REF_TO_TEXTURE
- GL_MAX_CLIP_DISTANCES
- GL_MAX_VARYING_COMPONENTS
- GL_CONTEXT_FLAG_FORWARD_COMPATIBLE_BIT
- GL_MAJOR_VERSION
- GL_MINOR_VERSION
- GL_NUM_EXTENSIONS
- GL_CONTEXT_FLAGS
- GL_DEPTH_BUFFER
- GL_STENCIL_BUFFER
- GL_RGBA32F
- GL_RGB32F
- GL_RGBA16F
- GL_RGB16F
- GL_VERTEX_ATTRIB_ARRAY_INTEGER
- GL_MAX_ARRAY_TEXTURE_LAYERS
- GL_MIN_PROGRAM_TEXEL_OFFSET
- GL_MAX_PROGRAM_TEXEL_OFFSET
- GL_CLAMP_VERTEX_COLOR

- GL_CLAMP_FRAGMENT_COLOR
- GL_CLAMP_READ_COLOR
- GL_FIXED_ONLY
- GL_TEXTURE_RED_TYPE
- GL_TEXTURE_GREEN_TYPE
- GL_TEXTURE_BLUE_TYPE
- GL_TEXTURE_ALPHA_TYPE
- GL_TEXTURE_LUMINANCE_TYPE
- GL_TEXTURE_INTENSITY_TYPE
- GL_TEXTURE_DEPTH_TYPE
- GL_TEXTURE_1D_ARRAY
- GL_PROXY_TEXTURE_1D_ARRAY
- GL_TEXTURE_2D_ARRAY
- GL_PROXY_TEXTURE_2D_ARRAY
- GL_TEXTURE_BINDING_1D_ARRAY
- GL_TEXTURE_BINDING_2D_ARRAY
- GL_R11F_G11F_B10F
- GL_UNSIGNED_INT_10F_11F_11F_REV
- GL_RGB9_E5
- GL_UNSIGNED_INT_5_9_9_9_REV
- GL_TEXTURE_SHARED_SIZE
- GL_TRANSFORM_FEEDBACK_VARYING_MAX_LENGTH
- GL_TRANSFORM_FEEDBACK_BUFFER_MODE
- GL_MAX_TRANSFORM_FEEDBACK_SEPARATE_COMPONENTS
- GL_TRANSFORM_FEEDBACK_VARYINGS
- GL_TRANSFORM_FEEDBACK_BUFFER_START
- GL_TRANSFORM_FEEDBACK_BUFFER_SIZE
- GL_PRIMITIVES_GENERATED
- GL_TRANSFORM_FEEDBACK_PRIMITIVES_WRITTEN
- GL_RASTERIZER_DISCARD
- GL_MAX_TRANSFORM_FEEDBACK_INTERLEAVED_COMPONENTS
- GL_MAX_TRANSFORM_FEEDBACK_SEPARATE_ATTRIBS
- GL_INTERLEAVED_ATTRIBS
- GL_SEPARATE_ATTRIBS
- GL_TRANSFORM_FEEDBACK_BUFFER
- GL_TRANSFORM_FEEDBACK_BUFFER_BINDING

- GL_RGBA32UI
- GL_RGB32UI
- GL_RGBA16UI
- GL_RGB16UI
- GL_RGBA8UI
- GL_RGB8UI
- GL_RGBA32I
- GL_RGB32I
- GL_RGBA16I
- GL_RGB16I
- GL_RGBA8I
- GL_RGB8I
- GL_RED_INTEGER
- GL_GREEN_INTEGER
- GL_BLUE_INTEGER
- GL_ALPHA_INTEGER
- GL_RGB_INTEGER
- GL_RGBA_INTEGER
- GL_BGR_INTEGER
- GL_BGRA_INTEGER
- GL_SAMPLER_1D_ARRAY
- GL_SAMPLER_2D_ARRAY
- GL_SAMPLER_1D_ARRAY_SHADOW
- GL_SAMPLER_2D_ARRAY_SHADOW
- GL_SAMPLER_CUBE_SHADOW
- GL_UNSIGNED_INT_VEC2
- GL_UNSIGNED_INT_VEC3
- GL_UNSIGNED_INT_VEC4
- GL_INT_SAMPLER_1D
- GL_INT_SAMPLER_2D
- GL_INT_SAMPLER_3D
- GL_INT_SAMPLER_CUBE
- GL_INT_SAMPLER_1D_ARRAY
- GL_INT_SAMPLER_2D_ARRAY
- GL_UNSIGNED_INT_SAMPLER_1D
- GL_UNSIGNED_INT_SAMPLER_2D

- GL_UNSIGNED_INT_SAMPLER_3D
- GL_UNSIGNED_INT_SAMPLER_CUBE
- GL_UNSIGNED_INT_SAMPLER_1D_ARRAY
- GL_UNSIGNED_INT_SAMPLER_2D_ARRAY
- GL_QUERY_WAIT
- GL_QUERY_NO_WAIT
- GL_QUERY_BY_REGION_WAIT
- GL_QUERY_BY_REGION_NO_WAIT
- GL_TEXTURE_RECTANGLE
- GL_TEXTURE_BINDING_RECTANGLE
- GL_PROXY_TEXTURE_RECTANGLE
- GL_MAX_RECTANGLE_TEXTURE_SIZE
- GL_SAMPLER_2D_RECT
- GL_SAMPLER_2D_RECT_SHADOW
- GL_TEXTURE_BUFFER
- GL_MAX_TEXTURE_BUFFER_SIZE
- GL_TEXTURE_BINDING_BUFFER
- GL_TEXTURE_BUFFER_DATA_STORE_BINDING
- GL_TEXTURE_BUFFER_FORMAT
- GL_SAMPLER_BUFFER
- GL_INT_SAMPLER_2D_RECT
- GL_INT_SAMPLER_BUFFER
- GL_UNSIGNED_INT_SAMPLER_2D_RECT
- GL_UNSIGNED_INT_SAMPLER_BUFFER
- GL_RED_SNORM
- GL_RG_SNORM
- GL_RGB_SNORM
- GL_RGBA_SNORM
- GL_R8_SNORM
- GL_RG8_SNORM
- GL_RGB8_SNORM
- GL_RGBA8_SNORM
- GL_R16_SNORM
- GL_RG16_SNORM
- GL_RGB16_SNORM
- GL_RGBA16_SNORM

- GL_SIGNED_NORMALIZED
- GL_PRIMITIVE_RESTART
- GL_PRIMITIVE_RESTART_INDEX
- GL_BUFFER_ACCESS_FLAGS
- GL_BUFFER_MAP_LENGTH
- GL_BUFFER_MAP_OFFSET
- GL_CONTEXT_CORE_PROFILE_BIT
- GL_CONTEXT_COMPATIBILITY_PROFILE_BIT
- GL_LINES_ADJACENCY
- GL_LINE_STRIP_ADJACENCY
- GL_TRIANGLES_ADJACENCY
- GL_TRIANGLE_STRIP_ADJACENCY
- GL_PROGRAM_POINT_SIZE
- GL_GEOMETRY_VERTICES_OUT
- GL_GEOMETRY_INPUT_TYPE
- GL_GEOMETRY_OUTPUT_TYPE
- GL_MAX_GEOMETRY_TEXTURE_IMAGE_UNITS
- GL_FRAMEBUFFER_ATTACHMENT_LAYERED
- GL_FRAMEBUFFER_INCOMPLETE_LAYER_TARGETS
- GL_GEOMETRY_SHADER
- GL_MAX_GEOMETRY_UNIFORM_COMPONENTS
- GL_MAX_GEOMETRY_OUTPUT_VERTICES
- GL_MAX_GEOMETRY_TOTAL_OUTPUT_COMPONENTS
- GL_MAX_VERTEX_OUTPUT_COMPONENTS
- GL_MAX_GEOMETRY_INPUT_COMPONENTS
- GL_MAX_GEOMETRY_OUTPUT_COMPONENTS
- GL_MAX_FRAGMENT_INPUT_COMPONENTS
- GL_CONTEXT_PROFILE_MASK
- GL_RGB10_A2UI
- void glAccum(GLenum op, GLfloat value)
- void glActiveTexture(GLenum texture)
- void glAlphaFunc(GLenum func, GLclampf ref)
- GLboolean glAreTexturesResident(GLsizei n, const GLuint * textures, GLboolean * residences)
- void glArrayElement(GLint i)
- void glAttachShader(GLuint program, GLuint shader)
- void glBegin(GLenum mode)

- void glBeginQuery(GLenum target,GLuint id)
- void glBindAttribLocation(GLuint program,GLuint index,const GLchar *name)
- void glBindBuffer(GLenum target,GLuint buffer)
- void glBindTexture(GLenum target,GLuint texture)
- void glBitmap(GLsizei width,GLsizei height,GLfloat xorig,GLfloat yorig,GLfloat xmove,GLfloat ymove,const GLubyte * bitmap)
- void glBlendColor(GLclampf red,GLclampf green,GLclampf blue,GLclampf alpha)
- void glBlendEquation(GLenum mode)
- void glBlendEquationSeparate(GLenum modeRGB,GLenum modeAlpha)
- void glBlendFunc(GLenum sfactor,GLenum dfactor)
- void glBlendFuncSeparate(GLenum srcRGB,GLenum dstRGB,GLenum srcAlpha,GLenum dstAlpha)
- void glBufferData(GLenum target,GLsizeiptr size,const GLvoid * data,GLenum usage)
- void glBufferSubData(GLenum target,GLintptr offset,GLsizeiptr size,const GLvoid * data)
- void glCallList(GLuint list)
- void glCallLists(GLsizei n,GLenum type,const GLvoid * lists)
- void glClear(GLbitfield mask)
- void glClearAccum(GLfloat red,GLfloat green,GLfloat blue,GLfloat alpha)
- void glClearColor(GLclampf red,GLclampf green,GLclampf blue,GLclampf alpha)
- void glClearDepth(GLclampd depth)
- void glClearIndex(GLfloat c)
- void glClearStencil(GLint s)
- void glClientActiveTexture(GLenum texture)
- void glClipPlane(GLenum plane,const GLdouble * equation)
- void glColor3b(GLbyte red,GLbyte green,GLbyte blue)
- void glColor3s(GLshort red,GLshort green,GLshort blue)
- void glColor3i(GLint red,GLint green,GLint blue)
- void glColor3f(GLfloat red,GLfloat green,GLfloat blue)
- void glColor3d(GLdouble red,GLdouble green,GLdouble blue)
- void glColor3ub(GLubyte red,GLubyte green,GLubyte blue)
- void glColor3us(GLushort red,GLushort green,GLushort blue)
- void glColor3ui(GLuint red,GLuint green,GLuint blue)
- void glColor4b(GLbyte red,GLbyte green,GLbyte blue,GLbyte alpha)
- void glColor4s(GLshort red,GLshort green,GLshort blue,GLshort alpha)
- void glColor4i(GLint red,GLint green,GLint blue,GLint alpha)
- void glColor4f(GLfloat red,GLfloat green,GLfloat blue,GLfloat alpha)
- void glColor4d(GLdouble red,GLdouble green,GLdouble blue,GLdouble alpha)

- void glColor4ub(GLubyte red, GLubyte green, GLubyte blue, GLubyte alpha)
- void glColor4us(GLushort red, GLushort green, GLushort blue, GLushort alpha)
- void glColor4ui(GLuint red, GLuint green, GLuint blue, GLuint alpha)
- void glColor3bv(const GLbyte * v)
- void glColor3sv(const GLshort * v)
- void glColor3iv(const GLint * v)
- void glColor3fv(const GLfloat * v)
- void glColor3dv(const GLdouble * v)
- void glColor3ubv(const GLubyte * v)
- void glColor3usv(const GLushort * v)
- void glColor3uiv(const GLuint * v)
- void glColor4bv(const GLbyte * v)
- void glColor4sv(const GLshort * v)
- void glColor4iv(const GLint * v)
- void glColor4fv(const GLfloat * v)
- void glColor4dv(const GLdouble * v)
- void glColor4ubv(const GLubyte * v)
- void glColor4usv(const GLushort * v)
- void glColor4uiv(const GLuint * v)
- void glColorMask(GLboolean red, GLboolean green, GLboolean blue, GLboolean alpha)
- void glColorMaterial(GLenum face, GLenum mode)
- void glColorPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)
- void glColorSubTable(GLenum target, GLsizei start, GLsizei count, GLenum format, GLenum type, const GLvoid * data)
- void glColorTable(GLenum target, GLenum internalformat, GLsizei width, GLenum format, GLenum type, const GLvoid * data)
- void glColorTableParameterfv(GLenum target, GLenum pname, const GLfloat * params)
- void glColorTableParameteriv(GLenum target, GLenum pname, const GLint * params)
- void glCompileShader(GLuint shader)
- void glCompressedTexImage1D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLint border, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexImage2D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLint border, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexImage3D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLsizei depth, GLint border, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLsizei imageSize, const GLvoid * data)

- `void glCompressedTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLsizei imageSize, const GLvoid * data)`
- `void glCompressedTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLsizei imageSize, const GLvoid * data)`
- `void glConvolutionFilter1D(GLenum target, GLenum internalformat, GLsizei width, GLenum format, GLenum type, const GLvoid * data)`
- `void glConvolutionFilter2D(GLenum target, GLenum internalformat, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)`
- `void glConvolutionParameterf(GLenum target, GLenum pname, GLfloat params)`
- `void glConvolutionParameteri(GLenum target, GLenum pname, GLint params)`
- `void glConvolutionParameterfv(GLenum target, GLenum pname, const GLfloat * params)`
- `void glConvolutionParameteriv(GLenum target, GLenum pname, const GLint * params)`
- `void glCopyColorSubTable(GLenum target, GLsizei start, GLint x, GLint y, GLsizei width)`
- `void glCopyColorTable(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width)`
- `void glCopyConvolutionFilter1D(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width)`
- `void glCopyConvolutionFilter2D(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width, GLsizei height)`
- `void glCopyPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum type)`
- `void glCopyTexImage1D(GLenum target, GLint level, GLenum internalformat, GLint x, GLint y, GLsizei width, GLint border)`
- `void glCopyTexImage2D(GLenum target, GLint level, GLenum internalformat, GLint x, GLint y, GLsizei width, GLsizei height, GLint border)`
- `void glCopyTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLint x, GLint y, GLsizei width)`
- `void glCopyTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint x, GLint y, GLsizei width, GLsizei height)`
- `void glCopyTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLint x, GLint y, GLsizei width, GLsizei height)`
- `GLuint glCreateProgram(void)`
- `GLuint glCreateShader(GLenum shaderType)`
- `void glCullFace(GLenum mode)`
- `void glDeleteBuffers(GLsizei n, const GLuint * buffers)`
- `void glDeleteLists(GLuint list, GLsizei range)`
- `void glDeleteProgram(GLuint program)`
- `void glDeleteQueries(GLsizei n, const GLuint * ids)`
- `void glDeleteShader(GLuint shader)`
- `void glDeleteTextures(GLsizei n, const GLuint * textures)`
- `void glDepthFunc(GLenum func)`
- `void glDepthMask(GLboolean flag)`
- `void glDepthRange(GLclampd nearVal, GLclampd farVal)`

- void glDetachShader(GLuint program,GLuint shader)
- void glEnable(GLenum cap)
- void glEnableClientState(GLenum cap)
- void glEnableVertexAttribArray(GLuint index)
- void glDisableVertexAttribArray(GLuint index)
- void glDrawArrays(GLenum mode,GLint first,GLsizei count)
- void glDrawBuffer(GLenum mode)
- void glDrawBuffers(GLsizei n,const GLenum *bufs)
- void glDrawElements(GLenum mode,GLsizei count,GLenum type,const GLvoid * indices)
- void glDrawPixels(GLsizei width,GLsizei height,GLenum format,GLenum type,const GLvoid * data)
- void glDrawRangeElements(GLenum mode,GLuint start,GLuint end,GLsizei count,GLenum type,const GLvoid * indices)
- void glEdgeFlag(GLboolean flag)
- void glEdgeFlagPointer(GLsizei stride,const GLvoid * pointer)
- void glEnd(void)
- void glEndList(void)
- void glEndQuery(GLenum target)
- void glEvalCoord1f(GLfloat u)
- void glEvalCoord1d(GLdouble u)
- void glEvalCoord2f(GLfloat u,GLfloat v)
- void glEvalCoord2d(GLdouble u,GLdouble v)
- void glEvalMesh1(GLenum mode,GLint i1,GLint i2)
- void glEvalPoint1(GLint i)
- void glEvalPoint2(GLint i,GLint j)
- void glFeedbackBuffer(GLsizei size,GLenum type,GLfloat * buffer)
- void glFinish(void)
- void glFlush(void)
- void glFogf(GLenum pname,GLfloat param)
- void glFogi(GLenum pname,GLint param)
- void glFogfv(GLenum pname,const GLfloat * params)
- void glFogiv(GLenum pname,const GLint * params)
- void glFogCoordd(GLdouble coord)
- void glFogCoordf(GLfloat coord)
- void glFogCoorddv(GLdouble * coord)
- void glFogCoordfv(GLfloat * coord)
- void glFogCoordPointer(GLenum type,GLsizei stride,GLvoid * pointer)

- void glFrontFace(GLenum mode)
- void glFrustum(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble nearVal, GLdouble farVal)
- void glGenBuffers(GLsizei n, GLuint * buffers)
- GLuint glGenLists(GLsizei range)
- void glGenQueries(GLsizei n, GLuint * ids)
- void glGenTextures(GLsizei n, GLuint * textures)
- void glGetBooleanv(GLenum pname, GLboolean * params)
- void glGetDoublev(GLenum pname, GLdouble * params)
- void glGetFloatv(GLenum pname, GLfloat * params)
- void glGetIntegerv(GLenum pname, GLint * params)
- void glGetActiveAttrib(GLuint program, GLuint index, GLsizei bufSize, GLsizei *length, GLint *size, GLenum *type, GLchar *name)
- void glGetActiveUniform(GLuint program, GLuint index, GLsizei bufSize, GLsizei *length, GLint *size, GLenum *type, GLchar *name)
- void glGetAttachedShaders(GLuint program, GLsizei maxCount, GLsizei *count, GLuint *shaders)
- GLint glGetAttribLocation(GLuint program, const GLchar *name)
- void glGetBufferParameteriv(GLenum target, GLenum value, GLint * data)
- void glGetBufferPointerv(GLenum target, GLenum pname, GLvoid ** params)
- void glGetBufferSubData(GLenum target, GLintptr offset, GLsizeiptr size, GLvoid * data)
- void glGetClipPlane(GLenum plane, GLdouble * equation)
- void glGetColorTable(GLenum target, GLenum format, GLenum type, GLvoid * table)
- void glGetColorTableParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetColorTableParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetCompressedTexImage(GLenum target, GLint lod, GLvoid * img)
- void glGetConvolutionFilter(GLenum target, GLenum format, GLenum type, GLvoid * image)
- void glGetConvolutionParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetConvolutionParameteriv(GLenum target, GLenum pname, GLint * params)
- GLenum glGetError(void)
- void glGetHistogram(GLenum target, GLboolean reset, GLenum format, GLenum type, GLvoid * values)
- void glGetHistogramParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetHistogramParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetLightfv(GLenum light, GLenum pname, GLfloat * params)
- void glGetLightiv(GLenum light, GLenum pname, GLint * params)
- void glGetMapdv(GLenum target, GLenum query, GLdouble * v)
- void glGetMapfv(GLenum target, GLenum query, GLfloat * v)
- void glGetMapiv(GLenum target, GLenum query, GLint * v)

- void glGetMaterialfv(GLenum face, GLenum pname, GLfloat * params)
- void glGetMaterialiv(GLenum face, GLenum pname, GLint * params)
- void glGetMinmax(GLenum target, GLboolean reset, GLenum format, GLenum types, GLvoid * values)
- void glGetMinmaxParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetMinmaxParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetPixelMapfv(GLenum map, GLfloat * data)
- void glGetPixelMapuiv(GLenum map, GLuint * data)
- void glGetPixelMapusv(GLenum map, GLushort * data)
- void glGetPointerv(GLenum pname, GLvoid ** params)
- void glGetPolygonStipple(GLubyte * pattern)
- void glGetProgramiv(GLuint program, GLenum pname, GLint *params)
- void glGetProgramInfoLog(GLuint program, GLsizei maxLength, GLsizei *length, GLchar *infoLog)
- void glGetQueryObjectiv(GLuint id, GLenum pname, GLint * params)
- void glGetQueryObjectuiv(GLuint id, GLenum pname, GLuint * params)
- void glGetQueryiv(GLenum target, GLenum pname, GLint * params)
- void glGetSeparableFilter(GLenum target, GLenum format, GLenum type, GLvoid * row, GLvoid * column, GLvoid * span)
- void glGetShaderiv(GLuint shader, GLenum pname, GLint *params)
- void glGetShaderInfoLog(GLuint shader, GLsizei maxLength, GLsizei *length, GLchar *infoLog)
- void glGetShaderSource(GLuint shader, GLsizei bufSize, GLsizei *length, GLchar *source)
- const GLubyte* glGetString(GLenum name)
- void glGetTexEnvfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetTexEnviv(GLenum target, GLenum pname, GLint * params)
- void glGetTexGendv(GLenum coord, GLenum pname, GLdouble * params)
- void glGetTexGenfv(GLenum coord, GLenum pname, GLfloat * params)
- void glGetTexGeniv(GLenum coord, GLenum pname, GLint * params)
- void glGetTexImage(GLenum target, GLint level, GLenum format, GLenum type, GLvoid * img)
- void glGetTexLevelParameterfv(GLenum target, GLint level, GLenum pname, GLfloat * params)
- void glGetTexLevelParameteriv(GLenum target, GLint level, GLenum pname, GLint * params)
- void glGetTexParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetTexParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetUniformfv(GLuint program, GLint location, GLfloat *params)
- void glGetUniformiv(GLuint program, GLint location, GLint *params)
- GLint glGetUniformLocation(GLuint program, const GLchar *name)
- void glGetVertexAttribdv(GLuint index, GLenum pname, GLdouble *params)
- void glGetVertexAttribfv(GLuint index, GLenum pname, GLfloat *params)

- void glGetVertexAttrib(GLuint index, GLenum pname, GLint *params)
- void glGetVertexAttribPointerv(GLuint index, GLenum pname, GLvoid **pointer)
- void glHint(GLenum target, GLenum mode)
- void glHistogram(GLenum target, GLsizei width, GLenum internalformat, GLboolean sink)
- void glIndexs(GLshort c)
- void glIndexi(GLint c)
- void glIndexf(GLfloat c)
- void glIndexd(GLdouble c)
- void glIndexub(GLubyte c)
- void glIndexsv(const GLshort * c)
- void glIndexiv(const GLint * c)
- void glIndexfv(const GLfloat * c)
- void glIndexdv(const GLdouble * c)
- void glIndexubv(const GLubyte * c)
- void glIndexMask(GLuint mask)
- void glIndexPointer(GLenum type, GLsizei stride, const GLvoid * pointer)
- void glInitNames(void)
- void glInterleavedArrays(GLenum format, GLsizei stride, const GLvoid * pointer)
- GLboolean glIsBuffer(GLuint buffer)
- GLboolean glIsEnabled(GLenum cap)
- GLboolean glIsList(GLuint list)
- GLboolean glIsProgram(GLuint program)
- GLboolean glIsQuery(GLuint id)
- GLboolean glIsShader(GLuint shader)
- GLboolean glIsTexture(GLuint texture)
- void glLightf(GLenum light, GLenum pname, GLfloat param)
- void glLighti(GLenum light, GLenum pname, GLint param)
- void glLightfv(GLenum light, GLenum pname, const GLfloat * params)
- void glLightiv(GLenum light, GLenum pname, const GLint * params)
- void glLightModelf(GLenum pname, GLfloat param)
- void glLightModeli(GLenum pname, GLint param)
- void glLightModelfv(GLenum pname, const GLfloat * params)
- void glLightModeliv(GLenum pname, const GLint * params)
- void glLineStipple(GLint factor, GLushort pattern)
- void glLineWidth(GLfloat width)
- void glLinkProgram(GLuint program)

- void glListBase(GLuint base)
- void glLoadIdentity(void)
- void glLoadMatrixd(const GLdouble * m)
- void glLoadMatrixf(const GLfloat * m)
- void glLoadName(GLuint name)
- void glLoadTransposeMatrixd(const GLdouble * m)
- void glLoadTransposeMatrixf(const GLfloat * m)
- void glLogicOp(GLenum opcode)
- void glMap1f(GLenum target, GLfloat u1, GLfloat u2, GLint stride, GLint order, const GLfloat * points)
- void glMap1d(GLenum target, GLdouble u1, GLdouble u2, GLint stride, GLint order, const GLdouble * points)
- void glMap2f(GLenum target, GLfloat u1, GLfloat u2, GLint ustride, GLint uorder, GLfloat v1, GLfloat v2, GLint vstride, GLint vorder, const GLfloat * points)
- void glMap2d(GLenum target, GLdouble u1, GLdouble u2, GLint ustride, GLint uorder, GLdouble v1, GLdouble v2, GLint vstride, GLint vorder, const GLdouble * points)
- void * glMapBuffer(GLenum target, GLenum access)
- void glMapGrid1d(GLint un, GLdouble u1, GLdouble u2)
- void glMapGrid1f(GLint un, GLfloat u1, GLfloat u2)
- void glMapGrid2d(GLint un, GLdouble u1, GLdouble u2, GLint vn, GLdouble v1, GLdouble v2)
- void glMapGrid2f(GLint un, GLfloat u1, GLfloat u2, GLint vn, GLfloat v1, GLfloat v2)
- void glMaterialf(GLenum face, GLenum pname, GLfloat param)
- void glMateriali(GLenum face, GLenum pname, GLint param)
- void glMatrixMode(GLenum mode)
- void glMinmax(GLenum target, GLenum internalformat, GLboolean sink)
- void glMultMatrixd(const GLdouble * m)
- void glMultMatrixf(const GLfloat * m)
- void glMultTransposeMatrixd(const GLdouble * m)
- void glMultTransposeMatrixf(const GLfloat * m)
- void glMultiDrawArrays(GLenum mode, GLint * first, GLsizei * count, GLsizei primcount)
- void glMultiDrawElements(GLenum mode, const GLsizei * count, GLenum type, const GLvoid ** indices, GLsizei primcount)
- void glMultiTexCoord1s(GLenum target, GLshort s)
- void glMultiTexCoord1i(GLenum target, GLint s)
- void glMultiTexCoord1f(GLenum target, GLfloat s)
- void glMultiTexCoord1d(GLenum target, GLdouble s)
- void glMultiTexCoord2s(GLenum target, GLshort s, GLshort t)
- void glMultiTexCoord2i(GLenum target, GLint s, GLint t)
- void glMultiTexCoord2f(GLenum target, GLfloat s, GLfloat t)

- `void glMultiTexCoord2d(GLenum target, GLdouble s, GLdouble t)`
- `void glMultiTexCoord3s(GLenum target, GLshort s, GLshort t, GLshort r)`
- `void glMultiTexCoord3i(GLenum target, GLint s, GLint t, GLint r)`
- `void glMultiTexCoord3f(GLenum target, GLfloat s, GLfloat t, GLfloat r)`
- `void glMultiTexCoord3d(GLenum target, GLdouble s, GLdouble t, GLdouble r)`
- `void glMultiTexCoord4s(GLenum target, GLshort s, GLshort t, GLshort r, GLshort q)`
- `void glMultiTexCoord4i(GLenum target, GLint s, GLint t, GLint r, GLint q)`
- `void glMultiTexCoord4f(GLenum target, GLfloat s, GLfloat t, GLfloat r, GLfloat q)`
- `void glMultiTexCoord4d(GLenum target, GLdouble s, GLdouble t, GLdouble r, GLdouble q)`
- `void glMultiTexCoord1sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord1iv(GLenum target, const GLint * v)`
- `void glMultiTexCoord1fv(GLenum target, const GLfloat * v)`
- `void glMultiTexCoord1dv(GLenum target, const GLdouble * v)`
- `void glMultiTexCoord2sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord2iv(GLenum target, const GLint * v)`
- `void glMultiTexCoord2fv(GLenum target, const GLfloat * v)`
- `void glMultiTexCoord2dv(GLenum target, const GLdouble * v)`
- `void glMultiTexCoord3sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord3iv(GLenum target, const GLint * v)`
- `void glMultiTexCoord3fv(GLenum target, const GLfloat * v)`
- `void glMultiTexCoord3dv(GLenum target, const GLdouble * v)`
- `void glMultiTexCoord4sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord4iv(GLenum target, const GLint * v)`
- `void glMultiTexCoord4fv(GLenum target, const GLfloat * v)`
- `void glMultiTexCoord4dv(GLenum target, const GLdouble * v)`
- `void glNewList(GLuint list, GLenum mode)`
- `void glNormal3b(GLbyte nx, GLbyte ny, GLbyte nz)`
- `void glNormal3d(GLdouble nx, GLdouble ny, GLdouble nz)`
- `void glNormal3f(GLfloat nx, GLfloat ny, GLfloat nz)`
- `void glNormal3i(GLint nx, GLint ny, GLint nz)`
- `void glNormal3s(GLshort nx, GLshort ny, GLshort nz)`
- `void glNormal3bv(const GLbyte * v)`
- `void glNormal3dv(const GLdouble * v)`
- `void glNormal3fv(const GLfloat * v)`
- `void glNormal3iv(const GLint * v)`
- `void glNormal3sv(const GLshort * v)`

- void glNormalPointer(GLenum type,GLsizei stride,const GLvoid * pointer)
- void glOrtho(GLdouble left,GLdouble right,GLdouble bottom,GLdouble top,GLdouble nearVal,GLdouble farVal)
- void glPassThrough(GLfloat token)
- void glPixelMapfv(GLenum map,GLsizei mapsize,const GLfloat * values)
- void glPixelMapuiv(GLenum map,GLsizei mapsize,const GLuint * values)
- void glPixelMapusv(GLenum map,GLsizei mapsize,const GLushort * values)
- void glPixelStoref(GLenum pname,GLfloat param)
- void glPixelStorei(GLenum pname,GLint param)
- void glPixelTransferf(GLenum pname,GLfloat param)
- void glPixelTransferi(GLenum pname,GLint param)
- void glPixelZoom(GLfloat xfactor,GLfloat yfactor)
- void glPointParameterf(GLenum pname,GLfloat param)
- void glPointParameteri(GLenum pname,GLint param)
- void glPointSize(GLfloat size)
- void glPolygonMode(GLenum face,GLenum mode)
- void glPolygonOffset(GLfloat factor,GLfloat units)
- void glPolygonStipple(const GLubyte * pattern)
- void glPushAttrib(GLbitfield mask)
- void glPushClientAttrib(GLbitfield mask)
- void glPushMatrix(void)
- void glPushName(GLuint name)
- void glPrioritizeTextures(GLsizei n,const GLuint * textures,const GLclampf * priorities)
- void glPopMatrix(void)
- void glRasterPos2s(GLshort x,GLshort y)
- void glRasterPos2i(GLint x,GLint y)
- void glRasterPos2f(GLfloat x,GLfloat y)
- void glRasterPos2d(GLdouble x,GLdouble y)
- void glRasterPos3s(GLshort x,GLshort y,GLshort z)
- void glRasterPos3i(GLint x,GLint y,GLint z)
- void glRasterPos3f(GLfloat x,GLfloat y,GLfloat z)
- void glRasterPos3d(GLdouble x,GLdouble y,GLdouble z)
- void glRasterPos4s(GLshort x,GLshort y,GLshort z,GLshort w)
- void glRasterPos4i(GLint x,GLint y,GLint z,GLint w)
- void glRasterPos4f(GLfloat x,GLfloat y,GLfloat z,GLfloat w)
- void glRasterPos4d(GLdouble x,GLdouble y,GLdouble z,GLdouble w)

- `void glReadBuffer(GLenum mode)`
- `void glReadPixels(GLint x,GLint y,GLsizei width,GLsizei height,GLenum format,GLenum type,GLvoid * data)`
- `void glRectd(GLdouble x1,GLdouble y1,GLdouble x2,GLdouble y2)`
- `void glRectf(GLfloat x1,GLfloat y1,GLfloat x2,GLfloat y2)`
- `void glRecti(GLint x1,GLint y1,GLint x2,GLint y2)`
- `void glRects(GLshort x1,GLshort y1,GLshort x2,GLshort y2)`
- `void glRectdv(const GLdouble * v1,const GLdouble * v2)`
- `void glRectfv(const GLfloat * v1,const GLfloat * v2)`
- `void glRectiv(const GLint * v1,const GLint * v2)`
- `void glRectsv(const GLshort * v1,const GLshort * v2)`
- `GLint glRenderMode(GLenum mode)`
- `void glResetHistogram(GLenum target)`
- `void glRotated(GLdouble angle,GLdouble x,GLdouble y,GLdouble z)`
- `void glRotatef(GLfloat angle,GLfloat x,GLfloat y,GLfloat z)`
- `void glSampleCoverage(GLclampf value,GLboolean invert)`
- `void glScaled(GLdouble x,GLdouble y,GLdouble z)`
- `void glScalef(GLfloat x,GLfloat y,GLfloat z)`
- `void glScissor(GLint x,GLint y,GLsizei width,GLsizei height)`
- `void glSecondaryColor3b(GLbyte red,GLbyte green,GLbyte blue)`
- `void glSecondaryColor3s(GLshort red,GLshort green,GLshort blue)`
- `void glSecondaryColor3i(GLint red,GLint green,GLint blue)`
- `void glSecondaryColor3f(GLfloat red,GLfloat green,GLfloat blue)`
- `void glSecondaryColor3d(GLdouble red,GLdouble green,GLdouble blue)`
- `void glSecondaryColor3ub(GLubyte red,GLubyte green,GLubyte blue)`
- `void glSecondaryColor3us(GLushort red,GLushort green,GLushort blue)`
- `void glSecondaryColor3ui(GLuint red,GLuint green,GLuint blue)`
- `void glSecondaryColor3bv(const GLbyte * v)`
- `void glSecondaryColor3sv(const GLshort * v)`
- `void glSecondaryColor3iv(const GLint * v)`
- `void glSecondaryColor3fv(const GLfloat * v)`
- `void glSecondaryColor3dv(const GLdouble * v)`
- `void glSecondaryColor3ubv(const GLubyte * v)`
- `void glSecondaryColor3usv(const GLushort * v)`
- `void glSecondaryColor3uiv(const GLuint * v)`
- `void glSecondaryColorPointer(GLint size,GLenum type,GLsizei stride,const GLvoid * pointer)`
- `void glSelectBuffer(GLsizei size,GLuint * buffer)`

- void glSeparableFilter2D(GLenum target, GLenum internalformat, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * row, const GLvoid * column)
- void glShadeModel(GLenum mode)
- void glShaderSource(GLuint shader, GLsizei count, const GLchar **string, const GLint *length)
- void glStencilFunc(GLenum func, GLint ref, GLuint mask)
- void glStencilFuncSeparate(GLenum face, GLenum func, GLint ref, GLuint mask)
- void glStencilMask(GLuint mask)
- void glStencilMaskSeparate(GLenum face, GLuint mask)
- void glStencilOp(GLenum sfail, GLenum dpfail, GLenum dppass)
- void glStencilOpSeparate(GLenum face, GLenum sfail, GLenum dpfail, GLenum dppass)
- void glTexCoord1s(GLshort s)
- void glTexCoord1i(GLint s)
- void glTexCoord1f(GLfloat s)
- void glTexCoord1d(GLdouble s)
- void glTexCoord2s(GLshort s, GLshort t)
- void glTexCoord2i(GLint s, GLint t)
- void glTexCoord2f(GLfloat s, GLfloat t)
- void glTexCoord2d(GLdouble s, GLdouble t)
- void glTexCoord3s(GLshort s, GLshort t, GLshort r)
- void glTexCoord3i(GLint s, GLint t, GLint r)
- void glTexCoord3f(GLfloat s, GLfloat t, GLfloat r)
- void glTexCoord3d(GLdouble s, GLdouble t, GLdouble r)
- void glTexCoord4s(GLshort s, GLshort t, GLshort r, GLshort q)
- void glTexCoord4i(GLint s, GLint t, GLint r, GLint q)
- void glTexCoord4f(GLfloat s, GLfloat t, GLfloat r, GLfloat q)
- void glTexCoord4d(GLdouble s, GLdouble t, GLdouble r, GLdouble q)
- void glTexCoord1sv(const GLshort * v)
- void glTexCoord1iv(const GLint * v)
- void glTexCoord1fv(const GLfloat * v)
- void glTexCoord1dv(const GLdouble * v)
- void glTexCoord2sv(const GLshort * v)
- void glTexCoord2iv(const GLint * v)
- void glTexCoord2fv(const GLfloat * v)
- void glTexCoord2dv(const GLdouble * v)
- void glTexCoord3sv(const GLshort * v)
- void glTexCoord3iv(const GLint * v)

- `void glTexCoord3fv(const GLfloat * v)`
- `void glTexCoord3dv(const GLdouble * v)`
- `void glTexCoord4sv(const GLshort * v)`
- `void glTexCoord4iv(const GLint * v)`
- `void glTexCoord4fv(const GLfloat * v)`
- `void glTexCoord4dv(const GLdouble * v)`
- `void glTexCoordPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)`
- `void glTexEnvf(GLenum target, GLenum pname, GLfloat param)`
- `void glTexEnvf(GLenum target, GLenum pname, GLint param)`
- `void glTexGeni(GLenum coord, GLenum pname, GLint param)`
- `void glTexGenf(GLenum coord, GLenum pname, GLfloat param)`
- `void glTexGend(GLenum coord, GLenum pname, GLdouble param)`
- `void glTexGeniv(GLenum coord, GLenum pname, const GLint * params)`
- `void glTexGenfv(GLenum coord, GLenum pname, const GLfloat * params)`
- `void glTexGendv(GLenum coord, GLenum pname, const GLdouble * params)`
- `void glTexImage1D(GLenum target, GLint level, GLint internalFormat, GLsizei width, GLint border, GLenum format, GLenum type, const GLvoid * data)`
- `void glTexImage2D(GLenum target, GLint level, GLint internalFormat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, const GLvoid * data)`
- `void glTexImage3D(GLenum target, GLint level, GLint internalFormat, GLsizei width, GLsizei height, GLsizei depth, GLint border, GLenum format, GLenum type, const GLvoid * data)`
- `void glTexParameterf(GLenum target, GLenum pname, GLfloat param)`
- `void glTexParameteri(GLenum target, GLenum pname, GLint param)`
- `void glTexParameterfv(GLenum target, GLenum pname, const GLfloat * params)`
- `void glTexParameteriv(GLenum target, GLenum pname, const GLint * params)`
- `void glTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLenum type, const GLvoid * data)`
- `void glTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)`
- `void glTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLenum type, const GLvoid * data)`
- `void glTranslated(GLdouble x, GLdouble y, GLdouble z)`
- `void glTranslatef(GLfloat x, GLfloat y, GLfloat z)`
- `void glUniform1f(GLint location, GLfloat v0)`
- `void glUniform2f(GLint location, GLfloat v0, GLfloat v1)`
- `void glUniform3f(GLint location, GLfloat v0, GLfloat v1, GLfloat v2)`
- `void glUniform4f(GLint location, GLfloat v0, GLfloat v1, GLfloat v2, GLfloat v3)`
- `void glUniform1i(GLint location, GLint v0)`

- void glUniform2i(GLint location,GLint v0,GLint v1)
- void glUniform3i(GLint location,GLint v0,GLint v1,GLint v2)
- void glUniform4i(GLint location,GLint v0,GLint v1,GLint v2,GLint v3)
- void glUniform1fv(GLint location,GLsizei count,const GLfloat *value)
- void glUniform2fv(GLint location,GLsizei count,const GLfloat *value)
- void glUniform3fv(GLint location,GLsizei count,const GLfloat *value)
- void glUniform4fv(GLint location,GLsizei count,const GLfloat *value)
- void glUniform1iv(GLint location,GLsizei count,const GLint *value)
- void glUniform2iv(GLint location,GLsizei count,const GLint *value)
- void glUniform3iv(GLint location,GLsizei count,const GLint *value)
- void glUniform4iv(GLint location,GLsizei count,const GLint *value)
- void glUniformMatrix2fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix3fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix4fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix2x3fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix3x2fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix2x4fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix4x2fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix3x4fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix4x3fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUseProgram(GLuint program)
- void glValidateProgram(GLuint program)
- void glVertex2s(GLshort x,GLshort y)
- void glVertex2i(GLint x,GLint y)
- void glVertex2f(GLfloat x,GLfloat y)
- void glVertex2d(GLdouble x,GLdouble y)
- void glVertex3s(GLshort x,GLshort y,GLshort z)
- void glVertex3i(GLint x,GLint y,GLint z)
- void glVertex3f(GLfloat x,GLfloat y,GLfloat z)
- void glVertex3d(GLdouble x,GLdouble y,GLdouble z)
- void glVertex4s(GLshort x,GLshort y,GLshort z,GLshort w)
- void glVertex4i(GLint x,GLint y,GLint z,GLint w)
- void glVertex4f(GLfloat x,GLfloat y,GLfloat z,GLfloat w)
- void glVertex4d(GLdouble x,GLdouble y,GLdouble z,GLdouble w)
- void glVertex2sv(const GLshort * v)
- void glVertex2iv(const GLint * v)

- `void glVertex2fv(const GLfloat * v)`
- `void glVertex2dv(const GLdouble * v)`
- `void glVertex3sv(const GLshort * v)`
- `void glVertex3iv(const GLint * v)`
- `void glVertex3fv(const GLfloat * v)`
- `void glVertex3dv(const GLdouble * v)`
- `void glVertex4sv(const GLshort * v)`
- `void glVertex4iv(const GLint * v)`
- `void glVertex4fv(const GLfloat * v)`
- `void glVertex4dv(const GLdouble * v)`
- `void glVertexAttrib1f(GLuint index, GLfloat v0)`
- `void glVertexAttrib1s(GLuint index, GLshort v0)`
- `void glVertexAttrib1d(GLuint index, GLdouble v0)`
- `void glVertexAttrib2f(GLuint index, GLfloat v0, GLfloat v1)`
- `void glVertexAttrib2s(GLuint index, GLshort v0, GLshort v1)`
- `void glVertexAttrib2d(GLuint index, GLdouble v0, GLdouble v1)`
- `void glVertexAttrib3f(GLuint index, GLfloat v0, GLfloat v1, GLfloat v2)`
- `void glVertexAttrib3s(GLuint index, GLshort v0, GLshort v1, GLshort v2)`
- `void glVertexAttrib3d(GLuint index, GLdouble v0, GLdouble v1, GLdouble v2)`
- `void glVertexAttrib4f(GLuint index, GLfloat v0, GLfloat v1, GLfloat v2, GLfloat v3)`
- `void glVertexAttrib4s(GLuint index, GLshort v0, GLshort v1, GLshort v2, GLshort v3)`
- `void glVertexAttrib4d(GLuint index, GLdouble v0, GLdouble v1, GLdouble v2, GLdouble v3)`
- `void glVertexAttrib4Nub(GLuint index, GLubyte v0, GLubyte v1, GLubyte v2, GLubyte v3)`
- `void glVertexAttrib1fv(GLuint index, const GLfloat *v)`
- `void glVertexAttrib1sv(GLuint index, const GLshort *v)`
- `void glVertexAttrib1dv(GLuint index, const GLdouble *v)`
- `void glVertexAttrib2fv(GLuint index, const GLfloat *v)`
- `void glVertexAttrib2sv(GLuint index, const GLshort *v)`
- `void glVertexAttrib2dv(GLuint index, const GLdouble *v)`
- `void glVertexAttrib3fv(GLuint index, const GLfloat *v)`
- `void glVertexAttrib3sv(GLuint index, const GLshort *v)`
- `void glVertexAttrib3dv(GLuint index, const GLdouble *v)`
- `void glVertexAttrib4fv(GLuint index, const GLfloat *v)`
- `void glVertexAttrib4sv(GLuint index, const GLshort *v)`
- `void glVertexAttrib4dv(GLuint index, const GLdouble *v)`
- `void glVertexAttrib4iv(GLuint index, const GLint *v)`

- void glVertexAttrib4bv(GLuint index,const GLbyte *v)
- void glVertexAttrib4ubv(GLuint index,const GLubyte *v)
- void glVertexAttrib4usv(GLuint index,const GLushort *v)
- void glVertexAttrib4uiv(GLuint index,const GLuint *v)
- void glVertexAttribPointer(GLuint index,GLint size,GLenum type,GLboolean normalized,GLsizei stride,const GLvoid * pointer)
- void glVertexPointer(GLint size,GLenum type,GLsizei stride,const GLvoid * pointer)
- void glViewport(GLint x,GLint y,GLsizei width,GLsizei height)
- void glWindowPos2s(GLshort x,GLshort y)
- void glWindowPos2i(GLint x,GLint y)
- void glWindowPos2f(GLfloat x,GLfloat y)
- void glWindowPos2d(GLdouble x,GLdouble y)
- void glWindowPos3s(GLshort x,GLshort y,GLshort z)
- void glWindowPos3i(GLint x,GLint y,GLint z)
- void glWindowPos3f(GLfloat x,GLfloat y,GLfloat z)
- void glWindowPos3d(GLdouble x,GLdouble y,GLdouble z)
- void glWindowPos2sv(const GLshort * v)
- void glWindowPos2iv(const GLint * v)
- void glWindowPos2fv(const GLfloat * v)
- void glWindowPos2dv(const GLdouble * v)
- void glWindowPos3sv(const GLshort * v)
- void glWindowPos3iv(const GLint * v)
- void glWindowPos3fv(const GLfloat * v)
- void glWindowPos3dv(const GLdouble * v)
- void gluBeginCurve(GLUnurbs* nurb)
- void gluBeginPolygon(GLUtesselator* tess)
- void gluBeginSurface(GLUnurbs* nurb)
- void gluBeginTrim(GLUnurbs* nurb)
- void gluCylinder(GLUquadric* quad,GLdouble base,GLdouble top,GLdouble height,GLint slices,GLint stacks)
- void gluDeleteNurbsRenderer(GLUnurbs* nurb)
- void gluDeleteQuadric(GLUquadric* quad)
- void gluDeleteTess(GLUtesselator* tess)
- void gluDisk(GLUquadric* quad,GLdouble inner,GLdouble outer,GLint slices,GLint loops)
- void gluEndCurve(GLUnurbs* nurb)
- void gluEndPolygon(GLUtesselator* tess)
- void gluEndSurface(GLUnurbs* nurb)

- void gluEndTrim(GLUnurbs* nurb)
- const GLubyte * gluErrorString(GLenum error)
- void gluGetNurbsProperty(GLUnurbs* nurb, GLenum property, GLfloat* data)
- const GLubyte * gluGetString(GLenum name)
- void gluGetTessProperty(GLUtesselator* tess, GLenum which, GLdouble* data)
- void gluLoadSamplingMatrices(GLUnurbs* nurb, const GLfloat * model, const GLfloat * perspective, const GLint * view)
- void gluLookAt(GLdouble eyeX, GLdouble eyeY, GLdouble eyeZ, GLdouble centerX, GLdouble centerY, GLdouble centerZ, GLdouble upX, GLdouble upY, GLdouble upZ)
- GLUnurbs *gluNewNurbsRenderer(void)
- GLUquadric *gluNewQuadric(void)
- GLUtesselator* gluNewTess(void)
- void gluNextContour(GLUtesselator* tess, GLenum type)
- void gluNurbsCurve(GLUnurbs* nurb, GLint knotCount, GLfloat * knots, GLint stride, GLfloat * control, GLint order, GLenum type)
- void gluNurbsProperty(GLUnurbs* nurb, GLenum property, GLfloat value)
- void gluNurbsSurface(GLUnurbs* nurb, GLint sKnotCount, GLfloat* sKnots, GLint tKnotCount, GLfloat* tKnots, GLint sStride, GLint tStride, GLfloat* control, GLint sOrder, GLint tOrder, GLenum type)
- void gluOrtho2D(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top)
- void gluPartialDisk(GLUquadric* quad, GLdouble inner, GLdouble outer, GLint slices, GLint loops, GLdouble start, GLdouble sweep)
- void gluPerspective(GLdouble fovy, GLdouble aspect, GLdouble zNear, GLdouble zFar)
- void gluPickMatrix(GLdouble x, GLdouble y, GLdouble delX, GLdouble delY, GLint * viewport)
- GLint gluProject(GLdouble objX, GLdouble objY, GLdouble objZ, const GLdouble * model, const GLdouble * proj, const GLint * view, GLdouble* winX, GLdouble* winY, GLdouble* winZ)
- void gluPwlCurve(GLUnurbs* nurb, GLint count, GLfloat* data, GLint stride, GLenum type)
- void gluQuadricDrawStyle(GLUquadric* quad, GLenum draw)
- void gluQuadricNormals(GLUquadric* quad, GLenum normal)
- void gluQuadricOrientation(GLUquadric* quad, GLenum orientation)
- void gluQuadricTexture(GLUquadric* quad, GLboolean texture)
- GLint gluScaleImage(GLenum format, GLsizei wIn, GLsizei hIn, GLenum typeIn, const void * dataIn, GLsizei wOut, GLsizei hOut, GLenum typeOut, GLvoid* dataOut)
- void gluSphere(GLUquadric* quad, GLdouble radius, GLint slices, GLint stacks)
- void gluTessBeginContour(GLUtesselator* tess)
- void gluTessBeginPolygon(GLUtesselator* tess, GLvoid* data)
- void gluTessEndContour(GLUtesselator* tess)
- void gluTessEndPolygon(GLUtesselator* tess)
- void gluTessNormal(GLUtesselator* tess, GLdouble valueX, GLdouble valueY, GLdouble valueZ)

- void gluTessProperty(GLUtesselator* tess, GLenum which, GLdouble data)
- void gluTessVertex(GLUtesselator* tess, GLdouble * location, GLvoid* data)
- GLint gluUnProject(GLdouble winX, GLdouble winY, GLdouble winZ, const GLdouble * model, const GLdouble * proj, const GLint * view, GLdouble* objX, GLdouble* objY, GLdouble* objZ)
- void glDisable(GLenum cap)

RINGOPENGL (OPENGL 4.0) FUNCTIONS REFERENCE

- GL_ZERO
- GL_FALSE
- GL_LOGIC_OP
- GL_NONE
- GL_TEXTURE_COMPONENTS
- GL_NO_ERROR
- GL_POINTS
- GL_CURRENT_BIT
- GL_TRUE
- GL_ONE
- GL_CLIENT_PIXEL_STORE_BIT
- GL_LINES
- GL_LINE_LOOP
- GL_POINT_BIT
- GL_CLIENT_VERTEX_ARRAY_BIT
- GL_LINE_STRIP
- GL_LINE_BIT
- GL_TRIANGLES
- GL_TRIANGLE_STRIP
- GL_TRIANGLE_FAN
- GL_QUADS
- GL_QUAD_STRIP
- GL_POLYGON_BIT
- GL_POLYGON
- GL_POLYGON_STIPPLE_BIT
- GL_PIXEL_MODE_BIT
- GL_LIGHTING_BIT

- GL_FOG_BIT
- GL_DEPTH_BUFFER_BIT
- GL_ACCUM
- GL_LOAD
- GL_RETURN
- GL_MULT
- GL_ADD
- GL_NEVER
- GL_ACCUM_BUFFER_BIT
- GL_LESS
- GL_EQUAL
- GL_LEQUAL
- GL_GREATER
- GL_NOTEQUAL
- GL_GEQUAL
- GL_ALWAYS
- GL_SRC_COLOR
- GL_ONE_MINUS_SRC_COLOR
- GL_SRC_ALPHA
- GL_ONE_MINUS_SRC_ALPHA
- GL_DST_ALPHA
- GL_ONE_MINUS_DST_ALPHA
- GL_DST_COLOR
- GL_ONE_MINUS_DST_COLOR
- GL_SRC_ALPHA_SATURATE
- GL_STENCIL_BUFFER_BIT
- GL_FRONT_LEFT
- GL_FRONT_RIGHT
- GL_BACK_LEFT
- GL_BACK_RIGHT
- GL_FRONT
- GL_BACK
- GL_LEFT
- GL_RIGHT
- GL_FRONT_AND_BACK
- GL_AUX0

- GL_AUX1
- GL_AUX2
- GL_AUX3
- GL_INVALID_ENUM
- GL_INVALID_VALUE
- GL_INVALID_OPERATION
- GL_STACK_OVERFLOW
- GL_STACK_UNDERFLOW
- GL_OUT_OF_MEMORY
- GL_2D
- GL_3D
- GL_3D_COLOR
- GL_3D_COLOR_TEXTURE
- GL_4D_COLOR_TEXTURE
- GL_PASS_THROUGH_TOKEN
- GL_POINT_TOKEN
- GL_LINE_TOKEN
- GL_POLYGON_TOKEN
- GL_BITMAP_TOKEN
- GL_DRAW_PIXEL_TOKEN
- GL_COPY_PIXEL_TOKEN
- GL_LINE_RESET_TOKEN
- GL_EXP
- GL_VIEWPORT_BIT
- GL_EXP2
- GL_CW
- GL_CCW
- GL_COEFF
- GL_ORDER
- GL_DOMAIN
- GL_CURRENT_COLOR
- GL_CURRENT_INDEX
- GL_CURRENT_NORMAL
- GL_CURRENT_TEXTURE_COORDS
- GL_CURRENT_RASTER_COLOR
- GL_CURRENT_RASTER_INDEX

- GL_CURRENT_RASTER_TEXTURE_COORDS
- GL_CURRENT_RASTER_POSITION
- GL_CURRENT_RASTER_POSITION_VALID
- GL_CURRENT_RASTER_DISTANCE
- GL_POINT_SMOOTH
- GL_POINT_SIZE
- GL_POINT_SIZE_RANGE
- GL_POINT_SIZE_GRANULARITY
- GL_LINE_SMOOTH
- GL_LINE_WIDTH
- GL_LINE_WIDTH_RANGE
- GL_LINE_WIDTH_GRANULARITY
- GL_LINE_STIPPLE
- GL_LINE_STIPPLE_PATTERN
- GL_LINE_STIPPLE_REPEAT
- GL_LIST_MODE
- GL_MAX_LIST_NESTING
- GL_LIST_BASE
- GL_LIST_INDEX
- GL_POLYGON_MODE
- GL_POLYGON_SMOOTH
- GL_POLYGON_STIPPLE
- GL_EDGE_FLAG
- GL_CULL_FACE
- GL_CULL_FACE_MODE
- GL_FRONT_FACE
- GL_LIGHTING
- GL_LIGHT_MODEL_LOCAL_VIEWER
- GL_LIGHT_MODEL_TWO_SIDE
- GL_LIGHT_MODEL_AMBIENT
- GL_SHADE_MODEL
- GL_COLOR_MATERIAL_FACE
- GL_COLOR_MATERIAL_PARAMETER
- GL_COLOR_MATERIAL
- GL_FOG
- GL_FOG_INDEX

- GL_FOG_DENSITY
- GL_FOG_START
- GL_FOG_END
- GL_FOG_MODE
- GL_FOG_COLOR
- GL_DEPTH_RANGE
- GL_DEPTH_TEST
- GL_DEPTH_WRITEMASK
- GL_DEPTH_CLEAR_VALUE
- GL_DEPTH_FUNC
- GL_ACCUM_CLEAR_VALUE
- GL_STENCIL_TEST
- GL_STENCIL_CLEAR_VALUE
- GL_STENCIL_FUNC
- GL_STENCIL_VALUE_MASK
- GL_STENCIL_FAIL
- GL_STENCIL_PASS_DEPTH_FAIL
- GL_STENCIL_PASS_DEPTH_PASS
- GL_STENCIL_REF
- GL_STENCIL_WRITEMASK
- GL_MATRIX_MODE
- GL_NORMALIZE
- GL_VIEWPORT
- GL_MODELVIEW_STACK_DEPTH
- GL_PROJECTION_STACK_DEPTH
- GL_TEXTURE_STACK_DEPTH
- GL_MODELVIEW_MATRIX
- GL_PROJECTION_MATRIX
- GL_TEXTURE_MATRIX
- GL_ATTRIB_STACK_DEPTH
- GL_CLIENT_ATTRIB_STACK_DEPTH
- GL_ALPHA_TEST
- GL_ALPHA_TEST_FUNC
- GL_ALPHA_TEST_REF
- GL_DITHER
- GL_BLEND_DST

- GL_BLEND_SRC
- GL_BLEND
- GL_LOGIC_OP_MODE
- GL_INDEX_LOGIC_OP
- GL_COLOR_LOGIC_OP
- GL_AUX_BUFFERS
- GL_DRAW_BUFFER
- GL_READ_BUFFER
- GL_SCISSOR_BOX
- GL_SCISSOR_TEST
- GL_INDEX_CLEAR_VALUE
- GL_INDEX_WRITEMASK
- GL_COLOR_CLEAR_VALUE
- GL_COLOR_WRITEMASK
- GL_INDEX_MODE
- GL_RGBA_MODE
- GL_DOUBLEBUFFER
- GL_STEREO
- GL_RENDER_MODE
- GL_PERSPECTIVE_CORRECTION_HINT
- GL_POINT_SMOOTH_HINT
- GL_LINE_SMOOTH_HINT
- GL_POLYGON_SMOOTH_HINT
- GL_FOG_HINT
- GL_TEXTURE_GEN_S
- GL_TEXTURE_GEN_T
- GL_TEXTURE_GEN_R
- GL_TEXTURE_GEN_Q
- GL_PIXEL_MAP_I_TO_I
- GL_PIXEL_MAP_S_TO_S
- GL_PIXEL_MAP_I_TO_R
- GL_PIXEL_MAP_I_TO_G
- GL_PIXEL_MAP_I_TO_B
- GL_PIXEL_MAP_I_TO_A
- GL_PIXEL_MAP_R_TO_R
- GL_PIXEL_MAP_G_TO_G

- GL_PIXEL_MAP_B_TO_B
- GL_PIXEL_MAP_A_TO_A
- GL_PIXEL_MAP_I_TO_I_SIZE
- GL_PIXEL_MAP_S_TO_S_SIZE
- GL_PIXEL_MAP_I_TO_R_SIZE
- GL_PIXEL_MAP_I_TO_G_SIZE
- GL_PIXEL_MAP_I_TO_B_SIZE
- GL_PIXEL_MAP_I_TO_A_SIZE
- GL_PIXEL_MAP_R_TO_R_SIZE
- GL_PIXEL_MAP_G_TO_G_SIZE
- GL_PIXEL_MAP_B_TO_B_SIZE
- GL_PIXEL_MAP_A_TO_A_SIZE
- GL_UNPACK_SWAP_BYTES
- GL_UNPACK_LSB_FIRST
- GL_UNPACK_ROW_LENGTH
- GL_UNPACK_SKIP_ROWS
- GL_UNPACK_SKIP_PIXELS
- GL_UNPACK_ALIGNMENT
- GL_PACK_SWAP_BYTES
- GL_PACK_LSB_FIRST
- GL_PACK_ROW_LENGTH
- GL_PACK_SKIP_ROWS
- GL_PACK_SKIP_PIXELS
- GL_PACK_ALIGNMENT
- GL_MAP_COLOR
- GL_MAP_STENCIL
- GL_INDEX_SHIFT
- GL_INDEX_OFFSET
- GL_RED_SCALE
- GL_RED_BIAS
- GL_ZOOM_X
- GL_ZOOM_Y
- GL_GREEN_SCALE
- GL_GREEN_BIAS
- GL_BLUE_SCALE
- GL_BLUE_BIAS

- GL_ALPHA_SCALE
- GL_ALPHA_BIAS
- GL_DEPTH_SCALE
- GL_DEPTH_BIAS
- GL_MAX_EVAL_ORDER
- GL_MAX_LIGHTS
- GL_MAX_CLIP_PLANES
- GL_MAX_TEXTURE_SIZE
- GL_MAX_PIXEL_MAP_TABLE
- GL_MAX_ATTRIB_STACK_DEPTH
- GL_MAX_MODELVIEW_STACK_DEPTH
- GL_MAX_NAME_STACK_DEPTH
- GL_MAX_PROJECTION_STACK_DEPTH
- GL_MAX_TEXTURE_STACK_DEPTH
- GL_MAX_VIEWPORT_DIMS
- GL_MAX_CLIENT_ATTRIB_STACK_DEPTH
- GL_SUBPIXEL_BITS
- GL_INDEX_BITS
- GL_RED_BITS
- GL_GREEN_BITS
- GL_BLUE_BITS
- GL_ALPHA_BITS
- GL_DEPTH_BITS
- GL_STENCIL_BITS
- GL_ACCUM_RED_BITS
- GL_ACCUM_GREEN_BITS
- GL_ACCUM_BLUE_BITS
- GL_ACCUM_ALPHA_BITS
- GL_NAME_STACK_DEPTH
- GL_AUTO_NORMAL
- GL_MAP1_COLOR_4
- GL_MAP1_INDEX
- GL_MAP1_NORMAL
- GL_MAP1_TEXTURE_COORD_1
- GL_MAP1_TEXTURE_COORD_2
- GL_MAP1_TEXTURE_COORD_3

- GL_MAP1_TEXTURE_COORD_4
- GL_MAP1_VERTEX_3
- GL_MAP1_VERTEX_4
- GL_MAP2_COLOR_4
- GL_MAP2_INDEX
- GL_MAP2_NORMAL
- GL_MAP2_TEXTURE_COORD_1
- GL_MAP2_TEXTURE_COORD_2
- GL_MAP2_TEXTURE_COORD_3
- GL_MAP2_TEXTURE_COORD_4
- GL_MAP2_VERTEX_3
- GL_MAP2_VERTEX_4
- GL_MAP1_GRID_DOMAIN
- GL_MAP1_GRID_SEGMENTS
- GL_MAP2_GRID_DOMAIN
- GL_MAP2_GRID_SEGMENTS
- GL_TEXTURE_1D
- GL_TEXTURE_2D
- GL_FEEDBACK_BUFFER_POINTER
- GL_FEEDBACK_BUFFER_SIZE
- GL_FEEDBACK_BUFFER_TYPE
- GL_SELECTION_BUFFER_POINTER
- GL_SELECTION_BUFFER_SIZE
- GL_TEXTURE_WIDTH
- GL_TRANSFORM_BIT
- GL_TEXTURE_HEIGHT
- GL_TEXTURE_INTERNAL_FORMAT
- GL_TEXTURE_BORDER_COLOR
- GL_TEXTURE_BORDER
- GL_DONT_CARE
- GL_FASTEST
- GL_NICEST
- GL_AMBIENT
- GL_DIFFUSE
- GL_SPECULAR
- GL_POSITION

- GL_SPOT_DIRECTION
- GL_SPOT_EXPONENT
- GL_SPOT_CUTOFF
- GL_CONSTANT_ATTENUATION
- GL_LINEAR_ATTENUATION
- GL_QUADRATIC_ATTENUATION
- GL_COMPILE
- GL_COMPILE_AND_EXECUTE
- GL_BYTE
- GL_UNSIGNED_BYTE
- GL_SHORT
- GL_UNSIGNED_SHORT
- GL_INT
- GL_UNSIGNED_INT
- GL_FLOAT
- GL_2_BYTES
- GL_3_BYTES
- GL_4_BYTES
- GL_DOUBLE
- GL_CLEAR
- GL_AND
- GL_AND_REVERSE
- GL_COPY
- GL_AND_INVERTED
- GL_NOOP
- GL_XOR
- GL_OR
- GL_NOR
- GL_EQUIV
- GL_INVERT
- GL_OR_REVERSE
- GL_COPY_INVERTED
- GL_OR_INVERTED
- GL_NAND
- GL_SET
- GL_EMISSION

- GL_SHININESS
- GL_AMBIENT_AND_DIFFUSE
- GL_COLOR_INDEXES
- GL_MODELVIEW
- GL_PROJECTION
- GL_TEXTURE
- GL_COLOR
- GL_DEPTH
- GL_STENCIL
- GL_COLOR_INDEX
- GL_STENCIL_INDEX
- GL_DEPTH_COMPONENT
- GL_RED
- GL_GREEN
- GL_BLUE
- GL_ALPHA
- GL_RGB
- GL_RGBA
- GL_LUMINANCE
- GL_LUMINANCE_ALPHA
- GL_BITMAP
- GL_POINT
- GL_LINE
- GL_FILL
- GL_RENDER
- GL_FEEDBACK
- GL_SELECT
- GL_FLAT
- GL_SMOOTH
- GL_KEEP
- GL_REPLACE
- GL_INCR
- GL_DECR
- GL_VENDOR
- GL_RENDERER
- GL_VERSION

- GL_EXTENSIONS
- GL_S
- GL_ENABLE_BIT
- GL_T
- GL_R
- GL_Q
- GL_MODULATE
- GL_DECAL
- GL_TEXTURE_ENV_MODE
- GL_TEXTURE_ENV_COLOR
- GL_TEXTURE_ENV
- GL_EYE_LINEAR
- GL_OBJECT_LINEAR
- GL_SPHERE_MAP
- GL_TEXTURE_GEN_MODE
- GL_OBJECT_PLANE
- GL_EYE_PLANE
- GL_NEAREST
- GL_LINEAR
- GL_NEAREST_MIPMAP_NEAREST
- GL_LINEAR_MIPMAP_NEAREST
- GL_NEAREST_MIPMAP_LINEAR
- GL_LINEAR_MIPMAP_LINEAR
- GL_TEXTURE_MAG_FILTER
- GL_TEXTURE_MIN_FILTER
- GL_TEXTURE_WRAP_S
- GL_TEXTURE_WRAP_T
- GL_CLAMP
- GL_REPEAT
- GL_POLYGON_OFFSET_UNITS
- GL_POLYGON_OFFSET_POINT
- GL_POLYGON_OFFSET_LINE
- GL_R3_G3_B2
- GL_V2F
- GL_V3F
- GL_C4UB_V2F

- GL_C4UB_V3F
- GL_C3F_V3F
- GL_N3F_V3F
- GL_C4F_N3F_V3F
- GL_T2F_V3F
- GL_T4F_V4F
- GL_T2F_C4UB_V3F
- GL_T2F_C3F_V3F
- GL_T2F_N3F_V3F
- GL_T2F_C4F_N3F_V3F
- GL_T4F_C4F_N3F_V4F
- GL_CLIP_PLANE0
- GL_CLIP_PLANE1
- GL_CLIP_PLANE2
- GL_CLIP_PLANE3
- GL_CLIP_PLANE4
- GL_CLIP_PLANE5
- GL_LIGHT0
- GL_COLOR_BUFFER_BIT
- GL_LIGHT1
- GL_LIGHT2
- GL_LIGHT3
- GL_LIGHT4
- GL_LIGHT5
- GL_LIGHT6
- GL_LIGHT7
- GL_HINT_BIT
- GL_POLYGON_OFFSET_FILL
- GL_POLYGON_OFFSET_FACTOR
- GL_ALPHA4
- GL_ALPHA8
- GL_ALPHA12
- GL_ALPHA16
- GL_LUMINANCE4
- GL_LUMINANCE8
- GL_LUMINANCE12

- GL_LUMINANCE16
- GL_LUMINANCE4_ALPHA4
- GL_LUMINANCE6_ALPHA2
- GL_LUMINANCE8_ALPHA8
- GL_LUMINANCE12_ALPHA4
- GL_LUMINANCE12_ALPHA12
- GL_LUMINANCE16_ALPHA16
- GL_INTENSITY
- GL_INTENSITY4
- GL_INTENSITY8
- GL_INTENSITY12
- GL_INTENSITY16
- GL_RGB4
- GL_RGB5
- GL_RGB8
- GL_RGB10
- GL_RGB12
- GL_RGB16
- GL_RGBA2
- GL_RGBA4
- GL_RGB5_A1
- GL_RGBA8
- GL_RGB10_A2
- GL_RGBA12
- GL_RGBA16
- GL_TEXTURE_RED_SIZE
- GL_TEXTURE_GREEN_SIZE
- GL_TEXTURE_BLUE_SIZE
- GL_TEXTURE_ALPHA_SIZE
- GL_TEXTURE_LUMINANCE_SIZE
- GL_TEXTURE_INTENSITY_SIZE
- GL_PROXY_TEXTURE_1D
- GL_PROXY_TEXTURE_2D
- GL_TEXTURE_PRIORITY
- GL_TEXTURE_RESIDENT
- GL_TEXTURE_BINDING_1D

- GL_TEXTURE_BINDING_2D
- GL_VERTEX_ARRAY
- GL_NORMAL_ARRAY
- GL_COLOR_ARRAY
- GL_INDEX_ARRAY
- GL_TEXTURE_COORD_ARRAY
- GL_EDGE_FLAG_ARRAY
- GL_VERTEX_ARRAY_SIZE
- GL_VERTEX_ARRAY_TYPE
- GL_VERTEX_ARRAY_STRIDE
- GL_NORMAL_ARRAY_TYPE
- GL_NORMAL_ARRAY_STRIDE
- GL_COLOR_ARRAY_SIZE
- GL_COLOR_ARRAY_TYPE
- GL_COLOR_ARRAY_STRIDE
- GL_INDEX_ARRAY_TYPE
- GL_INDEX_ARRAY_STRIDE
- GL_TEXTURE_COORD_ARRAY_SIZE
- GL_TEXTURE_COORD_ARRAY_TYPE
- GL_TEXTURE_COORD_ARRAY_STRIDE
- GL_EDGE_FLAG_ARRAY_STRIDE
- GL_VERTEX_ARRAY_POINTER
- GL_NORMAL_ARRAY_POINTER
- GL_COLOR_ARRAY_POINTER
- GL_INDEX_ARRAY_POINTER
- GL_TEXTURE_COORD_ARRAY_POINTER
- GL_EDGE_FLAG_ARRAY_POINTER
- GL_COLOR_INDEX1_EXT
- GL_COLOR_INDEX2_EXT
- GL_COLOR_INDEX4_EXT
- GL_COLOR_INDEX8_EXT
- GL_COLOR_INDEX12_EXT
- GL_COLOR_INDEX16_EXT
- GL_EVAL_BIT
- GL_LIST_BIT
- GL_TEXTURE_BIT

- GL_SCISSOR_BIT
- GL_ALL_ATTRIB_BITS
- GL_CLIENT_ALL_ATTRIB_BITS
- GL_SMOOTH_POINT_SIZE_RANGE
- GL_SMOOTH_POINT_SIZE_GRANULARITY
- GL_SMOOTH_LINE_WIDTH_RANGE
- GL_SMOOTH_LINE_WIDTH_GRANULARITY
- GL_UNSIGNED_BYTE_3_3_2
- GL_UNSIGNED_SHORT_4_4_4_4
- GL_UNSIGNED_SHORT_5_5_5_1
- GL_UNSIGNED_INT_8_8_8_8
- GL_UNSIGNED_INT_10_10_10_2
- GL_RESCALE_NORMAL
- GL_TEXTURE_BINDING_3D
- GL_PACK_SKIP_IMAGES
- GL_PACK_IMAGE_HEIGHT
- GL_UNPACK_SKIP_IMAGES
- GL_UNPACK_IMAGE_HEIGHT
- GL_TEXTURE_3D
- GL_PROXY_TEXTURE_3D
- GL_TEXTURE_DEPTH
- GL_TEXTURE_WRAP_R
- GL_MAX_3D_TEXTURE_SIZE
- GL_BGR
- GL_BGRA
- GL_MAX_ELEMENTS_VERTICES
- GL_MAX_ELEMENTS_INDICES
- GL_CLAMP_TO_EDGE
- GL_TEXTURE_MIN_LOD
- GL_TEXTURE_MAX_LOD
- GL_TEXTURE_BASE_LEVEL
- GL_TEXTURE_MAX_LEVEL
- GL_LIGHT_MODEL_COLOR_CONTROL
- GL_SINGLE_COLOR
- GL_SEPARATE_SPECULAR_COLOR
- GL_UNSIGNED_BYTE_2_3_3_REV

- GL_UNSIGNED_SHORT_5_6_5
- GL_UNSIGNED_SHORT_5_6_5_REV
- GL_UNSIGNED_SHORT_4_4_4_4_REV
- GL_UNSIGNED_SHORT_1_5_5_5_REV
- GL_UNSIGNED_INT_8_8_8_8_REV
- GL_ALIASED_POINT_SIZE_RANGE
- GL_ALIASED_LINE_WIDTH_RANGE
- GL_MULTISAMPLE
- GL_SAMPLE_ALPHA_TO_COVERAGE
- GL_SAMPLE_ALPHA_TO_ONE
- GL_SAMPLE_COVERAGE
- GL_SAMPLE_BUFFERS
- GL_SAMPLES
- GL_SAMPLE_COVERAGE_VALUE
- GL_SAMPLE_COVERAGE_INVERT
- GL_CLAMP_TO_BORDER
- GL_TEXTURE0
- GL_TEXTURE1
- GL_TEXTURE2
- GL_TEXTURE3
- GL_TEXTURE4
- GL_TEXTURE5
- GL_TEXTURE6
- GL_TEXTURE7
- GL_TEXTURE8
- GL_TEXTURE9
- GL_TEXTURE10
- GL_TEXTURE11
- GL_TEXTURE12
- GL_TEXTURE13
- GL_TEXTURE14
- GL_TEXTURE15
- GL_TEXTURE16
- GL_TEXTURE17
- GL_TEXTURE18
- GL_TEXTURE19

- GL_TEXTURE20
- GL_TEXTURE21
- GL_TEXTURE22
- GL_TEXTURE23
- GL_TEXTURE24
- GL_TEXTURE25
- GL_TEXTURE26
- GL_TEXTURE27
- GL_TEXTURE28
- GL_TEXTURE29
- GL_TEXTURE30
- GL_TEXTURE31
- GL_ACTIVE_TEXTURE
- GL_CLIENT_ACTIVE_TEXTURE
- GL_MAX_TEXTURE_UNITS
- GL_TRANSPOSE_MODELVIEW_MATRIX
- GL_TRANSPOSE_PROJECTION_MATRIX
- GL_TRANSPOSE_TEXTURE_MATRIX
- GL_TRANSPOSE_COLOR_MATRIX
- GL_SUBTRACT
- GL_COMPRESSED_ALPHA
- GL_COMPRESSED_LUMINANCE
- GL_COMPRESSED_LUMINANCE_ALPHA
- GL_COMPRESSED_INTENSITY
- GL_COMPRESSED_RGB
- GL_COMPRESSED_RGBA
- GL_TEXTURE_COMPRESSION_HINT
- GL_NORMAL_MAP
- GL_REFLECTION_MAP
- GL_TEXTURE_CUBE_MAP
- GL_TEXTURE_BINDING_CUBE_MAP
- GL_TEXTURE_CUBE_MAP_POSITIVE_X
- GL_TEXTURE_CUBE_MAP_NEGATIVE_X
- GL_TEXTURE_CUBE_MAP_POSITIVE_Y
- GL_TEXTURE_CUBE_MAP_NEGATIVE_Y
- GL_TEXTURE_CUBE_MAP_POSITIVE_Z

- GL_TEXTURE_CUBE_MAP_NEGATIVE_Z
- GL_PROXY_TEXTURE_CUBE_MAP
- GL_MAX_CUBE_MAP_TEXTURE_SIZE
- GL_COMBINE
- GL_COMBINE_RGB
- GL_COMBINE_ALPHA
- GL_RGB_SCALE
- GL_ADD_SIGNED
- GL_INTERPOLATE
- GL_CONSTANT
- GL_PRIMARY_COLOR
- GL_PREVIOUS
- GL_SOURCE0_RGB
- GL_SOURCE1_RGB
- GL_SOURCE2_RGB
- GL_SOURCE0_ALPHA
- GL_SOURCE1_ALPHA
- GL_SOURCE2_ALPHA
- GL_OPERAND0_RGB
- GL_OPERAND1_RGB
- GL_OPERAND2_RGB
- GL_OPERAND0_ALPHA
- GL_OPERAND1_ALPHA
- GL_OPERAND2_ALPHA
- GL_TEXTURE_COMPRESSED_IMAGE_SIZE
- GL_TEXTURE_COMPRESSED
- GL_NUM_COMPRESSED_TEXTURE_FORMATS
- GL_COMPRESSED_TEXTURE_FORMATS
- GL_DOT3_RGB
- GL_DOT3_RGBA
- GL_MULTISAMPLE_BIT
- GL_BLEND_DST_RGB
- GL_BLEND_SRC_RGB
- GL_BLEND_DST_ALPHA
- GL_BLEND_SRC_ALPHA
- GL_POINT_SIZE_MIN

- GL_POINT_SIZE_MAX
- GL_POINT_FADE_THRESHOLD_SIZE
- GL_POINT_DISTANCE_ATTENUATION
- GL_GENERATE_MIPMAP
- GL_GENERATE_MIPMAP_HINT
- GL_DEPTH_COMPONENT16
- GL_DEPTH_COMPONENT24
- GL_DEPTH_COMPONENT32
- GL_MIRRORED_REPEAT
- GL_FOG_COORDINATE_SOURCE
- GL_FOG_COORDINATE
- GL_FRAGMENT_DEPTH
- GL_CURRENT_FOG_COORDINATE
- GL_FOG_COORDINATE_ARRAY_TYPE
- GL_FOG_COORDINATE_ARRAY_STRIDE
- GL_FOG_COORDINATE_ARRAY_POINTER
- GL_FOG_COORDINATE_ARRAY
- GL_COLOR_SUM
- GL_CURRENT_SECONDARY_COLOR
- GL_SECONDARY_COLOR_ARRAY_SIZE
- GL_SECONDARY_COLOR_ARRAY_TYPE
- GL_SECONDARY_COLOR_ARRAY_STRIDE
- GL_SECONDARY_COLOR_ARRAY_POINTER
- GL_SECONDARY_COLOR_ARRAY
- GL_MAX_TEXTURE_LOD_BIAS
- GL_TEXTURE_FILTER_CONTROL
- GL_TEXTURE_LOD_BIAS
- GL_INCR_WRAP
- GL_DECR_WRAP
- GL_TEXTURE_DEPTH_SIZE
- GL_DEPTH_TEXTURE_MODE
- GL_TEXTURE_COMPARE_MODE
- GL_TEXTURE_COMPARE_FUNC
- GL_COMPARE_R_TO_TEXTURE
- GL_CURRENT_FOG_COORD
- GL_FOG_COORD

- GL_FOG_COORD_ARRAY
- GL_FOG_COORD_ARRAY_BUFFER_BINDING
- GL_FOG_COORD_ARRAY_POINTER
- GL_FOG_COORD_ARRAY_STRIDE
- GL_FOG_COORD_ARRAY_TYPE
- GL_FOG_COORD_SRC
- GL_SRC0_ALPHA
- GL_SRC0_RGB
- GL_SRC1_ALPHA
- GL_SRC1_RGB
- GL_SRC2_ALPHA
- GL_SRC2_RGB
- GL_BUFFER_SIZE
- GL_BUFFER_USAGE
- GL_QUERY_COUNTER_BITS
- GL_CURRENT_QUERY
- GL_QUERY_RESULT
- GL_QUERY_RESULT_AVAILABLE
- GL_ARRAY_BUFFER
- GL_ELEMENT_ARRAY_BUFFER
- GL_ARRAY_BUFFER_BINDING
- GL_ELEMENT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ARRAY_BUFFER_BINDING
- GL_NORMAL_ARRAY_BUFFER_BINDING
- GL_COLOR_ARRAY_BUFFER_BINDING
- GL_INDEX_ARRAY_BUFFER_BINDING
- GL_TEXTURE_COORD_ARRAY_BUFFER_BINDING
- GL_EDGE_FLAG_ARRAY_BUFFER_BINDING
- GL_SECONDARY_COLOR_ARRAY_BUFFER_BINDING
- GL_FOG_COORDINATE_ARRAY_BUFFER_BINDING
- GL_WEIGHT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ATTRIB_ARRAY_BUFFER_BINDING
- GL_READ_ONLY
- GL_WRITE_ONLY
- GL_READ_WRITE
- GL_BUFFER_ACCESS

- GL_BUFFER_MAPPED
- GL_BUFFER_MAP_POINTER
- GL_STREAM_DRAW
- GL_STREAM_READ
- GL_STREAM_COPY
- GL_STATIC_DRAW
- GL_STATIC_READ
- GL_STATIC_COPY
- GL_DYNAMIC_DRAW
- GL_DYNAMIC_READ
- GL_DYNAMIC_COPY
- GL_SAMPLES_PASSED
- GL_BLEND_EQUATION_RGB
- GL_VERTEX_ATTRIB_ARRAY_ENABLED
- GL_VERTEX_ATTRIB_ARRAY_SIZE
- GL_VERTEX_ATTRIB_ARRAY_STRIDE
- GL_VERTEX_ATTRIB_ARRAY_TYPE
- GL_CURRENT_VERTEX_ATTRIB
- GL_VERTEX_PROGRAM_POINT_SIZE
- GL_VERTEX_PROGRAM_TWO_SIDE
- GL_VERTEX_ATTRIB_ARRAY_POINTER
- GL_STENCIL_BACK_FUNC
- GL_STENCIL_BACK_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_PASS
- GL_MAX_DRAW_BUFFERS
- GL_DRAW_BUFFER0
- GL_DRAW_BUFFER1
- GL_DRAW_BUFFER2
- GL_DRAW_BUFFER3
- GL_DRAW_BUFFER4
- GL_DRAW_BUFFER5
- GL_DRAW_BUFFER6
- GL_DRAW_BUFFER7
- GL_DRAW_BUFFER8
- GL_DRAW_BUFFER9

- GL_DRAW_BUFFER10
- GL_DRAW_BUFFER11
- GL_DRAW_BUFFER12
- GL_DRAW_BUFFER13
- GL_DRAW_BUFFER14
- GL_DRAW_BUFFER15
- GL_BLEND_EQUATION_ALPHA
- GL_POINT_SPRITE
- GL_COORD_REPLACE
- GL_MAX_VERTEX_ATTRIBS
- GL_VERTEX_ATTRIB_ARRAY_NORMALIZED
- GL_MAX_TEXTURE_COORDS
- GL_MAX_TEXTURE_IMAGE_UNITS
- GL_FRAGMENT_SHADER
- GL_VERTEX_SHADER
- GL_MAX_FRAGMENT_UNIFORM_COMPONENTS
- GL_MAX_VERTEX_UNIFORM_COMPONENTS
- GL_MAX_VARYING_FLOATS
- GL_MAX_VERTEX_TEXTURE_IMAGE_UNITS
- GL_MAX_COMBINED_TEXTURE_IMAGE_UNITS
- GL_SHADER_TYPE
- GL_FLOAT_VEC2
- GL_FLOAT_VEC3
- GL_FLOAT_VEC4
- GL_INT_VEC2
- GL_INT_VEC3
- GL_INT_VEC4
- GL_BOOL
- GL_BOOL_VEC2
- GL_BOOL_VEC3
- GL_BOOL_VEC4
- GL_FLOAT_MAT2
- GL_FLOAT_MAT3
- GL_FLOAT_MAT4
- GL_SAMPLER_1D
- GL_SAMPLER_2D

- GL_SAMPLER_3D
- GL_SAMPLER_CUBE
- GL_SAMPLER_1D_SHADOW
- GL_SAMPLER_2D_SHADOW
- GL_DELETE_STATUS
- GL_COMPILE_STATUS
- GL_LINK_STATUS
- GL_VALIDATE_STATUS
- GL_INFO_LOG_LENGTH
- GL_ATTACHED_SHADERS
- GL_ACTIVE_UNIFORMS
- GL_ACTIVE_UNIFORM_MAX_LENGTH
- GL_SHADER_SOURCE_LENGTH
- GL_ACTIVE_ATTRIBUTES
- GL_ACTIVE_ATTRIBUTE_MAX_LENGTH
- GL_FRAGMENT_SHADER_DERIVATIVE_HINT
- GL_SHADING_LANGUAGE_VERSION
- GL_CURRENT_PROGRAM
- GL_POINT_SPRITE_COORD_ORIGIN
- GL_LOWER_LEFT
- GL_UPPER_LEFT
- GL_STENCIL_BACK_REF
- GL_STENCIL_BACK_VALUE_MASK
- GL_STENCIL_BACK_WRITEMASK
- GL_CURRENT_RASTER_SECONDARY_COLOR
- GL_PIXEL_PACK_BUFFER
- GL_PIXEL_UNPACK_BUFFER
- GL_PIXEL_PACK_BUFFER_BINDING
- GL_PIXEL_UNPACK_BUFFER_BINDING
- GL_FLOAT_MAT2x3
- GL_FLOAT_MAT2x4
- GL_FLOAT_MAT3x2
- GL_FLOAT_MAT3x4
- GL_FLOAT_MAT4x2
- GL_FLOAT_MAT4x3
- GL_SRGB

- GL_SRGB8
- GL_SRGB_ALPHA
- GL_SRGB8_ALPHA8
- GL_SLUMINANCE_ALPHA
- GL_SLUMINANCE8_ALPHA8
- GL_SLUMINANCE
- GL_SLUMINANCE8
- GL_COMPRESSED_SRGB
- GL_COMPRESSED_SRGB_ALPHA
- GL_COMPRESSED_SLUMINANCE
- GL_COMPRESSED_SLUMINANCE_ALPHA
- GL_CLIP_DISTANCE0
- GL_CLIP_DISTANCE1
- GL_CLIP_DISTANCE2
- GL_CLIP_DISTANCE3
- GL_CLIP_DISTANCE4
- GL_CLIP_DISTANCE5
- GL_COMPARE_REF_TO_TEXTURE
- GL_MAX_CLIP_DISTANCES
- GL_MAX_VARYING_COMPONENTS
- GL_CONTEXT_FLAG_FORWARD_COMPATIBLE_BIT
- GL_MAJOR_VERSION
- GL_MINOR_VERSION
- GL_NUM_EXTENSIONS
- GL_CONTEXT_FLAGS
- GL_DEPTH_BUFFER
- GL_STENCIL_BUFFER
- GL_RGBA32F
- GL_RGB32F
- GL_RGBA16F
- GL_RGB16F
- GL_VERTEX_ATTRIB_ARRAY_INTEGER
- GL_MAX_ARRAY_TEXTURE_LAYERS
- GL_MIN_PROGRAM_TEXEL_OFFSET
- GL_MAX_PROGRAM_TEXEL_OFFSET
- GL_CLAMP_VERTEX_COLOR

- GL_CLAMP_FRAGMENT_COLOR
- GL_CLAMP_READ_COLOR
- GL_FIXED_ONLY
- GL_TEXTURE_RED_TYPE
- GL_TEXTURE_GREEN_TYPE
- GL_TEXTURE_BLUE_TYPE
- GL_TEXTURE_ALPHA_TYPE
- GL_TEXTURE_LUMINANCE_TYPE
- GL_TEXTURE_INTENSITY_TYPE
- GL_TEXTURE_DEPTH_TYPE
- GL_TEXTURE_1D_ARRAY
- GL_PROXY_TEXTURE_1D_ARRAY
- GL_TEXTURE_2D_ARRAY
- GL_PROXY_TEXTURE_2D_ARRAY
- GL_TEXTURE_BINDING_1D_ARRAY
- GL_TEXTURE_BINDING_2D_ARRAY
- GL_R11F_G11F_B10F
- GL_UNSIGNED_INT_10F_11F_11F_REV
- GL_RGB9_E5
- GL_UNSIGNED_INT_5_9_9_9_REV
- GL_TEXTURE_SHARED_SIZE
- GL_TRANSFORM_FEEDBACK_VARYING_MAX_LENGTH
- GL_TRANSFORM_FEEDBACK_BUFFER_MODE
- GL_MAX_TRANSFORM_FEEDBACK_SEPARATE_COMPONENTS
- GL_TRANSFORM_FEEDBACK_VARYINGS
- GL_TRANSFORM_FEEDBACK_BUFFER_START
- GL_TRANSFORM_FEEDBACK_BUFFER_SIZE
- GL_PRIMITIVES_GENERATED
- GL_TRANSFORM_FEEDBACK_PRIMITIVES_WRITTEN
- GL_RASTERIZER_DISCARD
- GL_MAX_TRANSFORM_FEEDBACK_INTERLEAVED_COMPONENTS
- GL_MAX_TRANSFORM_FEEDBACK_SEPARATE_ATTRIBS
- GL_INTERLEAVED_ATTRIBS
- GL_SEPARATE_ATTRIBS
- GL_TRANSFORM_FEEDBACK_BUFFER
- GL_TRANSFORM_FEEDBACK_BUFFER_BINDING

- GL_RGBA32UI
- GL_RGB32UI
- GL_RGBA16UI
- GL_RGB16UI
- GL_RGBA8UI
- GL_RGB8UI
- GL_RGBA32I
- GL_RGB32I
- GL_RGBA16I
- GL_RGB16I
- GL_RGBA8I
- GL_RGB8I
- GL_RED_INTEGER
- GL_GREEN_INTEGER
- GL_BLUE_INTEGER
- GL_ALPHA_INTEGER
- GL_RGB_INTEGER
- GL_RGBA_INTEGER
- GL_BGR_INTEGER
- GL_BGRA_INTEGER
- GL_SAMPLER_1D_ARRAY
- GL_SAMPLER_2D_ARRAY
- GL_SAMPLER_1D_ARRAY_SHADOW
- GL_SAMPLER_2D_ARRAY_SHADOW
- GL_SAMPLER_CUBE_SHADOW
- GL_UNSIGNED_INT_VEC2
- GL_UNSIGNED_INT_VEC3
- GL_UNSIGNED_INT_VEC4
- GL_INT_SAMPLER_1D
- GL_INT_SAMPLER_2D
- GL_INT_SAMPLER_3D
- GL_INT_SAMPLER_CUBE
- GL_INT_SAMPLER_1D_ARRAY
- GL_INT_SAMPLER_2D_ARRAY
- GL_UNSIGNED_INT_SAMPLER_1D
- GL_UNSIGNED_INT_SAMPLER_2D

- GL_UNSIGNED_INT_SAMPLER_3D
- GL_UNSIGNED_INT_SAMPLER_CUBE
- GL_UNSIGNED_INT_SAMPLER_1D_ARRAY
- GL_UNSIGNED_INT_SAMPLER_2D_ARRAY
- GL_QUERY_WAIT
- GL_QUERY_NO_WAIT
- GL_QUERY_BY_REGION_WAIT
- GL_QUERY_BY_REGION_NO_WAIT
- GL_TEXTURE_RECTANGLE
- GL_TEXTURE_BINDING_RECTANGLE
- GL_PROXY_TEXTURE_RECTANGLE
- GL_MAX_RECTANGLE_TEXTURE_SIZE
- GL_SAMPLER_2D_RECT
- GL_SAMPLER_2D_RECT_SHADOW
- GL_TEXTURE_BUFFER
- GL_MAX_TEXTURE_BUFFER_SIZE
- GL_TEXTURE_BINDING_BUFFER
- GL_TEXTURE_BUFFER_DATA_STORE_BINDING
- GL_TEXTURE_BUFFER_FORMAT
- GL_SAMPLER_BUFFER
- GL_INT_SAMPLER_2D_RECT
- GL_INT_SAMPLER_BUFFER
- GL_UNSIGNED_INT_SAMPLER_2D_RECT
- GL_UNSIGNED_INT_SAMPLER_BUFFER
- GL_RED_SNORM
- GL_RG_SNORM
- GL_RGB_SNORM
- GL_RGBA_SNORM
- GL_R8_SNORM
- GL_RG8_SNORM
- GL_RGB8_SNORM
- GL_RGBA8_SNORM
- GL_R16_SNORM
- GL_RG16_SNORM
- GL_RGB16_SNORM
- GL_RGBA16_SNORM

- GL_SIGNED_NORMALIZED
- GL_PRIMITIVE_RESTART
- GL_PRIMITIVE_RESTART_INDEX
- GL_BUFFER_ACCESS_FLAGS
- GL_BUFFER_MAP_LENGTH
- GL_BUFFER_MAP_OFFSET
- GL_CONTEXT_CORE_PROFILE_BIT
- GL_CONTEXT_COMPATIBILITY_PROFILE_BIT
- GL_LINES_ADJACENCY
- GL_LINE_STRIP_ADJACENCY
- GL_TRIANGLES_ADJACENCY
- GL_TRIANGLE_STRIP_ADJACENCY
- GL_PROGRAM_POINT_SIZE
- GL_GEOMETRY_VERTICES_OUT
- GL_GEOMETRY_INPUT_TYPE
- GL_GEOMETRY_OUTPUT_TYPE
- GL_MAX_GEOMETRY_TEXTURE_IMAGE_UNITS
- GL_FRAMEBUFFER_ATTACHMENT_LAYERED
- GL_FRAMEBUFFER_INCOMPLETE_LAYER_TARGETS
- GL_GEOMETRY_SHADER
- GL_MAX_GEOMETRY_UNIFORM_COMPONENTS
- GL_MAX_GEOMETRY_OUTPUT_VERTICES
- GL_MAX_GEOMETRY_TOTAL_OUTPUT_COMPONENTS
- GL_MAX_VERTEX_OUTPUT_COMPONENTS
- GL_MAX_GEOMETRY_INPUT_COMPONENTS
- GL_MAX_GEOMETRY_OUTPUT_COMPONENTS
- GL_MAX_FRAGMENT_INPUT_COMPONENTS
- GL_CONTEXT_PROFILE_MASK
- GL_VERTEX_ATTRIB_ARRAY_DIVISOR
- GL_RGB10_A2UI
- GL_SAMPLE_SHADING
- GL_MIN_SAMPLE_SHADING_VALUE
- GL_MIN_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_COMPONENTS
- GL_TEXTURE_CUBE_MAP_ARRAY

- GL_TEXTURE_BINDING_CUBE_MAP_ARRAY
- GL_PROXY_TEXTURE_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY_SHADOW
- GL_INT_SAMPLER_CUBE_MAP_ARRAY
- GL_UNSIGNED_INT_SAMPLER_CUBE_MAP_ARRAY
- void glAccum(GLenum op, GLfloat value)
- void glActiveTexture(GLenum texture)
- void glAlphaFunc(GLenum func, GLclampf ref)
- GLboolean glAreTexturesResident(GLsizei n, const GLuint * textures, GLboolean * residences)
- void glArrayElement(GLint i)
- void glAttachShader(GLuint program, GLuint shader)
- void glBegin(GLenum mode)
- void glBeginQuery(GLenum target, GLuint id)
- void glBindAttribLocation(GLuint program, GLuint index, const GLchar * name)
- void glBindBuffer(GLenum target, GLuint buffer)
- void glBindTexture(GLenum target, GLuint texture)
- void glBitmap(GLsizei width, GLsizei height, GLfloat xorig, GLfloat yorig, GLfloat xmove, GLfloat ymove, const GLubyte * bitmap)
- void glBlendColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)
- void glBlendEquation(GLenum mode)
- void glBlendEquationSeparate(GLenum modeRGB, GLenum modeAlpha)
- void glBlendFunc(GLenum sfactor, GLenum dfactor)
- void glBlendFuncSeparate(GLenum srcRGB, GLenum dstRGB, GLenum srcAlpha, GLenum dstAlpha)
- void glBufferData(GLenum target, GLsizeiptr size, const GLvoid * data, GLenum usage)
- void glBufferSubData(GLenum target, GLintptr offset, GLsizeiptr size, const GLvoid * data)
- void glCallList(GLuint list)
- void glCallLists(GLsizei n, GLenum type, const GLvoid * lists)
- void glClear(GLbitfield mask)
- void glClearAccum(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glClearColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)
- void glClearDepth(GLclampd depth)
- void glClearIndex(GLfloat c)
- void glClearStencil(GLint s)
- void glClientActiveTexture(GLenum texture)
- void glClipPlane(GLenum plane, const GLdouble * equation)

- `void glColor3b(GLbyte red, GLbyte green, GLbyte blue)`
- `void glColor3s(GLshort red, GLshort green, GLshort blue)`
- `void glColor3i(GLint red, GLint green, GLint blue)`
- `void glColor3f(GLfloat red, GLfloat green, GLfloat blue)`
- `void glColor3d(GLdouble red, GLdouble green, GLdouble blue)`
- `void glColor3ub(GLubyte red, GLubyte green, GLubyte blue)`
- `void glColor3us(GLushort red, GLushort green, GLushort blue)`
- `void glColor3ui(GLuint red, GLuint green, GLuint blue)`
- `void glColor4b(GLbyte red, GLbyte green, GLbyte blue, GLbyte alpha)`
- `void glColor4s(GLshort red, GLshort green, GLshort blue, GLshort alpha)`
- `void glColor4i(GLint red, GLint green, GLint blue, GLint alpha)`
- `void glColor4f(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)`
- `void glColor4d(GLdouble red, GLdouble green, GLdouble blue, GLdouble alpha)`
- `void glColor4ub(GLubyte red, GLubyte green, GLubyte blue, GLubyte alpha)`
- `void glColor4us(GLushort red, GLushort green, GLushort blue, GLushort alpha)`
- `void glColor4ui(GLuint red, GLuint green, GLuint blue, GLuint alpha)`
- `void glColor3bv(const GLbyte * v)`
- `void glColor3sv(const GLshort * v)`
- `void glColor3iv(const GLint * v)`
- `void glColor3fv(const GLfloat * v)`
- `void glColor3dv(const GLdouble * v)`
- `void glColor3ubv(const GLubyte * v)`
- `void glColor3usv(const GLushort * v)`
- `void glColor3uiv(const GLuint * v)`
- `void glColor4bv(const GLbyte * v)`
- `void glColor4sv(const GLshort * v)`
- `void glColor4iv(const GLint * v)`
- `void glColor4fv(const GLfloat * v)`
- `void glColor4dv(const GLdouble * v)`
- `void glColor4ubv(const GLubyte * v)`
- `void glColor4usv(const GLushort * v)`
- `void glColor4uiv(const GLuint * v)`
- `void glColorMask(GLboolean red, GLboolean green, GLboolean blue, GLboolean alpha)`
- `void glColorMaterial(GLenum face, GLenum mode)`
- `void glColorPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)`

- void glColorSubTable(GLenum target, GLsizei start, GLsizei count, GLenum format, GLenum type, const GLvoid * data)
- void glColorTable(GLenum target, GLenum internalformat, GLsizei width, GLenum format, GLenum type, const GLvoid * data)
- void glColorTableParameterfv(GLenum target, GLenum pname, const GLfloat * params)
- void glColorTableParameteriv(GLenum target, GLenum pname, const GLint * params)
- void glCompileShader(GLuint shader)
- void glCompressedTexImage1D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLint border, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexImage2D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLint border, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexImage3D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLsizei depth, GLint border, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLsizei imageSize, const GLvoid * data)
- void glConvolutionFilter1D(GLenum target, GLenum internalformat, GLsizei width, GLenum format, GLenum type, const GLvoid * data)
- void glConvolutionFilter2D(GLenum target, GLenum internalformat, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)
- void glConvolutionParameterf(GLenum target, GLenum pname, GLfloat params)
- void glConvolutionParameteri(GLenum target, GLenum pname, GLint params)
- void glConvolutionParameterfv(GLenum target, GLenum pname, const GLfloat * params)
- void glConvolutionParameteriv(GLenum target, GLenum pname, const GLint * params)
- void glCopyColorSubTable(GLenum target, GLsizei start, GLint x, GLint y, GLsizei width)
- void glCopyColorTable(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width)
- void glCopyConvolutionFilter1D(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width)
- void glCopyConvolutionFilter2D(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width, GLsizei height)
- void glCopyPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum type)
- void glCopyTexImage1D(GLenum target, GLint level, GLenum internalformat, GLint x, GLint y, GLsizei width, GLint border)
- void glCopyTexImage2D(GLenum target, GLint level, GLenum internalformat, GLint x, GLint y, GLsizei width, GLsizei height, GLint border)
- void glCopyTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLint x, GLint y, GLsizei width)
- void glCopyTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint x, GLint y, GLsizei width, GLsizei height)

- `void glCopyTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLint x, GLint y, GLsizei width, GLsizei height)`
- `GLuint glCreateProgram(void)`
- `GLuint glCreateShader(GLenum shaderType)`
- `void glCullFace(GLenum mode)`
- `void glDeleteBuffers(GLsizei n, const GLuint * buffers)`
- `void glDeleteLists(GLuint list, GLsizei range)`
- `void glDeleteProgram(GLuint program)`
- `void glDeleteQueries(GLsizei n, const GLuint * ids)`
- `void glDeleteShader(GLuint shader)`
- `void glDeleteTextures(GLsizei n, const GLuint * textures)`
- `void glDepthFunc(GLenum func)`
- `void glDepthMask(GLboolean flag)`
- `void glDepthRange(GLclampd nearVal, GLclampd farVal)`
- `void glDetachShader(GLuint program, GLuint shader)`
- `void glEnable(GLenum cap)`
- `void glEnableClientState(GLenum cap)`
- `void glEnableVertexAttribArray(GLuint index)`
- `void glDisableVertexAttribArray(GLuint index)`
- `void glDrawArrays(GLenum mode, GLint first, GLsizei count)`
- `void glDrawBuffer(GLenum mode)`
- `void glDrawBuffers(GLsizei n, const GLenum *bufs)`
- `void glDrawElements(GLenum mode, GLsizei count, GLenum type, const GLvoid * indices)`
- `void glDrawPixels(GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)`
- `void glDrawRangeElements(GLenum mode, GLuint start, GLuint end, GLsizei count, GLenum type, const GLvoid * indices)`
- `void glEdgeFlag(GLboolean flag)`
- `void glEdgeFlagPointer(GLsizei stride, const GLvoid * pointer)`
- `void glEnd(void)`
- `void glEndList(void)`
- `void glEndQuery(GLenum target)`
- `void glEvalCoord1f(GLfloat u)`
- `void glEvalCoord1d(GLdouble u)`
- `void glEvalCoord2f(GLfloat u, GLfloat v)`
- `void glEvalCoord2d(GLdouble u, GLdouble v)`
- `void glEvalMesh1(GLenum mode, GLint i1, GLint i2)`
- `void glEvalPoint1(GLint i)`

- void glEvalPoint2(GLint i, GLint j)
- void glFeedbackBuffer(GLsizei size, GLenum type, GLfloat * buffer)
- void glFinish(void)
- void glFlush(void)
- void glFogf(GLenum pname, GLfloat param)
- void glFogi(GLenum pname, GLint param)
- void glFogfv(GLenum pname, const GLfloat * params)
- void glFogiv(GLenum pname, const GLint * params)
- void glFogCoordd(GLdouble coord)
- void glFogCoordf(GLfloat coord)
- void glFogCoorddv(GLdouble * coord)
- void glFogCoordfv(GLfloat * coord)
- void glFogCoordPointer(GLenum type, GLsizei stride, GLvoid * pointer)
- void glFrontFace(GLenum mode)
- void glFrustum(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble nearVal, GLdouble farVal)
- void glGenBuffers(GLsizei n, GLuint * buffers)
- GLuint glGenLists(GLsizei range)
- void glGenQueries(GLsizei n, GLuint * ids)
- void glGenTextures(GLsizei n, GLuint * textures)
- void glGetBooleanv(GLenum pname, GLboolean * params)
- void glGetDoublev(GLenum pname, GLdouble * params)
- void glGetFloatv(GLenum pname, GLfloat * params)
- void glGetIntegerv(GLenum pname, GLint * params)
- void glGetActiveAttrib(GLuint program, GLuint index, GLsizei bufSize, GLsizei *length, GLint *size, GLenum *type, GLchar *name)
- void glGetActiveUniform(GLuint program, GLuint index, GLsizei bufSize, GLsizei *length, GLint *size, GLenum *type, GLchar *name)
- void glGetAttachedShaders(GLuint program, GLsizei maxCount, GLsizei *count, GLuint *shaders)
- GLint glGetAttribLocation(GLuint program, const GLchar *name)
- void glGetBufferParameteriv(GLenum target, GLenum value, GLint * data)
- void glGetBufferPointerv(GLenum target, GLenum pname, GLvoid ** params)
- void glGetBufferSubData(GLenum target, GLintptr offset, GLsizeiptr size, GLvoid * data)
- void glGetClipPlane(GLenum plane, GLdouble * equation)
- void glGetColorTable(GLenum target, GLenum format, GLenum type, GLvoid * table)
- void glGetColorTableParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetColorTableParameteriv(GLenum target, GLenum pname, GLint * params)

- void glGetCompressedTexImage(GLenum target, GLint lod, GLvoid * img)
- void glGetConvolutionFilter(GLenum target, GLenum format, GLenum type, GLvoid * image)
- void glGetConvolutionParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetConvolutionParameteriv(GLenum target, GLenum pname, GLint * params)
- GLenum glGetError(void)
- void glGetHistogram(GLenum target, GLboolean reset, GLenum format, GLenum type, GLvoid * values)
- void glGetHistogramParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetHistogramParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetLightfv(GLenum light, GLenum pname, GLfloat * params)
- void glGetLightiv(GLenum light, GLenum pname, GLint * params)
- void glGetMapdv(GLenum target, GLenum query, GLdouble * v)
- void glGetMapfv(GLenum target, GLenum query, GLfloat * v)
- void glGetMapiv(GLenum target, GLenum query, GLint * v)
- void glGetMaterialfv(GLenum face, GLenum pname, GLfloat * params)
- void glGetMaterialiv(GLenum face, GLenum pname, GLint * params)
- void glGetMinmax(GLenum target, GLboolean reset, GLenum format, GLenum types, GLvoid * values)
- void glGetMinmaxParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetMinmaxParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetPixelMapfv(GLenum map, GLfloat * data)
- void glGetPixelMapuiv(GLenum map, GLuint * data)
- void glGetPixelMapusv(GLenum map, GLushort * data)
- void glGetPointerv(GLenum pname, GLvoid ** params)
- void glGetPolygonStipple(GLubyte * pattern)
- void glGetProgramiv(GLuint program, GLenum pname, GLint *params)
- void glGetProgramInfoLog(GLuint program, GLsizei maxLength, GLsizei *length, GLchar *infoLog)
- void glGetQueryObjectiv(GLuint id, GLenum pname, GLint * params)
- void glGetQueryObjectuiv(GLuint id, GLenum pname, GLuint * params)
- void glGetQueryiv(GLenum target, GLenum pname, GLint * params)
- void glGetSeparableFilter(GLenum target, GLenum format, GLenum type, GLvoid * row, GLvoid * column, GLvoid * span)
- void glGetShaderiv(GLuint shader, GLenum pname, GLint *params)
- void glGetShaderInfoLog(GLuint shader, GLsizei maxLength, GLsizei *length, GLchar *infoLog)
- void glGetShaderSource(GLuint shader, GLsizei bufSize, GLsizei *length, GLchar *source)
- const GLubyte* getString(GLenum name)
- void glGetTexEnvfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetTexEnviv(GLenum target, GLenum pname, GLint * params)

- void glGetTexGendv(GLenum coord, GLenum pname, GLdouble * params)
- void glGetTexGenfv(GLenum coord, GLenum pname, GLfloat * params)
- void glGetTexGeniv(GLenum coord, GLenum pname, GLint * params)
- void glGetTexImage(GLenum target, GLint level, GLenum format, GLenum type, GLvoid * img)
- void glGetTexLevelParameterfv(GLenum target, GLint level, GLenum pname, GLfloat * params)
- void glGetTexLevelParameteriv(GLenum target, GLint level, GLenum pname, GLint * params)
- void glGetTexParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetTexParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetUniformfv(GLuint program, GLint location, GLfloat *params)
- void glGetUniformiv(GLuint program, GLint location, GLint *params)
- GLint glGetUniformLocation(GLuint program, const GLchar *name)
- void glGetVertexAttribdv(GLuint index, GLenum pname, GLdouble *params)
- void glGetVertexAttribfv(GLuint index, GLenum pname, GLfloat *params)
- void glGetVertexAttribiv(GLuint index, GLenum pname, GLint *params)
- void glGetVertexAttribPointerv(GLuint index, GLenum pname, GLvoid **pointer)
- void glHint(GLenum target, GLenum mode)
- void glHistogram(GLenum target, GLsizei width, GLenum internalformat, GLboolean sink)
- void glIndexs(GLshort c)
- void glIndexi(GLint c)
- void glIndexf(GLfloat c)
- void glIndexd(GLdouble c)
- void glIndexub(GLubyte c)
- void glIndexsv(const GLshort * c)
- void glIndexiv(const GLint * c)
- void glIndexfv(const GLfloat * c)
- void glIndexdv(const GLdouble * c)
- void glIndexubv(const GLubyte * c)
- void glIndexMask(GLuint mask)
- void glIndexPointer(GLenum type, GLsizei stride, const GLvoid * pointer)
- void glInitNames(void)
- void glInterleavedArrays(GLenum format, GLsizei stride, const GLvoid * pointer)
- GLboolean glIsBuffer(GLuint buffer)
- GLboolean glIsEnabled(GLenum cap)
- GLboolean glIsList(GLuint list)
- GLboolean glIsProgram(GLuint program)
- GLboolean glIsQuery(GLuint id)

- GLboolean glIsShader(GLuint shader)
- GLboolean glIsTexture(GLuint texture)
- void glLightf(GLenum light, GLenum pname, GLfloat param)
- void glLighti(GLenum light, GLenum pname, GLint param)
- void glLightfv(GLenum light, GLenum pname, const GLfloat * params)
- void glLightiv(GLenum light, GLenum pname, const GLint * params)
- void glLightModelf(GLenum pname, GLfloat param)
- void glLightModeli(GLenum pname, GLint param)
- void glLightModelfv(GLenum pname, const GLfloat * params)
- void glLightModeliv(GLenum pname, const GLint * params)
- void glLineStipple(GLint factor, GLushort pattern)
- void glLineWidth(GLfloat width)
- void glLinkProgram(GLuint program)
- void glListBase(GLuint base)
- void glLoadIdentity(void)
- void glLoadMatrixd(const GLdouble * m)
- void glLoadMatrixf(const GLfloat * m)
- void glLoadName(GLuint name)
- void glLoadTransposeMatrixd(const GLdouble * m)
- void glLoadTransposeMatrixf(const GLfloat * m)
- void glLogicOp(GLenum opcode)
- void glMap1f(GLenum target, GLfloat u1, GLfloat u2, GLint stride, GLint order, const GLfloat * points)
- void glMap1d(GLenum target, GLdouble u1, GLdouble u2, GLint stride, GLint order, const GLdouble * points)
- void glMap2f(GLenum target, GLfloat u1, GLfloat u2, GLint ustride, GLint uorder, GLfloat v1, GLfloat v2, GLint vstride, GLint vorder, const GLfloat * points)
- void glMap2d(GLenum target, GLdouble u1, GLdouble u2, GLint ustride, GLint uorder, GLdouble v1, GLdouble v2, GLint vstride, GLint vorder, const GLdouble * points)
- void * glMapBuffer(GLenum target, GLenum access)
- void glMapGrid1d(GLint un, GLdouble u1, GLdouble u2)
- void glMapGrid1f(GLint un, GLfloat u1, GLfloat u2)
- void glMapGrid2d(GLint un, GLdouble u1, GLdouble u2, GLint vn, GLdouble v1, GLdouble v2)
- void glMapGrid2f(GLint un, GLfloat u1, GLfloat u2, GLint vn, GLfloat v1, GLfloat v2)
- void glMaterialf(GLenum face, GLenum pname, GLfloat param)
- void glMateriali(GLenum face, GLenum pname, GLint param)
- void glMatrixMode(GLenum mode)
- void glMinmax(GLenum target, GLenum internalformat, GLboolean sink)
- void glMultMatrixd(const GLdouble * m)

- `void glMultMatrixf(const GLfloat * m)`
- `void glMultTransposeMatrixd(const GLdouble * m)`
- `void glMultTransposeMatrixf(const GLfloat * m)`
- `void glMultiDrawArrays(GLenum mode, GLint * first, GLsizei * count, GLsizei primcount)`
- `void glMultiDrawElements(GLenum mode, const GLsizei * count, GLenum type, const GLvoid ** indices, GLsizei primcount)`
- `void glMultiTexCoord1s(GLenum target, GLshort s)`
- `void glMultiTexCoord1i(GLenum target, GLint s)`
- `void glMultiTexCoord1f(GLenum target, GLfloat s)`
- `void glMultiTexCoord1d(GLenum target, GLdouble s)`
- `void glMultiTexCoord2s(GLenum target, GLshort s, GLshort t)`
- `void glMultiTexCoord2i(GLenum target, GLint s, GLint t)`
- `void glMultiTexCoord2f(GLenum target, GLfloat s, GLfloat t)`
- `void glMultiTexCoord2d(GLenum target, GLdouble s, GLdouble t)`
- `void glMultiTexCoord3s(GLenum target, GLshort s, GLshort t, GLshort r)`
- `void glMultiTexCoord3i(GLenum target, GLint s, GLint t, GLint r)`
- `void glMultiTexCoord3f(GLenum target, GLfloat s, GLfloat t, GLfloat r)`
- `void glMultiTexCoord3d(GLenum target, GLdouble s, GLdouble t, GLdouble r)`
- `void glMultiTexCoord4s(GLenum target, GLshort s, GLshort t, GLshort r, GLshort q)`
- `void glMultiTexCoord4i(GLenum target, GLint s, GLint t, GLint r, GLint q)`
- `void glMultiTexCoord4f(GLenum target, GLfloat s, GLfloat t, GLfloat r, GLfloat q)`
- `void glMultiTexCoord4d(GLenum target, GLdouble s, GLdouble t, GLdouble r, GLdouble q)`
- `void glMultiTexCoord1sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord1iv(GLenum target, const GLint * v)`
- `void glMultiTexCoord1fv(GLenum target, const GLfloat * v)`
- `void glMultiTexCoord1dv(GLenum target, const GLdouble * v)`
- `void glMultiTexCoord2sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord2iv(GLenum target, const GLint * v)`
- `void glMultiTexCoord2fv(GLenum target, const GLfloat * v)`
- `void glMultiTexCoord2dv(GLenum target, const GLdouble * v)`
- `void glMultiTexCoord3sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord3iv(GLenum target, const GLint * v)`
- `void glMultiTexCoord3fv(GLenum target, const GLfloat * v)`
- `void glMultiTexCoord3dv(GLenum target, const GLdouble * v)`
- `void glMultiTexCoord4sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord4iv(GLenum target, const GLint * v)`

- void glMultiTexCoord4fv(GLenum target,const GLfloat * v)
- void glMultiTexCoord4dv(GLenum target,const GLdouble * v)
- void glNewList(GLuint list,GLenum mode)
- void glNormal3b(GLbyte nx,GLbyte ny,GLbyte nz)
- void glNormal3d(GLdouble nx,GLdouble ny,GLdouble nz)
- void glNormal3f(GLfloat nx,GLfloat ny,GLfloat nz)
- void glNormal3i(GLint nx,GLint ny,GLint nz)
- void glNormal3s(GLshort nx,GLshort ny,GLshort nz)
- void glNormal3bv(const GLbyte * v)
- void glNormal3dv(const GLdouble * v)
- void glNormal3fv(const GLfloat * v)
- void glNormal3iv(const GLint * v)
- void glNormal3sv(const GLshort * v)
- void glNormalPointer(GLenum type,GLsizei stride,const GLvoid * pointer)
- void glOrtho(GLdouble left,GLdouble right,GLdouble bottom,GLdouble top,GLdouble nearVal,GLdouble farVal)
- void glPassThrough(GLfloat token)
- void glPixelMapfv(GLenum map,GLsizei mapsize,const GLfloat * values)
- void glPixelMapuiv(GLenum map,GLsizei mapsize,const GLuint * values)
- void glPixelMapusv(GLenum map,GLsizei mapsize,const GLushort * values)
- void glPixelStoref(GLenum pname,GLfloat param)
- void glPixelStorei(GLenum pname,GLint param)
- void glPixelTransferf(GLenum pname,GLfloat param)
- void glPixelTransferi(GLenum pname,GLint param)
- void glPixelZoom(GLfloat xfactor,GLfloat yfactor)
- void glPointParameterf(GLenum pname,GLfloat param)
- void glPointParameteri(GLenum pname,GLint param)
- void glPointSize(GLfloat size)
- void glPolygonMode(GLenum face,GLenum mode)
- void glPolygonOffset(GLfloat factor,GLfloat units)
- void glPolygonStipple(const GLubyte * pattern)
- void glPushAttrib(GLbitfield mask)
- void glPushClientAttrib(GLbitfield mask)
- void glPushMatrix(void)
- void glPushName(GLuint name)
- void glPrioritizeTextures(GLsizei n,const GLuint * textures,const GLclampf * priorities)

- `void glPopMatrix(void)`
- `void glRasterPos2s(GLshort x, GLshort y)`
- `void glRasterPos2i(GLint x, GLint y)`
- `void glRasterPos2f(GLfloat x, GLfloat y)`
- `void glRasterPos2d(GLdouble x, GLdouble y)`
- `void glRasterPos3s(GLshort x, GLshort y, GLshort z)`
- `void glRasterPos3i(GLint x, GLint y, GLint z)`
- `void glRasterPos3f(GLfloat x, GLfloat y, GLfloat z)`
- `void glRasterPos3d(GLdouble x, GLdouble y, GLdouble z)`
- `void glRasterPos4s(GLshort x, GLshort y, GLshort z, GLshort w)`
- `void glRasterPos4i(GLint x, GLint y, GLint z, GLint w)`
- `void glRasterPos4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)`
- `void glRasterPos4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)`
- `void glReadBuffer(GLenum mode)`
- `void glReadPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum format, GLenum type, GLvoid * data)`
- `void glRectd(GLdouble x1, GLdouble y1, GLdouble x2, GLdouble y2)`
- `void glRectf(GLfloat x1, GLfloat y1, GLfloat x2, GLfloat y2)`
- `void glRecti(GLint x1, GLint y1, GLint x2, GLint y2)`
- `void glRects(GLshort x1, GLshort y1, GLshort x2, GLshort y2)`
- `void glRectdv(const GLdouble * v1, const GLdouble * v2)`
- `void glRectfv(const GLfloat * v1, const GLfloat * v2)`
- `void glRectiv(const GLint * v1, const GLint * v2)`
- `void glRectsv(const GLshort * v1, const GLshort * v2)`
- `GLint glRenderMode(GLenum mode)`
- `void glResetHistogram(GLenum target)`
- `void glRotated(GLdouble angle, GLdouble x, GLdouble y, GLdouble z)`
- `void glRotatef(GLfloat angle, GLfloat x, GLfloat y, GLfloat z)`
- `void glSampleCoverage(GLclampf value, GLboolean invert)`
- `void glScaled(GLdouble x, GLdouble y, GLdouble z)`
- `void glScalef(GLfloat x, GLfloat y, GLfloat z)`
- `void glScissor(GLint x, GLint y, GLsizei width, GLsizei height)`
- `void glSecondaryColor3b(GLbyte red, GLbyte green, GLbyte blue)`
- `void glSecondaryColor3s(GLshort red, GLshort green, GLshort blue)`
- `void glSecondaryColor3i(GLint red, GLint green, GLint blue)`
- `void glSecondaryColor3f(GLfloat red, GLfloat green, GLfloat blue)`
- `void glSecondaryColor3d(GLdouble red, GLdouble green, GLdouble blue)`

- void glSecondaryColor3ub(GLubyte red,GLubyte green,GLubyte blue)
- void glSecondaryColor3us(GLushort red,GLushort green,GLushort blue)
- void glSecondaryColor3ui(GLuint red,GLuint green,GLuint blue)
- void glSecondaryColor3bv(const GLbyte * v)
- void glSecondaryColor3sv(const GLshort * v)
- void glSecondaryColor3iv(const GLint * v)
- void glSecondaryColor3fv(const GLfloat * v)
- void glSecondaryColor3dv(const GLdouble * v)
- void glSecondaryColor3ubv(const GLubyte * v)
- void glSecondaryColor3usv(const GLushort * v)
- void glSecondaryColor3uiv(const GLuint * v)
- void glSecondaryColorPointer(GLint size,GLenum type,GLsizei stride,const GLvoid * pointer)
- void glSelectBuffer(GLsizei size,GLuint * buffer)
- void glSeparableFilter2D(GLenum target,GLenum internalformat,GLsizei width,GLsizei height,GLenum format,GLenum type,const GLvoid * row,const GLvoid * column)
- void glShadeModel(GLenum mode)
- void glShaderSource(GLuint shader,GLsizei count,const GLchar **string,const GLint *length)
- void glStencilFunc(GLenum func,GLint ref,GLuint mask)
- void glStencilFuncSeparate(GLenum face,GLenum func,GLint ref,GLuint mask)
- void glStencilMask(GLuint mask)
- void glStencilMaskSeparate(GLenum face,GLuint mask)
- void glStencilOp(GLenum sfail,GLenum dpfail,GLenum dppass)
- void glStencilOpSeparate(GLenum face,GLenum sfail,GLenum dpfail,GLenum dppass)
- void glTexCoord1s(GLshort s)
- void glTexCoord1i(GLint s)
- void glTexCoord1f(GLfloat s)
- void glTexCoord1d(GLdouble s)
- void glTexCoord2s(GLshort s,GLshort t)
- void glTexCoord2i(GLint s,GLint t)
- void glTexCoord2f(GLfloat s,GLfloat t)
- void glTexCoord2d(GLdouble s,GLdouble t)
- void glTexCoord3s(GLshort s,GLshort t,GLshort r)
- void glTexCoord3i(GLint s,GLint t,GLint r)
- void glTexCoord3f(GLfloat s,GLfloat t,GLfloat r)
- void glTexCoord3d(GLdouble s,GLdouble t,GLdouble r)
- void glTexCoord4s(GLshort s,GLshort t,GLshort r,GLshort q)

- `void glTexCoord4i(GLint s,GLint t,GLint r,GLint q)`
- `void glTexCoord4f(GLfloat s,GLfloat t,GLfloat r,GLfloat q)`
- `void glTexCoord4d(GLdouble s,GLdouble t,GLdouble r,GLdouble q)`
- `void glTexCoord1sv(const GLshort * v)`
- `void glTexCoord1iv(const GLint * v)`
- `void glTexCoord1fv(const GLfloat * v)`
- `void glTexCoord1dv(const GLdouble * v)`
- `void glTexCoord2sv(const GLshort * v)`
- `void glTexCoord2iv(const GLint * v)`
- `void glTexCoord2fv(const GLfloat * v)`
- `void glTexCoord2dv(const GLdouble * v)`
- `void glTexCoord3sv(const GLshort * v)`
- `void glTexCoord3iv(const GLint * v)`
- `void glTexCoord3fv(const GLfloat * v)`
- `void glTexCoord3dv(const GLdouble * v)`
- `void glTexCoord4sv(const GLshort * v)`
- `void glTexCoord4iv(const GLint * v)`
- `void glTexCoord4fv(const GLfloat * v)`
- `void glTexCoord4dv(const GLdouble * v)`
- `void glTexCoordPointer(GLint size,GLenum type,GLsizei stride,const GLvoid * pointer)`
- `void glTexEnvf(GLenum target,GLenum pname,GLfloat param)`
- `void glTexEnvi(GLenum target,GLenum pname,GLint param)`
- `void glTexGeni(GLenum coord,GLenum pname,GLint param)`
- `void glTexGenf(GLenum coord,GLenum pname,GLfloat param)`
- `void glTexGend(GLenum coord,GLenum pname,GLdouble param)`
- `void glTexGeniv(GLenum coord,GLenum pname,const GLint * params)`
- `void glTexGenfv(GLenum coord,GLenum pname,const GLfloat * params)`
- `void glTexGendv(GLenum coord,GLenum pname,const GLdouble * params)`
- `void glTexImage1D(GLenum target,GLint level,GLint internalFormat,GLsizei width,GLint border,GLenum format,GLenum type,const GLvoid * data)`
- `void glTexImage2D(GLenum target,GLint level,GLint internalFormat,GLsizei width,GLsizei height,GLint border,GLenum format,GLenum type,const GLvoid * data)`
- `void glTexImage3D(GLenum target,GLint level,GLint internalFormat,GLsizei width,GLsizei height,GLsizei depth,GLint border,GLenum format,GLenum type,const GLvoid * data)`
- `void glTexParameterf(GLenum target,GLenum pname,GLfloat param)`
- `void glTexParameteri(GLenum target,GLenum pname,GLint param)`
- `void glTexParameterfv(GLenum target,GLenum pname,const GLfloat * params)`

- void glTexParameteriv(GLenum target, GLenum pname, const GLint * params)
- void glTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLenum type, const GLvoid * data)
- void glTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)
- void glTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLenum type, const GLvoid * data)
- void glTranslated(GLdouble x, GLdouble y, GLdouble z)
- void glTranslatef(GLfloat x, GLfloat y, GLfloat z)
- void glUniform1f(GLint location, GLfloat v0)
- void glUniform2f(GLint location, GLfloat v0, GLfloat v1)
- void glUniform3f(GLint location, GLfloat v0, GLfloat v1, GLfloat v2)
- void glUniform4f(GLint location, GLfloat v0, GLfloat v1, GLfloat v2, GLfloat v3)
- void glUniform1i(GLint location, GLint v0)
- void glUniform2i(GLint location, GLint v0, GLint v1)
- void glUniform3i(GLint location, GLint v0, GLint v1, GLint v2)
- void glUniform4i(GLint location, GLint v0, GLint v1, GLint v2, GLint v3)
- void glUniform1fv(GLint location, GLsizei count, const GLfloat *value)
- void glUniform2fv(GLint location, GLsizei count, const GLfloat *value)
- void glUniform3fv(GLint location, GLsizei count, const GLfloat *value)
- void glUniform4fv(GLint location, GLsizei count, const GLfloat *value)
- void glUniform1iv(GLint location, GLsizei count, const GLint *value)
- void glUniform2iv(GLint location, GLsizei count, const GLint *value)
- void glUniform3iv(GLint location, GLsizei count, const GLint *value)
- void glUniform4iv(GLint location, GLsizei count, const GLint *value)
- void glUniformMatrix2fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)
- void glUniformMatrix3fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)
- void glUniformMatrix4fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)
- void glUniformMatrix2x3fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)
- void glUniformMatrix3x2fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)
- void glUniformMatrix2x4fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)
- void glUniformMatrix4x2fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)
- void glUniformMatrix3x4fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)
- void glUniformMatrix4x3fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)
- void glUseProgram(GLuint program)
- void glValidateProgram(GLuint program)
- void glVertex2s(GLshort x, GLshort y)

- `void glVertex2i(GLint x, GLint y)`
- `void glVertex2f(GLfloat x, GLfloat y)`
- `void glVertex2d(GLdouble x, GLdouble y)`
- `void glVertex3s(GLshort x, GLshort y, GLshort z)`
- `void glVertex3i(GLint x, GLint y, GLint z)`
- `void glVertex3f(GLfloat x, GLfloat y, GLfloat z)`
- `void glVertex3d(GLdouble x, GLdouble y, GLdouble z)`
- `void glVertex4s(GLshort x, GLshort y, GLshort z, GLshort w)`
- `void glVertex4i(GLint x, GLint y, GLint z, GLint w)`
- `void glVertex4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)`
- `void glVertex4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)`
- `void glVertex2sv(const GLshort * v)`
- `void glVertex2iv(const GLint * v)`
- `void glVertex2fv(const GLfloat * v)`
- `void glVertex2dv(const GLdouble * v)`
- `void glVertex3sv(const GLshort * v)`
- `void glVertex3iv(const GLint * v)`
- `void glVertex3fv(const GLfloat * v)`
- `void glVertex3dv(const GLdouble * v)`
- `void glVertex4sv(const GLshort * v)`
- `void glVertex4iv(const GLint * v)`
- `void glVertex4fv(const GLfloat * v)`
- `void glVertex4dv(const GLdouble * v)`
- `void glVertexAttrib1f(GLuint index, GLfloat v0)`
- `void glVertexAttrib1s(GLuint index, GLshort v0)`
- `void glVertexAttrib1d(GLuint index, GLdouble v0)`
- `void glVertexAttrib2f(GLuint index, GLfloat v0, GLfloat v1)`
- `void glVertexAttrib2s(GLuint index, GLshort v0, GLshort v1)`
- `void glVertexAttrib2d(GLuint index, GLdouble v0, GLdouble v1)`
- `void glVertexAttrib3f(GLuint index, GLfloat v0, GLfloat v1, GLfloat v2)`
- `void glVertexAttrib3s(GLuint index, GLshort v0, GLshort v1, GLshort v2)`
- `void glVertexAttrib3d(GLuint index, GLdouble v0, GLdouble v1, GLdouble v2)`
- `void glVertexAttrib4f(GLuint index, GLfloat v0, GLfloat v1, GLfloat v2, GLfloat v3)`
- `void glVertexAttrib4s(GLuint index, GLshort v0, GLshort v1, GLshort v2, GLshort v3)`
- `void glVertexAttrib4d(GLuint index, GLdouble v0, GLdouble v1, GLdouble v2, GLdouble v3)`
- `void glVertexAttrib4Nub(GLuint index, GLubyte v0, GLubyte v1, GLubyte v2, GLubyte v3)`

- void glVertexAttrib1fv(GLuint index,const GLfloat *v)
- void glVertexAttrib1sv(GLuint index,const GLshort *v)
- void glVertexAttrib1dv(GLuint index,const GLdouble *v)
- void glVertexAttrib2fv(GLuint index,const GLfloat *v)
- void glVertexAttrib2sv(GLuint index,const GLshort *v)
- void glVertexAttrib2dv(GLuint index,const GLdouble *v)
- void glVertexAttrib3fv(GLuint index,const GLfloat *v)
- void glVertexAttrib3sv(GLuint index,const GLshort *v)
- void glVertexAttrib3dv(GLuint index,const GLdouble *v)
- void glVertexAttrib4fv(GLuint index,const GLfloat *v)
- void glVertexAttrib4sv(GLuint index,const GLshort *v)
- void glVertexAttrib4dv(GLuint index,const GLdouble *v)
- void glVertexAttrib4iv(GLuint index,const GLint *v)
- void glVertexAttrib4bv(GLuint index,const GLbyte *v)
- void glVertexAttrib4ubv(GLuint index,const GLubyte *v)
- void glVertexAttrib4usv(GLuint index,const GLushort *v)
- void glVertexAttrib4uiv(GLuint index,const GLuint *v)
- void glVertexAttribPointer(GLuint index,GLint size,GLenum type,GLboolean normalized,GLsizei stride,const GLvoid * pointer)
- void glVertexPointer(GLint size,GLenum type,GLsizei stride,const GLvoid * pointer)
- void glViewport(GLint x,GLint y,GLsizei width,GLsizei height)
- void glWindowPos2s(GLshort x,GLshort y)
- void glWindowPos2i(GLint x,GLint y)
- void glWindowPos2f(GLfloat x,GLfloat y)
- void glWindowPos2d(GLdouble x,GLdouble y)
- void glWindowPos3s(GLshort x,GLshort y,GLshort z)
- void glWindowPos3i(GLint x,GLint y,GLint z)
- void glWindowPos3f(GLfloat x,GLfloat y,GLfloat z)
- void glWindowPos3d(GLdouble x,GLdouble y,GLdouble z)
- void glWindowPos2sv(const GLshort * v)
- void glWindowPos2iv(const GLint * v)
- void glWindowPos2fv(const GLfloat * v)
- void glWindowPos2dv(const GLdouble * v)
- void glWindowPos3sv(const GLshort * v)
- void glWindowPos3iv(const GLint * v)
- void glWindowPos3fv(const GLfloat * v)

- void glWindowPos3dv(const GLdouble * v)
- void gluBeginCurve(GLUnurbs* nurb)
- void gluBeginPolygon(GLUtesselator* tess)
- void gluBeginSurface(GLUnurbs* nurb)
- void gluBeginTrim(GLUnurbs* nurb)
- void gluCylinder(GLUquadric* quad, GLdouble base, GLdouble top, GLdouble height, GLint slices, GLint stacks)
- void gluDeleteNurbsRenderer(GLUnurbs* nurb)
- void gluDeleteQuadric(GLUquadric* quad)
- void gluDeleteTess(GLUtesselator* tess)
- void gluDisk(GLUquadric* quad, GLdouble inner, GLdouble outer, GLint slices, GLint loops)
- void gluEndCurve(GLUnurbs* nurb)
- void gluEndPolygon(GLUtesselator* tess)
- void gluEndSurface(GLUnurbs* nurb)
- void gluEndTrim(GLUnurbs* nurb)
- const GLubyte * gluErrorString(GLenum error)
- void gluGetNurbsProperty(GLUnurbs* nurb, GLenum property, GLfloat* data)
- const GLubyte * gluGetString(GLenum name)
- void gluGetTessProperty(GLUtesselator* tess, GLenum which, GLdouble* data)
- void gluLoadSamplingMatrices(GLUnurbs* nurb, const GLfloat * model, const GLfloat * perspective, const GLint * view)
- void gluLookAt(GLdouble eyeX, GLdouble eyeY, GLdouble eyeZ, GLdouble centerX, GLdouble centerY, GLdouble centerZ, GLdouble upX, GLdouble upY, GLdouble upZ)
- GLUnurbs *gluNewNurbsRenderer(void)
- GLUquadric *gluNewQuadric(void)
- GLUtesselator* gluNewTess(void)
- void gluNextContour(GLUtesselator* tess, GLenum type)
- void gluNurbsCurve(GLUnurbs* nurb, GLint knotCount, GLfloat * knots, GLint stride, GLfloat * control, GLint order, GLenum type)
- void gluNurbsProperty(GLUnurbs* nurb, GLenum property, GLfloat value)
- void gluNurbsSurface(GLUnurbs* nurb, GLint sKnotCount, GLfloat* sKnots, GLint tKnotCount, GLfloat* tKnots, GLint sStride, GLint tStride, GLfloat* control, GLint sOrder, GLint tOrder, GLenum type)
- void gluOrtho2D(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top)
- void gluPartialDisk(GLUquadric* quad, GLdouble inner, GLdouble outer, GLint slices, GLint loops, GLdouble start, GLdouble sweep)
- void gluPerspective(GLdouble fovy, GLdouble aspect, GLdouble zNear, GLdouble zFar)
- void gluPickMatrix(GLdouble x, GLdouble y, GLdouble delX, GLdouble delY, GLint * viewport)
- GLint gluProject(GLdouble objX, GLdouble objY, GLdouble objZ, const GLdouble * model, const GLdouble * proj, const GLint * view, GLdouble* winX, GLdouble* winY, GLdouble* winZ)

- void gluPwlCurve(GLUnurbs* nurb,GLint count,GLfloat* data,GLint stride,GLenum type)
- void gluQuadricDrawStyle(GLUquadric* quad,GLenum draw)
- void gluQuadricNormals(GLUquadric* quad,GLenum normal)
- void gluQuadricOrientation(GLUquadric* quad,GLenum orientation)
- void gluQuadricTexture(GLUquadric* quad,GLboolean texture)
- GLint gluScaleImage(GLenum format,GLsizei wIn,GLsizei hIn,GLenum typeIn,const void * dataIn,GLsizei wOut,GLsizei hOut,GLenum typeOut,GLvoid* dataOut)
- void gluSphere(GLUquadric* quad,GLdouble radius,GLint slices,GLint stacks)
- void gluTessBeginContour(GLUtesselator* tess)
- void gluTessBeginPolygon(GLUtesselator* tess,GLvoid* data)
- void gluTessEndContour(GLUtesselator* tess)
- void gluTessEndPolygon(GLUtesselator* tess)
- void gluTessNormal(GLUtesselator* tess,GLdouble valueX,GLdouble valueY,GLdouble valueZ)
- void gluTessProperty(GLUtesselator* tess,GLenum which,GLdouble data)
- void gluTessVertex(GLUtesselator* tess,GLdouble * location,GLvoid* data)
- GLint gluUnProject(GLdouble winX,GLdouble winY,GLdouble winZ,const GLdouble * model,const GLdouble * proj,const GLint * view,GLdouble* objX,GLdouble* objY,GLdouble* objZ)
- void glDisable(GLenum cap)

RINGOPENGL (OPENGL 4.1) FUNCTIONS REFERENCE

- GL_ZERO
- GL_FALSE
- GL_LOGIC_OP
- GL_NONE
- GL_TEXTURE_COMPONENTS
- GL_NO_ERROR
- GL_POINTS
- GL_CURRENT_BIT
- GL_TRUE
- GL_ONE
- GL_CLIENT_PIXEL_STORE_BIT
- GL_LINES
- GL_LINE_LOOP
- GL_POINT_BIT
- GL_CLIENT_VERTEX_ARRAY_BIT
- GL_LINE_STRIP
- GL_LINE_BIT
- GL_TRIANGLES
- GL_TRIANGLE_STRIP
- GL_TRIANGLE_FAN
- GL_QUADS
- GL_QUAD_STRIP
- GL_POLYGON_BIT
- GL_POLYGON
- GL_POLYGON_STIPPLE_BIT
- GL_PIXEL_MODE_BIT
- GL_LIGHTING_BIT

- GL_FOG_BIT
- GL_DEPTH_BUFFER_BIT
- GL_ACCUM
- GL_LOAD
- GL_RETURN
- GL_MULT
- GL_ADD
- GL_NEVER
- GL_ACCUM_BUFFER_BIT
- GL_LESS
- GL_EQUAL
- GL_LEQUAL
- GL_GREATER
- GL_NOTEQUAL
- GL_GEQUAL
- GL_ALWAYS
- GL_SRC_COLOR
- GL_ONE_MINUS_SRC_COLOR
- GL_SRC_ALPHA
- GL_ONE_MINUS_SRC_ALPHA
- GL_DST_ALPHA
- GL_ONE_MINUS_DST_ALPHA
- GL_DST_COLOR
- GL_ONE_MINUS_DST_COLOR
- GL_SRC_ALPHA_SATURATE
- GL_STENCIL_BUFFER_BIT
- GL_FRONT_LEFT
- GL_FRONT_RIGHT
- GL_BACK_LEFT
- GL_BACK_RIGHT
- GL_FRONT
- GL_BACK
- GL_LEFT
- GL_RIGHT
- GL_FRONT_AND_BACK
- GL_AUX0

- GL_AUX1
- GL_AUX2
- GL_AUX3
- GL_INVALID_ENUM
- GL_INVALID_VALUE
- GL_INVALID_OPERATION
- GL_STACK_OVERFLOW
- GL_STACK_UNDERFLOW
- GL_OUT_OF_MEMORY
- GL_2D
- GL_3D
- GL_3D_COLOR
- GL_3D_COLOR_TEXTURE
- GL_4D_COLOR_TEXTURE
- GL_PASS_THROUGH_TOKEN
- GL_POINT_TOKEN
- GL_LINE_TOKEN
- GL_POLYGON_TOKEN
- GL_BITMAP_TOKEN
- GL_DRAW_PIXEL_TOKEN
- GL_COPY_PIXEL_TOKEN
- GL_LINE_RESET_TOKEN
- GL_EXP
- GL_VIEWPORT_BIT
- GL_EXP2
- GL_CW
- GL_CCW
- GL_COEFF
- GL_ORDER
- GL_DOMAIN
- GL_CURRENT_COLOR
- GL_CURRENT_INDEX
- GL_CURRENT_NORMAL
- GL_CURRENT_TEXTURE_COORDS
- GL_CURRENT_RASTER_COLOR
- GL_CURRENT_RASTER_INDEX

- GL_CURRENT_RASTER_TEXTURE_COORDS
- GL_CURRENT_RASTER_POSITION
- GL_CURRENT_RASTER_POSITION_VALID
- GL_CURRENT_RASTER_DISTANCE
- GL_POINT_SMOOTH
- GL_POINT_SIZE
- GL_POINT_SIZE_RANGE
- GL_POINT_SIZE_GRANULARITY
- GL_LINE_SMOOTH
- GL_LINE_WIDTH
- GL_LINE_WIDTH_RANGE
- GL_LINE_WIDTH_GRANULARITY
- GL_LINE_STIPPLE
- GL_LINE_STIPPLE_PATTERN
- GL_LINE_STIPPLE_REPEAT
- GL_LIST_MODE
- GL_MAX_LIST_NESTING
- GL_LIST_BASE
- GL_LIST_INDEX
- GL_POLYGON_MODE
- GL_POLYGON_SMOOTH
- GL_POLYGON_STIPPLE
- GL_EDGE_FLAG
- GL_CULL_FACE
- GL_CULL_FACE_MODE
- GL_FRONT_FACE
- GL_LIGHTING
- GL_LIGHT_MODEL_LOCAL_VIEWER
- GL_LIGHT_MODEL_TWO_SIDE
- GL_LIGHT_MODEL_AMBIENT
- GL_SHADE_MODEL
- GL_COLOR_MATERIAL_FACE
- GL_COLOR_MATERIAL_PARAMETER
- GL_COLOR_MATERIAL
- GL_FOG
- GL_FOG_INDEX

- GL_FOG_DENSITY
- GL_FOG_START
- GL_FOG_END
- GL_FOG_MODE
- GL_FOG_COLOR
- GL_DEPTH_RANGE
- GL_DEPTH_TEST
- GL_DEPTH_WRITEMASK
- GL_DEPTH_CLEAR_VALUE
- GL_DEPTH_FUNC
- GL_ACCUM_CLEAR_VALUE
- GL_STENCIL_TEST
- GL_STENCIL_CLEAR_VALUE
- GL_STENCIL_FUNC
- GL_STENCIL_VALUE_MASK
- GL_STENCIL_FAIL
- GL_STENCIL_PASS_DEPTH_FAIL
- GL_STENCIL_PASS_DEPTH_PASS
- GL_STENCIL_REF
- GL_STENCIL_WRITEMASK
- GL_MATRIX_MODE
- GL_NORMALIZE
- GL_VIEWPORT
- GL_MODELVIEW_STACK_DEPTH
- GL_PROJECTION_STACK_DEPTH
- GL_TEXTURE_STACK_DEPTH
- GL_MODELVIEW_MATRIX
- GL_PROJECTION_MATRIX
- GL_TEXTURE_MATRIX
- GL_ATTRIB_STACK_DEPTH
- GL_CLIENT_ATTRIB_STACK_DEPTH
- GL_ALPHA_TEST
- GL_ALPHA_TEST_FUNC
- GL_ALPHA_TEST_REF
- GL_DITHER
- GL_BLEND_DST

- GL_BLEND_SRC
- GL_BLEND
- GL_LOGIC_OP_MODE
- GL_INDEX_LOGIC_OP
- GL_COLOR_LOGIC_OP
- GL_AUX_BUFFERS
- GL_DRAW_BUFFER
- GL_READ_BUFFER
- GL_SCISSOR_BOX
- GL_SCISSOR_TEST
- GL_INDEX_CLEAR_VALUE
- GL_INDEX_WRITEMASK
- GL_COLOR_CLEAR_VALUE
- GL_COLOR_WRITEMASK
- GL_INDEX_MODE
- GL_RGBA_MODE
- GL_DOUBLEBUFFER
- GL_STEREO
- GL_RENDER_MODE
- GL_PERSPECTIVE_CORRECTION_HINT
- GL_POINT_SMOOTH_HINT
- GL_LINE_SMOOTH_HINT
- GL_POLYGON_SMOOTH_HINT
- GL_FOG_HINT
- GL_TEXTURE_GEN_S
- GL_TEXTURE_GEN_T
- GL_TEXTURE_GEN_R
- GL_TEXTURE_GEN_Q
- GL_PIXEL_MAP_I_TO_I
- GL_PIXEL_MAP_S_TO_S
- GL_PIXEL_MAP_I_TO_R
- GL_PIXEL_MAP_I_TO_G
- GL_PIXEL_MAP_I_TO_B
- GL_PIXEL_MAP_I_TO_A
- GL_PIXEL_MAP_R_TO_R
- GL_PIXEL_MAP_G_TO_G

- GL_PIXEL_MAP_B_TO_B
- GL_PIXEL_MAP_A_TO_A
- GL_PIXEL_MAP_I_TO_I_SIZE
- GL_PIXEL_MAP_S_TO_S_SIZE
- GL_PIXEL_MAP_I_TO_R_SIZE
- GL_PIXEL_MAP_I_TO_G_SIZE
- GL_PIXEL_MAP_I_TO_B_SIZE
- GL_PIXEL_MAP_I_TO_A_SIZE
- GL_PIXEL_MAP_R_TO_R_SIZE
- GL_PIXEL_MAP_G_TO_G_SIZE
- GL_PIXEL_MAP_B_TO_B_SIZE
- GL_PIXEL_MAP_A_TO_A_SIZE
- GL_UNPACK_SWAP_BYTES
- GL_UNPACK_LSB_FIRST
- GL_UNPACK_ROW_LENGTH
- GL_UNPACK_SKIP_ROWS
- GL_UNPACK_SKIP_PIXELS
- GL_UNPACK_ALIGNMENT
- GL_PACK_SWAP_BYTES
- GL_PACK_LSB_FIRST
- GL_PACK_ROW_LENGTH
- GL_PACK_SKIP_ROWS
- GL_PACK_SKIP_PIXELS
- GL_PACK_ALIGNMENT
- GL_MAP_COLOR
- GL_MAP_STENCIL
- GL_INDEX_SHIFT
- GL_INDEX_OFFSET
- GL_RED_SCALE
- GL_RED_BIAS
- GL_ZOOM_X
- GL_ZOOM_Y
- GL_GREEN_SCALE
- GL_GREEN_BIAS
- GL_BLUE_SCALE
- GL_BLUE_BIAS

- GL_ALPHA_SCALE
- GL_ALPHA_BIAS
- GL_DEPTH_SCALE
- GL_DEPTH_BIAS
- GL_MAX_EVAL_ORDER
- GL_MAX_LIGHTS
- GL_MAX_CLIP_PLANES
- GL_MAX_TEXTURE_SIZE
- GL_MAX_PIXEL_MAP_TABLE
- GL_MAX_ATTRIB_STACK_DEPTH
- GL_MAX_MODELVIEW_STACK_DEPTH
- GL_MAX_NAME_STACK_DEPTH
- GL_MAX_PROJECTION_STACK_DEPTH
- GL_MAX_TEXTURE_STACK_DEPTH
- GL_MAX_VIEWPORT_DIMS
- GL_MAX_CLIENT_ATTRIB_STACK_DEPTH
- GL_SUBPIXEL_BITS
- GL_INDEX_BITS
- GL_RED_BITS
- GL_GREEN_BITS
- GL_BLUE_BITS
- GL_ALPHA_BITS
- GL_DEPTH_BITS
- GL_STENCIL_BITS
- GL_ACCUM_RED_BITS
- GL_ACCUM_GREEN_BITS
- GL_ACCUM_BLUE_BITS
- GL_ACCUM_ALPHA_BITS
- GL_NAME_STACK_DEPTH
- GL_AUTO_NORMAL
- GL_MAP1_COLOR_4
- GL_MAP1_INDEX
- GL_MAP1_NORMAL
- GL_MAP1_TEXTURE_COORD_1
- GL_MAP1_TEXTURE_COORD_2
- GL_MAP1_TEXTURE_COORD_3

- GL_MAP1_TEXTURE_COORD_4
- GL_MAP1_VERTEX_3
- GL_MAP1_VERTEX_4
- GL_MAP2_COLOR_4
- GL_MAP2_INDEX
- GL_MAP2_NORMAL
- GL_MAP2_TEXTURE_COORD_1
- GL_MAP2_TEXTURE_COORD_2
- GL_MAP2_TEXTURE_COORD_3
- GL_MAP2_TEXTURE_COORD_4
- GL_MAP2_VERTEX_3
- GL_MAP2_VERTEX_4
- GL_MAP1_GRID_DOMAIN
- GL_MAP1_GRID_SEGMENTS
- GL_MAP2_GRID_DOMAIN
- GL_MAP2_GRID_SEGMENTS
- GL_TEXTURE_1D
- GL_TEXTURE_2D
- GL_FEEDBACK_BUFFER_POINTER
- GL_FEEDBACK_BUFFER_SIZE
- GL_FEEDBACK_BUFFER_TYPE
- GL_SELECTION_BUFFER_POINTER
- GL_SELECTION_BUFFER_SIZE
- GL_TEXTURE_WIDTH
- GL_TRANSFORM_BIT
- GL_TEXTURE_HEIGHT
- GL_TEXTURE_INTERNAL_FORMAT
- GL_TEXTURE_BORDER_COLOR
- GL_TEXTURE_BORDER
- GL_DONT_CARE
- GL_FASTEST
- GL_NICEST
- GL_AMBIENT
- GL_DIFFUSE
- GL_SPECULAR
- GL_POSITION

- GL_SPOT_DIRECTION
- GL_SPOT_EXPONENT
- GL_SPOT_CUTOFF
- GL_CONSTANT_ATTENUATION
- GL_LINEAR_ATTENUATION
- GL_QUADRATIC_ATTENUATION
- GL_COMPILE
- GL_COMPILE_AND_EXECUTE
- GL_BYTE
- GL_UNSIGNED_BYTE
- GL_SHORT
- GL_UNSIGNED_SHORT
- GL_INT
- GL_UNSIGNED_INT
- GL_FLOAT
- GL_2_BYTES
- GL_3_BYTES
- GL_4_BYTES
- GL_DOUBLE
- GL_CLEAR
- GL_AND
- GL_AND_REVERSE
- GL_COPY
- GL_AND_INVERTED
- GL_NOOP
- GL_XOR
- GL_OR
- GL_NOR
- GL_EQUIV
- GL_INVERT
- GL_OR_REVERSE
- GL_COPY_INVERTED
- GL_OR_INVERTED
- GL_NAND
- GL_SET
- GL_EMISSION

- GL_SHININESS
- GL_AMBIENT_AND_DIFFUSE
- GL_COLOR_INDEXES
- GL_MODELVIEW
- GL_PROJECTION
- GL_TEXTURE
- GL_COLOR
- GL_DEPTH
- GL_STENCIL
- GL_COLOR_INDEX
- GL_STENCIL_INDEX
- GL_DEPTH_COMPONENT
- GL_RED
- GL_GREEN
- GL_BLUE
- GL_ALPHA
- GL_RGB
- GL_RGBA
- GL_LUMINANCE
- GL_LUMINANCE_ALPHA
- GL_BITMAP
- GL_POINT
- GL_LINE
- GL_FILL
- GL_RENDER
- GL_FEEDBACK
- GL_SELECT
- GL_FLAT
- GL_SMOOTH
- GL_KEEP
- GL_REPLACE
- GL_INCR
- GL_DECR
- GL_VENDOR
- GL_RENDERER
- GL_VERSION

- GL_EXTENSIONS
- GL_S
- GL_ENABLE_BIT
- GL_T
- GL_R
- GL_Q
- GL_MODULATE
- GL_DECAL
- GL_TEXTURE_ENV_MODE
- GL_TEXTURE_ENV_COLOR
- GL_TEXTURE_ENV
- GL_EYE_LINEAR
- GL_OBJECT_LINEAR
- GL_SPHERE_MAP
- GL_TEXTURE_GEN_MODE
- GL_OBJECT_PLANE
- GL_EYE_PLANE
- GL_NEAREST
- GL_LINEAR
- GL_NEAREST_MIPMAP_NEAREST
- GL_LINEAR_MIPMAP_NEAREST
- GL_NEAREST_MIPMAP_LINEAR
- GL_LINEAR_MIPMAP_LINEAR
- GL_TEXTURE_MAG_FILTER
- GL_TEXTURE_MIN_FILTER
- GL_TEXTURE_WRAP_S
- GL_TEXTURE_WRAP_T
- GL_CLAMP
- GL_REPEAT
- GL_POLYGON_OFFSET_UNITS
- GL_POLYGON_OFFSET_POINT
- GL_POLYGON_OFFSET_LINE
- GL_R3_G3_B2
- GL_V2F
- GL_V3F
- GL_C4UB_V2F

- GL_C4UB_V3F
- GL_C3F_V3F
- GL_N3F_V3F
- GL_C4F_N3F_V3F
- GL_T2F_V3F
- GL_T4F_V4F
- GL_T2F_C4UB_V3F
- GL_T2F_C3F_V3F
- GL_T2F_N3F_V3F
- GL_T2F_C4F_N3F_V3F
- GL_T4F_C4F_N3F_V4F
- GL_CLIP_PLANE0
- GL_CLIP_PLANE1
- GL_CLIP_PLANE2
- GL_CLIP_PLANE3
- GL_CLIP_PLANE4
- GL_CLIP_PLANE5
- GL_LIGHT0
- GL_COLOR_BUFFER_BIT
- GL_LIGHT1
- GL_LIGHT2
- GL_LIGHT3
- GL_LIGHT4
- GL_LIGHT5
- GL_LIGHT6
- GL_LIGHT7
- GL_HINT_BIT
- GL_POLYGON_OFFSET_FILL
- GL_POLYGON_OFFSET_FACTOR
- GL_ALPHA4
- GL_ALPHA8
- GL_ALPHA12
- GL_ALPHA16
- GL_LUMINANCE4
- GL_LUMINANCE8
- GL_LUMINANCE12

- GL_LUMINANCE16
- GL_LUMINANCE4_ALPHA4
- GL_LUMINANCE6_ALPHA2
- GL_LUMINANCE8_ALPHA8
- GL_LUMINANCE12_ALPHA4
- GL_LUMINANCE12_ALPHA12
- GL_LUMINANCE16_ALPHA16
- GL_INTENSITY
- GL_INTENSITY4
- GL_INTENSITY8
- GL_INTENSITY12
- GL_INTENSITY16
- GL_RGB4
- GL_RGB5
- GL_RGB8
- GL_RGB10
- GL_RGB12
- GL_RGB16
- GL_RGBA2
- GL_RGBA4
- GL_RGB5_A1
- GL_RGBA8
- GL_RGB10_A2
- GL_RGBA12
- GL_RGBA16
- GL_TEXTURE_RED_SIZE
- GL_TEXTURE_GREEN_SIZE
- GL_TEXTURE_BLUE_SIZE
- GL_TEXTURE_ALPHA_SIZE
- GL_TEXTURE_LUMINANCE_SIZE
- GL_TEXTURE_INTENSITY_SIZE
- GL_PROXY_TEXTURE_1D
- GL_PROXY_TEXTURE_2D
- GL_TEXTURE_PRIORITY
- GL_TEXTURE_RESIDENT
- GL_TEXTURE_BINDING_1D

- GL_TEXTURE_BINDING_2D
- GL_VERTEX_ARRAY
- GL_NORMAL_ARRAY
- GL_COLOR_ARRAY
- GL_INDEX_ARRAY
- GL_TEXTURE_COORD_ARRAY
- GL_EDGE_FLAG_ARRAY
- GL_VERTEX_ARRAY_SIZE
- GL_VERTEX_ARRAY_TYPE
- GL_VERTEX_ARRAY_STRIDE
- GL_NORMAL_ARRAY_TYPE
- GL_NORMAL_ARRAY_STRIDE
- GL_COLOR_ARRAY_SIZE
- GL_COLOR_ARRAY_TYPE
- GL_COLOR_ARRAY_STRIDE
- GL_INDEX_ARRAY_TYPE
- GL_INDEX_ARRAY_STRIDE
- GL_TEXTURE_COORD_ARRAY_SIZE
- GL_TEXTURE_COORD_ARRAY_TYPE
- GL_TEXTURE_COORD_ARRAY_STRIDE
- GL_EDGE_FLAG_ARRAY_STRIDE
- GL_VERTEX_ARRAY_POINTER
- GL_NORMAL_ARRAY_POINTER
- GL_COLOR_ARRAY_POINTER
- GL_INDEX_ARRAY_POINTER
- GL_TEXTURE_COORD_ARRAY_POINTER
- GL_EDGE_FLAG_ARRAY_POINTER
- GL_COLOR_INDEX1_EXT
- GL_COLOR_INDEX2_EXT
- GL_COLOR_INDEX4_EXT
- GL_COLOR_INDEX8_EXT
- GL_COLOR_INDEX12_EXT
- GL_COLOR_INDEX16_EXT
- GL_EVAL_BIT
- GL_LIST_BIT
- GL_TEXTURE_BIT

- GL_SCISSOR_BIT
- GL_ALL_ATTRIB_BITS
- GL_CLIENT_ALL_ATTRIB_BITS
- GL_SMOOTH_POINT_SIZE_RANGE
- GL_SMOOTH_POINT_SIZE_GRANULARITY
- GL_SMOOTH_LINE_WIDTH_RANGE
- GL_SMOOTH_LINE_WIDTH_GRANULARITY
- GL_UNSIGNED_BYTE_3_3_2
- GL_UNSIGNED_SHORT_4_4_4_4
- GL_UNSIGNED_SHORT_5_5_5_1
- GL_UNSIGNED_INT_8_8_8_8
- GL_UNSIGNED_INT_10_10_10_2
- GL_RESCALE_NORMAL
- GL_TEXTURE_BINDING_3D
- GL_PACK_SKIP_IMAGES
- GL_PACK_IMAGE_HEIGHT
- GL_UNPACK_SKIP_IMAGES
- GL_UNPACK_IMAGE_HEIGHT
- GL_TEXTURE_3D
- GL_PROXY_TEXTURE_3D
- GL_TEXTURE_DEPTH
- GL_TEXTURE_WRAP_R
- GL_MAX_3D_TEXTURE_SIZE
- GL_BGR
- GL_BGRA
- GL_MAX_ELEMENTS_VERTICES
- GL_MAX_ELEMENTS_INDICES
- GL_CLAMP_TO_EDGE
- GL_TEXTURE_MIN_LOD
- GL_TEXTURE_MAX_LOD
- GL_TEXTURE_BASE_LEVEL
- GL_TEXTURE_MAX_LEVEL
- GL_LIGHT_MODEL_COLOR_CONTROL
- GL_SINGLE_COLOR
- GL_SEPARATE_SPECULAR_COLOR
- GL_UNSIGNED_BYTE_2_3_3_REV

- GL_UNSIGNED_SHORT_5_6_5
- GL_UNSIGNED_SHORT_5_6_5_REV
- GL_UNSIGNED_SHORT_4_4_4_REV
- GL_UNSIGNED_SHORT_1_5_5_5_REV
- GL_UNSIGNED_INT_8_8_8_8_REV
- GL_ALIASED_POINT_SIZE_RANGE
- GL_ALIASED_LINE_WIDTH_RANGE
- GL_MULTISAMPLE
- GL_SAMPLE_ALPHA_TO_COVERAGE
- GL_SAMPLE_ALPHA_TO_ONE
- GL_SAMPLE_COVERAGE
- GL_SAMPLE_BUFFERS
- GL_SAMPLES
- GL_SAMPLE_COVERAGE_VALUE
- GL_SAMPLE_COVERAGE_INVERT
- GL_CLAMP_TO_BORDER
- GL_TEXTURE0
- GL_TEXTURE1
- GL_TEXTURE2
- GL_TEXTURE3
- GL_TEXTURE4
- GL_TEXTURE5
- GL_TEXTURE6
- GL_TEXTURE7
- GL_TEXTURE8
- GL_TEXTURE9
- GL_TEXTURE10
- GL_TEXTURE11
- GL_TEXTURE12
- GL_TEXTURE13
- GL_TEXTURE14
- GL_TEXTURE15
- GL_TEXTURE16
- GL_TEXTURE17
- GL_TEXTURE18
- GL_TEXTURE19

- GL_TEXTURE20
- GL_TEXTURE21
- GL_TEXTURE22
- GL_TEXTURE23
- GL_TEXTURE24
- GL_TEXTURE25
- GL_TEXTURE26
- GL_TEXTURE27
- GL_TEXTURE28
- GL_TEXTURE29
- GL_TEXTURE30
- GL_TEXTURE31
- GL_ACTIVE_TEXTURE
- GL_CLIENT_ACTIVE_TEXTURE
- GL_MAX_TEXTURE_UNITS
- GL_TRANSPOSE_MODELVIEW_MATRIX
- GL_TRANSPOSE_PROJECTION_MATRIX
- GL_TRANSPOSE_TEXTURE_MATRIX
- GL_TRANSPOSE_COLOR_MATRIX
- GL_SUBTRACT
- GL_COMPRESSED_ALPHA
- GL_COMPRESSED_LUMINANCE
- GL_COMPRESSED_LUMINANCE_ALPHA
- GL_COMPRESSED_INTENSITY
- GL_COMPRESSED_RGB
- GL_COMPRESSED_RGBA
- GL_TEXTURE_COMPRESSION_HINT
- GL_NORMAL_MAP
- GL_REFLECTION_MAP
- GL_TEXTURE_CUBE_MAP
- GL_TEXTURE_BINDING_CUBE_MAP
- GL_TEXTURE_CUBE_MAP_POSITIVE_X
- GL_TEXTURE_CUBE_MAP_NEGATIVE_X
- GL_TEXTURE_CUBE_MAP_POSITIVE_Y
- GL_TEXTURE_CUBE_MAP_NEGATIVE_Y
- GL_TEXTURE_CUBE_MAP_POSITIVE_Z

- GL_TEXTURE_CUBE_MAP_NEGATIVE_Z
- GL_PROXY_TEXTURE_CUBE_MAP
- GL_MAX_CUBE_MAP_TEXTURE_SIZE
- GL_COMBINE
- GL_COMBINE_RGB
- GL_COMBINE_ALPHA
- GL_RGB_SCALE
- GL_ADD_SIGNED
- GL_INTERPOLATE
- GL_CONSTANT
- GL_PRIMARY_COLOR
- GL_PREVIOUS
- GL_SOURCE0_RGB
- GL_SOURCE1_RGB
- GL_SOURCE2_RGB
- GL_SOURCE0_ALPHA
- GL_SOURCE1_ALPHA
- GL_SOURCE2_ALPHA
- GL_OPERAND0_RGB
- GL_OPERAND1_RGB
- GL_OPERAND2_RGB
- GL_OPERAND0_ALPHA
- GL_OPERAND1_ALPHA
- GL_OPERAND2_ALPHA
- GL_TEXTURE_COMPRESSED_IMAGE_SIZE
- GL_TEXTURE_COMPRESSED
- GL_NUM_COMPRESSED_TEXTURE_FORMATS
- GL_COMPRESSED_TEXTURE_FORMATS
- GL_DOT3_RGB
- GL_DOT3_RGBA
- GL_MULTISAMPLE_BIT
- GL_BLEND_DST_RGB
- GL_BLEND_SRC_RGB
- GL_BLEND_DST_ALPHA
- GL_BLEND_SRC_ALPHA
- GL_POINT_SIZE_MIN

- GL_POINT_SIZE_MAX
- GL_POINT_FADE_THRESHOLD_SIZE
- GL_POINT_DISTANCE_ATTENUATION
- GL_GENERATE_MIPMAP
- GL_GENERATE_MIPMAP_HINT
- GL_DEPTH_COMPONENT16
- GL_DEPTH_COMPONENT24
- GL_DEPTH_COMPONENT32
- GL_MIRRORED_REPEAT
- GL_FOG_COORDINATE_SOURCE
- GL_FOG_COORDINATE
- GL_FRAGMENT_DEPTH
- GL_CURRENT_FOG_COORDINATE
- GL_FOG_COORDINATE_ARRAY_TYPE
- GL_FOG_COORDINATE_ARRAY_STRIDE
- GL_FOG_COORDINATE_ARRAY_POINTER
- GL_FOG_COORDINATE_ARRAY
- GL_COLOR_SUM
- GL_CURRENT_SECONDARY_COLOR
- GL_SECONDARY_COLOR_ARRAY_SIZE
- GL_SECONDARY_COLOR_ARRAY_TYPE
- GL_SECONDARY_COLOR_ARRAY_STRIDE
- GL_SECONDARY_COLOR_ARRAY_POINTER
- GL_SECONDARY_COLOR_ARRAY
- GL_MAX_TEXTURE_LOD_BIAS
- GL_TEXTURE_FILTER_CONTROL
- GL_TEXTURE_LOD_BIAS
- GL_INCR_WRAP
- GL_DECR_WRAP
- GL_TEXTURE_DEPTH_SIZE
- GL_DEPTH_TEXTURE_MODE
- GL_TEXTURE_COMPARE_MODE
- GL_TEXTURE_COMPARE_FUNC
- GL_COMPARE_R_TO_TEXTURE
- GL_CURRENT_FOG_COORD
- GL_FOG_COORD

- GL_FOG_COORD_ARRAY
- GL_FOG_COORD_ARRAY_BUFFER_BINDING
- GL_FOG_COORD_ARRAY_POINTER
- GL_FOG_COORD_ARRAY_STRIDE
- GL_FOG_COORD_ARRAY_TYPE
- GL_FOG_COORD_SRC
- GL_SRC0_ALPHA
- GL_SRC0_RGB
- GL_SRC1_ALPHA
- GL_SRC1_RGB
- GL_SRC2_ALPHA
- GL_SRC2_RGB
- GL_BUFFER_SIZE
- GL_BUFFER_USAGE
- GL_QUERY_COUNTER_BITS
- GL_CURRENT_QUERY
- GL_QUERY_RESULT
- GL_QUERY_RESULT_AVAILABLE
- GL_ARRAY_BUFFER
- GL_ELEMENT_ARRAY_BUFFER
- GL_ARRAY_BUFFER_BINDING
- GL_ELEMENT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ARRAY_BUFFER_BINDING
- GL_NORMAL_ARRAY_BUFFER_BINDING
- GL_COLOR_ARRAY_BUFFER_BINDING
- GL_INDEX_ARRAY_BUFFER_BINDING
- GL_TEXTURE_COORD_ARRAY_BUFFER_BINDING
- GL_EDGE_FLAG_ARRAY_BUFFER_BINDING
- GL_SECONDARY_COLOR_ARRAY_BUFFER_BINDING
- GL_FOG_COORDINATE_ARRAY_BUFFER_BINDING
- GL_WEIGHT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ATTRIB_ARRAY_BUFFER_BINDING
- GL_READ_ONLY
- GL_WRITE_ONLY
- GL_READ_WRITE
- GL_BUFFER_ACCESS

- GL_BUFFER_MAPPED
- GL_BUFFER_MAP_POINTER
- GL_STREAM_DRAW
- GL_STREAM_READ
- GL_STREAM_COPY
- GL_STATIC_DRAW
- GL_STATIC_READ
- GL_STATIC_COPY
- GL_DYNAMIC_DRAW
- GL_DYNAMIC_READ
- GL_DYNAMIC_COPY
- GL_SAMPLES_PASSED
- GL_BLEND_EQUATION_RGB
- GL_VERTEX_ATTRIB_ARRAY_ENABLED
- GL_VERTEX_ATTRIB_ARRAY_SIZE
- GL_VERTEX_ATTRIB_ARRAY_STRIDE
- GL_VERTEX_ATTRIB_ARRAY_TYPE
- GL_CURRENT_VERTEX_ATTRIB
- GL_VERTEX_PROGRAM_POINT_SIZE
- GL_VERTEX_PROGRAM_TWO_SIDE
- GL_VERTEX_ATTRIB_ARRAY_POINTER
- GL_STENCIL_BACK_FUNC
- GL_STENCIL_BACK_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_PASS
- GL_MAX_DRAW_BUFFERS
- GL_DRAW_BUFFER0
- GL_DRAW_BUFFER1
- GL_DRAW_BUFFER2
- GL_DRAW_BUFFER3
- GL_DRAW_BUFFER4
- GL_DRAW_BUFFER5
- GL_DRAW_BUFFER6
- GL_DRAW_BUFFER7
- GL_DRAW_BUFFER8
- GL_DRAW_BUFFER9

- GL_DRAW_BUFFER10
- GL_DRAW_BUFFER11
- GL_DRAW_BUFFER12
- GL_DRAW_BUFFER13
- GL_DRAW_BUFFER14
- GL_DRAW_BUFFER15
- GL_BLEND_EQUATION_ALPHA
- GL_POINT_SPRITE
- GL_COORD_REPLACE
- GL_MAX_VERTEX_ATTRIBS
- GL_VERTEX_ATTRIB_ARRAY_NORMALIZED
- GL_MAX_TEXTURE_COORDS
- GL_MAX_TEXTURE_IMAGE_UNITS
- GL_FRAGMENT_SHADER
- GL_VERTEX_SHADER
- GL_MAX_FRAGMENT_UNIFORM_COMPONENTS
- GL_MAX_VERTEX_UNIFORM_COMPONENTS
- GL_MAX_VARYING_FLOATS
- GL_MAX_VERTEX_TEXTURE_IMAGE_UNITS
- GL_MAX_COMBINED_TEXTURE_IMAGE_UNITS
- GL_SHADER_TYPE
- GL_FLOAT_VEC2
- GL_FLOAT_VEC3
- GL_FLOAT_VEC4
- GL_INT_VEC2
- GL_INT_VEC3
- GL_INT_VEC4
- GL_BOOL
- GL_BOOL_VEC2
- GL_BOOL_VEC3
- GL_BOOL_VEC4
- GL_FLOAT_MAT2
- GL_FLOAT_MAT3
- GL_FLOAT_MAT4
- GL_SAMPLER_1D
- GL_SAMPLER_2D

- GL_SAMPLER_3D
- GL_SAMPLER_CUBE
- GL_SAMPLER_1D_SHADOW
- GL_SAMPLER_2D_SHADOW
- GL_DELETE_STATUS
- GL_COMPILE_STATUS
- GL_LINK_STATUS
- GL_VALIDATE_STATUS
- GL_INFO_LOG_LENGTH
- GL_ATTACHED_SHADERS
- GL_ACTIVE_UNIFORMS
- GL_ACTIVE_UNIFORM_MAX_LENGTH
- GL_SHADER_SOURCE_LENGTH
- GL_ACTIVE_ATTRIBUTES
- GL_ACTIVE_ATTRIBUTE_MAX_LENGTH
- GL_FRAGMENT_SHADER_DERIVATIVE_HINT
- GL_SHADING_LANGUAGE_VERSION
- GL_CURRENT_PROGRAM
- GL_POINT_SPRITE_COORD_ORIGIN
- GL_LOWER_LEFT
- GL_UPPER_LEFT
- GL_STENCIL_BACK_REF
- GL_STENCIL_BACK_VALUE_MASK
- GL_STENCIL_BACK_WRITEMASK
- GL_CURRENT_RASTER_SECONDARY_COLOR
- GL_PIXEL_PACK_BUFFER
- GL_PIXEL_UNPACK_BUFFER
- GL_PIXEL_PACK_BUFFER_BINDING
- GL_PIXEL_UNPACK_BUFFER_BINDING
- GL_FLOAT_MAT2x3
- GL_FLOAT_MAT2x4
- GL_FLOAT_MAT3x2
- GL_FLOAT_MAT3x4
- GL_FLOAT_MAT4x2
- GL_FLOAT_MAT4x3
- GL_SRGB

- GL_SRGB8
- GL_SRGB_ALPHA
- GL_SRGB8_ALPHA8
- GL_SLUMINANCE_ALPHA
- GL_SLUMINANCE8_ALPHA8
- GL_SLUMINANCE
- GL_SLUMINANCE8
- GL_COMPRESSED_SRGB
- GL_COMPRESSED_SRGB_ALPHA
- GL_COMPRESSED_SLUMINANCE
- GL_COMPRESSED_SLUMINANCE_ALPHA
- GL_CLIP_DISTANCE0
- GL_CLIP_DISTANCE1
- GL_CLIP_DISTANCE2
- GL_CLIP_DISTANCE3
- GL_CLIP_DISTANCE4
- GL_CLIP_DISTANCE5
- GL_COMPARE_REF_TO_TEXTURE
- GL_MAX_CLIP_DISTANCES
- GL_MAX_VARYING_COMPONENTS
- GL_CONTEXT_FLAG_FORWARD_COMPATIBLE_BIT
- GL_MAJOR_VERSION
- GL_MINOR_VERSION
- GL_NUM_EXTENSIONS
- GL_CONTEXT_FLAGS
- GL_DEPTH_BUFFER
- GL_STENCIL_BUFFER
- GL_RGBA32F
- GL_RGB32F
- GL_RGBA16F
- GL_RGB16F
- GL_VERTEX_ATTRIB_ARRAY_INTEGER
- GL_MAX_ARRAY_TEXTURE_LAYERS
- GL_MIN_PROGRAM_TEXEL_OFFSET
- GL_MAX_PROGRAM_TEXEL_OFFSET
- GL_CLAMP_VERTEX_COLOR

- GL_CLAMP_FRAGMENT_COLOR
- GL_CLAMP_READ_COLOR
- GL_FIXED_ONLY
- GL_TEXTURE_RED_TYPE
- GL_TEXTURE_GREEN_TYPE
- GL_TEXTURE_BLUE_TYPE
- GL_TEXTURE_ALPHA_TYPE
- GL_TEXTURE_LUMINANCE_TYPE
- GL_TEXTURE_INTENSITY_TYPE
- GL_TEXTURE_DEPTH_TYPE
- GL_TEXTURE_1D_ARRAY
- GL_PROXY_TEXTURE_1D_ARRAY
- GL_TEXTURE_2D_ARRAY
- GL_PROXY_TEXTURE_2D_ARRAY
- GL_TEXTURE_BINDING_1D_ARRAY
- GL_TEXTURE_BINDING_2D_ARRAY
- GL_R11F_G11F_B10F
- GL_UNSIGNED_INT_10F_11F_11F_REV
- GL_RGB9_E5
- GL_UNSIGNED_INT_5_9_9_9_REV
- GL_TEXTURE_SHARED_SIZE
- GL_TRANSFORM_FEEDBACK_VARYING_MAX_LENGTH
- GL_TRANSFORM_FEEDBACK_BUFFER_MODE
- GL_MAX_TRANSFORM_FEEDBACK_SEPARATE_COMPONENTS
- GL_TRANSFORM_FEEDBACK_VARYINGS
- GL_TRANSFORM_FEEDBACK_BUFFER_START
- GL_TRANSFORM_FEEDBACK_BUFFER_SIZE
- GL_PRIMITIVES_GENERATED
- GL_TRANSFORM_FEEDBACK_PRIMITIVES_WRITTEN
- GL_RASTERIZER_DISCARD
- GL_MAX_TRANSFORM_FEEDBACK_INTERLEAVED_COMPONENTS
- GL_MAX_TRANSFORM_FEEDBACK_SEPARATE_ATTRIBS
- GL_INTERLEAVED_ATTRIBS
- GL_SEPARATE_ATTRIBS
- GL_TRANSFORM_FEEDBACK_BUFFER
- GL_TRANSFORM_FEEDBACK_BUFFER_BINDING

- GL_RGBA32UI
- GL_RGB32UI
- GL_RGBA16UI
- GL_RGB16UI
- GL_RGBA8UI
- GL_RGB8UI
- GL_RGBA32I
- GL_RGB32I
- GL_RGBA16I
- GL_RGB16I
- GL_RGBA8I
- GL_RGB8I
- GL_RED_INTEGER
- GL_GREEN_INTEGER
- GL_BLUE_INTEGER
- GL_ALPHA_INTEGER
- GL_RGB_INTEGER
- GL_RGBA_INTEGER
- GL_BGR_INTEGER
- GL_BGRA_INTEGER
- GL_SAMPLER_1D_ARRAY
- GL_SAMPLER_2D_ARRAY
- GL_SAMPLER_1D_ARRAY_SHADOW
- GL_SAMPLER_2D_ARRAY_SHADOW
- GL_SAMPLER_CUBE_SHADOW
- GL_UNSIGNED_INT_VEC2
- GL_UNSIGNED_INT_VEC3
- GL_UNSIGNED_INT_VEC4
- GL_INT_SAMPLER_1D
- GL_INT_SAMPLER_2D
- GL_INT_SAMPLER_3D
- GL_INT_SAMPLER_CUBE
- GL_INT_SAMPLER_1D_ARRAY
- GL_INT_SAMPLER_2D_ARRAY
- GL_UNSIGNED_INT_SAMPLER_1D
- GL_UNSIGNED_INT_SAMPLER_2D

- GL_UNSIGNED_INT_SAMPLER_3D
- GL_UNSIGNED_INT_SAMPLER_CUBE
- GL_UNSIGNED_INT_SAMPLER_1D_ARRAY
- GL_UNSIGNED_INT_SAMPLER_2D_ARRAY
- GL_QUERY_WAIT
- GL_QUERY_NO_WAIT
- GL_QUERY_BY_REGION_WAIT
- GL_QUERY_BY_REGION_NO_WAIT
- GL_TEXTURE_RECTANGLE
- GL_TEXTURE_BINDING_RECTANGLE
- GL_PROXY_TEXTURE_RECTANGLE
- GL_MAX_RECTANGLE_TEXTURE_SIZE
- GL_SAMPLER_2D_RECT
- GL_SAMPLER_2D_RECT_SHADOW
- GL_TEXTURE_BUFFER
- GL_MAX_TEXTURE_BUFFER_SIZE
- GL_TEXTURE_BINDING_BUFFER
- GL_TEXTURE_BUFFER_DATA_STORE_BINDING
- GL_TEXTURE_BUFFER_FORMAT
- GL_SAMPLER_BUFFER
- GL_INT_SAMPLER_2D_RECT
- GL_INT_SAMPLER_BUFFER
- GL_UNSIGNED_INT_SAMPLER_2D_RECT
- GL_UNSIGNED_INT_SAMPLER_BUFFER
- GL_RED_SNORM
- GL_RG_SNORM
- GL_RGB_SNORM
- GL_RGBA_SNORM
- GL_R8_SNORM
- GL_RG8_SNORM
- GL_RGB8_SNORM
- GL_RGBA8_SNORM
- GL_R16_SNORM
- GL_RG16_SNORM
- GL_RGB16_SNORM
- GL_RGBA16_SNORM

- GL_SIGNED_NORMALIZED
- GL_PRIMITIVE_RESTART
- GL_PRIMITIVE_RESTART_INDEX
- GL_BUFFER_ACCESS_FLAGS
- GL_BUFFER_MAP_LENGTH
- GL_BUFFER_MAP_OFFSET
- GL_CONTEXT_CORE_PROFILE_BIT
- GL_CONTEXT_COMPATIBILITY_PROFILE_BIT
- GL_LINES_ADJACENCY
- GL_LINE_STRIP_ADJACENCY
- GL_TRIANGLES_ADJACENCY
- GL_TRIANGLE_STRIP_ADJACENCY
- GL_PROGRAM_POINT_SIZE
- GL_GEOMETRY_VERTICES_OUT
- GL_GEOMETRY_INPUT_TYPE
- GL_GEOMETRY_OUTPUT_TYPE
- GL_MAX_GEOMETRY_TEXTURE_IMAGE_UNITS
- GL_FRAMEBUFFER_ATTACHMENT_LAYERED
- GL_FRAMEBUFFER_INCOMPLETE_LAYER_TARGETS
- GL_GEOMETRY_SHADER
- GL_MAX_GEOMETRY_UNIFORM_COMPONENTS
- GL_MAX_GEOMETRY_OUTPUT_VERTICES
- GL_MAX_GEOMETRY_TOTAL_OUTPUT_COMPONENTS
- GL_MAX_VERTEX_OUTPUT_COMPONENTS
- GL_MAX_GEOMETRY_INPUT_COMPONENTS
- GL_MAX_GEOMETRY_OUTPUT_COMPONENTS
- GL_MAX_FRAGMENT_INPUT_COMPONENTS
- GL_CONTEXT_PROFILE_MASK
- GL_VERTEX_ATTRIB_ARRAY_DIVISOR
- GL_RGB10_A2UI
- GL_SAMPLE_SHADING
- GL_MIN_SAMPLE_SHADING_VALUE
- GL_MIN_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_COMPONENTS
- GL_TEXTURE_CUBE_MAP_ARRAY

- GL_TEXTURE_BINDING_CUBE_MAP_ARRAY
- GL_PROXY_TEXTURE_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY_SHADOW
- GL_INT_SAMPLER_CUBE_MAP_ARRAY
- GL_UNSIGNED_INT_SAMPLER_CUBE_MAP_ARRAY
- void glAccum(GLenum op, GLfloat value)
- void glActiveTexture(GLenum texture)
- void glAlphaFunc(GLenum func, GLclampf ref)
- GLboolean glAreTexturesResident(GLsizei n, const GLuint * textures, GLboolean * residences)
- void glArrayElement(GLint i)
- void glAttachShader(GLuint program, GLuint shader)
- void glBegin(GLenum mode)
- void glBeginQuery(GLenum target, GLuint id)
- void glBindAttribLocation(GLuint program, GLuint index, const GLchar * name)
- void glBindBuffer(GLenum target, GLuint buffer)
- void glBindTexture(GLenum target, GLuint texture)
- void glBitmap(GLsizei width, GLsizei height, GLfloat xorig, GLfloat yorig, GLfloat xmove, GLfloat ymove, const GLubyte * bitmap)
- void glBlendColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)
- void glBlendEquation(GLenum mode)
- void glBlendEquationSeparate(GLenum modeRGB, GLenum modeAlpha)
- void glBlendFunc(GLenum sfactor, GLenum dfactor)
- void glBlendFuncSeparate(GLenum srcRGB, GLenum dstRGB, GLenum srcAlpha, GLenum dstAlpha)
- void glBufferData(GLenum target, GLsizeiptr size, const GLvoid * data, GLenum usage)
- void glBufferSubData(GLenum target, GLintptr offset, GLsizeiptr size, const GLvoid * data)
- void glCallList(GLuint list)
- void glCallLists(GLsizei n, GLenum type, const GLvoid * lists)
- void glClear(GLbitfield mask)
- void glClearAccum(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glClearColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)
- void glClearDepth(GLclampd depth)
- void glClearIndex(GLfloat c)
- void glClearStencil(GLint s)
- void glClientActiveTexture(GLenum texture)
- void glClipPlane(GLenum plane, const GLdouble * equation)

- `void glColor3b(GLbyte red, GLbyte green, GLbyte blue)`
- `void glColor3s(GLshort red, GLshort green, GLshort blue)`
- `void glColor3i(GLint red, GLint green, GLint blue)`
- `void glColor3f(GLfloat red, GLfloat green, GLfloat blue)`
- `void glColor3d(GLdouble red, GLdouble green, GLdouble blue)`
- `void glColor3ub(GLubyte red, GLubyte green, GLubyte blue)`
- `void glColor3us(GLushort red, GLushort green, GLushort blue)`
- `void glColor3ui(GLuint red, GLuint green, GLuint blue)`
- `void glColor4b(GLbyte red, GLbyte green, GLbyte blue, GLbyte alpha)`
- `void glColor4s(GLshort red, GLshort green, GLshort blue, GLshort alpha)`
- `void glColor4i(GLint red, GLint green, GLint blue, GLint alpha)`
- `void glColor4f(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)`
- `void glColor4d(GLdouble red, GLdouble green, GLdouble blue, GLdouble alpha)`
- `void glColor4ub(GLubyte red, GLubyte green, GLubyte blue, GLubyte alpha)`
- `void glColor4us(GLushort red, GLushort green, GLushort blue, GLushort alpha)`
- `void glColor4ui(GLuint red, GLuint green, GLuint blue, GLuint alpha)`
- `void glColor3bv(const GLbyte * v)`
- `void glColor3sv(const GLshort * v)`
- `void glColor3iv(const GLint * v)`
- `void glColor3fv(const GLfloat * v)`
- `void glColor3dv(const GLdouble * v)`
- `void glColor3ubv(const GLubyte * v)`
- `void glColor3usv(const GLushort * v)`
- `void glColor3uiv(const GLuint * v)`
- `void glColor4bv(const GLbyte * v)`
- `void glColor4sv(const GLshort * v)`
- `void glColor4iv(const GLint * v)`
- `void glColor4fv(const GLfloat * v)`
- `void glColor4dv(const GLdouble * v)`
- `void glColor4ubv(const GLubyte * v)`
- `void glColor4usv(const GLushort * v)`
- `void glColor4uiv(const GLuint * v)`
- `void glColorMask(GLboolean red, GLboolean green, GLboolean blue, GLboolean alpha)`
- `void glColorMaterial(GLenum face, GLenum mode)`
- `void glColorPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)`

- void glColorSubTable(GLenum target, GLsizei start, GLsizei count, GLenum format, GLenum type, const GLvoid * data)
- void glColorTable(GLenum target, GLenum internalformat, GLsizei width, GLenum format, GLenum type, const GLvoid * data)
- void glColorTableParameterfv(GLenum target, GLenum pname, const GLfloat * params)
- void glColorTableParameteriv(GLenum target, GLenum pname, const GLint * params)
- void glCompileShader(GLuint shader)
- void glCompressedTexImage1D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLint border, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexImage2D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLint border, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexImage3D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLsizei depth, GLint border, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLsizei imageSize, const GLvoid * data)
- void glConvolutionFilter1D(GLenum target, GLenum internalformat, GLsizei width, GLenum format, GLenum type, const GLvoid * data)
- void glConvolutionFilter2D(GLenum target, GLenum internalformat, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)
- void glConvolutionParameterf(GLenum target, GLenum pname, GLfloat params)
- void glConvolutionParameteri(GLenum target, GLenum pname, GLint params)
- void glConvolutionParameterfv(GLenum target, GLenum pname, const GLfloat * params)
- void glConvolutionParameteriv(GLenum target, GLenum pname, const GLint * params)
- void glCopyColorSubTable(GLenum target, GLsizei start, GLint x, GLint y, GLsizei width)
- void glCopyColorTable(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width)
- void glCopyConvolutionFilter1D(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width)
- void glCopyConvolutionFilter2D(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width, GLsizei height)
- void glCopyPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum type)
- void glCopyTexImage1D(GLenum target, GLint level, GLenum internalformat, GLint x, GLint y, GLsizei width, GLint border)
- void glCopyTexImage2D(GLenum target, GLint level, GLenum internalformat, GLint x, GLint y, GLsizei width, GLsizei height, GLint border)
- void glCopyTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLint x, GLint y, GLsizei width)
- void glCopyTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint x, GLint y, GLsizei width, GLsizei height)

- `void glCopyTexSubImage3D(GLenum target,GLint level,GLint xoffset,GLint yoffset,GLint zoffset,GLint x,GLint y,GLsizei width,GLsizei height)`
- `GLuint glCreateProgram(void)`
- `GLuint glCreateShader(GLenum shaderType)`
- `void glCullFace(GLenum mode)`
- `void glDeleteBuffers(GLsizei n,const GLuint * buffers)`
- `void glDeleteLists(GLuint list,GLsizei range)`
- `void glDeleteProgram(GLuint program)`
- `void glDeleteQueries(GLsizei n,const GLuint * ids)`
- `void glDeleteShader(GLuint shader)`
- `void glDeleteTextures(GLsizei n,const GLuint * textures)`
- `void glDepthFunc(GLenum func)`
- `void glDepthMask(GLboolean flag)`
- `void glDepthRange(GLclampd nearVal,GLclampd farVal)`
- `void glDetachShader(GLuint program,GLuint shader)`
- `void glEnable(GLenum cap)`
- `void glEnableClientState(GLenum cap)`
- `void glEnableVertexAttribArray(GLuint index)`
- `void glDisableVertexAttribArray(GLuint index)`
- `void glDrawArrays(GLenum mode,GLint first,GLsizei count)`
- `void glDrawBuffer(GLenum mode)`
- `void glDrawBuffers(GLsizei n,const GLenum *bufs)`
- `void glDrawElements(GLenum mode,GLsizei count,GLenum type,const GLvoid * indices)`
- `void glDrawPixels(GLsizei width,GLsizei height,GLenum format,GLenum type,const GLvoid * data)`
- `void glDrawRangeElements(GLenum mode,GLuint start,GLuint end,GLsizei count,GLenum type,const GLvoid * indices)`
- `void glEdgeFlag(GLboolean flag)`
- `void glEdgeFlagPointer(GLsizei stride,const GLvoid * pointer)`
- `void glEnd(void)`
- `void glEndList(void)`
- `void glEndQuery(GLenum target)`
- `void glEvalCoord1f(GLfloat u)`
- `void glEvalCoord1d(GLdouble u)`
- `void glEvalCoord2f(GLfloat u,GLfloat v)`
- `void glEvalCoord2d(GLdouble u,GLdouble v)`
- `void glEvalMesh1(GLenum mode,GLint i1,GLint i2)`
- `void glEvalPoint1(GLint i)`

- void glEvalPoint2(GLint i, GLint j)
- void glFeedbackBuffer(GLsizei size, GLenum type, GLfloat * buffer)
- void glFinish(void)
- void glFlush(void)
- void glFogf(GLenum pname, GLfloat param)
- void glFogi(GLenum pname, GLint param)
- void glFogfv(GLenum pname, const GLfloat * params)
- void glFogiv(GLenum pname, const GLint * params)
- void glFogCoordd(GLdouble coord)
- void glFogCoordf(GLfloat coord)
- void glFogCoorddv(GLdouble * coord)
- void glFogCoordfv(GLfloat * coord)
- void glFogCoordPointer(GLenum type, GLsizei stride, GLvoid * pointer)
- void glFrontFace(GLenum mode)
- void glFrustum(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble nearVal, GLdouble farVal)
- void glGenBuffers(GLsizei n, GLuint * buffers)
- GLuint glGenLists(GLsizei range)
- void glGenQueries(GLsizei n, GLuint * ids)
- void glGenTextures(GLsizei n, GLuint * textures)
- void glGetBooleanv(GLenum pname, GLboolean * params)
- void glGetDoublev(GLenum pname, GLdouble * params)
- void glGetFloatv(GLenum pname, GLfloat * params)
- void glGetIntegerv(GLenum pname, GLint * params)
- void glGetActiveAttrib(GLuint program, GLuint index, GLsizei bufSize, GLsizei *length, GLint *size, GLenum *type, GLchar *name)
- void glGetActiveUniform(GLuint program, GLuint index, GLsizei bufSize, GLsizei *length, GLint *size, GLenum *type, GLchar *name)
- void glGetAttachedShaders(GLuint program, GLsizei maxCount, GLsizei *count, GLuint *shaders)
- GLint glGetAttribLocation(GLuint program, const GLchar *name)
- void glGetBufferParameteriv(GLenum target, GLenum value, GLint * data)
- void glGetBufferPointerv(GLenum target, GLenum pname, GLvoid ** params)
- void glGetBufferSubData(GLenum target, GLintptr offset, GLsizeiptr size, GLvoid * data)
- void glGetClipPlane(GLenum plane, GLdouble * equation)
- void glGetColorTable(GLenum target, GLenum format, GLenum type, GLvoid * table)
- void glGetColorTableParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetColorTableParameteriv(GLenum target, GLenum pname, GLint * params)

- void glGetCompressedTexImage(GLenum target, GLint lod, GLvoid * img)
- void glGetConvolutionFilter(GLenum target, GLenum format, GLenum type, GLvoid * image)
- void glGetConvolutionParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetConvolutionParameteriv(GLenum target, GLenum pname, GLint * params)
- GLenum glGetError(void)
- void glGetHistogram(GLenum target, GLboolean reset, GLenum format, GLenum type, GLvoid * values)
- void glGetHistogramParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetHistogramParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetLightfv(GLenum light, GLenum pname, GLfloat * params)
- void glGetLightiv(GLenum light, GLenum pname, GLint * params)
- void glGetMapdv(GLenum target, GLenum query, GLdouble * v)
- void glGetMapfv(GLenum target, GLenum query, GLfloat * v)
- void glGetMapiv(GLenum target, GLenum query, GLint * v)
- void glGetMaterialfv(GLenum face, GLenum pname, GLfloat * params)
- void glGetMaterialiv(GLenum face, GLenum pname, GLint * params)
- void glGetMinmax(GLenum target, GLboolean reset, GLenum format, GLenum types, GLvoid * values)
- void glGetMinmaxParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetMinmaxParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetPixelMapfv(GLenum map, GLfloat * data)
- void glGetPixelMapuiv(GLenum map, GLuint * data)
- void glGetPixelMapusv(GLenum map, GLushort * data)
- void glGetPointerv(GLenum pname, GLvoid ** params)
- void glGetPolygonStipple(GLubyte * pattern)
- void glGetProgramiv(GLuint program, GLenum pname, GLint *params)
- void glGetProgramInfoLog(GLuint program, GLsizei maxLength, GLsizei *length, GLchar *infoLog)
- void glGetQueryObjectiv(GLuint id, GLenum pname, GLint * params)
- void glGetQueryObjectuiv(GLuint id, GLenum pname, GLuint * params)
- void glGetQueryiv(GLenum target, GLenum pname, GLint * params)
- void glGetSeparableFilter(GLenum target, GLenum format, GLenum type, GLvoid * row, GLvoid * column, GLvoid * span)
- void glGetShaderiv(GLuint shader, GLenum pname, GLint *params)
- void glGetShaderInfoLog(GLuint shader, GLsizei maxLength, GLsizei *length, GLchar *infoLog)
- void glGetShaderSource(GLuint shader, GLsizei bufSize, GLsizei *length, GLchar *source)
- const GLubyte* getString(GLenum name)
- void glGetTexEnvfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetTexEnviv(GLenum target, GLenum pname, GLint * params)

- void glGetTexGendv(GLenum coord, GLenum pname, GLdouble * params)
- void glGetTexGenfv(GLenum coord, GLenum pname, GLfloat * params)
- void glGetTexGeniv(GLenum coord, GLenum pname, GLint * params)
- void glGetTexImage(GLenum target, GLint level, GLenum format, GLenum type, GLvoid * img)
- void glGetTexLevelParameterfv(GLenum target, GLint level, GLenum pname, GLfloat * params)
- void glGetTexLevelParameteriv(GLenum target, GLint level, GLenum pname, GLint * params)
- void glGetTexParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetTexParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetUniformfv(GLuint program, GLint location, GLfloat *params)
- void glGetUniformiv(GLuint program, GLint location, GLint *params)
- GLint glGetUniformLocation(GLuint program, const GLchar *name)
- void glGetVertexAttribdv(GLuint index, GLenum pname, GLdouble *params)
- void glGetVertexAttribfv(GLuint index, GLenum pname, GLfloat *params)
- void glGetVertexAttribiv(GLuint index, GLenum pname, GLint *params)
- void glGetVertexAttribPointerv(GLuint index, GLenum pname, GLvoid **pointer)
- void glHint(GLenum target, GLenum mode)
- void glHistogram(GLenum target, GLsizei width, GLenum internalformat, GLboolean sink)
- void glIndexs(GLshort c)
- void glIndexi(GLint c)
- void glIndexf(GLfloat c)
- void glIndexd(GLdouble c)
- void glIndexub(GLubyte c)
- void glIndexsv(const GLshort * c)
- void glIndexiv(const GLint * c)
- void glIndexfv(const GLfloat * c)
- void glIndexdv(const GLdouble * c)
- void glIndexubv(const GLubyte * c)
- void glIndexMask(GLuint mask)
- void glIndexPointer(GLenum type, GLsizei stride, const GLvoid * pointer)
- void glInitNames(void)
- void glInterleavedArrays(GLenum format, GLsizei stride, const GLvoid * pointer)
- GLboolean glIsBuffer(GLuint buffer)
- GLboolean glIsEnabled(GLenum cap)
- GLboolean glIsList(GLuint list)
- GLboolean glIsProgram(GLuint program)
- GLboolean glIsQuery(GLuint id)

- GLboolean glIsShader(GLuint shader)
- GLboolean glIsTexture(GLuint texture)
- void glLightf(GLenum light, GLenum pname, GLfloat param)
- void glLighti(GLenum light, GLenum pname, GLint param)
- void glLightfv(GLenum light, GLenum pname, const GLfloat * params)
- void glLightiv(GLenum light, GLenum pname, const GLint * params)
- void glLightModelf(GLenum pname, GLfloat param)
- void glLightModeli(GLenum pname, GLint param)
- void glLightModelfv(GLenum pname, const GLfloat * params)
- void glLightModeliv(GLenum pname, const GLint * params)
- void glLineStipple(GLint factor, GLushort pattern)
- void glLineWidth(GLfloat width)
- void glLinkProgram(GLuint program)
- void glListBase(GLuint base)
- void glLoadIdentity(void)
- void glLoadMatrixd(const GLdouble * m)
- void glLoadMatrixf(const GLfloat * m)
- void glLoadName(GLuint name)
- void glLoadTransposeMatrixd(const GLdouble * m)
- void glLoadTransposeMatrixf(const GLfloat * m)
- void glLogicOp(GLenum opcode)
- void glMap1f(GLenum target, GLfloat u1, GLfloat u2, GLint stride, GLint order, const GLfloat * points)
- void glMap1d(GLenum target, GLdouble u1, GLdouble u2, GLint stride, GLint order, const GLdouble * points)
- void glMap2f(GLenum target, GLfloat u1, GLfloat u2, GLint ustride, GLint uorder, GLfloat v1, GLfloat v2, GLint vstride, GLint vorder, const GLfloat * points)
- void glMap2d(GLenum target, GLdouble u1, GLdouble u2, GLint ustride, GLint uorder, GLdouble v1, GLdouble v2, GLint vstride, GLint vorder, const GLdouble * points)
- void * glMapBuffer(GLenum target, GLenum access)
- void glMapGrid1d(GLint un, GLdouble u1, GLdouble u2)
- void glMapGrid1f(GLint un, GLfloat u1, GLfloat u2)
- void glMapGrid2d(GLint un, GLdouble u1, GLdouble u2, GLint vn, GLdouble v1, GLdouble v2)
- void glMapGrid2f(GLint un, GLfloat u1, GLfloat u2, GLint vn, GLfloat v1, GLfloat v2)
- void glMaterialf(GLenum face, GLenum pname, GLfloat param)
- void glMateriali(GLenum face, GLenum pname, GLint param)
- void glMatrixMode(GLenum mode)
- void glMinmax(GLenum target, GLenum internalformat, GLboolean sink)
- void glMultMatrixd(const GLdouble * m)

- `void glMultMatrixf(const GLfloat * m)`
- `void glMultTransposeMatrixd(const GLdouble * m)`
- `void glMultTransposeMatrixf(const GLfloat * m)`
- `void glMultiDrawArrays(GLenum mode, GLint * first, GLsizei * count, GLsizei primcount)`
- `void glMultiDrawElements(GLenum mode, const GLsizei * count, GLenum type, const GLvoid ** indices, GLsizei primcount)`
- `void glMultiTexCoord1s(GLenum target, GLshort s)`
- `void glMultiTexCoord1i(GLenum target, GLint s)`
- `void glMultiTexCoord1f(GLenum target, GLfloat s)`
- `void glMultiTexCoord1d(GLenum target, GLdouble s)`
- `void glMultiTexCoord2s(GLenum target, GLshort s, GLshort t)`
- `void glMultiTexCoord2i(GLenum target, GLint s, GLint t)`
- `void glMultiTexCoord2f(GLenum target, GLfloat s, GLfloat t)`
- `void glMultiTexCoord2d(GLenum target, GLdouble s, GLdouble t)`
- `void glMultiTexCoord3s(GLenum target, GLshort s, GLshort t, GLshort r)`
- `void glMultiTexCoord3i(GLenum target, GLint s, GLint t, GLint r)`
- `void glMultiTexCoord3f(GLenum target, GLfloat s, GLfloat t, GLfloat r)`
- `void glMultiTexCoord3d(GLenum target, GLdouble s, GLdouble t, GLdouble r)`
- `void glMultiTexCoord4s(GLenum target, GLshort s, GLshort t, GLshort r, GLshort q)`
- `void glMultiTexCoord4i(GLenum target, GLint s, GLint t, GLint r, GLint q)`
- `void glMultiTexCoord4f(GLenum target, GLfloat s, GLfloat t, GLfloat r, GLfloat q)`
- `void glMultiTexCoord4d(GLenum target, GLdouble s, GLdouble t, GLdouble r, GLdouble q)`
- `void glMultiTexCoord1sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord1iv(GLenum target, const GLint * v)`
- `void glMultiTexCoord1fv(GLenum target, const GLfloat * v)`
- `void glMultiTexCoord1dv(GLenum target, const GLdouble * v)`
- `void glMultiTexCoord2sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord2iv(GLenum target, const GLint * v)`
- `void glMultiTexCoord2fv(GLenum target, const GLfloat * v)`
- `void glMultiTexCoord2dv(GLenum target, const GLdouble * v)`
- `void glMultiTexCoord3sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord3iv(GLenum target, const GLint * v)`
- `void glMultiTexCoord3fv(GLenum target, const GLfloat * v)`
- `void glMultiTexCoord3dv(GLenum target, const GLdouble * v)`
- `void glMultiTexCoord4sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord4iv(GLenum target, const GLint * v)`

- `void glMultiTexCoord4fv(GLenum target,const GLfloat * v)`
- `void glMultiTexCoord4dv(GLenum target,const GLdouble * v)`
- `void glNewList(GLuint list,GLenum mode)`
- `void glNormal3b(GLbyte nx,GLbyte ny,GLbyte nz)`
- `void glNormal3d(GLdouble nx,GLdouble ny,GLdouble nz)`
- `void glNormal3f(GLfloat nx,GLfloat ny,GLfloat nz)`
- `void glNormal3i(GLint nx,GLint ny,GLint nz)`
- `void glNormal3s(GLshort nx,GLshort ny,GLshort nz)`
- `void glNormal3bv(const GLbyte * v)`
- `void glNormal3dv(const GLdouble * v)`
- `void glNormal3fv(const GLfloat * v)`
- `void glNormal3iv(const GLint * v)`
- `void glNormal3sv(const GLshort * v)`
- `void glNormalPointer(GLenum type,GLsizei stride,const GLvoid * pointer)`
- `void glOrtho(GLdouble left,GLdouble right,GLdouble bottom,GLdouble top,GLdouble nearVal,GLdouble farVal)`
- `void glPassThrough(GLfloat token)`
- `void glPixelMapfv(GLenum map,GLsizei mapsize,const GLfloat * values)`
- `void glPixelMapuiv(GLenum map,GLsizei mapsize,const GLuint * values)`
- `void glPixelMapusv(GLenum map,GLsizei mapsize,const GLushort * values)`
- `void glPixelStoref(GLenum pname,GLfloat param)`
- `void glPixelStorei(GLenum pname,GLint param)`
- `void glPixelTransferf(GLenum pname,GLfloat param)`
- `void glPixelTransferi(GLenum pname,GLint param)`
- `void glPixelZoom(GLfloat xfactor,GLfloat yfactor)`
- `void glPointParameterf(GLenum pname,GLfloat param)`
- `void glPointParameteri(GLenum pname,GLint param)`
- `void glPointSize(GLfloat size)`
- `void glPolygonMode(GLenum face,GLenum mode)`
- `void glPolygonOffset(GLfloat factor,GLfloat units)`
- `void glPolygonStipple(const GLubyte * pattern)`
- `void glPushAttrib(GLbitfield mask)`
- `void glPushClientAttrib(GLbitfield mask)`
- `void glPushMatrix(void)`
- `void glPushName(GLuint name)`
- `void glPrioritizeTextures(GLsizei n,const GLuint * textures,const GLclampf * priorities)`

- `void glPopMatrix(void)`
- `void glRasterPos2s(GLshort x, GLshort y)`
- `void glRasterPos2i(GLint x, GLint y)`
- `void glRasterPos2f(GLfloat x, GLfloat y)`
- `void glRasterPos2d(GLdouble x, GLdouble y)`
- `void glRasterPos3s(GLshort x, GLshort y, GLshort z)`
- `void glRasterPos3i(GLint x, GLint y, GLint z)`
- `void glRasterPos3f(GLfloat x, GLfloat y, GLfloat z)`
- `void glRasterPos3d(GLdouble x, GLdouble y, GLdouble z)`
- `void glRasterPos4s(GLshort x, GLshort y, GLshort z, GLshort w)`
- `void glRasterPos4i(GLint x, GLint y, GLint z, GLint w)`
- `void glRasterPos4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)`
- `void glRasterPos4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)`
- `void glReadBuffer(GLenum mode)`
- `void glReadPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum format, GLenum type, GLvoid * data)`
- `void glRectd(GLdouble x1, GLdouble y1, GLdouble x2, GLdouble y2)`
- `void glRectf(GLfloat x1, GLfloat y1, GLfloat x2, GLfloat y2)`
- `void glRecti(GLint x1, GLint y1, GLint x2, GLint y2)`
- `void glRects(GLshort x1, GLshort y1, GLshort x2, GLshort y2)`
- `void glRectdv(const GLdouble * v1, const GLdouble * v2)`
- `void glRectfv(const GLfloat * v1, const GLfloat * v2)`
- `void glRectiv(const GLint * v1, const GLint * v2)`
- `void glRectsv(const GLshort * v1, const GLshort * v2)`
- `GLint glRenderMode(GLenum mode)`
- `void glResetHistogram(GLenum target)`
- `void glRotated(GLdouble angle, GLdouble x, GLdouble y, GLdouble z)`
- `void glRotatef(GLfloat angle, GLfloat x, GLfloat y, GLfloat z)`
- `void glSampleCoverage(GLclampf value, GLboolean invert)`
- `void glScaled(GLdouble x, GLdouble y, GLdouble z)`
- `void glScalef(GLfloat x, GLfloat y, GLfloat z)`
- `void glScissor(GLint x, GLint y, GLsizei width, GLsizei height)`
- `void glSecondaryColor3b(GLbyte red, GLbyte green, GLbyte blue)`
- `void glSecondaryColor3s(GLshort red, GLshort green, GLshort blue)`
- `void glSecondaryColor3i(GLint red, GLint green, GLint blue)`
- `void glSecondaryColor3f(GLfloat red, GLfloat green, GLfloat blue)`
- `void glSecondaryColor3d(GLdouble red, GLdouble green, GLdouble blue)`

- void glSecondaryColor3ub(GLubyte red,GLubyte green,GLubyte blue)
- void glSecondaryColor3us(GLushort red,GLushort green,GLushort blue)
- void glSecondaryColor3ui(GLuint red,GLuint green,GLuint blue)
- void glSecondaryColor3bv(const GLbyte * v)
- void glSecondaryColor3sv(const GLshort * v)
- void glSecondaryColor3iv(const GLint * v)
- void glSecondaryColor3fv(const GLfloat * v)
- void glSecondaryColor3dv(const GLdouble * v)
- void glSecondaryColor3ubv(const GLubyte * v)
- void glSecondaryColor3usv(const GLushort * v)
- void glSecondaryColor3uiv(const GLuint * v)
- void glSecondaryColorPointer(GLint size,GLenum type,GLsizei stride,const GLvoid * pointer)
- void glSelectBuffer(GLsizei size,GLuint * buffer)
- void glSeparableFilter2D(GLenum target,GLenum internalformat,GLsizei width,GLsizei height,GLenum format,GLenum type,const GLvoid * row,const GLvoid * column)
- void glShadeModel(GLenum mode)
- void glShaderSource(GLuint shader,GLsizei count,const GLchar **string,const GLint *length)
- void glStencilFunc(GLenum func,GLint ref,GLuint mask)
- void glStencilFuncSeparate(GLenum face,GLenum func,GLint ref,GLuint mask)
- void glStencilMask(GLuint mask)
- void glStencilMaskSeparate(GLenum face,GLuint mask)
- void glStencilOp(GLenum sfail,GLenum dpfail,GLenum dppass)
- void glStencilOpSeparate(GLenum face,GLenum sfail,GLenum dpfail,GLenum dppass)
- void glTexCoord1s(GLshort s)
- void glTexCoord1i(GLint s)
- void glTexCoord1f(GLfloat s)
- void glTexCoord1d(GLdouble s)
- void glTexCoord2s(GLshort s,GLshort t)
- void glTexCoord2i(GLint s,GLint t)
- void glTexCoord2f(GLfloat s,GLfloat t)
- void glTexCoord2d(GLdouble s,GLdouble t)
- void glTexCoord3s(GLshort s,GLshort t,GLshort r)
- void glTexCoord3i(GLint s,GLint t,GLint r)
- void glTexCoord3f(GLfloat s,GLfloat t,GLfloat r)
- void glTexCoord3d(GLdouble s,GLdouble t,GLdouble r)
- void glTexCoord4s(GLshort s,GLshort t,GLshort r,GLshort q)

- `void glTexCoord4i(GLint s,GLint t,GLint r,GLint q)`
- `void glTexCoord4f(GLfloat s,GLfloat t,GLfloat r,GLfloat q)`
- `void glTexCoord4d(GLdouble s,GLdouble t,GLdouble r,GLdouble q)`
- `void glTexCoord1sv(const GLshort * v)`
- `void glTexCoord1iv(const GLint * v)`
- `void glTexCoord1fv(const GLfloat * v)`
- `void glTexCoord1dv(const GLdouble * v)`
- `void glTexCoord2sv(const GLshort * v)`
- `void glTexCoord2iv(const GLint * v)`
- `void glTexCoord2fv(const GLfloat * v)`
- `void glTexCoord2dv(const GLdouble * v)`
- `void glTexCoord3sv(const GLshort * v)`
- `void glTexCoord3iv(const GLint * v)`
- `void glTexCoord3fv(const GLfloat * v)`
- `void glTexCoord3dv(const GLdouble * v)`
- `void glTexCoord4sv(const GLshort * v)`
- `void glTexCoord4iv(const GLint * v)`
- `void glTexCoord4fv(const GLfloat * v)`
- `void glTexCoord4dv(const GLdouble * v)`
- `void glTexCoordPointer(GLint size,GLenum type,GLsizei stride,const GLvoid * pointer)`
- `void glTexEnvf(GLenum target,GLenum pname,GLfloat param)`
- `void glTexEnvi(GLenum target,GLenum pname,GLint param)`
- `void glTexGeni(GLenum coord,GLenum pname,GLint param)`
- `void glTexGenf(GLenum coord,GLenum pname,GLfloat param)`
- `void glTexGend(GLenum coord,GLenum pname,GLdouble param)`
- `void glTexGeniv(GLenum coord,GLenum pname,const GLint * params)`
- `void glTexGenfv(GLenum coord,GLenum pname,const GLfloat * params)`
- `void glTexGendv(GLenum coord,GLenum pname,const GLdouble * params)`
- `void glTexImage1D(GLenum target,GLint level,GLint internalFormat,GLsizei width,GLint border,GLenum format,GLenum type,const GLvoid * data)`
- `void glTexImage2D(GLenum target,GLint level,GLint internalFormat,GLsizei width,GLsizei height,GLint border,GLenum format,GLenum type,const GLvoid * data)`
- `void glTexImage3D(GLenum target,GLint level,GLint internalFormat,GLsizei width,GLsizei height,GLsizei depth,GLint border,GLenum format,GLenum type,const GLvoid * data)`
- `void glTexParameterf(GLenum target,GLenum pname,GLfloat param)`
- `void glTexParameteri(GLenum target,GLenum pname,GLint param)`
- `void glTexParameterfv(GLenum target,GLenum pname,const GLfloat * params)`

- `void glTexParameteriv(GLenum target, GLenum pname, const GLint * params)`
- `void glTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLenum type, const GLvoid * data)`
- `void glTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)`
- `void glTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLenum type, const GLvoid * data)`
- `void glTranslated(GLdouble x, GLdouble y, GLdouble z)`
- `void glTranslatef(GLfloat x, GLfloat y, GLfloat z)`
- `void glUniform1f(GLint location, GLfloat v0)`
- `void glUniform2f(GLint location, GLfloat v0, GLfloat v1)`
- `void glUniform3f(GLint location, GLfloat v0, GLfloat v1, GLfloat v2)`
- `void glUniform4f(GLint location, GLfloat v0, GLfloat v1, GLfloat v2, GLfloat v3)`
- `void glUniform1i(GLint location, GLint v0)`
- `void glUniform2i(GLint location, GLint v0, GLint v1)`
- `void glUniform3i(GLint location, GLint v0, GLint v1, GLint v2)`
- `void glUniform4i(GLint location, GLint v0, GLint v1, GLint v2, GLint v3)`
- `void glUniform1fv(GLint location, GLsizei count, const GLfloat *value)`
- `void glUniform2fv(GLint location, GLsizei count, const GLfloat *value)`
- `void glUniform3fv(GLint location, GLsizei count, const GLfloat *value)`
- `void glUniform4fv(GLint location, GLsizei count, const GLfloat *value)`
- `void glUniform1iv(GLint location, GLsizei count, const GLint *value)`
- `void glUniform2iv(GLint location, GLsizei count, const GLint *value)`
- `void glUniform3iv(GLint location, GLsizei count, const GLint *value)`
- `void glUniform4iv(GLint location, GLsizei count, const GLint *value)`
- `void glUniformMatrix2fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix3fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix4fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix2x3fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix3x2fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix2x4fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix4x2fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix3x4fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix4x3fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUseProgram(GLuint program)`
- `void glValidateProgram(GLuint program)`
- `void glVertex2s(GLshort x, GLshort y)`

- `void glVertex2i(GLint x, GLint y)`
- `void glVertex2f(GLfloat x, GLfloat y)`
- `void glVertex2d(GLdouble x, GLdouble y)`
- `void glVertex3s(GLshort x, GLshort y, GLshort z)`
- `void glVertex3i(GLint x, GLint y, GLint z)`
- `void glVertex3f(GLfloat x, GLfloat y, GLfloat z)`
- `void glVertex3d(GLdouble x, GLdouble y, GLdouble z)`
- `void glVertex4s(GLshort x, GLshort y, GLshort z, GLshort w)`
- `void glVertex4i(GLint x, GLint y, GLint z, GLint w)`
- `void glVertex4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)`
- `void glVertex4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)`
- `void glVertex2sv(const GLshort * v)`
- `void glVertex2iv(const GLint * v)`
- `void glVertex2fv(const GLfloat * v)`
- `void glVertex2dv(const GLdouble * v)`
- `void glVertex3sv(const GLshort * v)`
- `void glVertex3iv(const GLint * v)`
- `void glVertex3fv(const GLfloat * v)`
- `void glVertex3dv(const GLdouble * v)`
- `void glVertex4sv(const GLshort * v)`
- `void glVertex4iv(const GLint * v)`
- `void glVertex4fv(const GLfloat * v)`
- `void glVertex4dv(const GLdouble * v)`
- `void glVertexAttrib1f(GLuint index, GLfloat v0)`
- `void glVertexAttrib1s(GLuint index, GLshort v0)`
- `void glVertexAttrib1d(GLuint index, GLdouble v0)`
- `void glVertexAttrib2f(GLuint index, GLfloat v0, GLfloat v1)`
- `void glVertexAttrib2s(GLuint index, GLshort v0, GLshort v1)`
- `void glVertexAttrib2d(GLuint index, GLdouble v0, GLdouble v1)`
- `void glVertexAttrib3f(GLuint index, GLfloat v0, GLfloat v1, GLfloat v2)`
- `void glVertexAttrib3s(GLuint index, GLshort v0, GLshort v1, GLshort v2)`
- `void glVertexAttrib3d(GLuint index, GLdouble v0, GLdouble v1, GLdouble v2)`
- `void glVertexAttrib4f(GLuint index, GLfloat v0, GLfloat v1, GLfloat v2, GLfloat v3)`
- `void glVertexAttrib4s(GLuint index, GLshort v0, GLshort v1, GLshort v2, GLshort v3)`
- `void glVertexAttrib4d(GLuint index, GLdouble v0, GLdouble v1, GLdouble v2, GLdouble v3)`
- `void glVertexAttrib4Nub(GLuint index, GLubyte v0, GLubyte v1, GLubyte v2, GLubyte v3)`

- void glVertexAttrib1fv(GLuint index,const GLfloat *v)
- void glVertexAttrib1sv(GLuint index,const GLshort *v)
- void glVertexAttrib1dv(GLuint index,const GLdouble *v)
- void glVertexAttrib2fv(GLuint index,const GLfloat *v)
- void glVertexAttrib2sv(GLuint index,const GLshort *v)
- void glVertexAttrib2dv(GLuint index,const GLdouble *v)
- void glVertexAttrib3fv(GLuint index,const GLfloat *v)
- void glVertexAttrib3sv(GLuint index,const GLshort *v)
- void glVertexAttrib3dv(GLuint index,const GLdouble *v)
- void glVertexAttrib4fv(GLuint index,const GLfloat *v)
- void glVertexAttrib4sv(GLuint index,const GLshort *v)
- void glVertexAttrib4dv(GLuint index,const GLdouble *v)
- void glVertexAttrib4iv(GLuint index,const GLint *v)
- void glVertexAttrib4bv(GLuint index,const GLbyte *v)
- void glVertexAttrib4ubv(GLuint index,const GLubyte *v)
- void glVertexAttrib4usv(GLuint index,const GLushort *v)
- void glVertexAttrib4uiv(GLuint index,const GLuint *v)
- void glVertexAttribPointer(GLuint index,GLint size,GLenum type,GLboolean normalized,GLsizei stride,const GLvoid * pointer)
- void glVertexPointer(GLint size,GLenum type,GLsizei stride,const GLvoid * pointer)
- void glViewport(GLint x,GLint y,GLsizei width,GLsizei height)
- void glWindowPos2s(GLshort x,GLshort y)
- void glWindowPos2i(GLint x,GLint y)
- void glWindowPos2f(GLfloat x,GLfloat y)
- void glWindowPos2d(GLdouble x,GLdouble y)
- void glWindowPos3s(GLshort x,GLshort y,GLshort z)
- void glWindowPos3i(GLint x,GLint y,GLint z)
- void glWindowPos3f(GLfloat x,GLfloat y,GLfloat z)
- void glWindowPos3d(GLdouble x,GLdouble y,GLdouble z)
- void glWindowPos2sv(const GLshort * v)
- void glWindowPos2iv(const GLint * v)
- void glWindowPos2fv(const GLfloat * v)
- void glWindowPos2dv(const GLdouble * v)
- void glWindowPos3sv(const GLshort * v)
- void glWindowPos3iv(const GLint * v)
- void glWindowPos3fv(const GLfloat * v)

- void glWindowPos3dv(const GLdouble * v)
- void gluBeginCurve(GLUnurbs* nurb)
- void gluBeginPolygon(GLUtesselator* tess)
- void gluBeginSurface(GLUnurbs* nurb)
- void gluBeginTrim(GLUnurbs* nurb)
- void gluCylinder(GLUquadric* quad, GLdouble base, GLdouble top, GLdouble height, GLint slices, GLint stacks)
- void gluDeleteNurbsRenderer(GLUnurbs* nurb)
- void gluDeleteQuadric(GLUquadric* quad)
- void gluDeleteTess(GLUtesselator* tess)
- void gluDisk(GLUquadric* quad, GLdouble inner, GLdouble outer, GLint slices, GLint loops)
- void gluEndCurve(GLUnurbs* nurb)
- void gluEndPolygon(GLUtesselator* tess)
- void gluEndSurface(GLUnurbs* nurb)
- void gluEndTrim(GLUnurbs* nurb)
- const GLubyte * gluErrorString(GLenum error)
- void gluGetNurbsProperty(GLUnurbs* nurb, GLenum property, GLfloat* data)
- const GLubyte * gluGetString(GLenum name)
- void gluGetTessProperty(GLUtesselator* tess, GLenum which, GLdouble* data)
- void gluLoadSamplingMatrices(GLUnurbs* nurb, const GLfloat * model, const GLfloat * perspective, const GLint * view)
- void gluLookAt(GLdouble eyeX, GLdouble eyeY, GLdouble eyeZ, GLdouble centerX, GLdouble centerY, GLdouble centerZ, GLdouble upX, GLdouble upY, GLdouble upZ)
- GLUnurbs *gluNewNurbsRenderer(void)
- GLUquadric *gluNewQuadric(void)
- GLUtesselator* gluNewTess(void)
- void gluNextContour(GLUtesselator* tess, GLenum type)
- void gluNurbsCurve(GLUnurbs* nurb, GLint knotCount, GLfloat * knots, GLint stride, GLfloat * control, GLint order, GLenum type)
- void gluNurbsProperty(GLUnurbs* nurb, GLenum property, GLfloat value)
- void gluNurbsSurface(GLUnurbs* nurb, GLint sKnotCount, GLfloat* sKnots, GLint tKnotCount, GLfloat* tKnots, GLint sStride, GLint tStride, GLfloat* control, GLint sOrder, GLint tOrder, GLenum type)
- void gluOrtho2D(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top)
- void gluPartialDisk(GLUquadric* quad, GLdouble inner, GLdouble outer, GLint slices, GLint loops, GLdouble start, GLdouble sweep)
- void gluPerspective(GLdouble fovy, GLdouble aspect, GLdouble zNear, GLdouble zFar)
- void gluPickMatrix(GLdouble x, GLdouble y, GLdouble delX, GLdouble delY, GLint * viewport)
- GLint gluProject(GLdouble objX, GLdouble objY, GLdouble objZ, const GLdouble * model, const GLdouble * proj, const GLint * view, GLdouble* winX, GLdouble* winY, GLdouble* winZ)

- void gluPwlCurve(GLUnurbs* nurb,GLint count,GLfloat* data,GLint stride,GLenum type)
- void gluQuadricDrawStyle(GLUquadric* quad,GLenum draw)
- void gluQuadricNormals(GLUquadric* quad,GLenum normal)
- void gluQuadricOrientation(GLUquadric* quad,GLenum orientation)
- void gluQuadricTexture(GLUquadric* quad,GLboolean texture)
- GLint gluScaleImage(GLenum format,GLsizei wIn,GLsizei hIn,GLenum typeIn,const void * dataIn,GLsizei wOut,GLsizei hOut,GLenum typeOut,GLvoid* dataOut)
- void gluSphere(GLUquadric* quad,GLdouble radius,GLint slices,GLint stacks)
- void gluTessBeginContour(GLUtesselator* tess)
- void gluTessBeginPolygon(GLUtesselator* tess,GLvoid* data)
- void gluTessEndContour(GLUtesselator* tess)
- void gluTessEndPolygon(GLUtesselator* tess)
- void gluTessNormal(GLUtesselator* tess,GLdouble valueX,GLdouble valueY,GLdouble valueZ)
- void gluTessProperty(GLUtesselator* tess,GLenum which,GLdouble data)
- void gluTessVertex(GLUtesselator* tess,GLdouble * location,GLvoid* data)
- GLint gluUnProject(GLdouble winX,GLdouble winY,GLdouble winZ,const GLdouble * model,const GLdouble * proj,const GLint * view,GLdouble* objX,GLdouble* objY,GLdouble* objZ)
- void glDisable(GLenum cap)

RINGOPENGL (OPENGL 4.2) FUNCTIONS REFERENCE

- GL_ZERO
- GL_FALSE
- GL_LOGIC_OP
- GL_NONE
- GL_TEXTURE_COMPONENTS
- GL_NO_ERROR
- GL_POINTS
- GL_CURRENT_BIT
- GL_TRUE
- GL_ONE
- GL_CLIENT_PIXEL_STORE_BIT
- GL_LINES
- GL_LINE_LOOP
- GL_POINT_BIT
- GL_CLIENT_VERTEX_ARRAY_BIT
- GL_LINE_STRIP
- GL_LINE_BIT
- GL_TRIANGLES
- GL_TRIANGLE_STRIP
- GL_TRIANGLE_FAN
- GL_QUADS
- GL_QUAD_STRIP
- GL_POLYGON_BIT
- GL_POLYGON
- GL_POLYGON_STIPPLE_BIT
- GL_PIXEL_MODE_BIT
- GL_LIGHTING_BIT

- GL_FOG_BIT
- GL_DEPTH_BUFFER_BIT
- GL_ACCUM
- GL_LOAD
- GL_RETURN
- GL_MULT
- GL_ADD
- GL_NEVER
- GL_ACCUM_BUFFER_BIT
- GL_LESS
- GL_EQUAL
- GL_LEQUAL
- GL_GREATER
- GL_NOTEQUAL
- GL_GEQUAL
- GL_ALWAYS
- GL_SRC_COLOR
- GL_ONE_MINUS_SRC_COLOR
- GL_SRC_ALPHA
- GL_ONE_MINUS_SRC_ALPHA
- GL_DST_ALPHA
- GL_ONE_MINUS_DST_ALPHA
- GL_DST_COLOR
- GL_ONE_MINUS_DST_COLOR
- GL_SRC_ALPHA_SATURATE
- GL_STENCIL_BUFFER_BIT
- GL_FRONT_LEFT
- GL_FRONT_RIGHT
- GL_BACK_LEFT
- GL_BACK_RIGHT
- GL_FRONT
- GL_BACK
- GL_LEFT
- GL_RIGHT
- GL_FRONT_AND_BACK
- GL_AUX0

- GL_AUX1
- GL_AUX2
- GL_AUX3
- GL_INVALID_ENUM
- GL_INVALID_VALUE
- GL_INVALID_OPERATION
- GL_STACK_OVERFLOW
- GL_STACK_UNDERFLOW
- GL_OUT_OF_MEMORY
- GL_2D
- GL_3D
- GL_3D_COLOR
- GL_3D_COLOR_TEXTURE
- GL_4D_COLOR_TEXTURE
- GL_PASS_THROUGH_TOKEN
- GL_POINT_TOKEN
- GL_LINE_TOKEN
- GL_POLYGON_TOKEN
- GL_BITMAP_TOKEN
- GL_DRAW_PIXEL_TOKEN
- GL_COPY_PIXEL_TOKEN
- GL_LINE_RESET_TOKEN
- GL_EXP
- GL_VIEWPORT_BIT
- GL_EXP2
- GL_CW
- GL_CCW
- GL_COEFF
- GL_ORDER
- GL_DOMAIN
- GL_CURRENT_COLOR
- GL_CURRENT_INDEX
- GL_CURRENT_NORMAL
- GL_CURRENT_TEXTURE_COORDS
- GL_CURRENT_RASTER_COLOR
- GL_CURRENT_RASTER_INDEX

- GL_CURRENT_RASTER_TEXTURE_COORDS
- GL_CURRENT_RASTER_POSITION
- GL_CURRENT_RASTER_POSITION_VALID
- GL_CURRENT_RASTER_DISTANCE
- GL_POINT_SMOOTH
- GL_POINT_SIZE
- GL_POINT_SIZE_RANGE
- GL_POINT_SIZE_GRANULARITY
- GL_LINE_SMOOTH
- GL_LINE_WIDTH
- GL_LINE_WIDTH_RANGE
- GL_LINE_WIDTH_GRANULARITY
- GL_LINE_STIPPLE
- GL_LINE_STIPPLE_PATTERN
- GL_LINE_STIPPLE_REPEAT
- GL_LIST_MODE
- GL_MAX_LIST_NESTING
- GL_LIST_BASE
- GL_LIST_INDEX
- GL_POLYGON_MODE
- GL_POLYGON_SMOOTH
- GL_POLYGON_STIPPLE
- GL_EDGE_FLAG
- GL_CULL_FACE
- GL_CULL_FACE_MODE
- GL_FRONT_FACE
- GL_LIGHTING
- GL_LIGHT_MODEL_LOCAL_VIEWER
- GL_LIGHT_MODEL_TWO_SIDE
- GL_LIGHT_MODEL_AMBIENT
- GL_SHADE_MODEL
- GL_COLOR_MATERIAL_FACE
- GL_COLOR_MATERIAL_PARAMETER
- GL_COLOR_MATERIAL
- GL_FOG
- GL_FOG_INDEX

- GL_FOG_DENSITY
- GL_FOG_START
- GL_FOG_END
- GL_FOG_MODE
- GL_FOG_COLOR
- GL_DEPTH_RANGE
- GL_DEPTH_TEST
- GL_DEPTH_WRITEMASK
- GL_DEPTH_CLEAR_VALUE
- GL_DEPTH_FUNC
- GL_ACCUM_CLEAR_VALUE
- GL_STENCIL_TEST
- GL_STENCIL_CLEAR_VALUE
- GL_STENCIL_FUNC
- GL_STENCIL_VALUE_MASK
- GL_STENCIL_FAIL
- GL_STENCIL_PASS_DEPTH_FAIL
- GL_STENCIL_PASS_DEPTH_PASS
- GL_STENCIL_REF
- GL_STENCIL_WRITEMASK
- GL_MATRIX_MODE
- GL_NORMALIZE
- GL_VIEWPORT
- GL_MODELVIEW_STACK_DEPTH
- GL_PROJECTION_STACK_DEPTH
- GL_TEXTURE_STACK_DEPTH
- GL_MODELVIEW_MATRIX
- GL_PROJECTION_MATRIX
- GL_TEXTURE_MATRIX
- GL_ATTRIB_STACK_DEPTH
- GL_CLIENT_ATTRIB_STACK_DEPTH
- GL_ALPHA_TEST
- GL_ALPHA_TEST_FUNC
- GL_ALPHA_TEST_REF
- GL_DITHER
- GL_BLEND_DST

- GL_BLEND_SRC
- GL_BLEND
- GL_LOGIC_OP_MODE
- GL_INDEX_LOGIC_OP
- GL_COLOR_LOGIC_OP
- GL_AUX_BUFFERS
- GL_DRAW_BUFFER
- GL_READ_BUFFER
- GL_SCISSOR_BOX
- GL_SCISSOR_TEST
- GL_INDEX_CLEAR_VALUE
- GL_INDEX_WRITEMASK
- GL_COLOR_CLEAR_VALUE
- GL_COLOR_WRITEMASK
- GL_INDEX_MODE
- GL_RGBA_MODE
- GL_DOUBLEBUFFER
- GL_STEREO
- GL_RENDER_MODE
- GL_PERSPECTIVE_CORRECTION_HINT
- GL_POINT_SMOOTH_HINT
- GL_LINE_SMOOTH_HINT
- GL_POLYGON_SMOOTH_HINT
- GL_FOG_HINT
- GL_TEXTURE_GEN_S
- GL_TEXTURE_GEN_T
- GL_TEXTURE_GEN_R
- GL_TEXTURE_GEN_Q
- GL_PIXEL_MAP_I_TO_I
- GL_PIXEL_MAP_S_TO_S
- GL_PIXEL_MAP_I_TO_R
- GL_PIXEL_MAP_I_TO_G
- GL_PIXEL_MAP_I_TO_B
- GL_PIXEL_MAP_I_TO_A
- GL_PIXEL_MAP_R_TO_R
- GL_PIXEL_MAP_G_TO_G

- GL_PIXEL_MAP_B_TO_B
- GL_PIXEL_MAP_A_TO_A
- GL_PIXEL_MAP_I_TO_I_SIZE
- GL_PIXEL_MAP_S_TO_S_SIZE
- GL_PIXEL_MAP_I_TO_R_SIZE
- GL_PIXEL_MAP_I_TO_G_SIZE
- GL_PIXEL_MAP_I_TO_B_SIZE
- GL_PIXEL_MAP_I_TO_A_SIZE
- GL_PIXEL_MAP_R_TO_R_SIZE
- GL_PIXEL_MAP_G_TO_G_SIZE
- GL_PIXEL_MAP_B_TO_B_SIZE
- GL_PIXEL_MAP_A_TO_A_SIZE
- GL_UNPACK_SWAP_BYTES
- GL_UNPACK_LSB_FIRST
- GL_UNPACK_ROW_LENGTH
- GL_UNPACK_SKIP_ROWS
- GL_UNPACK_SKIP_PIXELS
- GL_UNPACK_ALIGNMENT
- GL_PACK_SWAP_BYTES
- GL_PACK_LSB_FIRST
- GL_PACK_ROW_LENGTH
- GL_PACK_SKIP_ROWS
- GL_PACK_SKIP_PIXELS
- GL_PACK_ALIGNMENT
- GL_MAP_COLOR
- GL_MAP_STENCIL
- GL_INDEX_SHIFT
- GL_INDEX_OFFSET
- GL_RED_SCALE
- GL_RED_BIAS
- GL_ZOOM_X
- GL_ZOOM_Y
- GL_GREEN_SCALE
- GL_GREEN_BIAS
- GL_BLUE_SCALE
- GL_BLUE_BIAS

- GL_ALPHA_SCALE
- GL_ALPHA_BIAS
- GL_DEPTH_SCALE
- GL_DEPTH_BIAS
- GL_MAX_EVAL_ORDER
- GL_MAX_LIGHTS
- GL_MAX_CLIP_PLANES
- GL_MAX_TEXTURE_SIZE
- GL_MAX_PIXEL_MAP_TABLE
- GL_MAX_ATTRIB_STACK_DEPTH
- GL_MAX_MODELVIEW_STACK_DEPTH
- GL_MAX_NAME_STACK_DEPTH
- GL_MAX_PROJECTION_STACK_DEPTH
- GL_MAX_TEXTURE_STACK_DEPTH
- GL_MAX_VIEWPORT_DIMS
- GL_MAX_CLIENT_ATTRIB_STACK_DEPTH
- GL_SUBPIXEL_BITS
- GL_INDEX_BITS
- GL_RED_BITS
- GL_GREEN_BITS
- GL_BLUE_BITS
- GL_ALPHA_BITS
- GL_DEPTH_BITS
- GL_STENCIL_BITS
- GL_ACCUM_RED_BITS
- GL_ACCUM_GREEN_BITS
- GL_ACCUM_BLUE_BITS
- GL_ACCUM_ALPHA_BITS
- GL_NAME_STACK_DEPTH
- GL_AUTO_NORMAL
- GL_MAP1_COLOR_4
- GL_MAP1_INDEX
- GL_MAP1_NORMAL
- GL_MAP1_TEXTURE_COORD_1
- GL_MAP1_TEXTURE_COORD_2
- GL_MAP1_TEXTURE_COORD_3

- GL_MAP1_TEXTURE_COORD_4
- GL_MAP1_VERTEX_3
- GL_MAP1_VERTEX_4
- GL_MAP2_COLOR_4
- GL_MAP2_INDEX
- GL_MAP2_NORMAL
- GL_MAP2_TEXTURE_COORD_1
- GL_MAP2_TEXTURE_COORD_2
- GL_MAP2_TEXTURE_COORD_3
- GL_MAP2_TEXTURE_COORD_4
- GL_MAP2_VERTEX_3
- GL_MAP2_VERTEX_4
- GL_MAP1_GRID_DOMAIN
- GL_MAP1_GRID_SEGMENTS
- GL_MAP2_GRID_DOMAIN
- GL_MAP2_GRID_SEGMENTS
- GL_TEXTURE_1D
- GL_TEXTURE_2D
- GL_FEEDBACK_BUFFER_POINTER
- GL_FEEDBACK_BUFFER_SIZE
- GL_FEEDBACK_BUFFER_TYPE
- GL_SELECTION_BUFFER_POINTER
- GL_SELECTION_BUFFER_SIZE
- GL_TEXTURE_WIDTH
- GL_TRANSFORM_BIT
- GL_TEXTURE_HEIGHT
- GL_TEXTURE_INTERNAL_FORMAT
- GL_TEXTURE_BORDER_COLOR
- GL_TEXTURE_BORDER
- GL_DONT_CARE
- GL_FASTEST
- GL_NICEST
- GL_AMBIENT
- GL_DIFFUSE
- GL_SPECULAR
- GL_POSITION

- GL_SPOT_DIRECTION
- GL_SPOT_EXPONENT
- GL_SPOT_CUTOFF
- GL_CONSTANT_ATTENUATION
- GL_LINEAR_ATTENUATION
- GL_QUADRATIC_ATTENUATION
- GL_COMPILE
- GL_COMPILE_AND_EXECUTE
- GL_BYTE
- GL_UNSIGNED_BYTE
- GL_SHORT
- GL_UNSIGNED_SHORT
- GL_INT
- GL_UNSIGNED_INT
- GL_FLOAT
- GL_2_BYTES
- GL_3_BYTES
- GL_4_BYTES
- GL_DOUBLE
- GL_CLEAR
- GL_AND
- GL_AND_REVERSE
- GL_COPY
- GL_AND_INVERTED
- GL_NOOP
- GL_XOR
- GL_OR
- GL_NOR
- GL_EQUIV
- GL_INVERT
- GL_OR_REVERSE
- GL_COPY_INVERTED
- GL_OR_INVERTED
- GL_NAND
- GL_SET
- GL_EMISSION

- GL_SHININESS
- GL_AMBIENT_AND_DIFFUSE
- GL_COLOR_INDEXES
- GL_MODELVIEW
- GL_PROJECTION
- GL_TEXTURE
- GL_COLOR
- GL_DEPTH
- GL_STENCIL
- GL_COLOR_INDEX
- GL_STENCIL_INDEX
- GL_DEPTH_COMPONENT
- GL_RED
- GL_GREEN
- GL_BLUE
- GL_ALPHA
- GL_RGB
- GL_RGBA
- GL_LUMINANCE
- GL_LUMINANCE_ALPHA
- GL_BITMAP
- GL_POINT
- GL_LINE
- GL_FILL
- GL_RENDER
- GL_FEEDBACK
- GL_SELECT
- GL_FLAT
- GL_SMOOTH
- GL_KEEP
- GL_REPLACE
- GL_INCR
- GL_DECR
- GL_VENDOR
- GL_RENDERER
- GL_VERSION

- GL_EXTENSIONS
- GL_S
- GL_ENABLE_BIT
- GL_T
- GL_R
- GL_Q
- GL_MODULATE
- GL_DECAL
- GL_TEXTURE_ENV_MODE
- GL_TEXTURE_ENV_COLOR
- GL_TEXTURE_ENV
- GL_EYE_LINEAR
- GL_OBJECT_LINEAR
- GL_SPHERE_MAP
- GL_TEXTURE_GEN_MODE
- GL_OBJECT_PLANE
- GL_EYE_PLANE
- GL_NEAREST
- GL_LINEAR
- GL_NEAREST_MIPMAP_NEAREST
- GL_LINEAR_MIPMAP_NEAREST
- GL_NEAREST_MIPMAP_LINEAR
- GL_LINEAR_MIPMAP_LINEAR
- GL_TEXTURE_MAG_FILTER
- GL_TEXTURE_MIN_FILTER
- GL_TEXTURE_WRAP_S
- GL_TEXTURE_WRAP_T
- GL_CLAMP
- GL_REPEAT
- GL_POLYGON_OFFSET_UNITS
- GL_POLYGON_OFFSET_POINT
- GL_POLYGON_OFFSET_LINE
- GL_R3_G3_B2
- GL_V2F
- GL_V3F
- GL_C4UB_V2F

- GL_C4UB_V3F
- GL_C3F_V3F
- GL_N3F_V3F
- GL_C4F_N3F_V3F
- GL_T2F_V3F
- GL_T4F_V4F
- GL_T2F_C4UB_V3F
- GL_T2F_C3F_V3F
- GL_T2F_N3F_V3F
- GL_T2F_C4F_N3F_V3F
- GL_T4F_C4F_N3F_V4F
- GL_CLIP_PLANE0
- GL_CLIP_PLANE1
- GL_CLIP_PLANE2
- GL_CLIP_PLANE3
- GL_CLIP_PLANE4
- GL_CLIP_PLANE5
- GL_LIGHT0
- GL_COLOR_BUFFER_BIT
- GL_LIGHT1
- GL_LIGHT2
- GL_LIGHT3
- GL_LIGHT4
- GL_LIGHT5
- GL_LIGHT6
- GL_LIGHT7
- GL_HINT_BIT
- GL_POLYGON_OFFSET_FILL
- GL_POLYGON_OFFSET_FACTOR
- GL_ALPHA4
- GL_ALPHA8
- GL_ALPHA12
- GL_ALPHA16
- GL_LUMINANCE4
- GL_LUMINANCE8
- GL_LUMINANCE12

- GL_LUMINANCE16
- GL_LUMINANCE4_ALPHA4
- GL_LUMINANCE6_ALPHA2
- GL_LUMINANCE8_ALPHA8
- GL_LUMINANCE12_ALPHA4
- GL_LUMINANCE12_ALPHA12
- GL_LUMINANCE16_ALPHA16
- GL_INTENSITY
- GL_INTENSITY4
- GL_INTENSITY8
- GL_INTENSITY12
- GL_INTENSITY16
- GL_RGB4
- GL_RGB5
- GL_RGB8
- GL_RGB10
- GL_RGB12
- GL_RGB16
- GL_RGBA2
- GL_RGBA4
- GL_RGB5_A1
- GL_RGBA8
- GL_RGB10_A2
- GL_RGBA12
- GL_RGBA16
- GL_TEXTURE_RED_SIZE
- GL_TEXTURE_GREEN_SIZE
- GL_TEXTURE_BLUE_SIZE
- GL_TEXTURE_ALPHA_SIZE
- GL_TEXTURE_LUMINANCE_SIZE
- GL_TEXTURE_INTENSITY_SIZE
- GL_PROXY_TEXTURE_1D
- GL_PROXY_TEXTURE_2D
- GL_TEXTURE_PRIORITY
- GL_TEXTURE_RESIDENT
- GL_TEXTURE_BINDING_1D

- GL_TEXTURE_BINDING_2D
- GL_VERTEX_ARRAY
- GL_NORMAL_ARRAY
- GL_COLOR_ARRAY
- GL_INDEX_ARRAY
- GL_TEXTURE_COORD_ARRAY
- GL_EDGE_FLAG_ARRAY
- GL_VERTEX_ARRAY_SIZE
- GL_VERTEX_ARRAY_TYPE
- GL_VERTEX_ARRAY_STRIDE
- GL_NORMAL_ARRAY_TYPE
- GL_NORMAL_ARRAY_STRIDE
- GL_COLOR_ARRAY_SIZE
- GL_COLOR_ARRAY_TYPE
- GL_COLOR_ARRAY_STRIDE
- GL_INDEX_ARRAY_TYPE
- GL_INDEX_ARRAY_STRIDE
- GL_TEXTURE_COORD_ARRAY_SIZE
- GL_TEXTURE_COORD_ARRAY_TYPE
- GL_TEXTURE_COORD_ARRAY_STRIDE
- GL_EDGE_FLAG_ARRAY_STRIDE
- GL_VERTEX_ARRAY_POINTER
- GL_NORMAL_ARRAY_POINTER
- GL_COLOR_ARRAY_POINTER
- GL_INDEX_ARRAY_POINTER
- GL_TEXTURE_COORD_ARRAY_POINTER
- GL_EDGE_FLAG_ARRAY_POINTER
- GL_COLOR_INDEX1_EXT
- GL_COLOR_INDEX2_EXT
- GL_COLOR_INDEX4_EXT
- GL_COLOR_INDEX8_EXT
- GL_COLOR_INDEX12_EXT
- GL_COLOR_INDEX16_EXT
- GL_EVAL_BIT
- GL_LIST_BIT
- GL_TEXTURE_BIT

- GL_SCISSOR_BIT
- GL_ALL_ATTRIB_BITS
- GL_CLIENT_ALL_ATTRIB_BITS
- GL_SMOOTH_POINT_SIZE_RANGE
- GL_SMOOTH_POINT_SIZE_GRANULARITY
- GL_SMOOTH_LINE_WIDTH_RANGE
- GL_SMOOTH_LINE_WIDTH_GRANULARITY
- GL_UNSIGNED_BYTE_3_3_2
- GL_UNSIGNED_SHORT_4_4_4_4
- GL_UNSIGNED_SHORT_5_5_5_1
- GL_UNSIGNED_INT_8_8_8_8
- GL_UNSIGNED_INT_10_10_10_2
- GL_RESCALE_NORMAL
- GL_TEXTURE_BINDING_3D
- GL_PACK_SKIP_IMAGES
- GL_PACK_IMAGE_HEIGHT
- GL_UNPACK_SKIP_IMAGES
- GL_UNPACK_IMAGE_HEIGHT
- GL_TEXTURE_3D
- GL_PROXY_TEXTURE_3D
- GL_TEXTURE_DEPTH
- GL_TEXTURE_WRAP_R
- GL_MAX_3D_TEXTURE_SIZE
- GL_BGR
- GL_BGRA
- GL_MAX_ELEMENTS_VERTICES
- GL_MAX_ELEMENTS_INDICES
- GL_CLAMP_TO_EDGE
- GL_TEXTURE_MIN_LOD
- GL_TEXTURE_MAX_LOD
- GL_TEXTURE_BASE_LEVEL
- GL_TEXTURE_MAX_LEVEL
- GL_LIGHT_MODEL_COLOR_CONTROL
- GL_SINGLE_COLOR
- GL_SEPARATE_SPECULAR_COLOR
- GL_UNSIGNED_BYTE_2_3_3_REV

- GL_UNSIGNED_SHORT_5_6_5
- GL_UNSIGNED_SHORT_5_6_5_REV
- GL_UNSIGNED_SHORT_4_4_4_4_REV
- GL_UNSIGNED_SHORT_1_5_5_5_REV
- GL_UNSIGNED_INT_8_8_8_8_REV
- GL_ALIASED_POINT_SIZE_RANGE
- GL_ALIASED_LINE_WIDTH_RANGE
- GL_MULTISAMPLE
- GL_SAMPLE_ALPHA_TO_COVERAGE
- GL_SAMPLE_ALPHA_TO_ONE
- GL_SAMPLE_COVERAGE
- GL_SAMPLE_BUFFERS
- GL_SAMPLES
- GL_SAMPLE_COVERAGE_VALUE
- GL_SAMPLE_COVERAGE_INVERT
- GL_CLAMP_TO_BORDER
- GL_TEXTURE0
- GL_TEXTURE1
- GL_TEXTURE2
- GL_TEXTURE3
- GL_TEXTURE4
- GL_TEXTURE5
- GL_TEXTURE6
- GL_TEXTURE7
- GL_TEXTURE8
- GL_TEXTURE9
- GL_TEXTURE10
- GL_TEXTURE11
- GL_TEXTURE12
- GL_TEXTURE13
- GL_TEXTURE14
- GL_TEXTURE15
- GL_TEXTURE16
- GL_TEXTURE17
- GL_TEXTURE18
- GL_TEXTURE19

- GL_TEXTURE20
- GL_TEXTURE21
- GL_TEXTURE22
- GL_TEXTURE23
- GL_TEXTURE24
- GL_TEXTURE25
- GL_TEXTURE26
- GL_TEXTURE27
- GL_TEXTURE28
- GL_TEXTURE29
- GL_TEXTURE30
- GL_TEXTURE31
- GL_ACTIVE_TEXTURE
- GL_CLIENT_ACTIVE_TEXTURE
- GL_MAX_TEXTURE_UNITS
- GL_TRANSPOSE_MODELVIEW_MATRIX
- GL_TRANSPOSE_PROJECTION_MATRIX
- GL_TRANSPOSE_TEXTURE_MATRIX
- GL_TRANSPOSE_COLOR_MATRIX
- GL_SUBTRACT
- GL_COMPRESSED_ALPHA
- GL_COMPRESSED_LUMINANCE
- GL_COMPRESSED_LUMINANCE_ALPHA
- GL_COMPRESSED_INTENSITY
- GL_COMPRESSED_RGB
- GL_COMPRESSED_RGBA
- GL_TEXTURE_COMPRESSION_HINT
- GL_NORMAL_MAP
- GL_REFLECTION_MAP
- GL_TEXTURE_CUBE_MAP
- GL_TEXTURE_BINDING_CUBE_MAP
- GL_TEXTURE_CUBE_MAP_POSITIVE_X
- GL_TEXTURE_CUBE_MAP_NEGATIVE_X
- GL_TEXTURE_CUBE_MAP_POSITIVE_Y
- GL_TEXTURE_CUBE_MAP_NEGATIVE_Y
- GL_TEXTURE_CUBE_MAP_POSITIVE_Z

- GL_TEXTURE_CUBE_MAP_NEGATIVE_Z
- GL_PROXY_TEXTURE_CUBE_MAP
- GL_MAX_CUBE_MAP_TEXTURE_SIZE
- GL_COMBINE
- GL_COMBINE_RGB
- GL_COMBINE_ALPHA
- GL_RGB_SCALE
- GL_ADD_SIGNED
- GL_INTERPOLATE
- GL_CONSTANT
- GL_PRIMARY_COLOR
- GL_PREVIOUS
- GL_SOURCE0_RGB
- GL_SOURCE1_RGB
- GL_SOURCE2_RGB
- GL_SOURCE0_ALPHA
- GL_SOURCE1_ALPHA
- GL_SOURCE2_ALPHA
- GL_OPERAND0_RGB
- GL_OPERAND1_RGB
- GL_OPERAND2_RGB
- GL_OPERAND0_ALPHA
- GL_OPERAND1_ALPHA
- GL_OPERAND2_ALPHA
- GL_TEXTURE_COMPRESSED_IMAGE_SIZE
- GL_TEXTURE_COMPRESSED
- GL_NUM_COMPRESSED_TEXTURE_FORMATS
- GL_COMPRESSED_TEXTURE_FORMATS
- GL_DOT3_RGB
- GL_DOT3_RGBA
- GL_MULTISAMPLE_BIT
- GL_BLEND_DST_RGB
- GL_BLEND_SRC_RGB
- GL_BLEND_DST_ALPHA
- GL_BLEND_SRC_ALPHA
- GL_POINT_SIZE_MIN

- GL_POINT_SIZE_MAX
- GL_POINT_FADE_THRESHOLD_SIZE
- GL_POINT_DISTANCE_ATTENUATION
- GL_GENERATE_MIPMAP
- GL_GENERATE_MIPMAP_HINT
- GL_DEPTH_COMPONENT16
- GL_DEPTH_COMPONENT24
- GL_DEPTH_COMPONENT32
- GL_MIRRORED_REPEAT
- GL_FOG_COORDINATE_SOURCE
- GL_FOG_COORDINATE
- GL_FRAGMENT_DEPTH
- GL_CURRENT_FOG_COORDINATE
- GL_FOG_COORDINATE_ARRAY_TYPE
- GL_FOG_COORDINATE_ARRAY_STRIDE
- GL_FOG_COORDINATE_ARRAY_POINTER
- GL_FOG_COORDINATE_ARRAY
- GL_COLOR_SUM
- GL_CURRENT_SECONDARY_COLOR
- GL_SECONDARY_COLOR_ARRAY_SIZE
- GL_SECONDARY_COLOR_ARRAY_TYPE
- GL_SECONDARY_COLOR_ARRAY_STRIDE
- GL_SECONDARY_COLOR_ARRAY_POINTER
- GL_SECONDARY_COLOR_ARRAY
- GL_MAX_TEXTURE_LOD_BIAS
- GL_TEXTURE_FILTER_CONTROL
- GL_TEXTURE_LOD_BIAS
- GL_INCR_WRAP
- GL_DECR_WRAP
- GL_TEXTURE_DEPTH_SIZE
- GL_DEPTH_TEXTURE_MODE
- GL_TEXTURE_COMPARE_MODE
- GL_TEXTURE_COMPARE_FUNC
- GL_COMPARE_R_TO_TEXTURE
- GL_CURRENT_FOG_COORD
- GL_FOG_COORD

- GL_FOG_COORD_ARRAY
- GL_FOG_COORD_ARRAY_BUFFER_BINDING
- GL_FOG_COORD_ARRAY_POINTER
- GL_FOG_COORD_ARRAY_STRIDE
- GL_FOG_COORD_ARRAY_TYPE
- GL_FOG_COORD_SRC
- GL_SRC0_ALPHA
- GL_SRC0_RGB
- GL_SRC1_ALPHA
- GL_SRC1_RGB
- GL_SRC2_ALPHA
- GL_SRC2_RGB
- GL_BUFFER_SIZE
- GL_BUFFER_USAGE
- GL_QUERY_COUNTER_BITS
- GL_CURRENT_QUERY
- GL_QUERY_RESULT
- GL_QUERY_RESULT_AVAILABLE
- GL_ARRAY_BUFFER
- GL_ELEMENT_ARRAY_BUFFER
- GL_ARRAY_BUFFER_BINDING
- GL_ELEMENT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ARRAY_BUFFER_BINDING
- GL_NORMAL_ARRAY_BUFFER_BINDING
- GL_COLOR_ARRAY_BUFFER_BINDING
- GL_INDEX_ARRAY_BUFFER_BINDING
- GL_TEXTURE_COORD_ARRAY_BUFFER_BINDING
- GL_EDGE_FLAG_ARRAY_BUFFER_BINDING
- GL_SECONDARY_COLOR_ARRAY_BUFFER_BINDING
- GL_FOG_COORDINATE_ARRAY_BUFFER_BINDING
- GL_WEIGHT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ATTRIB_ARRAY_BUFFER_BINDING
- GL_READ_ONLY
- GL_WRITE_ONLY
- GL_READ_WRITE
- GL_BUFFER_ACCESS

- GL_BUFFER_MAPPED
- GL_BUFFER_MAP_POINTER
- GL_STREAM_DRAW
- GL_STREAM_READ
- GL_STREAM_COPY
- GL_STATIC_DRAW
- GL_STATIC_READ
- GL_STATIC_COPY
- GL_DYNAMIC_DRAW
- GL_DYNAMIC_READ
- GL_DYNAMIC_COPY
- GL_SAMPLES_PASSED
- GL_BLEND_EQUATION_RGB
- GL_VERTEX_ATTRIB_ARRAY_ENABLED
- GL_VERTEX_ATTRIB_ARRAY_SIZE
- GL_VERTEX_ATTRIB_ARRAY_STRIDE
- GL_VERTEX_ATTRIB_ARRAY_TYPE
- GL_CURRENT_VERTEX_ATTRIB
- GL_VERTEX_PROGRAM_POINT_SIZE
- GL_VERTEX_PROGRAM_TWO_SIDE
- GL_VERTEX_ATTRIB_ARRAY_POINTER
- GL_STENCIL_BACK_FUNC
- GL_STENCIL_BACK_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_PASS
- GL_MAX_DRAW_BUFFERS
- GL_DRAW_BUFFER0
- GL_DRAW_BUFFER1
- GL_DRAW_BUFFER2
- GL_DRAW_BUFFER3
- GL_DRAW_BUFFER4
- GL_DRAW_BUFFER5
- GL_DRAW_BUFFER6
- GL_DRAW_BUFFER7
- GL_DRAW_BUFFER8
- GL_DRAW_BUFFER9

- GL_DRAW_BUFFER10
- GL_DRAW_BUFFER11
- GL_DRAW_BUFFER12
- GL_DRAW_BUFFER13
- GL_DRAW_BUFFER14
- GL_DRAW_BUFFER15
- GL_BLEND_EQUATION_ALPHA
- GL_POINT_SPRITE
- GL_COORD_REPLACE
- GL_MAX_VERTEX_ATTRIBS
- GL_VERTEX_ATTRIB_ARRAY_NORMALIZED
- GL_MAX_TEXTURE_COORDS
- GL_MAX_TEXTURE_IMAGE_UNITS
- GL_FRAGMENT_SHADER
- GL_VERTEX_SHADER
- GL_MAX_FRAGMENT_UNIFORM_COMPONENTS
- GL_MAX_VERTEX_UNIFORM_COMPONENTS
- GL_MAX_VARYING_FLOATS
- GL_MAX_VERTEX_TEXTURE_IMAGE_UNITS
- GL_MAX_COMBINED_TEXTURE_IMAGE_UNITS
- GL_SHADER_TYPE
- GL_FLOAT_VEC2
- GL_FLOAT_VEC3
- GL_FLOAT_VEC4
- GL_INT_VEC2
- GL_INT_VEC3
- GL_INT_VEC4
- GL_BOOL
- GL_BOOL_VEC2
- GL_BOOL_VEC3
- GL_BOOL_VEC4
- GL_FLOAT_MAT2
- GL_FLOAT_MAT3
- GL_FLOAT_MAT4
- GL_SAMPLER_1D
- GL_SAMPLER_2D

- GL_SAMPLER_3D
- GL_SAMPLER_CUBE
- GL_SAMPLER_1D_SHADOW
- GL_SAMPLER_2D_SHADOW
- GL_DELETE_STATUS
- GL_COMPILE_STATUS
- GL_LINK_STATUS
- GL_VALIDATE_STATUS
- GL_INFO_LOG_LENGTH
- GL_ATTACHED_SHADERS
- GL_ACTIVE_UNIFORMS
- GL_ACTIVE_UNIFORM_MAX_LENGTH
- GL_SHADER_SOURCE_LENGTH
- GL_ACTIVE_ATTRIBUTES
- GL_ACTIVE_ATTRIBUTE_MAX_LENGTH
- GL_FRAGMENT_SHADER_DERIVATIVE_HINT
- GL_SHADING_LANGUAGE_VERSION
- GL_CURRENT_PROGRAM
- GL_POINT_SPRITE_COORD_ORIGIN
- GL_LOWER_LEFT
- GL_UPPER_LEFT
- GL_STENCIL_BACK_REF
- GL_STENCIL_BACK_VALUE_MASK
- GL_STENCIL_BACK_WRITEMASK
- GL_CURRENT_RASTER_SECONDARY_COLOR
- GL_PIXEL_PACK_BUFFER
- GL_PIXEL_UNPACK_BUFFER
- GL_PIXEL_PACK_BUFFER_BINDING
- GL_PIXEL_UNPACK_BUFFER_BINDING
- GL_FLOAT_MAT2x3
- GL_FLOAT_MAT2x4
- GL_FLOAT_MAT3x2
- GL_FLOAT_MAT3x4
- GL_FLOAT_MAT4x2
- GL_FLOAT_MAT4x3
- GL_SRGB

- GL_SRGB8
- GL_SRGB_ALPHA
- GL_SRGB8_ALPHA8
- GL_SLUMINANCE_ALPHA
- GL_SLUMINANCE8_ALPHA8
- GL_SLUMINANCE
- GL_SLUMINANCE8
- GL_COMPRESSED_SRGB
- GL_COMPRESSED_SRGB_ALPHA
- GL_COMPRESSED_SLUMINANCE
- GL_COMPRESSED_SLUMINANCE_ALPHA
- GL_CLIP_DISTANCE0
- GL_CLIP_DISTANCE1
- GL_CLIP_DISTANCE2
- GL_CLIP_DISTANCE3
- GL_CLIP_DISTANCE4
- GL_CLIP_DISTANCE5
- GL_COMPARE_REF_TO_TEXTURE
- GL_MAX_CLIP_DISTANCES
- GL_MAX_VARYING_COMPONENTS
- GL_CONTEXT_FLAG_FORWARD_COMPATIBLE_BIT
- GL_MAJOR_VERSION
- GL_MINOR_VERSION
- GL_NUM_EXTENSIONS
- GL_CONTEXT_FLAGS
- GL_DEPTH_BUFFER
- GL_STENCIL_BUFFER
- GL_RGBA32F
- GL_RGB32F
- GL_RGBA16F
- GL_RGB16F
- GL_VERTEX_ATTRIB_ARRAY_INTEGER
- GL_MAX_ARRAY_TEXTURE_LAYERS
- GL_MIN_PROGRAM_TEXEL_OFFSET
- GL_MAX_PROGRAM_TEXEL_OFFSET
- GL_CLAMP_VERTEX_COLOR

- GL_CLAMP_FRAGMENT_COLOR
- GL_CLAMP_READ_COLOR
- GL_FIXED_ONLY
- GL_TEXTURE_RED_TYPE
- GL_TEXTURE_GREEN_TYPE
- GL_TEXTURE_BLUE_TYPE
- GL_TEXTURE_ALPHA_TYPE
- GL_TEXTURE_LUMINANCE_TYPE
- GL_TEXTURE_INTENSITY_TYPE
- GL_TEXTURE_DEPTH_TYPE
- GL_TEXTURE_1D_ARRAY
- GL_PROXY_TEXTURE_1D_ARRAY
- GL_TEXTURE_2D_ARRAY
- GL_PROXY_TEXTURE_2D_ARRAY
- GL_TEXTURE_BINDING_1D_ARRAY
- GL_TEXTURE_BINDING_2D_ARRAY
- GL_R11F_G11F_B10F
- GL_UNSIGNED_INT_10F_11F_11F_REV
- GL_RGB9_E5
- GL_UNSIGNED_INT_5_9_9_9_REV
- GL_TEXTURE_SHARED_SIZE
- GL_TRANSFORM_FEEDBACK_VARYING_MAX_LENGTH
- GL_TRANSFORM_FEEDBACK_BUFFER_MODE
- GL_MAX_TRANSFORM_FEEDBACK_SEPARATE_COMPONENTS
- GL_TRANSFORM_FEEDBACK_VARYINGS
- GL_TRANSFORM_FEEDBACK_BUFFER_START
- GL_TRANSFORM_FEEDBACK_BUFFER_SIZE
- GL_PRIMITIVES_GENERATED
- GL_TRANSFORM_FEEDBACK_PRIMITIVES_WRITTEN
- GL_RASTERIZER_DISCARD
- GL_MAX_TRANSFORM_FEEDBACK_INTERLEAVED_COMPONENTS
- GL_MAX_TRANSFORM_FEEDBACK_SEPARATE_ATTRIBS
- GL_INTERLEAVED_ATTRIBS
- GL_SEPARATE_ATTRIBS
- GL_TRANSFORM_FEEDBACK_BUFFER
- GL_TRANSFORM_FEEDBACK_BUFFER_BINDING

- GL_RGBA32UI
- GL_RGB32UI
- GL_RGBA16UI
- GL_RGB16UI
- GL_RGBA8UI
- GL_RGB8UI
- GL_RGBA32I
- GL_RGB32I
- GL_RGBA16I
- GL_RGB16I
- GL_RGBA8I
- GL_RGB8I
- GL_RED_INTEGER
- GL_GREEN_INTEGER
- GL_BLUE_INTEGER
- GL_ALPHA_INTEGER
- GL_RGB_INTEGER
- GL_RGBA_INTEGER
- GL_BGR_INTEGER
- GL_BGRA_INTEGER
- GL_SAMPLER_1D_ARRAY
- GL_SAMPLER_2D_ARRAY
- GL_SAMPLER_1D_ARRAY_SHADOW
- GL_SAMPLER_2D_ARRAY_SHADOW
- GL_SAMPLER_CUBE_SHADOW
- GL_UNSIGNED_INT_VEC2
- GL_UNSIGNED_INT_VEC3
- GL_UNSIGNED_INT_VEC4
- GL_INT_SAMPLER_1D
- GL_INT_SAMPLER_2D
- GL_INT_SAMPLER_3D
- GL_INT_SAMPLER_CUBE
- GL_INT_SAMPLER_1D_ARRAY
- GL_INT_SAMPLER_2D_ARRAY
- GL_UNSIGNED_INT_SAMPLER_1D
- GL_UNSIGNED_INT_SAMPLER_2D

- GL_UNSIGNED_INT_SAMPLER_3D
- GL_UNSIGNED_INT_SAMPLER_CUBE
- GL_UNSIGNED_INT_SAMPLER_1D_ARRAY
- GL_UNSIGNED_INT_SAMPLER_2D_ARRAY
- GL_QUERY_WAIT
- GL_QUERY_NO_WAIT
- GL_QUERY_BY_REGION_WAIT
- GL_QUERY_BY_REGION_NO_WAIT
- GL_TEXTURE_RECTANGLE
- GL_TEXTURE_BINDING_RECTANGLE
- GL_PROXY_TEXTURE_RECTANGLE
- GL_MAX_RECTANGLE_TEXTURE_SIZE
- GL_SAMPLER_2D_RECT
- GL_SAMPLER_2D_RECT_SHADOW
- GL_TEXTURE_BUFFER
- GL_MAX_TEXTURE_BUFFER_SIZE
- GL_TEXTURE_BINDING_BUFFER
- GL_TEXTURE_BUFFER_DATA_STORE_BINDING
- GL_TEXTURE_BUFFER_FORMAT
- GL_SAMPLER_BUFFER
- GL_INT_SAMPLER_2D_RECT
- GL_INT_SAMPLER_BUFFER
- GL_UNSIGNED_INT_SAMPLER_2D_RECT
- GL_UNSIGNED_INT_SAMPLER_BUFFER
- GL_RED_SNORM
- GL_RG_SNORM
- GL_RGB_SNORM
- GL_RGBA_SNORM
- GL_R8_SNORM
- GL_RG8_SNORM
- GL_RGB8_SNORM
- GL_RGBA8_SNORM
- GL_R16_SNORM
- GL_RG16_SNORM
- GL_RGB16_SNORM
- GL_RGBA16_SNORM

- GL_SIGNED_NORMALIZED
- GL_PRIMITIVE_RESTART
- GL_PRIMITIVE_RESTART_INDEX
- GL_BUFFER_ACCESS_FLAGS
- GL_BUFFER_MAP_LENGTH
- GL_BUFFER_MAP_OFFSET
- GL_CONTEXT_CORE_PROFILE_BIT
- GL_CONTEXT_COMPATIBILITY_PROFILE_BIT
- GL_LINES_ADJACENCY
- GL_LINE_STRIP_ADJACENCY
- GL_TRIANGLES_ADJACENCY
- GL_TRIANGLE_STRIP_ADJACENCY
- GL_PROGRAM_POINT_SIZE
- GL_GEOMETRY_VERTICES_OUT
- GL_GEOMETRY_INPUT_TYPE
- GL_GEOMETRY_OUTPUT_TYPE
- GL_MAX_GEOMETRY_TEXTURE_IMAGE_UNITS
- GL_FRAMEBUFFER_ATTACHMENT_LAYERED
- GL_FRAMEBUFFER_INCOMPLETE_LAYER_TARGETS
- GL_GEOMETRY_SHADER
- GL_MAX_GEOMETRY_UNIFORM_COMPONENTS
- GL_MAX_GEOMETRY_OUTPUT_VERTICES
- GL_MAX_GEOMETRY_TOTAL_OUTPUT_COMPONENTS
- GL_MAX_VERTEX_OUTPUT_COMPONENTS
- GL_MAX_GEOMETRY_INPUT_COMPONENTS
- GL_MAX_GEOMETRY_OUTPUT_COMPONENTS
- GL_MAX_FRAGMENT_INPUT_COMPONENTS
- GL_CONTEXT_PROFILE_MASK
- GL_VERTEX_ATTRIB_ARRAY_DIVISOR
- GL_RGB10_A2UI
- GL_SAMPLE_SHADING
- GL_MIN_SAMPLE_SHADING_VALUE
- GL_MIN_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_COMPONENTS
- GL_TEXTURE_CUBE_MAP_ARRAY

- GL_TEXTURE_BINDING_CUBE_MAP_ARRAY
- GL_PROXY_TEXTURE_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY_SHADOW
- GL_INT_SAMPLER_CUBE_MAP_ARRAY
- GL_UNSIGNED_INT_SAMPLER_CUBE_MAP_ARRAY
- GL_TRANSFORM_FEEDBACK_PAUSED
- GL_TRANSFORM_FEEDBACK_ACTIVE
- GL_COMPRESSED_RGBA_BPTC_UNORM
- GL_COMPRESSED_SRGB_ALPHA_BPTC_UNORM
- GL_COMPRESSED_RGB_BPTC_SIGNED_FLOAT
- GL_COMPRESSED_RGB_BPTC_UNSIGNED_FLOAT
- GL_COPY_READ_BUFFER_BINDING
- GL_COPY_WRITE_BUFFER_BINDING
- void glAccum(GLenum op, GLfloat value)
- void glActiveTexture(GLenum texture)
- void glAlphaFunc(GLenum func, GLclampf ref)
- GLboolean glAreTexturesResident(GLsizei n, const GLuint * textures, GLboolean * residences)
- void glArrayElement(GLint i)
- void glAttachShader(GLuint program, GLuint shader)
- void glBegin(GLenum mode)
- void glBeginQuery(GLenum target, GLuint id)
- void glBindAttribLocation(GLuint program, GLuint index, const GLchar * name)
- void glBindBuffer(GLenum target, GLuint buffer)
- void glBindTexture(GLenum target, GLuint texture)
- void glBitmap(GLsizei width, GLsizei height, GLfloat xorig, GLfloat yorig, GLfloat xmove, GLfloat ymove, const GLubyte * bitmap)
- void glBlendColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)
- void glBlendEquation(GLenum mode)
- void glBlendEquationSeparate(GLenum modeRGB, GLenum modeAlpha)
- void glBlendFunc(GLenum sfactor, GLenum dfactor)
- void glBlendFuncSeparate(GLenum srcRGB, GLenum dstRGB, GLenum srcAlpha, GLenum dstAlpha)
- void glBufferData(GLenum target, GLsizeiptr size, const GLvoid * data, GLenum usage)
- void glBufferSubData(GLenum target, GLintptr offset, GLsizeiptr size, const GLvoid * data)
- void glCallList(GLuint list)
- void glCallLists(GLsizei n, GLenum type, const GLvoid * lists)

- `void glClear(GLbitfield mask)`
- `void glClearAccum(GLfloat red,GLfloat green,GLfloat blue,GLfloat alpha)`
- `void glClearColor(GLclampf red,GLclampf green,GLclampf blue,GLclampf alpha)`
- `void glClearDepth(GLclampd depth)`
- `void glClearIndex(GLfloat c)`
- `void glClearStencil(GLint s)`
- `void glClientActiveTexture(GLenum texture)`
- `void glClipPlane(GLenum plane,const GLdouble * equation)`
- `void glColor3b(GLbyte red,GLbyte green,GLbyte blue)`
- `void glColor3s(GLshort red,GLshort green,GLshort blue)`
- `void glColor3i(GLint red,GLint green,GLint blue)`
- `void glColor3f(GLfloat red,GLfloat green,GLfloat blue)`
- `void glColor3d(GLdouble red,GLdouble green,GLdouble blue)`
- `void glColor3ub(GLubyte red,GLubyte green,GLubyte blue)`
- `void glColor3us(GLushort red,GLushort green,GLushort blue)`
- `void glColor3ui(GLuint red,GLuint green,GLuint blue)`
- `void glColor4b(GLbyte red,GLbyte green,GLbyte blue,GLbyte alpha)`
- `void glColor4s(GLshort red,GLshort green,GLshort blue,GLshort alpha)`
- `void glColor4i(GLint red,GLint green,GLint blue,GLint alpha)`
- `void glColor4f(GLfloat red,GLfloat green,GLfloat blue,GLfloat alpha)`
- `void glColor4d(GLdouble red,GLdouble green,GLdouble blue,GLdouble alpha)`
- `void glColor4ub(GLubyte red,GLubyte green,GLubyte blue,GLubyte alpha)`
- `void glColor4us(GLushort red,GLushort green,GLushort blue,GLushort alpha)`
- `void glColor4ui(GLuint red,GLuint green,GLuint blue,GLuint alpha)`
- `void glColor3bv(const GLbyte * v)`
- `void glColor3sv(const GLshort * v)`
- `void glColor3iv(const GLint * v)`
- `void glColor3fv(const GLfloat * v)`
- `void glColor3dv(const GLdouble * v)`
- `void glColor3ubv(const GLubyte * v)`
- `void glColor3usv(const GLushort * v)`
- `void glColor3uiv(const GLuint * v)`
- `void glColor4bv(const GLbyte * v)`
- `void glColor4sv(const GLshort * v)`
- `void glColor4iv(const GLint * v)`
- `void glColor4fv(const GLfloat * v)`

- `void glColor4dv(const GLdouble * v)`
- `void glColor4ubv(const GLubyte * v)`
- `void glColor4usv(const GLushort * v)`
- `void glColor4uiv(const GLuint * v)`
- `void glColorMask(GLboolean red, GLboolean green, GLboolean blue, GLboolean alpha)`
- `void glColorMaterial(GLenum face, GLenum mode)`
- `void glColorPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)`
- `void glColorSubTable(GLenum target, GLsizei start, GLsizei count, GLenum format, GLenum type, const GLvoid * data)`
- `void glColorTable(GLenum target, GLenum internalformat, GLsizei width, GLenum format, GLenum type, const GLvoid * data)`
- `void glColorTableParameterfv(GLenum target, GLenum pname, const GLfloat * params)`
- `void glColorTableParameteriv(GLenum target, GLenum pname, const GLint * params)`
- `void glCompileShader(GLuint shader)`
- `void glCompressedTexImage1D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLint border, GLsizei imageSize, const GLvoid * data)`
- `void glCompressedTexImage2D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLint border, GLsizei imageSize, const GLvoid * data)`
- `void glCompressedTexImage3D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLsizei depth, GLint border, GLsizei imageSize, const GLvoid * data)`
- `void glCompressedTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLsizei imageSize, const GLvoid * data)`
- `void glCompressedTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLsizei imageSize, const GLvoid * data)`
- `void glCompressedTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLsizei imageSize, const GLvoid * data)`
- `void glConvolutionFilter1D(GLenum target, GLenum internalformat, GLsizei width, GLenum format, GLenum type, const GLvoid * data)`
- `void glConvolutionFilter2D(GLenum target, GLenum internalformat, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)`
- `void glConvolutionParameterf(GLenum target, GLenum pname, GLfloat params)`
- `void glConvolutionParameteri(GLenum target, GLenum pname, GLint params)`
- `void glConvolutionParameterfv(GLenum target, GLenum pname, const GLfloat * params)`
- `void glConvolutionParameteriv(GLenum target, GLenum pname, const GLint * params)`
- `void glCopyColorSubTable(GLenum target, GLsizei start, GLint x, GLint y, GLsizei width)`
- `void glCopyColorTable(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width)`
- `void glCopyConvolutionFilter1D(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width)`
- `void glCopyConvolutionFilter2D(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width, GLsizei height)`
- `void glCopyPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum type)`

- void glCopyTexImage1D(GLenum target,GLint level,GLenum internalformat,GLint x,GLint y,GLsizei width,GLint border)
- void glCopyTexImage2D(GLenum target,GLint level,GLenum internalformat,GLint x,GLint y,GLsizei width,GLsizei height,GLint border)
- void glCopyTexSubImage1D(GLenum target,GLint level,GLint xoffset,GLint x,GLint y,GLsizei width)
- void glCopyTexSubImage2D(GLenum target,GLint level,GLint xoffset,GLint yoffset,GLint x,GLint y,GLsizei width,GLsizei height)
- void glCopyTexSubImage3D(GLenum target,GLint level,GLint xoffset,GLint yoffset,GLint zoffset,GLint x,GLint y,GLsizei width,GLsizei height)
- GLuint glCreateProgram(void)
- GLuint glCreateShader(GLenum shaderType)
- void glCullFace(GLenum mode)
- void glDeleteBuffers(GLsizei n,const GLuint * buffers)
- void glDeleteLists(GLuint list,GLsizei range)
- void glDeleteProgram(GLuint program)
- void glDeleteQueries(GLsizei n,const GLuint * ids)
- void glDeleteShader(GLuint shader)
- void glDeleteTextures(GLsizei n,const GLuint * textures)
- void glDepthFunc(GLenum func)
- void glDepthMask(GLboolean flag)
- void glDepthRange(GLclampd nearVal,GLclampd farVal)
- void glDetachShader(GLuint program,GLuint shader)
- void glEnable(GLenum cap)
- void glEnableClientState(GLenum cap)
- void glEnableVertexAttribArray(GLuint index)
- void glDisableVertexAttribArray(GLuint index)
- void glDrawArrays(GLenum mode,GLint first,GLsizei count)
- void glDrawBuffer(GLenum mode)
- void glDrawBuffers(GLsizei n,const GLenum *bufs)
- void glDrawElements(GLenum mode,GLsizei count,GLenum type,const GLvoid * indices)
- void glDrawPixels(GLsizei width,GLsizei height,GLenum format,GLenum type,const GLvoid * data)
- void glDrawRangeElements(GLenum mode,GLuint start,GLuint end,GLsizei count,GLenum type,const GLvoid * indices)
- void glEdgeFlag(GLboolean flag)
- void glEdgeFlagPointer(GLsizei stride,const GLvoid * pointer)
- void glEnd(void)
- void glEndList(void)
- void glEndQuery(GLenum target)

- void glEvalCoord1f(GLfloat u)
- void glEvalCoord1d(GLdouble u)
- void glEvalCoord2f(GLfloat u,GLfloat v)
- void glEvalCoord2d(GLdouble u,GLdouble v)
- void glEvalMesh1(GLenum mode,GLint i1,GLint i2)
- void glEvalPoint1(GLint i)
- void glEvalPoint2(GLint i,GLint j)
- void glFeedbackBuffer(GLsizei size,GLenum type,GLfloat * buffer)
- void glFinish(void)
- void glFlush(void)
- void glFogf(GLenum pname,GLfloat param)
- void glFogi(GLenum pname,GLint param)
- void glFogfv(GLenum pname,const GLfloat * params)
- void glFogiv(GLenum pname,const GLint * params)
- void glFogCoordd(GLdouble coord)
- void glFogCoordf(GLfloat coord)
- void glFogCoorddv(GLdouble * coord)
- void glFogCoordfv(GLfloat * coord)
- void glFogCoordPointer(GLenum type,GLsizei stride,GLvoid * pointer)
- void glFrontFace(GLenum mode)
- void glFrustum(GLdouble left,GLdouble right,GLdouble bottom,GLdouble top,GLdouble nearVal,GLdouble farVal)
- void glGenBuffers(GLsizei n,GLuint * buffers)
- GLuint glGenLists(GLsizei range)
- void glGenQueries(GLsizei n,GLuint * ids)
- void glGenTextures(GLsizei n,GLuint * textures)
- void glGetBooleany(GLenum pname,GLboolean * params)
- void glGetDoublev(GLenum pname,GLdouble * params)
- void glGetFloatv(GLenum pname,GLfloat * params)
- void glGetIntegerv(GLenum pname,GLint * params)
- void glGetActiveAttrib(GLuint program,GLuint index,GLsizei bufSize,GLsizei *length,GLint *size,GLenum *type,GLchar *name)
- void glGetActiveUniform(GLuint program,GLuint index,GLsizei bufSize,GLsizei *length,GLint *size,GLenum *type,GLchar *name)
- void glGetAttachedShaders(GLuint program,GLsizei maxCount,GLsizei *count,GLuint *shaders)
- GLint glGetAttribLocation(GLuint program,const GLchar *name)
- void glGetBufferParameteriv(GLenum target,GLenum value,GLint * data)

- void glGetBufferPointerv(GLenum target, GLenum pname, GLvoid ** params)
- void glGetBufferSubData(GLenum target, GLintptr offset, GLsizeiptr size, GLvoid * data)
- void glGetClipPlane(GLenum plane, GLdouble * equation)
- void glGetColorTable(GLenum target, GLenum format, GLenum type, GLvoid * table)
- void glGetColorTableParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetColorTableParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetCompressedTexImage(GLenum target, GLint lod, GLvoid * img)
- void glGetConvolutionFilter(GLenum target, GLenum format, GLenum type, GLvoid * image)
- void glGetConvolutionParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetConvolutionParameteriv(GLenum target, GLenum pname, GLint * params)
- GLenum glGetError(void)
- void glGetHistogram(GLenum target, GLboolean reset, GLenum format, GLenum type, GLvoid * values)
- void glGetHistogramParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetHistogramParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetLightfv(GLenum light, GLenum pname, GLfloat * params)
- void glGetLightiv(GLenum light, GLenum pname, GLint * params)
- void glGetMapdv(GLenum target, GLenum query, GLdouble * v)
- void glGetMapfv(GLenum target, GLenum query, GLfloat * v)
- void glGetMapiv(GLenum target, GLenum query, GLint * v)
- void glGetMaterialfv(GLenum face, GLenum pname, GLfloat * params)
- void glGetMaterialiv(GLenum face, GLenum pname, GLint * params)
- void glGetMinmax(GLenum target, GLboolean reset, GLenum format, GLenum types, GLvoid * values)
- void glGetMinmaxParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetMinmaxParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetPixelMapfv(GLenum map, GLfloat * data)
- void glGetPixelMapuiv(GLenum map, GLuint * data)
- void glGetPixelMapusv(GLenum map, GLushort * data)
- void glGetPointerv(GLenum pname, GLvoid ** params)
- void glGetPolygonStipple(GLubyte * pattern)
- void glGetProgramiv(GLuint program, GLenum pname, GLint *params)
- void glGetProgramInfoLog(GLuint program, GLsizei maxLength, GLsizei *length, GLchar *infoLog)
- void glGetQueryObjectiv(GLuint id, GLenum pname, GLint * params)
- void glGetQueryObjectuiv(GLuint id, GLenum pname, GLuint * params)
- void glGetQueryiv(GLenum target, GLenum pname, GLint * params)
- void glGetSeparableFilter(GLenum target, GLenum format, GLenum type, GLvoid * row, GLvoid * column, GLvoid * span)

- void glGetShaderiv(GLuint shader, GLenum pname, GLint *params)
- void glGetShaderInfoLog(GLuint shader, GLsizei maxLength, GLsizei *length, GLchar *infoLog)
- void glGetShaderSource(GLuint shader, GLsizei bufSize, GLsizei *length, GLchar *source)
- const GLubyte* glGetString(GLenum name)
- void glGetTexEnvfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetTexEnviv(GLenum target, GLenum pname, GLint * params)
- void glGetTexGendv(GLenum coord, GLenum pname, GLdouble * params)
- void glGetTexGenfv(GLenum coord, GLenum pname, GLfloat * params)
- void glGetTexGeniv(GLenum coord, GLenum pname, GLint * params)
- void glGetTexImage(GLenum target, GLint level, GLenum format, GLenum type, GLvoid * img)
- void glGetTexLevelParameterfv(GLenum target, GLint level, GLenum pname, GLfloat * params)
- void glGetTexLevelParameteriv(GLenum target, GLint level, GLenum pname, GLint * params)
- void glGetTexParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetTexParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetUniformfv(GLuint program, GLint location, GLfloat *params)
- void glGetUniformiv(GLuint program, GLint location, GLint *params)
- GLint glGetUniformLocation(GLuint program, const GLchar *name)
- void glGetVertexAttribdv(GLuint index, GLenum pname, GLdouble *params)
- void glGetVertexAttribfv(GLuint index, GLenum pname, GLfloat *params)
- void glGetVertexAttribiv(GLuint index, GLenum pname, GLint *params)
- void glGetVertexAttribPointerv(GLuint index, GLenum pname, GLvoid **pointer)
- void glHint(GLenum target, GLenum mode)
- void glHistogram(GLenum target, GLsizei width, GLenum internalformat, GLboolean sink)
- void glIndexs(GLshort c)
- void glIndexi(GLint c)
- void glIndexf(GLfloat c)
- void glIndexd(GLdouble c)
- void glIndexub(GLubyte c)
- void glIndexsv(const GLshort * c)
- void glIndexiv(const GLint * c)
- void glIndexfv(const GLfloat * c)
- void glIndexdv(const GLdouble * c)
- void glIndexubv(const GLubyte * c)
- void glIndexMask(GLuint mask)
- void glIndexPointer(GLenum type, GLsizei stride, const GLvoid * pointer)
- void glInitNames(void)

- `void glInterleavedArrays(GLenum format, GLsizei stride, const GLvoid * pointer)`
- `GLboolean glIsBuffer(GLuint buffer)`
- `GLboolean glIsEnabled(GLenum cap)`
- `GLboolean glIsList(GLuint list)`
- `GLboolean glIsProgram(GLuint program)`
- `GLboolean glIsQuery(GLuint id)`
- `GLboolean glIsShader(GLuint shader)`
- `GLboolean glIsTexture(GLuint texture)`
- `void glLightf(GLenum light, GLenum pname, GLfloat param)`
- `void glLighti(GLenum light, GLenum pname, GLint param)`
- `void glLightfv(GLenum light, GLenum pname, const GLfloat * params)`
- `void glLightiv(GLenum light, GLenum pname, const GLint * params)`
- `void glLightModelf(GLenum pname, GLfloat param)`
- `void glLightModeli(GLenum pname, GLint param)`
- `void glLightModelfv(GLenum pname, const GLfloat * params)`
- `void glLightModeliv(GLenum pname, const GLint * params)`
- `void glLineStipple(GLint factor, GLushort pattern)`
- `void glLineWidth(GLfloat width)`
- `void glLinkProgram(GLuint program)`
- `void glListBase(GLuint base)`
- `void glLoadIdentity(void)`
- `void glLoadMatrixd(const GLdouble * m)`
- `void glLoadMatrixf(const GLfloat * m)`
- `void glLoadName(GLuint name)`
- `void glLoadTransposeMatrixd(const GLdouble * m)`
- `void glLoadTransposeMatrixf(const GLfloat * m)`
- `void glLogicOp(GLenum opcode)`
- `void glMap1f(GLenum target, GLfloat u1, GLfloat u2, GLint stride, GLint order, const GLfloat * points)`
- `void glMap1d(GLenum target, GLdouble u1, GLdouble u2, GLint stride, GLint order, const GLdouble * points)`
- `void glMap2f(GLenum target, GLfloat u1, GLfloat u2, GLint ustride, GLint uorder, GLfloat v1, GLfloat v2, GLint vstride, GLint vorder, const GLfloat * points)`
- `void glMap2d(GLenum target, GLdouble u1, GLdouble u2, GLint ustride, GLint uorder, GLdouble v1, GLdouble v2, GLint vstride, GLint vorder, const GLdouble * points)`
- `void * glMapBuffer(GLenum target, GLenum access)`
- `void glMapGrid1d(GLint un, GLdouble u1, GLdouble u2)`
- `void glMapGrid1f(GLint un, GLfloat u1, GLfloat u2)`
- `void glMapGrid2d(GLint un, GLdouble u1, GLdouble u2, GLint vn, GLdouble v1, GLdouble v2)`

- void glMapGrid2f(GLint un,GLfloat u1,GLfloat u2,GLint vn,GLfloat v1,GLfloat v2)
- void glMaterialf(GLenum face,GLenum pname,GLfloat param)
- void glMateriali(GLenum face,GLenum pname,GLint param)
- void glMatrixMode(GLenum mode)
- void glMinmax(GLenum target,GLenum internalformat,GLboolean sink)
- void glMultMatrixd(const GLdouble * m)
- void glMultMatrixf(const GLfloat * m)
- void glMultTransposeMatrixd(const GLdouble * m)
- void glMultTransposeMatrixf(const GLfloat * m)
- void glMultiDrawArrays(GLenum mode,GLint * first,GLsizei * count,GLsizei primcount)
- void glMultiDrawElements(GLenum mode,const GLsizei * count,GLenum type,const GLvoid ** indices,GLsizei primcount)
- void glMultiTexCoord1s(GLenum target,GLshort s)
- void glMultiTexCoord1i(GLenum target,GLint s)
- void glMultiTexCoord1f(GLenum target,GLfloat s)
- void glMultiTexCoord1d(GLenum target,GLdouble s)
- void glMultiTexCoord2s(GLenum target,GLshort s,GLshort t)
- void glMultiTexCoord2i(GLenum target,GLint s,GLint t)
- void glMultiTexCoord2f(GLenum target,GLfloat s,GLfloat t)
- void glMultiTexCoord2d(GLenum target,GLdouble s,GLdouble t)
- void glMultiTexCoord3s(GLenum target,GLshort s,GLshort t,GLshort r)
- void glMultiTexCoord3i(GLenum target,GLint s,GLint t,GLint r)
- void glMultiTexCoord3f(GLenum target,GLfloat s,GLfloat t,GLfloat r)
- void glMultiTexCoord3d(GLenum target,GLdouble s,GLdouble t,GLdouble r)
- void glMultiTexCoord4s(GLenum target,GLshort s,GLshort t,GLshort r,GLshort q)
- void glMultiTexCoord4i(GLenum target,GLint s,GLint t,GLint r,GLint q)
- void glMultiTexCoord4f(GLenum target,GLfloat s,GLfloat t,GLfloat r,GLfloat q)
- void glMultiTexCoord4d(GLenum target,GLdouble s,GLdouble t,GLdouble r,GLdouble q)
- void glMultiTexCoord1sv(GLenum target,const GLshort * v)
- void glMultiTexCoord1iv(GLenum target,const GLint * v)
- void glMultiTexCoord1fv(GLenum target,const GLfloat * v)
- void glMultiTexCoord1dv(GLenum target,const GLdouble * v)
- void glMultiTexCoord2sv(GLenum target,const GLshort * v)
- void glMultiTexCoord2iv(GLenum target,const GLint * v)
- void glMultiTexCoord2fv(GLenum target,const GLfloat * v)
- void glMultiTexCoord2dv(GLenum target,const GLdouble * v)

- `void glMultiTexCoord3sv(GLenum target,const GLshort * v)`
- `void glMultiTexCoord3iv(GLenum target,const GLint * v)`
- `void glMultiTexCoord3fv(GLenum target,const GLfloat * v)`
- `void glMultiTexCoord3dv(GLenum target,const GLdouble * v)`
- `void glMultiTexCoord4sv(GLenum target,const GLshort * v)`
- `void glMultiTexCoord4iv(GLenum target,const GLint * v)`
- `void glMultiTexCoord4fv(GLenum target,const GLfloat * v)`
- `void glMultiTexCoord4dv(GLenum target,const GLdouble * v)`
- `void glNewList(GLuint list,GLenum mode)`
- `void glNormal3b(GLbyte nx,GLbyte ny,GLbyte nz)`
- `void glNormal3d(GLdouble nx,GLdouble ny,GLdouble nz)`
- `void glNormal3f(GLfloat nx,GLfloat ny,GLfloat nz)`
- `void glNormal3i(GLint nx,GLint ny,GLint nz)`
- `void glNormal3s(GLshort nx,GLshort ny,GLshort nz)`
- `void glNormal3bv(const GLbyte * v)`
- `void glNormal3dv(const GLdouble * v)`
- `void glNormal3fv(const GLfloat * v)`
- `void glNormal3iv(const GLint * v)`
- `void glNormal3sv(const GLshort * v)`
- `void glNormalPointer(GLenum type,GLsizei stride,const GLvoid * pointer)`
- `void glOrtho(GLdouble left,GLdouble right,GLdouble bottom,GLdouble top,GLdouble nearVal,GLdouble farVal)`
- `void glPassThrough(GLfloat token)`
- `void glPixelMapfv(GLenum map,GLsizei mapsize,const GLfloat * values)`
- `void glPixelMapuiv(GLenum map,GLsizei mapsize,const GLuint * values)`
- `void glPixelMapusv(GLenum map,GLsizei mapsize,const GLushort * values)`
- `void glPixelStoref(GLenum pname,GLfloat param)`
- `void glPixelStorei(GLenum pname,GLint param)`
- `void glPixelTransferf(GLenum pname,GLfloat param)`
- `void glPixelTransferi(GLenum pname,GLint param)`
- `void glPixelZoom(GLfloat xfactor,GLfloat yfactor)`
- `void glPointParameterf(GLenum pname,GLfloat param)`
- `void glPointParameteri(GLenum pname,GLint param)`
- `void glPointSize(GLfloat size)`
- `void glPolygonMode(GLenum face,GLenum mode)`
- `void glPolygonOffset(GLfloat factor,GLfloat units)`

- void glPolygonStipple(const GLubyte * pattern)
- void glPushAttrib(GLbitfield mask)
- void glPushClientAttrib(GLbitfield mask)
- void glPushMatrix(void)
- void glPushName(GLuint name)
- void glPrioritizeTextures(GLsizei n,const GLuint * textures,const GLclampf * priorities)
- void glPopMatrix(void)
- void glRasterPos2s(GLshort x,GLshort y)
- void glRasterPos2i(GLint x,GLint y)
- void glRasterPos2f(GLfloat x,GLfloat y)
- void glRasterPos2d(GLdouble x,GLdouble y)
- void glRasterPos3s(GLshort x,GLshort y,GLshort z)
- void glRasterPos3i(GLint x,GLint y,GLint z)
- void glRasterPos3f(GLfloat x,GLfloat y,GLfloat z)
- void glRasterPos3d(GLdouble x,GLdouble y,GLdouble z)
- void glRasterPos4s(GLshort x,GLshort y,GLshort z,GLshort w)
- void glRasterPos4i(GLint x,GLint y,GLint z,GLint w)
- void glRasterPos4f(GLfloat x,GLfloat y,GLfloat z,GLfloat w)
- void glRasterPos4d(GLdouble x,GLdouble y,GLdouble z,GLdouble w)
- void glReadBuffer(GLenum mode)
- void glReadPixels(GLint x,GLint y,GLsizei width,GLsizei height,GLenum format,GLenum type,GLvoid * data)
- void glRectd(GLdouble x1,GLdouble y1,GLdouble x2,GLdouble y2)
- void glRectf(GLfloat x1,GLfloat y1,GLfloat x2,GLfloat y2)
- void glRecti(GLint x1,GLint y1,GLint x2,GLint y2)
- void glRects(GLshort x1,GLshort y1,GLshort x2,GLshort y2)
- void glRectdv(const GLdouble * v1,const GLdouble * v2)
- void glRectfv(const GLfloat * v1,const GLfloat * v2)
- void glRectiv(const GLint * v1,const GLint * v2)
- void glRectsv(const GLshort * v1,const GLshort * v2)
- GLint glRenderMode(GLenum mode)
- void glResetHistogram(GLenum target)
- void glRotated(GLdouble angle,GLdouble x,GLdouble y,GLdouble z)
- void glRotatef(GLfloat angle,GLfloat x,GLfloat y,GLfloat z)
- void glSampleCoverage(GLclampf value,GLboolean invert)
- void glScaled(GLdouble x,GLdouble y,GLdouble z)
- void glScalef(GLfloat x,GLfloat y,GLfloat z)

- void glScissor(GLint x,GLint y,GLsizei width,GLsizei height)
- void glSecondaryColor3b(GLbyte red,GLbyte green,GLbyte blue)
- void glSecondaryColor3s(GLshort red,GLshort green,GLshort blue)
- void glSecondaryColor3i(GLint red,GLint green,GLint blue)
- void glSecondaryColor3f(GLfloat red,GLfloat green,GLfloat blue)
- void glSecondaryColor3d(GLdouble red,GLdouble green,GLdouble blue)
- void glSecondaryColor3ub(GLubyte red,GLubyte green,GLubyte blue)
- void glSecondaryColor3us(GLushort red,GLushort green,GLushort blue)
- void glSecondaryColor3ui(GLuint red,GLuint green,GLuint blue)
- void glSecondaryColor3bv(const GLbyte * v)
- void glSecondaryColor3sv(const GLshort * v)
- void glSecondaryColor3iv(const GLint * v)
- void glSecondaryColor3fv(const GLfloat * v)
- void glSecondaryColor3dv(const GLdouble * v)
- void glSecondaryColor3ubv(const GLubyte * v)
- void glSecondaryColor3usv(const GLushort * v)
- void glSecondaryColor3uiv(const GLuint * v)
- void glSecondaryColorPointer(GLint size,GLenum type,GLsizei stride,const GLvoid * pointer)
- void glSelectBuffer(GLsizei size,GLuint * buffer)
- void glSeparableFilter2D(GLenum target,GLenum internalformat,GLsizei width,GLsizei height,GLenum format,GLenum type,const GLvoid * row,const GLvoid * column)
- void glShadeModel(GLenum mode)
- void glShaderSource(GLuint shader,GLsizei count,const GLchar **string,const GLint *length)
- void glStencilFunc(GLenum func,GLint ref,GLuint mask)
- void glStencilFuncSeparate(GLenum face,GLenum func,GLint ref,GLuint mask)
- void glStencilMask(GLuint mask)
- void glStencilMaskSeparate(GLenum face,GLuint mask)
- void glStencilOp(GLenum sfail,GLenum dpfail,GLenum dppass)
- void glStencilOpSeparate(GLenum face,GLenum sfail,GLenum dpfail,GLenum dppass)
- void glTexCoord1s(GLshort s)
- void glTexCoord1i(GLint s)
- void glTexCoord1f(GLfloat s)
- void glTexCoord1d(GLdouble s)
- void glTexCoord2s(GLshort s,GLshort t)
- void glTexCoord2i(GLint s,GLint t)
- void glTexCoord2f(GLfloat s,GLfloat t)

- `void glTexCoord2d(GLdouble s, GLdouble t)`
- `void glTexCoord3s(GLshort s, GLshort t, GLshort r)`
- `void glTexCoord3i(GLint s, GLint t, GLint r)`
- `void glTexCoord3f(GLfloat s, GLfloat t, GLfloat r)`
- `void glTexCoord3d(GLdouble s, GLdouble t, GLdouble r)`
- `void glTexCoord4s(GLshort s, GLshort t, GLshort r, GLshort q)`
- `void glTexCoord4i(GLint s, GLint t, GLint r, GLint q)`
- `void glTexCoord4f(GLfloat s, GLfloat t, GLfloat r, GLfloat q)`
- `void glTexCoord4d(GLdouble s, GLdouble t, GLdouble r, GLdouble q)`
- `void glTexCoord1sv(const GLshort * v)`
- `void glTexCoord1iv(const GLint * v)`
- `void glTexCoord1fv(const GLfloat * v)`
- `void glTexCoord1dv(const GLdouble * v)`
- `void glTexCoord2sv(const GLshort * v)`
- `void glTexCoord2iv(const GLint * v)`
- `void glTexCoord2fv(const GLfloat * v)`
- `void glTexCoord2dv(const GLdouble * v)`
- `void glTexCoord3sv(const GLshort * v)`
- `void glTexCoord3iv(const GLint * v)`
- `void glTexCoord3fv(const GLfloat * v)`
- `void glTexCoord3dv(const GLdouble * v)`
- `void glTexCoord4sv(const GLshort * v)`
- `void glTexCoord4iv(const GLint * v)`
- `void glTexCoord4fv(const GLfloat * v)`
- `void glTexCoord4dv(const GLdouble * v)`
- `void glTexCoordPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)`
- `void glTexEnvf(GLenum target, GLenum pname, GLfloat param)`
- `void glTexEnvf(GLenum target, GLenum pname, GLint param)`
- `void glTexGeni(GLenum coord, GLenum pname, GLint param)`
- `void glTexGenf(GLenum coord, GLenum pname, GLfloat param)`
- `void glTexGend(GLenum coord, GLenum pname, GLdouble param)`
- `void glTexGeniv(GLenum coord, GLenum pname, const GLint * params)`
- `void glTexGenfv(GLenum coord, GLenum pname, const GLfloat * params)`
- `void glTexGendv(GLenum coord, GLenum pname, const GLdouble * params)`
- `void glTexImage1D(GLenum target, GLint level, GLint internalFormat, GLsizei width, GLint border, GLenum format, GLenum type, const GLvoid * data)`

- void glTexImage2D(GLenum target, GLint level, GLint internalFormat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, const GLvoid * data)
- void glTexImage3D(GLenum target, GLint level, GLint internalFormat, GLsizei width, GLsizei height, GLsizei depth, GLint border, GLenum format, GLenum type, const GLvoid * data)
- void glTexParameterf(GLenum target, GLenum pname, GLfloat param)
- void glTexParameteri(GLenum target, GLenum pname, GLint param)
- void glTexParameterfv(GLenum target, GLenum pname, const GLfloat * params)
- void glTexParameteriv(GLenum target, GLenum pname, const GLint * params)
- void glTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLenum type, const GLvoid * data)
- void glTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)
- void glTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLenum type, const GLvoid * data)
- void glTranslated(GLdouble x, GLdouble y, GLdouble z)
- void glTranslatef(GLfloat x, GLfloat y, GLfloat z)
- void glUniform1f(GLint location, GLfloat v0)
- void glUniform2f(GLint location, GLfloat v0, GLfloat v1)
- void glUniform3f(GLint location, GLfloat v0, GLfloat v1, GLfloat v2)
- void glUniform4f(GLint location, GLfloat v0, GLfloat v1, GLfloat v2, GLfloat v3)
- void glUniform1i(GLint location, GLint v0)
- void glUniform2i(GLint location, GLint v0, GLint v1)
- void glUniform3i(GLint location, GLint v0, GLint v1, GLint v2)
- void glUniform4i(GLint location, GLint v0, GLint v1, GLint v2, GLint v3)
- void glUniform1fv(GLint location, GLsizei count, const GLfloat *value)
- void glUniform2fv(GLint location, GLsizei count, const GLfloat *value)
- void glUniform3fv(GLint location, GLsizei count, const GLfloat *value)
- void glUniform4fv(GLint location, GLsizei count, const GLfloat *value)
- void glUniform1iv(GLint location, GLsizei count, const GLint *value)
- void glUniform2iv(GLint location, GLsizei count, const GLint *value)
- void glUniform3iv(GLint location, GLsizei count, const GLint *value)
- void glUniform4iv(GLint location, GLsizei count, const GLint *value)
- void glUniformMatrix2fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)
- void glUniformMatrix3fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)
- void glUniformMatrix4fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)
- void glUniformMatrix2x3fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)
- void glUniformMatrix3x2fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)
- void glUniformMatrix2x4fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)

- `void glUniformMatrix4x2fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix3x4fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix4x3fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUseProgram(GLuint program)`
- `void glValidateProgram(GLuint program)`
- `void glVertex2s(GLshort x, GLshort y)`
- `void glVertex2i(GLint x, GLint y)`
- `void glVertex2f(GLfloat x, GLfloat y)`
- `void glVertex2d(GLdouble x, GLdouble y)`
- `void glVertex3s(GLshort x, GLshort y, GLshort z)`
- `void glVertex3i(GLint x, GLint y, GLint z)`
- `void glVertex3f(GLfloat x, GLfloat y, GLfloat z)`
- `void glVertex3d(GLdouble x, GLdouble y, GLdouble z)`
- `void glVertex4s(GLshort x, GLshort y, GLshort z, GLshort w)`
- `void glVertex4i(GLint x, GLint y, GLint z, GLint w)`
- `void glVertex4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)`
- `void glVertex4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)`
- `void glVertex2sv(const GLshort * v)`
- `void glVertex2iv(const GLint * v)`
- `void glVertex2fv(const GLfloat * v)`
- `void glVertex2dv(const GLdouble * v)`
- `void glVertex3sv(const GLshort * v)`
- `void glVertex3iv(const GLint * v)`
- `void glVertex3fv(const GLfloat * v)`
- `void glVertex3dv(const GLdouble * v)`
- `void glVertex4sv(const GLshort * v)`
- `void glVertex4iv(const GLint * v)`
- `void glVertex4fv(const GLfloat * v)`
- `void glVertex4dv(const GLdouble * v)`
- `void glVertexAttrib1f(GLuint index, GLfloat v0)`
- `void glVertexAttrib1s(GLuint index, GLshort v0)`
- `void glVertexAttrib1d(GLuint index, GLdouble v0)`
- `void glVertexAttrib2f(GLuint index, GLfloat v0, GLfloat v1)`
- `void glVertexAttrib2s(GLuint index, GLshort v0, GLshort v1)`
- `void glVertexAttrib2d(GLuint index, GLdouble v0, GLdouble v1)`
- `void glVertexAttrib3f(GLuint index, GLfloat v0, GLfloat v1, GLfloat v2)`

- `void glVertexAttrib3s(GLuint index, GLshort v0, GLshort v1, GLshort v2)`
- `void glVertexAttrib3d(GLuint index, GLdouble v0, GLdouble v1, GLdouble v2)`
- `void glVertexAttrib4f(GLuint index, GLfloat v0, GLfloat v1, GLfloat v2, GLfloat v3)`
- `void glVertexAttrib4s(GLuint index, GLshort v0, GLshort v1, GLshort v2, GLshort v3)`
- `void glVertexAttrib4d(GLuint index, GLdouble v0, GLdouble v1, GLdouble v2, GLdouble v3)`
- `void glVertexAttrib4Nub(GLuint index, GLubyte v0, GLubyte v1, GLubyte v2, GLubyte v3)`
- `void glVertexAttrib1fv(GLuint index, const GLfloat *v)`
- `void glVertexAttrib1sv(GLuint index, const GLshort *v)`
- `void glVertexAttrib1dv(GLuint index, const GLdouble *v)`
- `void glVertexAttrib2fv(GLuint index, const GLfloat *v)`
- `void glVertexAttrib2sv(GLuint index, const GLshort *v)`
- `void glVertexAttrib2dv(GLuint index, const GLdouble *v)`
- `void glVertexAttrib3fv(GLuint index, const GLfloat *v)`
- `void glVertexAttrib3sv(GLuint index, const GLshort *v)`
- `void glVertexAttrib3dv(GLuint index, const GLdouble *v)`
- `void glVertexAttrib4fv(GLuint index, const GLfloat *v)`
- `void glVertexAttrib4sv(GLuint index, const GLshort *v)`
- `void glVertexAttrib4dv(GLuint index, const GLdouble *v)`
- `void glVertexAttrib4iv(GLuint index, const GLint *v)`
- `void glVertexAttrib4bv(GLuint index, const GLbyte *v)`
- `void glVertexAttrib4ubv(GLuint index, const GLubyte *v)`
- `void glVertexAttrib4usv(GLuint index, const GLushort *v)`
- `void glVertexAttrib4uiv(GLuint index, const GLuint *v)`
- `void glVertexAttribPointer(GLuint index, GLint size, GLenum type, GLboolean normalized, GLsizei stride, const GLvoid * pointer)`
- `void glVertexPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)`
- `void glViewport(GLint x, GLint y, GLsizei width, GLsizei height)`
- `void glWindowPos2s(GLshort x, GLshort y)`
- `void glWindowPos2i(GLint x, GLint y)`
- `void glWindowPos2f(GLfloat x, GLfloat y)`
- `void glWindowPos2d(GLdouble x, GLdouble y)`
- `void glWindowPos3s(GLshort x, GLshort y, GLshort z)`
- `void glWindowPos3i(GLint x, GLint y, GLint z)`
- `void glWindowPos3f(GLfloat x, GLfloat y, GLfloat z)`
- `void glWindowPos3d(GLdouble x, GLdouble y, GLdouble z)`
- `void glWindowPos2sv(const GLshort * v)`

- void glWindowPos2iv(const GLint * v)
- void glWindowPos2fv(const GLfloat * v)
- void glWindowPos2dv(const GLdouble * v)
- void glWindowPos3sv(const GLshort * v)
- void glWindowPos3iv(const GLint * v)
- void glWindowPos3fv(const GLfloat * v)
- void glWindowPos3dv(const GLdouble * v)
- void gluBeginCurve(GLUnurbs* nurb)
- void gluBeginPolygon(GLUtesselator* tess)
- void gluBeginSurface(GLUnurbs* nurb)
- void gluBeginTrim(GLUnurbs* nurb)
- void gluCylinder(GLUquadric* quad, GLdouble base, GLdouble top, GLdouble height, GLint slices, GLint stacks)
- void gluDeleteNurbsRenderer(GLUnurbs* nurb)
- void gluDeleteQuadric(GLUquadric* quad)
- void gluDeleteTess(GLUtesselator* tess)
- void gluDisk(GLUquadric* quad, GLdouble inner, GLdouble outer, GLint slices, GLint loops)
- void gluEndCurve(GLUnurbs* nurb)
- void gluEndPolygon(GLUtesselator* tess)
- void gluEndSurface(GLUnurbs* nurb)
- void gluEndTrim(GLUnurbs* nurb)
- const GLubyte * gluErrorString(GLenum error)
- void gluGetNurbsProperty(GLUnurbs* nurb, GLenum property, GLfloat* data)
- const GLubyte * gluGetString(GLenum name)
- void gluGetTessProperty(GLUtesselator* tess, GLenum which, GLdouble* data)
- void gluLoadSamplingMatrices(GLUnurbs* nurb, const GLfloat * model, const GLfloat * perspective, const GLint * view)
- void gluLookAt(GLdouble eyeX, GLdouble eyeY, GLdouble eyeZ, GLdouble centerX, GLdouble centerY, GLdouble centerZ, GLdouble upX, GLdouble upY, GLdouble upZ)
- GLUnurbs * gluNewNurbsRenderer(void)
- GLUquadric * gluNewQuadric(void)
- GLUtesselator* gluNewTess(void)
- void gluNextContour(GLUtesselator* tess, GLenum type)
- void gluNurbsCurve(GLUnurbs* nurb, GLint knotCount, GLfloat * knots, GLint stride, GLfloat * control, GLint order, GLenum type)
- void gluNurbsProperty(GLUnurbs* nurb, GLenum property, GLfloat value)
- void gluNurbsSurface(GLUnurbs* nurb, GLint sKnotCount, GLfloat* sKnots, GLint tKnotCount, GLfloat* tKnots, GLint sStride, GLint tStride, GLfloat* control, GLint sOrder, GLint tOrder, GLenum type)

- void gluOrtho2D(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top)
- void gluPartialDisk(GLUquadric* quad, GLdouble inner, GLdouble outer, GLint slices, GLint loops, GLdouble start, GLdouble sweep)
- void gluPerspective(GLdouble fovy, GLdouble aspect, GLdouble zNear, GLdouble zFar)
- void gluPickMatrix(GLdouble x, GLdouble y, GLdouble delX, GLdouble delY, GLint * viewport)
- GLint gluProject(GLdouble objX, GLdouble objY, GLdouble objZ, const GLdouble * model, const GLdouble * proj, const GLint * view, GLdouble* winX, GLdouble* winY, GLdouble* winZ)
- void gluPwlCurve(GLUnurbs* nurb, GLint count, GLfloat* data, GLint stride, GLenum type)
- void gluQuadricDrawStyle(GLUquadric* quad, GLenum draw)
- void gluQuadricNormals(GLUquadric* quad, GLenum normal)
- void gluQuadricOrientation(GLUquadric* quad, GLenum orientation)
- void gluQuadricTexture(GLUquadric* quad, GLboolean texture)
- GLint gluScaleImage(GLenum format, GLsizei wIn, GLsizei hIn, GLenum typeIn, const void * dataIn, GLsizei wOut, GLsizei hOut, GLenum typeOut, GLvoid* dataOut)
- void gluSphere(GLUquadric* quad, GLdouble radius, GLint slices, GLint stacks)
- void gluTessBeginContour(GLUtesselator* tess)
- void gluTessBeginPolygon(GLUtesselator* tess, GLvoid* data)
- void gluTessEndContour(GLUtesselator* tess)
- void gluTessEndPolygon(GLUtesselator* tess)
- void gluTessNormal(GLUtesselator* tess, GLdouble valueX, GLdouble valueY, GLdouble valueZ)
- void gluTessProperty(GLUtesselator* tess, GLenum which, GLdouble data)
- void gluTessVertex(GLUtesselator* tess, GLdouble * location, GLvoid* data)
- GLint gluUnProject(GLdouble winX, GLdouble winY, GLdouble winZ, const GLdouble * model, const GLdouble * proj, const GLint * view, GLdouble* objX, GLdouble* objY, GLdouble* objZ)
- void glDisable(GLenum cap)

RINGOPENGL (OPENGL 4.3) FUNCTIONS REFERENCE

- GL_ZERO
- GL_FALSE
- GL_LOGIC_OP
- GL_NONE
- GL_TEXTURE_COMPONENTS
- GL_NO_ERROR
- GL_POINTS
- GL_CURRENT_BIT
- GL_TRUE
- GL_ONE
- GL_CLIENT_PIXEL_STORE_BIT
- GL_LINES
- GL_LINE_LOOP
- GL_POINT_BIT
- GL_CLIENT_VERTEX_ARRAY_BIT
- GL_LINE_STRIP
- GL_LINE_BIT
- GL_TRIANGLES
- GL_TRIANGLE_STRIP
- GL_TRIANGLE_FAN
- GL_QUADS
- GL_QUAD_STRIP
- GL_POLYGON_BIT
- GL_POLYGON
- GL_POLYGON_STIPPLE_BIT
- GL_PIXEL_MODE_BIT
- GL_LIGHTING_BIT

- GL_FOG_BIT
- GL_DEPTH_BUFFER_BIT
- GL_ACCUM
- GL_LOAD
- GL_RETURN
- GL_MULT
- GL_ADD
- GL_NEVER
- GL_ACCUM_BUFFER_BIT
- GL_LESS
- GL_EQUAL
- GL_LEQUAL
- GL_GREATER
- GL_NOTEQUAL
- GL_GEQUAL
- GL_ALWAYS
- GL_SRC_COLOR
- GL_ONE_MINUS_SRC_COLOR
- GL_SRC_ALPHA
- GL_ONE_MINUS_SRC_ALPHA
- GL_DST_ALPHA
- GL_ONE_MINUS_DST_ALPHA
- GL_DST_COLOR
- GL_ONE_MINUS_DST_COLOR
- GL_SRC_ALPHA_SATURATE
- GL_STENCIL_BUFFER_BIT
- GL_FRONT_LEFT
- GL_FRONT_RIGHT
- GL_BACK_LEFT
- GL_BACK_RIGHT
- GL_FRONT
- GL_BACK
- GL_LEFT
- GL_RIGHT
- GL_FRONT_AND_BACK
- GL_AUX0

- GL_AUX1
- GL_AUX2
- GL_AUX3
- GL_INVALID_ENUM
- GL_INVALID_VALUE
- GL_INVALID_OPERATION
- GL_STACK_OVERFLOW
- GL_STACK_UNDERFLOW
- GL_OUT_OF_MEMORY
- GL_2D
- GL_3D
- GL_3D_COLOR
- GL_3D_COLOR_TEXTURE
- GL_4D_COLOR_TEXTURE
- GL_PASS_THROUGH_TOKEN
- GL_POINT_TOKEN
- GL_LINE_TOKEN
- GL_POLYGON_TOKEN
- GL_BITMAP_TOKEN
- GL_DRAW_PIXEL_TOKEN
- GL_COPY_PIXEL_TOKEN
- GL_LINE_RESET_TOKEN
- GL_EXP
- GL_VIEWPORT_BIT
- GL_EXP2
- GL_CW
- GL_CCW
- GL_COEFF
- GL_ORDER
- GL_DOMAIN
- GL_CURRENT_COLOR
- GL_CURRENT_INDEX
- GL_CURRENT_NORMAL
- GL_CURRENT_TEXTURE_COORDS
- GL_CURRENT_RASTER_COLOR
- GL_CURRENT_RASTER_INDEX

- GL_CURRENT_RASTER_TEXTURE_COORDS
- GL_CURRENT_RASTER_POSITION
- GL_CURRENT_RASTER_POSITION_VALID
- GL_CURRENT_RASTER_DISTANCE
- GL_POINT_SMOOTH
- GL_POINT_SIZE
- GL_POINT_SIZE_RANGE
- GL_POINT_SIZE_GRANULARITY
- GL_LINE_SMOOTH
- GL_LINE_WIDTH
- GL_LINE_WIDTH_RANGE
- GL_LINE_WIDTH_GRANULARITY
- GL_LINE_STIPPLE
- GL_LINE_STIPPLE_PATTERN
- GL_LINE_STIPPLE_REPEAT
- GL_LIST_MODE
- GL_MAX_LIST_NESTING
- GL_LIST_BASE
- GL_LIST_INDEX
- GL_POLYGON_MODE
- GL_POLYGON_SMOOTH
- GL_POLYGON_STIPPLE
- GL_EDGE_FLAG
- GL_CULL_FACE
- GL_CULL_FACE_MODE
- GL_FRONT_FACE
- GL_LIGHTING
- GL_LIGHT_MODEL_LOCAL_VIEWER
- GL_LIGHT_MODEL_TWO_SIDE
- GL_LIGHT_MODEL_AMBIENT
- GL_SHADE_MODEL
- GL_COLOR_MATERIAL_FACE
- GL_COLOR_MATERIAL_PARAMETER
- GL_COLOR_MATERIAL
- GL_FOG
- GL_FOG_INDEX

- GL_FOG_DENSITY
- GL_FOG_START
- GL_FOG_END
- GL_FOG_MODE
- GL_FOG_COLOR
- GL_DEPTH_RANGE
- GL_DEPTH_TEST
- GL_DEPTH_WRITEMASK
- GL_DEPTH_CLEAR_VALUE
- GL_DEPTH_FUNC
- GL_ACCUM_CLEAR_VALUE
- GL_STENCIL_TEST
- GL_STENCIL_CLEAR_VALUE
- GL_STENCIL_FUNC
- GL_STENCIL_VALUE_MASK
- GL_STENCIL_FAIL
- GL_STENCIL_PASS_DEPTH_FAIL
- GL_STENCIL_PASS_DEPTH_PASS
- GL_STENCIL_REF
- GL_STENCIL_WRITEMASK
- GL_MATRIX_MODE
- GL_NORMALIZE
- GL_VIEWPORT
- GL_MODELVIEW_STACK_DEPTH
- GL_PROJECTION_STACK_DEPTH
- GL_TEXTURE_STACK_DEPTH
- GL_MODELVIEW_MATRIX
- GL_PROJECTION_MATRIX
- GL_TEXTURE_MATRIX
- GL_ATTRIB_STACK_DEPTH
- GL_CLIENT_ATTRIB_STACK_DEPTH
- GL_ALPHA_TEST
- GL_ALPHA_TEST_FUNC
- GL_ALPHA_TEST_REF
- GL_DITHER
- GL_BLEND_DST

- GL_BLEND_SRC
- GL_BLEND
- GL_LOGIC_OP_MODE
- GL_INDEX_LOGIC_OP
- GL_COLOR_LOGIC_OP
- GL_AUX_BUFFERS
- GL_DRAW_BUFFER
- GL_READ_BUFFER
- GL_SCISSOR_BOX
- GL_SCISSOR_TEST
- GL_INDEX_CLEAR_VALUE
- GL_INDEX_WRITEMASK
- GL_COLOR_CLEAR_VALUE
- GL_COLOR_WRITEMASK
- GL_INDEX_MODE
- GL_RGBA_MODE
- GL_DOUBLEBUFFER
- GL_STEREO
- GL_RENDER_MODE
- GL_PERSPECTIVE_CORRECTION_HINT
- GL_POINT_SMOOTH_HINT
- GL_LINE_SMOOTH_HINT
- GL_POLYGON_SMOOTH_HINT
- GL_FOG_HINT
- GL_TEXTURE_GEN_S
- GL_TEXTURE_GEN_T
- GL_TEXTURE_GEN_R
- GL_TEXTURE_GEN_Q
- GL_PIXEL_MAP_I_TO_I
- GL_PIXEL_MAP_S_TO_S
- GL_PIXEL_MAP_I_TO_R
- GL_PIXEL_MAP_I_TO_G
- GL_PIXEL_MAP_I_TO_B
- GL_PIXEL_MAP_I_TO_A
- GL_PIXEL_MAP_R_TO_R
- GL_PIXEL_MAP_G_TO_G

- GL_PIXEL_MAP_B_TO_B
- GL_PIXEL_MAP_A_TO_A
- GL_PIXEL_MAP_I_TO_I_SIZE
- GL_PIXEL_MAP_S_TO_S_SIZE
- GL_PIXEL_MAP_I_TO_R_SIZE
- GL_PIXEL_MAP_I_TO_G_SIZE
- GL_PIXEL_MAP_I_TO_B_SIZE
- GL_PIXEL_MAP_I_TO_A_SIZE
- GL_PIXEL_MAP_R_TO_R_SIZE
- GL_PIXEL_MAP_G_TO_G_SIZE
- GL_PIXEL_MAP_B_TO_B_SIZE
- GL_PIXEL_MAP_A_TO_A_SIZE
- GL_UNPACK_SWAP_BYTES
- GL_UNPACK_LSB_FIRST
- GL_UNPACK_ROW_LENGTH
- GL_UNPACK_SKIP_ROWS
- GL_UNPACK_SKIP_PIXELS
- GL_UNPACK_ALIGNMENT
- GL_PACK_SWAP_BYTES
- GL_PACK_LSB_FIRST
- GL_PACK_ROW_LENGTH
- GL_PACK_SKIP_ROWS
- GL_PACK_SKIP_PIXELS
- GL_PACK_ALIGNMENT
- GL_MAP_COLOR
- GL_MAP_STENCIL
- GL_INDEX_SHIFT
- GL_INDEX_OFFSET
- GL_RED_SCALE
- GL_RED_BIAS
- GL_ZOOM_X
- GL_ZOOM_Y
- GL_GREEN_SCALE
- GL_GREEN_BIAS
- GL_BLUE_SCALE
- GL_BLUE_BIAS

- GL_ALPHA_SCALE
- GL_ALPHA_BIAS
- GL_DEPTH_SCALE
- GL_DEPTH_BIAS
- GL_MAX_EVAL_ORDER
- GL_MAX_LIGHTS
- GL_MAX_CLIP_PLANES
- GL_MAX_TEXTURE_SIZE
- GL_MAX_PIXEL_MAP_TABLE
- GL_MAX_ATTRIB_STACK_DEPTH
- GL_MAX_MODELVIEW_STACK_DEPTH
- GL_MAX_NAME_STACK_DEPTH
- GL_MAX_PROJECTION_STACK_DEPTH
- GL_MAX_TEXTURE_STACK_DEPTH
- GL_MAX_VIEWPORT_DIMS
- GL_MAX_CLIENT_ATTRIB_STACK_DEPTH
- GL_SUBPIXEL_BITS
- GL_INDEX_BITS
- GL_RED_BITS
- GL_GREEN_BITS
- GL_BLUE_BITS
- GL_ALPHA_BITS
- GL_DEPTH_BITS
- GL_STENCIL_BITS
- GL_ACCUM_RED_BITS
- GL_ACCUM_GREEN_BITS
- GL_ACCUM_BLUE_BITS
- GL_ACCUM_ALPHA_BITS
- GL_NAME_STACK_DEPTH
- GL_AUTO_NORMAL
- GL_MAP1_COLOR_4
- GL_MAP1_INDEX
- GL_MAP1_NORMAL
- GL_MAP1_TEXTURE_COORD_1
- GL_MAP1_TEXTURE_COORD_2
- GL_MAP1_TEXTURE_COORD_3

- GL_MAP1_TEXTURE_COORD_4
- GL_MAP1_VERTEX_3
- GL_MAP1_VERTEX_4
- GL_MAP2_COLOR_4
- GL_MAP2_INDEX
- GL_MAP2_NORMAL
- GL_MAP2_TEXTURE_COORD_1
- GL_MAP2_TEXTURE_COORD_2
- GL_MAP2_TEXTURE_COORD_3
- GL_MAP2_TEXTURE_COORD_4
- GL_MAP2_VERTEX_3
- GL_MAP2_VERTEX_4
- GL_MAP1_GRID_DOMAIN
- GL_MAP1_GRID_SEGMENTS
- GL_MAP2_GRID_DOMAIN
- GL_MAP2_GRID_SEGMENTS
- GL_TEXTURE_1D
- GL_TEXTURE_2D
- GL_FEEDBACK_BUFFER_POINTER
- GL_FEEDBACK_BUFFER_SIZE
- GL_FEEDBACK_BUFFER_TYPE
- GL_SELECTION_BUFFER_POINTER
- GL_SELECTION_BUFFER_SIZE
- GL_TEXTURE_WIDTH
- GL_TRANSFORM_BIT
- GL_TEXTURE_HEIGHT
- GL_TEXTURE_INTERNAL_FORMAT
- GL_TEXTURE_BORDER_COLOR
- GL_TEXTURE_BORDER
- GL_DONT_CARE
- GL_FASTEST
- GL_NICEST
- GL_AMBIENT
- GL_DIFFUSE
- GL_SPECULAR
- GL_POSITION

- GL_SPOT_DIRECTION
- GL_SPOT_EXPONENT
- GL_SPOT_CUTOFF
- GL_CONSTANT_ATTENUATION
- GL_LINEAR_ATTENUATION
- GL_QUADRATIC_ATTENUATION
- GL_COMPILE
- GL_COMPILE_AND_EXECUTE
- GL_BYTE
- GL_UNSIGNED_BYTE
- GL_SHORT
- GL_UNSIGNED_SHORT
- GL_INT
- GL_UNSIGNED_INT
- GL_FLOAT
- GL_2_BYTES
- GL_3_BYTES
- GL_4_BYTES
- GL_DOUBLE
- GL_CLEAR
- GL_AND
- GL_AND_REVERSE
- GL_COPY
- GL_AND_INVERTED
- GL_NOOP
- GL_XOR
- GL_OR
- GL_NOR
- GL_EQUIV
- GL_INVERT
- GL_OR_REVERSE
- GL_COPY_INVERTED
- GL_OR_INVERTED
- GL_NAND
- GL_SET
- GL_EMISSION

- GL_SHININESS
- GL_AMBIENT_AND_DIFFUSE
- GL_COLOR_INDEXES
- GL_MODELVIEW
- GL_PROJECTION
- GL_TEXTURE
- GL_COLOR
- GL_DEPTH
- GL_STENCIL
- GL_COLOR_INDEX
- GL_STENCIL_INDEX
- GL_DEPTH_COMPONENT
- GL_RED
- GL_GREEN
- GL_BLUE
- GL_ALPHA
- GL_RGB
- GL_RGBA
- GL_LUMINANCE
- GL_LUMINANCE_ALPHA
- GL_BITMAP
- GL_POINT
- GL_LINE
- GL_FILL
- GL_RENDER
- GL_FEEDBACK
- GL_SELECT
- GL_FLAT
- GL_SMOOTH
- GL_KEEP
- GL_REPLACE
- GL_INCR
- GL_DECR
- GL_VENDOR
- GL_RENDERER
- GL_VERSION

- GL_EXTENSIONS
- GL_S
- GL_ENABLE_BIT
- GL_T
- GL_R
- GL_Q
- GL_MODULATE
- GL_DECAL
- GL_TEXTURE_ENV_MODE
- GL_TEXTURE_ENV_COLOR
- GL_TEXTURE_ENV
- GL_EYE_LINEAR
- GL_OBJECT_LINEAR
- GL_SPHERE_MAP
- GL_TEXTURE_GEN_MODE
- GL_OBJECT_PLANE
- GL_EYE_PLANE
- GL_NEAREST
- GL_LINEAR
- GL_NEAREST_MIPMAP_NEAREST
- GL_LINEAR_MIPMAP_NEAREST
- GL_NEAREST_MIPMAP_LINEAR
- GL_LINEAR_MIPMAP_LINEAR
- GL_TEXTURE_MAG_FILTER
- GL_TEXTURE_MIN_FILTER
- GL_TEXTURE_WRAP_S
- GL_TEXTURE_WRAP_T
- GL_CLAMP
- GL_REPEAT
- GL_POLYGON_OFFSET_UNITS
- GL_POLYGON_OFFSET_POINT
- GL_POLYGON_OFFSET_LINE
- GL_R3_G3_B2
- GL_V2F
- GL_V3F
- GL_C4UB_V2F

- GL_C4UB_V3F
- GL_C3F_V3F
- GL_N3F_V3F
- GL_C4F_N3F_V3F
- GL_T2F_V3F
- GL_T4F_V4F
- GL_T2F_C4UB_V3F
- GL_T2F_C3F_V3F
- GL_T2F_N3F_V3F
- GL_T2F_C4F_N3F_V3F
- GL_T4F_C4F_N3F_V4F
- GL_CLIP_PLANE0
- GL_CLIP_PLANE1
- GL_CLIP_PLANE2
- GL_CLIP_PLANE3
- GL_CLIP_PLANE4
- GL_CLIP_PLANE5
- GL_LIGHT0
- GL_COLOR_BUFFER_BIT
- GL_LIGHT1
- GL_LIGHT2
- GL_LIGHT3
- GL_LIGHT4
- GL_LIGHT5
- GL_LIGHT6
- GL_LIGHT7
- GL_HINT_BIT
- GL_POLYGON_OFFSET_FILL
- GL_POLYGON_OFFSET_FACTOR
- GL_ALPHA4
- GL_ALPHA8
- GL_ALPHA12
- GL_ALPHA16
- GL_LUMINANCE4
- GL_LUMINANCE8
- GL_LUMINANCE12

- GL_LUMINANCE16
- GL_LUMINANCE4_ALPHA4
- GL_LUMINANCE6_ALPHA2
- GL_LUMINANCE8_ALPHA8
- GL_LUMINANCE12_ALPHA4
- GL_LUMINANCE12_ALPHA12
- GL_LUMINANCE16_ALPHA16
- GL_INTENSITY
- GL_INTENSITY4
- GL_INTENSITY8
- GL_INTENSITY12
- GL_INTENSITY16
- GL_RGB4
- GL_RGB5
- GL_RGB8
- GL_RGB10
- GL_RGB12
- GL_RGB16
- GL_RGBA2
- GL_RGBA4
- GL_RGB5_A1
- GL_RGBA8
- GL_RGB10_A2
- GL_RGBA12
- GL_RGBA16
- GL_TEXTURE_RED_SIZE
- GL_TEXTURE_GREEN_SIZE
- GL_TEXTURE_BLUE_SIZE
- GL_TEXTURE_ALPHA_SIZE
- GL_TEXTURE_LUMINANCE_SIZE
- GL_TEXTURE_INTENSITY_SIZE
- GL_PROXY_TEXTURE_1D
- GL_PROXY_TEXTURE_2D
- GL_TEXTURE_PRIORITY
- GL_TEXTURE_RESIDENT
- GL_TEXTURE_BINDING_1D

- GL_TEXTURE_BINDING_2D
- GL_VERTEX_ARRAY
- GL_NORMAL_ARRAY
- GL_COLOR_ARRAY
- GL_INDEX_ARRAY
- GL_TEXTURE_COORD_ARRAY
- GL_EDGE_FLAG_ARRAY
- GL_VERTEX_ARRAY_SIZE
- GL_VERTEX_ARRAY_TYPE
- GL_VERTEX_ARRAY_STRIDE
- GL_NORMAL_ARRAY_TYPE
- GL_NORMAL_ARRAY_STRIDE
- GL_COLOR_ARRAY_SIZE
- GL_COLOR_ARRAY_TYPE
- GL_COLOR_ARRAY_STRIDE
- GL_INDEX_ARRAY_TYPE
- GL_INDEX_ARRAY_STRIDE
- GL_TEXTURE_COORD_ARRAY_SIZE
- GL_TEXTURE_COORD_ARRAY_TYPE
- GL_TEXTURE_COORD_ARRAY_STRIDE
- GL_EDGE_FLAG_ARRAY_STRIDE
- GL_VERTEX_ARRAY_POINTER
- GL_NORMAL_ARRAY_POINTER
- GL_COLOR_ARRAY_POINTER
- GL_INDEX_ARRAY_POINTER
- GL_TEXTURE_COORD_ARRAY_POINTER
- GL_EDGE_FLAG_ARRAY_POINTER
- GL_COLOR_INDEX1_EXT
- GL_COLOR_INDEX2_EXT
- GL_COLOR_INDEX4_EXT
- GL_COLOR_INDEX8_EXT
- GL_COLOR_INDEX12_EXT
- GL_COLOR_INDEX16_EXT
- GL_EVAL_BIT
- GL_LIST_BIT
- GL_TEXTURE_BIT

- GL_SCISSOR_BIT
- GL_ALL_ATTRIB_BITS
- GL_CLIENT_ALL_ATTRIB_BITS
- GL_SMOOTH_POINT_SIZE_RANGE
- GL_SMOOTH_POINT_SIZE_GRANULARITY
- GL_SMOOTH_LINE_WIDTH_RANGE
- GL_SMOOTH_LINE_WIDTH_GRANULARITY
- GL_UNSIGNED_BYTE_3_3_2
- GL_UNSIGNED_SHORT_4_4_4_4
- GL_UNSIGNED_SHORT_5_5_5_1
- GL_UNSIGNED_INT_8_8_8_8
- GL_UNSIGNED_INT_10_10_10_2
- GL_RESCALE_NORMAL
- GL_TEXTURE_BINDING_3D
- GL_PACK_SKIP_IMAGES
- GL_PACK_IMAGE_HEIGHT
- GL_UNPACK_SKIP_IMAGES
- GL_UNPACK_IMAGE_HEIGHT
- GL_TEXTURE_3D
- GL_PROXY_TEXTURE_3D
- GL_TEXTURE_DEPTH
- GL_TEXTURE_WRAP_R
- GL_MAX_3D_TEXTURE_SIZE
- GL_BGR
- GL_BGRA
- GL_MAX_ELEMENTS_VERTICES
- GL_MAX_ELEMENTS_INDICES
- GL_CLAMP_TO_EDGE
- GL_TEXTURE_MIN_LOD
- GL_TEXTURE_MAX_LOD
- GL_TEXTURE_BASE_LEVEL
- GL_TEXTURE_MAX_LEVEL
- GL_LIGHT_MODEL_COLOR_CONTROL
- GL_SINGLE_COLOR
- GL_SEPARATE_SPECULAR_COLOR
- GL_UNSIGNED_BYTE_2_3_3_REV

- GL_UNSIGNED_SHORT_5_6_5
- GL_UNSIGNED_SHORT_5_6_5_REV
- GL_UNSIGNED_SHORT_4_4_4_REV
- GL_UNSIGNED_SHORT_1_5_5_5_REV
- GL_UNSIGNED_INT_8_8_8_8_REV
- GL_ALIASED_POINT_SIZE_RANGE
- GL_ALIASED_LINE_WIDTH_RANGE
- GL_MULTISAMPLE
- GL_SAMPLE_ALPHA_TO_COVERAGE
- GL_SAMPLE_ALPHA_TO_ONE
- GL_SAMPLE_COVERAGE
- GL_SAMPLE_BUFFERS
- GL_SAMPLES
- GL_SAMPLE_COVERAGE_VALUE
- GL_SAMPLE_COVERAGE_INVERT
- GL_CLAMP_TO_BORDER
- GL_TEXTURE0
- GL_TEXTURE1
- GL_TEXTURE2
- GL_TEXTURE3
- GL_TEXTURE4
- GL_TEXTURE5
- GL_TEXTURE6
- GL_TEXTURE7
- GL_TEXTURE8
- GL_TEXTURE9
- GL_TEXTURE10
- GL_TEXTURE11
- GL_TEXTURE12
- GL_TEXTURE13
- GL_TEXTURE14
- GL_TEXTURE15
- GL_TEXTURE16
- GL_TEXTURE17
- GL_TEXTURE18
- GL_TEXTURE19

- GL_TEXTURE20
- GL_TEXTURE21
- GL_TEXTURE22
- GL_TEXTURE23
- GL_TEXTURE24
- GL_TEXTURE25
- GL_TEXTURE26
- GL_TEXTURE27
- GL_TEXTURE28
- GL_TEXTURE29
- GL_TEXTURE30
- GL_TEXTURE31
- GL_ACTIVE_TEXTURE
- GL_CLIENT_ACTIVE_TEXTURE
- GL_MAX_TEXTURE_UNITS
- GL_TRANSPOSE_MODELVIEW_MATRIX
- GL_TRANSPOSE_PROJECTION_MATRIX
- GL_TRANSPOSE_TEXTURE_MATRIX
- GL_TRANSPOSE_COLOR_MATRIX
- GL_SUBTRACT
- GL_COMPRESSED_ALPHA
- GL_COMPRESSED_LUMINANCE
- GL_COMPRESSED_LUMINANCE_ALPHA
- GL_COMPRESSED_INTENSITY
- GL_COMPRESSED_RGB
- GL_COMPRESSED_RGBA
- GL_TEXTURE_COMPRESSION_HINT
- GL_NORMAL_MAP
- GL_REFLECTION_MAP
- GL_TEXTURE_CUBE_MAP
- GL_TEXTURE_BINDING_CUBE_MAP
- GL_TEXTURE_CUBE_MAP_POSITIVE_X
- GL_TEXTURE_CUBE_MAP_NEGATIVE_X
- GL_TEXTURE_CUBE_MAP_POSITIVE_Y
- GL_TEXTURE_CUBE_MAP_NEGATIVE_Y
- GL_TEXTURE_CUBE_MAP_POSITIVE_Z

- GL_TEXTURE_CUBE_MAP_NEGATIVE_Z
- GL_PROXY_TEXTURE_CUBE_MAP
- GL_MAX_CUBE_MAP_TEXTURE_SIZE
- GL_COMBINE
- GL_COMBINE_RGB
- GL_COMBINE_ALPHA
- GL_RGB_SCALE
- GL_ADD_SIGNED
- GL_INTERPOLATE
- GL_CONSTANT
- GL_PRIMARY_COLOR
- GL_PREVIOUS
- GL_SOURCE0_RGB
- GL_SOURCE1_RGB
- GL_SOURCE2_RGB
- GL_SOURCE0_ALPHA
- GL_SOURCE1_ALPHA
- GL_SOURCE2_ALPHA
- GL_OPERAND0_RGB
- GL_OPERAND1_RGB
- GL_OPERAND2_RGB
- GL_OPERAND0_ALPHA
- GL_OPERAND1_ALPHA
- GL_OPERAND2_ALPHA
- GL_TEXTURE_COMPRESSED_IMAGE_SIZE
- GL_TEXTURE_COMPRESSED
- GL_NUM_COMPRESSED_TEXTURE_FORMATS
- GL_COMPRESSED_TEXTURE_FORMATS
- GL_DOT3_RGB
- GL_DOT3_RGBA
- GL_MULTISAMPLE_BIT
- GL_BLEND_DST_RGB
- GL_BLEND_SRC_RGB
- GL_BLEND_DST_ALPHA
- GL_BLEND_SRC_ALPHA
- GL_POINT_SIZE_MIN

- GL_POINT_SIZE_MAX
- GL_POINT_FADE_THRESHOLD_SIZE
- GL_POINT_DISTANCE_ATTENUATION
- GL_GENERATE_MIPMAP
- GL_GENERATE_MIPMAP_HINT
- GL_DEPTH_COMPONENT16
- GL_DEPTH_COMPONENT24
- GL_DEPTH_COMPONENT32
- GL_MIRRORED_REPEAT
- GL_FOG_COORDINATE_SOURCE
- GL_FOG_COORDINATE
- GL_FRAGMENT_DEPTH
- GL_CURRENT_FOG_COORDINATE
- GL_FOG_COORDINATE_ARRAY_TYPE
- GL_FOG_COORDINATE_ARRAY_STRIDE
- GL_FOG_COORDINATE_ARRAY_POINTER
- GL_FOG_COORDINATE_ARRAY
- GL_COLOR_SUM
- GL_CURRENT_SECONDARY_COLOR
- GL_SECONDARY_COLOR_ARRAY_SIZE
- GL_SECONDARY_COLOR_ARRAY_TYPE
- GL_SECONDARY_COLOR_ARRAY_STRIDE
- GL_SECONDARY_COLOR_ARRAY_POINTER
- GL_SECONDARY_COLOR_ARRAY
- GL_MAX_TEXTURE_LOD_BIAS
- GL_TEXTURE_FILTER_CONTROL
- GL_TEXTURE_LOD_BIAS
- GL_INCR_WRAP
- GL_DECR_WRAP
- GL_TEXTURE_DEPTH_SIZE
- GL_DEPTH_TEXTURE_MODE
- GL_TEXTURE_COMPARE_MODE
- GL_TEXTURE_COMPARE_FUNC
- GL_COMPARE_R_TO_TEXTURE
- GL_CURRENT_FOG_COORD
- GL_FOG_COORD

- GL_FOG_COORD_ARRAY
- GL_FOG_COORD_ARRAY_BUFFER_BINDING
- GL_FOG_COORD_ARRAY_POINTER
- GL_FOG_COORD_ARRAY_STRIDE
- GL_FOG_COORD_ARRAY_TYPE
- GL_FOG_COORD_SRC
- GL_SRC0_ALPHA
- GL_SRC0_RGB
- GL_SRC1_ALPHA
- GL_SRC1_RGB
- GL_SRC2_ALPHA
- GL_SRC2_RGB
- GL_BUFFER_SIZE
- GL_BUFFER_USAGE
- GL_QUERY_COUNTER_BITS
- GL_CURRENT_QUERY
- GL_QUERY_RESULT
- GL_QUERY_RESULT_AVAILABLE
- GL_ARRAY_BUFFER
- GL_ELEMENT_ARRAY_BUFFER
- GL_ARRAY_BUFFER_BINDING
- GL_ELEMENT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ARRAY_BUFFER_BINDING
- GL_NORMAL_ARRAY_BUFFER_BINDING
- GL_COLOR_ARRAY_BUFFER_BINDING
- GL_INDEX_ARRAY_BUFFER_BINDING
- GL_TEXTURE_COORD_ARRAY_BUFFER_BINDING
- GL_EDGE_FLAG_ARRAY_BUFFER_BINDING
- GL_SECONDARY_COLOR_ARRAY_BUFFER_BINDING
- GL_FOG_COORDINATE_ARRAY_BUFFER_BINDING
- GL_WEIGHT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ATTRIB_ARRAY_BUFFER_BINDING
- GL_READ_ONLY
- GL_WRITE_ONLY
- GL_READ_WRITE
- GL_BUFFER_ACCESS

- GL_BUFFER_MAPPED
- GL_BUFFER_MAP_POINTER
- GL_STREAM_DRAW
- GL_STREAM_READ
- GL_STREAM_COPY
- GL_STATIC_DRAW
- GL_STATIC_READ
- GL_STATIC_COPY
- GL_DYNAMIC_DRAW
- GL_DYNAMIC_READ
- GL_DYNAMIC_COPY
- GL_SAMPLES_PASSED
- GL_BLEND_EQUATION_RGB
- GL_VERTEX_ATTRIB_ARRAY_ENABLED
- GL_VERTEX_ATTRIB_ARRAY_SIZE
- GL_VERTEX_ATTRIB_ARRAY_STRIDE
- GL_VERTEX_ATTRIB_ARRAY_TYPE
- GL_CURRENT_VERTEX_ATTRIB
- GL_VERTEX_PROGRAM_POINT_SIZE
- GL_VERTEX_PROGRAM_TWO_SIDE
- GL_VERTEX_ATTRIB_ARRAY_POINTER
- GL_STENCIL_BACK_FUNC
- GL_STENCIL_BACK_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_PASS
- GL_MAX_DRAW_BUFFERS
- GL_DRAW_BUFFER0
- GL_DRAW_BUFFER1
- GL_DRAW_BUFFER2
- GL_DRAW_BUFFER3
- GL_DRAW_BUFFER4
- GL_DRAW_BUFFER5
- GL_DRAW_BUFFER6
- GL_DRAW_BUFFER7
- GL_DRAW_BUFFER8
- GL_DRAW_BUFFER9

- GL_DRAW_BUFFER10
- GL_DRAW_BUFFER11
- GL_DRAW_BUFFER12
- GL_DRAW_BUFFER13
- GL_DRAW_BUFFER14
- GL_DRAW_BUFFER15
- GL_BLEND_EQUATION_ALPHA
- GL_POINT_SPRITE
- GL_COORD_REPLACE
- GL_MAX_VERTEX_ATTRIBS
- GL_VERTEX_ATTRIB_ARRAY_NORMALIZED
- GL_MAX_TEXTURE_COORDS
- GL_MAX_TEXTURE_IMAGE_UNITS
- GL_FRAGMENT_SHADER
- GL_VERTEX_SHADER
- GL_MAX_FRAGMENT_UNIFORM_COMPONENTS
- GL_MAX_VERTEX_UNIFORM_COMPONENTS
- GL_MAX_VARYING_FLOATS
- GL_MAX_VERTEX_TEXTURE_IMAGE_UNITS
- GL_MAX_COMBINED_TEXTURE_IMAGE_UNITS
- GL_SHADER_TYPE
- GL_FLOAT_VEC2
- GL_FLOAT_VEC3
- GL_FLOAT_VEC4
- GL_INT_VEC2
- GL_INT_VEC3
- GL_INT_VEC4
- GL_BOOL
- GL_BOOL_VEC2
- GL_BOOL_VEC3
- GL_BOOL_VEC4
- GL_FLOAT_MAT2
- GL_FLOAT_MAT3
- GL_FLOAT_MAT4
- GL_SAMPLER_1D
- GL_SAMPLER_2D

- GL_SAMPLER_3D
- GL_SAMPLER_CUBE
- GL_SAMPLER_1D_SHADOW
- GL_SAMPLER_2D_SHADOW
- GL_DELETE_STATUS
- GL_COMPILE_STATUS
- GL_LINK_STATUS
- GL_VALIDATE_STATUS
- GL_INFO_LOG_LENGTH
- GL_ATTACHED_SHADERS
- GL_ACTIVE_UNIFORMS
- GL_ACTIVE_UNIFORM_MAX_LENGTH
- GL_SHADER_SOURCE_LENGTH
- GL_ACTIVE_ATTRIBUTES
- GL_ACTIVE_ATTRIBUTE_MAX_LENGTH
- GL_FRAGMENT_SHADER_DERIVATIVE_HINT
- GL_SHADING_LANGUAGE_VERSION
- GL_CURRENT_PROGRAM
- GL_POINT_SPRITE_COORD_ORIGIN
- GL_LOWER_LEFT
- GL_UPPER_LEFT
- GL_STENCIL_BACK_REF
- GL_STENCIL_BACK_VALUE_MASK
- GL_STENCIL_BACK_WRITEMASK
- GL_CURRENT_RASTER_SECONDARY_COLOR
- GL_PIXEL_PACK_BUFFER
- GL_PIXEL_UNPACK_BUFFER
- GL_PIXEL_PACK_BUFFER_BINDING
- GL_PIXEL_UNPACK_BUFFER_BINDING
- GL_FLOAT_MAT2x3
- GL_FLOAT_MAT2x4
- GL_FLOAT_MAT3x2
- GL_FLOAT_MAT3x4
- GL_FLOAT_MAT4x2
- GL_FLOAT_MAT4x3
- GL_SRGB

- GL_SRGB8
- GL_SRGB_ALPHA
- GL_SRGB8_ALPHA8
- GL_SLUMINANCE_ALPHA
- GL_SLUMINANCE8_ALPHA8
- GL_SLUMINANCE
- GL_SLUMINANCE8
- GL_COMPRESSED_SRGB
- GL_COMPRESSED_SRGB_ALPHA
- GL_COMPRESSED_SLUMINANCE
- GL_COMPRESSED_SLUMINANCE_ALPHA
- GL_CLIP_DISTANCE0
- GL_CLIP_DISTANCE1
- GL_CLIP_DISTANCE2
- GL_CLIP_DISTANCE3
- GL_CLIP_DISTANCE4
- GL_CLIP_DISTANCE5
- GL_COMPARE_REF_TO_TEXTURE
- GL_MAX_CLIP_DISTANCES
- GL_MAX_VARYING_COMPONENTS
- GL_CONTEXT_FLAG_FORWARD_COMPATIBLE_BIT
- GL_MAJOR_VERSION
- GL_MINOR_VERSION
- GL_NUM_EXTENSIONS
- GL_CONTEXT_FLAGS
- GL_DEPTH_BUFFER
- GL_STENCIL_BUFFER
- GL_RGBA32F
- GL_RGB32F
- GL_RGBA16F
- GL_RGB16F
- GL_VERTEX_ATTRIB_ARRAY_INTEGER
- GL_MAX_ARRAY_TEXTURE_LAYERS
- GL_MIN_PROGRAM_TEXEL_OFFSET
- GL_MAX_PROGRAM_TEXEL_OFFSET
- GL_CLAMP_VERTEX_COLOR

- GL_CLAMP_FRAGMENT_COLOR
- GL_CLAMP_READ_COLOR
- GL_FIXED_ONLY
- GL_TEXTURE_RED_TYPE
- GL_TEXTURE_GREEN_TYPE
- GL_TEXTURE_BLUE_TYPE
- GL_TEXTURE_ALPHA_TYPE
- GL_TEXTURE_LUMINANCE_TYPE
- GL_TEXTURE_INTENSITY_TYPE
- GL_TEXTURE_DEPTH_TYPE
- GL_TEXTURE_1D_ARRAY
- GL_PROXY_TEXTURE_1D_ARRAY
- GL_TEXTURE_2D_ARRAY
- GL_PROXY_TEXTURE_2D_ARRAY
- GL_TEXTURE_BINDING_1D_ARRAY
- GL_TEXTURE_BINDING_2D_ARRAY
- GL_R11F_G11F_B10F
- GL_UNSIGNED_INT_10F_11F_11F_REV
- GL_RGB9_E5
- GL_UNSIGNED_INT_5_9_9_9_REV
- GL_TEXTURE_SHARED_SIZE
- GL_TRANSFORM_FEEDBACK_VARYING_MAX_LENGTH
- GL_TRANSFORM_FEEDBACK_BUFFER_MODE
- GL_MAX_TRANSFORM_FEEDBACK_SEPARATE_COMPONENTS
- GL_TRANSFORM_FEEDBACK_VARYINGS
- GL_TRANSFORM_FEEDBACK_BUFFER_START
- GL_TRANSFORM_FEEDBACK_BUFFER_SIZE
- GL_PRIMITIVES_GENERATED
- GL_TRANSFORM_FEEDBACK_PRIMITIVES_WRITTEN
- GL_RASTERIZER_DISCARD
- GL_MAX_TRANSFORM_FEEDBACK_INTERLEAVED_COMPONENTS
- GL_MAX_TRANSFORM_FEEDBACK_SEPARATE_ATTRIBS
- GL_INTERLEAVED_ATTRIBS
- GL_SEPARATE_ATTRIBS
- GL_TRANSFORM_FEEDBACK_BUFFER
- GL_TRANSFORM_FEEDBACK_BUFFER_BINDING

- GL_RGBA32UI
- GL_RGB32UI
- GL_RGBA16UI
- GL_RGB16UI
- GL_RGBA8UI
- GL_RGB8UI
- GL_RGBA32I
- GL_RGB32I
- GL_RGBA16I
- GL_RGB16I
- GL_RGBA8I
- GL_RGB8I
- GL_RED_INTEGER
- GL_GREEN_INTEGER
- GL_BLUE_INTEGER
- GL_ALPHA_INTEGER
- GL_RGB_INTEGER
- GL_RGBA_INTEGER
- GL_BGR_INTEGER
- GL_BGRA_INTEGER
- GL_SAMPLER_1D_ARRAY
- GL_SAMPLER_2D_ARRAY
- GL_SAMPLER_1D_ARRAY_SHADOW
- GL_SAMPLER_2D_ARRAY_SHADOW
- GL_SAMPLER_CUBE_SHADOW
- GL_UNSIGNED_INT_VEC2
- GL_UNSIGNED_INT_VEC3
- GL_UNSIGNED_INT_VEC4
- GL_INT_SAMPLER_1D
- GL_INT_SAMPLER_2D
- GL_INT_SAMPLER_3D
- GL_INT_SAMPLER_CUBE
- GL_INT_SAMPLER_1D_ARRAY
- GL_INT_SAMPLER_2D_ARRAY
- GL_UNSIGNED_INT_SAMPLER_1D
- GL_UNSIGNED_INT_SAMPLER_2D

- GL_UNSIGNED_INT_SAMPLER_3D
- GL_UNSIGNED_INT_SAMPLER_CUBE
- GL_UNSIGNED_INT_SAMPLER_1D_ARRAY
- GL_UNSIGNED_INT_SAMPLER_2D_ARRAY
- GL_QUERY_WAIT
- GL_QUERY_NO_WAIT
- GL_QUERY_BY_REGION_WAIT
- GL_QUERY_BY_REGION_NO_WAIT
- GL_TEXTURE_RECTANGLE
- GL_TEXTURE_BINDING_RECTANGLE
- GL_PROXY_TEXTURE_RECTANGLE
- GL_MAX_RECTANGLE_TEXTURE_SIZE
- GL_SAMPLER_2D_RECT
- GL_SAMPLER_2D_RECT_SHADOW
- GL_TEXTURE_BUFFER
- GL_MAX_TEXTURE_BUFFER_SIZE
- GL_TEXTURE_BINDING_BUFFER
- GL_TEXTURE_BUFFER_DATA_STORE_BINDING
- GL_TEXTURE_BUFFER_FORMAT
- GL_SAMPLER_BUFFER
- GL_INT_SAMPLER_2D_RECT
- GL_INT_SAMPLER_BUFFER
- GL_UNSIGNED_INT_SAMPLER_2D_RECT
- GL_UNSIGNED_INT_SAMPLER_BUFFER
- GL_RED_SNORM
- GL_RG_SNORM
- GL_RGB_SNORM
- GL_RGBA_SNORM
- GL_R8_SNORM
- GL_RG8_SNORM
- GL_RGB8_SNORM
- GL_RGBA8_SNORM
- GL_R16_SNORM
- GL_RG16_SNORM
- GL_RGB16_SNORM
- GL_RGBA16_SNORM

- GL_SIGNED_NORMALIZED
- GL_PRIMITIVE_RESTART
- GL_PRIMITIVE_RESTART_INDEX
- GL_BUFFER_ACCESS_FLAGS
- GL_BUFFER_MAP_LENGTH
- GL_BUFFER_MAP_OFFSET
- GL_CONTEXT_CORE_PROFILE_BIT
- GL_CONTEXT_COMPATIBILITY_PROFILE_BIT
- GL_LINES_ADJACENCY
- GL_LINE_STRIP_ADJACENCY
- GL_TRIANGLES_ADJACENCY
- GL_TRIANGLE_STRIP_ADJACENCY
- GL_PROGRAM_POINT_SIZE
- GL_GEOMETRY_VERTICES_OUT
- GL_GEOMETRY_INPUT_TYPE
- GL_GEOMETRY_OUTPUT_TYPE
- GL_MAX_GEOMETRY_TEXTURE_IMAGE_UNITS
- GL_FRAMEBUFFER_ATTACHMENT_LAYERED
- GL_FRAMEBUFFER_INCOMPLETE_LAYER_TARGETS
- GL_GEOMETRY_SHADER
- GL_MAX_GEOMETRY_UNIFORM_COMPONENTS
- GL_MAX_GEOMETRY_OUTPUT_VERTICES
- GL_MAX_GEOMETRY_TOTAL_OUTPUT_COMPONENTS
- GL_MAX_VERTEX_OUTPUT_COMPONENTS
- GL_MAX_GEOMETRY_INPUT_COMPONENTS
- GL_MAX_GEOMETRY_OUTPUT_COMPONENTS
- GL_MAX_FRAGMENT_INPUT_COMPONENTS
- GL_CONTEXT_PROFILE_MASK
- GL_VERTEX_ATTRIB_ARRAY_DIVISOR
- GL_RGB10_A2UI
- GL_SAMPLE_SHADING
- GL_MIN_SAMPLE_SHADING_VALUE
- GL_MIN_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_COMPONENTS
- GL_TEXTURE_CUBE_MAP_ARRAY

- GL_TEXTURE_BINDING_CUBE_MAP_ARRAY
- GL_PROXY_TEXTURE_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY_SHADOW
- GL_INT_SAMPLER_CUBE_MAP_ARRAY
- GL_UNSIGNED_INT_SAMPLER_CUBE_MAP_ARRAY
- GL_TRANSFORM_FEEDBACK_PAUSED
- GL_TRANSFORM_FEEDBACK_ACTIVE
- GL_COMPRESSED_RGBA_BPTC_UNORM
- GL_COMPRESSED_SRGB_ALPHA_BPTC_UNORM
- GL_COMPRESSED_RGB_BPTC_SIGNED_FLOAT
- GL_COMPRESSED_RGB_BPTC_UNSIGNED_FLOAT
- GL_COPY_READ_BUFFER_BINDING
- GL_COPY_WRITE_BUFFER_BINDING
- GL_NUM_SHADING_LANGUAGE_VERSIONS
- GL_VERTEX_ATTRIB_ARRAY_LONG
- void glAccum(GLenum op, GLfloat value)
- void glActiveTexture(GLenum texture)
- void glAlphaFunc(GLenum func, GLclampf ref)
- GLboolean glAreTexturesResident(GLsizei n, const GLuint * textures, GLboolean * residences)
- void glArrayElement(GLint i)
- void glAttachShader(GLuint program, GLuint shader)
- void glBegin(GLenum mode)
- void glBeginQuery(GLenum target, GLuint id)
- void glBindAttribLocation(GLuint program, GLuint index, const GLchar *name)
- void glBindBuffer(GLenum target, GLuint buffer)
- void glBindTexture(GLenum target, GLuint texture)
- void glBitmap(GLsizei width, GLsizei height, GLfloat xorig, GLfloat yorig, GLfloat xmove, GLfloat ymove, const GLubyte * bitmap)
- void glBlendColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)
- void glBlendEquation(GLenum mode)
- void glBlendEquationSeparate(GLenum modeRGB, GLenum modeAlpha)
- void glBlendFunc(GLenum sfactor, GLenum dfactor)
- void glBlendFuncSeparate(GLenum srcRGB, GLenum dstRGB, GLenum srcAlpha, GLenum dstAlpha)
- void glBufferData(GLenum target, GLsizeiptr size, const GLvoid * data, GLenum usage)
- void glBufferSubData(GLenum target, GLintptr offset, GLsizeiptr size, const GLvoid * data)

- void glCallList(GLuint list)
- void glCallLists(GLsizei n, GLenum type, const GLvoid * lists)
- void glClear(GLbitfield mask)
- void glClearAccum(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glClearColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)
- void glClearDepth(GLclampd depth)
- void glClearIndex(GLfloat c)
- void glClearStencil(GLint s)
- void glClientActiveTexture(GLenum texture)
- void glClipPlane(GLenum plane, const GLdouble * equation)
- void glColor3b(GLbyte red, GLbyte green, GLbyte blue)
- void glColor3s(GLshort red, GLshort green, GLshort blue)
- void glColor3i(GLint red, GLint green, GLint blue)
- void glColor3f(GLfloat red, GLfloat green, GLfloat blue)
- void glColor3d(GLdouble red, GLdouble green, GLdouble blue)
- void glColor3ub(GLubyte red, GLubyte green, GLubyte blue)
- void glColor3us(GLushort red, GLushort green, GLushort blue)
- void glColor3ui(GLuint red, GLuint green, GLuint blue)
- void glColor4b(GLbyte red, GLbyte green, GLbyte blue, GLbyte alpha)
- void glColor4s(GLshort red, GLshort green, GLshort blue, GLshort alpha)
- void glColor4i(GLint red, GLint green, GLint blue, GLint alpha)
- void glColor4f(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glColor4d(GLdouble red, GLdouble green, GLdouble blue, GLdouble alpha)
- void glColor4ub(GLubyte red, GLubyte green, GLubyte blue, GLubyte alpha)
- void glColor4us(GLushort red, GLushort green, GLushort blue, GLushort alpha)
- void glColor4ui(GLuint red, GLuint green, GLuint blue, GLuint alpha)
- void glColor3bv(const GLbyte * v)
- void glColor3sv(const GLshort * v)
- void glColor3iv(const GLint * v)
- void glColor3fv(const GLfloat * v)
- void glColor3dv(const GLdouble * v)
- void glColor3ubv(const GLubyte * v)
- void glColor3usv(const GLushort * v)
- void glColor3uiv(const GLuint * v)
- void glColor4bv(const GLbyte * v)
- void glColor4sv(const GLshort * v)

- `void glColor4iv(const GLint * v)`
- `void glColor4fv(const GLfloat * v)`
- `void glColor4dv(const GLdouble * v)`
- `void glColor4ubv(const GLubyte * v)`
- `void glColor4usv(const GLushort * v)`
- `void glColor4uiv(const GLuint * v)`
- `void glColorMask(GLboolean red, GLboolean green, GLboolean blue, GLboolean alpha)`
- `void glColorMaterial(GLenum face, GLenum mode)`
- `void glColorPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)`
- `void glColorSubTable(GLenum target, GLsizei start, GLsizei count, GLenum format, GLenum type, const GLvoid * data)`
- `void glColorTable(GLenum target, GLenum internalformat, GLsizei width, GLenum format, GLenum type, const GLvoid * data)`
- `void glColorTableParameterfv(GLenum target, GLenum pname, const GLfloat * params)`
- `void glColorTableParameteriv(GLenum target, GLenum pname, const GLint * params)`
- `void glCompileShader(GLuint shader)`
- `void glCompressedTexImage1D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLint border, GLsizei imageSize, const GLvoid * data)`
- `void glCompressedTexImage2D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLint border, GLsizei imageSize, const GLvoid * data)`
- `void glCompressedTexImage3D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLsizei depth, GLint border, GLsizei imageSize, const GLvoid * data)`
- `void glCompressedTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLsizei imageSize, const GLvoid * data)`
- `void glCompressedTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLsizei imageSize, const GLvoid * data)`
- `void glCompressedTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLsizei imageSize, const GLvoid * data)`
- `void glConvolutionFilter1D(GLenum target, GLenum internalformat, GLsizei width, GLenum format, GLenum type, const GLvoid * data)`
- `void glConvolutionFilter2D(GLenum target, GLenum internalformat, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)`
- `void glConvolutionParameterf(GLenum target, GLenum pname, GLfloat params)`
- `void glConvolutionParameteri(GLenum target, GLenum pname, GLint params)`
- `void glConvolutionParameterfv(GLenum target, GLenum pname, const GLfloat * params)`
- `void glConvolutionParameteriv(GLenum target, GLenum pname, const GLint * params)`
- `void glCopyColorSubTable(GLenum target, GLsizei start, GLint x, GLint y, GLsizei width)`
- `void glCopyColorTable(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width)`
- `void glCopyConvolutionFilter1D(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width)`

- void glCopyConvolutionFilter2D(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width, GLsizei height)
- void glCopyPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum type)
- void glCopyTexImage1D(GLenum target, GLint level, GLenum internalformat, GLint x, GLint y, GLsizei width, GLint border)
- void glCopyTexImage2D(GLenum target, GLint level, GLenum internalformat, GLint x, GLint y, GLsizei width, GLsizei height, GLint border)
- void glCopyTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLint x, GLint y, GLsizei width)
- void glCopyTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint x, GLint y, GLsizei width, GLsizei height)
- void glCopyTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLint x, GLint y, GLsizei width, GLsizei height)
- GLuint glCreateProgram(void)
- GLuint glCreateShader(GLenum shaderType)
- void glCullFace(GLenum mode)
- void glDeleteBuffers(GLsizei n, const GLuint * buffers)
- void glDeleteLists(GLuint list, GLsizei range)
- void glDeleteProgram(GLuint program)
- void glDeleteQueries(GLsizei n, const GLuint * ids)
- void glDeleteShader(GLuint shader)
- void glDeleteTextures(GLsizei n, const GLuint * textures)
- void glDepthFunc(GLenum func)
- void glDepthMask(GLboolean flag)
- void glDepthRange(GLclampd nearVal, GLclampd farVal)
- void glDetachShader(GLuint program, GLuint shader)
- void glEnable(GLenum cap)
- void glEnableClientState(GLenum cap)
- void glEnableVertexAttribArray(GLuint index)
- void glDisableVertexAttribArray(GLuint index)
- void glDrawArrays(GLenum mode, GLint first, GLsizei count)
- void glDrawBuffer(GLenum mode)
- void glDrawBuffers(GLsizei n, const GLenum *bufs)
- void glDrawElements(GLenum mode, GLsizei count, GLenum type, const GLvoid * indices)
- void glDrawPixels(GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)
- void glDrawRangeElements(GLenum mode, GLuint start, GLuint end, GLsizei count, GLenum type, const GLvoid * indices)
- void glEdgeFlag(GLboolean flag)
- void glEdgeFlagPointer(GLsizei stride, const GLvoid * pointer)

- void glEnd(void)
- void glEndList(void)
- void glEndQuery(GLenum target)
- void glEvalCoord1f(GLfloat u)
- void glEvalCoord1d(GLdouble u)
- void glEvalCoord2f(GLfloat u,GLfloat v)
- void glEvalCoord2d(GLdouble u,GLdouble v)
- void glEvalMesh1(GLenum mode,GLint i1,GLint i2)
- void glEvalPoint1(GLint i)
- void glEvalPoint2(GLint i,GLint j)
- void glFeedbackBuffer(GLsizei size,GLenum type,GLfloat * buffer)
- void glFinish(void)
- void glFlush(void)
- void glFogf(GLenum pname,GLfloat param)
- void glFogi(GLenum pname,GLint param)
- void glFogfv(GLenum pname,const GLfloat * params)
- void glFogiv(GLenum pname,const GLint * params)
- void glFogCoordd(GLdouble coord)
- void glFogCoordf(GLfloat coord)
- void glFogCoorddv(GLdouble * coord)
- void glFogCoordfv(GLfloat * coord)
- void glFogCoordPointer(GLenum type,GLsizei stride,GLvoid * pointer)
- void glFrontFace(GLenum mode)
- void glFrustum(GLdouble left,GLdouble right,GLdouble bottom,GLdouble top,GLdouble nearVal,GLdouble farVal)
- void glGenBuffers(GLsizei n,GLuint * buffers)
- GLuint glGenLists(GLsizei range)
- void glGenQueries(GLsizei n,GLuint * ids)
- void glGenTextures(GLsizei n,GLuint * textures)
- void glGetBooleanv(GLenum pname,GLboolean * params)
- void glGetDoublev(GLenum pname,GLdouble * params)
- void glGetFloatv(GLenum pname,GLfloat * params)
- void glGetIntegerv(GLenum pname,GLint * params)
- void glGetActiveAttrib(GLuint program,GLuint index,GLsizei bufSize,GLsizei *length,GLint *size,GLenum *type,GLchar *name)
- void glGetActiveUniform(GLuint program,GLuint index,GLsizei bufSize,GLsizei *length,GLint *size,GLenum *type,GLchar *name)

- void glGetAttachedShaders(GLuint program, GLsizei maxCount, GLsizei *count, GLuint *shaders)
- GLint glGetAttribLocation(GLuint program, const GLchar *name)
- void glGetBufferParameteriv(GLenum target, GLenum value, GLint * data)
- void glGetBufferPointerv(GLenum target, GLenum pname, GLvoid ** params)
- void glGetBufferSubData(GLenum target, GLintptr offset, GLsizeiptr size, GLvoid * data)
- void glGetClipPlane(GLenum plane, GLdouble * equation)
- void glGetColorTable(GLenum target, GLenum format, GLenum type, GLvoid * table)
- void glGetColorTableParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetColorTableParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetCompressedTexImage(GLenum target, GLint lod, GLvoid * img)
- void glGetConvolutionFilter(GLenum target, GLenum format, GLenum type, GLvoid * image)
- void glGetConvolutionParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetConvolutionParameteriv(GLenum target, GLenum pname, GLint * params)
- GLenum glGetError(void)
- void glGetHistogram(GLenum target, GLboolean reset, GLenum format, GLenum type, GLvoid * values)
- void glGetHistogramParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetHistogramParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetLightfv(GLenum light, GLenum pname, GLfloat * params)
- void glGetLightiv(GLenum light, GLenum pname, GLint * params)
- void glGetMapdv(GLenum target, GLenum query, GLdouble * v)
- void glGetMapfv(GLenum target, GLenum query, GLfloat * v)
- void glGetMapiv(GLenum target, GLenum query, GLint * v)
- void glGetMaterialfv(GLenum face, GLenum pname, GLfloat * params)
- void glGetMaterialiv(GLenum face, GLenum pname, GLint * params)
- void glGetMinmax(GLenum target, GLboolean reset, GLenum format, GLenum types, GLvoid * values)
- void glGetMinmaxParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetMinmaxParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetPixelMapfv(GLenum map, GLfloat * data)
- void glGetPixelMapuiv(GLenum map, GLuint * data)
- void glGetPixelMapusv(GLenum map, GLushort * data)
- void glGetPointerv(GLenum pname, GLvoid ** params)
- void glGetPolygonStipple(GLubyte * pattern)
- void glGetProgramiv(GLuint program, GLenum pname, GLint *params)
- void glGetProgramInfoLog(GLuint program, GLsizei maxLength, GLsizei *length, GLchar *infoLog)
- void glGetQueryObjectiv(GLuint id, GLenum pname, GLint * params)
- void glGetQueryObjectuiv(GLuint id, GLenum pname, GLuint * params)

- void glGetQueryiv(GLenum target, GLenum pname, GLint * params)
- void glGetSeparableFilter(GLenum target, GLenum format, GLenum type, GLvoid * row, GLvoid * column, GLvoid * span)
- void glGetShaderiv(GLuint shader, GLenum pname, GLint *params)
- void glGetShaderInfoLog(GLuint shader, GLsizei maxLength, GLsizei *length, GLchar *infoLog)
- void glGetShaderSource(GLuint shader, GLsizei bufSize, GLsizei *length, GLchar *source)
- const GLubyte* glGetString(GLenum name)
- void glGetTexEnvfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetTexEnviv(GLenum target, GLenum pname, GLint * params)
- void glGetTexGendv(GLenum coord, GLenum pname, GLdouble * params)
- void glGetTexGenfv(GLenum coord, GLenum pname, GLfloat * params)
- void glGetTexGeniv(GLenum coord, GLenum pname, GLint * params)
- void glGetTexImage(GLenum target, GLint level, GLenum format, GLenum type, GLvoid * img)
- void glGetTexLevelParameterfv(GLenum target, GLint level, GLenum pname, GLfloat * params)
- void glGetTexLevelParameteriv(GLenum target, GLint level, GLenum pname, GLint * params)
- void glGetTexParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetTexParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetUniformfv(GLuint program, GLint location, GLfloat *params)
- void glGetUniformiv(GLuint program, GLint location, GLint *params)
- GLint glGetUniformLocation(GLuint program, const GLchar *name)
- void glGetVertexAttribdv(GLuint index, GLenum pname, GLdouble *params)
- void glGetVertexAttribfv(GLuint index, GLenum pname, GLfloat *params)
- void glGetVertexAttribiv(GLuint index, GLenum pname, GLint *params)
- void glGetVertexAttribPointerv(GLuint index, GLenum pname, GLvoid **pointer)
- void glHint(GLenum target, GLenum mode)
- void glHistogram(GLenum target, GLsizei width, GLenum internalformat, GLboolean sink)
- void glIndexs(GLshort c)
- void glIndexi(GLint c)
- void glIndexf(GLfloat c)
- void glIndexd(GLdouble c)
- void glIndexub(GLubyte c)
- void glIndexsv(const GLshort * c)
- void glIndexiv(const GLint * c)
- void glIndexfv(const GLfloat * c)
- void glIndexdv(const GLdouble * c)
- void glIndexubv(const GLubyte * c)

- `void glIndexMask(GLuint mask)`
- `void glIndexPointer(GLenum type, GLsizei stride, const GLvoid * pointer)`
- `void glInitNames(void)`
- `void glInterleavedArrays(GLenum format, GLsizei stride, const GLvoid * pointer)`
- `GLboolean glIsBuffer(GLuint buffer)`
- `GLboolean glIsEnabled(GLenum cap)`
- `GLboolean glIsList(GLuint list)`
- `GLboolean glIsProgram(GLuint program)`
- `GLboolean glIsQuery(GLuint id)`
- `GLboolean glIsShader(GLuint shader)`
- `GLboolean glIsTexture(GLuint texture)`
- `void glLightf(GLenum light, GLenum pname, GLfloat param)`
- `void glLighti(GLenum light, GLenum pname, GLint param)`
- `void glLightfv(GLenum light, GLenum pname, const GLfloat * params)`
- `void glLightiv(GLenum light, GLenum pname, const GLint * params)`
- `void glLightModelf(GLenum pname, GLfloat param)`
- `void glLightModeli(GLenum pname, GLint param)`
- `void glLightModelfv(GLenum pname, const GLfloat * params)`
- `void glLightModeliv(GLenum pname, const GLint * params)`
- `void glLineStipple(GLint factor, GLushort pattern)`
- `void glLineWidth(GLfloat width)`
- `void glLinkProgram(GLuint program)`
- `void glListBase(GLuint base)`
- `void glLoadIdentity(void)`
- `void glLoadMatrixd(const GLdouble * m)`
- `void glLoadMatrixf(const GLfloat * m)`
- `void glLoadName(GLuint name)`
- `void glLoadTransposeMatrixd(const GLdouble * m)`
- `void glLoadTransposeMatrixf(const GLfloat * m)`
- `void glLogicOp(GLenum opcode)`
- `void glMap1f(GLenum target, GLfloat u1, GLfloat u2, GLint stride, GLint order, const GLfloat * points)`
- `void glMap1d(GLenum target, GLdouble u1, GLdouble u2, GLint stride, GLint order, const GLdouble * points)`
- `void glMap2f(GLenum target, GLfloat u1, GLfloat u2, GLint ustride, GLint uorder, GLfloat v1, GLfloat v2, GLint vstride, GLint vorder, const GLfloat * points)`
- `void glMap2d(GLenum target, GLdouble u1, GLdouble u2, GLint ustride, GLint uorder, GLdouble v1, GLdouble v2, GLint vstride, GLint vorder, const GLdouble * points)`
- `void * glMapBuffer(GLenum target, GLenum access)`

- `void glMapGrid1d(GLint un,GLdouble u1,GLdouble u2)`
- `void glMapGrid1f(GLint un,GLfloat u1,GLfloat u2)`
- `void glMapGrid2d(GLint un,GLdouble u1,GLdouble u2,GLint vn,GLdouble v1,GLdouble v2)`
- `void glMapGrid2f(GLint un,GLfloat u1,GLfloat u2,GLint vn,GLfloat v1,GLfloat v2)`
- `void glMaterialf(GLenum face,GLenum pname,GLfloat param)`
- `void glMateriali(GLenum face,GLenum pname,GLint param)`
- `void glMatrixMode(GLenum mode)`
- `void glMinmax(GLenum target,GLenum internalformat,GLboolean sink)`
- `void glMultMatrixd(const GLdouble * m)`
- `void glMultMatrixf(const GLfloat * m)`
- `void glMultTransposeMatrixd(const GLdouble * m)`
- `void glMultTransposeMatrixf(const GLfloat * m)`
- `void glMultiDrawArrays(GLenum mode,GLint * first,GLsizei * count,GLsizei primcount)`
- `void glMultiDrawElements(GLenum mode,const GLsizei * count,GLenum type,const GLvoid ** indices,GLsizei primcount)`
- `void glMultiTexCoord1s(GLenum target,GLshort s)`
- `void glMultiTexCoord1i(GLenum target,GLint s)`
- `void glMultiTexCoord1f(GLenum target,GLfloat s)`
- `void glMultiTexCoord1d(GLenum target,GLdouble s)`
- `void glMultiTexCoord2s(GLenum target,GLshort s,GLshort t)`
- `void glMultiTexCoord2i(GLenum target,GLint s,GLint t)`
- `void glMultiTexCoord2f(GLenum target,GLfloat s,GLfloat t)`
- `void glMultiTexCoord2d(GLenum target,GLdouble s,GLdouble t)`
- `void glMultiTexCoord3s(GLenum target,GLshort s,GLshort t,GLshort r)`
- `void glMultiTexCoord3i(GLenum target,GLint s,GLint t,GLint r)`
- `void glMultiTexCoord3f(GLenum target,GLfloat s,GLfloat t,GLfloat r)`
- `void glMultiTexCoord3d(GLenum target,GLdouble s,GLdouble t,GLdouble r)`
- `void glMultiTexCoord4s(GLenum target,GLshort s,GLshort t,GLshort r,GLshort q)`
- `void glMultiTexCoord4i(GLenum target,GLint s,GLint t,GLint r,GLint q)`
- `void glMultiTexCoord4f(GLenum target,GLfloat s,GLfloat t,GLfloat r,GLfloat q)`
- `void glMultiTexCoord4d(GLenum target,GLdouble s,GLdouble t,GLdouble r,GLdouble q)`
- `void glMultiTexCoord1sv(GLenum target,const GLshort * v)`
- `void glMultiTexCoord1iv(GLenum target,const GLint * v)`
- `void glMultiTexCoord1fv(GLenum target,const GLfloat * v)`
- `void glMultiTexCoord1dv(GLenum target,const GLdouble * v)`
- `void glMultiTexCoord2sv(GLenum target,const GLshort * v)`

- void glMultiTexCoord2iv(GLenum target,const GLint * v)
- void glMultiTexCoord2fv(GLenum target,const GLfloat * v)
- void glMultiTexCoord2dv(GLenum target,const GLdouble * v)
- void glMultiTexCoord3sv(GLenum target,const GLshort * v)
- void glMultiTexCoord3iv(GLenum target,const GLint * v)
- void glMultiTexCoord3fv(GLenum target,const GLfloat * v)
- void glMultiTexCoord3dv(GLenum target,const GLdouble * v)
- void glMultiTexCoord4sv(GLenum target,const GLshort * v)
- void glMultiTexCoord4iv(GLenum target,const GLint * v)
- void glMultiTexCoord4fv(GLenum target,const GLfloat * v)
- void glMultiTexCoord4dv(GLenum target,const GLdouble * v)
- void glNewList(GLuint list,GLenum mode)
- void glNormal3b(GLbyte nx,GLbyte ny,GLbyte nz)
- void glNormal3d(GLdouble nx,GLdouble ny,GLdouble nz)
- void glNormal3f(GLfloat nx,GLfloat ny,GLfloat nz)
- void glNormal3i(GLint nx,GLint ny,GLint nz)
- void glNormal3s(GLshort nx,GLshort ny,GLshort nz)
- void glNormal3bv(const GLbyte * v)
- void glNormal3dv(const GLdouble * v)
- void glNormal3fv(const GLfloat * v)
- void glNormal3iv(const GLint * v)
- void glNormal3sv(const GLshort * v)
- void glNormalPointer(GLenum type,GLsizei stride,const GLvoid * pointer)
- void glOrtho(GLdouble left,GLdouble right,GLdouble bottom,GLdouble top,GLdouble nearVal,GLdouble farVal)
- void glPassThrough(GLfloat token)
- void glPixelMapfv(GLenum map,GLsizei mapsize,const GLfloat * values)
- void glPixelMapuiv(GLenum map,GLsizei mapsize,const GLuint * values)
- void glPixelMapusv(GLenum map,GLsizei mapsize,const GLushort * values)
- void glPixelStoref(GLenum pname,GLfloat param)
- void glPixelStorei(GLenum pname,GLint param)
- void glPixelTransferf(GLenum pname,GLfloat param)
- void glPixelTransferi(GLenum pname,GLint param)
- void glPixelZoom(GLfloat xfactor,GLfloat yfactor)
- void glPointParameterf(GLenum pname,GLfloat param)
- void glPointParameteri(GLenum pname,GLint param)

- void glPointSize(GLfloat size)
- void glPolygonMode(GLenum face, GLenum mode)
- void glPolygonOffset(GLfloat factor, GLfloat units)
- void glPolygonStipple(const GLubyte * pattern)
- void glPushAttrib(GLbitfield mask)
- void glPushClientAttrib(GLbitfield mask)
- void glPushMatrix(void)
- void glPushName(GLuint name)
- void glPrioritizeTextures(GLsizei n, const GLuint * textures, const GLclampf * priorities)
- void glPopMatrix(void)
- void glRasterPos2s(GLshort x, GLshort y)
- void glRasterPos2i(GLint x, GLint y)
- void glRasterPos2f(GLfloat x, GLfloat y)
- void glRasterPos2d(GLdouble x, GLdouble y)
- void glRasterPos3s(GLshort x, GLshort y, GLshort z)
- void glRasterPos3i(GLint x, GLint y, GLint z)
- void glRasterPos3f(GLfloat x, GLfloat y, GLfloat z)
- void glRasterPos3d(GLdouble x, GLdouble y, GLdouble z)
- void glRasterPos4s(GLshort x, GLshort y, GLshort z, GLshort w)
- void glRasterPos4i(GLint x, GLint y, GLint z, GLint w)
- void glRasterPos4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- void glRasterPos4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)
- void glReadBuffer(GLenum mode)
- void glReadPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum format, GLenum type, GLvoid * data)
- void glRectd(GLdouble x1, GLdouble y1, GLdouble x2, GLdouble y2)
- void glRectf(GLfloat x1, GLfloat y1, GLfloat x2, GLfloat y2)
- void glRecti(GLint x1, GLint y1, GLint x2, GLint y2)
- void glRects(GLshort x1, GLshort y1, GLshort x2, GLshort y2)
- void glRectdv(const GLdouble * v1, const GLdouble * v2)
- void glRectfv(const GLfloat * v1, const GLfloat * v2)
- void glRectiv(const GLint * v1, const GLint * v2)
- void glRectsv(const GLshort * v1, const GLshort * v2)
- GLint glRenderMode(GLenum mode)
- void glResetHistogram(GLenum target)
- void glRotated(GLdouble angle, GLdouble x, GLdouble y, GLdouble z)
- void glRotatef(GLfloat angle, GLfloat x, GLfloat y, GLfloat z)

- void glSampleCoverage(GLclampf value, GLboolean invert)
- void glScaled(GLdouble x, GLdouble y, GLdouble z)
- void glScalef(GLfloat x, GLfloat y, GLfloat z)
- void glScissor(GLint x, GLint y, GLsizei width, GLsizei height)
- void glSecondaryColor3b(GLbyte red, GLbyte green, GLbyte blue)
- void glSecondaryColor3s(GLshort red, GLshort green, GLshort blue)
- void glSecondaryColor3i(GLint red, GLint green, GLint blue)
- void glSecondaryColor3f(GLfloat red, GLfloat green, GLfloat blue)
- void glSecondaryColor3d(GLdouble red, GLdouble green, GLdouble blue)
- void glSecondaryColor3ub(GLubyte red, GLubyte green, GLubyte blue)
- void glSecondaryColor3us(GLushort red, GLushort green, GLushort blue)
- void glSecondaryColor3ui(GLuint red, GLuint green, GLuint blue)
- void glSecondaryColor3bv(const GLbyte * v)
- void glSecondaryColor3sv(const GLshort * v)
- void glSecondaryColor3iv(const GLint * v)
- void glSecondaryColor3fv(const GLfloat * v)
- void glSecondaryColor3dv(const GLdouble * v)
- void glSecondaryColor3ubv(const GLubyte * v)
- void glSecondaryColor3usv(const GLushort * v)
- void glSecondaryColor3uiv(const GLuint * v)
- void glSecondaryColorPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)
- void glSelectBuffer(GLsizei size, GLuint * buffer)
- void glSeparableFilter2D(GLenum target, GLenum internalformat, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * row, const GLvoid * column)
- void glShadeModel(GLenum mode)
- void glShaderSource(GLuint shader, GLsizei count, const GLchar **string, const GLint *length)
- void glStencilFunc(GLenum func, GLint ref, GLuint mask)
- void glStencilFuncSeparate(GLenum face, GLenum func, GLint ref, GLuint mask)
- void glStencilMask(GLuint mask)
- void glStencilMaskSeparate(GLenum face, GLuint mask)
- void glStencilOp(GLenum sfail, GLenum dpfail, GLenum dppass)
- void glStencilOpSeparate(GLenum face, GLenum sfail, GLenum dpfail, GLenum dppass)
- void glTexCoord1s(GLshort s)
- void glTexCoord1i(GLint s)
- void glTexCoord1f(GLfloat s)
- void glTexCoord1d(GLdouble s)

- `void glTexCoord2s(GLshort s, GLshort t)`
- `void glTexCoord2i(GLint s, GLint t)`
- `void glTexCoord2f(GLfloat s, GLfloat t)`
- `void glTexCoord2d(GLdouble s, GLdouble t)`
- `void glTexCoord3s(GLshort s, GLshort t, GLshort r)`
- `void glTexCoord3i(GLint s, GLint t, GLint r)`
- `void glTexCoord3f(GLfloat s, GLfloat t, GLfloat r)`
- `void glTexCoord3d(GLdouble s, GLdouble t, GLdouble r)`
- `void glTexCoord4s(GLshort s, GLshort t, GLshort r, GLshort q)`
- `void glTexCoord4i(GLint s, GLint t, GLint r, GLint q)`
- `void glTexCoord4f(GLfloat s, GLfloat t, GLfloat r, GLfloat q)`
- `void glTexCoord4d(GLdouble s, GLdouble t, GLdouble r, GLdouble q)`
- `void glTexCoord1sv(const GLshort * v)`
- `void glTexCoord1iv(const GLint * v)`
- `void glTexCoord1fv(const GLfloat * v)`
- `void glTexCoord1dv(const GLdouble * v)`
- `void glTexCoord2sv(const GLshort * v)`
- `void glTexCoord2iv(const GLint * v)`
- `void glTexCoord2fv(const GLfloat * v)`
- `void glTexCoord2dv(const GLdouble * v)`
- `void glTexCoord3sv(const GLshort * v)`
- `void glTexCoord3iv(const GLint * v)`
- `void glTexCoord3fv(const GLfloat * v)`
- `void glTexCoord3dv(const GLdouble * v)`
- `void glTexCoord4sv(const GLshort * v)`
- `void glTexCoord4iv(const GLint * v)`
- `void glTexCoord4fv(const GLfloat * v)`
- `void glTexCoord4dv(const GLdouble * v)`
- `void glTexCoordPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)`
- `void glTexEnvf(GLenum target, GLenum pname, GLfloat param)`
- `void glTexEnvf(GLenum target, GLenum pname, GLint param)`
- `void glTexGeni(GLenum coord, GLenum pname, GLint param)`
- `void glTexGenf(GLenum coord, GLenum pname, GLfloat param)`
- `void glTexGend(GLenum coord, GLenum pname, GLdouble param)`
- `void glTexGeniv(GLenum coord, GLenum pname, const GLint * params)`
- `void glTexGenfv(GLenum coord, GLenum pname, const GLfloat * params)`

- `void glTexGendv(GLenum coord, GLenum pname, const GLdouble * params)`
- `void glTexImage1D(GLenum target, GLint level, GLint internalFormat, GLsizei width, GLint border, GLenum format, GLenum type, const GLvoid * data)`
- `void glTexImage2D(GLenum target, GLint level, GLint internalFormat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, const GLvoid * data)`
- `void glTexImage3D(GLenum target, GLint level, GLint internalFormat, GLsizei width, GLsizei height, GLsizei depth, GLint border, GLenum format, GLenum type, const GLvoid * data)`
- `void glTexParameterf(GLenum target, GLenum pname, GLfloat param)`
- `void glTexParameteri(GLenum target, GLenum pname, GLint param)`
- `void glTexParameterfv(GLenum target, GLenum pname, const GLfloat * params)`
- `void glTexParameteriv(GLenum target, GLenum pname, const GLint * params)`
- `void glTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLenum type, const GLvoid * data)`
- `void glTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)`
- `void glTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLenum type, const GLvoid * data)`
- `void glTranslated(GLdouble x, GLdouble y, GLdouble z)`
- `void glTranslatef(GLfloat x, GLfloat y, GLfloat z)`
- `void glUniform1f(GLint location, GLfloat v0)`
- `void glUniform2f(GLint location, GLfloat v0, GLfloat v1)`
- `void glUniform3f(GLint location, GLfloat v0, GLfloat v1, GLfloat v2)`
- `void glUniform4f(GLint location, GLfloat v0, GLfloat v1, GLfloat v2, GLfloat v3)`
- `void glUniform1i(GLint location, GLint v0)`
- `void glUniform2i(GLint location, GLint v0, GLint v1)`
- `void glUniform3i(GLint location, GLint v0, GLint v1, GLint v2)`
- `void glUniform4i(GLint location, GLint v0, GLint v1, GLint v2, GLint v3)`
- `void glUniform1fv(GLint location, GLsizei count, const GLfloat *value)`
- `void glUniform2fv(GLint location, GLsizei count, const GLfloat *value)`
- `void glUniform3fv(GLint location, GLsizei count, const GLfloat *value)`
- `void glUniform4fv(GLint location, GLsizei count, const GLfloat *value)`
- `void glUniform1iv(GLint location, GLsizei count, const GLint *value)`
- `void glUniform2iv(GLint location, GLsizei count, const GLint *value)`
- `void glUniform3iv(GLint location, GLsizei count, const GLint *value)`
- `void glUniform4iv(GLint location, GLsizei count, const GLint *value)`
- `void glUniformMatrix2fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix3fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix4fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`

- void glUniformMatrix2x3fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix3x2fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix2x4fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix4x2fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix3x4fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix4x3fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUseProgram(GLuint program)
- void glValidateProgram(GLuint program)
- void glVertex2s(GLshort x,GLshort y)
- void glVertex2i(GLint x,GLint y)
- void glVertex2f(GLfloat x,GLfloat y)
- void glVertex2d(GLdouble x,GLdouble y)
- void glVertex3s(GLshort x,GLshort y,GLshort z)
- void glVertex3i(GLint x,GLint y,GLint z)
- void glVertex3f(GLfloat x,GLfloat y,GLfloat z)
- void glVertex3d(GLdouble x,GLdouble y,GLdouble z)
- void glVertex4s(GLshort x,GLshort y,GLshort z,GLshort w)
- void glVertex4i(GLint x,GLint y,GLint z,GLint w)
- void glVertex4f(GLfloat x,GLfloat y,GLfloat z,GLfloat w)
- void glVertex4d(GLdouble x,GLdouble y,GLdouble z,GLdouble w)
- void glVertex2sv(const GLshort * v)
- void glVertex2iv(const GLint * v)
- void glVertex2fv(const GLfloat * v)
- void glVertex2dv(const GLdouble * v)
- void glVertex3sv(const GLshort * v)
- void glVertex3iv(const GLint * v)
- void glVertex3fv(const GLfloat * v)
- void glVertex3dv(const GLdouble * v)
- void glVertex4sv(const GLshort * v)
- void glVertex4iv(const GLint * v)
- void glVertex4fv(const GLfloat * v)
- void glVertex4dv(const GLdouble * v)
- void glVertexAttrib1f(GLuint index,GLfloat v0)
- void glVertexAttrib1s(GLuint index,GLshort v0)
- void glVertexAttrib1d(GLuint index,GLdouble v0)
- void glVertexAttrib2f(GLuint index,GLfloat v0,GLfloat v1)

- void glVertexAttrib2s(GLuint index, GLshort v0, GLshort v1)
- void glVertexAttrib2d(GLuint index, GLdouble v0, GLdouble v1)
- void glVertexAttrib3f(GLuint index, GLfloat v0, GLfloat v1, GLfloat v2)
- void glVertexAttrib3s(GLuint index, GLshort v0, GLshort v1, GLshort v2)
- void glVertexAttrib3d(GLuint index, GLdouble v0, GLdouble v1, GLdouble v2)
- void glVertexAttrib4f(GLuint index, GLfloat v0, GLfloat v1, GLfloat v2, GLfloat v3)
- void glVertexAttrib4s(GLuint index, GLshort v0, GLshort v1, GLshort v2, GLshort v3)
- void glVertexAttrib4d(GLuint index, GLdouble v0, GLdouble v1, GLdouble v2, GLdouble v3)
- void glVertexAttrib4Nub(GLuint index, GLubyte v0, GLubyte v1, GLubyte v2, GLubyte v3)
- void glVertexAttrib1fv(GLuint index, const GLfloat *v)
- void glVertexAttrib1sv(GLuint index, const GLshort *v)
- void glVertexAttrib1dv(GLuint index, const GLdouble *v)
- void glVertexAttrib2fv(GLuint index, const GLfloat *v)
- void glVertexAttrib2sv(GLuint index, const GLshort *v)
- void glVertexAttrib2dv(GLuint index, const GLdouble *v)
- void glVertexAttrib3fv(GLuint index, const GLfloat *v)
- void glVertexAttrib3sv(GLuint index, const GLshort *v)
- void glVertexAttrib3dv(GLuint index, const GLdouble *v)
- void glVertexAttrib4fv(GLuint index, const GLfloat *v)
- void glVertexAttrib4sv(GLuint index, const GLshort *v)
- void glVertexAttrib4dv(GLuint index, const GLdouble *v)
- void glVertexAttrib4iv(GLuint index, const GLint *v)
- void glVertexAttrib4bv(GLuint index, const GLbyte *v)
- void glVertexAttrib4ubv(GLuint index, const GLubyte *v)
- void glVertexAttrib4usv(GLuint index, const GLushort *v)
- void glVertexAttrib4uiv(GLuint index, const GLuint *v)
- void glVertexAttribPointer(GLuint index, GLint size, GLenum type, GLboolean normalized, GLsizei stride, const GLvoid * pointer)
- void glVertexPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)
- void glViewport(GLint x, GLint y, GLsizei width, GLsizei height)
- void glWindowPos2s(GLshort x, GLshort y)
- void glWindowPos2i(GLint x, GLint y)
- void glWindowPos2f(GLfloat x, GLfloat y)
- void glWindowPos2d(GLdouble x, GLdouble y)
- void glWindowPos3s(GLshort x, GLshort y, GLshort z)
- void glWindowPos3i(GLint x, GLint y, GLint z)

- void glWindowPos3f(GLfloat x,GLfloat y,GLfloat z)
- void glWindowPos3d(GLdouble x,GLdouble y,GLdouble z)
- void glWindowPos2sv(const GLshort * v)
- void glWindowPos2iv(const GLint * v)
- void glWindowPos2fv(const GLfloat * v)
- void glWindowPos2dv(const GLdouble * v)
- void glWindowPos3sv(const GLshort * v)
- void glWindowPos3iv(const GLint * v)
- void glWindowPos3fv(const GLfloat * v)
- void glWindowPos3dv(const GLdouble * v)
- void gluBeginCurve(GLUnurbs* nurb)
- void gluBeginPolygon(GLUtesselator* tess)
- void gluBeginSurface(GLUnurbs* nurb)
- void gluBeginTrim(GLUnurbs* nurb)
- void gluCylinder(GLUquadric* quad,GLdouble base,GLdouble top,GLdouble height,GLint slices,GLint stacks)
- void gluDeleteNurbsRenderer(GLUnurbs* nurb)
- void gluDeleteQuadric(GLUquadric* quad)
- void gluDeleteTess(GLUtesselator* tess)
- void gluDisk(GLUquadric* quad,GLdouble inner,GLdouble outer,GLint slices,GLint loops)
- void gluEndCurve(GLUnurbs* nurb)
- void gluEndPolygon(GLUtesselator* tess)
- void gluEndSurface(GLUnurbs* nurb)
- void gluEndTrim(GLUnurbs* nurb)
- const GLubyte * gluErrorString(GLenum error)
- void gluGetNurbsProperty(GLUnurbs* nurb,GLenum property,GLfloat* data)
- const GLubyte * gluGetString(GLenum name)
- void gluGetTessProperty(GLUtesselator* tess,GLenum which,GLdouble* data)
- void gluLoadSamplingMatrices(GLUnurbs* nurb,const GLfloat * model,const GLfloat * perspective,const GLint * view)
- void gluLookAt(GLdouble eyeX,GLdouble eyeY,GLdouble eyeZ,GLdouble centerX,GLdouble centerY,GLdouble centerZ,GLdouble upX,GLdouble upY,GLdouble upZ)
- GLUnurbs *gluNewNurbsRenderer(void)
- GLUquadric *gluNewQuadric(void)
- GLUtesselator* gluNewTess(void)
- void gluNextContour(GLUtesselator* tess,GLenum type)
- void gluNurbsCurve(GLUnurbs* nurb,GLint knotCount,GLfloat * knots,GLint stride,GLfloat * control,GLint order,GLenum type)

- void gluNurbsProperty(GLUnurbs* nurb, GLenum property, GLfloat value)
- void gluNurbsSurface(GLUnurbs* nurb, GLint sKnotCount, GLfloat* sKnots, GLint tKnotCount, GLfloat* tKnots, GLint sStride, GLint tStride, GLfloat* control, GLint sOrder, GLint tOrder, GLenum type)
- void gluOrtho2D(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top)
- void gluPartialDisk(GLUquadric* quad, GLdouble inner, GLdouble outer, GLint slices, GLint loops, GLdouble start, GLdouble sweep)
- void gluPerspective(GLdouble fovy, GLdouble aspect, GLdouble zNear, GLdouble zFar)
- void gluPickMatrix(GLdouble x, GLdouble y, GLdouble delX, GLdouble delY, GLint * viewport)
- GLint gluProject(GLdouble objX, GLdouble objY, GLdouble objZ, const GLdouble * model, const GLdouble * proj, const GLint * view, GLdouble* winX, GLdouble* winY, GLdouble* winZ)
- void gluPwlCurve(GLUnurbs* nurb, GLint count, GLfloat* data, GLint stride, GLenum type)
- void gluQuadricDrawStyle(GLUquadric* quad, GLenum draw)
- void gluQuadricNormals(GLUquadric* quad, GLenum normal)
- void gluQuadricOrientation(GLUquadric* quad, GLenum orientation)
- void gluQuadricTexture(GLUquadric* quad, GLboolean texture)
- GLint gluScaleImage(GLenum format, GLsizei wIn, GLsizei hIn, GLenum typeIn, const void * dataIn, GLsizei wOut, GLsizei hOut, GLenum typeOut, GLvoid* dataOut)
- void gluSphere(GLUquadric* quad, GLdouble radius, GLint slices, GLint stacks)
- void gluTessBeginContour(GLUtesselator* tess)
- void gluTessBeginPolygon(GLUtesselator* tess, GLvoid* data)
- void gluTessEndContour(GLUtesselator* tess)
- void gluTessEndPolygon(GLUtesselator* tess)
- void gluTessNormal(GLUtesselator* tess, GLdouble valueX, GLdouble valueY, GLdouble valueZ)
- void gluTessProperty(GLUtesselator* tess, GLenum which, GLdouble data)
- void gluTessVertex(GLUtesselator* tess, GLdouble * location, GLvoid* data)
- GLint gluUnProject(GLdouble winX, GLdouble winY, GLdouble winZ, const GLdouble * model, const GLdouble * proj, const GLint * view, GLdouble* objX, GLdouble* objY, GLdouble* objZ)
- void glDisable(GLenum cap)

RINGOPENGL (OPENGL 4.4) FUNCTIONS REFERENCE

- GL_ZERO
- GL_FALSE
- GL_LOGIC_OP
- GL_NONE
- GL_TEXTURE_COMPONENTS
- GL_NO_ERROR
- GL_POINTS
- GL_CURRENT_BIT
- GL_TRUE
- GL_ONE
- GL_CLIENT_PIXEL_STORE_BIT
- GL_LINES
- GL_LINE_LOOP
- GL_POINT_BIT
- GL_CLIENT_VERTEX_ARRAY_BIT
- GL_LINE_STRIP
- GL_LINE_BIT
- GL_TRIANGLES
- GL_TRIANGLE_STRIP
- GL_TRIANGLE_FAN
- GL_QUADS
- GL_QUAD_STRIP
- GL_POLYGON_BIT
- GL_POLYGON
- GL_POLYGON_STIPPLE_BIT
- GL_PIXEL_MODE_BIT
- GL_LIGHTING_BIT

- GL_FOG_BIT
- GL_DEPTH_BUFFER_BIT
- GL_ACCUM
- GL_LOAD
- GL_RETURN
- GL_MULT
- GL_ADD
- GL_NEVER
- GL_ACCUM_BUFFER_BIT
- GL_LESS
- GL_EQUAL
- GL_LEQUAL
- GL_GREATER
- GL_NOTEQUAL
- GL_GEQUAL
- GL_ALWAYS
- GL_SRC_COLOR
- GL_ONE_MINUS_SRC_COLOR
- GL_SRC_ALPHA
- GL_ONE_MINUS_SRC_ALPHA
- GL_DST_ALPHA
- GL_ONE_MINUS_DST_ALPHA
- GL_DST_COLOR
- GL_ONE_MINUS_DST_COLOR
- GL_SRC_ALPHA_SATURATE
- GL_STENCIL_BUFFER_BIT
- GL_FRONT_LEFT
- GL_FRONT_RIGHT
- GL_BACK_LEFT
- GL_BACK_RIGHT
- GL_FRONT
- GL_BACK
- GL_LEFT
- GL_RIGHT
- GL_FRONT_AND_BACK
- GL_AUX0

- GL_AUX1
- GL_AUX2
- GL_AUX3
- GL_INVALID_ENUM
- GL_INVALID_VALUE
- GL_INVALID_OPERATION
- GL_STACK_OVERFLOW
- GL_STACK_UNDERFLOW
- GL_OUT_OF_MEMORY
- GL_2D
- GL_3D
- GL_3D_COLOR
- GL_3D_COLOR_TEXTURE
- GL_4D_COLOR_TEXTURE
- GL_PASS_THROUGH_TOKEN
- GL_POINT_TOKEN
- GL_LINE_TOKEN
- GL_POLYGON_TOKEN
- GL_BITMAP_TOKEN
- GL_DRAW_PIXEL_TOKEN
- GL_COPY_PIXEL_TOKEN
- GL_LINE_RESET_TOKEN
- GL_EXP
- GL_VIEWPORT_BIT
- GL_EXP2
- GL_CW
- GL_CCW
- GL_COEFF
- GL_ORDER
- GL_DOMAIN
- GL_CURRENT_COLOR
- GL_CURRENT_INDEX
- GL_CURRENT_NORMAL
- GL_CURRENT_TEXTURE_COORDS
- GL_CURRENT_RASTER_COLOR
- GL_CURRENT_RASTER_INDEX

- GL_CURRENT_RASTER_TEXTURE_COORDS
- GL_CURRENT_RASTER_POSITION
- GL_CURRENT_RASTER_POSITION_VALID
- GL_CURRENT_RASTER_DISTANCE
- GL_POINT_SMOOTH
- GL_POINT_SIZE
- GL_POINT_SIZE_RANGE
- GL_POINT_SIZE_GRANULARITY
- GL_LINE_SMOOTH
- GL_LINE_WIDTH
- GL_LINE_WIDTH_RANGE
- GL_LINE_WIDTH_GRANULARITY
- GL_LINE_STIPPLE
- GL_LINE_STIPPLE_PATTERN
- GL_LINE_STIPPLE_REPEAT
- GL_LIST_MODE
- GL_MAX_LIST_NESTING
- GL_LIST_BASE
- GL_LIST_INDEX
- GL_POLYGON_MODE
- GL_POLYGON_SMOOTH
- GL_POLYGON_STIPPLE
- GL_EDGE_FLAG
- GL_CULL_FACE
- GL_CULL_FACE_MODE
- GL_FRONT_FACE
- GL_LIGHTING
- GL_LIGHT_MODEL_LOCAL_VIEWER
- GL_LIGHT_MODEL_TWO_SIDE
- GL_LIGHT_MODEL_AMBIENT
- GL_SHADE_MODEL
- GL_COLOR_MATERIAL_FACE
- GL_COLOR_MATERIAL_PARAMETER
- GL_COLOR_MATERIAL
- GL_FOG
- GL_FOG_INDEX

- GL_FOG_DENSITY
- GL_FOG_START
- GL_FOG_END
- GL_FOG_MODE
- GL_FOG_COLOR
- GL_DEPTH_RANGE
- GL_DEPTH_TEST
- GL_DEPTH_WRITEMASK
- GL_DEPTH_CLEAR_VALUE
- GL_DEPTH_FUNC
- GL_ACCUM_CLEAR_VALUE
- GL_STENCIL_TEST
- GL_STENCIL_CLEAR_VALUE
- GL_STENCIL_FUNC
- GL_STENCIL_VALUE_MASK
- GL_STENCIL_FAIL
- GL_STENCIL_PASS_DEPTH_FAIL
- GL_STENCIL_PASS_DEPTH_PASS
- GL_STENCIL_REF
- GL_STENCIL_WRITEMASK
- GL_MATRIX_MODE
- GL_NORMALIZE
- GL_VIEWPORT
- GL_MODELVIEW_STACK_DEPTH
- GL_PROJECTION_STACK_DEPTH
- GL_TEXTURE_STACK_DEPTH
- GL_MODELVIEW_MATRIX
- GL_PROJECTION_MATRIX
- GL_TEXTURE_MATRIX
- GL_ATTRIB_STACK_DEPTH
- GL_CLIENT_ATTRIB_STACK_DEPTH
- GL_ALPHA_TEST
- GL_ALPHA_TEST_FUNC
- GL_ALPHA_TEST_REF
- GL_DITHER
- GL_BLEND_DST

- GL_BLEND_SRC
- GL_BLEND
- GL_LOGIC_OP_MODE
- GL_INDEX_LOGIC_OP
- GL_COLOR_LOGIC_OP
- GL_AUX_BUFFERS
- GL_DRAW_BUFFER
- GL_READ_BUFFER
- GL_SCISSOR_BOX
- GL_SCISSOR_TEST
- GL_INDEX_CLEAR_VALUE
- GL_INDEX_WRITEMASK
- GL_COLOR_CLEAR_VALUE
- GL_COLOR_WRITEMASK
- GL_INDEX_MODE
- GL_RGBA_MODE
- GL_DOUBLEBUFFER
- GL_STEREO
- GL_RENDER_MODE
- GL_PERSPECTIVE_CORRECTION_HINT
- GL_POINT_SMOOTH_HINT
- GL_LINE_SMOOTH_HINT
- GL_POLYGON_SMOOTH_HINT
- GL_FOG_HINT
- GL_TEXTURE_GEN_S
- GL_TEXTURE_GEN_T
- GL_TEXTURE_GEN_R
- GL_TEXTURE_GEN_Q
- GL_PIXEL_MAP_I_TO_I
- GL_PIXEL_MAP_S_TO_S
- GL_PIXEL_MAP_I_TO_R
- GL_PIXEL_MAP_I_TO_G
- GL_PIXEL_MAP_I_TO_B
- GL_PIXEL_MAP_I_TO_A
- GL_PIXEL_MAP_R_TO_R
- GL_PIXEL_MAP_G_TO_G

- GL_PIXEL_MAP_B_TO_B
- GL_PIXEL_MAP_A_TO_A
- GL_PIXEL_MAP_I_TO_I_SIZE
- GL_PIXEL_MAP_S_TO_S_SIZE
- GL_PIXEL_MAP_I_TO_R_SIZE
- GL_PIXEL_MAP_I_TO_G_SIZE
- GL_PIXEL_MAP_I_TO_B_SIZE
- GL_PIXEL_MAP_I_TO_A_SIZE
- GL_PIXEL_MAP_R_TO_R_SIZE
- GL_PIXEL_MAP_G_TO_G_SIZE
- GL_PIXEL_MAP_B_TO_B_SIZE
- GL_PIXEL_MAP_A_TO_A_SIZE
- GL_UNPACK_SWAP_BYTES
- GL_UNPACK_LSB_FIRST
- GL_UNPACK_ROW_LENGTH
- GL_UNPACK_SKIP_ROWS
- GL_UNPACK_SKIP_PIXELS
- GL_UNPACK_ALIGNMENT
- GL_PACK_SWAP_BYTES
- GL_PACK_LSB_FIRST
- GL_PACK_ROW_LENGTH
- GL_PACK_SKIP_ROWS
- GL_PACK_SKIP_PIXELS
- GL_PACK_ALIGNMENT
- GL_MAP_COLOR
- GL_MAP_STENCIL
- GL_INDEX_SHIFT
- GL_INDEX_OFFSET
- GL_RED_SCALE
- GL_RED_BIAS
- GL_ZOOM_X
- GL_ZOOM_Y
- GL_GREEN_SCALE
- GL_GREEN_BIAS
- GL_BLUE_SCALE
- GL_BLUE_BIAS

- GL_ALPHA_SCALE
- GL_ALPHA_BIAS
- GL_DEPTH_SCALE
- GL_DEPTH_BIAS
- GL_MAX_EVAL_ORDER
- GL_MAX_LIGHTS
- GL_MAX_CLIP_PLANES
- GL_MAX_TEXTURE_SIZE
- GL_MAX_PIXEL_MAP_TABLE
- GL_MAX_ATTRIB_STACK_DEPTH
- GL_MAX_MODELVIEW_STACK_DEPTH
- GL_MAX_NAME_STACK_DEPTH
- GL_MAX_PROJECTION_STACK_DEPTH
- GL_MAX_TEXTURE_STACK_DEPTH
- GL_MAX_VIEWPORT_DIMS
- GL_MAX_CLIENT_ATTRIB_STACK_DEPTH
- GL_SUBPIXEL_BITS
- GL_INDEX_BITS
- GL_RED_BITS
- GL_GREEN_BITS
- GL_BLUE_BITS
- GL_ALPHA_BITS
- GL_DEPTH_BITS
- GL_STENCIL_BITS
- GL_ACCUM_RED_BITS
- GL_ACCUM_GREEN_BITS
- GL_ACCUM_BLUE_BITS
- GL_ACCUM_ALPHA_BITS
- GL_NAME_STACK_DEPTH
- GL_AUTO_NORMAL
- GL_MAP1_COLOR_4
- GL_MAP1_INDEX
- GL_MAP1_NORMAL
- GL_MAP1_TEXTURE_COORD_1
- GL_MAP1_TEXTURE_COORD_2
- GL_MAP1_TEXTURE_COORD_3

- GL_MAP1_TEXTURE_COORD_4
- GL_MAP1_VERTEX_3
- GL_MAP1_VERTEX_4
- GL_MAP2_COLOR_4
- GL_MAP2_INDEX
- GL_MAP2_NORMAL
- GL_MAP2_TEXTURE_COORD_1
- GL_MAP2_TEXTURE_COORD_2
- GL_MAP2_TEXTURE_COORD_3
- GL_MAP2_TEXTURE_COORD_4
- GL_MAP2_VERTEX_3
- GL_MAP2_VERTEX_4
- GL_MAP1_GRID_DOMAIN
- GL_MAP1_GRID_SEGMENTS
- GL_MAP2_GRID_DOMAIN
- GL_MAP2_GRID_SEGMENTS
- GL_TEXTURE_1D
- GL_TEXTURE_2D
- GL_FEEDBACK_BUFFER_POINTER
- GL_FEEDBACK_BUFFER_SIZE
- GL_FEEDBACK_BUFFER_TYPE
- GL_SELECTION_BUFFER_POINTER
- GL_SELECTION_BUFFER_SIZE
- GL_TEXTURE_WIDTH
- GL_TRANSFORM_BIT
- GL_TEXTURE_HEIGHT
- GL_TEXTURE_INTERNAL_FORMAT
- GL_TEXTURE_BORDER_COLOR
- GL_TEXTURE_BORDER
- GL_DONT_CARE
- GL_FASTEST
- GL_NICEST
- GL_AMBIENT
- GL_DIFFUSE
- GL_SPECULAR
- GL_POSITION

- GL_SPOT_DIRECTION
- GL_SPOT_EXPONENT
- GL_SPOT_CUTOFF
- GL_CONSTANT_ATTENUATION
- GL_LINEAR_ATTENUATION
- GL_QUADRATIC_ATTENUATION
- GL_COMPILE
- GL_COMPILE_AND_EXECUTE
- GL_BYTE
- GL_UNSIGNED_BYTE
- GL_SHORT
- GL_UNSIGNED_SHORT
- GL_INT
- GL_UNSIGNED_INT
- GL_FLOAT
- GL_2_BYTES
- GL_3_BYTES
- GL_4_BYTES
- GL_DOUBLE
- GL_CLEAR
- GL_AND
- GL_AND_REVERSE
- GL_COPY
- GL_AND_INVERTED
- GL_NOOP
- GL_XOR
- GL_OR
- GL_NOR
- GL_EQUIV
- GL_INVERT
- GL_OR_REVERSE
- GL_COPY_INVERTED
- GL_OR_INVERTED
- GL_NAND
- GL_SET
- GL_EMISSION

- GL_SHININESS
- GL_AMBIENT_AND_DIFFUSE
- GL_COLOR_INDEXES
- GL_MODELVIEW
- GL_PROJECTION
- GL_TEXTURE
- GL_COLOR
- GL_DEPTH
- GL_STENCIL
- GL_COLOR_INDEX
- GL_STENCIL_INDEX
- GL_DEPTH_COMPONENT
- GL_RED
- GL_GREEN
- GL_BLUE
- GL_ALPHA
- GL_RGB
- GL_RGBA
- GL_LUMINANCE
- GL_LUMINANCE_ALPHA
- GL_BITMAP
- GL_POINT
- GL_LINE
- GL_FILL
- GL_RENDER
- GL_FEEDBACK
- GL_SELECT
- GL_FLAT
- GL_SMOOTH
- GL_KEEP
- GL_REPLACE
- GL_INCR
- GL_DECR
- GL_VENDOR
- GL_RENDERER
- GL_VERSION

- GL_EXTENSIONS
- GL_S
- GL_ENABLE_BIT
- GL_T
- GL_R
- GL_Q
- GL_MODULATE
- GL_DECAL
- GL_TEXTURE_ENV_MODE
- GL_TEXTURE_ENV_COLOR
- GL_TEXTURE_ENV
- GL_EYE_LINEAR
- GL_OBJECT_LINEAR
- GL_SPHERE_MAP
- GL_TEXTURE_GEN_MODE
- GL_OBJECT_PLANE
- GL_EYE_PLANE
- GL_NEAREST
- GL_LINEAR
- GL_NEAREST_MIPMAP_NEAREST
- GL_LINEAR_MIPMAP_NEAREST
- GL_NEAREST_MIPMAP_LINEAR
- GL_LINEAR_MIPMAP_LINEAR
- GL_TEXTURE_MAG_FILTER
- GL_TEXTURE_MIN_FILTER
- GL_TEXTURE_WRAP_S
- GL_TEXTURE_WRAP_T
- GL_CLAMP
- GL_REPEAT
- GL_POLYGON_OFFSET_UNITS
- GL_POLYGON_OFFSET_POINT
- GL_POLYGON_OFFSET_LINE
- GL_R3_G3_B2
- GL_V2F
- GL_V3F
- GL_C4UB_V2F

- GL_C4UB_V3F
- GL_C3F_V3F
- GL_N3F_V3F
- GL_C4F_N3F_V3F
- GL_T2F_V3F
- GL_T4F_V4F
- GL_T2F_C4UB_V3F
- GL_T2F_C3F_V3F
- GL_T2F_N3F_V3F
- GL_T2F_C4F_N3F_V3F
- GL_T4F_C4F_N3F_V4F
- GL_CLIP_PLANE0
- GL_CLIP_PLANE1
- GL_CLIP_PLANE2
- GL_CLIP_PLANE3
- GL_CLIP_PLANE4
- GL_CLIP_PLANE5
- GL_LIGHT0
- GL_COLOR_BUFFER_BIT
- GL_LIGHT1
- GL_LIGHT2
- GL_LIGHT3
- GL_LIGHT4
- GL_LIGHT5
- GL_LIGHT6
- GL_LIGHT7
- GL_HINT_BIT
- GL_POLYGON_OFFSET_FILL
- GL_POLYGON_OFFSET_FACTOR
- GL_ALPHA4
- GL_ALPHA8
- GL_ALPHA12
- GL_ALPHA16
- GL_LUMINANCE4
- GL_LUMINANCE8
- GL_LUMINANCE12

- GL_LUMINANCE16
- GL_LUMINANCE4_ALPHA4
- GL_LUMINANCE6_ALPHA2
- GL_LUMINANCE8_ALPHA8
- GL_LUMINANCE12_ALPHA4
- GL_LUMINANCE12_ALPHA12
- GL_LUMINANCE16_ALPHA16
- GL_INTENSITY
- GL_INTENSITY4
- GL_INTENSITY8
- GL_INTENSITY12
- GL_INTENSITY16
- GL_RGB4
- GL_RGB5
- GL_RGB8
- GL_RGB10
- GL_RGB12
- GL_RGB16
- GL_RGBA2
- GL_RGBA4
- GL_RGB5_A1
- GL_RGBA8
- GL_RGB10_A2
- GL_RGBA12
- GL_RGBA16
- GL_TEXTURE_RED_SIZE
- GL_TEXTURE_GREEN_SIZE
- GL_TEXTURE_BLUE_SIZE
- GL_TEXTURE_ALPHA_SIZE
- GL_TEXTURE_LUMINANCE_SIZE
- GL_TEXTURE_INTENSITY_SIZE
- GL_PROXY_TEXTURE_1D
- GL_PROXY_TEXTURE_2D
- GL_TEXTURE_PRIORITY
- GL_TEXTURE_RESIDENT
- GL_TEXTURE_BINDING_1D

- GL_TEXTURE_BINDING_2D
- GL_VERTEX_ARRAY
- GL_NORMAL_ARRAY
- GL_COLOR_ARRAY
- GL_INDEX_ARRAY
- GL_TEXTURE_COORD_ARRAY
- GL_EDGE_FLAG_ARRAY
- GL_VERTEX_ARRAY_SIZE
- GL_VERTEX_ARRAY_TYPE
- GL_VERTEX_ARRAY_STRIDE
- GL_NORMAL_ARRAY_TYPE
- GL_NORMAL_ARRAY_STRIDE
- GL_COLOR_ARRAY_SIZE
- GL_COLOR_ARRAY_TYPE
- GL_COLOR_ARRAY_STRIDE
- GL_INDEX_ARRAY_TYPE
- GL_INDEX_ARRAY_STRIDE
- GL_TEXTURE_COORD_ARRAY_SIZE
- GL_TEXTURE_COORD_ARRAY_TYPE
- GL_TEXTURE_COORD_ARRAY_STRIDE
- GL_EDGE_FLAG_ARRAY_STRIDE
- GL_VERTEX_ARRAY_POINTER
- GL_NORMAL_ARRAY_POINTER
- GL_COLOR_ARRAY_POINTER
- GL_INDEX_ARRAY_POINTER
- GL_TEXTURE_COORD_ARRAY_POINTER
- GL_EDGE_FLAG_ARRAY_POINTER
- GL_COLOR_INDEX1_EXT
- GL_COLOR_INDEX2_EXT
- GL_COLOR_INDEX4_EXT
- GL_COLOR_INDEX8_EXT
- GL_COLOR_INDEX12_EXT
- GL_COLOR_INDEX16_EXT
- GL_EVAL_BIT
- GL_LIST_BIT
- GL_TEXTURE_BIT

- GL_SCISSOR_BIT
- GL_ALL_ATTRIB_BITS
- GL_CLIENT_ALL_ATTRIB_BITS
- GL_SMOOTH_POINT_SIZE_RANGE
- GL_SMOOTH_POINT_SIZE_GRANULARITY
- GL_SMOOTH_LINE_WIDTH_RANGE
- GL_SMOOTH_LINE_WIDTH_GRANULARITY
- GL_UNSIGNED_BYTE_3_3_2
- GL_UNSIGNED_SHORT_4_4_4_4
- GL_UNSIGNED_SHORT_5_5_5_1
- GL_UNSIGNED_INT_8_8_8_8
- GL_UNSIGNED_INT_10_10_10_2
- GL_RESCALE_NORMAL
- GL_TEXTURE_BINDING_3D
- GL_PACK_SKIP_IMAGES
- GL_PACK_IMAGE_HEIGHT
- GL_UNPACK_SKIP_IMAGES
- GL_UNPACK_IMAGE_HEIGHT
- GL_TEXTURE_3D
- GL_PROXY_TEXTURE_3D
- GL_TEXTURE_DEPTH
- GL_TEXTURE_WRAP_R
- GL_MAX_3D_TEXTURE_SIZE
- GL_BGR
- GL_BGRA
- GL_MAX_ELEMENTS_VERTICES
- GL_MAX_ELEMENTS_INDICES
- GL_CLAMP_TO_EDGE
- GL_TEXTURE_MIN_LOD
- GL_TEXTURE_MAX_LOD
- GL_TEXTURE_BASE_LEVEL
- GL_TEXTURE_MAX_LEVEL
- GL_LIGHT_MODEL_COLOR_CONTROL
- GL_SINGLE_COLOR
- GL_SEPARATE_SPECULAR_COLOR
- GL_UNSIGNED_BYTE_2_3_3_REV

- GL_UNSIGNED_SHORT_5_6_5
- GL_UNSIGNED_SHORT_5_6_5_REV
- GL_UNSIGNED_SHORT_4_4_4_4_REV
- GL_UNSIGNED_SHORT_1_5_5_5_REV
- GL_UNSIGNED_INT_8_8_8_8_REV
- GL_ALIASED_POINT_SIZE_RANGE
- GL_ALIASED_LINE_WIDTH_RANGE
- GL_MULTISAMPLE
- GL_SAMPLE_ALPHA_TO_COVERAGE
- GL_SAMPLE_ALPHA_TO_ONE
- GL_SAMPLE_COVERAGE
- GL_SAMPLE_BUFFERS
- GL_SAMPLES
- GL_SAMPLE_COVERAGE_VALUE
- GL_SAMPLE_COVERAGE_INVERT
- GL_CLAMP_TO_BORDER
- GL_TEXTURE0
- GL_TEXTURE1
- GL_TEXTURE2
- GL_TEXTURE3
- GL_TEXTURE4
- GL_TEXTURE5
- GL_TEXTURE6
- GL_TEXTURE7
- GL_TEXTURE8
- GL_TEXTURE9
- GL_TEXTURE10
- GL_TEXTURE11
- GL_TEXTURE12
- GL_TEXTURE13
- GL_TEXTURE14
- GL_TEXTURE15
- GL_TEXTURE16
- GL_TEXTURE17
- GL_TEXTURE18
- GL_TEXTURE19

- GL_TEXTURE20
- GL_TEXTURE21
- GL_TEXTURE22
- GL_TEXTURE23
- GL_TEXTURE24
- GL_TEXTURE25
- GL_TEXTURE26
- GL_TEXTURE27
- GL_TEXTURE28
- GL_TEXTURE29
- GL_TEXTURE30
- GL_TEXTURE31
- GL_ACTIVE_TEXTURE
- GL_CLIENT_ACTIVE_TEXTURE
- GL_MAX_TEXTURE_UNITS
- GL_TRANSPOSE_MODELVIEW_MATRIX
- GL_TRANSPOSE_PROJECTION_MATRIX
- GL_TRANSPOSE_TEXTURE_MATRIX
- GL_TRANSPOSE_COLOR_MATRIX
- GL_SUBTRACT
- GL_COMPRESSED_ALPHA
- GL_COMPRESSED_LUMINANCE
- GL_COMPRESSED_LUMINANCE_ALPHA
- GL_COMPRESSED_INTENSITY
- GL_COMPRESSED_RGB
- GL_COMPRESSED_RGBA
- GL_TEXTURE_COMPRESSION_HINT
- GL_NORMAL_MAP
- GL_REFLECTION_MAP
- GL_TEXTURE_CUBE_MAP
- GL_TEXTURE_BINDING_CUBE_MAP
- GL_TEXTURE_CUBE_MAP_POSITIVE_X
- GL_TEXTURE_CUBE_MAP_NEGATIVE_X
- GL_TEXTURE_CUBE_MAP_POSITIVE_Y
- GL_TEXTURE_CUBE_MAP_NEGATIVE_Y
- GL_TEXTURE_CUBE_MAP_POSITIVE_Z

- GL_TEXTURE_CUBE_MAP_NEGATIVE_Z
- GL_PROXY_TEXTURE_CUBE_MAP
- GL_MAX_CUBE_MAP_TEXTURE_SIZE
- GL_COMBINE
- GL_COMBINE_RGB
- GL_COMBINE_ALPHA
- GL_RGB_SCALE
- GL_ADD_SIGNED
- GL_INTERPOLATE
- GL_CONSTANT
- GL_PRIMARY_COLOR
- GL_PREVIOUS
- GL_SOURCE0_RGB
- GL_SOURCE1_RGB
- GL_SOURCE2_RGB
- GL_SOURCE0_ALPHA
- GL_SOURCE1_ALPHA
- GL_SOURCE2_ALPHA
- GL_OPERAND0_RGB
- GL_OPERAND1_RGB
- GL_OPERAND2_RGB
- GL_OPERAND0_ALPHA
- GL_OPERAND1_ALPHA
- GL_OPERAND2_ALPHA
- GL_TEXTURE_COMPRESSED_IMAGE_SIZE
- GL_TEXTURE_COMPRESSED
- GL_NUM_COMPRESSED_TEXTURE_FORMATS
- GL_COMPRESSED_TEXTURE_FORMATS
- GL_DOT3_RGB
- GL_DOT3_RGBA
- GL_MULTISAMPLE_BIT
- GL_BLEND_DST_RGB
- GL_BLEND_SRC_RGB
- GL_BLEND_DST_ALPHA
- GL_BLEND_SRC_ALPHA
- GL_POINT_SIZE_MIN

- GL_POINT_SIZE_MAX
- GL_POINT_FADE_THRESHOLD_SIZE
- GL_POINT_DISTANCE_ATTENUATION
- GL_GENERATE_MIPMAP
- GL_GENERATE_MIPMAP_HINT
- GL_DEPTH_COMPONENT16
- GL_DEPTH_COMPONENT24
- GL_DEPTH_COMPONENT32
- GL_MIRRORED_REPEAT
- GL_FOG_COORDINATE_SOURCE
- GL_FOG_COORDINATE
- GL_FRAGMENT_DEPTH
- GL_CURRENT_FOG_COORDINATE
- GL_FOG_COORDINATE_ARRAY_TYPE
- GL_FOG_COORDINATE_ARRAY_STRIDE
- GL_FOG_COORDINATE_ARRAY_POINTER
- GL_FOG_COORDINATE_ARRAY
- GL_COLOR_SUM
- GL_CURRENT_SECONDARY_COLOR
- GL_SECONDARY_COLOR_ARRAY_SIZE
- GL_SECONDARY_COLOR_ARRAY_TYPE
- GL_SECONDARY_COLOR_ARRAY_STRIDE
- GL_SECONDARY_COLOR_ARRAY_POINTER
- GL_SECONDARY_COLOR_ARRAY
- GL_MAX_TEXTURE_LOD_BIAS
- GL_TEXTURE_FILTER_CONTROL
- GL_TEXTURE_LOD_BIAS
- GL_INCR_WRAP
- GL_DECR_WRAP
- GL_TEXTURE_DEPTH_SIZE
- GL_DEPTH_TEXTURE_MODE
- GL_TEXTURE_COMPARE_MODE
- GL_TEXTURE_COMPARE_FUNC
- GL_COMPARE_R_TO_TEXTURE
- GL_CURRENT_FOG_COORD
- GL_FOG_COORD

- GL_FOG_COORD_ARRAY
- GL_FOG_COORD_ARRAY_BUFFER_BINDING
- GL_FOG_COORD_ARRAY_POINTER
- GL_FOG_COORD_ARRAY_STRIDE
- GL_FOG_COORD_ARRAY_TYPE
- GL_FOG_COORD_SRC
- GL_SRC0_ALPHA
- GL_SRC0_RGB
- GL_SRC1_ALPHA
- GL_SRC1_RGB
- GL_SRC2_ALPHA
- GL_SRC2_RGB
- GL_BUFFER_SIZE
- GL_BUFFER_USAGE
- GL_QUERY_COUNTER_BITS
- GL_CURRENT_QUERY
- GL_QUERY_RESULT
- GL_QUERY_RESULT_AVAILABLE
- GL_ARRAY_BUFFER
- GL_ELEMENT_ARRAY_BUFFER
- GL_ARRAY_BUFFER_BINDING
- GL_ELEMENT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ARRAY_BUFFER_BINDING
- GL_NORMAL_ARRAY_BUFFER_BINDING
- GL_COLOR_ARRAY_BUFFER_BINDING
- GL_INDEX_ARRAY_BUFFER_BINDING
- GL_TEXTURE_COORD_ARRAY_BUFFER_BINDING
- GL_EDGE_FLAG_ARRAY_BUFFER_BINDING
- GL_SECONDARY_COLOR_ARRAY_BUFFER_BINDING
- GL_FOG_COORDINATE_ARRAY_BUFFER_BINDING
- GL_WEIGHT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ATTRIB_ARRAY_BUFFER_BINDING
- GL_READ_ONLY
- GL_WRITE_ONLY
- GL_READ_WRITE
- GL_BUFFER_ACCESS

- GL_BUFFER_MAPPED
- GL_BUFFER_MAP_POINTER
- GL_STREAM_DRAW
- GL_STREAM_READ
- GL_STREAM_COPY
- GL_STATIC_DRAW
- GL_STATIC_READ
- GL_STATIC_COPY
- GL_DYNAMIC_DRAW
- GL_DYNAMIC_READ
- GL_DYNAMIC_COPY
- GL_SAMPLES_PASSED
- GL_BLEND_EQUATION_RGB
- GL_VERTEX_ATTRIB_ARRAY_ENABLED
- GL_VERTEX_ATTRIB_ARRAY_SIZE
- GL_VERTEX_ATTRIB_ARRAY_STRIDE
- GL_VERTEX_ATTRIB_ARRAY_TYPE
- GL_CURRENT_VERTEX_ATTRIB
- GL_VERTEX_PROGRAM_POINT_SIZE
- GL_VERTEX_PROGRAM_TWO_SIDE
- GL_VERTEX_ATTRIB_ARRAY_POINTER
- GL_STENCIL_BACK_FUNC
- GL_STENCIL_BACK_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_PASS
- GL_MAX_DRAW_BUFFERS
- GL_DRAW_BUFFER0
- GL_DRAW_BUFFER1
- GL_DRAW_BUFFER2
- GL_DRAW_BUFFER3
- GL_DRAW_BUFFER4
- GL_DRAW_BUFFER5
- GL_DRAW_BUFFER6
- GL_DRAW_BUFFER7
- GL_DRAW_BUFFER8
- GL_DRAW_BUFFER9

- GL_DRAW_BUFFER10
- GL_DRAW_BUFFER11
- GL_DRAW_BUFFER12
- GL_DRAW_BUFFER13
- GL_DRAW_BUFFER14
- GL_DRAW_BUFFER15
- GL_BLEND_EQUATION_ALPHA
- GL_POINT_SPRITE
- GL_COORD_REPLACE
- GL_MAX_VERTEX_ATTRIBS
- GL_VERTEX_ATTRIB_ARRAY_NORMALIZED
- GL_MAX_TEXTURE_COORDS
- GL_MAX_TEXTURE_IMAGE_UNITS
- GL_FRAGMENT_SHADER
- GL_VERTEX_SHADER
- GL_MAX_FRAGMENT_UNIFORM_COMPONENTS
- GL_MAX_VERTEX_UNIFORM_COMPONENTS
- GL_MAX_VARYING_FLOATS
- GL_MAX_VERTEX_TEXTURE_IMAGE_UNITS
- GL_MAX_COMBINED_TEXTURE_IMAGE_UNITS
- GL_SHADER_TYPE
- GL_FLOAT_VEC2
- GL_FLOAT_VEC3
- GL_FLOAT_VEC4
- GL_INT_VEC2
- GL_INT_VEC3
- GL_INT_VEC4
- GL_BOOL
- GL_BOOL_VEC2
- GL_BOOL_VEC3
- GL_BOOL_VEC4
- GL_FLOAT_MAT2
- GL_FLOAT_MAT3
- GL_FLOAT_MAT4
- GL_SAMPLER_1D
- GL_SAMPLER_2D

- GL_SAMPLER_3D
- GL_SAMPLER_CUBE
- GL_SAMPLER_1D_SHADOW
- GL_SAMPLER_2D_SHADOW
- GL_DELETE_STATUS
- GL_COMPILE_STATUS
- GL_LINK_STATUS
- GL_VALIDATE_STATUS
- GL_INFO_LOG_LENGTH
- GL_ATTACHED_SHADERS
- GL_ACTIVE_UNIFORMS
- GL_ACTIVE_UNIFORM_MAX_LENGTH
- GL_SHADER_SOURCE_LENGTH
- GL_ACTIVE_ATTRIBUTES
- GL_ACTIVE_ATTRIBUTE_MAX_LENGTH
- GL_FRAGMENT_SHADER_DERIVATIVE_HINT
- GL_SHADING_LANGUAGE_VERSION
- GL_CURRENT_PROGRAM
- GL_POINT_SPRITE_COORD_ORIGIN
- GL_LOWER_LEFT
- GL_UPPER_LEFT
- GL_STENCIL_BACK_REF
- GL_STENCIL_BACK_VALUE_MASK
- GL_STENCIL_BACK_WRITEMASK
- GL_CURRENT_RASTER_SECONDARY_COLOR
- GL_PIXEL_PACK_BUFFER
- GL_PIXEL_UNPACK_BUFFER
- GL_PIXEL_PACK_BUFFER_BINDING
- GL_PIXEL_UNPACK_BUFFER_BINDING
- GL_FLOAT_MAT2x3
- GL_FLOAT_MAT2x4
- GL_FLOAT_MAT3x2
- GL_FLOAT_MAT3x4
- GL_FLOAT_MAT4x2
- GL_FLOAT_MAT4x3
- GL_SRGB

- GL_SRGB8
- GL_SRGB_ALPHA
- GL_SRGB8_ALPHA8
- GL_SLUMINANCE_ALPHA
- GL_SLUMINANCE8_ALPHA8
- GL_SLUMINANCE
- GL_SLUMINANCE8
- GL_COMPRESSED_SRGB
- GL_COMPRESSED_SRGB_ALPHA
- GL_COMPRESSED_SLUMINANCE
- GL_COMPRESSED_SLUMINANCE_ALPHA
- GL_CLIP_DISTANCE0
- GL_CLIP_DISTANCE1
- GL_CLIP_DISTANCE2
- GL_CLIP_DISTANCE3
- GL_CLIP_DISTANCE4
- GL_CLIP_DISTANCE5
- GL_COMPARE_REF_TO_TEXTURE
- GL_MAX_CLIP_DISTANCES
- GL_MAX_VARYING_COMPONENTS
- GL_CONTEXT_FLAG_FORWARD_COMPATIBLE_BIT
- GL_MAJOR_VERSION
- GL_MINOR_VERSION
- GL_NUM_EXTENSIONS
- GL_CONTEXT_FLAGS
- GL_DEPTH_BUFFER
- GL_STENCIL_BUFFER
- GL_RGBA32F
- GL_RGB32F
- GL_RGBA16F
- GL_RGB16F
- GL_VERTEX_ATTRIB_ARRAY_INTEGER
- GL_MAX_ARRAY_TEXTURE_LAYERS
- GL_MIN_PROGRAM_TEXEL_OFFSET
- GL_MAX_PROGRAM_TEXEL_OFFSET
- GL_CLAMP_VERTEX_COLOR

- GL_CLAMP_FRAGMENT_COLOR
- GL_CLAMP_READ_COLOR
- GL_FIXED_ONLY
- GL_TEXTURE_RED_TYPE
- GL_TEXTURE_GREEN_TYPE
- GL_TEXTURE_BLUE_TYPE
- GL_TEXTURE_ALPHA_TYPE
- GL_TEXTURE_LUMINANCE_TYPE
- GL_TEXTURE_INTENSITY_TYPE
- GL_TEXTURE_DEPTH_TYPE
- GL_TEXTURE_1D_ARRAY
- GL_PROXY_TEXTURE_1D_ARRAY
- GL_TEXTURE_2D_ARRAY
- GL_PROXY_TEXTURE_2D_ARRAY
- GL_TEXTURE_BINDING_1D_ARRAY
- GL_TEXTURE_BINDING_2D_ARRAY
- GL_R11F_G11F_B10F
- GL_UNSIGNED_INT_10F_11F_11F_REV
- GL_RGB9_E5
- GL_UNSIGNED_INT_5_9_9_9_REV
- GL_TEXTURE_SHARED_SIZE
- GL_TRANSFORM_FEEDBACK_VARYING_MAX_LENGTH
- GL_TRANSFORM_FEEDBACK_BUFFER_MODE
- GL_MAX_TRANSFORM_FEEDBACK_SEPARATE_COMPONENTS
- GL_TRANSFORM_FEEDBACK_VARYINGS
- GL_TRANSFORM_FEEDBACK_BUFFER_START
- GL_TRANSFORM_FEEDBACK_BUFFER_SIZE
- GL_PRIMITIVES_GENERATED
- GL_TRANSFORM_FEEDBACK_PRIMITIVES_WRITTEN
- GL_RASTERIZER_DISCARD
- GL_MAX_TRANSFORM_FEEDBACK_INTERLEAVED_COMPONENTS
- GL_MAX_TRANSFORM_FEEDBACK_SEPARATE_ATTRIBS
- GL_INTERLEAVED_ATTRIBS
- GL_SEPARATE_ATTRIBS
- GL_TRANSFORM_FEEDBACK_BUFFER
- GL_TRANSFORM_FEEDBACK_BUFFER_BINDING

- GL_RGBA32UI
- GL_RGB32UI
- GL_RGBA16UI
- GL_RGB16UI
- GL_RGBA8UI
- GL_RGB8UI
- GL_RGBA32I
- GL_RGB32I
- GL_RGBA16I
- GL_RGB16I
- GL_RGBA8I
- GL_RGB8I
- GL_RED_INTEGER
- GL_GREEN_INTEGER
- GL_BLUE_INTEGER
- GL_ALPHA_INTEGER
- GL_RGB_INTEGER
- GL_RGBA_INTEGER
- GL_BGR_INTEGER
- GL_BGRA_INTEGER
- GL_SAMPLER_1D_ARRAY
- GL_SAMPLER_2D_ARRAY
- GL_SAMPLER_1D_ARRAY_SHADOW
- GL_SAMPLER_2D_ARRAY_SHADOW
- GL_SAMPLER_CUBE_SHADOW
- GL_UNSIGNED_INT_VEC2
- GL_UNSIGNED_INT_VEC3
- GL_UNSIGNED_INT_VEC4
- GL_INT_SAMPLER_1D
- GL_INT_SAMPLER_2D
- GL_INT_SAMPLER_3D
- GL_INT_SAMPLER_CUBE
- GL_INT_SAMPLER_1D_ARRAY
- GL_INT_SAMPLER_2D_ARRAY
- GL_UNSIGNED_INT_SAMPLER_1D
- GL_UNSIGNED_INT_SAMPLER_2D

- GL_UNSIGNED_INT_SAMPLER_3D
- GL_UNSIGNED_INT_SAMPLER_CUBE
- GL_UNSIGNED_INT_SAMPLER_1D_ARRAY
- GL_UNSIGNED_INT_SAMPLER_2D_ARRAY
- GL_QUERY_WAIT
- GL_QUERY_NO_WAIT
- GL_QUERY_BY_REGION_WAIT
- GL_QUERY_BY_REGION_NO_WAIT
- GL_TEXTURE_RECTANGLE
- GL_TEXTURE_BINDING_RECTANGLE
- GL_PROXY_TEXTURE_RECTANGLE
- GL_MAX_RECTANGLE_TEXTURE_SIZE
- GL_SAMPLER_2D_RECT
- GL_SAMPLER_2D_RECT_SHADOW
- GL_TEXTURE_BUFFER
- GL_MAX_TEXTURE_BUFFER_SIZE
- GL_TEXTURE_BINDING_BUFFER
- GL_TEXTURE_BUFFER_DATA_STORE_BINDING
- GL_TEXTURE_BUFFER_FORMAT
- GL_SAMPLER_BUFFER
- GL_INT_SAMPLER_2D_RECT
- GL_INT_SAMPLER_BUFFER
- GL_UNSIGNED_INT_SAMPLER_2D_RECT
- GL_UNSIGNED_INT_SAMPLER_BUFFER
- GL_RED_SNORM
- GL_RG_SNORM
- GL_RGB_SNORM
- GL_RGBA_SNORM
- GL_R8_SNORM
- GL_RG8_SNORM
- GL_RGB8_SNORM
- GL_RGBA8_SNORM
- GL_R16_SNORM
- GL_RG16_SNORM
- GL_RGB16_SNORM
- GL_RGBA16_SNORM

- GL_SIGNED_NORMALIZED
- GL_PRIMITIVE_RESTART
- GL_PRIMITIVE_RESTART_INDEX
- GL_BUFFER_ACCESS_FLAGS
- GL_BUFFER_MAP_LENGTH
- GL_BUFFER_MAP_OFFSET
- GL_CONTEXT_CORE_PROFILE_BIT
- GL_CONTEXT_COMPATIBILITY_PROFILE_BIT
- GL_LINES_ADJACENCY
- GL_LINE_STRIP_ADJACENCY
- GL_TRIANGLES_ADJACENCY
- GL_TRIANGLE_STRIP_ADJACENCY
- GL_PROGRAM_POINT_SIZE
- GL_GEOMETRY_VERTICES_OUT
- GL_GEOMETRY_INPUT_TYPE
- GL_GEOMETRY_OUTPUT_TYPE
- GL_MAX_GEOMETRY_TEXTURE_IMAGE_UNITS
- GL_FRAMEBUFFER_ATTACHMENT_LAYERED
- GL_FRAMEBUFFER_INCOMPLETE_LAYER_TARGETS
- GL_GEOMETRY_SHADER
- GL_MAX_GEOMETRY_UNIFORM_COMPONENTS
- GL_MAX_GEOMETRY_OUTPUT_VERTICES
- GL_MAX_GEOMETRY_TOTAL_OUTPUT_COMPONENTS
- GL_MAX_VERTEX_OUTPUT_COMPONENTS
- GL_MAX_GEOMETRY_INPUT_COMPONENTS
- GL_MAX_GEOMETRY_OUTPUT_COMPONENTS
- GL_MAX_FRAGMENT_INPUT_COMPONENTS
- GL_CONTEXT_PROFILE_MASK
- GL_VERTEX_ATTRIB_ARRAY_DIVISOR
- GL_RGB10_A2UI
- GL_SAMPLE_SHADING
- GL_MIN_SAMPLE_SHADING_VALUE
- GL_MIN_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_COMPONENTS
- GL_TEXTURE_CUBE_MAP_ARRAY

- GL_TEXTURE_BINDING_CUBE_MAP_ARRAY
- GL_PROXY_TEXTURE_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY_SHADOW
- GL_INT_SAMPLER_CUBE_MAP_ARRAY
- GL_UNSIGNED_INT_SAMPLER_CUBE_MAP_ARRAY
- GL_TRANSFORM_FEEDBACK_PAUSED
- GL_TRANSFORM_FEEDBACK_ACTIVE
- GL_COMPRESSED_RGBA_BPTC_UNORM
- GL_COMPRESSED_SRGB_ALPHA_BPTC_UNORM
- GL_COMPRESSED_RGB_BPTC_SIGNED_FLOAT
- GL_COMPRESSED_RGB_BPTC_UNSIGNED_FLOAT
- GL_COPY_READ_BUFFER_BINDING
- GL_COPY_WRITE_BUFFER_BINDING
- GL_NUM_SHADING_LANGUAGE_VERSIONS
- GL_VERTEX_ATTRIB_ARRAY_LONG
- GL_PRIMITIVE_RESTART_FOR_PATCHES_SUPPORTED
- GL_MAX_VERTEX_ATTRIB_STRIDE
- GL_TEXTURE_BUFFER_BINDING
- void glAccum(GLenum op, GLfloat value)
- void glActiveTexture(GLenum texture)
- void glAlphaFunc(GLenum func, GLclampf ref)
- GLboolean glAreTexturesResident(GLsizei n, const GLuint * textures, GLboolean * residences)
- void glArrayElement(GLint i)
- void glAttachShader(GLuint program, GLuint shader)
- void glBegin(GLenum mode)
- void glBeginQuery(GLenum target, GLuint id)
- void glBindAttribLocation(GLuint program, GLuint index, const GLchar *name)
- void glBindBuffer(GLenum target, GLuint buffer)
- void glBindTexture(GLenum target, GLuint texture)
- void glBitmap(GLsizei width, GLsizei height, GLfloat xorig, GLfloat yorig, GLfloat xmove, GLfloat ymove, const GLubyte * bitmap)
- void glBlendColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)
- void glBlendEquation(GLenum mode)
- void glBlendEquationSeparate(GLenum modeRGB, GLenum modeAlpha)
- void glBlendFunc(GLenum sfactor, GLenum dfactor)

- `void glBlendFuncSeparate(GLenum srcRGB, GLenum dstRGB, GLenum srcAlpha, GLenum dstAlpha)`
- `void glBufferData(GLenum target, GLsizeiptr size, const GLvoid * data, GLenum usage)`
- `void glBufferSubData(GLenum target, GLintptr offset, GLsizeiptr size, const GLvoid * data)`
- `void glCallList(GLuint list)`
- `void glCallLists(GLsizei n, GLenum type, const GLvoid * lists)`
- `void glClear(GLbitfield mask)`
- `void glClearAccum(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)`
- `void glClearColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)`
- `void glClearDepth(GLclampd depth)`
- `void glClearIndex(GLfloat c)`
- `void glClearStencil(GLint s)`
- `void glClientActiveTexture(GLenum texture)`
- `void glClipPlane(GLenum plane, const GLdouble * equation)`
- `void glColor3b(GLbyte red, GLbyte green, GLbyte blue)`
- `void glColor3s(GLshort red, GLshort green, GLshort blue)`
- `void glColor3i(GLint red, GLint green, GLint blue)`
- `void glColor3f(GLfloat red, GLfloat green, GLfloat blue)`
- `void glColor3d(GLdouble red, GLdouble green, GLdouble blue)`
- `void glColor3ub(GLubyte red, GLubyte green, GLubyte blue)`
- `void glColor3us(GLushort red, GLushort green, GLushort blue)`
- `void glColor3ui(GLuint red, GLuint green, GLuint blue)`
- `void glColor4b(GLbyte red, GLbyte green, GLbyte blue, GLbyte alpha)`
- `void glColor4s(GLshort red, GLshort green, GLshort blue, GLshort alpha)`
- `void glColor4i(GLint red, GLint green, GLint blue, GLint alpha)`
- `void glColor4f(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)`
- `void glColor4d(GLdouble red, GLdouble green, GLdouble blue, GLdouble alpha)`
- `void glColor4ub(GLubyte red, GLubyte green, GLubyte blue, GLubyte alpha)`
- `void glColor4us(GLushort red, GLushort green, GLushort blue, GLushort alpha)`
- `void glColor4ui(GLuint red, GLuint green, GLuint blue, GLuint alpha)`
- `void glColor3bv(const GLbyte * v)`
- `void glColor3sv(const GLshort * v)`
- `void glColor3iv(const GLint * v)`
- `void glColor3fv(const GLfloat * v)`
- `void glColor3dv(const GLdouble * v)`
- `void glColor3ubv(const GLubyte * v)`
- `void glColor3usv(const GLushort * v)`

- `void glColor3uiv(const GLuint * v)`
- `void glColor4bv(const GLbyte * v)`
- `void glColor4sv(const GLshort * v)`
- `void glColor4iv(const GLint * v)`
- `void glColor4fv(const GLfloat * v)`
- `void glColor4dv(const GLdouble * v)`
- `void glColor4ubv(const GLubyte * v)`
- `void glColor4usv(const GLushort * v)`
- `void glColor4uiv(const GLuint * v)`
- `void glColorMask(GLboolean red, GLboolean green, GLboolean blue, GLboolean alpha)`
- `void glColorMaterial(GLenum face, GLenum mode)`
- `void glColorPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)`
- `void glColorSubTable(GLenum target, GLsizei start, GLsizei count, GLenum format, GLenum type, const GLvoid * data)`
- `void glColorTable(GLenum target, GLenum internalformat, GLsizei width, GLenum format, GLenum type, const GLvoid * data)`
- `void glColorTableParameterfv(GLenum target, GLenum pname, const GLfloat * params)`
- `void glColorTableParameteriv(GLenum target, GLenum pname, const GLint * params)`
- `void glCompileShader(GLuint shader)`
- `void glCompressedTexImage1D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLint border, GLsizei imageSize, const GLvoid * data)`
- `void glCompressedTexImage2D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLint border, GLsizei imageSize, const GLvoid * data)`
- `void glCompressedTexImage3D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLsizei depth, GLint border, GLsizei imageSize, const GLvoid * data)`
- `void glCompressedTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLsizei imageSize, const GLvoid * data)`
- `void glCompressedTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLsizei imageSize, const GLvoid * data)`
- `void glCompressedTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLsizei imageSize, const GLvoid * data)`
- `void glConvolutionFilter1D(GLenum target, GLenum internalformat, GLsizei width, GLenum format, GLenum type, const GLvoid * data)`
- `void glConvolutionFilter2D(GLenum target, GLenum internalformat, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)`
- `void glConvolutionParameterf(GLenum target, GLenum pname, GLfloat params)`
- `void glConvolutionParameteri(GLenum target, GLenum pname, GLint params)`
- `void glConvolutionParameterfv(GLenum target, GLenum pname, const GLfloat * params)`
- `void glConvolutionParameteriv(GLenum target, GLenum pname, const GLint * params)`

- void glCopyColorSubTable(GLenum target,GLsizei start,GLint x,GLint y,GLsizei width)
- void glCopyColorTable(GLenum target,GLenum internalformat,GLint x,GLint y,GLsizei width)
- void glCopyConvolutionFilter1D(GLenum target,GLenum internalformat,GLint x,GLint y,GLsizei width)
- void glCopyConvolutionFilter2D(GLenum target,GLenum internalformat,GLint x,GLint y,GLsizei width,GLsizei height)
- void glCopyPixels(GLint x,GLint y,GLsizei width,GLsizei height,GLenum type)
- void glCopyTexImage1D(GLenum target,GLint level,GLenum internalformat,GLint x,GLint y,GLsizei width,GLint border)
- void glCopyTexImage2D(GLenum target,GLint level,GLenum internalformat,GLint x,GLint y,GLsizei width,GLsizei height,GLint border)
- void glCopyTexSubImage1D(GLenum target,GLint level,GLint xoffset,GLint x,GLint y,GLsizei width)
- void glCopyTexSubImage2D(GLenum target,GLint level,GLint xoffset,GLint yoffset,GLint x,GLint y,GLsizei width,GLsizei height)
- void glCopyTexSubImage3D(GLenum target,GLint level,GLint xoffset,GLint yoffset,GLint zoffset,GLint x,GLint y,GLsizei width,GLsizei height)
- GLuint glCreateProgram(void)
- GLuint glCreateShader(GLenum shaderType)
- void glCullFace(GLenum mode)
- void glDeleteBuffers(GLsizei n,const GLuint * buffers)
- void glDeleteLists(GLuint list,GLsizei range)
- void glDeleteProgram(GLuint program)
- void glDeleteQueries(GLsizei n,const GLuint * ids)
- void glDeleteShader(GLuint shader)
- void glDeleteTextures(GLsizei n,const GLuint * textures)
- void glDepthFunc(GLenum func)
- void glDepthMask(GLboolean flag)
- void glDepthRange(GLclampd nearVal,GLclampd farVal)
- void glDetachShader(GLuint program,GLuint shader)
- void glEnable(GLenum cap)
- void glEnableClientState(GLenum cap)
- void glEnableVertexAttribArray(GLuint index)
- void glDisableVertexAttribArray(GLuint index)
- void glDrawArrays(GLenum mode,GLint first,GLsizei count)
- void glDrawBuffer(GLenum mode)
- void glDrawBuffers(GLsizei n,const GLenum *bufs)
- void glDrawElements(GLenum mode,GLsizei count,GLenum type,const GLvoid * indices)
- void glDrawPixels(GLsizei width,GLsizei height,GLenum format,GLenum type,const GLvoid * data)

- void glDrawRangeElements(GLenum mode,GLuint start,GLuint end,GLsizei count,GLenum type,const GLvoid * indices)
- void glEdgeFlag(GLboolean flag)
- void glEdgeFlagPointer(GLsizei stride,const GLvoid * pointer)
- void glEnd(void)
- void glEndList(void)
- void glEndQuery(GLenum target)
- void glEvalCoord1f(GLfloat u)
- void glEvalCoord1d(GLdouble u)
- void glEvalCoord2f(GLfloat u,GLfloat v)
- void glEvalCoord2d(GLdouble u,GLdouble v)
- void glEvalMesh1(GLenum mode,GLint i1,GLint i2)
- void glEvalPoint1(GLint i)
- void glEvalPoint2(GLint i,GLint j)
- void glFeedbackBuffer(GLsizei size,GLenum type,GLfloat * buffer)
- void glFinish(void)
- void glFlush(void)
- void glFogf(GLenum pname,GLfloat param)
- void glFogi(GLenum pname,GLint param)
- void glFogfv(GLenum pname,const GLfloat * params)
- void glFogiv(GLenum pname,const GLint * params)
- void glFogCoordd(GLdouble coord)
- void glFogCoordf(GLfloat coord)
- void glFogCoorddv(GLdouble * coord)
- void glFogCoordfv(GLfloat * coord)
- void glFogCoordPointer(GLenum type,GLsizei stride,GLvoid * pointer)
- void glFrontFace(GLenum mode)
- void glFrustum(GLdouble left,GLdouble right,GLdouble bottom,GLdouble top,GLdouble nearVal,GLdouble farVal)
- void glGenBuffers(GLsizei n,GLuint * buffers)
- GLuint glGenLists(GLsizei range)
- void glGenQueries(GLsizei n,GLuint * ids)
- void glGenTextures(GLsizei n,GLuint * textures)
- void glGetBooleany(GLenum pname,GLboolean * params)
- void glGetDoublev(GLenum pname,GLdouble * params)
- void glGetFloatv(GLenum pname,GLfloat * params)
- void glGetIntegerv(GLenum pname,GLint * params)

- void glGetActiveAttrib(GLuint program,GLuint index,GLsizei bufSize,GLsizei *length,GLint *size,GLenum *type,GLchar *name)
- void glGetActiveUniform(GLuint program,GLuint index,GLsizei bufSize,GLsizei *length,GLint *size,GLenum *type,GLchar *name)
- void glGetAttachedShaders(GLuint program,GLsizei maxCount,GLsizei *count,GLuint *shaders)
- GLint glGetAttribLocation(GLuint program,const GLchar *name)
- void glGetBufferParameteriv(GLenum target,GLenum value,GLint * data)
- void glGetBufferPointerv(GLenum target,GLenum pname,GLvoid ** params)
- void glGetBufferSubData(GLenum target,GLintptr offset,GLsizei size,GLvoid * data)
- void glGetClipPlane(GLenum plane,GLdouble * equation)
- void glGetColorTable(GLenum target,GLenum format,GLenum type,GLvoid * table)
- void glGetColorTableParameterfv(GLenum target,GLenum pname,GLfloat * params)
- void glGetColorTableParameteriv(GLenum target,GLenum pname,GLint * params)
- void glGetCompressedTexImage(GLenum target,GLint lod,GLvoid * img)
- void glGetConvolutionFilter(GLenum target,GLenum format,GLenum type,GLvoid * image)
- void glGetConvolutionParameterfv(GLenum target,GLenum pname,GLfloat * params)
- void glGetConvolutionParameteriv(GLenum target,GLenum pname,GLint * params)
- GLenum glGetError(void)
- void glGetHistogram(GLenum target,GLboolean reset,GLenum format,GLenum type,GLvoid * values)
- void glGetHistogramParameterfv(GLenum target,GLenum pname,GLfloat * params)
- void glGetHistogramParameteriv(GLenum target,GLenum pname,GLint * params)
- void glGetLightfv(GLenum light,GLenum pname,GLfloat * params)
- void glGetLightiv(GLenum light,GLenum pname,GLint * params)
- void glGetMapdv(GLenum target,GLenum query,GLdouble * v)
- void glGetMapfv(GLenum target,GLenum query,GLfloat * v)
- void glGetMapiv(GLenum target,GLenum query,GLint * v)
- void glGetMaterialfv(GLenum face,GLenum pname,GLfloat * params)
- void glGetMaterialiv(GLenum face,GLenum pname,GLint * params)
- void glGetMinmax(GLenum target,GLboolean reset,GLenum format,GLenum types,GLvoid * values)
- void glGetMinmaxParameterfv(GLenum target,GLenum pname,GLfloat * params)
- void glGetMinmaxParameteriv(GLenum target,GLenum pname,GLint * params)
- void glGetPixelMapfv(GLenum map,GLfloat * data)
- void glGetPixelMapuiv(GLenum map,GLuint * data)
- void glGetPixelMapusv(GLenum map,GLushort * data)
- void glGetPointerv(GLenum pname,GLvoid ** params)
- void glGetPolygonStipple(GLubyte * pattern)
- void glGetProgramiv(GLuint program,GLenum pname,GLint *params)

- `void glGetProgramInfoLog(GLuint program, GLsizei maxLength, GLsizei *length, GLchar *infoLog)`
- `void glGetQueryObjectiv(GLuint id, GLenum pname, GLint * params)`
- `void glGetQueryObjectuiv(GLuint id, GLenum pname, GLuint * params)`
- `void glGetQueryiv(GLenum target, GLenum pname, GLint * params)`
- `void glGetSeparableFilter(GLenum target, GLenum format, GLenum type, GLvoid * row, GLvoid * column, GLvoid * span)`
- `void glGetShaderiv(GLuint shader, GLenum pname, GLint *params)`
- `void glGetShaderInfoLog(GLuint shader, GLsizei maxLength, GLsizei *length, GLchar *infoLog)`
- `void glGetShaderSource(GLuint shader, GLsizei bufSize, GLsizei *length, GLchar *source)`
- `const GLubyte* glGetString(GLenum name)`
- `void glGetTexEnvfv(GLenum target, GLenum pname, GLfloat * params)`
- `void glGetTexEnviv(GLenum target, GLenum pname, GLint * params)`
- `void glGetTexGendv(GLenum coord, GLenum pname, GLdouble * params)`
- `void glGetTexGenfv(GLenum coord, GLenum pname, GLfloat * params)`
- `void glGetTexGeniv(GLenum coord, GLenum pname, GLint * params)`
- `void glGetTexImage(GLenum target, GLint level, GLenum format, GLenum type, GLvoid * img)`
- `void glGetTexLevelParameterfv(GLenum target, GLint level, GLenum pname, GLfloat * params)`
- `void glGetTexLevelParameteriv(GLenum target, GLint level, GLenum pname, GLint * params)`
- `void glGetTexParameterfv(GLenum target, GLenum pname, GLfloat * params)`
- `void glGetTexParameteriv(GLenum target, GLenum pname, GLint * params)`
- `void glGetUniformfv(GLuint program, GLint location, GLfloat *params)`
- `void glGetUniformiv(GLuint program, GLint location, GLint *params)`
- `GLint glGetUniformLocation(GLuint program, const GLchar *name)`
- `void glGetVertexAttribdv(GLuint index, GLenum pname, GLdouble *params)`
- `void glGetVertexAttribfv(GLuint index, GLenum pname, GLfloat *params)`
- `void glGetVertexAttribiv(GLuint index, GLenum pname, GLint *params)`
- `void glGetVertexAttribPointerv(GLuint index, GLenum pname, GLvoid **pointer)`
- `void glHint(GLenum target, GLenum mode)`
- `void glHistogram(GLenum target, GLsizei width, GLenum internalformat, GLboolean sink)`
- `void glIndexs(GLshort c)`
- `void glIndexi(GLint c)`
- `void glIndexf(GLfloat c)`
- `void glIndexd(GLdouble c)`
- `void glIndexub(GLubyte c)`
- `void glIndexsv(const GLshort * c)`
- `void glIndexiv(const GLint * c)`

- `void glIndexfv(const GLfloat * c)`
- `void glIndexdv(const GLdouble * c)`
- `void glIndexbv(const GLubyte * c)`
- `void glIndexMask(GLuint mask)`
- `void glIndexPointer(GLenum type,GLsizei stride,const GLvoid * pointer)`
- `void glInitNames(void)`
- `void glInterleavedArrays(GLenum format,GLsizei stride,const GLvoid * pointer)`
- `GLboolean glIsBuffer(GLuint buffer)`
- `GLboolean glIsEnabled(GLenum cap)`
- `GLboolean glIsList(GLuint list)`
- `GLboolean glIsProgram(GLuint program)`
- `GLboolean glIsQuery(GLuint id)`
- `GLboolean glIsShader(GLuint shader)`
- `GLboolean glIsTexture(GLuint texture)`
- `void glLightf(GLenum light,GLenum pname,GLfloat param)`
- `void glLighti(GLenum light,GLenum pname,GLint param)`
- `void glLightfv(GLenum light,GLenum pname,const GLfloat * params)`
- `void glLightiv(GLenum light,GLenum pname,const GLint * params)`
- `void glLightModelf(GLenum pname,GLfloat param)`
- `void glLightModeli(GLenum pname,GLint param)`
- `void glLightModelfv(GLenum pname,const GLfloat * params)`
- `void glLightModeliv(GLenum pname,const GLint * params)`
- `void glLineStipple(GLint factor,GLushort pattern)`
- `void glLineWidth(GLfloat width)`
- `void glLinkProgram(GLuint program)`
- `void glListBase(GLuint base)`
- `void glLoadIdentity(void)`
- `void glLoadMatrixd(const GLdouble * m)`
- `void glLoadMatrixf(const GLfloat * m)`
- `void glLoadName(GLuint name)`
- `void glLoadTransposeMatrixd(const GLdouble * m)`
- `void glLoadTransposeMatrixf(const GLfloat * m)`
- `void glLogicOp(GLenum opcode)`
- `void glMap1f(GLenum target,GLfloat u1,GLfloat u2,GLint stride,GLint order,const GLfloat * points)`
- `void glMap1d(GLenum target,GLdouble u1,GLdouble u2,GLint stride,GLint order,const GLdouble * points)`

- `void glMap2f(GLenum target, GLfloat u1, GLfloat u2, GLint ustride, GLint uorder, GLfloat v1, GLfloat v2, GLint vstride, GLint vorder, const GLfloat * points)`
- `void glMap2d(GLenum target, GLdouble u1, GLdouble u2, GLint ustride, GLint uorder, GLdouble v1, GLdouble v2, GLint vstride, GLint vorder, const GLdouble * points)`
- `void * glMapBuffer(GLenum target, GLenum access)`
- `void glMapGrid1d(GLint un, GLdouble u1, GLdouble u2)`
- `void glMapGrid1f(GLint un, GLfloat u1, GLfloat u2)`
- `void glMapGrid2d(GLint un, GLdouble u1, GLdouble u2, GLint vn, GLdouble v1, GLdouble v2)`
- `void glMapGrid2f(GLint un, GLfloat u1, GLfloat u2, GLint vn, GLfloat v1, GLfloat v2)`
- `void glMaterialf(GLenum face, GLenum pname, GLfloat param)`
- `void glMateriali(GLenum face, GLenum pname, GLint param)`
- `void glMatrixMode(GLenum mode)`
- `void glMinmax(GLenum target, GLenum internalformat, GLboolean sink)`
- `void glMultMatrixd(const GLdouble * m)`
- `void glMultMatrixf(const GLfloat * m)`
- `void glMultTransposeMatrixd(const GLdouble * m)`
- `void glMultTransposeMatrixf(const GLfloat * m)`
- `void glMultiDrawArrays(GLenum mode, GLint * first, GLsizei * count, GLsizei primcount)`
- `void glMultiDrawElements(GLenum mode, const GLsizei * count, GLenum type, const GLvoid ** indices, GLsizei primcount)`
- `void glMultiTexCoord1s(GLenum target, GLshort s)`
- `void glMultiTexCoord1i(GLenum target, GLint s)`
- `void glMultiTexCoord1f(GLenum target, GLfloat s)`
- `void glMultiTexCoord1d(GLenum target, GLdouble s)`
- `void glMultiTexCoord2s(GLenum target, GLshort s, GLshort t)`
- `void glMultiTexCoord2i(GLenum target, GLint s, GLint t)`
- `void glMultiTexCoord2f(GLenum target, GLfloat s, GLfloat t)`
- `void glMultiTexCoord2d(GLenum target, GLdouble s, GLdouble t)`
- `void glMultiTexCoord3s(GLenum target, GLshort s, GLshort t, GLshort r)`
- `void glMultiTexCoord3i(GLenum target, GLint s, GLint t, GLint r)`
- `void glMultiTexCoord3f(GLenum target, GLfloat s, GLfloat t, GLfloat r)`
- `void glMultiTexCoord3d(GLenum target, GLdouble s, GLdouble t, GLdouble r)`
- `void glMultiTexCoord4s(GLenum target, GLshort s, GLshort t, GLshort r, GLshort q)`
- `void glMultiTexCoord4i(GLenum target, GLint s, GLint t, GLint r, GLint q)`
- `void glMultiTexCoord4f(GLenum target, GLfloat s, GLfloat t, GLfloat r, GLfloat q)`
- `void glMultiTexCoord4d(GLenum target, GLdouble s, GLdouble t, GLdouble r, GLdouble q)`
- `void glMultiTexCoord1sv(GLenum target, const GLshort * v)`

- `void glMultiTexCoord1iv(GLenum target,const GLint * v)`
- `void glMultiTexCoord1fv(GLenum target,const GLfloat * v)`
- `void glMultiTexCoord1dv(GLenum target,const GLdouble * v)`
- `void glMultiTexCoord2sv(GLenum target,const GLshort * v)`
- `void glMultiTexCoord2iv(GLenum target,const GLint * v)`
- `void glMultiTexCoord2fv(GLenum target,const GLfloat * v)`
- `void glMultiTexCoord2dv(GLenum target,const GLdouble * v)`
- `void glMultiTexCoord3sv(GLenum target,const GLshort * v)`
- `void glMultiTexCoord3iv(GLenum target,const GLint * v)`
- `void glMultiTexCoord3fv(GLenum target,const GLfloat * v)`
- `void glMultiTexCoord3dv(GLenum target,const GLdouble * v)`
- `void glMultiTexCoord4sv(GLenum target,const GLshort * v)`
- `void glMultiTexCoord4iv(GLenum target,const GLint * v)`
- `void glMultiTexCoord4fv(GLenum target,const GLfloat * v)`
- `void glMultiTexCoord4dv(GLenum target,const GLdouble * v)`
- `void glNewList(GLuint list,GLenum mode)`
- `void glNormal3b(GLbyte nx,GLbyte ny,GLbyte nz)`
- `void glNormal3d(GLdouble nx,GLdouble ny,GLdouble nz)`
- `void glNormal3f(GLfloat nx,GLfloat ny,GLfloat nz)`
- `void glNormal3i(GLint nx,GLint ny,GLint nz)`
- `void glNormal3s(GLshort nx,GLshort ny,GLshort nz)`
- `void glNormal3bv(const GLbyte * v)`
- `void glNormal3dv(const GLdouble * v)`
- `void glNormal3fv(const GLfloat * v)`
- `void glNormal3iv(const GLint * v)`
- `void glNormal3sv(const GLshort * v)`
- `void glNormalPointer(GLenum type,GLsizei stride,const GLvoid * pointer)`
- `void glOrtho(GLdouble left,GLdouble right,GLdouble bottom,GLdouble top,GLdouble nearVal,GLdouble farVal)`
- `void glPassThrough(GLfloat token)`
- `void glPixelMapfv(GLenum map,GLsizei mapsize,const GLfloat * values)`
- `void glPixelMapuiv(GLenum map,GLsizei mapsize,const GLuint * values)`
- `void glPixelMapusv(GLenum map,GLsizei mapsize,const GLushort * values)`
- `void glPixelStoref(GLenum pname,GLfloat param)`
- `void glPixelStorei(GLenum pname,GLint param)`
- `void glPixelTransferf(GLenum pname,GLfloat param)`

- void glPixelTransferi(GLenum pname,GLint param)
- void glPixelZoom(GLfloat xfactor,GLfloat yfactor)
- void glPointParameterf(GLenum pname,GLfloat param)
- void glPointParameteri(GLenum pname,GLint param)
- void glPointSize(GLfloat size)
- void glPolygonMode(GLenum face,GLenum mode)
- void glPolygonOffset(GLfloat factor,GLfloat units)
- void glPolygonStipple(const GLubyte * pattern)
- void glPushAttrib(GLbitfield mask)
- void glPushClientAttrib(GLbitfield mask)
- void glPushMatrix(void)
- void glPushName(GLuint name)
- void glPrioritizeTextures(GLsizei n,const GLuint * textures,const GLclampf * priorities)
- void glPopMatrix(void)
- void glRasterPos2s(GLshort x,GLshort y)
- void glRasterPos2i(GLint x,GLint y)
- void glRasterPos2f(GLfloat x,GLfloat y)
- void glRasterPos2d(GLdouble x,GLdouble y)
- void glRasterPos3s(GLshort x,GLshort y,GLshort z)
- void glRasterPos3i(GLint x,GLint y,GLint z)
- void glRasterPos3f(GLfloat x,GLfloat y,GLfloat z)
- void glRasterPos3d(GLdouble x,GLdouble y,GLdouble z)
- void glRasterPos4s(GLshort x,GLshort y,GLshort z,GLshort w)
- void glRasterPos4i(GLint x,GLint y,GLint z,GLint w)
- void glRasterPos4f(GLfloat x,GLfloat y,GLfloat z,GLfloat w)
- void glRasterPos4d(GLdouble x,GLdouble y,GLdouble z,GLdouble w)
- void glReadBuffer(GLenum mode)
- void glReadPixels(GLint x,GLint y,GLsizei width,GLsizei height,GLenum format,GLenum type,GLvoid * data)
- void glRectd(GLdouble x1,GLdouble y1,GLdouble x2,GLdouble y2)
- void glRectf(GLfloat x1,GLfloat y1,GLfloat x2,GLfloat y2)
- void glRecti(GLint x1,GLint y1,GLint x2,GLint y2)
- void glRects(GLshort x1,GLshort y1,GLshort x2,GLshort y2)
- void glRectdv(const GLdouble * v1,const GLdouble * v2)
- void glRectfv(const GLfloat * v1,const GLfloat * v2)
- void glRectiv(const GLint * v1,const GLint * v2)
- void glRectsv(const GLshort * v1,const GLshort * v2)

- `GLint glRenderMode(GLenum mode)`
- `void glResetHistogram(GLenum target)`
- `void glRotated(GLdouble angle, GLdouble x, GLdouble y, GLdouble z)`
- `void glRotatef(GLfloat angle, GLfloat x, GLfloat y, GLfloat z)`
- `void glSampleCoverage(GLclampf value, GLboolean invert)`
- `void glScaled(GLdouble x, GLdouble y, GLdouble z)`
- `void glScalef(GLfloat x, GLfloat y, GLfloat z)`
- `void glScissor(GLint x, GLint y, GLsizei width, GLsizei height)`
- `void glSecondaryColor3b(GLbyte red, GLbyte green, GLbyte blue)`
- `void glSecondaryColor3s(GLshort red, GLshort green, GLshort blue)`
- `void glSecondaryColor3i(GLint red, GLint green, GLint blue)`
- `void glSecondaryColor3f(GLfloat red, GLfloat green, GLfloat blue)`
- `void glSecondaryColor3d(GLdouble red, GLdouble green, GLdouble blue)`
- `void glSecondaryColor3ub(GLubyte red, GLubyte green, GLubyte blue)`
- `void glSecondaryColor3us(GLushort red, GLushort green, GLushort blue)`
- `void glSecondaryColor3ui(GLuint red, GLuint green, GLuint blue)`
- `void glSecondaryColor3bv(const GLbyte * v)`
- `void glSecondaryColor3sv(const GLshort * v)`
- `void glSecondaryColor3iv(const GLint * v)`
- `void glSecondaryColor3fv(const GLfloat * v)`
- `void glSecondaryColor3dv(const GLdouble * v)`
- `void glSecondaryColor3ubv(const GLubyte * v)`
- `void glSecondaryColor3usv(const GLushort * v)`
- `void glSecondaryColor3uiv(const GLuint * v)`
- `void glSecondaryColorPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)`
- `void glSelectBuffer(GLsizei size, GLuint * buffer)`
- `void glSeparableFilter2D(GLenum target, GLenum internalformat, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * row, const GLvoid * column)`
- `void glShadeModel(GLenum mode)`
- `void glShaderSource(GLuint shader, GLsizei count, const GLchar **string, const GLint *length)`
- `void glStencilFunc(GLenum func, GLint ref, GLuint mask)`
- `void glStencilFuncSeparate(GLenum face, GLenum func, GLint ref, GLuint mask)`
- `void glStencilMask(GLuint mask)`
- `void glStencilMaskSeparate(GLenum face, GLuint mask)`
- `void glStencilOp(GLenum sfail, GLenum dpfail, GLenum dppass)`
- `void glStencilOpSeparate(GLenum face, GLenum sfail, GLenum dpfail, GLenum dppass)`

- `void glTexCoord1s(GLshort s)`
- `void glTexCoord1i(GLint s)`
- `void glTexCoord1f(GLfloat s)`
- `void glTexCoord1d(GLdouble s)`
- `void glTexCoord2s(GLshort s, GLshort t)`
- `void glTexCoord2i(GLint s, GLint t)`
- `void glTexCoord2f(GLfloat s, GLfloat t)`
- `void glTexCoord2d(GLdouble s, GLdouble t)`
- `void glTexCoord3s(GLshort s, GLshort t, GLshort r)`
- `void glTexCoord3i(GLint s, GLint t, GLint r)`
- `void glTexCoord3f(GLfloat s, GLfloat t, GLfloat r)`
- `void glTexCoord3d(GLdouble s, GLdouble t, GLdouble r)`
- `void glTexCoord4s(GLshort s, GLshort t, GLshort r, GLshort q)`
- `void glTexCoord4i(GLint s, GLint t, GLint r, GLint q)`
- `void glTexCoord4f(GLfloat s, GLfloat t, GLfloat r, GLfloat q)`
- `void glTexCoord4d(GLdouble s, GLdouble t, GLdouble r, GLdouble q)`
- `void glTexCoord1sv(const GLshort * v)`
- `void glTexCoord1iv(const GLint * v)`
- `void glTexCoord1fv(const GLfloat * v)`
- `void glTexCoord1dv(const GLdouble * v)`
- `void glTexCoord2sv(const GLshort * v)`
- `void glTexCoord2iv(const GLint * v)`
- `void glTexCoord2fv(const GLfloat * v)`
- `void glTexCoord2dv(const GLdouble * v)`
- `void glTexCoord3sv(const GLshort * v)`
- `void glTexCoord3iv(const GLint * v)`
- `void glTexCoord3fv(const GLfloat * v)`
- `void glTexCoord3dv(const GLdouble * v)`
- `void glTexCoord4sv(const GLshort * v)`
- `void glTexCoord4iv(const GLint * v)`
- `void glTexCoord4fv(const GLfloat * v)`
- `void glTexCoord4dv(const GLdouble * v)`
- `void glTexCoordPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)`
- `void glTexEnvf(GLenum target, GLenum pname, GLfloat param)`
- `void glTexEnvf(GLenum target, GLenum pname, GLint param)`
- `void glTexGeni(GLenum coord, GLenum pname, GLint param)`

- void glTexGenf(GLenum coord, GLenum pname, GLfloat param)
- void glTexGend(GLenum coord, GLenum pname, GLdouble param)
- void glTexGeniv(GLenum coord, GLenum pname, const GLint * params)
- void glTexGenfv(GLenum coord, GLenum pname, const GLfloat * params)
- void glTexGendv(GLenum coord, GLenum pname, const GLdouble * params)
- void glTexImage1D(GLenum target, GLint level, GLint internalFormat, GLsizei width, GLint border, GLenum format, GLenum type, const GLvoid * data)
- void glTexImage2D(GLenum target, GLint level, GLint internalFormat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, const GLvoid * data)
- void glTexImage3D(GLenum target, GLint level, GLint internalFormat, GLsizei width, GLsizei height, GLsizei depth, GLint border, GLenum format, GLenum type, const GLvoid * data)
- void glTexParameterf(GLenum target, GLenum pname, GLfloat param)
- void glTexParameteri(GLenum target, GLenum pname, GLint param)
- void glTexParameterfv(GLenum target, GLenum pname, const GLfloat * params)
- void glTexParameteriv(GLenum target, GLenum pname, const GLint * params)
- void glTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLenum type, const GLvoid * data)
- void glTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)
- void glTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLenum type, const GLvoid * data)
- void glTranslated(GLdouble x, GLdouble y, GLdouble z)
- void glTranslatef(GLfloat x, GLfloat y, GLfloat z)
- void glUniform1f(GLint location, GLfloat v0)
- void glUniform2f(GLint location, GLfloat v0, GLfloat v1)
- void glUniform3f(GLint location, GLfloat v0, GLfloat v1, GLfloat v2)
- void glUniform4f(GLint location, GLfloat v0, GLfloat v1, GLfloat v2, GLfloat v3)
- void glUniform1i(GLint location, GLint v0)
- void glUniform2i(GLint location, GLint v0, GLint v1)
- void glUniform3i(GLint location, GLint v0, GLint v1, GLint v2)
- void glUniform4i(GLint location, GLint v0, GLint v1, GLint v2, GLint v3)
- void glUniform1fv(GLint location, GLsizei count, const GLfloat *value)
- void glUniform2fv(GLint location, GLsizei count, const GLfloat *value)
- void glUniform3fv(GLint location, GLsizei count, const GLfloat *value)
- void glUniform4fv(GLint location, GLsizei count, const GLfloat *value)
- void glUniform1iv(GLint location, GLsizei count, const GLint *value)
- void glUniform2iv(GLint location, GLsizei count, const GLint *value)
- void glUniform3iv(GLint location, GLsizei count, const GLint *value)

- `void glUniform4iv(GLint location, GLsizei count, const GLint *value)`
- `void glUniformMatrix2fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix3fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix4fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix2x3fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix3x2fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix2x4fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix4x2fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix3x4fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix4x3fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUseProgram(GLuint program)`
- `void glValidateProgram(GLuint program)`
- `void glVertex2s(GLshort x, GLshort y)`
- `void glVertex2i(GLint x, GLint y)`
- `void glVertex2f(GLfloat x, GLfloat y)`
- `void glVertex2d(GLdouble x, GLdouble y)`
- `void glVertex3s(GLshort x, GLshort y, GLshort z)`
- `void glVertex3i(GLint x, GLint y, GLint z)`
- `void glVertex3f(GLfloat x, GLfloat y, GLfloat z)`
- `void glVertex3d(GLdouble x, GLdouble y, GLdouble z)`
- `void glVertex4s(GLshort x, GLshort y, GLshort z, GLshort w)`
- `void glVertex4i(GLint x, GLint y, GLint z, GLint w)`
- `void glVertex4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)`
- `void glVertex4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)`
- `void glVertex2sv(const GLshort * v)`
- `void glVertex2iv(const GLint * v)`
- `void glVertex2fv(const GLfloat * v)`
- `void glVertex2dv(const GLdouble * v)`
- `void glVertex3sv(const GLshort * v)`
- `void glVertex3iv(const GLint * v)`
- `void glVertex3fv(const GLfloat * v)`
- `void glVertex3dv(const GLdouble * v)`
- `void glVertex4sv(const GLshort * v)`
- `void glVertex4iv(const GLint * v)`
- `void glVertex4fv(const GLfloat * v)`
- `void glVertex4dv(const GLdouble * v)`

- `void glVertexAttrib1f(GLuint index, GLfloat v0)`
- `void glVertexAttrib1s(GLuint index, GLshort v0)`
- `void glVertexAttrib1d(GLuint index, GLdouble v0)`
- `void glVertexAttrib2f(GLuint index, GLfloat v0, GLfloat v1)`
- `void glVertexAttrib2s(GLuint index, GLshort v0, GLshort v1)`
- `void glVertexAttrib2d(GLuint index, GLdouble v0, GLdouble v1)`
- `void glVertexAttrib3f(GLuint index, GLfloat v0, GLfloat v1, GLfloat v2)`
- `void glVertexAttrib3s(GLuint index, GLshort v0, GLshort v1, GLshort v2)`
- `void glVertexAttrib3d(GLuint index, GLdouble v0, GLdouble v1, GLdouble v2)`
- `void glVertexAttrib4f(GLuint index, GLfloat v0, GLfloat v1, GLfloat v2, GLfloat v3)`
- `void glVertexAttrib4s(GLuint index, GLshort v0, GLshort v1, GLshort v2, GLshort v3)`
- `void glVertexAttrib4d(GLuint index, GLdouble v0, GLdouble v1, GLdouble v2, GLdouble v3)`
- `void glVertexAttrib4Nub(GLuint index, GLubyte v0, GLubyte v1, GLubyte v2, GLubyte v3)`
- `void glVertexAttrib1fv(GLuint index, const GLfloat *v)`
- `void glVertexAttrib1sv(GLuint index, const GLshort *v)`
- `void glVertexAttrib1dv(GLuint index, const GLdouble *v)`
- `void glVertexAttrib2fv(GLuint index, const GLfloat *v)`
- `void glVertexAttrib2sv(GLuint index, const GLshort *v)`
- `void glVertexAttrib2dv(GLuint index, const GLdouble *v)`
- `void glVertexAttrib3fv(GLuint index, const GLfloat *v)`
- `void glVertexAttrib3sv(GLuint index, const GLshort *v)`
- `void glVertexAttrib3dv(GLuint index, const GLdouble *v)`
- `void glVertexAttrib4fv(GLuint index, const GLfloat *v)`
- `void glVertexAttrib4sv(GLuint index, const GLshort *v)`
- `void glVertexAttrib4dv(GLuint index, const GLdouble *v)`
- `void glVertexAttrib4iv(GLuint index, const GLint *v)`
- `void glVertexAttrib4bv(GLuint index, const GLbyte *v)`
- `void glVertexAttrib4ubv(GLuint index, const GLubyte *v)`
- `void glVertexAttrib4usv(GLuint index, const GLushort *v)`
- `void glVertexAttrib4uiv(GLuint index, const GLuint *v)`
- `void glVertexAttribPointer(GLuint index, GLint size, GLenum type, GLboolean normalized, GLsizei stride, const GLvoid * pointer)`
- `void glVertexPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)`
- `void glViewport(GLint x, GLint y, GLsizei width, GLsizei height)`
- `void glWindowPos2s(GLshort x, GLshort y)`
- `void glWindowPos2i(GLint x, GLint y)`

- void glWindowPos2f(GLfloat x,GLfloat y)
- void glWindowPos2d(GLdouble x,GLdouble y)
- void glWindowPos3s(GLshort x,GLshort y,GLshort z)
- void glWindowPos3i(GLint x,GLint y,GLint z)
- void glWindowPos3f(GLfloat x,GLfloat y,GLfloat z)
- void glWindowPos3d(GLdouble x,GLdouble y,GLdouble z)
- void glWindowPos2sv(const GLshort * v)
- void glWindowPos2iv(const GLint * v)
- void glWindowPos2fv(const GLfloat * v)
- void glWindowPos2dv(const GLdouble * v)
- void glWindowPos3sv(const GLshort * v)
- void glWindowPos3iv(const GLint * v)
- void glWindowPos3fv(const GLfloat * v)
- void glWindowPos3dv(const GLdouble * v)
- void gluBeginCurve(GLUnurbs* nurb)
- void gluBeginPolygon(GLUtesselator* tess)
- void gluBeginSurface(GLUnurbs* nurb)
- void gluBeginTrim(GLUnurbs* nurb)
- void gluCylinder(GLUquadric* quad,GLdouble base,GLdouble top,GLdouble height,GLint slices,GLint stacks)
- void gluDeleteNurbsRenderer(GLUnurbs* nurb)
- void gluDeleteQuadric(GLUquadric* quad)
- void gluDeleteTess(GLUtesselator* tess)
- void gluDisk(GLUquadric* quad,GLdouble inner,GLdouble outer,GLint slices,GLint loops)
- void gluEndCurve(GLUnurbs* nurb)
- void gluEndPolygon(GLUtesselator* tess)
- void gluEndSurface(GLUnurbs* nurb)
- void gluEndTrim(GLUnurbs* nurb)
- const GLubyte * gluErrorString(GLenum error)
- void gluGetNurbsProperty(GLUnurbs* nurb,GLenum property,GLfloat* data)
- const GLubyte * gluGetString(GLenum name)
- void gluGetTessProperty(GLUtesselator* tess,GLenum which,GLdouble* data)
- void gluLoadSamplingMatrices(GLUnurbs* nurb,const GLfloat * model,const GLfloat * perspective,const GLint * view)
- void gluLookAt(GLdouble eyeX,GLdouble eyeY,GLdouble eyeZ,GLdouble centerX,GLdouble centerY,GLdouble centerZ,GLdouble upX,GLdouble upY,GLdouble upZ)
- GLUnurbs *gluNewNurbsRenderer(void)
- GLUquadric *gluNewQuadric(void)

- `GLUtesselator* gluNewTess(void)`
- `void gluNextContour(GLUtesselator* tess, GLenum type)`
- `void gluNurbsCurve(GLUnurbs* nurb, GLint knotCount, GLfloat * knots, GLint stride, GLfloat * control, GLint order, GLenum type)`
- `void gluNurbsProperty(GLUnurbs* nurb, GLenum property, GLfloat value)`
- `void gluNurbsSurface(GLUnurbs* nurb, GLint sKnotCount, GLfloat* sKnots, GLint tKnotCount, GLfloat* tKnots, GLint sStride, GLint tStride, GLfloat* control, GLint sOrder, GLint tOrder, GLenum type)`
- `void gluOrtho2D(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top)`
- `void gluPartialDisk(GLUquadric* quad, GLdouble inner, GLdouble outer, GLint slices, GLint loops, GLdouble start, GLdouble sweep)`
- `void gluPerspective(GLdouble fovy, GLdouble aspect, GLdouble zNear, GLdouble zFar)`
- `void gluPickMatrix(GLdouble x, GLdouble y, GLdouble delX, GLdouble delY, GLint * viewport)`
- `GLint gluProject(GLdouble objX, GLdouble objY, GLdouble objZ, const GLdouble * model, const GLdouble * proj, const GLint * view, GLdouble* winX, GLdouble* winY, GLdouble* winZ)`
- `void gluPwlCurve(GLUnurbs* nurb, GLint count, GLfloat* data, GLint stride, GLenum type)`
- `void gluQuadricDrawStyle(GLUquadric* quad, GLenum draw)`
- `void gluQuadricNormals(GLUquadric* quad, GLenum normal)`
- `void gluQuadricOrientation(GLUquadric* quad, GLenum orientation)`
- `void gluQuadricTexture(GLUquadric* quad, GLboolean texture)`
- `GLint gluScaleImage(GLenum format, GLsizei wIn, GLsizei hIn, GLenum typeIn, const void * dataIn, GLsizei wOut, GLsizei hOut, GLenum typeOut, GLvoid* dataOut)`
- `void gluSphere(GLUquadric* quad, GLdouble radius, GLint slices, GLint stacks)`
- `void gluTessBeginContour(GLUtesselator* tess)`
- `void gluTessBeginPolygon(GLUtesselator* tess, GLvoid* data)`
- `void gluTessEndContour(GLUtesselator* tess)`
- `void gluTessEndPolygon(GLUtesselator* tess)`
- `void gluTessNormal(GLUtesselator* tess, GLdouble valueX, GLdouble valueY, GLdouble valueZ)`
- `void gluTessProperty(GLUtesselator* tess, GLenum which, GLdouble data)`
- `void gluTessVertex(GLUtesselator* tess, GLdouble * location, GLvoid* data)`
- `GLint gluUnProject(GLdouble winX, GLdouble winY, GLdouble winZ, const GLdouble * model, const GLdouble * proj, const GLint * view, GLdouble* objX, GLdouble* objY, GLdouble* objZ)`
- `void glDisable(GLenum cap)`

RINGOPENGL (OPENGL 4.5) FUNCTIONS REFERENCE

- GL_ZERO
- GL_FALSE
- GL_LOGIC_OP
- GL_NONE
- GL_TEXTURE_COMPONENTS
- GL_NO_ERROR
- GL_POINTS
- GL_CURRENT_BIT
- GL_TRUE
- GL_ONE
- GL_CLIENT_PIXEL_STORE_BIT
- GL_LINES
- GL_LINE_LOOP
- GL_POINT_BIT
- GL_CLIENT_VERTEX_ARRAY_BIT
- GL_LINE_STRIP
- GL_LINE_BIT
- GL_TRIANGLES
- GL_TRIANGLE_STRIP
- GL_TRIANGLE_FAN
- GL_QUADS
- GL_QUAD_STRIP
- GL_POLYGON_BIT
- GL_POLYGON
- GL_POLYGON_STIPPLE_BIT
- GL_PIXEL_MODE_BIT
- GL_LIGHTING_BIT

- GL_FOG_BIT
- GL_DEPTH_BUFFER_BIT
- GL_ACCUM
- GL_LOAD
- GL_RETURN
- GL_MULT
- GL_ADD
- GL_NEVER
- GL_ACCUM_BUFFER_BIT
- GL_LESS
- GL_EQUAL
- GL_LEQUAL
- GL_GREATER
- GL_NOTEQUAL
- GL_GEQUAL
- GL_ALWAYS
- GL_SRC_COLOR
- GL_ONE_MINUS_SRC_COLOR
- GL_SRC_ALPHA
- GL_ONE_MINUS_SRC_ALPHA
- GL_DST_ALPHA
- GL_ONE_MINUS_DST_ALPHA
- GL_DST_COLOR
- GL_ONE_MINUS_DST_COLOR
- GL_SRC_ALPHA_SATURATE
- GL_STENCIL_BUFFER_BIT
- GL_FRONT_LEFT
- GL_FRONT_RIGHT
- GL_BACK_LEFT
- GL_BACK_RIGHT
- GL_FRONT
- GL_BACK
- GL_LEFT
- GL_RIGHT
- GL_FRONT_AND_BACK
- GL_AUX0

- GL_AUX1
- GL_AUX2
- GL_AUX3
- GL_INVALID_ENUM
- GL_INVALID_VALUE
- GL_INVALID_OPERATION
- GL_STACK_OVERFLOW
- GL_STACK_UNDERFLOW
- GL_OUT_OF_MEMORY
- GL_2D
- GL_3D
- GL_3D_COLOR
- GL_3D_COLOR_TEXTURE
- GL_4D_COLOR_TEXTURE
- GL_PASS_THROUGH_TOKEN
- GL_POINT_TOKEN
- GL_LINE_TOKEN
- GL_POLYGON_TOKEN
- GL_BITMAP_TOKEN
- GL_DRAW_PIXEL_TOKEN
- GL_COPY_PIXEL_TOKEN
- GL_LINE_RESET_TOKEN
- GL_EXP
- GL_VIEWPORT_BIT
- GL_EXP2
- GL_CW
- GL_CCW
- GL_COEFF
- GL_ORDER
- GL_DOMAIN
- GL_CURRENT_COLOR
- GL_CURRENT_INDEX
- GL_CURRENT_NORMAL
- GL_CURRENT_TEXTURE_COORDS
- GL_CURRENT_RASTER_COLOR
- GL_CURRENT_RASTER_INDEX

- GL_CURRENT_RASTER_TEXTURE_COORDS
- GL_CURRENT_RASTER_POSITION
- GL_CURRENT_RASTER_POSITION_VALID
- GL_CURRENT_RASTER_DISTANCE
- GL_POINT_SMOOTH
- GL_POINT_SIZE
- GL_POINT_SIZE_RANGE
- GL_POINT_SIZE_GRANULARITY
- GL_LINE_SMOOTH
- GL_LINE_WIDTH
- GL_LINE_WIDTH_RANGE
- GL_LINE_WIDTH_GRANULARITY
- GL_LINE_STIPPLE
- GL_LINE_STIPPLE_PATTERN
- GL_LINE_STIPPLE_REPEAT
- GL_LIST_MODE
- GL_MAX_LIST_NESTING
- GL_LIST_BASE
- GL_LIST_INDEX
- GL_POLYGON_MODE
- GL_POLYGON_SMOOTH
- GL_POLYGON_STIPPLE
- GL_EDGE_FLAG
- GL_CULL_FACE
- GL_CULL_FACE_MODE
- GL_FRONT_FACE
- GL_LIGHTING
- GL_LIGHT_MODEL_LOCAL_VIEWER
- GL_LIGHT_MODEL_TWO_SIDE
- GL_LIGHT_MODEL_AMBIENT
- GL_SHADE_MODEL
- GL_COLOR_MATERIAL_FACE
- GL_COLOR_MATERIAL_PARAMETER
- GL_COLOR_MATERIAL
- GL_FOG
- GL_FOG_INDEX

- GL_FOG_DENSITY
- GL_FOG_START
- GL_FOG_END
- GL_FOG_MODE
- GL_FOG_COLOR
- GL_DEPTH_RANGE
- GL_DEPTH_TEST
- GL_DEPTH_WRITEMASK
- GL_DEPTH_CLEAR_VALUE
- GL_DEPTH_FUNC
- GL_ACCUM_CLEAR_VALUE
- GL_STENCIL_TEST
- GL_STENCIL_CLEAR_VALUE
- GL_STENCIL_FUNC
- GL_STENCIL_VALUE_MASK
- GL_STENCIL_FAIL
- GL_STENCIL_PASS_DEPTH_FAIL
- GL_STENCIL_PASS_DEPTH_PASS
- GL_STENCIL_REF
- GL_STENCIL_WRITEMASK
- GL_MATRIX_MODE
- GL_NORMALIZE
- GL_VIEWPORT
- GL_MODELVIEW_STACK_DEPTH
- GL_PROJECTION_STACK_DEPTH
- GL_TEXTURE_STACK_DEPTH
- GL_MODELVIEW_MATRIX
- GL_PROJECTION_MATRIX
- GL_TEXTURE_MATRIX
- GL_ATTRIB_STACK_DEPTH
- GL_CLIENT_ATTRIB_STACK_DEPTH
- GL_ALPHA_TEST
- GL_ALPHA_TEST_FUNC
- GL_ALPHA_TEST_REF
- GL_DITHER
- GL_BLEND_DST

- GL_BLEND_SRC
- GL_BLEND
- GL_LOGIC_OP_MODE
- GL_INDEX_LOGIC_OP
- GL_COLOR_LOGIC_OP
- GL_AUX_BUFFERS
- GL_DRAW_BUFFER
- GL_READ_BUFFER
- GL_SCISSOR_BOX
- GL_SCISSOR_TEST
- GL_INDEX_CLEAR_VALUE
- GL_INDEX_WRITEMASK
- GL_COLOR_CLEAR_VALUE
- GL_COLOR_WRITEMASK
- GL_INDEX_MODE
- GL_RGBA_MODE
- GL_DOUBLEBUFFER
- GL_STEREO
- GL_RENDER_MODE
- GL_PERSPECTIVE_CORRECTION_HINT
- GL_POINT_SMOOTH_HINT
- GL_LINE_SMOOTH_HINT
- GL_POLYGON_SMOOTH_HINT
- GL_FOG_HINT
- GL_TEXTURE_GEN_S
- GL_TEXTURE_GEN_T
- GL_TEXTURE_GEN_R
- GL_TEXTURE_GEN_Q
- GL_PIXEL_MAP_I_TO_I
- GL_PIXEL_MAP_S_TO_S
- GL_PIXEL_MAP_I_TO_R
- GL_PIXEL_MAP_I_TO_G
- GL_PIXEL_MAP_I_TO_B
- GL_PIXEL_MAP_I_TO_A
- GL_PIXEL_MAP_R_TO_R
- GL_PIXEL_MAP_G_TO_G

- GL_PIXEL_MAP_B_TO_B
- GL_PIXEL_MAP_A_TO_A
- GL_PIXEL_MAP_I_TO_I_SIZE
- GL_PIXEL_MAP_S_TO_S_SIZE
- GL_PIXEL_MAP_I_TO_R_SIZE
- GL_PIXEL_MAP_I_TO_G_SIZE
- GL_PIXEL_MAP_I_TO_B_SIZE
- GL_PIXEL_MAP_I_TO_A_SIZE
- GL_PIXEL_MAP_R_TO_R_SIZE
- GL_PIXEL_MAP_G_TO_G_SIZE
- GL_PIXEL_MAP_B_TO_B_SIZE
- GL_PIXEL_MAP_A_TO_A_SIZE
- GL_UNPACK_SWAP_BYTES
- GL_UNPACK_LSB_FIRST
- GL_UNPACK_ROW_LENGTH
- GL_UNPACK_SKIP_ROWS
- GL_UNPACK_SKIP_PIXELS
- GL_UNPACK_ALIGNMENT
- GL_PACK_SWAP_BYTES
- GL_PACK_LSB_FIRST
- GL_PACK_ROW_LENGTH
- GL_PACK_SKIP_ROWS
- GL_PACK_SKIP_PIXELS
- GL_PACK_ALIGNMENT
- GL_MAP_COLOR
- GL_MAP_STENCIL
- GL_INDEX_SHIFT
- GL_INDEX_OFFSET
- GL_RED_SCALE
- GL_RED_BIAS
- GL_ZOOM_X
- GL_ZOOM_Y
- GL_GREEN_SCALE
- GL_GREEN_BIAS
- GL_BLUE_SCALE
- GL_BLUE_BIAS

- GL_ALPHA_SCALE
- GL_ALPHA_BIAS
- GL_DEPTH_SCALE
- GL_DEPTH_BIAS
- GL_MAX_EVAL_ORDER
- GL_MAX_LIGHTS
- GL_MAX_CLIP_PLANES
- GL_MAX_TEXTURE_SIZE
- GL_MAX_PIXEL_MAP_TABLE
- GL_MAX_ATTRIB_STACK_DEPTH
- GL_MAX_MODELVIEW_STACK_DEPTH
- GL_MAX_NAME_STACK_DEPTH
- GL_MAX_PROJECTION_STACK_DEPTH
- GL_MAX_TEXTURE_STACK_DEPTH
- GL_MAX_VIEWPORT_DIMS
- GL_MAX_CLIENT_ATTRIB_STACK_DEPTH
- GL_SUBPIXEL_BITS
- GL_INDEX_BITS
- GL_RED_BITS
- GL_GREEN_BITS
- GL_BLUE_BITS
- GL_ALPHA_BITS
- GL_DEPTH_BITS
- GL_STENCIL_BITS
- GL_ACCUM_RED_BITS
- GL_ACCUM_GREEN_BITS
- GL_ACCUM_BLUE_BITS
- GL_ACCUM_ALPHA_BITS
- GL_NAME_STACK_DEPTH
- GL_AUTO_NORMAL
- GL_MAP1_COLOR_4
- GL_MAP1_INDEX
- GL_MAP1_NORMAL
- GL_MAP1_TEXTURE_COORD_1
- GL_MAP1_TEXTURE_COORD_2
- GL_MAP1_TEXTURE_COORD_3

- GL_MAP1_TEXTURE_COORD_4
- GL_MAP1_VERTEX_3
- GL_MAP1_VERTEX_4
- GL_MAP2_COLOR_4
- GL_MAP2_INDEX
- GL_MAP2_NORMAL
- GL_MAP2_TEXTURE_COORD_1
- GL_MAP2_TEXTURE_COORD_2
- GL_MAP2_TEXTURE_COORD_3
- GL_MAP2_TEXTURE_COORD_4
- GL_MAP2_VERTEX_3
- GL_MAP2_VERTEX_4
- GL_MAP1_GRID_DOMAIN
- GL_MAP1_GRID_SEGMENTS
- GL_MAP2_GRID_DOMAIN
- GL_MAP2_GRID_SEGMENTS
- GL_TEXTURE_1D
- GL_TEXTURE_2D
- GL_FEEDBACK_BUFFER_POINTER
- GL_FEEDBACK_BUFFER_SIZE
- GL_FEEDBACK_BUFFER_TYPE
- GL_SELECTION_BUFFER_POINTER
- GL_SELECTION_BUFFER_SIZE
- GL_TEXTURE_WIDTH
- GL_TRANSFORM_BIT
- GL_TEXTURE_HEIGHT
- GL_TEXTURE_INTERNAL_FORMAT
- GL_TEXTURE_BORDER_COLOR
- GL_TEXTURE_BORDER
- GL_DONT_CARE
- GL_FASTEST
- GL_NICEST
- GL_AMBIENT
- GL_DIFFUSE
- GL_SPECULAR
- GL_POSITION

- GL_SPOT_DIRECTION
- GL_SPOT_EXPONENT
- GL_SPOT_CUTOFF
- GL_CONSTANT_ATTENUATION
- GL_LINEAR_ATTENUATION
- GL_QUADRATIC_ATTENUATION
- GL_COMPILE
- GL_COMPILE_AND_EXECUTE
- GL_BYTE
- GL_UNSIGNED_BYTE
- GL_SHORT
- GL_UNSIGNED_SHORT
- GL_INT
- GL_UNSIGNED_INT
- GL_FLOAT
- GL_2_BYTES
- GL_3_BYTES
- GL_4_BYTES
- GL_DOUBLE
- GL_CLEAR
- GL_AND
- GL_AND_REVERSE
- GL_COPY
- GL_AND_INVERTED
- GL_NOOP
- GL_XOR
- GL_OR
- GL_NOR
- GL_EQUIV
- GL_INVERT
- GL_OR_REVERSE
- GL_COPY_INVERTED
- GL_OR_INVERTED
- GL_NAND
- GL_SET
- GL_EMISSION

- GL_SHININESS
- GL_AMBIENT_AND_DIFFUSE
- GL_COLOR_INDEXES
- GL_MODELVIEW
- GL_PROJECTION
- GL_TEXTURE
- GL_COLOR
- GL_DEPTH
- GL_STENCIL
- GL_COLOR_INDEX
- GL_STENCIL_INDEX
- GL_DEPTH_COMPONENT
- GL_RED
- GL_GREEN
- GL_BLUE
- GL_ALPHA
- GL_RGB
- GL_RGBA
- GL_LUMINANCE
- GL_LUMINANCE_ALPHA
- GL_BITMAP
- GL_POINT
- GL_LINE
- GL_FILL
- GL_RENDER
- GL_FEEDBACK
- GL_SELECT
- GL_FLAT
- GL_SMOOTH
- GL_KEEP
- GL_REPLACE
- GL_INCR
- GL_DECR
- GL_VENDOR
- GL_RENDERER
- GL_VERSION

- GL_EXTENSIONS
- GL_S
- GL_ENABLE_BIT
- GL_T
- GL_R
- GL_Q
- GL_MODULATE
- GL_DECAL
- GL_TEXTURE_ENV_MODE
- GL_TEXTURE_ENV_COLOR
- GL_TEXTURE_ENV
- GL_EYE_LINEAR
- GL_OBJECT_LINEAR
- GL_SPHERE_MAP
- GL_TEXTURE_GEN_MODE
- GL_OBJECT_PLANE
- GL_EYE_PLANE
- GL_NEAREST
- GL_LINEAR
- GL_NEAREST_MIPMAP_NEAREST
- GL_LINEAR_MIPMAP_NEAREST
- GL_NEAREST_MIPMAP_LINEAR
- GL_LINEAR_MIPMAP_LINEAR
- GL_TEXTURE_MAG_FILTER
- GL_TEXTURE_MIN_FILTER
- GL_TEXTURE_WRAP_S
- GL_TEXTURE_WRAP_T
- GL_CLAMP
- GL_REPEAT
- GL_POLYGON_OFFSET_UNITS
- GL_POLYGON_OFFSET_POINT
- GL_POLYGON_OFFSET_LINE
- GL_R3_G3_B2
- GL_V2F
- GL_V3F
- GL_C4UB_V2F

- GL_C4UB_V3F
- GL_C3F_V3F
- GL_N3F_V3F
- GL_C4F_N3F_V3F
- GL_T2F_V3F
- GL_T4F_V4F
- GL_T2F_C4UB_V3F
- GL_T2F_C3F_V3F
- GL_T2F_N3F_V3F
- GL_T2F_C4F_N3F_V3F
- GL_T4F_C4F_N3F_V4F
- GL_CLIP_PLANE0
- GL_CLIP_PLANE1
- GL_CLIP_PLANE2
- GL_CLIP_PLANE3
- GL_CLIP_PLANE4
- GL_CLIP_PLANE5
- GL_LIGHT0
- GL_COLOR_BUFFER_BIT
- GL_LIGHT1
- GL_LIGHT2
- GL_LIGHT3
- GL_LIGHT4
- GL_LIGHT5
- GL_LIGHT6
- GL_LIGHT7
- GL_HINT_BIT
- GL_POLYGON_OFFSET_FILL
- GL_POLYGON_OFFSET_FACTOR
- GL_ALPHA4
- GL_ALPHA8
- GL_ALPHA12
- GL_ALPHA16
- GL_LUMINANCE4
- GL_LUMINANCE8
- GL_LUMINANCE12

- GL_LUMINANCE16
- GL_LUMINANCE4_ALPHA4
- GL_LUMINANCE6_ALPHA2
- GL_LUMINANCE8_ALPHA8
- GL_LUMINANCE12_ALPHA4
- GL_LUMINANCE12_ALPHA12
- GL_LUMINANCE16_ALPHA16
- GL_INTENSITY
- GL_INTENSITY4
- GL_INTENSITY8
- GL_INTENSITY12
- GL_INTENSITY16
- GL_RGB4
- GL_RGB5
- GL_RGB8
- GL_RGB10
- GL_RGB12
- GL_RGB16
- GL_RGBA2
- GL_RGBA4
- GL_RGB5_A1
- GL_RGBA8
- GL_RGB10_A2
- GL_RGBA12
- GL_RGBA16
- GL_TEXTURE_RED_SIZE
- GL_TEXTURE_GREEN_SIZE
- GL_TEXTURE_BLUE_SIZE
- GL_TEXTURE_ALPHA_SIZE
- GL_TEXTURE_LUMINANCE_SIZE
- GL_TEXTURE_INTENSITY_SIZE
- GL_PROXY_TEXTURE_1D
- GL_PROXY_TEXTURE_2D
- GL_TEXTURE_PRIORITY
- GL_TEXTURE_RESIDENT
- GL_TEXTURE_BINDING_1D

- GL_TEXTURE_BINDING_2D
- GL_VERTEX_ARRAY
- GL_NORMAL_ARRAY
- GL_COLOR_ARRAY
- GL_INDEX_ARRAY
- GL_TEXTURE_COORD_ARRAY
- GL_EDGE_FLAG_ARRAY
- GL_VERTEX_ARRAY_SIZE
- GL_VERTEX_ARRAY_TYPE
- GL_VERTEX_ARRAY_STRIDE
- GL_NORMAL_ARRAY_TYPE
- GL_NORMAL_ARRAY_STRIDE
- GL_COLOR_ARRAY_SIZE
- GL_COLOR_ARRAY_TYPE
- GL_COLOR_ARRAY_STRIDE
- GL_INDEX_ARRAY_TYPE
- GL_INDEX_ARRAY_STRIDE
- GL_TEXTURE_COORD_ARRAY_SIZE
- GL_TEXTURE_COORD_ARRAY_TYPE
- GL_TEXTURE_COORD_ARRAY_STRIDE
- GL_EDGE_FLAG_ARRAY_STRIDE
- GL_VERTEX_ARRAY_POINTER
- GL_NORMAL_ARRAY_POINTER
- GL_COLOR_ARRAY_POINTER
- GL_INDEX_ARRAY_POINTER
- GL_TEXTURE_COORD_ARRAY_POINTER
- GL_EDGE_FLAG_ARRAY_POINTER
- GL_COLOR_INDEX1_EXT
- GL_COLOR_INDEX2_EXT
- GL_COLOR_INDEX4_EXT
- GL_COLOR_INDEX8_EXT
- GL_COLOR_INDEX12_EXT
- GL_COLOR_INDEX16_EXT
- GL_EVAL_BIT
- GL_LIST_BIT
- GL_TEXTURE_BIT

- GL_SCISSOR_BIT
- GL_ALL_ATTRIB_BITS
- GL_CLIENT_ALL_ATTRIB_BITS
- GL_SMOOTH_POINT_SIZE_RANGE
- GL_SMOOTH_POINT_SIZE_GRANULARITY
- GL_SMOOTH_LINE_WIDTH_RANGE
- GL_SMOOTH_LINE_WIDTH_GRANULARITY
- GL_UNSIGNED_BYTE_3_3_2
- GL_UNSIGNED_SHORT_4_4_4_4
- GL_UNSIGNED_SHORT_5_5_5_1
- GL_UNSIGNED_INT_8_8_8_8
- GL_UNSIGNED_INT_10_10_10_2
- GL_RESCALE_NORMAL
- GL_TEXTURE_BINDING_3D
- GL_PACK_SKIP_IMAGES
- GL_PACK_IMAGE_HEIGHT
- GL_UNPACK_SKIP_IMAGES
- GL_UNPACK_IMAGE_HEIGHT
- GL_TEXTURE_3D
- GL_PROXY_TEXTURE_3D
- GL_TEXTURE_DEPTH
- GL_TEXTURE_WRAP_R
- GL_MAX_3D_TEXTURE_SIZE
- GL_BGR
- GL_BGRA
- GL_MAX_ELEMENTS_VERTICES
- GL_MAX_ELEMENTS_INDICES
- GL_CLAMP_TO_EDGE
- GL_TEXTURE_MIN_LOD
- GL_TEXTURE_MAX_LOD
- GL_TEXTURE_BASE_LEVEL
- GL_TEXTURE_MAX_LEVEL
- GL_LIGHT_MODEL_COLOR_CONTROL
- GL_SINGLE_COLOR
- GL_SEPARATE_SPECULAR_COLOR
- GL_UNSIGNED_BYTE_2_3_3_REV

- GL_UNSIGNED_SHORT_5_6_5
- GL_UNSIGNED_SHORT_5_6_5_REV
- GL_UNSIGNED_SHORT_4_4_4_4_REV
- GL_UNSIGNED_SHORT_1_5_5_5_REV
- GL_UNSIGNED_INT_8_8_8_8_REV
- GL_ALIASED_POINT_SIZE_RANGE
- GL_ALIASED_LINE_WIDTH_RANGE
- GL_MULTISAMPLE
- GL_SAMPLE_ALPHA_TO_COVERAGE
- GL_SAMPLE_ALPHA_TO_ONE
- GL_SAMPLE_COVERAGE
- GL_SAMPLE_BUFFERS
- GL_SAMPLES
- GL_SAMPLE_COVERAGE_VALUE
- GL_SAMPLE_COVERAGE_INVERT
- GL_CLAMP_TO_BORDER
- GL_TEXTURE0
- GL_TEXTURE1
- GL_TEXTURE2
- GL_TEXTURE3
- GL_TEXTURE4
- GL_TEXTURE5
- GL_TEXTURE6
- GL_TEXTURE7
- GL_TEXTURE8
- GL_TEXTURE9
- GL_TEXTURE10
- GL_TEXTURE11
- GL_TEXTURE12
- GL_TEXTURE13
- GL_TEXTURE14
- GL_TEXTURE15
- GL_TEXTURE16
- GL_TEXTURE17
- GL_TEXTURE18
- GL_TEXTURE19

- GL_TEXTURE20
- GL_TEXTURE21
- GL_TEXTURE22
- GL_TEXTURE23
- GL_TEXTURE24
- GL_TEXTURE25
- GL_TEXTURE26
- GL_TEXTURE27
- GL_TEXTURE28
- GL_TEXTURE29
- GL_TEXTURE30
- GL_TEXTURE31
- GL_ACTIVE_TEXTURE
- GL_CLIENT_ACTIVE_TEXTURE
- GL_MAX_TEXTURE_UNITS
- GL_TRANSPOSE_MODELVIEW_MATRIX
- GL_TRANSPOSE_PROJECTION_MATRIX
- GL_TRANSPOSE_TEXTURE_MATRIX
- GL_TRANSPOSE_COLOR_MATRIX
- GL_SUBTRACT
- GL_COMPRESSED_ALPHA
- GL_COMPRESSED_LUMINANCE
- GL_COMPRESSED_LUMINANCE_ALPHA
- GL_COMPRESSED_INTENSITY
- GL_COMPRESSED_RGB
- GL_COMPRESSED_RGBA
- GL_TEXTURE_COMPRESSION_HINT
- GL_NORMAL_MAP
- GL_REFLECTION_MAP
- GL_TEXTURE_CUBE_MAP
- GL_TEXTURE_BINDING_CUBE_MAP
- GL_TEXTURE_CUBE_MAP_POSITIVE_X
- GL_TEXTURE_CUBE_MAP_NEGATIVE_X
- GL_TEXTURE_CUBE_MAP_POSITIVE_Y
- GL_TEXTURE_CUBE_MAP_NEGATIVE_Y
- GL_TEXTURE_CUBE_MAP_POSITIVE_Z

- GL_TEXTURE_CUBE_MAP_NEGATIVE_Z
- GL_PROXY_TEXTURE_CUBE_MAP
- GL_MAX_CUBE_MAP_TEXTURE_SIZE
- GL_COMBINE
- GL_COMBINE_RGB
- GL_COMBINE_ALPHA
- GL_RGB_SCALE
- GL_ADD_SIGNED
- GL_INTERPOLATE
- GL_CONSTANT
- GL_PRIMARY_COLOR
- GL_PREVIOUS
- GL_SOURCE0_RGB
- GL_SOURCE1_RGB
- GL_SOURCE2_RGB
- GL_SOURCE0_ALPHA
- GL_SOURCE1_ALPHA
- GL_SOURCE2_ALPHA
- GL_OPERAND0_RGB
- GL_OPERAND1_RGB
- GL_OPERAND2_RGB
- GL_OPERAND0_ALPHA
- GL_OPERAND1_ALPHA
- GL_OPERAND2_ALPHA
- GL_TEXTURE_COMPRESSED_IMAGE_SIZE
- GL_TEXTURE_COMPRESSED
- GL_NUM_COMPRESSED_TEXTURE_FORMATS
- GL_COMPRESSED_TEXTURE_FORMATS
- GL_DOT3_RGB
- GL_DOT3_RGBA
- GL_MULTISAMPLE_BIT
- GL_BLEND_DST_RGB
- GL_BLEND_SRC_RGB
- GL_BLEND_DST_ALPHA
- GL_BLEND_SRC_ALPHA
- GL_POINT_SIZE_MIN

- GL_POINT_SIZE_MAX
- GL_POINT_FADE_THRESHOLD_SIZE
- GL_POINT_DISTANCE_ATTENUATION
- GL_GENERATE_MIPMAP
- GL_GENERATE_MIPMAP_HINT
- GL_DEPTH_COMPONENT16
- GL_DEPTH_COMPONENT24
- GL_DEPTH_COMPONENT32
- GL_MIRRORED_REPEAT
- GL_FOG_COORDINATE_SOURCE
- GL_FOG_COORDINATE
- GL_FRAGMENT_DEPTH
- GL_CURRENT_FOG_COORDINATE
- GL_FOG_COORDINATE_ARRAY_TYPE
- GL_FOG_COORDINATE_ARRAY_STRIDE
- GL_FOG_COORDINATE_ARRAY_POINTER
- GL_FOG_COORDINATE_ARRAY
- GL_COLOR_SUM
- GL_CURRENT_SECONDARY_COLOR
- GL_SECONDARY_COLOR_ARRAY_SIZE
- GL_SECONDARY_COLOR_ARRAY_TYPE
- GL_SECONDARY_COLOR_ARRAY_STRIDE
- GL_SECONDARY_COLOR_ARRAY_POINTER
- GL_SECONDARY_COLOR_ARRAY
- GL_MAX_TEXTURE_LOD_BIAS
- GL_TEXTURE_FILTER_CONTROL
- GL_TEXTURE_LOD_BIAS
- GL_INCR_WRAP
- GL_DECR_WRAP
- GL_TEXTURE_DEPTH_SIZE
- GL_DEPTH_TEXTURE_MODE
- GL_TEXTURE_COMPARE_MODE
- GL_TEXTURE_COMPARE_FUNC
- GL_COMPARE_R_TO_TEXTURE
- GL_CURRENT_FOG_COORD
- GL_FOG_COORD

- GL_FOG_COORD_ARRAY
- GL_FOG_COORD_ARRAY_BUFFER_BINDING
- GL_FOG_COORD_ARRAY_POINTER
- GL_FOG_COORD_ARRAY_STRIDE
- GL_FOG_COORD_ARRAY_TYPE
- GL_FOG_COORD_SRC
- GL_SRC0_ALPHA
- GL_SRC0_RGB
- GL_SRC1_ALPHA
- GL_SRC1_RGB
- GL_SRC2_ALPHA
- GL_SRC2_RGB
- GL_BUFFER_SIZE
- GL_BUFFER_USAGE
- GL_QUERY_COUNTER_BITS
- GL_CURRENT_QUERY
- GL_QUERY_RESULT
- GL_QUERY_RESULT_AVAILABLE
- GL_ARRAY_BUFFER
- GL_ELEMENT_ARRAY_BUFFER
- GL_ARRAY_BUFFER_BINDING
- GL_ELEMENT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ARRAY_BUFFER_BINDING
- GL_NORMAL_ARRAY_BUFFER_BINDING
- GL_COLOR_ARRAY_BUFFER_BINDING
- GL_INDEX_ARRAY_BUFFER_BINDING
- GL_TEXTURE_COORD_ARRAY_BUFFER_BINDING
- GL_EDGE_FLAG_ARRAY_BUFFER_BINDING
- GL_SECONDARY_COLOR_ARRAY_BUFFER_BINDING
- GL_FOG_COORDINATE_ARRAY_BUFFER_BINDING
- GL_WEIGHT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ATTRIB_ARRAY_BUFFER_BINDING
- GL_READ_ONLY
- GL_WRITE_ONLY
- GL_READ_WRITE
- GL_BUFFER_ACCESS

- GL_BUFFER_MAPPED
- GL_BUFFER_MAP_POINTER
- GL_STREAM_DRAW
- GL_STREAM_READ
- GL_STREAM_COPY
- GL_STATIC_DRAW
- GL_STATIC_READ
- GL_STATIC_COPY
- GL_DYNAMIC_DRAW
- GL_DYNAMIC_READ
- GL_DYNAMIC_COPY
- GL_SAMPLES_PASSED
- GL_BLEND_EQUATION_RGB
- GL_VERTEX_ATTRIB_ARRAY_ENABLED
- GL_VERTEX_ATTRIB_ARRAY_SIZE
- GL_VERTEX_ATTRIB_ARRAY_STRIDE
- GL_VERTEX_ATTRIB_ARRAY_TYPE
- GL_CURRENT_VERTEX_ATTRIB
- GL_VERTEX_PROGRAM_POINT_SIZE
- GL_VERTEX_PROGRAM_TWO_SIDE
- GL_VERTEX_ATTRIB_ARRAY_POINTER
- GL_STENCIL_BACK_FUNC
- GL_STENCIL_BACK_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_PASS
- GL_MAX_DRAW_BUFFERS
- GL_DRAW_BUFFER0
- GL_DRAW_BUFFER1
- GL_DRAW_BUFFER2
- GL_DRAW_BUFFER3
- GL_DRAW_BUFFER4
- GL_DRAW_BUFFER5
- GL_DRAW_BUFFER6
- GL_DRAW_BUFFER7
- GL_DRAW_BUFFER8
- GL_DRAW_BUFFER9

- GL_DRAW_BUFFER10
- GL_DRAW_BUFFER11
- GL_DRAW_BUFFER12
- GL_DRAW_BUFFER13
- GL_DRAW_BUFFER14
- GL_DRAW_BUFFER15
- GL_BLEND_EQUATION_ALPHA
- GL_POINT_SPRITE
- GL_COORD_REPLACE
- GL_MAX_VERTEX_ATTRIBS
- GL_VERTEX_ATTRIB_ARRAY_NORMALIZED
- GL_MAX_TEXTURE_COORDS
- GL_MAX_TEXTURE_IMAGE_UNITS
- GL_FRAGMENT_SHADER
- GL_VERTEX_SHADER
- GL_MAX_FRAGMENT_UNIFORM_COMPONENTS
- GL_MAX_VERTEX_UNIFORM_COMPONENTS
- GL_MAX_VARYING_FLOATS
- GL_MAX_VERTEX_TEXTURE_IMAGE_UNITS
- GL_MAX_COMBINED_TEXTURE_IMAGE_UNITS
- GL_SHADER_TYPE
- GL_FLOAT_VEC2
- GL_FLOAT_VEC3
- GL_FLOAT_VEC4
- GL_INT_VEC2
- GL_INT_VEC3
- GL_INT_VEC4
- GL_BOOL
- GL_BOOL_VEC2
- GL_BOOL_VEC3
- GL_BOOL_VEC4
- GL_FLOAT_MAT2
- GL_FLOAT_MAT3
- GL_FLOAT_MAT4
- GL_SAMPLER_1D
- GL_SAMPLER_2D

- GL_SAMPLER_3D
- GL_SAMPLER_CUBE
- GL_SAMPLER_1D_SHADOW
- GL_SAMPLER_2D_SHADOW
- GL_DELETE_STATUS
- GL_COMPILE_STATUS
- GL_LINK_STATUS
- GL_VALIDATE_STATUS
- GL_INFO_LOG_LENGTH
- GL_ATTACHED_SHADERS
- GL_ACTIVE_UNIFORMS
- GL_ACTIVE_UNIFORM_MAX_LENGTH
- GL_SHADER_SOURCE_LENGTH
- GL_ACTIVE_ATTRIBUTES
- GL_ACTIVE_ATTRIBUTE_MAX_LENGTH
- GL_FRAGMENT_SHADER_DERIVATIVE_HINT
- GL_SHADING_LANGUAGE_VERSION
- GL_CURRENT_PROGRAM
- GL_POINT_SPRITE_COORD_ORIGIN
- GL_LOWER_LEFT
- GL_UPPER_LEFT
- GL_STENCIL_BACK_REF
- GL_STENCIL_BACK_VALUE_MASK
- GL_STENCIL_BACK_WRITEMASK
- GL_CURRENT_RASTER_SECONDARY_COLOR
- GL_PIXEL_PACK_BUFFER
- GL_PIXEL_UNPACK_BUFFER
- GL_PIXEL_PACK_BUFFER_BINDING
- GL_PIXEL_UNPACK_BUFFER_BINDING
- GL_FLOAT_MAT2x3
- GL_FLOAT_MAT2x4
- GL_FLOAT_MAT3x2
- GL_FLOAT_MAT3x4
- GL_FLOAT_MAT4x2
- GL_FLOAT_MAT4x3
- GL_SRGB

- GL_SRGB8
- GL_SRGB_ALPHA
- GL_SRGB8_ALPHA8
- GL_SLUMINANCE_ALPHA
- GL_SLUMINANCE8_ALPHA8
- GL_SLUMINANCE
- GL_SLUMINANCE8
- GL_COMPRESSED_SRGB
- GL_COMPRESSED_SRGB_ALPHA
- GL_COMPRESSED_SLUMINANCE
- GL_COMPRESSED_SLUMINANCE_ALPHA
- GL_CLIP_DISTANCE0
- GL_CLIP_DISTANCE1
- GL_CLIP_DISTANCE2
- GL_CLIP_DISTANCE3
- GL_CLIP_DISTANCE4
- GL_CLIP_DISTANCE5
- GL_COMPARE_REF_TO_TEXTURE
- GL_MAX_CLIP_DISTANCES
- GL_MAX_VARYING_COMPONENTS
- GL_CONTEXT_FLAG_FORWARD_COMPATIBLE_BIT
- GL_MAJOR_VERSION
- GL_MINOR_VERSION
- GL_NUM_EXTENSIONS
- GL_CONTEXT_FLAGS
- GL_DEPTH_BUFFER
- GL_STENCIL_BUFFER
- GL_RGBA32F
- GL_RGB32F
- GL_RGBA16F
- GL_RGB16F
- GL_VERTEX_ATTRIB_ARRAY_INTEGER
- GL_MAX_ARRAY_TEXTURE_LAYERS
- GL_MIN_PROGRAM_TEXEL_OFFSET
- GL_MAX_PROGRAM_TEXEL_OFFSET
- GL_CLAMP_VERTEX_COLOR

- GL_CLAMP_FRAGMENT_COLOR
- GL_CLAMP_READ_COLOR
- GL_FIXED_ONLY
- GL_TEXTURE_RED_TYPE
- GL_TEXTURE_GREEN_TYPE
- GL_TEXTURE_BLUE_TYPE
- GL_TEXTURE_ALPHA_TYPE
- GL_TEXTURE_LUMINANCE_TYPE
- GL_TEXTURE_INTENSITY_TYPE
- GL_TEXTURE_DEPTH_TYPE
- GL_TEXTURE_1D_ARRAY
- GL_PROXY_TEXTURE_1D_ARRAY
- GL_TEXTURE_2D_ARRAY
- GL_PROXY_TEXTURE_2D_ARRAY
- GL_TEXTURE_BINDING_1D_ARRAY
- GL_TEXTURE_BINDING_2D_ARRAY
- GL_R11F_G11F_B10F
- GL_UNSIGNED_INT_10F_11F_11F_REV
- GL_RGB9_E5
- GL_UNSIGNED_INT_5_9_9_9_REV
- GL_TEXTURE_SHARED_SIZE
- GL_TRANSFORM_FEEDBACK_VARYING_MAX_LENGTH
- GL_TRANSFORM_FEEDBACK_BUFFER_MODE
- GL_MAX_TRANSFORM_FEEDBACK_SEPARATE_COMPONENTS
- GL_TRANSFORM_FEEDBACK_VARYINGS
- GL_TRANSFORM_FEEDBACK_BUFFER_START
- GL_TRANSFORM_FEEDBACK_BUFFER_SIZE
- GL_PRIMITIVES_GENERATED
- GL_TRANSFORM_FEEDBACK_PRIMITIVES_WRITTEN
- GL_RASTERIZER_DISCARD
- GL_MAX_TRANSFORM_FEEDBACK_INTERLEAVED_COMPONENTS
- GL_MAX_TRANSFORM_FEEDBACK_SEPARATE_ATTRIBS
- GL_INTERLEAVED_ATTRIBS
- GL_SEPARATE_ATTRIBS
- GL_TRANSFORM_FEEDBACK_BUFFER
- GL_TRANSFORM_FEEDBACK_BUFFER_BINDING

- GL_RGBA32UI
- GL_RGB32UI
- GL_RGBA16UI
- GL_RGB16UI
- GL_RGBA8UI
- GL_RGB8UI
- GL_RGBA32I
- GL_RGB32I
- GL_RGBA16I
- GL_RGB16I
- GL_RGBA8I
- GL_RGB8I
- GL_RED_INTEGER
- GL_GREEN_INTEGER
- GL_BLUE_INTEGER
- GL_ALPHA_INTEGER
- GL_RGB_INTEGER
- GL_RGBA_INTEGER
- GL_BGR_INTEGER
- GL_BGRA_INTEGER
- GL_SAMPLER_1D_ARRAY
- GL_SAMPLER_2D_ARRAY
- GL_SAMPLER_1D_ARRAY_SHADOW
- GL_SAMPLER_2D_ARRAY_SHADOW
- GL_SAMPLER_CUBE_SHADOW
- GL_UNSIGNED_INT_VEC2
- GL_UNSIGNED_INT_VEC3
- GL_UNSIGNED_INT_VEC4
- GL_INT_SAMPLER_1D
- GL_INT_SAMPLER_2D
- GL_INT_SAMPLER_3D
- GL_INT_SAMPLER_CUBE
- GL_INT_SAMPLER_1D_ARRAY
- GL_INT_SAMPLER_2D_ARRAY
- GL_UNSIGNED_INT_SAMPLER_1D
- GL_UNSIGNED_INT_SAMPLER_2D

- GL_UNSIGNED_INT_SAMPLER_3D
- GL_UNSIGNED_INT_SAMPLER_CUBE
- GL_UNSIGNED_INT_SAMPLER_1D_ARRAY
- GL_UNSIGNED_INT_SAMPLER_2D_ARRAY
- GL_QUERY_WAIT
- GL_QUERY_NO_WAIT
- GL_QUERY_BY_REGION_WAIT
- GL_QUERY_BY_REGION_NO_WAIT
- GL_TEXTURE_RECTANGLE
- GL_TEXTURE_BINDING_RECTANGLE
- GL_PROXY_TEXTURE_RECTANGLE
- GL_MAX_RECTANGLE_TEXTURE_SIZE
- GL_SAMPLER_2D_RECT
- GL_SAMPLER_2D_RECT_SHADOW
- GL_TEXTURE_BUFFER
- GL_MAX_TEXTURE_BUFFER_SIZE
- GL_TEXTURE_BINDING_BUFFER
- GL_TEXTURE_BUFFER_DATA_STORE_BINDING
- GL_TEXTURE_BUFFER_FORMAT
- GL_SAMPLER_BUFFER
- GL_INT_SAMPLER_2D_RECT
- GL_INT_SAMPLER_BUFFER
- GL_UNSIGNED_INT_SAMPLER_2D_RECT
- GL_UNSIGNED_INT_SAMPLER_BUFFER
- GL_RED_SNORM
- GL_RG_SNORM
- GL_RGB_SNORM
- GL_RGBA_SNORM
- GL_R8_SNORM
- GL_RG8_SNORM
- GL_RGB8_SNORM
- GL_RGBA8_SNORM
- GL_R16_SNORM
- GL_RG16_SNORM
- GL_RGB16_SNORM
- GL_RGBA16_SNORM

- GL_SIGNED_NORMALIZED
- GL_PRIMITIVE_RESTART
- GL_PRIMITIVE_RESTART_INDEX
- GL_BUFFER_ACCESS_FLAGS
- GL_BUFFER_MAP_LENGTH
- GL_BUFFER_MAP_OFFSET
- GL_CONTEXT_CORE_PROFILE_BIT
- GL_CONTEXT_COMPATIBILITY_PROFILE_BIT
- GL_LINES_ADJACENCY
- GL_LINE_STRIP_ADJACENCY
- GL_TRIANGLES_ADJACENCY
- GL_TRIANGLE_STRIP_ADJACENCY
- GL_PROGRAM_POINT_SIZE
- GL_GEOMETRY_VERTICES_OUT
- GL_GEOMETRY_INPUT_TYPE
- GL_GEOMETRY_OUTPUT_TYPE
- GL_MAX_GEOMETRY_TEXTURE_IMAGE_UNITS
- GL_FRAMEBUFFER_ATTACHMENT_LAYERED
- GL_FRAMEBUFFER_INCOMPLETE_LAYER_TARGETS
- GL_GEOMETRY_SHADER
- GL_MAX_GEOMETRY_UNIFORM_COMPONENTS
- GL_MAX_GEOMETRY_OUTPUT_VERTICES
- GL_MAX_GEOMETRY_TOTAL_OUTPUT_COMPONENTS
- GL_MAX_VERTEX_OUTPUT_COMPONENTS
- GL_MAX_GEOMETRY_INPUT_COMPONENTS
- GL_MAX_GEOMETRY_OUTPUT_COMPONENTS
- GL_MAX_FRAGMENT_INPUT_COMPONENTS
- GL_CONTEXT_PROFILE_MASK
- GL_VERTEX_ATTRIB_ARRAY_DIVISOR
- GL_RGB10_A2UI
- GL_SAMPLE_SHADING
- GL_MIN_SAMPLE_SHADING_VALUE
- GL_MIN_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_COMPONENTS
- GL_TEXTURE_CUBE_MAP_ARRAY

- GL_TEXTURE_BINDING_CUBE_MAP_ARRAY
- GL_PROXY_TEXTURE_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY_SHADOW
- GL_INT_SAMPLER_CUBE_MAP_ARRAY
- GL_UNSIGNED_INT_SAMPLER_CUBE_MAP_ARRAY
- GL_TRANSFORM_FEEDBACK_PAUSED
- GL_TRANSFORM_FEEDBACK_ACTIVE
- GL_COMPRESSED_RGBA_BPTC_UNORM
- GL_COMPRESSED_SRGB_ALPHA_BPTC_UNORM
- GL_COMPRESSED_RGB_BPTC_SIGNED_FLOAT
- GL_COMPRESSED_RGB_BPTC_UNSIGNED_FLOAT
- GL_COPY_READ_BUFFER_BINDING
- GL_COPY_WRITE_BUFFER_BINDING
- GL_NUM_SHADING_LANGUAGE_VERSIONS
- GL_VERTEX_ATTRIB_ARRAY_LONG
- GL_PRIMITIVE_RESTART_FOR_PATCHES_SUPPORTED
- GL_MAX_VERTEX_ATTRIB_STRIDE
- GL_TEXTURE_BUFFER_BINDING
- GL_CONTEXT_FLAG_ROBUST_ACCESS_BIT
- void glAccum(GLenum op, GLfloat value)
- void glActiveTexture(GLenum texture)
- void glAlphaFunc(GLenum func, GLclampf ref)
- GLboolean glAreTexturesResident(GLsizei n, const GLuint * textures, GLboolean * residences)
- void glArrayElement(GLint i)
- void glAttachShader(GLuint program, GLuint shader)
- void glBegin(GLenum mode)
- void glBeginQuery(GLenum target, GLuint id)
- void glBindAttribLocation(GLuint program, GLuint index, const GLchar *name)
- void glBindBuffer(GLenum target, GLuint buffer)
- void glBindTexture(GLenum target, GLuint texture)
- void glBitmap(GLsizei width, GLsizei height, GLfloat xorig, GLfloat yorig, GLfloat xmove, GLfloat ymove, const GLubyte * bitmap)
- void glBlendColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)
- void glBlendEquation(GLenum mode)
- void glBlendEquationSeparate(GLenum modeRGB, GLenum modeAlpha)

- void glBlendFunc(GLenum sfactor, GLenum dfactor)
- void glBlendFuncSeparate(GLenum srcRGB, GLenum dstRGB, GLenum srcAlpha, GLenum dstAlpha)
- void glBufferData(GLenum target, GLsizeiptr size, const GLvoid * data, GLenum usage)
- void glBufferSubData(GLenum target, GLintptr offset, GLsizeiptr size, const GLvoid * data)
- void glCallList(GLuint list)
- void glCallLists(GLsizei n, GLenum type, const GLvoid * lists)
- void glClear(GLbitfield mask)
- void glClearAccum(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glClearColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)
- void glClearDepth(GLclampd depth)
- void glClearIndex(GLfloat c)
- void glClearStencil(GLint s)
- void glClientActiveTexture(GLenum texture)
- void glClipPlane(GLenum plane, const GLdouble * equation)
- void glColor3b(GLbyte red, GLbyte green, GLbyte blue)
- void glColor3s(GLshort red, GLshort green, GLshort blue)
- void glColor3i(GLint red, GLint green, GLint blue)
- void glColor3f(GLfloat red, GLfloat green, GLfloat blue)
- void glColor3d(GLdouble red, GLdouble green, GLdouble blue)
- void glColor3ub(GLubyte red, GLubyte green, GLubyte blue)
- void glColor3us(GLushort red, GLushort green, GLushort blue)
- void glColor3ui(GLuint red, GLuint green, GLuint blue)
- void glColor4b(GLbyte red, GLbyte green, GLbyte blue, GLbyte alpha)
- void glColor4s(GLshort red, GLshort green, GLshort blue, GLshort alpha)
- void glColor4i(GLint red, GLint green, GLint blue, GLint alpha)
- void glColor4f(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glColor4d(GLdouble red, GLdouble green, GLdouble blue, GLdouble alpha)
- void glColor4ub(GLubyte red, GLubyte green, GLubyte blue, GLubyte alpha)
- void glColor4us(GLushort red, GLushort green, GLushort blue, GLushort alpha)
- void glColor4ui(GLuint red, GLuint green, GLuint blue, GLuint alpha)
- void glColor3bv(const GLbyte * v)
- void glColor3sv(const GLshort * v)
- void glColor3iv(const GLint * v)
- void glColor3fv(const GLfloat * v)
- void glColor3dv(const GLdouble * v)
- void glColor3ubv(const GLubyte * v)

- `void glColor3usv(const GLushort * v)`
- `void glColor3uiv(const GLuint * v)`
- `void glColor4bv(const GLbyte * v)`
- `void glColor4sv(const GLshort * v)`
- `void glColor4iv(const GLint * v)`
- `void glColor4fv(const GLfloat * v)`
- `void glColor4dv(const GLdouble * v)`
- `void glColor4ubv(const GLubyte * v)`
- `void glColor4usv(const GLushort * v)`
- `void glColor4uiv(const GLuint * v)`
- `void glColorMask(GLboolean red, GLboolean green, GLboolean blue, GLboolean alpha)`
- `void glColorMaterial(GLenum face, GLenum mode)`
- `void glColorPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)`
- `void glColorSubTable(GLenum target, GLsizei start, GLsizei count, GLenum format, GLenum type, const GLvoid * data)`
- `void glColorTable(GLenum target, GLenum internalformat, GLsizei width, GLenum format, GLenum type, const GLvoid * data)`
- `void glColorTableParameterfv(GLenum target, GLenum pname, const GLfloat * params)`
- `void glColorTableParameteriv(GLenum target, GLenum pname, const GLint * params)`
- `void glCompileShader(GLuint shader)`
- `void glCompressedTexImage1D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLint border, GLsizei imageSize, const GLvoid * data)`
- `void glCompressedTexImage2D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLint border, GLsizei imageSize, const GLvoid * data)`
- `void glCompressedTexImage3D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLsizei depth, GLint border, GLsizei imageSize, const GLvoid * data)`
- `void glCompressedTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLsizei imageSize, const GLvoid * data)`
- `void glCompressedTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLsizei imageSize, const GLvoid * data)`
- `void glCompressedTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLsizei imageSize, const GLvoid * data)`
- `void glConvolutionFilter1D(GLenum target, GLenum internalformat, GLsizei width, GLenum format, GLenum type, const GLvoid * data)`
- `void glConvolutionFilter2D(GLenum target, GLenum internalformat, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)`
- `void glConvolutionParameterf(GLenum target, GLenum pname, GLfloat params)`
- `void glConvolutionParameteri(GLenum target, GLenum pname, GLint params)`
- `void glConvolutionParameterfv(GLenum target, GLenum pname, const GLfloat * params)`

- void glConvolutionParameteriv(GLenum target, GLenum pname, const GLint * params)
- void glCopyColorSubTable(GLenum target, GLsizei start, GLint x, GLint y, GLsizei width)
- void glCopyColorTable(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width)
- void glCopyConvolutionFilter1D(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width)
- void glCopyConvolutionFilter2D(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width, GLsizei height)
- void glCopyPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum type)
- void glCopyTexImage1D(GLenum target, GLint level, GLenum internalformat, GLint x, GLint y, GLsizei width, GLint border)
- void glCopyTexImage2D(GLenum target, GLint level, GLenum internalformat, GLint x, GLint y, GLsizei width, GLsizei height, GLint border)
- void glCopyTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLint x, GLint y, GLsizei width)
- void glCopyTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint x, GLint y, GLsizei width, GLsizei height)
- void glCopyTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLint x, GLint y, GLsizei width, GLsizei height)
- GLuint glCreateProgram(void)
- GLuint glCreateShader(GLenum shaderType)
- void glCullFace(GLenum mode)
- void glDeleteBuffers(GLsizei n, const GLuint * buffers)
- void glDeleteLists(GLuint list, GLsizei range)
- void glDeleteProgram(GLuint program)
- void glDeleteQueries(GLsizei n, const GLuint * ids)
- void glDeleteShader(GLuint shader)
- void glDeleteTextures(GLsizei n, const GLuint * textures)
- void glDepthFunc(GLenum func)
- void glDepthMask(GLboolean flag)
- void glDepthRange(GLclampd nearVal, GLclampd farVal)
- void glDetachShader(GLuint program, GLuint shader)
- void glEnable(GLenum cap)
- void glEnableClientState(GLenum cap)
- void glEnableVertexAttribArray(GLuint index)
- void glDisableVertexAttribArray(GLuint index)
- void glDrawArrays(GLenum mode, GLint first, GLsizei count)
- void glDrawBuffer(GLenum mode)
- void glDrawBuffers(GLsizei n, const GLenum *bufs)
- void glDrawElements(GLenum mode, GLsizei count, GLenum type, const GLvoid * indices)
- void glDrawPixels(GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)

- void glDrawRangeElements(GLenum mode,GLuint start,GLuint end,GLsizei count,GLenum type,const GLvoid * indices)
- void glEdgeFlag(GLboolean flag)
- void glEdgeFlagPointer(GLsizei stride,const GLvoid * pointer)
- void glEnd(void)
- void glEndList(void)
- void glEndQuery(GLenum target)
- void glEvalCoord1f(GLfloat u)
- void glEvalCoord1d(GLdouble u)
- void glEvalCoord2f(GLfloat u,GLfloat v)
- void glEvalCoord2d(GLdouble u,GLdouble v)
- void glEvalMesh1(GLenum mode,GLint i1,GLint i2)
- void glEvalPoint1(GLint i)
- void glEvalPoint2(GLint i,GLint j)
- void glFeedbackBuffer(GLsizei size,GLenum type,GLfloat * buffer)
- void glFinish(void)
- void glFlush(void)
- void glFogf(GLenum pname,GLfloat param)
- void glFogi(GLenum pname,GLint param)
- void glFogfv(GLenum pname,const GLfloat * params)
- void glFogiv(GLenum pname,const GLint * params)
- void glFogCoordd(GLdouble coord)
- void glFogCoordf(GLfloat coord)
- void glFogCoorddv(GLdouble * coord)
- void glFogCoordfv(GLfloat * coord)
- void glFogCoordPointer(GLenum type,GLsizei stride,GLvoid * pointer)
- void glFrontFace(GLenum mode)
- void glFrustum(GLdouble left,GLdouble right,GLdouble bottom,GLdouble top,GLdouble nearVal,GLdouble farVal)
- void glGenBuffers(GLsizei n,GLuint * buffers)
- GLuint glGenLists(GLsizei range)
- void glGenQueries(GLsizei n,GLuint * ids)
- void glGenTextures(GLsizei n,GLuint * textures)
- void glGetBooleany(GLenum pname,GLboolean * params)
- void glGetDoublev(GLenum pname,GLdouble * params)
- void glGetFloatv(GLenum pname,GLfloat * params)
- void glGetIntegerv(GLenum pname,GLint * params)

- void glGetActiveAttrib(GLuint program,GLuint index,GLsizei bufSize,GLsizei *length,GLint *size,GLenum *type,GLchar *name)
- void glGetActiveUniform(GLuint program,GLuint index,GLsizei bufSize,GLsizei *length,GLint *size,GLenum *type,GLchar *name)
- void glGetAttachedShaders(GLuint program,GLsizei maxCount,GLsizei *count,GLuint *shaders)
- GLint glGetAttribLocation(GLuint program,const GLchar *name)
- void glGetBufferParameteriv(GLenum target,GLenum value,GLint * data)
- void glGetBufferPointerv(GLenum target,GLenum pname,GLvoid ** params)
- void glGetBufferSubData(GLenum target,GLintptr offset,GLsizei size,GLvoid * data)
- void glGetClipPlane(GLenum plane,GLdouble * equation)
- void glGetColorTable(GLenum target,GLenum format,GLenum type,GLvoid * table)
- void glGetColorTableParameterfv(GLenum target,GLenum pname,GLfloat * params)
- void glGetColorTableParameteriv(GLenum target,GLenum pname,GLint * params)
- void glGetCompressedTexImage(GLenum target,GLint lod,GLvoid * img)
- void glGetConvolutionFilter(GLenum target,GLenum format,GLenum type,GLvoid * image)
- void glGetConvolutionParameterfv(GLenum target,GLenum pname,GLfloat * params)
- void glGetConvolutionParameteriv(GLenum target,GLenum pname,GLint * params)
- GLenum glGetError(void)
- void glGetHistogram(GLenum target,GLboolean reset,GLenum format,GLenum type,GLvoid * values)
- void glGetHistogramParameterfv(GLenum target,GLenum pname,GLfloat * params)
- void glGetHistogramParameteriv(GLenum target,GLenum pname,GLint * params)
- void glGetLightfv(GLenum light,GLenum pname,GLfloat * params)
- void glGetLightiv(GLenum light,GLenum pname,GLint * params)
- void glGetMapdv(GLenum target,GLenum query,GLdouble * v)
- void glGetMapfv(GLenum target,GLenum query,GLfloat * v)
- void glGetMapiv(GLenum target,GLenum query,GLint * v)
- void glGetMaterialfv(GLenum face,GLenum pname,GLfloat * params)
- void glGetMaterialiv(GLenum face,GLenum pname,GLint * params)
- void glGetMinmax(GLenum target,GLboolean reset,GLenum format,GLenum types,GLvoid * values)
- void glGetMinmaxParameterfv(GLenum target,GLenum pname,GLfloat * params)
- void glGetMinmaxParameteriv(GLenum target,GLenum pname,GLint * params)
- void glGetPixelMapfv(GLenum map,GLfloat * data)
- void glGetPixelMapuiv(GLenum map,GLuint * data)
- void glGetPixelMapusv(GLenum map,GLushort * data)
- void glGetPointerv(GLenum pname,GLvoid ** params)
- void glGetPolygonStipple(GLubyte * pattern)
- void glGetProgramiv(GLuint program,GLenum pname,GLint *params)

- void glGetProgramInfoLog(GLuint program, GLsizei maxLength, GLsizei *length, GLchar *infoLog)
- void glGetQueryObjectiv(GLuint id, GLenum pname, GLint * params)
- void glGetQueryObjectuiv(GLuint id, GLenum pname, GLuint * params)
- void glGetQueryiv(GLenum target, GLenum pname, GLint * params)
- void glGetSeparableFilter(GLenum target, GLenum format, GLenum type, GLvoid * row, GLvoid * column, GLvoid * span)
- void glGetShaderiv(GLuint shader, GLenum pname, GLint *params)
- void glGetShaderInfoLog(GLuint shader, GLsizei maxLength, GLsizei *length, GLchar *infoLog)
- void glGetShaderSource(GLuint shader, GLsizei bufSize, GLsizei *length, GLchar *source)
- const GLubyte* glGetString(GLenum name)
- void glGetTexEnvfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetTexEnviv(GLenum target, GLenum pname, GLint * params)
- void glGetTexGendv(GLenum coord, GLenum pname, GLdouble * params)
- void glGetTexGenfv(GLenum coord, GLenum pname, GLfloat * params)
- void glGetTexGeniv(GLenum coord, GLenum pname, GLint * params)
- void glGetTexImage(GLenum target, GLint level, GLenum format, GLenum type, GLvoid * img)
- void glGetTexLevelParameterfv(GLenum target, GLint level, GLenum pname, GLfloat * params)
- void glGetTexLevelParameteriv(GLenum target, GLint level, GLenum pname, GLint * params)
- void glGetTexParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetTexParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetUniformfv(GLuint program, GLint location, GLfloat *params)
- void glGetUniformiv(GLuint program, GLint location, GLint *params)
- GLint glGetUniformLocation(GLuint program, const GLchar *name)
- void glGetVertexAttribdv(GLuint index, GLenum pname, GLdouble *params)
- void glGetVertexAttribfv(GLuint index, GLenum pname, GLfloat *params)
- void glGetVertexAttribiv(GLuint index, GLenum pname, GLint *params)
- void glGetVertexAttribPointerv(GLuint index, GLenum pname, GLvoid **pointer)
- void glHint(GLenum target, GLenum mode)
- void glHistogram(GLenum target, GLsizei width, GLenum internalformat, GLboolean sink)
- void glIndexs(GLshort c)
- void glIndexi(GLint c)
- void glIndexf(GLfloat c)
- void glIndexd(GLdouble c)
- void glIndexub(GLubyte c)
- void glIndexsv(const GLshort * c)
- void glIndexiv(const GLint * c)

- `void glIndexfv(const GLfloat * c)`
- `void glIndexdv(const GLdouble * c)`
- `void glIndexbv(const GLubyte * c)`
- `void glIndexMask(GLuint mask)`
- `void glIndexPointer(GLenum type, GLsizei stride, const GLvoid * pointer)`
- `void glInitNames(void)`
- `void glInterleavedArrays(GLenum format, GLsizei stride, const GLvoid * pointer)`
- `GLboolean glIsBuffer(GLuint buffer)`
- `GLboolean glIsEnabled(GLenum cap)`
- `GLboolean glIsList(GLuint list)`
- `GLboolean glIsProgram(GLuint program)`
- `GLboolean glIsQuery(GLuint id)`
- `GLboolean glIsShader(GLuint shader)`
- `GLboolean glIsTexture(GLuint texture)`
- `void glLightf(GLenum light, GLenum pname, GLfloat param)`
- `void glLighti(GLenum light, GLenum pname, GLint param)`
- `void glLightfv(GLenum light, GLenum pname, const GLfloat * params)`
- `void glLightiv(GLenum light, GLenum pname, const GLint * params)`
- `void glLightModelf(GLenum pname, GLfloat param)`
- `void glLightModeli(GLenum pname, GLint param)`
- `void glLightModelfv(GLenum pname, const GLfloat * params)`
- `void glLightModeliv(GLenum pname, const GLint * params)`
- `void glLineStipple(GLint factor, GLushort pattern)`
- `void glLineWidth(GLfloat width)`
- `void glLinkProgram(GLuint program)`
- `void glListBase(GLuint base)`
- `void glLoadIdentity(void)`
- `void glLoadMatrixd(const GLdouble * m)`
- `void glLoadMatrixf(const GLfloat * m)`
- `void glLoadName(GLuint name)`
- `void glLoadTransposeMatrixd(const GLdouble * m)`
- `void glLoadTransposeMatrixf(const GLfloat * m)`
- `void glLogicOp(GLenum opcode)`
- `void glMap1f(GLenum target, GLfloat u1, GLfloat u2, GLint stride, GLint order, const GLfloat * points)`
- `void glMap1d(GLenum target, GLdouble u1, GLdouble u2, GLint stride, GLint order, const GLdouble * points)`

- `void glMap2f(GLenum target, GLfloat u1, GLfloat u2, GLint ustride, GLint uorder, GLfloat v1, GLfloat v2, GLint vstride, GLint vorder, const GLfloat * points)`
- `void glMap2d(GLenum target, GLdouble u1, GLdouble u2, GLint ustride, GLint uorder, GLdouble v1, GLdouble v2, GLint vstride, GLint vorder, const GLdouble * points)`
- `void * glMapBuffer(GLenum target, GLenum access)`
- `void glMapGrid1d(GLint un, GLdouble u1, GLdouble u2)`
- `void glMapGrid1f(GLint un, GLfloat u1, GLfloat u2)`
- `void glMapGrid2d(GLint un, GLdouble u1, GLdouble u2, GLint vn, GLdouble v1, GLdouble v2)`
- `void glMapGrid2f(GLint un, GLfloat u1, GLfloat u2, GLint vn, GLfloat v1, GLfloat v2)`
- `void glMaterialf(GLenum face, GLenum pname, GLfloat param)`
- `void glMateriali(GLenum face, GLenum pname, GLint param)`
- `void glMatrixMode(GLenum mode)`
- `void glMinmax(GLenum target, GLenum internalformat, GLboolean sink)`
- `void glMultMatrixd(const GLdouble * m)`
- `void glMultMatrixf(const GLfloat * m)`
- `void glMultTransposeMatrixd(const GLdouble * m)`
- `void glMultTransposeMatrixf(const GLfloat * m)`
- `void glMultiDrawArrays(GLenum mode, GLint * first, GLsizei * count, GLsizei primcount)`
- `void glMultiDrawElements(GLenum mode, const GLsizei * count, GLenum type, const GLvoid ** indices, GLsizei primcount)`
- `void glMultiTexCoord1s(GLenum target, GLshort s)`
- `void glMultiTexCoord1i(GLenum target, GLint s)`
- `void glMultiTexCoord1f(GLenum target, GLfloat s)`
- `void glMultiTexCoord1d(GLenum target, GLdouble s)`
- `void glMultiTexCoord2s(GLenum target, GLshort s, GLshort t)`
- `void glMultiTexCoord2i(GLenum target, GLint s, GLint t)`
- `void glMultiTexCoord2f(GLenum target, GLfloat s, GLfloat t)`
- `void glMultiTexCoord2d(GLenum target, GLdouble s, GLdouble t)`
- `void glMultiTexCoord3s(GLenum target, GLshort s, GLshort t, GLshort r)`
- `void glMultiTexCoord3i(GLenum target, GLint s, GLint t, GLint r)`
- `void glMultiTexCoord3f(GLenum target, GLfloat s, GLfloat t, GLfloat r)`
- `void glMultiTexCoord3d(GLenum target, GLdouble s, GLdouble t, GLdouble r)`
- `void glMultiTexCoord4s(GLenum target, GLshort s, GLshort t, GLshort r, GLshort q)`
- `void glMultiTexCoord4i(GLenum target, GLint s, GLint t, GLint r, GLint q)`
- `void glMultiTexCoord4f(GLenum target, GLfloat s, GLfloat t, GLfloat r, GLfloat q)`
- `void glMultiTexCoord4d(GLenum target, GLdouble s, GLdouble t, GLdouble r, GLdouble q)`
- `void glMultiTexCoord1sv(GLenum target, const GLshort * v)`

- `void glMultiTexCoord1iv(GLenum target,const GLint * v)`
- `void glMultiTexCoord1fv(GLenum target,const GLfloat * v)`
- `void glMultiTexCoord1dv(GLenum target,const GLdouble * v)`
- `void glMultiTexCoord2sv(GLenum target,const GLshort * v)`
- `void glMultiTexCoord2iv(GLenum target,const GLint * v)`
- `void glMultiTexCoord2fv(GLenum target,const GLfloat * v)`
- `void glMultiTexCoord2dv(GLenum target,const GLdouble * v)`
- `void glMultiTexCoord3sv(GLenum target,const GLshort * v)`
- `void glMultiTexCoord3iv(GLenum target,const GLint * v)`
- `void glMultiTexCoord3fv(GLenum target,const GLfloat * v)`
- `void glMultiTexCoord3dv(GLenum target,const GLdouble * v)`
- `void glMultiTexCoord4sv(GLenum target,const GLshort * v)`
- `void glMultiTexCoord4iv(GLenum target,const GLint * v)`
- `void glMultiTexCoord4fv(GLenum target,const GLfloat * v)`
- `void glMultiTexCoord4dv(GLenum target,const GLdouble * v)`
- `void glNewList(GLuint list,GLenum mode)`
- `void glNormal3b(GLbyte nx,GLbyte ny,GLbyte nz)`
- `void glNormal3d(GLdouble nx,GLdouble ny,GLdouble nz)`
- `void glNormal3f(GLfloat nx,GLfloat ny,GLfloat nz)`
- `void glNormal3i(GLint nx,GLint ny,GLint nz)`
- `void glNormal3s(GLshort nx,GLshort ny,GLshort nz)`
- `void glNormal3bv(const GLbyte * v)`
- `void glNormal3dv(const GLdouble * v)`
- `void glNormal3fv(const GLfloat * v)`
- `void glNormal3iv(const GLint * v)`
- `void glNormal3sv(const GLshort * v)`
- `void glNormalPointer(GLenum type,GLsizei stride,const GLvoid * pointer)`
- `void glOrtho(GLdouble left,GLdouble right,GLdouble bottom,GLdouble top,GLdouble nearVal,GLdouble farVal)`
- `void glPassThrough(GLfloat token)`
- `void glPixelMapfv(GLenum map,GLsizei mapsize,const GLfloat * values)`
- `void glPixelMapuiv(GLenum map,GLsizei mapsize,const GLuint * values)`
- `void glPixelMapusv(GLenum map,GLsizei mapsize,const GLushort * values)`
- `void glPixelStoref(GLenum pname,GLfloat param)`
- `void glPixelStorei(GLenum pname,GLint param)`
- `void glPixelTransferf(GLenum pname,GLfloat param)`

- void glPixelTransferi(GLenum pname,GLint param)
- void glPixelZoom(GLfloat xfactor,GLfloat yfactor)
- void glPointParameterf(GLenum pname,GLfloat param)
- void glPointParameteri(GLenum pname,GLint param)
- void glPointSize(GLfloat size)
- void glPolygonMode(GLenum face,GLenum mode)
- void glPolygonOffset(GLfloat factor,GLfloat units)
- void glPolygonStipple(const GLubyte * pattern)
- void glPushAttrib(GLbitfield mask)
- void glPushClientAttrib(GLbitfield mask)
- void glPushMatrix(void)
- void glPushName(GLuint name)
- void glPrioritizeTextures(GLsizei n,const GLuint * textures,const GLclampf * priorities)
- void glPopMatrix(void)
- void glRasterPos2s(GLshort x,GLshort y)
- void glRasterPos2i(GLint x,GLint y)
- void glRasterPos2f(GLfloat x,GLfloat y)
- void glRasterPos2d(GLdouble x,GLdouble y)
- void glRasterPos3s(GLshort x,GLshort y,GLshort z)
- void glRasterPos3i(GLint x,GLint y,GLint z)
- void glRasterPos3f(GLfloat x,GLfloat y,GLfloat z)
- void glRasterPos3d(GLdouble x,GLdouble y,GLdouble z)
- void glRasterPos4s(GLshort x,GLshort y,GLshort z,GLshort w)
- void glRasterPos4i(GLint x,GLint y,GLint z,GLint w)
- void glRasterPos4f(GLfloat x,GLfloat y,GLfloat z,GLfloat w)
- void glRasterPos4d(GLdouble x,GLdouble y,GLdouble z,GLdouble w)
- void glReadBuffer(GLenum mode)
- void glReadPixels(GLint x,GLint y,GLsizei width,GLsizei height,GLenum format,GLenum type,GLvoid * data)
- void glRectd(GLdouble x1,GLdouble y1,GLdouble x2,GLdouble y2)
- void glRectf(GLfloat x1,GLfloat y1,GLfloat x2,GLfloat y2)
- void glRecti(GLint x1,GLint y1,GLint x2,GLint y2)
- void glRects(GLshort x1,GLshort y1,GLshort x2,GLshort y2)
- void glRectdv(const GLdouble * v1,const GLdouble * v2)
- void glRectfv(const GLfloat * v1,const GLfloat * v2)
- void glRectiv(const GLint * v1,const GLint * v2)
- void glRectsv(const GLshort * v1,const GLshort * v2)

- GLint glRenderMode(GLenum mode)
- void glResetHistogram(GLenum target)
- void glRotated(GLdouble angle, GLdouble x, GLdouble y, GLdouble z)
- void glRotatef(GLfloat angle, GLfloat x, GLfloat y, GLfloat z)
- void glSampleCoverage(GLclampf value, GLboolean invert)
- void glScaled(GLdouble x, GLdouble y, GLdouble z)
- void glScalef(GLfloat x, GLfloat y, GLfloat z)
- void glScissor(GLint x, GLint y, GLsizei width, GLsizei height)
- void glSecondaryColor3b(GLbyte red, GLbyte green, GLbyte blue)
- void glSecondaryColor3s(GLshort red, GLshort green, GLshort blue)
- void glSecondaryColor3i(GLint red, GLint green, GLint blue)
- void glSecondaryColor3f(GLfloat red, GLfloat green, GLfloat blue)
- void glSecondaryColor3d(GLdouble red, GLdouble green, GLdouble blue)
- void glSecondaryColor3ub(GLubyte red, GLubyte green, GLubyte blue)
- void glSecondaryColor3us(GLushort red, GLushort green, GLushort blue)
- void glSecondaryColor3ui(GLuint red, GLuint green, GLuint blue)
- void glSecondaryColor3bv(const GLbyte * v)
- void glSecondaryColor3sv(const GLshort * v)
- void glSecondaryColor3iv(const GLint * v)
- void glSecondaryColor3fv(const GLfloat * v)
- void glSecondaryColor3dv(const GLdouble * v)
- void glSecondaryColor3ubv(const GLubyte * v)
- void glSecondaryColor3usv(const GLushort * v)
- void glSecondaryColor3uiv(const GLuint * v)
- void glSecondaryColorPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)
- void glSelectBuffer(GLsizei size, GLuint * buffer)
- void glSeparableFilter2D(GLenum target, GLenum internalformat, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * row, const GLvoid * column)
- void glShadeModel(GLenum mode)
- void glShaderSource(GLuint shader, GLsizei count, const GLchar **string, const GLint *length)
- void glStencilFunc(GLenum func, GLint ref, GLuint mask)
- void glStencilFuncSeparate(GLenum face, GLenum func, GLint ref, GLuint mask)
- void glStencilMask(GLuint mask)
- void glStencilMaskSeparate(GLenum face, GLuint mask)
- void glStencilOp(GLenum sfail, GLenum dpfail, GLenum dppass)
- void glStencilOpSeparate(GLenum face, GLenum sfail, GLenum dpfail, GLenum dppass)

- `void glTexCoord1s(GLshort s)`
- `void glTexCoord1i(GLint s)`
- `void glTexCoord1f(GLfloat s)`
- `void glTexCoord1d(GLdouble s)`
- `void glTexCoord2s(GLshort s, GLshort t)`
- `void glTexCoord2i(GLint s, GLint t)`
- `void glTexCoord2f(GLfloat s, GLfloat t)`
- `void glTexCoord2d(GLdouble s, GLdouble t)`
- `void glTexCoord3s(GLshort s, GLshort t, GLshort r)`
- `void glTexCoord3i(GLint s, GLint t, GLint r)`
- `void glTexCoord3f(GLfloat s, GLfloat t, GLfloat r)`
- `void glTexCoord3d(GLdouble s, GLdouble t, GLdouble r)`
- `void glTexCoord4s(GLshort s, GLshort t, GLshort r, GLshort q)`
- `void glTexCoord4i(GLint s, GLint t, GLint r, GLint q)`
- `void glTexCoord4f(GLfloat s, GLfloat t, GLfloat r, GLfloat q)`
- `void glTexCoord4d(GLdouble s, GLdouble t, GLdouble r, GLdouble q)`
- `void glTexCoord1sv(const GLshort * v)`
- `void glTexCoord1iv(const GLint * v)`
- `void glTexCoord1fv(const GLfloat * v)`
- `void glTexCoord1dv(const GLdouble * v)`
- `void glTexCoord2sv(const GLshort * v)`
- `void glTexCoord2iv(const GLint * v)`
- `void glTexCoord2fv(const GLfloat * v)`
- `void glTexCoord2dv(const GLdouble * v)`
- `void glTexCoord3sv(const GLshort * v)`
- `void glTexCoord3iv(const GLint * v)`
- `void glTexCoord3fv(const GLfloat * v)`
- `void glTexCoord3dv(const GLdouble * v)`
- `void glTexCoord4sv(const GLshort * v)`
- `void glTexCoord4iv(const GLint * v)`
- `void glTexCoord4fv(const GLfloat * v)`
- `void glTexCoord4dv(const GLdouble * v)`
- `void glTexCoordPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)`
- `void glTexEnvf(GLenum target, GLenum pname, GLfloat param)`
- `void glTexEnvf(GLenum target, GLenum pname, GLint param)`
- `void glTexGeni(GLenum coord, GLenum pname, GLint param)`

- void glTexGenf(GLenum coord, GLenum pname, GLfloat param)
- void glTexGend(GLenum coord, GLenum pname, GLdouble param)
- void glTexGeniv(GLenum coord, GLenum pname, const GLint * params)
- void glTexGenfv(GLenum coord, GLenum pname, const GLfloat * params)
- void glTexGendv(GLenum coord, GLenum pname, const GLdouble * params)
- void glTexImage1D(GLenum target, GLint level, GLint internalFormat, GLsizei width, GLint border, GLenum format, GLenum type, const GLvoid * data)
- void glTexImage2D(GLenum target, GLint level, GLint internalFormat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, const GLvoid * data)
- void glTexImage3D(GLenum target, GLint level, GLint internalFormat, GLsizei width, GLsizei height, GLsizei depth, GLint border, GLenum format, GLenum type, const GLvoid * data)
- void glTexParameterf(GLenum target, GLenum pname, GLfloat param)
- void glTexParameteri(GLenum target, GLenum pname, GLint param)
- void glTexParameterfv(GLenum target, GLenum pname, const GLfloat * params)
- void glTexParameteriv(GLenum target, GLenum pname, const GLint * params)
- void glTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLenum type, const GLvoid * data)
- void glTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)
- void glTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLenum type, const GLvoid * data)
- void glTranslated(GLdouble x, GLdouble y, GLdouble z)
- void glTranslatef(GLfloat x, GLfloat y, GLfloat z)
- void glUniform1f(GLint location, GLfloat v0)
- void glUniform2f(GLint location, GLfloat v0, GLfloat v1)
- void glUniform3f(GLint location, GLfloat v0, GLfloat v1, GLfloat v2)
- void glUniform4f(GLint location, GLfloat v0, GLfloat v1, GLfloat v2, GLfloat v3)
- void glUniform1i(GLint location, GLint v0)
- void glUniform2i(GLint location, GLint v0, GLint v1)
- void glUniform3i(GLint location, GLint v0, GLint v1, GLint v2)
- void glUniform4i(GLint location, GLint v0, GLint v1, GLint v2, GLint v3)
- void glUniform1fv(GLint location, GLsizei count, const GLfloat *value)
- void glUniform2fv(GLint location, GLsizei count, const GLfloat *value)
- void glUniform3fv(GLint location, GLsizei count, const GLfloat *value)
- void glUniform4fv(GLint location, GLsizei count, const GLfloat *value)
- void glUniform1iv(GLint location, GLsizei count, const GLint *value)
- void glUniform2iv(GLint location, GLsizei count, const GLint *value)
- void glUniform3iv(GLint location, GLsizei count, const GLint *value)

- void glUniform4iv(GLint location,GLsizei count,const GLint *value)
- void glUniformMatrix2fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix3fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix4fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix2x3fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix3x2fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix2x4fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix4x2fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix3x4fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix4x3fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUseProgram(GLuint program)
- void glValidateProgram(GLuint program)
- void glVertex2s(GLshort x,GLshort y)
- void glVertex2i(GLint x,GLint y)
- void glVertex2f(GLfloat x,GLfloat y)
- void glVertex2d(GLdouble x,GLdouble y)
- void glVertex3s(GLshort x,GLshort y,GLshort z)
- void glVertex3i(GLint x,GLint y,GLint z)
- void glVertex3f(GLfloat x,GLfloat y,GLfloat z)
- void glVertex3d(GLdouble x,GLdouble y,GLdouble z)
- void glVertex4s(GLshort x,GLshort y,GLshort z,GLshort w)
- void glVertex4i(GLint x,GLint y,GLint z,GLint w)
- void glVertex4f(GLfloat x,GLfloat y,GLfloat z,GLfloat w)
- void glVertex4d(GLdouble x,GLdouble y,GLdouble z,GLdouble w)
- void glVertex2sv(const GLshort * v)
- void glVertex2iv(const GLint * v)
- void glVertex2fv(const GLfloat * v)
- void glVertex2dv(const GLdouble * v)
- void glVertex3sv(const GLshort * v)
- void glVertex3iv(const GLint * v)
- void glVertex3fv(const GLfloat * v)
- void glVertex3dv(const GLdouble * v)
- void glVertex4sv(const GLshort * v)
- void glVertex4iv(const GLint * v)
- void glVertex4fv(const GLfloat * v)
- void glVertex4dv(const GLdouble * v)

- `void glVertexAttrib1f(GLuint index, GLfloat v0)`
- `void glVertexAttrib1s(GLuint index, GLshort v0)`
- `void glVertexAttrib1d(GLuint index, GLdouble v0)`
- `void glVertexAttrib2f(GLuint index, GLfloat v0, GLfloat v1)`
- `void glVertexAttrib2s(GLuint index, GLshort v0, GLshort v1)`
- `void glVertexAttrib2d(GLuint index, GLdouble v0, GLdouble v1)`
- `void glVertexAttrib3f(GLuint index, GLfloat v0, GLfloat v1, GLfloat v2)`
- `void glVertexAttrib3s(GLuint index, GLshort v0, GLshort v1, GLshort v2)`
- `void glVertexAttrib3d(GLuint index, GLdouble v0, GLdouble v1, GLdouble v2)`
- `void glVertexAttrib4f(GLuint index, GLfloat v0, GLfloat v1, GLfloat v2, GLfloat v3)`
- `void glVertexAttrib4s(GLuint index, GLshort v0, GLshort v1, GLshort v2, GLshort v3)`
- `void glVertexAttrib4d(GLuint index, GLdouble v0, GLdouble v1, GLdouble v2, GLdouble v3)`
- `void glVertexAttrib4Nub(GLuint index, GLubyte v0, GLubyte v1, GLubyte v2, GLubyte v3)`
- `void glVertexAttrib1fv(GLuint index, const GLfloat *v)`
- `void glVertexAttrib1sv(GLuint index, const GLshort *v)`
- `void glVertexAttrib1dv(GLuint index, const GLdouble *v)`
- `void glVertexAttrib2fv(GLuint index, const GLfloat *v)`
- `void glVertexAttrib2sv(GLuint index, const GLshort *v)`
- `void glVertexAttrib2dv(GLuint index, const GLdouble *v)`
- `void glVertexAttrib3fv(GLuint index, const GLfloat *v)`
- `void glVertexAttrib3sv(GLuint index, const GLshort *v)`
- `void glVertexAttrib3dv(GLuint index, const GLdouble *v)`
- `void glVertexAttrib4fv(GLuint index, const GLfloat *v)`
- `void glVertexAttrib4sv(GLuint index, const GLshort *v)`
- `void glVertexAttrib4dv(GLuint index, const GLdouble *v)`
- `void glVertexAttrib4iv(GLuint index, const GLint *v)`
- `void glVertexAttrib4bv(GLuint index, const GLbyte *v)`
- `void glVertexAttrib4ubv(GLuint index, const GLubyte *v)`
- `void glVertexAttrib4usv(GLuint index, const GLushort *v)`
- `void glVertexAttrib4uiv(GLuint index, const GLuint *v)`
- `void glVertexAttribPointer(GLuint index, GLint size, GLenum type, GLboolean normalized, GLsizei stride, const GLvoid * pointer)`
- `void glVertexPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)`
- `void glViewport(GLint x, GLint y, GLsizei width, GLsizei height)`
- `void glWindowPos2s(GLshort x, GLshort y)`
- `void glWindowPos2i(GLint x, GLint y)`

- void glWindowPos2f(GLfloat x,GLfloat y)
- void glWindowPos2d(GLdouble x,GLdouble y)
- void glWindowPos3s(GLshort x,GLshort y,GLshort z)
- void glWindowPos3i(GLint x,GLint y,GLint z)
- void glWindowPos3f(GLfloat x,GLfloat y,GLfloat z)
- void glWindowPos3d(GLdouble x,GLdouble y,GLdouble z)
- void glWindowPos2sv(const GLshort * v)
- void glWindowPos2iv(const GLint * v)
- void glWindowPos2fv(const GLfloat * v)
- void glWindowPos2dv(const GLdouble * v)
- void glWindowPos3sv(const GLshort * v)
- void glWindowPos3iv(const GLint * v)
- void glWindowPos3fv(const GLfloat * v)
- void glWindowPos3dv(const GLdouble * v)
- void gluBeginCurve(GLUnurbs* nurb)
- void gluBeginPolygon(GLUtesselator* tess)
- void gluBeginSurface(GLUnurbs* nurb)
- void gluBeginTrim(GLUnurbs* nurb)
- void gluCylinder(GLUquadric* quad,GLdouble base,GLdouble top,GLdouble height,GLint slices,GLint stacks)
- void gluDeleteNurbsRenderer(GLUnurbs* nurb)
- void gluDeleteQuadric(GLUquadric* quad)
- void gluDeleteTess(GLUtesselator* tess)
- void gluDisk(GLUquadric* quad,GLdouble inner,GLdouble outer,GLint slices,GLint loops)
- void gluEndCurve(GLUnurbs* nurb)
- void gluEndPolygon(GLUtesselator* tess)
- void gluEndSurface(GLUnurbs* nurb)
- void gluEndTrim(GLUnurbs* nurb)
- const GLubyte * gluErrorString(GLenum error)
- void gluGetNurbsProperty(GLUnurbs* nurb,GLenum property,GLfloat* data)
- const GLubyte * gluGetString(GLenum name)
- void gluGetTessProperty(GLUtesselator* tess,GLenum which,GLdouble* data)
- void gluLoadSamplingMatrices(GLUnurbs* nurb,const GLfloat * model,const GLfloat * perspective,const GLint * view)
- void gluLookAt(GLdouble eyeX,GLdouble eyeY,GLdouble eyeZ,GLdouble centerX,GLdouble centerY,GLdouble centerZ,GLdouble upX,GLdouble upY,GLdouble upZ)
- GLUnurbs *gluNewNurbsRenderer(void)
- GLUquadric *gluNewQuadric(void)

- `GLUtesselator* gluNewTess(void)`
- `void gluNextContour(GLUtesselator* tess, GLenum type)`
- `void gluNurbsCurve(GLUnurbs* nurb, GLint knotCount, GLfloat * knots, GLint stride, GLfloat * control, GLint order, GLenum type)`
- `void gluNurbsProperty(GLUnurbs* nurb, GLenum property, GLfloat value)`
- `void gluNurbsSurface(GLUnurbs* nurb, GLint sKnotCount, GLfloat* sKnots, GLint tKnotCount, GLfloat* tKnots, GLint sStride, GLint tStride, GLfloat* control, GLint sOrder, GLint tOrder, GLenum type)`
- `void gluOrtho2D(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top)`
- `void gluPartialDisk(GLUquadric* quad, GLdouble inner, GLdouble outer, GLint slices, GLint loops, GLdouble start, GLdouble sweep)`
- `void gluPerspective(GLdouble fovy, GLdouble aspect, GLdouble zNear, GLdouble zFar)`
- `void gluPickMatrix(GLdouble x, GLdouble y, GLdouble delX, GLdouble delY, GLint * viewport)`
- `GLint gluProject(GLdouble objX, GLdouble objY, GLdouble objZ, const GLdouble * model, const GLdouble * proj, const GLint * view, GLdouble* winX, GLdouble* winY, GLdouble* winZ)`
- `void gluPwlCurve(GLUnurbs* nurb, GLint count, GLfloat* data, GLint stride, GLenum type)`
- `void gluQuadricDrawStyle(GLUquadric* quad, GLenum draw)`
- `void gluQuadricNormals(GLUquadric* quad, GLenum normal)`
- `void gluQuadricOrientation(GLUquadric* quad, GLenum orientation)`
- `void gluQuadricTexture(GLUquadric* quad, GLboolean texture)`
- `GLint gluScaleImage(GLenum format, GLsizei wIn, GLsizei hIn, GLenum typeIn, const void * dataIn, GLsizei wOut, GLsizei hOut, GLenum typeOut, GLvoid* dataOut)`
- `void gluSphere(GLUquadric* quad, GLdouble radius, GLint slices, GLint stacks)`
- `void gluTessBeginContour(GLUtesselator* tess)`
- `void gluTessBeginPolygon(GLUtesselator* tess, GLvoid* data)`
- `void gluTessEndContour(GLUtesselator* tess)`
- `void gluTessEndPolygon(GLUtesselator* tess)`
- `void gluTessNormal(GLUtesselator* tess, GLdouble valueX, GLdouble valueY, GLdouble valueZ)`
- `void gluTessProperty(GLUtesselator* tess, GLenum which, GLdouble data)`
- `void gluTessVertex(GLUtesselator* tess, GLdouble * location, GLvoid* data)`
- `GLint gluUnProject(GLdouble winX, GLdouble winY, GLdouble winZ, const GLdouble * model, const GLdouble * proj, const GLint * view, GLdouble* objX, GLdouble* objY, GLdouble* objZ)`
- `void glDisable(GLenum cap)`

RINGOPENGL (OPENGL 4.6) FUNCTIONS REFERENCE

- GL_ZERO
- GL_FALSE
- GL_LOGIC_OP
- GL_NONE
- GL_TEXTURE_COMPONENTS
- GL_NO_ERROR
- GL_POINTS
- GL_CURRENT_BIT
- GL_TRUE
- GL_ONE
- GL_CLIENT_PIXEL_STORE_BIT
- GL_LINES
- GL_LINE_LOOP
- GL_POINT_BIT
- GL_CLIENT_VERTEX_ARRAY_BIT
- GL_LINE_STRIP
- GL_LINE_BIT
- GL_TRIANGLES
- GL_TRIANGLE_STRIP
- GL_TRIANGLE_FAN
- GL_QUADS
- GL_QUAD_STRIP
- GL_POLYGON_BIT
- GL_POLYGON
- GL_POLYGON_STIPPLE_BIT
- GL_PIXEL_MODE_BIT
- GL_LIGHTING_BIT

- GL_FOG_BIT
- GL_DEPTH_BUFFER_BIT
- GL_ACCUM
- GL_LOAD
- GL_RETURN
- GL_MULT
- GL_ADD
- GL_NEVER
- GL_ACCUM_BUFFER_BIT
- GL_LESS
- GL_EQUAL
- GL_LEQUAL
- GL_GREATER
- GL_NOTEQUAL
- GL_GEQUAL
- GL_ALWAYS
- GL_SRC_COLOR
- GL_ONE_MINUS_SRC_COLOR
- GL_SRC_ALPHA
- GL_ONE_MINUS_SRC_ALPHA
- GL_DST_ALPHA
- GL_ONE_MINUS_DST_ALPHA
- GL_DST_COLOR
- GL_ONE_MINUS_DST_COLOR
- GL_SRC_ALPHA_SATURATE
- GL_STENCIL_BUFFER_BIT
- GL_FRONT_LEFT
- GL_FRONT_RIGHT
- GL_BACK_LEFT
- GL_BACK_RIGHT
- GL_FRONT
- GL_BACK
- GL_LEFT
- GL_RIGHT
- GL_FRONT_AND_BACK
- GL_AUX0

- GL_AUX1
- GL_AUX2
- GL_AUX3
- GL_INVALID_ENUM
- GL_INVALID_VALUE
- GL_INVALID_OPERATION
- GL_STACK_OVERFLOW
- GL_STACK_UNDERFLOW
- GL_OUT_OF_MEMORY
- GL_2D
- GL_3D
- GL_3D_COLOR
- GL_3D_COLOR_TEXTURE
- GL_4D_COLOR_TEXTURE
- GL_PASS_THROUGH_TOKEN
- GL_POINT_TOKEN
- GL_LINE_TOKEN
- GL_POLYGON_TOKEN
- GL_BITMAP_TOKEN
- GL_DRAW_PIXEL_TOKEN
- GL_COPY_PIXEL_TOKEN
- GL_LINE_RESET_TOKEN
- GL_EXP
- GL_VIEWPORT_BIT
- GL_EXP2
- GL_CW
- GL_CCW
- GL_COEFF
- GL_ORDER
- GL_DOMAIN
- GL_CURRENT_COLOR
- GL_CURRENT_INDEX
- GL_CURRENT_NORMAL
- GL_CURRENT_TEXTURE_COORDS
- GL_CURRENT_RASTER_COLOR
- GL_CURRENT_RASTER_INDEX

- GL_CURRENT_RASTER_TEXTURE_COORDS
- GL_CURRENT_RASTER_POSITION
- GL_CURRENT_RASTER_POSITION_VALID
- GL_CURRENT_RASTER_DISTANCE
- GL_POINT_SMOOTH
- GL_POINT_SIZE
- GL_POINT_SIZE_RANGE
- GL_POINT_SIZE_GRANULARITY
- GL_LINE_SMOOTH
- GL_LINE_WIDTH
- GL_LINE_WIDTH_RANGE
- GL_LINE_WIDTH_GRANULARITY
- GL_LINE_STIPPLE
- GL_LINE_STIPPLE_PATTERN
- GL_LINE_STIPPLE_REPEAT
- GL_LIST_MODE
- GL_MAX_LIST_NESTING
- GL_LIST_BASE
- GL_LIST_INDEX
- GL_POLYGON_MODE
- GL_POLYGON_SMOOTH
- GL_POLYGON_STIPPLE
- GL_EDGE_FLAG
- GL_CULL_FACE
- GL_CULL_FACE_MODE
- GL_FRONT_FACE
- GL_LIGHTING
- GL_LIGHT_MODEL_LOCAL_VIEWER
- GL_LIGHT_MODEL_TWO_SIDE
- GL_LIGHT_MODEL_AMBIENT
- GL_SHADE_MODEL
- GL_COLOR_MATERIAL_FACE
- GL_COLOR_MATERIAL_PARAMETER
- GL_COLOR_MATERIAL
- GL_FOG
- GL_FOG_INDEX

- GL_FOG_DENSITY
- GL_FOG_START
- GL_FOG_END
- GL_FOG_MODE
- GL_FOG_COLOR
- GL_DEPTH_RANGE
- GL_DEPTH_TEST
- GL_DEPTH_WRITEMASK
- GL_DEPTH_CLEAR_VALUE
- GL_DEPTH_FUNC
- GL_ACCUM_CLEAR_VALUE
- GL_STENCIL_TEST
- GL_STENCIL_CLEAR_VALUE
- GL_STENCIL_FUNC
- GL_STENCIL_VALUE_MASK
- GL_STENCIL_FAIL
- GL_STENCIL_PASS_DEPTH_FAIL
- GL_STENCIL_PASS_DEPTH_PASS
- GL_STENCIL_REF
- GL_STENCIL_WRITEMASK
- GL_MATRIX_MODE
- GL_NORMALIZE
- GL_VIEWPORT
- GL_MODELVIEW_STACK_DEPTH
- GL_PROJECTION_STACK_DEPTH
- GL_TEXTURE_STACK_DEPTH
- GL_MODELVIEW_MATRIX
- GL_PROJECTION_MATRIX
- GL_TEXTURE_MATRIX
- GL_ATTRIB_STACK_DEPTH
- GL_CLIENT_ATTRIB_STACK_DEPTH
- GL_ALPHA_TEST
- GL_ALPHA_TEST_FUNC
- GL_ALPHA_TEST_REF
- GL_DITHER
- GL_BLEND_DST

- GL_BLEND_SRC
- GL_BLEND
- GL_LOGIC_OP_MODE
- GL_INDEX_LOGIC_OP
- GL_COLOR_LOGIC_OP
- GL_AUX_BUFFERS
- GL_DRAW_BUFFER
- GL_READ_BUFFER
- GL_SCISSOR_BOX
- GL_SCISSOR_TEST
- GL_INDEX_CLEAR_VALUE
- GL_INDEX_WRITEMASK
- GL_COLOR_CLEAR_VALUE
- GL_COLOR_WRITEMASK
- GL_INDEX_MODE
- GL_RGBA_MODE
- GL_DOUBLEBUFFER
- GL_STEREO
- GL_RENDER_MODE
- GL_PERSPECTIVE_CORRECTION_HINT
- GL_POINT_SMOOTH_HINT
- GL_LINE_SMOOTH_HINT
- GL_POLYGON_SMOOTH_HINT
- GL_FOG_HINT
- GL_TEXTURE_GEN_S
- GL_TEXTURE_GEN_T
- GL_TEXTURE_GEN_R
- GL_TEXTURE_GEN_Q
- GL_PIXEL_MAP_I_TO_I
- GL_PIXEL_MAP_S_TO_S
- GL_PIXEL_MAP_I_TO_R
- GL_PIXEL_MAP_I_TO_G
- GL_PIXEL_MAP_I_TO_B
- GL_PIXEL_MAP_I_TO_A
- GL_PIXEL_MAP_R_TO_R
- GL_PIXEL_MAP_G_TO_G

- GL_PIXEL_MAP_B_TO_B
- GL_PIXEL_MAP_A_TO_A
- GL_PIXEL_MAP_I_TO_I_SIZE
- GL_PIXEL_MAP_S_TO_S_SIZE
- GL_PIXEL_MAP_I_TO_R_SIZE
- GL_PIXEL_MAP_I_TO_G_SIZE
- GL_PIXEL_MAP_I_TO_B_SIZE
- GL_PIXEL_MAP_I_TO_A_SIZE
- GL_PIXEL_MAP_R_TO_R_SIZE
- GL_PIXEL_MAP_G_TO_G_SIZE
- GL_PIXEL_MAP_B_TO_B_SIZE
- GL_PIXEL_MAP_A_TO_A_SIZE
- GL_UNPACK_SWAP_BYTES
- GL_UNPACK_LSB_FIRST
- GL_UNPACK_ROW_LENGTH
- GL_UNPACK_SKIP_ROWS
- GL_UNPACK_SKIP_PIXELS
- GL_UNPACK_ALIGNMENT
- GL_PACK_SWAP_BYTES
- GL_PACK_LSB_FIRST
- GL_PACK_ROW_LENGTH
- GL_PACK_SKIP_ROWS
- GL_PACK_SKIP_PIXELS
- GL_PACK_ALIGNMENT
- GL_MAP_COLOR
- GL_MAP_STENCIL
- GL_INDEX_SHIFT
- GL_INDEX_OFFSET
- GL_RED_SCALE
- GL_RED_BIAS
- GL_ZOOM_X
- GL_ZOOM_Y
- GL_GREEN_SCALE
- GL_GREEN_BIAS
- GL_BLUE_SCALE
- GL_BLUE_BIAS

- GL_ALPHA_SCALE
- GL_ALPHA_BIAS
- GL_DEPTH_SCALE
- GL_DEPTH_BIAS
- GL_MAX_EVAL_ORDER
- GL_MAX_LIGHTS
- GL_MAX_CLIP_PLANES
- GL_MAX_TEXTURE_SIZE
- GL_MAX_PIXEL_MAP_TABLE
- GL_MAX_ATTRIB_STACK_DEPTH
- GL_MAX_MODELVIEW_STACK_DEPTH
- GL_MAX_NAME_STACK_DEPTH
- GL_MAX_PROJECTION_STACK_DEPTH
- GL_MAX_TEXTURE_STACK_DEPTH
- GL_MAX_VIEWPORT_DIMS
- GL_MAX_CLIENT_ATTRIB_STACK_DEPTH
- GL_SUBPIXEL_BITS
- GL_INDEX_BITS
- GL_RED_BITS
- GL_GREEN_BITS
- GL_BLUE_BITS
- GL_ALPHA_BITS
- GL_DEPTH_BITS
- GL_STENCIL_BITS
- GL_ACCUM_RED_BITS
- GL_ACCUM_GREEN_BITS
- GL_ACCUM_BLUE_BITS
- GL_ACCUM_ALPHA_BITS
- GL_NAME_STACK_DEPTH
- GL_AUTO_NORMAL
- GL_MAP1_COLOR_4
- GL_MAP1_INDEX
- GL_MAP1_NORMAL
- GL_MAP1_TEXTURE_COORD_1
- GL_MAP1_TEXTURE_COORD_2
- GL_MAP1_TEXTURE_COORD_3

- GL_MAP1_TEXTURE_COORD_4
- GL_MAP1_VERTEX_3
- GL_MAP1_VERTEX_4
- GL_MAP2_COLOR_4
- GL_MAP2_INDEX
- GL_MAP2_NORMAL
- GL_MAP2_TEXTURE_COORD_1
- GL_MAP2_TEXTURE_COORD_2
- GL_MAP2_TEXTURE_COORD_3
- GL_MAP2_TEXTURE_COORD_4
- GL_MAP2_VERTEX_3
- GL_MAP2_VERTEX_4
- GL_MAP1_GRID_DOMAIN
- GL_MAP1_GRID_SEGMENTS
- GL_MAP2_GRID_DOMAIN
- GL_MAP2_GRID_SEGMENTS
- GL_TEXTURE_1D
- GL_TEXTURE_2D
- GL_FEEDBACK_BUFFER_POINTER
- GL_FEEDBACK_BUFFER_SIZE
- GL_FEEDBACK_BUFFER_TYPE
- GL_SELECTION_BUFFER_POINTER
- GL_SELECTION_BUFFER_SIZE
- GL_TEXTURE_WIDTH
- GL_TRANSFORM_BIT
- GL_TEXTURE_HEIGHT
- GL_TEXTURE_INTERNAL_FORMAT
- GL_TEXTURE_BORDER_COLOR
- GL_TEXTURE_BORDER
- GL_DONT_CARE
- GL_FASTEST
- GL_NICEST
- GL_AMBIENT
- GL_DIFFUSE
- GL_SPECULAR
- GL_POSITION

- GL_SPOT_DIRECTION
- GL_SPOT_EXPONENT
- GL_SPOT_CUTOFF
- GL_CONSTANT_ATTENUATION
- GL_LINEAR_ATTENUATION
- GL_QUADRATIC_ATTENUATION
- GL_COMPILE
- GL_COMPILE_AND_EXECUTE
- GL_BYTE
- GL_UNSIGNED_BYTE
- GL_SHORT
- GL_UNSIGNED_SHORT
- GL_INT
- GL_UNSIGNED_INT
- GL_FLOAT
- GL_2_BYTES
- GL_3_BYTES
- GL_4_BYTES
- GL_DOUBLE
- GL_CLEAR
- GL_AND
- GL_AND_REVERSE
- GL_COPY
- GL_AND_INVERTED
- GL_NOOP
- GL_XOR
- GL_OR
- GL_NOR
- GL_EQUIV
- GL_INVERT
- GL_OR_REVERSE
- GL_COPY_INVERTED
- GL_OR_INVERTED
- GL_NAND
- GL_SET
- GL_EMISSION

- GL_SHININESS
- GL_AMBIENT_AND_DIFFUSE
- GL_COLOR_INDEXES
- GL_MODELVIEW
- GL_PROJECTION
- GL_TEXTURE
- GL_COLOR
- GL_DEPTH
- GL_STENCIL
- GL_COLOR_INDEX
- GL_STENCIL_INDEX
- GL_DEPTH_COMPONENT
- GL_RED
- GL_GREEN
- GL_BLUE
- GL_ALPHA
- GL_RGB
- GL_RGBA
- GL_LUMINANCE
- GL_LUMINANCE_ALPHA
- GL_BITMAP
- GL_POINT
- GL_LINE
- GL_FILL
- GL_RENDER
- GL_FEEDBACK
- GL_SELECT
- GL_FLAT
- GL_SMOOTH
- GL_KEEP
- GL_REPLACE
- GL_INCR
- GL_DECR
- GL_VENDOR
- GL_RENDERER
- GL_VERSION

- GL_EXTENSIONS
- GL_S
- GL_ENABLE_BIT
- GL_T
- GL_R
- GL_Q
- GL_MODULATE
- GL_DECAL
- GL_TEXTURE_ENV_MODE
- GL_TEXTURE_ENV_COLOR
- GL_TEXTURE_ENV
- GL_EYE_LINEAR
- GL_OBJECT_LINEAR
- GL_SPHERE_MAP
- GL_TEXTURE_GEN_MODE
- GL_OBJECT_PLANE
- GL_EYE_PLANE
- GL_NEAREST
- GL_LINEAR
- GL_NEAREST_MIPMAP_NEAREST
- GL_LINEAR_MIPMAP_NEAREST
- GL_NEAREST_MIPMAP_LINEAR
- GL_LINEAR_MIPMAP_LINEAR
- GL_TEXTURE_MAG_FILTER
- GL_TEXTURE_MIN_FILTER
- GL_TEXTURE_WRAP_S
- GL_TEXTURE_WRAP_T
- GL_CLAMP
- GL_REPEAT
- GL_POLYGON_OFFSET_UNITS
- GL_POLYGON_OFFSET_POINT
- GL_POLYGON_OFFSET_LINE
- GL_R3_G3_B2
- GL_V2F
- GL_V3F
- GL_C4UB_V2F

- GL_C4UB_V3F
- GL_C3F_V3F
- GL_N3F_V3F
- GL_C4F_N3F_V3F
- GL_T2F_V3F
- GL_T4F_V4F
- GL_T2F_C4UB_V3F
- GL_T2F_C3F_V3F
- GL_T2F_N3F_V3F
- GL_T2F_C4F_N3F_V3F
- GL_T4F_C4F_N3F_V4F
- GL_CLIP_PLANE0
- GL_CLIP_PLANE1
- GL_CLIP_PLANE2
- GL_CLIP_PLANE3
- GL_CLIP_PLANE4
- GL_CLIP_PLANE5
- GL_LIGHT0
- GL_COLOR_BUFFER_BIT
- GL_LIGHT1
- GL_LIGHT2
- GL_LIGHT3
- GL_LIGHT4
- GL_LIGHT5
- GL_LIGHT6
- GL_LIGHT7
- GL_HINT_BIT
- GL_POLYGON_OFFSET_FILL
- GL_POLYGON_OFFSET_FACTOR
- GL_ALPHA4
- GL_ALPHA8
- GL_ALPHA12
- GL_ALPHA16
- GL_LUMINANCE4
- GL_LUMINANCE8
- GL_LUMINANCE12

- GL_LUMINANCE16
- GL_LUMINANCE4_ALPHA4
- GL_LUMINANCE6_ALPHA2
- GL_LUMINANCE8_ALPHA8
- GL_LUMINANCE12_ALPHA4
- GL_LUMINANCE12_ALPHA12
- GL_LUMINANCE16_ALPHA16
- GL_INTENSITY
- GL_INTENSITY4
- GL_INTENSITY8
- GL_INTENSITY12
- GL_INTENSITY16
- GL_RGB4
- GL_RGB5
- GL_RGB8
- GL_RGB10
- GL_RGB12
- GL_RGB16
- GL_RGBA2
- GL_RGBA4
- GL_RGB5_A1
- GL_RGBA8
- GL_RGB10_A2
- GL_RGBA12
- GL_RGBA16
- GL_TEXTURE_RED_SIZE
- GL_TEXTURE_GREEN_SIZE
- GL_TEXTURE_BLUE_SIZE
- GL_TEXTURE_ALPHA_SIZE
- GL_TEXTURE_LUMINANCE_SIZE
- GL_TEXTURE_INTENSITY_SIZE
- GL_PROXY_TEXTURE_1D
- GL_PROXY_TEXTURE_2D
- GL_TEXTURE_PRIORITY
- GL_TEXTURE_RESIDENT
- GL_TEXTURE_BINDING_1D

- GL_TEXTURE_BINDING_2D
- GL_VERTEX_ARRAY
- GL_NORMAL_ARRAY
- GL_COLOR_ARRAY
- GL_INDEX_ARRAY
- GL_TEXTURE_COORD_ARRAY
- GL_EDGE_FLAG_ARRAY
- GL_VERTEX_ARRAY_SIZE
- GL_VERTEX_ARRAY_TYPE
- GL_VERTEX_ARRAY_STRIDE
- GL_NORMAL_ARRAY_TYPE
- GL_NORMAL_ARRAY_STRIDE
- GL_COLOR_ARRAY_SIZE
- GL_COLOR_ARRAY_TYPE
- GL_COLOR_ARRAY_STRIDE
- GL_INDEX_ARRAY_TYPE
- GL_INDEX_ARRAY_STRIDE
- GL_TEXTURE_COORD_ARRAY_SIZE
- GL_TEXTURE_COORD_ARRAY_TYPE
- GL_TEXTURE_COORD_ARRAY_STRIDE
- GL_EDGE_FLAG_ARRAY_STRIDE
- GL_VERTEX_ARRAY_POINTER
- GL_NORMAL_ARRAY_POINTER
- GL_COLOR_ARRAY_POINTER
- GL_INDEX_ARRAY_POINTER
- GL_TEXTURE_COORD_ARRAY_POINTER
- GL_EDGE_FLAG_ARRAY_POINTER
- GL_COLOR_INDEX1_EXT
- GL_COLOR_INDEX2_EXT
- GL_COLOR_INDEX4_EXT
- GL_COLOR_INDEX8_EXT
- GL_COLOR_INDEX12_EXT
- GL_COLOR_INDEX16_EXT
- GL_EVAL_BIT
- GL_LIST_BIT
- GL_TEXTURE_BIT

- GL_SCISSOR_BIT
- GL_ALL_ATTRIB_BITS
- GL_CLIENT_ALL_ATTRIB_BITS
- GL_SMOOTH_POINT_SIZE_RANGE
- GL_SMOOTH_POINT_SIZE_GRANULARITY
- GL_SMOOTH_LINE_WIDTH_RANGE
- GL_SMOOTH_LINE_WIDTH_GRANULARITY
- GL_UNSIGNED_BYTE_3_3_2
- GL_UNSIGNED_SHORT_4_4_4_4
- GL_UNSIGNED_SHORT_5_5_5_1
- GL_UNSIGNED_INT_8_8_8_8
- GL_UNSIGNED_INT_10_10_10_2
- GL_RESCALE_NORMAL
- GL_TEXTURE_BINDING_3D
- GL_PACK_SKIP_IMAGES
- GL_PACK_IMAGE_HEIGHT
- GL_UNPACK_SKIP_IMAGES
- GL_UNPACK_IMAGE_HEIGHT
- GL_TEXTURE_3D
- GL_PROXY_TEXTURE_3D
- GL_TEXTURE_DEPTH
- GL_TEXTURE_WRAP_R
- GL_MAX_3D_TEXTURE_SIZE
- GL_BGR
- GL_BGRA
- GL_MAX_ELEMENTS_VERTICES
- GL_MAX_ELEMENTS_INDICES
- GL_CLAMP_TO_EDGE
- GL_TEXTURE_MIN_LOD
- GL_TEXTURE_MAX_LOD
- GL_TEXTURE_BASE_LEVEL
- GL_TEXTURE_MAX_LEVEL
- GL_LIGHT_MODEL_COLOR_CONTROL
- GL_SINGLE_COLOR
- GL_SEPARATE_SPECULAR_COLOR
- GL_UNSIGNED_BYTE_2_3_3_REV

- GL_UNSIGNED_SHORT_5_6_5
- GL_UNSIGNED_SHORT_5_6_5_REV
- GL_UNSIGNED_SHORT_4_4_4_4_REV
- GL_UNSIGNED_SHORT_1_5_5_5_REV
- GL_UNSIGNED_INT_8_8_8_8_REV
- GL_ALIASED_POINT_SIZE_RANGE
- GL_ALIASED_LINE_WIDTH_RANGE
- GL_MULTISAMPLE
- GL_SAMPLE_ALPHA_TO_COVERAGE
- GL_SAMPLE_ALPHA_TO_ONE
- GL_SAMPLE_COVERAGE
- GL_SAMPLE_BUFFERS
- GL_SAMPLES
- GL_SAMPLE_COVERAGE_VALUE
- GL_SAMPLE_COVERAGE_INVERT
- GL_CLAMP_TO_BORDER
- GL_TEXTURE0
- GL_TEXTURE1
- GL_TEXTURE2
- GL_TEXTURE3
- GL_TEXTURE4
- GL_TEXTURE5
- GL_TEXTURE6
- GL_TEXTURE7
- GL_TEXTURE8
- GL_TEXTURE9
- GL_TEXTURE10
- GL_TEXTURE11
- GL_TEXTURE12
- GL_TEXTURE13
- GL_TEXTURE14
- GL_TEXTURE15
- GL_TEXTURE16
- GL_TEXTURE17
- GL_TEXTURE18
- GL_TEXTURE19

- GL_TEXTURE20
- GL_TEXTURE21
- GL_TEXTURE22
- GL_TEXTURE23
- GL_TEXTURE24
- GL_TEXTURE25
- GL_TEXTURE26
- GL_TEXTURE27
- GL_TEXTURE28
- GL_TEXTURE29
- GL_TEXTURE30
- GL_TEXTURE31
- GL_ACTIVE_TEXTURE
- GL_CLIENT_ACTIVE_TEXTURE
- GL_MAX_TEXTURE_UNITS
- GL_TRANSPOSE_MODELVIEW_MATRIX
- GL_TRANSPOSE_PROJECTION_MATRIX
- GL_TRANSPOSE_TEXTURE_MATRIX
- GL_TRANSPOSE_COLOR_MATRIX
- GL_SUBTRACT
- GL_COMPRESSED_ALPHA
- GL_COMPRESSED_LUMINANCE
- GL_COMPRESSED_LUMINANCE_ALPHA
- GL_COMPRESSED_INTENSITY
- GL_COMPRESSED_RGB
- GL_COMPRESSED_RGBA
- GL_TEXTURE_COMPRESSION_HINT
- GL_NORMAL_MAP
- GL_REFLECTION_MAP
- GL_TEXTURE_CUBE_MAP
- GL_TEXTURE_BINDING_CUBE_MAP
- GL_TEXTURE_CUBE_MAP_POSITIVE_X
- GL_TEXTURE_CUBE_MAP_NEGATIVE_X
- GL_TEXTURE_CUBE_MAP_POSITIVE_Y
- GL_TEXTURE_CUBE_MAP_NEGATIVE_Y
- GL_TEXTURE_CUBE_MAP_POSITIVE_Z

- GL_TEXTURE_CUBE_MAP_NEGATIVE_Z
- GL_PROXY_TEXTURE_CUBE_MAP
- GL_MAX_CUBE_MAP_TEXTURE_SIZE
- GL_COMBINE
- GL_COMBINE_RGB
- GL_COMBINE_ALPHA
- GL_RGB_SCALE
- GL_ADD_SIGNED
- GL_INTERPOLATE
- GL_CONSTANT
- GL_PRIMARY_COLOR
- GL_PREVIOUS
- GL_SOURCE0_RGB
- GL_SOURCE1_RGB
- GL_SOURCE2_RGB
- GL_SOURCE0_ALPHA
- GL_SOURCE1_ALPHA
- GL_SOURCE2_ALPHA
- GL_OPERAND0_RGB
- GL_OPERAND1_RGB
- GL_OPERAND2_RGB
- GL_OPERAND0_ALPHA
- GL_OPERAND1_ALPHA
- GL_OPERAND2_ALPHA
- GL_TEXTURE_COMPRESSED_IMAGE_SIZE
- GL_TEXTURE_COMPRESSED
- GL_NUM_COMPRESSED_TEXTURE_FORMATS
- GL_COMPRESSED_TEXTURE_FORMATS
- GL_DOT3_RGB
- GL_DOT3_RGBA
- GL_MULTISAMPLE_BIT
- GL_BLEND_DST_RGB
- GL_BLEND_SRC_RGB
- GL_BLEND_DST_ALPHA
- GL_BLEND_SRC_ALPHA
- GL_POINT_SIZE_MIN

- GL_POINT_SIZE_MAX
- GL_POINT_FADE_THRESHOLD_SIZE
- GL_POINT_DISTANCE_ATTENUATION
- GL_GENERATE_MIPMAP
- GL_GENERATE_MIPMAP_HINT
- GL_DEPTH_COMPONENT16
- GL_DEPTH_COMPONENT24
- GL_DEPTH_COMPONENT32
- GL_MIRRORED_REPEAT
- GL_FOG_COORDINATE_SOURCE
- GL_FOG_COORDINATE
- GL_FRAGMENT_DEPTH
- GL_CURRENT_FOG_COORDINATE
- GL_FOG_COORDINATE_ARRAY_TYPE
- GL_FOG_COORDINATE_ARRAY_STRIDE
- GL_FOG_COORDINATE_ARRAY_POINTER
- GL_FOG_COORDINATE_ARRAY
- GL_COLOR_SUM
- GL_CURRENT_SECONDARY_COLOR
- GL_SECONDARY_COLOR_ARRAY_SIZE
- GL_SECONDARY_COLOR_ARRAY_TYPE
- GL_SECONDARY_COLOR_ARRAY_STRIDE
- GL_SECONDARY_COLOR_ARRAY_POINTER
- GL_SECONDARY_COLOR_ARRAY
- GL_MAX_TEXTURE_LOD_BIAS
- GL_TEXTURE_FILTER_CONTROL
- GL_TEXTURE_LOD_BIAS
- GL_INCR_WRAP
- GL_DECR_WRAP
- GL_TEXTURE_DEPTH_SIZE
- GL_DEPTH_TEXTURE_MODE
- GL_TEXTURE_COMPARE_MODE
- GL_TEXTURE_COMPARE_FUNC
- GL_COMPARE_R_TO_TEXTURE
- GL_CURRENT_FOG_COORD
- GL_FOG_COORD

- GL_FOG_COORD_ARRAY
- GL_FOG_COORD_ARRAY_BUFFER_BINDING
- GL_FOG_COORD_ARRAY_POINTER
- GL_FOG_COORD_ARRAY_STRIDE
- GL_FOG_COORD_ARRAY_TYPE
- GL_FOG_COORD_SRC
- GL_SRC0_ALPHA
- GL_SRC0_RGB
- GL_SRC1_ALPHA
- GL_SRC1_RGB
- GL_SRC2_ALPHA
- GL_SRC2_RGB
- GL_BUFFER_SIZE
- GL_BUFFER_USAGE
- GL_QUERY_COUNTER_BITS
- GL_CURRENT_QUERY
- GL_QUERY_RESULT
- GL_QUERY_RESULT_AVAILABLE
- GL_ARRAY_BUFFER
- GL_ELEMENT_ARRAY_BUFFER
- GL_ARRAY_BUFFER_BINDING
- GL_ELEMENT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ARRAY_BUFFER_BINDING
- GL_NORMAL_ARRAY_BUFFER_BINDING
- GL_COLOR_ARRAY_BUFFER_BINDING
- GL_INDEX_ARRAY_BUFFER_BINDING
- GL_TEXTURE_COORD_ARRAY_BUFFER_BINDING
- GL_EDGE_FLAG_ARRAY_BUFFER_BINDING
- GL_SECONDARY_COLOR_ARRAY_BUFFER_BINDING
- GL_FOG_COORDINATE_ARRAY_BUFFER_BINDING
- GL_WEIGHT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ATTRIB_ARRAY_BUFFER_BINDING
- GL_READ_ONLY
- GL_WRITE_ONLY
- GL_READ_WRITE
- GL_BUFFER_ACCESS

- GL_BUFFER_MAPPED
- GL_BUFFER_MAP_POINTER
- GL_STREAM_DRAW
- GL_STREAM_READ
- GL_STREAM_COPY
- GL_STATIC_DRAW
- GL_STATIC_READ
- GL_STATIC_COPY
- GL_DYNAMIC_DRAW
- GL_DYNAMIC_READ
- GL_DYNAMIC_COPY
- GL_SAMPLES_PASSED
- GL_BLEND_EQUATION_RGB
- GL_VERTEX_ATTRIB_ARRAY_ENABLED
- GL_VERTEX_ATTRIB_ARRAY_SIZE
- GL_VERTEX_ATTRIB_ARRAY_STRIDE
- GL_VERTEX_ATTRIB_ARRAY_TYPE
- GL_CURRENT_VERTEX_ATTRIB
- GL_VERTEX_PROGRAM_POINT_SIZE
- GL_VERTEX_PROGRAM_TWO_SIDE
- GL_VERTEX_ATTRIB_ARRAY_POINTER
- GL_STENCIL_BACK_FUNC
- GL_STENCIL_BACK_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_PASS
- GL_MAX_DRAW_BUFFERS
- GL_DRAW_BUFFER0
- GL_DRAW_BUFFER1
- GL_DRAW_BUFFER2
- GL_DRAW_BUFFER3
- GL_DRAW_BUFFER4
- GL_DRAW_BUFFER5
- GL_DRAW_BUFFER6
- GL_DRAW_BUFFER7
- GL_DRAW_BUFFER8
- GL_DRAW_BUFFER9

- GL_DRAW_BUFFER10
- GL_DRAW_BUFFER11
- GL_DRAW_BUFFER12
- GL_DRAW_BUFFER13
- GL_DRAW_BUFFER14
- GL_DRAW_BUFFER15
- GL_BLEND_EQUATION_ALPHA
- GL_POINT_SPRITE
- GL_COORD_REPLACE
- GL_MAX_VERTEX_ATTRIBS
- GL_VERTEX_ATTRIB_ARRAY_NORMALIZED
- GL_MAX_TEXTURE_COORDS
- GL_MAX_TEXTURE_IMAGE_UNITS
- GL_FRAGMENT_SHADER
- GL_VERTEX_SHADER
- GL_MAX_FRAGMENT_UNIFORM_COMPONENTS
- GL_MAX_VERTEX_UNIFORM_COMPONENTS
- GL_MAX_VARYING_FLOATS
- GL_MAX_VERTEX_TEXTURE_IMAGE_UNITS
- GL_MAX_COMBINED_TEXTURE_IMAGE_UNITS
- GL_SHADER_TYPE
- GL_FLOAT_VEC2
- GL_FLOAT_VEC3
- GL_FLOAT_VEC4
- GL_INT_VEC2
- GL_INT_VEC3
- GL_INT_VEC4
- GL_BOOL
- GL_BOOL_VEC2
- GL_BOOL_VEC3
- GL_BOOL_VEC4
- GL_FLOAT_MAT2
- GL_FLOAT_MAT3
- GL_FLOAT_MAT4
- GL_SAMPLER_1D
- GL_SAMPLER_2D

- GL_SAMPLER_3D
- GL_SAMPLER_CUBE
- GL_SAMPLER_1D_SHADOW
- GL_SAMPLER_2D_SHADOW
- GL_DELETE_STATUS
- GL_COMPILE_STATUS
- GL_LINK_STATUS
- GL_VALIDATE_STATUS
- GL_INFO_LOG_LENGTH
- GL_ATTACHED_SHADERS
- GL_ACTIVE_UNIFORMS
- GL_ACTIVE_UNIFORM_MAX_LENGTH
- GL_SHADER_SOURCE_LENGTH
- GL_ACTIVE_ATTRIBUTES
- GL_ACTIVE_ATTRIBUTE_MAX_LENGTH
- GL_FRAGMENT_SHADER_DERIVATIVE_HINT
- GL_SHADING_LANGUAGE_VERSION
- GL_CURRENT_PROGRAM
- GL_POINT_SPRITE_COORD_ORIGIN
- GL_LOWER_LEFT
- GL_UPPER_LEFT
- GL_STENCIL_BACK_REF
- GL_STENCIL_BACK_VALUE_MASK
- GL_STENCIL_BACK_WRITEMASK
- GL_CURRENT_RASTER_SECONDARY_COLOR
- GL_PIXEL_PACK_BUFFER
- GL_PIXEL_UNPACK_BUFFER
- GL_PIXEL_PACK_BUFFER_BINDING
- GL_PIXEL_UNPACK_BUFFER_BINDING
- GL_FLOAT_MAT2x3
- GL_FLOAT_MAT2x4
- GL_FLOAT_MAT3x2
- GL_FLOAT_MAT3x4
- GL_FLOAT_MAT4x2
- GL_FLOAT_MAT4x3
- GL_SRGB

- GL_SRGB8
- GL_SRGB_ALPHA
- GL_SRGB8_ALPHA8
- GL_SLUMINANCE_ALPHA
- GL_SLUMINANCE8_ALPHA8
- GL_SLUMINANCE
- GL_SLUMINANCE8
- GL_COMPRESSED_SRGB
- GL_COMPRESSED_SRGB_ALPHA
- GL_COMPRESSED_SLUMINANCE
- GL_COMPRESSED_SLUMINANCE_ALPHA
- GL_CLIP_DISTANCE0
- GL_CLIP_DISTANCE1
- GL_CLIP_DISTANCE2
- GL_CLIP_DISTANCE3
- GL_CLIP_DISTANCE4
- GL_CLIP_DISTANCE5
- GL_COMPARE_REF_TO_TEXTURE
- GL_MAX_CLIP_DISTANCES
- GL_MAX_VARYING_COMPONENTS
- GL_CONTEXT_FLAG_FORWARD_COMPATIBLE_BIT
- GL_MAJOR_VERSION
- GL_MINOR_VERSION
- GL_NUM_EXTENSIONS
- GL_CONTEXT_FLAGS
- GL_DEPTH_BUFFER
- GL_STENCIL_BUFFER
- GL_RGBA32F
- GL_RGB32F
- GL_RGBA16F
- GL_RGB16F
- GL_VERTEX_ATTRIB_ARRAY_INTEGER
- GL_MAX_ARRAY_TEXTURE_LAYERS
- GL_MIN_PROGRAM_TEXEL_OFFSET
- GL_MAX_PROGRAM_TEXEL_OFFSET
- GL_CLAMP_VERTEX_COLOR

- GL_CLAMP_FRAGMENT_COLOR
- GL_CLAMP_READ_COLOR
- GL_FIXED_ONLY
- GL_TEXTURE_RED_TYPE
- GL_TEXTURE_GREEN_TYPE
- GL_TEXTURE_BLUE_TYPE
- GL_TEXTURE_ALPHA_TYPE
- GL_TEXTURE_LUMINANCE_TYPE
- GL_TEXTURE_INTENSITY_TYPE
- GL_TEXTURE_DEPTH_TYPE
- GL_TEXTURE_1D_ARRAY
- GL_PROXY_TEXTURE_1D_ARRAY
- GL_TEXTURE_2D_ARRAY
- GL_PROXY_TEXTURE_2D_ARRAY
- GL_TEXTURE_BINDING_1D_ARRAY
- GL_TEXTURE_BINDING_2D_ARRAY
- GL_R11F_G11F_B10F
- GL_UNSIGNED_INT_10F_11F_11F_REV
- GL_RGB9_E5
- GL_UNSIGNED_INT_5_9_9_9_REV
- GL_TEXTURE_SHARED_SIZE
- GL_TRANSFORM_FEEDBACK_VARYING_MAX_LENGTH
- GL_TRANSFORM_FEEDBACK_BUFFER_MODE
- GL_MAX_TRANSFORM_FEEDBACK_SEPARATE_COMPONENTS
- GL_TRANSFORM_FEEDBACK_VARYINGS
- GL_TRANSFORM_FEEDBACK_BUFFER_START
- GL_TRANSFORM_FEEDBACK_BUFFER_SIZE
- GL_PRIMITIVES_GENERATED
- GL_TRANSFORM_FEEDBACK_PRIMITIVES_WRITTEN
- GL_RASTERIZER_DISCARD
- GL_MAX_TRANSFORM_FEEDBACK_INTERLEAVED_COMPONENTS
- GL_MAX_TRANSFORM_FEEDBACK_SEPARATE_ATTRIBS
- GL_INTERLEAVED_ATTRIBS
- GL_SEPARATE_ATTRIBS
- GL_TRANSFORM_FEEDBACK_BUFFER
- GL_TRANSFORM_FEEDBACK_BUFFER_BINDING

- GL_RGBA32UI
- GL_RGB32UI
- GL_RGBA16UI
- GL_RGB16UI
- GL_RGBA8UI
- GL_RGB8UI
- GL_RGBA32I
- GL_RGB32I
- GL_RGBA16I
- GL_RGB16I
- GL_RGBA8I
- GL_RGB8I
- GL_RED_INTEGER
- GL_GREEN_INTEGER
- GL_BLUE_INTEGER
- GL_ALPHA_INTEGER
- GL_RGB_INTEGER
- GL_RGBA_INTEGER
- GL_BGR_INTEGER
- GL_BGRA_INTEGER
- GL_SAMPLER_1D_ARRAY
- GL_SAMPLER_2D_ARRAY
- GL_SAMPLER_1D_ARRAY_SHADOW
- GL_SAMPLER_2D_ARRAY_SHADOW
- GL_SAMPLER_CUBE_SHADOW
- GL_UNSIGNED_INT_VEC2
- GL_UNSIGNED_INT_VEC3
- GL_UNSIGNED_INT_VEC4
- GL_INT_SAMPLER_1D
- GL_INT_SAMPLER_2D
- GL_INT_SAMPLER_3D
- GL_INT_SAMPLER_CUBE
- GL_INT_SAMPLER_1D_ARRAY
- GL_INT_SAMPLER_2D_ARRAY
- GL_UNSIGNED_INT_SAMPLER_1D
- GL_UNSIGNED_INT_SAMPLER_2D

- GL_UNSIGNED_INT_SAMPLER_3D
- GL_UNSIGNED_INT_SAMPLER_CUBE
- GL_UNSIGNED_INT_SAMPLER_1D_ARRAY
- GL_UNSIGNED_INT_SAMPLER_2D_ARRAY
- GL_QUERY_WAIT
- GL_QUERY_NO_WAIT
- GL_QUERY_BY_REGION_WAIT
- GL_QUERY_BY_REGION_NO_WAIT
- GL_TEXTURE_RECTANGLE
- GL_TEXTURE_BINDING_RECTANGLE
- GL_PROXY_TEXTURE_RECTANGLE
- GL_MAX_RECTANGLE_TEXTURE_SIZE
- GL_SAMPLER_2D_RECT
- GL_SAMPLER_2D_RECT_SHADOW
- GL_TEXTURE_BUFFER
- GL_MAX_TEXTURE_BUFFER_SIZE
- GL_TEXTURE_BINDING_BUFFER
- GL_TEXTURE_BUFFER_DATA_STORE_BINDING
- GL_TEXTURE_BUFFER_FORMAT
- GL_SAMPLER_BUFFER
- GL_INT_SAMPLER_2D_RECT
- GL_INT_SAMPLER_BUFFER
- GL_UNSIGNED_INT_SAMPLER_2D_RECT
- GL_UNSIGNED_INT_SAMPLER_BUFFER
- GL_RED_SNORM
- GL_RG_SNORM
- GL_RGB_SNORM
- GL_RGBA_SNORM
- GL_R8_SNORM
- GL_RG8_SNORM
- GL_RGB8_SNORM
- GL_RGBA8_SNORM
- GL_R16_SNORM
- GL_RG16_SNORM
- GL_RGB16_SNORM
- GL_RGBA16_SNORM

- GL_SIGNED_NORMALIZED
- GL_PRIMITIVE_RESTART
- GL_PRIMITIVE_RESTART_INDEX
- GL_BUFFER_ACCESS_FLAGS
- GL_BUFFER_MAP_LENGTH
- GL_BUFFER_MAP_OFFSET
- GL_CONTEXT_CORE_PROFILE_BIT
- GL_CONTEXT_COMPATIBILITY_PROFILE_BIT
- GL_LINES_ADJACENCY
- GL_LINE_STRIP_ADJACENCY
- GL_TRIANGLES_ADJACENCY
- GL_TRIANGLE_STRIP_ADJACENCY
- GL_PROGRAM_POINT_SIZE
- GL_GEOMETRY_VERTICES_OUT
- GL_GEOMETRY_INPUT_TYPE
- GL_GEOMETRY_OUTPUT_TYPE
- GL_MAX_GEOMETRY_TEXTURE_IMAGE_UNITS
- GL_FRAMEBUFFER_ATTACHMENT_LAYERED
- GL_FRAMEBUFFER_INCOMPLETE_LAYER_TARGETS
- GL_GEOMETRY_SHADER
- GL_MAX_GEOMETRY_UNIFORM_COMPONENTS
- GL_MAX_GEOMETRY_OUTPUT_VERTICES
- GL_MAX_GEOMETRY_TOTAL_OUTPUT_COMPONENTS
- GL_MAX_VERTEX_OUTPUT_COMPONENTS
- GL_MAX_GEOMETRY_INPUT_COMPONENTS
- GL_MAX_GEOMETRY_OUTPUT_COMPONENTS
- GL_MAX_FRAGMENT_INPUT_COMPONENTS
- GL_CONTEXT_PROFILE_MASK
- GL_VERTEX_ATTRIB_ARRAY_DIVISOR
- GL_RGB10_A2UI
- GL_SAMPLE_SHADING
- GL_MIN_SAMPLE_SHADING_VALUE
- GL_MIN_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_COMPONENTS
- GL_TEXTURE_CUBE_MAP_ARRAY

- GL_TEXTURE_BINDING_CUBE_MAP_ARRAY
- GL_PROXY_TEXTURE_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY_SHADOW
- GL_INT_SAMPLER_CUBE_MAP_ARRAY
- GL_UNSIGNED_INT_SAMPLER_CUBE_MAP_ARRAY
- GL_TRANSFORM_FEEDBACK_PAUSED
- GL_TRANSFORM_FEEDBACK_ACTIVE
- GL_COMPRESSED_RGBA_BPTC_UNORM
- GL_COMPRESSED_SRGB_ALPHA_BPTC_UNORM
- GL_COMPRESSED_RGB_BPTC_SIGNED_FLOAT
- GL_COMPRESSED_RGB_BPTC_UNSIGNED_FLOAT
- GL_COPY_READ_BUFFER_BINDING
- GL_COPY_WRITE_BUFFER_BINDING
- GL_NUM_SHADING_LANGUAGE_VERSIONS
- GL_VERTEX_ATTRIB_ARRAY_LONG
- GL_PRIMITIVE_RESTART_FOR_PATCHES_SUPPORTED
- GL_MAX_VERTEX_ATTRIB_STRIDE
- GL_TEXTURE_BUFFER_BINDING
- GL_CONTEXT_FLAG_ROBUST_ACCESS_BIT
- GL_PARAMETER_BUFFER
- GL_PARAMETER_BUFFER_BINDING
- GL_TRANSFORM_FEEDBACK_OVERFLOW
- GL_TRANSFORM_FEEDBACK_STREAM_OVERFLOW
- GL_VERTICES_SUBMITTED
- GL_PRIMITIVES_SUBMITTED
- GL_VERTEX_SHADER_INVOCATIONS
- GL_TESS_CONTROL_SHADER_PATCHES
- GL_TESS_EVALUATION_SHADER_INVOCATIONS
- GL_GEOMETRY_SHADER_PRIMITIVES_EMITTED
- GL_FRAGMENT_SHADER_INVOCATIONS
- GL_COMPUTE_SHADER_INVOCATIONS
- GL_CLIPPING_INPUT_PRIMITIVES
- GL_CLIPPING_OUTPUT_PRIMITIVES
- GL_TEXTURE_MAX_ANISOTROPY
- GL_MAX_TEXTURE_MAX_ANISOTROPY

- GL_POLYGON_OFFSET_CLAMP
- GL_SHADER_BINARY_FORMAT_SPIR_V
- GL_SPIR_V_BINARY
- GL_SPIR_V_EXTENSIONS
- GL_NUM_SPIR_V_EXTENSIONS
- void glAccum(GLenum op, GLfloat value)
- void glActiveTexture(GLenum texture)
- void glAlphaFunc(GLenum func, GLclampf ref)
- GLboolean glAreTexturesResident(GLsizei n, const GLuint * textures, GLboolean * residences)
- void glArrayElement(GLint i)
- void glAttachShader(GLuint program, GLuint shader)
- void glBegin(GLenum mode)
- void glBeginQuery(GLenum target, GLuint id)
- void glBindAttribLocation(GLuint program, GLuint index, const GLchar * name)
- void glBindBuffer(GLenum target, GLuint buffer)
- void glBindTexture(GLenum target, GLuint texture)
- void glBitmap(GLsizei width, GLsizei height, GLfloat xorig, GLfloat yorig, GLfloat xmove, GLfloat ymove, const GLubyte * bitmap)
- void glBlendColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)
- void glBlendEquation(GLenum mode)
- void glBlendEquationSeparate(GLenum modeRGB, GLenum modeAlpha)
- void glBlendFunc(GLenum sfactor, GLenum dfactor)
- void glBlendFuncSeparate(GLenum srcRGB, GLenum dstRGB, GLenum srcAlpha, GLenum dstAlpha)
- void glBufferData(GLenum target, GLsizeiptr size, const GLvoid * data, GLenum usage)
- void glBufferSubData(GLenum target, GLintptr offset, GLsizeiptr size, const GLvoid * data)
- void glCallList(GLuint list)
- void glCallLists(GLsizei n, GLenum type, const GLvoid * lists)
- void glClear(GLbitfield mask)
- void glClearAccum(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glClearColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)
- void glClearDepth(GLclampd depth)
- void glClearIndex(GLfloat c)
- void glClearStencil(GLint s)
- void glClientActiveTexture(GLenum texture)
- void glClipPlane(GLenum plane, const GLdouble * equation)
- void glColor3b(GLbyte red, GLbyte green, GLbyte blue)

- `void glColor3s(GLshort red, GLshort green, GLshort blue)`
- `void glColor3i(GLint red, GLint green, GLint blue)`
- `void glColor3f(GLfloat red, GLfloat green, GLfloat blue)`
- `void glColor3d(GLdouble red, GLdouble green, GLdouble blue)`
- `void glColor3ub(GLubyte red, GLubyte green, GLubyte blue)`
- `void glColor3us(GLushort red, GLushort green, GLushort blue)`
- `void glColor3ui(GLuint red, GLuint green, GLuint blue)`
- `void glColor4b(GLbyte red, GLbyte green, GLbyte blue, GLbyte alpha)`
- `void glColor4s(GLshort red, GLshort green, GLshort blue, GLshort alpha)`
- `void glColor4i(GLint red, GLint green, GLint blue, GLint alpha)`
- `void glColor4f(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)`
- `void glColor4d(GLdouble red, GLdouble green, GLdouble blue, GLdouble alpha)`
- `void glColor4ub(GLubyte red, GLubyte green, GLubyte blue, GLubyte alpha)`
- `void glColor4us(GLushort red, GLushort green, GLushort blue, GLushort alpha)`
- `void glColor4ui(GLuint red, GLuint green, GLuint blue, GLuint alpha)`
- `void glColor3bv(const GLbyte * v)`
- `void glColor3sv(const GLshort * v)`
- `void glColor3iv(const GLint * v)`
- `void glColor3fv(const GLfloat * v)`
- `void glColor3dv(const GLdouble * v)`
- `void glColor3ubv(const GLubyte * v)`
- `void glColor3usv(const GLushort * v)`
- `void glColor3uiv(const GLuint * v)`
- `void glColor4bv(const GLbyte * v)`
- `void glColor4sv(const GLshort * v)`
- `void glColor4iv(const GLint * v)`
- `void glColor4fv(const GLfloat * v)`
- `void glColor4dv(const GLdouble * v)`
- `void glColor4ubv(const GLubyte * v)`
- `void glColor4usv(const GLushort * v)`
- `void glColor4uiv(const GLuint * v)`
- `void glColorMask(GLboolean red, GLboolean green, GLboolean blue, GLboolean alpha)`
- `void glColorMaterial(GLenum face, GLenum mode)`
- `void glColorPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)`
- `void glColorSubTable(GLenum target, GLsizei start, GLsizei count, GLenum format, GLenum type, const GLvoid * data)`

- void glColorTable(GLenum target, GLenum internalformat, GLsizei width, GLenum format, GLenum type, const GLvoid * data)
- void glColorTableParameterfv(GLenum target, GLenum pname, const GLfloat * params)
- void glColorTableParameteriv(GLenum target, GLenum pname, const GLint * params)
- void glCompileShader(GLuint shader)
- void glCompressedTexImage1D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLint border, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexImage2D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLint border, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexImage3D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLsizei depth, GLint border, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLsizei imageSize, const GLvoid * data)
- void glCompressedTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLsizei imageSize, const GLvoid * data)
- void glConvolutionFilter1D(GLenum target, GLenum internalformat, GLsizei width, GLenum format, GLenum type, const GLvoid * data)
- void glConvolutionFilter2D(GLenum target, GLenum internalformat, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)
- void glConvolutionParameterf(GLenum target, GLenum pname, GLfloat params)
- void glConvolutionParameteri(GLenum target, GLenum pname, GLint params)
- void glConvolutionParameterfv(GLenum target, GLenum pname, const GLfloat * params)
- void glConvolutionParameteriv(GLenum target, GLenum pname, const GLint * params)
- void glCopyColorSubTable(GLenum target, GLsizei start, GLint x, GLint y, GLsizei width)
- void glCopyColorTable(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width)
- void glCopyConvolutionFilter1D(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width)
- void glCopyConvolutionFilter2D(GLenum target, GLenum internalformat, GLint x, GLint y, GLsizei width, GLsizei height)
- void glCopyPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum type)
- void glCopyTexImage1D(GLenum target, GLint level, GLenum internalformat, GLint x, GLint y, GLsizei width, GLint border)
- void glCopyTexImage2D(GLenum target, GLint level, GLenum internalformat, GLint x, GLint y, GLsizei width, GLsizei height, GLint border)
- void glCopyTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLint x, GLint y, GLsizei width)
- void glCopyTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint x, GLint y, GLsizei width, GLsizei height)
- void glCopyTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLint x, GLint y, GLsizei width, GLsizei height)
- GLuint glCreateProgram(void)

- `GLuint glCreateShader(GLenum shaderType)`
- `void glCullFace(GLenum mode)`
- `void glDeleteBuffers(GLsizei n,const GLuint * buffers)`
- `void glDeleteLists(GLuint list,GLsizei range)`
- `void glDeleteProgram(GLuint program)`
- `void glDeleteQueries(GLsizei n,const GLuint * ids)`
- `void glDeleteShader(GLuint shader)`
- `void glDeleteTextures(GLsizei n,const GLuint * textures)`
- `void glDepthFunc(GLenum func)`
- `void glDepthMask(GLboolean flag)`
- `void glDepthRange(GLclampd nearVal,GLclampd farVal)`
- `void glDetachShader(GLuint program,GLuint shader)`
- `void glEnable(GLenum cap)`
- `void glEnableClientState(GLenum cap)`
- `void glEnableVertexAttribArray(GLuint index)`
- `void glDisableVertexAttribArray(GLuint index)`
- `void glDrawArrays(GLenum mode,GLint first,GLsizei count)`
- `void glDrawBuffer(GLenum mode)`
- `void glDrawBuffers(GLsizei n,const GLenum *bufs)`
- `void glDrawElements(GLenum mode,GLsizei count,GLenum type,const GLvoid * indices)`
- `void glDrawPixels(GLsizei width,GLsizei height,GLenum format,GLenum type,const GLvoid * data)`
- `void glDrawRangeElements(GLenum mode,GLuint start,GLuint end,GLsizei count,GLenum type,const GLvoid * indices)`
- `void glEdgeFlag(GLboolean flag)`
- `void glEdgeFlagPointer(GLsizei stride,const GLvoid * pointer)`
- `void glEnd(void)`
- `void glEndList(void)`
- `void glEndQuery(GLenum target)`
- `void glEvalCoord1f(GLfloat u)`
- `void glEvalCoord1d(GLdouble u)`
- `void glEvalCoord2f(GLfloat u,GLfloat v)`
- `void glEvalCoord2d(GLdouble u,GLdouble v)`
- `void glEvalMesh1(GLenum mode,GLint i1,GLint i2)`
- `void glEvalPoint1(GLint i)`
- `void glEvalPoint2(GLint i,GLint j)`
- `void glFeedbackBuffer(GLsizei size,GLenum type,GLfloat * buffer)`

- void glFinish(void)
- void glFlush(void)
- void glFogf(GLenum pname,GLfloat param)
- void glFogi(GLenum pname,GLint param)
- void glFogfv(GLenum pname,const GLfloat * params)
- void glFogiv(GLenum pname,const GLint * params)
- void glFogCoordd(GLdouble coord)
- void glFogCoordf(GLfloat coord)
- void glFogCoorddv(GLdouble * coord)
- void glFogCoordfv(GLfloat * coord)
- void glFogCoordPointer(GLenum type,GLsizei stride,GLvoid * pointer)
- void glFrontFace(GLenum mode)
- void glFrustum(GLdouble left,GLdouble right,GLdouble bottom,GLdouble top,GLdouble nearVal,GLdouble farVal)
- void glGenBuffers(GLsizei n,GLuint * buffers)
- GLuint glGenLists(GLsizei range)
- void glGenQueries(GLsizei n,GLuint * ids)
- void glGenTextures(GLsizei n,GLuint * textures)
- void glGetBooleany(GLenum pname,GLboolean * params)
- void glGetDoublev(GLenum pname,GLdouble * params)
- void glGetFloatv(GLenum pname,GLfloat * params)
- void glGetIntegerv(GLenum pname,GLint * params)
- void glGetActiveAttrib(GLuint program,GLuint index,GLsizei bufSize,GLsizei *length,GLint *size,GLenum *type,GLchar *name)
- void glGetActiveUniform(GLuint program,GLuint index,GLsizei bufSize,GLsizei *length,GLint *size,GLenum *type,GLchar *name)
- void glGetAttachedShaders(GLuint program,GLsizei maxCount,GLsizei *count,GLuint *shaders)
- GLint glGetAttribLocation(GLuint program,const GLchar *name)
- void glGetBufferParameteriv(GLenum target,GLenum value,GLint * data)
- void glGetBufferPointerv(GLenum target,GLenum pname,GLvoid ** params)
- void glGetBufferSubData(GLenum target,GLintptr offset,GLsizeiptr size,GLvoid * data)
- void glGetClipPlane(GLenum plane,GLdouble * equation)
- void glGetColorTable(GLenum target,GLenum format,GLenum type,GLvoid * table)
- void glGetColorTableParameterfv(GLenum target,GLenum pname,GLfloat * params)
- void glGetColorTableParameteriv(GLenum target,GLenum pname,GLint * params)
- void glGetCompressedTexImage(GLenum target,GLint lod,GLvoid * img)
- void glGetConvolutionFilter(GLenum target,GLenum format,GLenum type,GLvoid * image)

- void glGetConvolutionParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetConvolutionParameteriv(GLenum target, GLenum pname, GLint * params)
- GLenum glGetError(void)
- void glGetHistogram(GLenum target, GLboolean reset, GLenum format, GLenum type, GLvoid * values)
- void glGetHistogramParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetHistogramParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetLightfv(GLenum light, GLenum pname, GLfloat * params)
- void glGetLightiv(GLenum light, GLenum pname, GLint * params)
- void glGetMapdv(GLenum target, GLenum query, GLdouble * v)
- void glGetMapfv(GLenum target, GLenum query, GLfloat * v)
- void glGetMapiv(GLenum target, GLenum query, GLint * v)
- void glGetMaterialfv(GLenum face, GLenum pname, GLfloat * params)
- void glGetMaterialiv(GLenum face, GLenum pname, GLint * params)
- void glGetMinmax(GLenum target, GLboolean reset, GLenum format, GLenum types, GLvoid * values)
- void glGetMinmaxParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetMinmaxParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetPixelMapfv(GLenum map, GLfloat * data)
- void glGetPixelMapuiv(GLenum map, GLuint * data)
- void glGetPixelMapusv(GLenum map, GLushort * data)
- void glGetPointerv(GLenum pname, GLvoid ** params)
- void glGetPolygonStipple(GLubyte * pattern)
- void glGetProgramiv(GLuint program, GLenum pname, GLint *params)
- void glGetProgramInfoLog(GLuint program, GLsizei maxLength, GLsizei *length, GLchar *infoLog)
- void glGetQueryObjectiv(GLuint id, GLenum pname, GLint * params)
- void glGetQueryObjectuiv(GLuint id, GLenum pname, GLuint * params)
- void glGetQueryiv(GLenum target, GLenum pname, GLint * params)
- void glGetSeparableFilter(GLenum target, GLenum format, GLenum type, GLvoid * row, GLvoid * column, GLvoid * span)
- void glGetShaderiv(GLuint shader, GLenum pname, GLint *params)
- void glGetShaderInfoLog(GLuint shader, GLsizei maxLength, GLsizei *length, GLchar *infoLog)
- void glGetShaderSource(GLuint shader, GLsizei bufSize, GLsizei *length, GLchar *source)
- const GLubyte* getString(GLenum name)
- void glGetTexEnvfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetTexEnviv(GLenum target, GLenum pname, GLint * params)
- void glGetTexGendv(GLenum coord, GLenum pname, GLdouble * params)
- void glGetTexGenfv(GLenum coord, GLenum pname, GLfloat * params)

- void glGetTexGeniv(GLenum coord, GLenum pname, GLint * params)
- void glGetTexImage(GLenum target, GLint level, GLenum format, GLenum type, GLvoid * img)
- void glGetTexLevelParameterfv(GLenum target, GLint level, GLenum pname, GLfloat * params)
- void glGetTexLevelParameteriv(GLenum target, GLint level, GLenum pname, GLint * params)
- void glGetTexParameterfv(GLenum target, GLenum pname, GLfloat * params)
- void glGetTexParameteriv(GLenum target, GLenum pname, GLint * params)
- void glGetUniformfv(GLuint program, GLint location, GLfloat *params)
- void glGetUniformiv(GLuint program, GLint location, GLint *params)
- GLint glGetUniformLocation(GLuint program, const GLchar *name)
- void glGetVertexAttribdv(GLuint index, GLenum pname, GLdouble *params)
- void glGetVertexAttribfv(GLuint index, GLenum pname, GLfloat *params)
- void glGetVertexAttribiv(GLuint index, GLenum pname, GLint *params)
- void glGetVertexAttribPointerv(GLuint index, GLenum pname, GLvoid **pointer)
- void glHint(GLenum target, GLenum mode)
- void glHistogram(GLenum target, GLsizei width, GLenum internalformat, GLboolean sink)
- void glIndexs(GLshort c)
- void glIndexi(GLint c)
- void glIndexf(GLfloat c)
- void glIndexd(GLdouble c)
- void glIndexub(GLubyte c)
- void glIndexsv(const GLshort * c)
- void glIndexiv(const GLint * c)
- void glIndexfv(const GLfloat * c)
- void glIndexdv(const GLdouble * c)
- void glIndexubv(const GLubyte * c)
- void glIndexMask(GLuint mask)
- void glIndexPointer(GLenum type, GLsizei stride, const GLvoid * pointer)
- void glInitNames(void)
- void glInterleavedArrays(GLenum format, GLsizei stride, const GLvoid * pointer)
- GLboolean glIsBuffer(GLuint buffer)
- GLboolean glIsEnabled(GLenum cap)
- GLboolean glIsList(GLuint list)
- GLboolean glIsProgram(GLuint program)
- GLboolean glIsQuery(GLuint id)
- GLboolean glIsShader(GLuint shader)
- GLboolean glIsTexture(GLuint texture)

- void glLightf(GLenum light,GLenum pname,GLfloat param)
- void glLighti(GLenum light,GLenum pname,GLint param)
- void glLightfv(GLenum light,GLenum pname,const GLfloat * params)
- void glLightiv(GLenum light,GLenum pname,const GLint * params)
- void glLightModelf(GLenum pname,GLfloat param)
- void glLightModeli(GLenum pname,GLint param)
- void glLightModelfv(GLenum pname,const GLfloat * params)
- void glLightModeliv(GLenum pname,const GLint * params)
- void glLineStipple(GLint factor,GLushort pattern)
- void glLineWidth(GLfloat width)
- void glLinkProgram(GLuint program)
- void glListBase(GLuint base)
- void glLoadIdentity(void)
- void glLoadMatrixd(const GLdouble * m)
- void glLoadMatrixf(const GLfloat * m)
- void glLoadName(GLuint name)
- void glLoadTransposeMatrixd(const GLdouble * m)
- void glLoadTransposeMatrixf(const GLfloat * m)
- void glLogicOp(GLenum opcode)
- void glMap1f(GLenum target,GLfloat u1,GLfloat u2,GLint stride,GLint order,const GLfloat * points)
- void glMap1d(GLenum target,GLdouble u1,GLdouble u2,GLint stride,GLint order,const GLdouble * points)
- void glMap2f(GLenum target,GLfloat u1,GLfloat u2,GLint ustride,GLint uorder,GLfloat v1,GLfloat v2,GLint vstride,GLint vorder,const GLfloat * points)
- void glMap2d(GLenum target,GLdouble u1,GLdouble u2,GLint ustride,GLint uorder,GLdouble v1,GLdouble v2,GLint vstride,GLint vorder,const GLdouble * points)
- void * glMapBuffer(GLenum target,GLenum access)
- void glMapGrid1d(GLint un,GLdouble u1,GLdouble u2)
- void glMapGrid1f(GLint un,GLfloat u1,GLfloat u2)
- void glMapGrid2d(GLint un,GLdouble u1,GLdouble u2,GLint vn,GLdouble v1,GLdouble v2)
- void glMapGrid2f(GLint un,GLfloat u1,GLfloat u2,GLint vn,GLfloat v1,GLfloat v2)
- void glMaterialf(GLenum face,GLenum pname,GLfloat param)
- void glMateriali(GLenum face,GLenum pname,GLint param)
- void glMatrixMode(GLenum mode)
- void glMinmax(GLenum target,GLenum internalformat,GLboolean sink)
- void glMultMatrixd(const GLdouble * m)
- void glMultMatrixf(const GLfloat * m)
- void glMultTransposeMatrixd(const GLdouble * m)

- `void glMultTransposeMatrixf(const GLfloat * m)`
- `void glMultiDrawArrays(GLenum mode, GLint * first, GLsizei * count, GLsizei primcount)`
- `void glMultiDrawElements(GLenum mode, const GLsizei * count, GLenum type, const GLvoid ** indices, GLsizei primcount)`
- `void glMultiTexCoord1s(GLenum target, GLshort s)`
- `void glMultiTexCoord1i(GLenum target, GLint s)`
- `void glMultiTexCoord1f(GLenum target, GLfloat s)`
- `void glMultiTexCoord1d(GLenum target, GLdouble s)`
- `void glMultiTexCoord2s(GLenum target, GLshort s, GLshort t)`
- `void glMultiTexCoord2i(GLenum target, GLint s, GLint t)`
- `void glMultiTexCoord2f(GLenum target, GLfloat s, GLfloat t)`
- `void glMultiTexCoord2d(GLenum target, GLdouble s, GLdouble t)`
- `void glMultiTexCoord3s(GLenum target, GLshort s, GLshort t, GLshort r)`
- `void glMultiTexCoord3i(GLenum target, GLint s, GLint t, GLint r)`
- `void glMultiTexCoord3f(GLenum target, GLfloat s, GLfloat t, GLfloat r)`
- `void glMultiTexCoord3d(GLenum target, GLdouble s, GLdouble t, GLdouble r)`
- `void glMultiTexCoord4s(GLenum target, GLshort s, GLshort t, GLshort r, GLshort q)`
- `void glMultiTexCoord4i(GLenum target, GLint s, GLint t, GLint r, GLint q)`
- `void glMultiTexCoord4f(GLenum target, GLfloat s, GLfloat t, GLfloat r, GLfloat q)`
- `void glMultiTexCoord4d(GLenum target, GLdouble s, GLdouble t, GLdouble r, GLdouble q)`
- `void glMultiTexCoord1sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord1iv(GLenum target, const GLint * v)`
- `void glMultiTexCoord1fv(GLenum target, const GLfloat * v)`
- `void glMultiTexCoord1dv(GLenum target, const GLdouble * v)`
- `void glMultiTexCoord2sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord2iv(GLenum target, const GLint * v)`
- `void glMultiTexCoord2fv(GLenum target, const GLfloat * v)`
- `void glMultiTexCoord2dv(GLenum target, const GLdouble * v)`
- `void glMultiTexCoord3sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord3iv(GLenum target, const GLint * v)`
- `void glMultiTexCoord3fv(GLenum target, const GLfloat * v)`
- `void glMultiTexCoord3dv(GLenum target, const GLdouble * v)`
- `void glMultiTexCoord4sv(GLenum target, const GLshort * v)`
- `void glMultiTexCoord4iv(GLenum target, const GLint * v)`
- `void glMultiTexCoord4fv(GLenum target, const GLfloat * v)`
- `void glMultiTexCoord4dv(GLenum target, const GLdouble * v)`

- void glNewList(GLuint list, GLenum mode)
- void glNormal3b(GLbyte nx, GLbyte ny, GLbyte nz)
- void glNormal3d(GLdouble nx, GLdouble ny, GLdouble nz)
- void glNormal3f(GLfloat nx, GLfloat ny, GLfloat nz)
- void glNormal3i(GLint nx, GLint ny, GLint nz)
- void glNormal3s(GLshort nx, GLshort ny, GLshort nz)
- void glNormal3bv(const GLbyte * v)
- void glNormal3dv(const GLdouble * v)
- void glNormal3fv(const GLfloat * v)
- void glNormal3iv(const GLint * v)
- void glNormal3sv(const GLshort * v)
- void glNormalPointer(GLenum type, GLsizei stride, const GLvoid * pointer)
- void glOrtho(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble nearVal, GLdouble farVal)
- void glPassThrough(GLfloat token)
- void glPixelMapfv(GLenum map, GLsizei mapsize, const GLfloat * values)
- void glPixelMapuiv(GLenum map, GLsizei mapsize, const GLuint * values)
- void glPixelMapusv(GLenum map, GLsizei mapsize, const GLushort * values)
- void glPixelStoref(GLenum pname, GLfloat param)
- void glPixelStorei(GLenum pname, GLint param)
- void glPixelTransferf(GLenum pname, GLfloat param)
- void glPixelTransferi(GLenum pname, GLint param)
- void glPixelZoom(GLfloat xfactor, GLfloat yfactor)
- void glPointParameterf(GLenum pname, GLfloat param)
- void glPointParameteri(GLenum pname, GLint param)
- void glPointSize(GLfloat size)
- void glPolygonMode(GLenum face, GLenum mode)
- void glPolygonOffset(GLfloat factor, GLfloat units)
- void glPolygonStipple(const GLubyte * pattern)
- void glPushAttrib(GLbitfield mask)
- void glPushClientAttrib(GLbitfield mask)
- void glPushMatrix(void)
- void glPushName(GLuint name)
- void glPrioritizeTextures(GLsizei n, const GLuint * textures, const GLclampf * priorities)
- void glPopMatrix(void)
- void glRasterPos2s(GLshort x, GLshort y)

- `void glRasterPos2i(GLint x, GLint y)`
- `void glRasterPos2f(GLfloat x, GLfloat y)`
- `void glRasterPos2d(GLdouble x, GLdouble y)`
- `void glRasterPos3s(GLshort x, GLshort y, GLshort z)`
- `void glRasterPos3i(GLint x, GLint y, GLint z)`
- `void glRasterPos3f(GLfloat x, GLfloat y, GLfloat z)`
- `void glRasterPos3d(GLdouble x, GLdouble y, GLdouble z)`
- `void glRasterPos4s(GLshort x, GLshort y, GLshort z, GLshort w)`
- `void glRasterPos4i(GLint x, GLint y, GLint z, GLint w)`
- `void glRasterPos4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)`
- `void glRasterPos4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)`
- `void glReadBuffer(GLenum mode)`
- `void glReadPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum format, GLenum type, GLvoid * data)`
- `void glRectd(GLdouble x1, GLdouble y1, GLdouble x2, GLdouble y2)`
- `void glRectf(GLfloat x1, GLfloat y1, GLfloat x2, GLfloat y2)`
- `void glRecti(GLint x1, GLint y1, GLint x2, GLint y2)`
- `void glRects(GLshort x1, GLshort y1, GLshort x2, GLshort y2)`
- `void glRectdv(const GLdouble * v1, const GLdouble * v2)`
- `void glRectfv(const GLfloat * v1, const GLfloat * v2)`
- `void glRectiv(const GLint * v1, const GLint * v2)`
- `void glRectsv(const GLshort * v1, const GLshort * v2)`
- `GLint glRenderMode(GLenum mode)`
- `void glResetHistogram(GLenum target)`
- `void glRotated(GLdouble angle, GLdouble x, GLdouble y, GLdouble z)`
- `void glRotatef(GLfloat angle, GLfloat x, GLfloat y, GLfloat z)`
- `void glSampleCoverage(GLclampf value, GLboolean invert)`
- `void glScaled(GLdouble x, GLdouble y, GLdouble z)`
- `void glScalef(GLfloat x, GLfloat y, GLfloat z)`
- `void glScissor(GLint x, GLint y, GLsizei width, GLsizei height)`
- `void glSecondaryColor3b(GLbyte red, GLbyte green, GLbyte blue)`
- `void glSecondaryColor3s(GLshort red, GLshort green, GLshort blue)`
- `void glSecondaryColor3i(GLint red, GLint green, GLint blue)`
- `void glSecondaryColor3f(GLfloat red, GLfloat green, GLfloat blue)`
- `void glSecondaryColor3d(GLdouble red, GLdouble green, GLdouble blue)`
- `void glSecondaryColor3ub(GLubyte red, GLubyte green, GLubyte blue)`
- `void glSecondaryColor3us(GLushort red, GLushort green, GLushort blue)`

- void glSecondaryColor3ui(GLuint red,GLuint green,GLuint blue)
- void glSecondaryColor3bv(const GLbyte * v)
- void glSecondaryColor3sv(const GLshort * v)
- void glSecondaryColor3iv(const GLint * v)
- void glSecondaryColor3fv(const GLfloat * v)
- void glSecondaryColor3dv(const GLdouble * v)
- void glSecondaryColor3ubv(const GLubyte * v)
- void glSecondaryColor3usv(const GLushort * v)
- void glSecondaryColor3uiv(const GLuint * v)
- void glSecondaryColorPointer(GLint size,GLenum type,GLsizei stride,const GLvoid * pointer)
- void glSelectBuffer(GLsizei size,GLuint * buffer)
- void glSeparableFilter2D(GLenum target,GLenum internalformat,GLsizei width,GLsizei height,GLenum format,GLenum type,const GLvoid * row,const GLvoid * column)
- void glShadeModel(GLenum mode)
- void glShaderSource(GLuint shader,GLsizei count,const GLchar **string,const GLint *length)
- void glStencilFunc(GLenum func,GLint ref,GLuint mask)
- void glStencilFuncSeparate(GLenum face,GLenum func,GLint ref,GLuint mask)
- void glStencilMask(GLuint mask)
- void glStencilMaskSeparate(GLenum face,GLuint mask)
- void glStencilOp(GLenum sfail,GLenum dpfail,GLenum dppass)
- void glStencilOpSeparate(GLenum face,GLenum sfail,GLenum dpfail,GLenum dppass)
- void glTexCoord1s(GLshort s)
- void glTexCoord1i(GLint s)
- void glTexCoord1f(GLfloat s)
- void glTexCoord1d(GLdouble s)
- void glTexCoord2s(GLshort s,GLshort t)
- void glTexCoord2i(GLint s,GLint t)
- void glTexCoord2f(GLfloat s,GLfloat t)
- void glTexCoord2d(GLdouble s,GLdouble t)
- void glTexCoord3s(GLshort s,GLshort t,GLshort r)
- void glTexCoord3i(GLint s,GLint t,GLint r)
- void glTexCoord3f(GLfloat s,GLfloat t,GLfloat r)
- void glTexCoord3d(GLdouble s,GLdouble t,GLdouble r)
- void glTexCoord4s(GLshort s,GLshort t,GLshort r,GLshort q)
- void glTexCoord4i(GLint s,GLint t,GLint r,GLint q)
- void glTexCoord4f(GLfloat s,GLfloat t,GLfloat r,GLfloat q)

- `void glTexCoord4d(GLdouble s, GLdouble t, GLdouble r, GLdouble q)`
- `void glTexCoord1sv(const GLshort * v)`
- `void glTexCoord1iv(const GLint * v)`
- `void glTexCoord1fv(const GLfloat * v)`
- `void glTexCoord1dv(const GLdouble * v)`
- `void glTexCoord2sv(const GLshort * v)`
- `void glTexCoord2iv(const GLint * v)`
- `void glTexCoord2fv(const GLfloat * v)`
- `void glTexCoord2dv(const GLdouble * v)`
- `void glTexCoord3sv(const GLshort * v)`
- `void glTexCoord3iv(const GLint * v)`
- `void glTexCoord3fv(const GLfloat * v)`
- `void glTexCoord3dv(const GLdouble * v)`
- `void glTexCoord4sv(const GLshort * v)`
- `void glTexCoord4iv(const GLint * v)`
- `void glTexCoord4fv(const GLfloat * v)`
- `void glTexCoord4dv(const GLdouble * v)`
- `void glTexCoordPointer(GLint size, GLenum type, GLsizei stride, const GLvoid * pointer)`
- `void glTexEnvf(GLenum target, GLenum pname, GLfloat param)`
- `void glTexEnvf(GLenum target, GLenum pname, GLint param)`
- `void glTexGenf(GLenum coord, GLenum pname, GLfloat param)`
- `void glTexGenf(GLenum coord, GLenum pname, GLdouble param)`
- `void glTexGeniv(GLenum coord, GLenum pname, const GLint * params)`
- `void glTexGenfv(GLenum coord, GLenum pname, const GLfloat * params)`
- `void glTexGendv(GLenum coord, GLenum pname, const GLdouble * params)`
- `void glTexImage1D(GLenum target, GLint level, GLint internalFormat, GLsizei width, GLint border, GLenum format, GLenum type, const GLvoid * data)`
- `void glTexImage2D(GLenum target, GLint level, GLint internalFormat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, const GLvoid * data)`
- `void glTexImage3D(GLenum target, GLint level, GLint internalFormat, GLsizei width, GLsizei height, GLsizei depth, GLint border, GLenum format, GLenum type, const GLvoid * data)`
- `void glTexParameterf(GLenum target, GLenum pname, GLfloat param)`
- `void glTexParameteri(GLenum target, GLenum pname, GLint param)`
- `void glTexParameterfv(GLenum target, GLenum pname, const GLfloat * params)`
- `void glTexParameteriv(GLenum target, GLenum pname, const GLint * params)`

- `void glTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLenum type, const GLvoid * data)`
- `void glTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid * data)`
- `void glTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLenum type, const GLvoid * data)`
- `void glTranslated(GLdouble x, GLdouble y, GLdouble z)`
- `void glTranslatef(GLfloat x, GLfloat y, GLfloat z)`
- `void glUniform1f(GLint location, GLfloat v0)`
- `void glUniform2f(GLint location, GLfloat v0, GLfloat v1)`
- `void glUniform3f(GLint location, GLfloat v0, GLfloat v1, GLfloat v2)`
- `void glUniform4f(GLint location, GLfloat v0, GLfloat v1, GLfloat v2, GLfloat v3)`
- `void glUniform1i(GLint location, GLint v0)`
- `void glUniform2i(GLint location, GLint v0, GLint v1)`
- `void glUniform3i(GLint location, GLint v0, GLint v1, GLint v2)`
- `void glUniform4i(GLint location, GLint v0, GLint v1, GLint v2, GLint v3)`
- `void glUniform1fv(GLint location, GLsizei count, const GLfloat *value)`
- `void glUniform2fv(GLint location, GLsizei count, const GLfloat *value)`
- `void glUniform3fv(GLint location, GLsizei count, const GLfloat *value)`
- `void glUniform4fv(GLint location, GLsizei count, const GLfloat *value)`
- `void glUniform1iv(GLint location, GLsizei count, const GLint *value)`
- `void glUniform2iv(GLint location, GLsizei count, const GLint *value)`
- `void glUniform3iv(GLint location, GLsizei count, const GLint *value)`
- `void glUniform4iv(GLint location, GLsizei count, const GLint *value)`
- `void glUniformMatrix2fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix3fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix4fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix2x3fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix3x2fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix2x4fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix4x2fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix3x4fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUniformMatrix4x3fv(GLint location, GLsizei count, GLboolean transpose, const GLfloat *value)`
- `void glUseProgram(GLuint program)`
- `void glValidateProgram(GLuint program)`
- `void glVertex2s(GLshort x, GLshort y)`
- `void glVertex2i(GLint x, GLint y)`

- `void glVertex2f(GLfloat x,GLfloat y)`
- `void glVertex2d(GLdouble x,GLdouble y)`
- `void glVertex3s(GLshort x,GLshort y,GLshort z)`
- `void glVertex3i(GLint x,GLint y,GLint z)`
- `void glVertex3f(GLfloat x,GLfloat y,GLfloat z)`
- `void glVertex3d(GLdouble x,GLdouble y,GLdouble z)`
- `void glVertex4s(GLshort x,GLshort y,GLshort z,GLshort w)`
- `void glVertex4i(GLint x,GLint y,GLint z,GLint w)`
- `void glVertex4f(GLfloat x,GLfloat y,GLfloat z,GLfloat w)`
- `void glVertex4d(GLdouble x,GLdouble y,GLdouble z,GLdouble w)`
- `void glVertex2sv(const GLshort * v)`
- `void glVertex2iv(const GLint * v)`
- `void glVertex2fv(const GLfloat * v)`
- `void glVertex2dv(const GLdouble * v)`
- `void glVertex3sv(const GLshort * v)`
- `void glVertex3iv(const GLint * v)`
- `void glVertex3fv(const GLfloat * v)`
- `void glVertex3dv(const GLdouble * v)`
- `void glVertex4sv(const GLshort * v)`
- `void glVertex4iv(const GLint * v)`
- `void glVertex4fv(const GLfloat * v)`
- `void glVertex4dv(const GLdouble * v)`
- `void glVertexAttrib1f(GLuint index,GLfloat v0)`
- `void glVertexAttrib1s(GLuint index,GLshort v0)`
- `void glVertexAttrib1d(GLuint index,GLdouble v0)`
- `void glVertexAttrib2f(GLuint index,GLfloat v0,GLfloat v1)`
- `void glVertexAttrib2s(GLuint index,GLshort v0,GLshort v1)`
- `void glVertexAttrib2d(GLuint index,GLdouble v0,GLdouble v1)`
- `void glVertexAttrib3f(GLuint index,GLfloat v0,GLfloat v1,GLfloat v2)`
- `void glVertexAttrib3s(GLuint index,GLshort v0,GLshort v1,GLshort v2)`
- `void glVertexAttrib3d(GLuint index,GLdouble v0,GLdouble v1,GLdouble v2)`
- `void glVertexAttrib4f(GLuint index,GLfloat v0,GLfloat v1,GLfloat v2,GLfloat v3)`
- `void glVertexAttrib4s(GLuint index,GLshort v0,GLshort v1,GLshort v2,GLshort v3)`
- `void glVertexAttrib4d(GLuint index,GLdouble v0,GLdouble v1,GLdouble v2,GLdouble v3)`
- `void glVertexAttrib4Nub(GLuint index,GLubyte v0,GLubyte v1,GLubyte v2,GLubyte v3)`
- `void glVertexAttrib1fv(GLuint index,const GLfloat *v)`

- void glVertexAttrib1sv(GLuint index,const GLshort *v)
- void glVertexAttrib1dv(GLuint index,const GLdouble *v)
- void glVertexAttrib2fv(GLuint index,const GLfloat *v)
- void glVertexAttrib2sv(GLuint index,const GLshort *v)
- void glVertexAttrib2dv(GLuint index,const GLdouble *v)
- void glVertexAttrib3fv(GLuint index,const GLfloat *v)
- void glVertexAttrib3sv(GLuint index,const GLshort *v)
- void glVertexAttrib3dv(GLuint index,const GLdouble *v)
- void glVertexAttrib4fv(GLuint index,const GLfloat *v)
- void glVertexAttrib4sv(GLuint index,const GLshort *v)
- void glVertexAttrib4dv(GLuint index,const GLdouble *v)
- void glVertexAttrib4iv(GLuint index,const GLint *v)
- void glVertexAttrib4bv(GLuint index,const GLbyte *v)
- void glVertexAttrib4ubv(GLuint index,const GLubyte *v)
- void glVertexAttrib4usv(GLuint index,const GLushort *v)
- void glVertexAttrib4uiv(GLuint index,const GLuint *v)
- void glVertexAttribPointer(GLuint index,GLint size,GLenum type,GLboolean normalized,GLsizei stride,const GLvoid * pointer)
- void glVertexPointer(GLint size,GLenum type,GLsizei stride,const GLvoid * pointer)
- void glViewport(GLint x,GLint y,GLsizei width,GLsizei height)
- void glWindowPos2s(GLshort x,GLshort y)
- void glWindowPos2i(GLint x,GLint y)
- void glWindowPos2f(GLfloat x,GLfloat y)
- void glWindowPos2d(GLdouble x,GLdouble y)
- void glWindowPos3s(GLshort x,GLshort y,GLshort z)
- void glWindowPos3i(GLint x,GLint y,GLint z)
- void glWindowPos3f(GLfloat x,GLfloat y,GLfloat z)
- void glWindowPos3d(GLdouble x,GLdouble y,GLdouble z)
- void glWindowPos2sv(const GLshort * v)
- void glWindowPos2iv(const GLint * v)
- void glWindowPos2fv(const GLfloat * v)
- void glWindowPos2dv(const GLdouble * v)
- void glWindowPos3sv(const GLshort * v)
- void glWindowPos3iv(const GLint * v)
- void glWindowPos3fv(const GLfloat * v)
- void glWindowPos3dv(const GLdouble * v)

- void gluBeginCurve(GLUnurbs* nurb)
- void gluBeginPolygon(GLUtesselator* tess)
- void gluBeginSurface(GLUnurbs* nurb)
- void gluBeginTrim(GLUnurbs* nurb)
- void gluCylinder(GLUquadric* quad, GLdouble base, GLdouble top, GLdouble height, GLint slices, GLint stacks)
- void gluDeleteNurbsRenderer(GLUnurbs* nurb)
- void gluDeleteQuadric(GLUquadric* quad)
- void gluDeleteTess(GLUtesselator* tess)
- void gluDisk(GLUquadric* quad, GLdouble inner, GLdouble outer, GLint slices, GLint loops)
- void gluEndCurve(GLUnurbs* nurb)
- void gluEndPolygon(GLUtesselator* tess)
- void gluEndSurface(GLUnurbs* nurb)
- void gluEndTrim(GLUnurbs* nurb)
- const GLubyte * gluErrorString(GLenum error)
- void gluGetNurbsProperty(GLUnurbs* nurb, GLenum property, GLfloat* data)
- const GLubyte * gluGetString(GLenum name)
- void gluGetTessProperty(GLUtesselator* tess, GLenum which, GLdouble* data)
- void gluLoadSamplingMatrices(GLUnurbs* nurb, const GLfloat * model, const GLfloat * perspective, const GLint * view)
- void gluLookAt(GLdouble eyeX, GLdouble eyeY, GLdouble eyeZ, GLdouble centerX, GLdouble centerY, GLdouble centerZ, GLdouble upX, GLdouble upY, GLdouble upZ)
- GLUnurbs * gluNewNurbsRenderer(void)
- GLUquadric * gluNewQuadric(void)
- GLUtesselator* gluNewTess(void)
- void gluNextContour(GLUtesselator* tess, GLenum type)
- void gluNurbsCurve(GLUnurbs* nurb, GLint knotCount, GLfloat * knots, GLint stride, GLfloat * control, GLint order, GLenum type)
- void gluNurbsProperty(GLUnurbs* nurb, GLenum property, GLfloat value)
- void gluNurbsSurface(GLUnurbs* nurb, GLint sKnotCount, GLfloat* sKnots, GLint tKnotCount, GLfloat* tKnots, GLint sStride, GLint tStride, GLfloat* control, GLint sOrder, GLint tOrder, GLenum type)
- void gluOrtho2D(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top)
- void gluPartialDisk(GLUquadric* quad, GLdouble inner, GLdouble outer, GLint slices, GLint loops, GLdouble start, GLdouble sweep)
- void gluPerspective(GLdouble fovy, GLdouble aspect, GLdouble zNear, GLdouble zFar)
- void gluPickMatrix(GLdouble x, GLdouble y, GLdouble delX, GLdouble delY, GLint * viewport)
- GLint gluProject(GLdouble objX, GLdouble objY, GLdouble objZ, const GLdouble * model, const GLdouble * proj, const GLint * view, GLdouble* winX, GLdouble* winY, GLdouble* winZ)
- void gluPwlCurve(GLUnurbs* nurb, GLint count, GLfloat* data, GLint stride, GLenum type)

- void gluQuadricDrawStyle(GLUquadric* quad, GLenum draw)
- void gluQuadricNormals(GLUquadric* quad, GLenum normal)
- void gluQuadricOrientation(GLUquadric* quad, GLenum orientation)
- void gluQuadricTexture(GLUquadric* quad, GLboolean texture)
- GLint gluScaleImage(GLenum format, GLsizei wIn, GLsizei hIn, GLenum typeIn, const void * dataIn, GLsizei wOut, GLsizei hOut, GLenum typeOut, GLvoid* dataOut)
- void gluSphere(GLUquadric* quad, GLdouble radius, GLint slices, GLint stacks)
- void gluTessBeginContour(GLUtesselator* tess)
- void gluTessBeginPolygon(GLUtesselator* tess, GLvoid* data)
- void gluTessEndContour(GLUtesselator* tess)
- void gluTessEndPolygon(GLUtesselator* tess)
- void gluTessNormal(GLUtesselator* tess, GLdouble valueX, GLdouble valueY, GLdouble valueZ)
- void gluTessProperty(GLUtesselator* tess, GLenum which, GLdouble data)
- void gluTessVertex(GLUtesselator* tess, GLdouble * location, GLvoid* data)
- GLint gluUnProject(GLdouble winX, GLdouble winY, GLdouble winZ, const GLdouble * model, const GLdouble * proj, const GLint * view, GLdouble* objX, GLdouble* objY, GLdouble* objZ)
- void glDisable(GLenum cap)

RINGQT CLASSES REFERENCE

99.1 QApp Class

C++ Reference : <http://doc.qt.io/qt-5/QApplication.html>

Parent Class : QGuiApplication

- void exec(void)
- void quit(void)
- void processEvents(void)
- void styleWindows(void)
- void styleWindowsVista(void)
- void styleFusion(void)
- void styleFusionBlack(void)
- void styleFusionCustom(QColor,QColor,QColor,QColor,QColor,QColor,QColor,QColor,QColor,QColor,QColor,QColor)
- void closeAllWindows(void)
- Qt::KeyboardModifiers keyboardModifiers(void)

99.2 QDesktopServices Class

C++ Reference : <http://doc.qt.io/qt-5/QDesktopServices.html>

- bool openUrl(QUrl)
- void setUrlHandler(QString, QObject *receiver, const char *method)
- void unsetUrlHandler(QString)

99.3 QTest Class

C++ Reference : <http://doc.qt.io/qt-5/QTest.html>

- void qsleep(int)

99.4 QObject Class

C++ Reference : <http://doc.qt.io/qt-5/QObject.html>

Parameters : void

- bool blockSignals(bool block)
- QObjectList children(void)
- void dumpObjectInfo(void)
- void dumpObjectTree(void)
- bool inherits(const char *className)
- void installEventFilter(QObject *filterObj)
- bool isWidgetType(void)
- void killTimer(int id)
- void moveToThread(QThread *targetThread)
- QString objectName(void)
- QObject *parent(void)
- QVariant property(const char *name)
- void removeEventFilter(QObject *obj)
- void setObjectName(QString)
- void setParent(QObject *parent)
- bool setProperty(const char *name, QVariant)
- bool signalsBlocked(void)
- int startTimer(int interval)
- QThread *thread(void)
- void deleteLater(void)

99.5 QWidget Class

C++ Reference : <http://doc.qt.io/qt-5/QWidget.html>

Parameters : void

Parent Class : QObject

- bool acceptDrops(void)
- QString accessibleDescription(void)
- QString accessibleName(void)
- void activateWindow(void)
- void addAction(QAction *action)
- void adjustSize(void)
- bool autoFillBackground(void)

- `int backgroundRole(void)`
- `QSize baseSize(void)`
- `QWidget *childAt(int x, int y)`
- `QRect childrenRect(void)`
- `QRegion childrenRegion(void)`
- `void clearFocus(void)`
- `void clearMask(void)`
- `QMargins contentsMargins(void)`
- `QRect contentsRect(void)`
- `int contextMenuPolicy(void)`
- `QCursor cursor(void)`
- `int effectiveWinId(void)`
- `void ensurePolished(void)`
- `int focusPolicy(void)`
- `QWidget *focusProxy(void)`
- `QWidget *focusWidget(void)`
- `QFont font(void)`
- `QFontInfo fontInfo(void)`
- `int foregroundRole(void)`
- `QRect frameGeometry(void)`
- `QSize frameSize(void)`
- `QRect geometry(void)`
- `void getContentsMargins(int *left, int *top, int *right, int *bottom)`
- `QPixmap grab(QRect)`
- `void grabGesture(Qt::GestureType gesture, Qt::GestureFlag flags)`
- `void grabKeyboard(void)`
- `void grabMouse(void)`
- `int grabShortcut(QKeySequence , Qt::ShortcutContext context)`
- `QGraphicsEffect *graphicsEffect(void)`
- `QGraphicsProxyWidget *graphicsProxyWidget(void)`
- `bool hasFocus(void)`
- `bool hasMouseTracking(void)`
- `int height(void)`
- `int heightForWidth(int w)`
- `int inputMethodHints(void)`
- `QVariant inputMethodQuery(Qt::InputMethodQuery query)`

- void insertAction(QAction *before, QAction *action)
- bool isActiveWindow(void)
- bool isAncestorOf(QWidget *child)
- bool isEnabled(void)
- bool isEnabledTo(QWidget *ancestor)
- bool isFullScreen(void)
- bool isHidden(void)
- bool isMaximized(void)
- bool isMinimized(void)
- bool isModal(void)
- bool isVisible(void)
- bool isVisibleTo(QWidget *ancestor)
- bool isWindow(void)
- bool isWindowModified(void)
- QLayout *layout(void)
- int layoutDirection(void)
- QLocale locale(void)
- QPoint mapFrom(QWidget *parent, QPoint)
- QPoint mapFromGlobal(QPoint)
- QPoint mapFromParent(QPoint)
- QPoint mapTo(QWidget *parent, QPoint)
- QPoint mapToGlobal(QPoint pos)
- QPoint mapToParent(QPoint pos)
- QRegion mask(void)
- int maximumHeight(void)
- QSize maximumSize(void)
- int maximumWidth(void)
- int minimumHeight(void)
- QSize minimumSize(void)
- int minimumWidth(void)
- void move(int x, int y)
- QWidget *nativeParentWidget(void)
- QWidget *nextInFocusChain(void)
- QRect normalGeometry(void)
- void overrideWindowFlags(Qt::WindowType flags)
- QPalette palette(void)

- QWidget *parentWidget(void)
- QPoint pos(void)
- QWidget *previousInFocusChain(void)
- QRect rect(void)
- void releaseKeyboard(void)
- void releaseMouse(void)
- void releaseShortcut(int id)
- void removeAction(QAction *action)
- void render(QPaintDevice *target, QPoint, QRegion, QWidget::RenderFlag)
- void repaint(void)
- void resize(int w, int h)
- bool restoreGeometry(QByteArray)
- QByteArray saveGeometry(void)
- void scroll(int dx, int dy)
- void setAcceptDrops(bool on)
- void setAccessibleDescription(QString)
- void setAccessibleName(QString)
- void setAttribute(Qt::WidgetAttribute attribute, bool on)
- void setAutoFillBackground(bool enabled)
- void setBackgroundRole(QPalette::ColorRole role)
- void setBaseSize(int basew, int baseh)
- void setContentsMargins(int left, int top, int right, int bottom)
- void setContextMenuPolicy(Qt::ContextMenuPolicy policy)
- void setCursor(QCursor)
- void setFixedHeight(int h)
- void setFixedSize(int w, int h)
- void setFixedWidth(int w)
- void setFocus(Qt::FocusReason reason)
- void setFocusPolicy(Qt::FocusPolicy policy)
- void setFocusProxy(QWidget *w)
- void setFont(QFont)
- void setForegroundRole(QPalette::ColorRole role)
- void setGeometry(int x, int y, int w, int h)
- void setGraphicsEffect(QGraphicsEffect *effect)
- void setInputMethodHints(Qt::InputMethodHint hints)
- void setLayout(QLayout *layout)

- void setLayoutDirection(Qt::LayoutDirection direction)
- void setLocale(QLocale)
- void setMask(QBitmap)
- void setMaximumHeight(int maxh)
- void setMaximumSize(int maxw, int maxh)
- void setMaximumWidth(int maxw)
- void setMinimumHeight(int minh)
- void setMinimumSize(int minw, int minh)
- void setMinimumWidth(int minw)
- void setMouseTracking(bool enable)
- void setPalette(QPalette)
- void setParent(QWidget *parent)
- void setShortcutAutoRepeat(int id, bool enable)
- void setShortcutEnabled(int id, bool enable)
- void setSizeIncrement(int w, int h)
- void setSizePolicy(QSizePolicy::Policy horizontal, QSizePolicy::Policy vertical)
- void setStatusTip(QString)
- void setStyle(QStyle *style)
- void setToolTip(QString)
- void setUpdatesEnabled(bool enable)
- void setWhatsThis(QString)
- void setWindowFilePath(QString)
- void setWindowFlags(Qt::WindowType type)
- void setWindowIcon(QIcon)
- void setWindowIconText(QString)
- void setWindowModality(Qt::WindowModality windowModality)
- void setWindowOpacity(double level)
- void setWindowRole(QString)
- void setWindowState(Qt::WindowState windowState)
- QSize size(void)
- QSize sizeIncrement(void)
- QSizePolicy sizePolicy(void)
- void stackUnder(QWidget *w)
- QString statusTip(void)
- QStyle *style(void)
- QString styleSheet(void)

- `bool testAttribute(Qt::WidgetAttribute attribute)`
- `QString toolTip(void)`
- `bool underMouse(void)`
- `void ungrabGesture(Qt::GestureType gesture)`
- `void unsetCursor(void)`
- `void unsetLayoutDirection(void)`
- `void unsetLocale(void)`
- `void update(int x, int y, int w, int h)`
- `void updateGeometry(void)`
- `bool updatesEnabled(void)`
- `QRegion visibleRegion(void)`
- `QString whatsThis(void)`
- `int width(void)`
- `int winId(void)`
- `QWidget *window(void)`
- `QString windowFilePath(void)`
- `int windowFlags(void)`
- `QWindow *windowHandle(void)`
- `QIcon windowIcon(void)`
- `QString windowIconText(void)`
- `int windowModality(void)`
- `double windowOpacity(void)`
- `QString windowRole(void)`
- `int windowState(void)`
- `QString windowTitle(void)`
- `int windowType(void)`
- `int x(void)`
- `int y(void)`
- `bool close(void)`
- `void hide(void)`
- `void lower(void)`
- `void raise(void)`
- `void setDisabled(bool disable)`
- `void setEnabled(bool)`
- `void setHidden(bool hidden)`
- `void setStyleSheet(QString)`

- void setWindowModified(bool)
- void setWindowTitle(QString)
- void show(void)
- void showFullScreen(void)
- void showMaximized(void)
- void showMinimized(void)
- void showNormal(void)
- QWidget *find(int id)
- QWidget *keyboardGrabber(void)
- QWidget *mouseGrabber(void)
- void setTabOrder(QWidget *first, QWidget *second)

99.6 QLabel Class

C++ Reference : <http://doc.qt.io/qt-5/QLabel.html>

Parameters : QWidget *

Parent Class : QWidget

- QWidget *buddy(void)
- bool hasScaledContents(void)
- bool hasSelectedText(void)
- int indent(void)
- int margin(void)
- QMovie *movie(void)
- bool openExternalLinks(void)
- QPicture *picture(void)
- QPixmap *pixmap(void)
- QString selectedText(void)
- int selectionStart(void)
- void setAlignment(Qt::AlignmentFlag)
- void setBuddy(QWidget *buddy)
- void setIndent(int)
- void setMargin(int)
- void setOpenExternalLinks(bool open)
- void setScaledContents(bool)
- void setSelection(int start, int length)
- void setTextFormat(Qt::TextFormat)

- void setTextInteractionFlags(Qt::TextInteractionFlag flags)
- void setWordWrap(bool on)
- QString text(void)
- int textFormat(void)
- int textInteractionFlags(void)
- bool wordWrap(void)
- void clear(void)
- void setMovie(QMovie *movie)
- void setNum(double num)
- void setPicture(QPicture)
- void setPixmap(QPixmap)
- void setText(QString)

99.7 QPushButton Class

C++ Reference : <http://doc.qt.io/qt-5/QPushButton.html>

Parameters : QWidget *

Parent Class : QAbstractButton

- void setClickEvent(const char *)
- void setIcon(QIcon)
- void setIconSize(QSize)
- const char *getClickEvent(void)

99.8 QBitmap Class

C++ Reference : <http://doc.qt.io/qt-5/QBitmap.html>

Parameters : void

Parent Class : QPixmap

- void clear(void)
- void swap(QBitmap)
- QBitmap transformed(QTransform)
- QBitmap fromData(QSize, const uchar * bits, QImage::Format monoFormat)
- QBitmap fromImage(QImage, Qt::ImageConversionFlags flags)

99.9 QPixmap Class

C++ Reference : <http://doc.qt.io/qt-5/QPixmap.html>

Parameters : const char *

- QPixmap copy(int x, int y, int width, int height)
- QPixmap scaled(int width, int height, Qt::AspectRatioMode aspectRatioMode, Qt::TransformationMode transformMode)
- int width(void)
- int height(void)
- QBitmap createMaskFromColor(QColor , Qt::MaskMode)
- QBitmap mask(void)
- void setMask(QBitmap)
- void fill(QColor)
- QPixmap fromImage(QImage, Qt::ImageConversionFlags)
- bool load(QString, const char *, Qt::ImageConversionFlags)
- qint64 cacheKey(void)
- bool convertFromImage(QImage image, Qt::ImageConversionFlags flags)
- QPixmap copy_2(QRect rectangle)
- QBitmap createHeuristicMask(bool clipTight)
- int depth(void)
- void detach(void)
- qreal devicePixelRatio(void)
- bool hasAlpha(void)
- bool hasAlphaChannel(void)
- bool isNull(void)
- bool isQBitmap(void)
- bool loadFromData(uchar *data, uint len, char *format, Qt::ImageConversionFlags flags)
- bool loadFromData_2(QByteArray data, char *format, Qt::ImageConversionFlags flags)
- QRect rect(void)
- bool save(QString fileName, char *format, int quality)
- bool save_2(QIODevice *device, char *format, int quality)
- QPixmap scaled_2(QSize size, Qt::AspectRatioMode aspectRatioMode, Qt::TransformationMode transformMode)
- QPixmap scaledToHeight(int height, Qt::TransformationMode mode)
- QPixmap scaledToWidth(int width, Qt::TransformationMode mode)
- void scroll(int dx, int dy, int x, int y, int width, int height, QRegion *exposed)
- void scroll_2(int dx, int dy, QRect rect, QRegion *exposed)

- void setDevicePixelRatio(qreal scaleFactor)
- QSize size(void)
- void swap(QPixmap other)
- QImage toImage(void)
- QPixmap transformed(QTransform transform, Qt::TransformationMode mode)
- QPixmap transformed_2(QMatrix matrix, Qt::TransformationMode mode)
- int defaultDepth(void)
- QPixmap fromImage_2(QImage image, Qt::ImageConversionFlags flags)
- QPixmap fromImageReader(QImageReader *imageReader, Qt::ImageConversionFlags flags)
- QTransform trueMatrix(QTransform matrix, int width, int height)
- QMatrix trueMatrix_2(QMatrix m, int w, int h)

99.10 QPixmap2 Class

C++ Reference : <http://doc.qt.io/qt-5/QPixmap2.html>

Parameters : int width, int height

Parent Class : QPixmap

99.11 QIcon Class

C++ Reference : <http://doc.qt.io/qt-5/QIcon.html>

Parameters : QPixmap

99.12 QSize Class

C++ Reference : <http://doc.qt.io/qt-5/QSize.html>

Parameters : int width, int height

99.13 QLineEdit Class

C++ Reference : <http://doc.qt.io/qt-5/QLineEdit.html>

Parameters : QWidget *

Parent Class : QWidget

- int alignment(void)
- void backspace(void)
- QCompleter *completer(void)
- QMenu *createStandardContextMenu(void)

- void cursorBackward(bool mark, int steps)
- void cursorForward(bool mark, int steps)
- int cursorMoveStyle(void)
- int cursorPosition(void)
- int cursorPositionAt(QPoint)
- void cursorWordBackward(bool mark)
- void cursorWordForward(bool mark)
- void del(void)
- void deselect(void)
- QString displayText(void)
- bool dragEnabled(void)
- int echoMode(void)
- void end(bool mark) # In RingQt use : void endtext(bool mark)
- void getTextMargins(int *left, int *top, int *right, int *bottom)
- bool hasAcceptableInput(void)
- bool hasFrame(void)
- bool hasSelectedText(void)
- void home(bool mark)
- QString inputMask(void)
- void insert(QString)
- bool isModified(void)
- bool isReadOnly(void)
- bool isRedoAvailable(void)
- bool isUndoAvailable(void)
- int maxLength(void)
- QString placeholderText(void)
- QString selectedText(void)
- int selectionStart(void)
- void setAlignment(Qt::AlignmentFlag flag)
- void setCompleter(QCompleter *c)
- void setCursorMoveStyle(Qt::CursorMoveStyle style)
- void setCursorPosition(int)
- void setDragEnabled(bool b)
- void setEchoMode(QLineEdit::EchoMode)
- void setFrame(bool)
- void setInputMask(QString)

- void setMaxLength(int)
- void setModified(bool)
- void setPlaceholderText(QString)
- void setReadOnly(bool)
- void setSelection(int start, int length)
- void setTextMargins(int left, int top, int right, int bottom)
- void setValidator(QValidator *v)
- QString text(void)
- QMargins textMargins(void)
- QValidator *validator(void)
- void clear(void)
- void copy(void)
- void cut(void)
- void paste(void)
- void redo(void)
- void selectAll(void)
- void setText(QString)
- void undo(void)
- void setTextChangedEvent(const char *)
- void setCursorPositionChangedEvent(const char *)
- void seteditingFinishedEvent(const char *)
- void setreturnPressedEvent(const char *)
- void setselectionChangedEvent(const char *)
- void settextEditedEvent(const char *)
- const char *getTextChangedEvent(void)
- const char *getCursorPositionChangedEvent(void)
- const char *geteditingFinishedEvent(void)
- const char *getreturnPressedEvent(void)
- const char *getselectionChangedEvent(void)
- const char *gettextEditedEvent(void)

99.14 QVBoxLayout Class

C++ Reference : <http://doc.qt.io/qt-5/QVBoxLayout.html>

Parameters : void

Parent Class : QBoxLayout

- void addLayout(QLayout *)

99.15 QHBoxLayout Class

C++ Reference : <http://doc.qt.io/qt-5/QHBoxLayout.html>

Parameters : void

Parent Class : QVBoxLayout

- void addLayout(QLayout *)

99.16 QTextEdit Class

C++ Reference : <http://doc.qt.io/qt-5/QTextEdit.html>

Parameters : QWidget *

Parent Class : QAbstractScrollArea

- bool acceptRichText(void)
- int alignment(void)
- QString anchorAt(QPoint)
- bool canPaste(void)
- QTextCharFormat currentCharFormat(void)
- QFont currentFont(void)
- QTextCursor cursorForPosition(QPoint)
- QRect cursorRect(void)
- int cursorWidth(void)
- QTextDocument *document(void)
- QString documentTitle(void)
- void ensureCursorVisible(void)
- bool find(QString, QTextDocument::FindFlag)
- QString fontFamily(void)
- bool fontItalic(void)
- double fontPointSize(void)
- bool fontUnderline(void)
- int fontWeight(void)
- bool isReadOnly(void)
- bool isUndoRedoEnabled(void)
- int lineWrapColumnOrWidth(void)
- QVariant loadResource(int, QUrl)
- void mergeCurrentCharFormat(QTextCharFormat)

- void moveCursor(QTextCursor::MoveOperation operation, QTextCursor::MoveMode mode)
- bool overwriteMode(void)
- void print(QPrinter * printer)
- void setAcceptRichText(bool accept)
- void setCurrentCharFormat(QTextCharFormat)
- void setCursorWidth(int width)
- void setDocument(QTextDocument *document)
- void setDocumentTitle(QString)
- void setLineWrapColumnOrWidth(int w)
- void setLineWrapMode(QTextEdit::LineWrapMode)
- void setOverwriteMode(bool overwrite)
- void setReadOnly(bool)
- void setTabChangesFocus(bool)
- void setTabStopWidth(int width)
- void setTextCursor(QTextCursor)
- void setTextInteractionFlags(Qt::TextInteractionFlag flags)
- void setUndoRedoEnabled(bool enable)
- void setWordWrapMode(QTextOption::WrapMode policy)
- bool tabChangesFocus(void)
- int tabStopWidth(void)
- QColor textBackgroundColor(void)
- QColor textColor(void)
- QTextCursor textCursor(void)
- int textInteractionFlags(void)
- QString toHtml(void)
- QString toPlainText(void)
- int wordWrapMode(void)
- void append(QString)
- void clear(void)
- void copy(void)
- void cut(void)
- void insertHtml(QString)
- void insertPlainText(QString)
- void paste(void)
- void redo(void)
- void scrollToAnchor(QString)

- void selectAll(void)
- void setAlignment(Qt::AlignmentFlag a)
- void setCurrentFont(QFont)
- void setFontFamily(QString)
- void setFontItalic(bool italic)
- void setFontSize(double s)
- void setFontUnderline(bool underline)
- void setFontWeight(int weight)
- void setHtml(QString)
- void setPlainText(QString)
- void setText(QString)
- void setTextBackgroundColor(QColor)
- void setTextColor(QColor)
- void undo(void)
- void zoomIn(int range)
- void zoomOut(int range)
- void setcopyAvailableEvent(const char *)
- void setcurrentCharFormatChangedEvent(const char *)
- void setcursorPositionChangedEvent(const char *)
- void setredoAvailableEvent(const char *)
- void setselectionChangedEvent(const char *)
- void settextChangedEvent(const char *)
- void setundoAvailableEvent(const char *)
- const char *getcopyAvailableEvent(void)
- const char *getcurrentCharFormatChangedEvent(void)
- const char *getcursorPositionChangedEvent(void)
- const char *getredoAvailableEvent(void)
- const char *getselectionChangedEvent(void)
- const char *gettextChangedEvent(void)
- const char *getundoAvailableEvent(void)
- void cyanline(void)
- void setactivelinecolor(QColor)

99.17 QListWidget Class

C++ Reference : <http://doc.qt.io/qt-5/QListWidget.html>

Parameters : QWidget *

Parent Class : QListView

- void addItem(QString)
- int count(void)
- void editItem(QListWidgetItem *item)
- bool isSortingEnabled(void)
- QListWidgetItem *item(int row)
- QListWidgetItem *itemAt(int x, int y)
- QWidget *itemWidget(QListWidgetItem *item)
- void openPersistentEditor(QListWidgetItem *item)
- void removeItemWidget(QListWidgetItem *item)
- int row(QListWidgetItem *item)
- void setCurrentRow(int row, QItemSelectionModel::SelectionFlag command)
- void setItemWidget(QListWidgetItem *item, QWidget *widget)
- void setSortingEnabled(bool enable)
- void sortItems(Qt::SortOrder order)
- QListWidgetItem *takeItem(int row)
- QRect visualItemRect(QListWidgetItem *item)
- void clear(void)
- void scrollToItem(QListWidgetItem *item, QAbstractItemView::ScrollHint hint)
- void setCurrentItemChangedEvent(const char *)
- void setCurrentRowChangedEvent(const char *)
- void setCurrentTextChangedEvent(const char *)
- void setItemActivatedEvent(const char *)
- void setItemChangedEvent(const char *)
- void setItemClickedEvent(const char *)
- void setItemDoubleClickedEvent(const char *)
- void setItemEnteredEvent(const char *)
- void setItemPressedEvent(const char *)
- void setItemSelectionChangedEvent(const char *)
- const char *getCurrentItemChangedEvent(void)
- const char *getCurrentRowChangedEvent(void)
- const char *getCurrentTextChangedEvent(void)

- `const char *getitemActivatedEvent(void)`
- `const char *getitemChangedEvent(void)`
- `const char *getitemClickedEvent(void)`
- `const char *getitemDoubleClickedEvent(void)`
- `const char *getitemEnteredEvent(void)`
- `const char *getitemPressedEvent(void)`
- `const char *getitemSelectionChangedEvent(void)`

99.18 QTreeView Class

C++ Reference : <http://doc.qt.io/qt-5/QTreeView.html>

Parameters : QWidget *

Parent Class : QAbstractItemView

- `bool allColumnsShowFocus(void)`
- `int autoExpandDelay(void)`
- `int columnAt(int x)`
- `int columnViewportPosition(int column)`
- `int columnWidth(int column)`
- `bool expandsOnDoubleClick(void)`
- `QHeaderView *header(void)`
- `int indentation(void)`
- `QModelIndex indexAbove(QModelIndex)`
- `QModelIndex indexBelow(QModelIndex)`
- `bool isAnimated(void)`
- `bool isColumnHidden(int column)`
- `bool isExpanded(QModelIndex)`
- `bool isFirstColumnSpanned(int row, QModelIndex)`
- `bool isHeaderHidden(void)`
- `bool isRowHidden(int row, QModelIndex)`
- `bool isSortingEnabled(void)`
- `bool itemsExpandable(void)`
- `bool rootIsDecorated(void)`
- `void setAllColumnsShowFocus(bool enable)`
- `void setAnimated(bool enable)`
- `void setAutoExpandDelay(int delay)`
- `void setColumnHidden(int column, bool hide)`

- void setColumnWidth(int column, int width)
- void setExpanded(QModelIndex, bool expanded)
- void setExpandsOnDoubleClick(bool enable)
- void setFirstColumnSpanned(int row, QModelIndex, bool span)
- void setHeader(QHeaderView * header)
- void setHeaderHidden(bool hide)
- void setIndentation(int i)
- void setItemsExpandable(bool enable)
- void setRootIsDecorated(bool show)
- void setRowHidden(int row, QModelIndex, bool hide)
- void setSortingEnabled(bool enable)
- void setUniformRowHeights(bool uniform)
- void setWordWrap(bool on)
- void sortByColumn(int column, Qt::SortOrder order)
- bool uniformRowHeights(void)
- bool wordWrap(void)
- void dataChanged(QModelIndex, QModelIndex)
- QModelIndex indexAt(QPoint)
- void keyboardSearch(QString)
- void reset(void)
- void scrollTo(QModelIndex, QAbstractItemView::ScrollHint)
- void selectAll(void)
- void setModel(QAbstractItemModel *model)
- void setRootIndex(QModelIndex)
- void setSelectionModel(QItemSelectionModel *selectionModel)
- QRect visualRect(QModelIndex)
- void collapse(QModelIndex)
- void collapseAll(void)
- void expand(QModelIndex)
- void expandAll(void)
- void expandToDepth(int depth)
- void hideColumn(int column)
- void resizeColumnToContents(int column)
- void showColumn(int column)
- void setcollapsedEvent(const char *)
- void setexpandedEvent(const char *)

- void setactivatedEvent(const char *)
- void setclickedEvent(const char *)
- void setdoubleClickedEvent(const char *)
- void setenteredEvent(const char *)
- void setpressedEvent(const char *)
- void setviewportEnteredEvent(const char *)
- const char *getcollapsedEvent(void)
- const char *getexpandedEvent(void)
- const char *getactivatedEvent(void)
- const char *getclickedEvent(void)
- const char *getdoubleClickedEvent(void)
- const char *getenteredEvent(void)
- const char *getpressedEvent(void)
- const char *getviewportEnteredEvent(void)

99.19 QDir Class

C++ Reference : <http://doc.qt.io/qt-5/QDir.html>

Parameters : void

- void setNameFilters(QStringList)

99.20 QFileSystemModel Class

C++ Reference : <http://doc.qt.io/qt-5/QFileSystemModel.html>

Parameters : void

- QIcon fileIcon(QModelIndex)
- QFileInfo fileInfo(QModelIndex)
- QString fileName(QModelIndex)
- QString filePath(QModelIndex)
- int filter(void)
- QFileIconProvider *iconProvider(void)
- QModelIndex index(QString, int column)
- bool isDir(QModelIndex)
- bool isReadOnly(void)
- QDateTime lastModified(QModelIndex)
- QModelIndex mkdir(QModelIndex, QString)
- QVariant myComputer(int role)

- bool nameFilterDisables(void)
- QStringList nameFilters(void)
- int permissions(QModelIndex)
- bool remove(QModelIndex)
- bool resolveSymlinks(void)
- bool rmdir(QModelIndex)
- QDir rootDirectory(void)
- QString rootPath(void)
- void setFilter(QDir::Filter filters)
- void setIconProvider(QFileIconProvider *provider)
- void setNameFilterDisables(bool enable)
- void setNameFilters(QStringList)
- void setReadOnly(bool enable)
- void setResolveSymlinks(bool enable)
- QModelIndex setRootPath(QString)
- int size(QModelIndex)
- QString type(QModelIndex)
- bool canFetchMore(QModelIndex)
- int columnCount(void)
- QVariant data(QModelIndex index, int role)
- bool dropMimeData(QMimeData *data, Qt::DropAction action, int row, int column, QModelIndex parent)
- void fetchMore(QModelIndex parent)
- int flags(QModelIndex index)
- bool hasChildren(QModelIndex parent)
- QVariant headerData(int section, Qt::Orientation orientation, int role)
- QMimeData * mimeTypeData(QModelIndexList indexes)
- QStringList mimeTypeTypes(void)
- QModelIndex parent(QModelIndex index)
- int rowCount(QModelIndex parent)
- bool setData(QModelIndex idx, QVariant value, int role)
- void sort(int column, Qt::SortOrder order)
- int supportedDropActions(void)

99.21 QTreeWidget Class

C++ Reference : <http://doc.qt.io/qt-5/QTreeWidget.html>

Parameters : QWidget *

Parent Class : QTreeView

- void addTopLevelItem(QTreeWidgetItem *item)
- void closePersistentEditor(QTreeWidgetItem *item, int column)
- int columnCount(void)
- int currentColumn(void)
- QTreeWidgetItem *currentItem(void)
- void editItem(QTreeWidgetItem *item, int column)
- QTreeWidgetItem *headerItem(void)
- int indexOfTopLevelItem(QTreeWidgetItem *item)
- void insertTopLevelItem(int index, QTreeWidgetItem *item)
- QTreeWidgetItem *invisibleRootItem(void)
- bool isFirstItemColumnSpanned(QTreeWidgetItem *item)
- QTreeWidgetItem *itemAbove(QTreeWidgetItem *item)
- QTreeWidgetItem *itemAt(int x, int y)
- QTreeWidgetItem *itemBelow(QTreeWidgetItem *item)
- QWidget *itemWidget(QTreeWidgetItem *item, int column)
- void openPersistentEditor(QTreeWidgetItem *item, int column)
- void removeItemWidget(QTreeWidgetItem *item, int column)
- void setColumnCount(int columns)
- void setCurrentItem(QTreeWidgetItem * item, QItemSelectionModel::SelectionFlag column)
- void setFirstItemColumnSpanned(QTreeWidgetItem *item, bool span)
- void setHeaderItem(QTreeWidgetItem *item)
- void setHeaderLabel(QString)
- void setHeaderLabels(QStringList)
- void setItemWidget(QTreeWidgetItem *item, int column, QWidget * widget)
- int sortColumn(void)
- void sortItems(int column, Qt::SortOrder order)
- QTreeWidgetItem *takeTopLevelItem(int index)
- QTreeWidgetItem *topLevelItem(int index)
- int topLevelItemCount(void)
- QRect visualItemRect(QTreeWidgetItem *item)
- void setSelectionModel(QItemSelectionModel *selectionModel)

- void clear(void)
- void collapseItem(QTreeWidgetItem *item)
- void expandItem(QTreeWidgetItem *item)
- void scrollToItem(QTreeWidgetItem *item, QAbstractItemView::ScrollHint hint)
- void setcollapsedEvent(const char *)
- void setexpandedEvent(const char *)
- void setactivatedEvent(const char *)
- void setclickedEvent(const char *)
- void setdoubleClickedEvent(const char *)
- void setenteredEvent(const char *)
- void setpressedEvent(const char *)
- void setviewportEnteredEvent(const char *)
- void setCurrentItemChangedEvent(const char *)
- void setitemActivatedEvent(const char *)
- void setitemChangedEvent(const char *)
- void setitemClickedEvent(const char *)
- void setitemCollapsedEvent(const char *)
- void setitemDoubleClickedEvent(const char *)
- void setitemEnteredEvent(const char *)
- void setitemExpandedEvent(const char *)
- void setitemPressedEvent(const char *)
- void setitemSelectionChangedEvent(const char *)
- const char *getcollapsedEvent(void)
- const char *getexpandedEvent(void)
- const char *getactivatedEvent(void)
- const char *getclickedEvent(void)
- const char *getdoubleClickedEvent(void)
- const char *getenteredEvent(void)
- const char *getpressedEvent(void)
- const char *getviewportEnteredEvent(void)
- const char *getCurrentItemChangedEvent(void)
- const char *getitemActivatedEvent(void)
- const char *getitemChangedEvent(void)
- const char *getitemClickedEvent(void)
- const char *getitemCollapsedEvent(void)
- const char *getitemDoubleClickedEvent(void)

- `const char *getitemEnteredEvent(void)`
- `const char *getitemExpandedEvent(void)`
- `const char *getitemPressedEvent(void)`
- `const char *getitemSelectionChangedEvent(void)`

99.22 QTreeWidgetItem Class

C++ Reference : <http://doc.qt.io/qt-5/QTreeWidgetItem.html>

Parameters : void

- `void addChild(QTreeWidgetItem *child)`
- `QBrush background(int column)`
- `int checkState(int column)`
- `QTreeWidgetItem *child(int index)`
- `int childCount(void)`
- `int childIndicatorPolicy(void)`
- `QTreeWidgetItem *clone(void)`
- `int columnCount(void)`
- `QVariant data(int column, int role)`
- `int flags(void)`
- `QFont font(int column)`
- `QBrush foreground(int column)`
- `QIcon icon(int column)`
- `int indexOfChild(QTreeWidgetItem *child)`
- `void insertChild(int index, QTreeWidgetItem *child)`
- `bool isDisabled(void)`
- `bool isExpanded(void)`
- `bool isFirstColumnSpanned(void)`
- `bool isHidden(void)`
- `bool isSelected(void)`
- `QTreeWidgetItem *parent(void)`
- `void read(QDataStream)`
- `void removeChild(QTreeWidgetItem *child)`
- `void setBackground(int column, QBrush)`
- `void setCheckState(int column, Qt::CheckState state)`
- `void setChildIndicatorPolicy(QTreeWidgetItem::ChildIndicatorPolicy policy)`
- `void setData(int column, int role, QVariant)`

- void setDisabled(bool disabled)
- void setExpanded(bool expand)
- void setFirstColumnSpanned(bool span)
- void setFlags(Qt::ItemFlag flags)
- void setFont(int column, QFont)
- void setForeground(int column, QBrush)
- void setHidden(bool hide)
- void setIcon(int column, QIcon)
- void setSelected(bool select)
- void setSizeHint(int column, QSize)
- void setStatusTip(int column, QString)
- void setText(int column, QString)
- void setTextAlignment(int column, int alignment)
- void setToolTip(int column, QString)
- void setWhatsThis(int column, QString)
- QSize sizeHint(int column)
- void sortChildren(int column, Qt::SortOrder order)
- QString statusTip(int column)
- QTreeWidgetItem *takeChild(int index)
- QString text(int column)
- int textAlignment(int column)
- QString toolTip(int column)
- QTreeWidgetItem *treeWidget(void)
- int type(void)
- QString whatsThis(int column)
- void write(QDataStream)

99.23 QComboBox Class

C++ Reference : <http://doc.qt.io/qt-5/QComboBox.html>

Parameters : QWidget *

Parent Class : QWidget

- void addItem(QString,int)
- QCompleter *completer(void)
- int count(void)
- int currentIndex(void)

- QString currentText(void)
- bool duplicatesEnabled(void)
- int findData(QVariant, int role, Qt::MatchFlag flags)
- int findText(QString, Qt::MatchFlag flags)
- bool hasFrame(void)
- void hidePopup(void)
- QSize iconSize(void)
- void insertItem(int index, QString, QVariant)
- bool isEditable(void)
- QVariant itemData(int index, int role)
- QAbstractItemDelegate *itemDelegate(void)
- QIcon itemIcon(int index)
- QString itemText(int index)
- QLineEdit *lineEdit(void)
- int maxCount(void)
- int maxVisibleItems(void)
- int minimumContentsLength(void)
- QAbstractItemModel *model(void)
- int modelColumn(void)
- void removeItem(int index)
- QModelIndex rootModelIndex(void)
- void setCompleter(QCompleter *completer)
- void setDuplicatesEnabled(bool enable)
- void setEditable(bool editable)
- void setFrame(bool)
- void setIconSize(QSize)
- void setItemDelegate(QAbstractItemDelegate *delegate)
- void setItemIcon(int index, QIcon)
- void setItemText(int index, QString)
- void setLineEdit(QLineEdit *edit)
- void setMaxCount(int max)
- void setMaxVisibleItems(int maxItems)
- void setMinimumContentsLength(int characters)
- void setModel(QAbstractItemModel *model)
- void setModelColumn(int visibleColumn)
- void setRootModelIndex(QModelIndex)

- void setView(QAbstractItemView *itemView)
- void showPopup(void)
- QAbstractItemView *view(void)
- void clear(void)
- void clearEditText(void)
- void setCurrentIndex(int index)
- void setEditText(QString)
- void setactivatedEvent(const char *)
- void setcurrentIndexChangedEvent(const char *)
- void seteditTextChangedEvent(const char *)
- void sethighlightedEvent(const char *)
- const char *getactivatedEvent(void)
- const char *getcurrentIndexChangedEvent(void)
- const char *geteditTextChangedEvent(void)
- const char *gethighlightedEvent(void)

99.24 QMenuBar Class

C++ Reference : <http://doc.qt.io/qt-5/QMenuBar.html>

Parameters : QWidget *

Parent Class : QWidget

- QAction *actionAt(QPoint)
- QRect actionGeometry(QAction *act)
- QAction *activeAction(void)
- QAction *addAction(QString)
- QAction *addSeparator(void)
- void clear(void)
- QWidget *cornerWidget(Qt::Corner)
- QAction *insertSeparator(QAction *before)
- bool isDefaultUp(void)
- bool isNativeMenuBar(void)
- void setActiveAction(QAction *act)
- void setCornerWidget(QWidget *widget, Qt::Corner)
- void setNativeMenuBar(bool nativeMenuBar)

99.25 QMenu Class

C++ Reference : <http://doc.qt.io/qt-5/QMenu.html>

Parameters : QWidget *

Parent Class : QWidget

- QAction *actionAt(QPoint)
- QRect actionGeometry(QAction *act)
- QAction *activeAction(void)
- void addAction(QAction *)
- QMenu *addMenu(QString)
- QAction *addSeparator(void)
- void clear(void)
- QAction *defaultAction(void)
- QAction *exec(const QPoint &)
- QAction *exec_2(void)
- QAction *exec_3(const QPoint &, QAction *)
- void hideTearOffMenu(void)
- QIcon icon(void)
- QAction *insertMenu(QAction *before, QMenu *menu)
- QAction *insertSeparator(QAction *before)
- bool isEmpty(void)
- bool isTearOffEnabled(void)
- bool isTearOffMenuVisible(void)
- QAction *menuAction(void)
- void popup(QPoint, QAction *atAction)
- bool separatorsCollapsible(void)
- void setActiveAction(QAction *act)
- void setDefaultAction(QAction *act)
- void setIcon(QIcon)
- void setSeparatorsCollapsible(bool collapse)
- void setTearOffEnabled(bool)
- void setTitle(QString)
- QString title(void)

99.26 QToolBar Class

C++ Reference : <http://doc.qt.io/qt-5/QToolBar.html>

Parameters : QWidget *

Parent Class : QWidget

- QAction *actionAt(int x, int y)
- QAction *addAction(QString)
- QAction *addSeparator(void)
- QAction *addWidget(QWidget *widget)
- int allowedAreas(void)
- void clear(void)
- QSize iconSize(void)
- QAction *insertSeparator(QAction *before)
- QAction *insertWidget(QAction *before, QWidget *widget)
- bool isAreaAllowed(Qt::ToolBarArea area)
- bool isFloatable(void)
- bool isFloating(void)
- bool isMovable(void)
- int orientation(void)
- void setAllowedAreas(Qt::ToolBarArea areas)
- void setFloatable(bool floatable)
- void setMovable(bool movable)
- void setOrientation(Qt::Orientation orientation)
- QAction *toggleViewAction(void)
- int toolButtonStyle(void)
- QWidget *widgetForAction(QAction *action)
- void setIconSize(QSize)
- void setToolButtonStyle(Qt::ToolButtonStyle toolButtonStyle)

99.27 QMainWindow Class

C++ Reference : <http://doc.qt.io/qt-5/QMainWindow.html>

Parameters : void

Parent Class : QWidget

- void addDockWidget(Qt::DockWidgetArea area, QDockWidget *dockwidget, Qt::Orientation orientation)
- QToolBar *addToolBar(QString)
- void addToolBar_2(Qt::ToolBarArea area, QToolBar *toolbar)

- void addToolBarBreak(Qt::ToolBarArea)
- QWidget *centralWidget(void)
- int corner(Qt::Corner corner)
- QMenu *createPopupMenu(void)
- int dockOptions(void)
- int dockWidgetArea(QDockWidget *dockwidget)
- bool documentMode(void)
- QSize iconSize(void)
- void insertToolBar(QToolBar *before, QToolBar *toolbar)
- void insertToolBarBreak(QToolBar *before)
- bool isAnimated(void)
- bool isDockNestingEnabled(void)
- QMenuBar *menuBar(void)
- QWidget *menuWidget(void)
- void removeDockWidget(QDockWidget *dockwidget)
- void removeToolBar(QToolBar *toolbar)
- void removeToolBarBreak(QToolBar *before)
- bool restoreDockWidget(QDockWidget *dockwidget)
- bool restoreState(QByteArray state, int version)
- QByteArray saveState(int version)
- void setCentralWidget(QWidget *widget)
- void setCorner(Qt::Corner corner, Qt::DockWidgetArea area)
- void setDockOptions(QMainWindow::DockOption options)
- void setDocumentMode(bool enabled)
- void setIconSize(QSize)
- void setMenuBar(QMenuBar *menuBar)
- void setMenuWidget(QWidget *menuBar)
- void setStatusBar(QStatusBar *statusbar)
- void setTabPosition(Qt::DockWidgetArea areas, QTabWidget::TabPosition tabPosition)
- void setTabShape(QTabWidget::TabShape tabShape)
- void setToolButtonStyle(Qt::ToolButtonStyle toolButtonStyle)
- void setUnifiedTitleAndToolBarOnMac(bool set)
- void splitDockWidget(QDockWidget *first, QDockWidget *second, Qt::Orientation orientation)
- QStatusBar *statusBar(void)
- int tabPosition(Qt::DockWidgetArea area)
- int tabShape(void)

- void tabifyDockWidget(QDockWidget *first, QDockWidget *second)
- int toolBarArea(QToolBar *toolbar)
- bool toolBarBreak(QToolBar *toolbar)
- int toolButtonStyle(void)
- bool unifiedTitleAndToolBarOnMac(void)

99.28 QStatusBar Class

C++ Reference : <http://doc.qt.io/qt-5/QStatusBar.html>

Parameters : QWidget *

Parent Class : QWidget

- void addPermanentWidget(QWidget * widget, int stretch)
- void addWidget(QWidget * widget, int stretch)
- QString currentMessage(void)
- int insertPermanentWidget(int index, QWidget * widget, int stretch)
- int insertWidget(int index, QWidget * widget, int stretch)
- bool isSizeGripEnabled(void)
- void removeWidget(QWidget *widget)
- void setSizeGripEnabled(bool)
- void clearMessage(void)
- void showMessage(QString , int timeout)

99.29 QDockWidget Class

C++ Reference : <http://doc.qt.io/qt-5/QDockWidget.html>

Parameters : QWidget *parent, Qt::WindowType flag

Parent Class : QWidget

- int allowedAreas(void)
- int features(void)
- bool isAreaAllowed(Qt::DockWidgetArea area)
- bool isFloating(void)
- void setAllowedAreas(Qt::DockWidgetArea areas)
- void setFeatures(QDockWidget::DockWidgetFeature features)
- void setFloating(bool floating)
- void setTitleBarWidget(QWidget *widget)
- void setWidget(QWidget *widget)
- QWidget *titleBarWidget(void)

- QAction *toggleViewAction(void)
- QWidget *widget(void)
- void allowedAreasChanged(Qt::DockWidgetArea allowedAreas)
- void dockLocationChanged(Qt::DockWidgetArea area)
- void featuresChanged(QDockWidget::DockWidgetFeature features)
- void topLevelChanged(bool topLevel)
- void visibilityChanged(bool visible)
- void setallowedAreasChangedEvent(const char *)
- void setdockLocationChangedEvent(const char *)
- void setfeaturesChangedEvent(const char *)
- void settopLevelChangedEvent(const char *)
- void setvisibilityChangedEvent(const char *)
- const char *getallowedAreasChangedEvent(void)
- const char *getdockLocationChangedEvent(void)
- const char *getfeaturesChangedEvent(void)
- const char *gettopLevelChangedEvent(void)
- const char *getvisibilityChangedEvent(void)

99.30 QTabWidget Class

C++ Reference : <http://doc.qt.io/qt-5/QTabWidget.html>

Parameters : QWidget *parent

Parent Class : QWidget

- int addTab(QWidget *page, QString)
- void clear(void)
- QWidget *cornerWidget(Qt::Corner corner)
- int count(void)
- int currentIndex(void)
- QWidget *currentWidget(void)
- bool documentMode(void)
- int elideMode(void)
- QSize iconSize(void)
- int indexOf(QWidget *w)
- int insertTab(int index, QWidget *page, QString)
- bool isMovable(void)
- bool isTabEnabled(int index)

- void removeTab(int index)
- void setCornerWidget(QWidget *widget, Qt::Corner corner)
- void setDocumentMode(bool set)
- void setElideMode(Qt::TextElideMode)
- void setIconSize(QSize)
- void setMovable(bool movable)
- void setTabEnabled(int index, bool enable)
- void setTabIcon(int index, QIcon)
- void setTabText(int index, QString)
- void setTabToolTip(int index, QString)
- void setTabWhatsThis(int index, QString)
- void setTabsClosable(bool closeable)
- void setUsesScrollButtons(bool useButtons)
- QIcon tabIcon(int index)
- QString tabText(int index)
- QString tabToolTip(int index)
- QString tabWhatsThis(int index)
- bool tabsClosable(void)
- bool usesScrollButtons(void)
- QWidget *widget(int index)
- int heightForWidth(int width)
- QSize minimumSizeHint(void)
- QSize sizeHint(void)
- void setCurrentIndex(int index)
- void setCurrentWidget(QWidget *widget)
- void setcurrentChangedEvent(const char *)
- void settabCloseRequestedEvent(const char *)
- const char *getcurrentChangedEvent(void)
- const char *gettabCloseRequestedEvent(void)

99.31 QTableWidgetItem Class

C++ Reference : <http://doc.qt.io/qt-5/QTableWidgetItem.html>

Parameters : QString

- QBrush background(void)
- int checkState(void)

- QTableWidgetItem *clone(void)
- int column(void)
- QVariant data(int role)
- int flags(void)
- QFont font(void)
- QBrush foreground(void)
- QIcon icon(void)
- bool isSelected(void)
- void read(QDataStream)
- int row(void)
- void setBackground(QBrush)
- void setCheckState(Qt::CheckState state)
- void setData(int role, QVariant)
- void setFlags(Qt::ItemFlag flags)
- void setFont(QFont)
- void setForeground(QBrush)
- void setIcon(QIcon)
- void setSelected(bool select)
- void setSizeHint(QSize)
- void setStatusTip(QString)
- void setText(QString)
- void setTextAlignment(int alignment)
- void setToolTip(QString)
- void setWhatsThis(QString)
- QSize sizeHint(void)
- QString statusTip(void)
- QTableWidgetItem *tableWidget(void)
- QString text(void)
- int textAlignment(void)
- QString toolTip(void)
- int type(void)
- QString whatsThis(void)
- void write(QDataStream)

99.32 QFrame Class

C++ Reference : <http://doc.qt.io/qt-5/QFrame.html>

Parameters : QWidget *parent, Qt::WindowType flag

Parent Class : QWidget

- int frameShadow(void)
- int frameShape(void)
- int frameStyle(void)
- int frameWidth(void)
- int lineWidth(void)
- int midLineWidth(void)
- void setFrameRect(QRect)
- void setFrameShadow(QFrame::Shadow)
- void setFrameShape(QFrame::Shape)
- void setFrameStyle(int style)
- void setLineWidth(int)
- void setMidLineWidth(int)
- QSize sizeHint(void)

99.33 QFrame2 Class

C++ Reference : <http://doc.qt.io/qt-5/QFrame2.html>

Parameters : void

Parent Class : QFrame

99.34 QFrame3 Class

C++ Reference : <http://doc.qt.io/qt-5/QFrame3.html>

Parameters : QWidget *parent

Parent Class : QFrame

99.35 QAbstractScrollArea Class

C++ Reference : <http://doc.qt.io/qt-5/QAbstractScrollArea.html>

Parameters : QWidget *parent

Parent Class : QFrame

- QWidget *cornerWidget(void)

- `QScrollBar *horizontalScrollBar(void)`
- `int horizontalScrollBarPolicy(void)`
- `QSize maximumViewportSize(void)`
- `QWidgetList scrollBarWidgets(Qt::AlignmentFlag)`
- `void setCornerWidget(QWidget *widget)`
- `void setHorizontalScrollBar(QScrollBar *scrollBar)`
- `void setHorizontalScrollBarPolicy(Qt::ScrollBarPolicy)`
- `void setVerticalScrollBar(QScrollBar *scrollBar)`
- `void setVerticalScrollBarPolicy(Qt::ScrollBarPolicy)`
- `void setViewport(QWidget *widget)`
- `QScrollBar *verticalScrollBar(void)`
- `int verticalScrollBarPolicy(void)`
- `QWidget *viewport(void)`

99.36 QAbstractItemView Class

C++ Reference : <http://doc.qt.io/qt-5/QAbstractItemView.html>

Parameters : `QWidget *parent`

Parent Class : `QAbstractScrollArea`

- `bool alternatingRowColors(void)`
- `int autoScrollMargin(void)`
- `void closePersistentEditor(QModelIndex)`
- `QModelIndex currentIndex(void)`
- `int defaultDropAction(void)`
- `int dragDropMode(void)`
- `bool dragDropOverwriteMode(void)`
- `bool dragEnabled(void)`
- `int editTriggers(void)`
- `bool hasAutoScroll(void)`
- `int horizontalScrollMode(void)`
- `QSize iconSize(void)`
- `QModelIndex indexAt(QPoint)`
- `QWidget *indexWidget(QModelIndex)`
- `QAbstractItemDelegate *itemDelegate(QModelIndex)`
- `QAbstractItemDelegate *itemDelegateForColumn(int column)`
- `QAbstractItemDelegate *itemDelegateForRow(int row)`

- void keyboardSearch(QString)
- QAbstractItemModel *model(void)
- void openPersistentEditor(QModelIndex)
- QModelIndex rootIndex(void)
- void scrollTo(QModelIndex, QAbstractItemView::ScrollHint)
- int selectionBehavior(void)
- int selectionMode(void)
- QItemSelectionModel *selectionModel(void)
- void setAlternatingRowColors(bool enable)
- void setAutoScroll(bool enable)
- void setAutoScrollMargin(int margin)
- void setDefaultDropAction(Qt::DropAction dropAction)
- void setDragDropMode(QAbstractItemView::DragDropMode behavior)
- void setDragDropOverwriteMode(bool overwrite)
- void setDragEnabled(bool enable)
- void setDropIndicatorShown(bool enable)
- void setEditTriggers(QAbstractItemView::EditTrigger triggers)
- void setHorizontalScrollMode(QAbstractItemView::ScrollMode mode)
- void setIconSize(QSize)
- void setIndexWidget(QModelIndex, QWidget *widget)
- void setItemDelegate(QAbstractItemDelegate *delegate)
- void setItemDelegateForColumn(int column, QAbstractItemDelegate *delegate)
- void setItemDelegateForRow(int row, QAbstractItemDelegate *delegate)
- void setModel(QAbstractItemModel *model)
- void setSelectionBehavior(QAbstractItemView::SelectionBehavior behavior)
- void setSelectionMode(QAbstractItemView::SelectionMode mode)
- void setSelectionModel(QItemSelectionModel *selectionModel)
- void setTabKeyNavigation(bool enable)
- void setTextElideMode(Qt::TextElideMode mode)
- void setVerticalScrollMode(QAbstractItemView::ScrollMode mode)
- bool showDropIndicator(void)
- int sizeHintForColumn(int column)
- QSize sizeHintForIndex(QModelIndex)
- int sizeHintForRow(int row)
- bool tabKeyNavigation(void)
- int textElideMode(void)

- int verticalScrollMode(void)
- QRect visualRect(QModelIndex)
- void clearSelection(void)
- void edit(QModelIndex)
- void scrollToBottom(void)
- void scrollToTop(void)
- void setCurrentIndex(QModelIndex)
- void update(QModelIndex)

99.37 QTableView Class

C++ Reference : <http://doc.qt.io/qt-5/QTableView.html>

Parameters : QWidget *parent

Parent Class : QAbstractItemView

- void clearSpans(void)
- int columnAt(int x)
- int columnSpan(int row, int column)
- int columnViewportPosition(int column)
- int columnWidth(int column)
- Qt::PenStyle gridStyle(void)
- QHeaderView *horizontalHeader(void)
- bool isColumnHidden(int column)
- bool isCornerButtonEnabled(void)
- bool isRowHidden(int row)
- bool isSortingEnabled(void)
- int rowAt(int y)
- int rowHeight(int row)
- int rowSpan(int row, int column)
- int rowViewportPosition(int row)
- void setColumnHidden(int column, bool hide)
- void setColumnWidth(int column, int width)
- void setCornerButtonEnabled(bool enable)
- void setGridStyle(Qt::PenStyle style)
- void setHorizontalHeader(QHeaderView *header)
- void setRowHeight(int row, int height)
- void setRowHidden(int row, bool hide)

- void setSortingEnabled(bool enable)
- void setSpan(int row, int column, int rowSpanCount, int columnSpanCount)
- void setVerticalHeader(QHeaderView *header)
- void setWordWrap(bool on)
- bool showGrid(void)
- void sortByColumn(int column, Qt::SortOrder order)
- QHeaderView *verticalHeader(void)
- bool wordWrap(void)
- void hideColumn(int column)
- void hideRow(int row)
- void resizeColumnToContents(int column)
- void resizeColumnsToContents(void)
- void resizeRowToContents(int row)
- void resizeRowsToContents(void)
- void selectColumn(int column)
- void selectRow(int row)
- void setShowGrid(bool show)
- void showColumn(int column)
- void showRow(int row)

99.38 QTableWidget Class

C++ Reference : <http://doc.qt.io/qt-5/QTableWidget.html>

Parameters : QWidget *parent

Parent Class : QTableView

- QWidget *cellWidget(int row, int column)
- void closePersistentEditor(QTableWidgetItem *item)
- int column(QTableWidgetItem *item)
- int columnCount(void)
- int currentColumn(void)
- QTableWidgetItem *currentItem(void)
- int currentRow(void)
- void editItem(QTableWidgetItem *item)
- QTableWidgetItem *horizontalHeaderItem(int column)
- QTableWidgetItem *item(int row, int column)
- QTableWidgetItem *itemAt(int ax, int ay)

- QTableWidgetItem *itemPrototype(void)
- void openPersistentEditor(QTableWidgetItem *item)
- void removeCellWidget(int row, int column)
- int row(const QTableWidgetItem *item)
- int rowCount(void)
- QList<QTableWidgetItem *> selectedItems(void)
- QList<QTableWidgetItemSelectionRange> selectedRanges(void)
- void setCellWidget(int row, int column, QWidget *widget)
- void setColumnCount(int columns)
- void setCurrentCell(int row, int column)
- void setCurrentItem(QTableWidgetItem * item)
- void setHorizontalHeaderItem(int column, QTableWidgetItem *item)
- void setHorizontalHeaderLabels(QStringList)
- void setItem(int row, int column, QTableWidgetItem *item)
- void setItemPrototype(QTableWidgetItem *item)
- void setRowCount(int rows)
- void setVerticalHeaderItem(int row, QTableWidgetItem *item)
- void sortItems(int column, Qt::SortOrder order)
- QTableWidgetItem *takeHorizontalHeaderItem(int column)
- QTableWidgetItem *takeItem(int row, int column)
- QTableWidgetItem *takeVerticalHeaderItem(int row)
- QTableWidgetItem *verticalHeaderItem(int row)
- int visualColumn(int logicalColumn)
- QRect visualItemRect(QTableWidgetItem *)
- int visualRow(int logicalRow)
- void clear(void)
- void clearContents(void)
- void insertColumn(int column)
- void insertRow(int row)
- void removeColumn(int column)
- void removeRow(int row)
- void scrollToItem(QTableWidgetItem *item, QAbstractItemView::ScrollHint hint)
- void setcellActivatedEvent(const char *)
- void setcellChangedEvent(const char *)
- void setcellClickedEvent(const char *)
- void setcellDoubleClickedEvent(const char *)

- void setcellEnteredEvent(const char *)
- void setcellPressedEvent(const char *)
- void setcurrentCellChangedEvent(const char *)
- void setcurrentItemChangedEvent(const char *)
- void setitemActivatedEvent(const char *)
- void setitemChangedEvent(const char *)
- void setitemClickedEvent(const char *)
- void setitemDoubleClickedEvent(const char *)
- void setitemEnteredEvent(const char *)
- void setitemPressedEvent(const char *)
- void setitemSelectionChangedEvent(const char *)
- const char *getCellActivatedEvent(void)
- const char *getCellChangedEvent(void)
- const char *getCellClickedEvent(void)
- const char *getCellDoubleClickedEvent(void)
- const char *getCellEnteredEvent(void)
- const char *getCellPressedEvent(void)
- const char *getcurrentCellChangedEvent(void)
- const char *getcurrentItemChangedEvent(void)
- const char *getitemActivatedEvent(void)
- const char *getitemChangedEvent(void)
- const char *getitemClickedEvent(void)
- const char *getitemDoubleClickedEvent(void)
- const char *getitemEnteredEvent(void)
- const char *getitemPressedEvent(void)
- const char *getitemSelectionChangedEvent(void)

99.39 QProgressBar Class

C++ Reference : <http://doc.qt.io/qt-5/QProgressBar.html>

Parameters : QWidget *

Parent Class : QWidget

- int alignment(void)
- QString format(void)
- bool invertedAppearance(void)
- bool isTextVisible(void)

- `int maximum(void)`
- `int minimum(void)`
- `int orientation(void)`
- `void resetFormat(void)`
- `void setAlignment(Qt::AlignmentFlag alignment)`
- `void setFormat(QString)`
- `void setInvertedAppearance(bool invert)`
- `void setTextDirection(QProgressBar::Direction textDirection)`
- `void setTextVisible(bool visible)`
- `QString text(void)`
- `int textDirection(void)`
- `int value(void)`
- `void reset(void)`
- `void setMaximum(int maximum)`
- `void setMinimum(int minimum)`
- `void setOrientation(Qt::Orientation)`
- `void setRange(int minimum, int maximum)`
- `void setValue(int value)`
- `void setValueChangedEvent(const char *)`
- `const char *getValueChangedEvent(void)`

99.40 QSpinBox Class

C++ Reference : <http://doc.qt.io/qt-5/QSpinBox.html>

Parameters : `QWidget *`parent

Parent Class : `QWidget`

- `QString cleanText(void)`
- `int displayIntegerBase(void)`
- `int maximum(void)`
- `int minimum(void)`
- `QString prefix(void)`
- `void setDisplayIntegerBase(int base)`
- `void setMaximum(int max)`
- `void setMinimum(int min)`
- `void setPrefix(QString)`
- `void setRange(int minimum, int maximum)`

- void setSingleStep(int val)
- void setSuffix(QString)
- int singleStep(void)
- QString suffix(void)
- int value(void)
- void setValue(int val)
- void setValueChangedEvent(const char *)
- const char *getValueChangedEvent(void)

99.41 QAbstractSlider Class

C++ Reference : <http://doc.qt.io/qt-5/QAbstractSlider.html>

Parameters : QWidget *parent

Parent Class : QWidget

- bool invertedAppearance(void)
- bool invertedControls(void)
- bool isSliderDown(void)
- int maximum(void)
- int minimum(void)
- int orientation(void)
- int pageStep(void)
- void setInvertedAppearance(bool)
- void setInvertedControls(bool)
- void setMaximum(int)
- void setMinimum(int)
- void setPageStep(int)
- void setSingleStep(int)
- void setSliderDown(bool)
- void setSliderPosition(int)
- void setTracking(bool enable)
- int singleStep(void)
- int sliderPosition(void)
- void triggerAction(QAbstractSlider::SliderAction action)
- int value(void)
- void setOrientation(Qt::Orientation)
- void setRange(int min, int max)

- void setValue(int)

99.42 QSlider Class

C++ Reference : <http://doc.qt.io/qt-5/QSlider.html>

Parameters : QWidget *parent

Parent Class : QAbstractSlider

- void setTickInterval(int ti)
- void setTickPosition(QSlider::TickPosition position)
- int tickInterval(void)
- int tickPosition(void)
- QSize minimumSizeHint(void)
- QSize sizeHint(void)
- void setactionTriggeredEvent(const char *)
- void setrangeChangedEvent(const char *)
- void setsliderMovedEvent(const char *)
- void setsliderPressedEvent(const char *)
- void setsliderReleasedEvent(const char *)
- void setvalueChangedEvent(const char *)
- const char *getactionTriggeredEvent(void)
- const char *getrangeChangedEvent(void)
- const char *getsliderMovedEvent(void)
- const char *getsliderPressedEvent(void)
- const char *getsliderReleasedEvent(void)
- const char *getvalueChangedEvent(void)

99.43 QDateEdit Class

C++ Reference : <http://doc.qt.io/qt-5/QDateEdit.html>

Parameters : QWidget *parent

Parent Class : QDateTimeEdit

99.44 QDateTimeEdit Class

C++ Reference : <http://doc.qt.io/qt-5/QDateTimeEdit.html>

Parameters : QWidget *parent

Parent Class : QAbstractSpinBox

- `bool calendarPopup(void)`
- `QCalendarWidget *calendarWidget(void)`
- `void clearMaximumDate(void)`
- `void clearMaximumDateTime(void)`
- `void clearMaximumTime(void)`
- `void clearMinimumDate(void)`
- `void clearMinimumDateTime(void)`
- `void clearMinimumTime(void)`
- `int currentSection(void)`
- `int currentSectionIndex(void)`
- `QDate date(void)`
- `QDateTime dateTime(void)`
- `QString displayFormat(void)`
- `int displayedSections(void)`
- `QDate maximumDate(void)`
- `QDateTime maximumDateTime(void)`
- `QTime maximumTime(void)`
- `QDate minimumDate(void)`
- `QDateTime minimumDateTime(void)`
- `QTime minimumTime(void)`
- `int sectionAt(int index)`
- `int sectionCount(void)`
- `QString sectionText(QDateTimeEdit::Section section)`
- `void setCalendarPopup(bool enable)`
- `void setCalendarWidget(QCalendarWidget *calendarWidget)`
- `void setCurrentSection(QDateTimeEdit::Section section)`
- `void setCurrentSectionIndex(int index)`
- `void setDateRange(QDate, QDate)`
- `void setDateTimeRange(QDateTime, QDateTime)`
- `void setDisplayFormat(QString)`
- `void setMaximumDate(QDate)`
- `void setMaximumDateTime(QDateTime)`
- `void setMaximumTime(QTime)`
- `void setMinimumDate(QDate)`
- `void setMinimumDateTime(QDateTime)`
- `void setMinimumTime(QTime)`

- void setSelectedSection(QDateTimeEdit::Section section)
- void setTimeRange(QTime, QTime)
- void setTimeSpec(Qt::TimeSpec spec)
- QTime time(void)
- Qt::TimeSpec timeSpec(void)
- void setDate(QDate)
- void setDateTime(QDateTime)
- void setTime(QTime)

99.45 QAbstractSpinBox Class

C++ Reference : <http://doc.qt.io/qt-5/QAbstractSpinBox.html>

Parameters : QWidget *parent

Parent Class : QWidget

- int alignment(void)
- int buttonSymbols(void)
- int correctionMode(void)
- bool hasAcceptableInput(void)
- bool hasFrame(void)
- void interpretText(void)
- bool isAccelerated(void)
- bool keyboardTracking(void)
- void setAccelerated(bool on)
- void setAlignment(Qt::AlignmentFlag flag)
- void setButtonSymbols(QAbstractSpinBox::ButtonSymbols bs)
- void setCorrectionMode(QAbstractSpinBox::CorrectionMode cm)
- void setFrame(bool)
- void setReadOnly(bool r)
- void setSpecialValueText(QString)
- void setWrapping(bool w)
- QString specialValueText(void)
- void stepBy(int steps)
- QString text(void)
- bool wrapping(void)
- void clear(void)
- void selectAll(void)

- void stepDown(void)
- void stepUp(void)

99.46 QDial Class

C++ Reference : <http://doc.qt.io/qt-5/QDial.html>

Parameters : QWidget *parent

Parent Class : QAbstractSlider

- int notchSize(void)
- qreal notchTarget(void)
- bool notchesVisible(void)
- void setNotchTarget(double target)
- bool wrapping(void)
- QSize minimumSizeHint(void)
- QSize sizeHint(void)
- void setNotchesVisible(bool visible)
- void setWrapping(bool on)
- void setactionTriggeredEvent(const char *)
- void setrangeChangedEvent(const char *)
- void setsliderMovedEvent(const char *)
- void setsliderPressedEvent(const char *)
- void setsliderReleasedEvent(const char *)
- void setValueChangedEvent(const char *)
- const char *getactionTriggeredEvent(void)
- const char *getrangeChangedEvent(void)
- const char *getsliderMovedEvent(void)
- const char *getsliderPressedEvent(void)
- const char *getsliderReleasedEvent(void)
- const char *getvalueChangedEvent(void)

99.47 QWebView Class

C++ Reference : <http://doc.qt.io/archives/qt-5.5/qwebview.html>

Parameters : QWidget *parent

Parent Class : QWidget

- QWebHistory *history(void)
- QAction *pageAction(QWebPage::WebAction action)

- void setContent(QByteArray,QString,QString)
- void setHtml(QString,QString)
- void setPage(QWebPage *page)
- void setZoomFactor(qreal factor)
- QWebSettings *settings(void)
- void triggerPageAction(QWebPage::WebAction action, bool checked)
- QString url(void)
- qreal zoomFactor(void)
- void back(void)
- void forward(void)
- void print(QPrinter *printer)
- void reload(void)
- void stop(void)
- void setloadProgressEvent(const char *)
- void setloadStartedEvent(const char *)
- void setselectionChangedEvent(const char *)
- void seturlChangedEvent(const char *)
- const char *getloadFinishedEvent(void)
- const char *getloadProgressEvent(void)
- const char *getloadStartedEvent(void)
- const char *getselectionChangedEvent(void)
- const char *gettitleChangedEvent(void)
- const char *geturlChangedEvent(void)

99.48 QUrl Class

C++ Reference : <http://doc.qt.io/qt-5/QUrl.html>

Parameters : QString

- void clear(void)
- QString errorString(void)
- QString fileName(QUrl::ComponentFormattingOption options)
- QString fragment(QUrl::ComponentFormattingOption options)
- bool hasFragment(void)
- bool hasQuery(void)
- QString host(QUrl::ComponentFormattingOption options)
- bool isEmpty(void)

- bool isLocalFile(void)
- bool isParentOf(QUrl)
- bool isRelative(void)
- bool isValid(void)
- QString path(QUrl::ComponentFormattingOption options)
- int port(int defaultPort)
- QString query(QUrl::ComponentFormattingOption options)
- QUrl resolved(QUrl)
- QString scheme(void)
- void setAuthority(QString, QUrl::ParsingMode mode)
- void setFragment(QString, QUrl::ParsingMode mode)
- void setHost(QString, QUrl::ParsingMode mode)
- void setPassword(QString, QUrl::ParsingMode mode)
- void setPath(QString, QUrl::ParsingMode mode)
- void setPort(int port)
- void setQuery(QString, QUrl::ParsingMode mode)
- void setScheme(QString)
- void setUrl(QString, QUrl::ParsingMode parsingMode)
- void setUserInfo(QString, QUrl::ParsingMode mode)
- void setUsername(QString, QUrl::ParsingMode mode)
- void swap(QUrl)
- QString topLevelDomain(QUrl::ComponentFormattingOption options)
- QString userInfo(QUrl::ComponentFormattingOption options)
- QString userName(QUrl::ComponentFormattingOption options)
- QUrl fromLocalFile(QString)

99.49 QCheckBox Class

C++ Reference : <http://doc.qt.io/qt-5/QCheckBox.html>

Parameters : QWidget *parent

Parent Class : QAbstractButton

- int checkState(void)
- bool isTristate(void)
- void setCheckState(Qt::CheckState state)
- void setTristate(bool y)
- QSize minimumSizeHint(void)

- QSize sizeHint(void)
- void setstateChangedEvent(const char *)
- void setclickedEvent(const char *)
- void setpressedEvent(const char *)
- void setreleasedEvent(const char *)
- void settoggledEvent(const char *)
- const char *getstateChangedEvent(void)
- const char *getclickedEvent(void)
- const char *getpressedEvent(void)
- const char *getreleasedEvent(void)
- const char *gettoggledEvent(void)

99.50 QAbstractButton Class

C++ Reference : <http://doc.qt.io/qt-5/QAbstractButton.html>

Parameters : QWidget *parent

Parent Class : QWidget

- bool autoExclusive(void)
- bool autoRepeat(void)
- int autoRepeatDelay(void)
- int autoRepeatInterval(void)
- QButtonGroup *group(void)
- QIcon icon(void)
- QSize iconSize(void)
- bool isCheckable(void)
- bool isChecked(void)
- bool isDown(void)
- void setAutoExclusive(bool)
- void setAutoRepeat(bool)
- void setAutoRepeatDelay(int)
- void setAutoRepeatInterval(int)
- void setCheckable(bool)
- void setDown(bool)
- void setIcon(QIcon)
- void setShortcut(QKeySequence)
- void setText(QString)

- QKeySequence shortcut(void)
- QString text(void)
- void animateClick(int msec)
- void click(void)
- void setChecked(bool)
- void setIconSize(QSize)
- void toggle(void)

99.51 QRadioButton Class

C++ Reference : <http://doc.qt.io/qt-5/QRadioButton.html>

Parameters : QWidget *parent

Parent Class : QAbstractButton

- QSize minimumSizeHint(void)
- QSize sizeHint(void)
- void setClickedEvent(const char *)
- void setPressedEvent(const char *)
- void setReleasedEvent(const char *)
- void setToggledEvent(const char *)
- const char *getClickedEvent(void)
- const char *getPressedEvent(void)
- const char *getReleasedEvent(void)
- const char *getToggledEvent(void)

99.52 QButtonGroup Class

C++ Reference : <http://doc.qt.io/qt-5/QButtonGroup.html>

Parameters : QObject *parent

- void addButton(QAbstractButton *button, int id)
- QAbstractButton *button(int id)
- QAbstractButton *checkedButton(void)
- int checkedId(void)
- bool exclusive(void)
- int id(QAbstractButton *button)
- void removeButton(QAbstractButton *button)
- void setExclusive(bool)
- void setId(QAbstractButton *button, int id)

- void setbuttonClickedEvent(const char *)
- void setbuttonPressedEvent(const char *)
- void setbuttonReleasedEvent(const char *)
- const char *getbuttonClickedEvent(void)
- const char *getbuttonPressedEvent(void)
- const char *getbuttonReleasedEvent(void)

99.53 QMediaPlayer Class

C++ Reference : <http://doc.qt.io/qt-5/QMediaPlayer.html>

Parameters : void

- int bufferSize(void)
- QMediaContent currentMedia(void)
- QNetworkConfiguration currentNetworkConfiguration(void)
- int duration(void)
- int error(void)
- QString errorString(void)
- bool isAudioAvailable(void)
- bool isMuted(void)
- bool isSeekable(void)
- bool isVideoAvailable(void)
- QMediaContent media(void)
- int mediaStatus(void)
- QIODevice *mediaStream(void)
- qreal playbackRate(void)
- QMediaPlaylist *playlist(void)
- int position(void)
- void setVideoOutput(QVideoWidget *output)
- int volume(void)
- void pause(void)
- void play(void)
- void setMuted(bool muted)
- void setPlaylist(QMediaPlaylist *playlist)
- void setPosition(int position)
- void setVolume(int volume)
- void stop(void)

99.54 QMediaPlayer Class

C++ Reference : <http://doc.qt.io/qt-5/QMediaPlayer.html>

Parameters : void

- int currentIndex(void)
- QMediaContent currentMedia(void)
- int error(void)
- QString errorString(void)
- bool insertMedia(int pos, QMediaContent)
- bool isReadOnly(void)
- QMediaContent media(int index)
- int mediaCount(void)
- int nextIndex(int steps)
- int playbackMode(void)
- int previousIndex(int steps)
- bool save(QUrl, const char * format)
- void next(void) # In RingQt use : void movenext(void)
- void previous(void)
- void setCurrentIndex(int playlistPosition)
- void shuffle(void)

99.55 QVideoWidget Class

C++ Reference : <http://doc.qt.io/qt-5/QVideoWidget.html>

Parameters : QWidget *parent

Parent Class : QWidget

- int aspectRatioMode(void)
- int brightness(void)
- int contrast(void)
- int hue(void)
- bool isFullScreen(void)
- int saturation(void)
- void setAspectRatioMode(Qt::AspectRatioMode mode)
- void setBrightness(int brightness)
- void setContrast(int contrast)
- void setFullScreen(bool fullScreen)
- void setHue(int hue)

- void setSaturation(int saturation)
- void setbrightnessChangedEvent(const char *)
- void setcontrastChangedEvent(const char *)
- void setfullScreenChangedEvent(const char *)
- void sethueChangedEvent(const char *)
- void setsaturationChangedEvent(const char *)
- const char *getbrightnessChangedEvent(void)
- const char *getcontrastChangedEvent(void)
- const char *getfullScreenChangedEvent(void)
- const char *gethueChangedEvent(void)
- const char *getsaturationChangedEvent(void)

99.56 QAction Class

C++ Reference : <http://doc.qt.io/qt-5/QAction.html>

Parameters : QWidget *parent

- QActionGroup *actionGroup(void)
- void activate(QAction::ActionEvent event)
- bool autoRepeat(void)
- QVariant data(void)
- QFont font(void)
- QIcon icon(void)
- QString iconText(void)
- bool isCheckable(void)
- bool isChecked(void)
- bool isEnabled(void)
- bool isIconVisibleInMenu(void)
- bool isSeparator(void)
- bool isVisible(void)
- QMenu *menu(void)
- int menuRole(void)
- QWidget *parentWidget(void)
- int priority(void)
- void setActionGroup(QActionGroup *group)
- void setAutoRepeat(bool)
- void setCheckable(bool)

- void setData(QVariant)
- void setFont(QFont)
- void setIcon(QIcon)
- void setIconText(QString)
- void setIconVisibleInMenu(bool visible)
- void setMenu(QMenu *menu)
- void setMenuRole(QAction::MenuRole menuRole)
- void setPriority(QAction::Priority priority)
- void setSeparator(bool b)
- void setShortcut(QKeySequence)
- void setShortcutContext(Qt::ShortcutContext context)
- void setShortcuts(QKeySequence::StandardKey key)
- void setStatusTip(QString)
- void setText(QString)
- void setToolTip(QString)
- void setWhatsThis(QString)
- QKeySequence shortcut(void)
- int shortcutContext(void)
- bool showStatusText(QWidget *widget)
- QString statusTip(void)
- QString text(void)
- QString toolTip(void)
- QString whatsThis(void)
- void hover(void)
- void setChecked(bool)
- void setDisabled(bool)
- void setEnabled(bool)
- void setVisible(bool)
- void toggle(void)
- void trigger(void)
- void setClickEvent(const char *)
- const char *getClickEvent(void)

99.57 QEvent Class

C++ Reference : <http://doc.qt.io/qt-5/QEvent.html>

Parameters : QEvent::Type Type

- void accept(void)
- void ignore(void)
- bool isAccepted(void)
- void setAccepted(bool accepted)
- bool spontaneous(void)
- int type(void)

99.58 QMessageBox Class

C++ Reference : <http://doc.qt.io/qt-5/QMessageBox.html>

Parameters : QWidget *parent

Parent Class : QDialog

- void addButton(QAbstractButton *button, QMessageBox::ButtonRole role)
- QAbstractButton *button(QMessageBox::StandardButton which)
- int buttonRole(QAbstractButton *button)
- QAbstractButton *clickedButton(void)
- QPushButton *defaultButton(void)
- QString detailedText(void)
- QAbstractButton *escapeButton(void)
- QPixmap iconPixmap(void)
- QString informativeText(void)
- void open(QObject *receiver, const char *member)
- void removeButton(QAbstractButton *button)
- void setDefaultButton(QPushButton *button)
- void setDetailedText(QString)
- void setEscapeButton(QAbstractButton *button)
- void setIconPixmap(QPixmap)
- void setInformativeText(QString)
- void setStandardButtons(QMessageBox::StandardButton buttons)
- void setText(QString)
- void setTextFormat(Qt::TextFormat format)
- void setWindowModality(Qt::WindowModality windowModality)
- void setWindowTitle(QString)

- `int standardButton(QAbstractButton *button)`
- `int standardButtons(void)`
- `QString text(void)`
- `int textFormat(void)`
- `int exec(void)`
- `void about(QWidget *parent, QString,QString)`
- `void aboutQt(QWidget *parent, QString)`
- `int critical(QWidget * parent, QString , QString, int buttons, int defaultButton)`
- `int information(QWidget * parent, QString ,QString, int buttons,int defaultButton)`
- `int question(QWidget * parent,QString,QString, int buttons ,int defaultButton)`
- `int warning(QWidget *parent, QString,QString, int buttons,int defaultButton)`

99.59 QTimer Class

C++ Reference : <http://doc.qt.io/qt-5/QTimer.html>

Parameters : `QObject *parent`

- `int interval(void)`
- `bool isActive(void)`
- `bool isSingleShot(void)`
- `void setInterval(int msec)`
- `void setSingleShot(bool singleShot)`
- `int timerId(void)`
- `void start(void)`
- `void stop(void)`
- `void setTimeoutEvent(const char *)`
- `const char *getTimeoutEvent(void)`

99.60 QFileDialog Class

C++ Reference : <http://doc.qt.io/qt-5/QFileDialog.html>

Parameters : `QWidget *parent`

Parent Class : `QWidget`

- `int acceptMode(void)`
- `QString defaultSuffix(void)`
- `QDir directory(void)`
- `QUrl directoryUrl(void)`
- `int fileMode(void)`

- int filter(void)
- QStringList history(void)
- QFileIconProvider *iconProvider(void)
- QAbstractItemDelegate *itemDelegate(void)
- QString labelText(QFileDialog::DialogLabel label)
- QStringList mimeTypeFilters(void)
- QStringList nameFilters(void)
- void open(QObject *receiver, const char *member)
- int options(void)
- QAbstractProxyModel *proxyModel(void)
- bool restoreState(QByteArray)
- QByteArray saveState(void)
- void selectFile(QString)
- void selectMimeTypeFilter(QString)
- void selectNameFilter(QString)
- void selectUrl(QUrl)
- QStringList selectedFiles(void)
- QString selectedNameFilter(void)
- void setDefaultSuffix(QString)
- void setDirectory(QString)
- void setDirectoryUrl(QUrl)
- void setFileMode(QFileDialog::FileMode mode)
- void setFilter(QDir::Filter filters)
- void setHistory(QStringList)
- void setIconProvider(QFileIconProvider *provider)
- void setItemDelegate(QAbstractItemDelegate *delegate)
- void setLabelText(QFileDialog::DialogLabel label, QString)
- void setMimeTypeFilters(QStringList)
- void setNameFilter(QString)
- void setNameFilters(QStringList)
- void setOption(QFileDialog::Option option, bool)
- void setOptions(QFileDialog::Option options)
- void setProxyModel(QAbstractProxyModel *proxyModel)
- int viewMode(void)
- QString getExistingDirectory(QWidget *, QString, QString, QFileDialog::Option)
- QUrl getExistingDirectoryUrl(QWidget *, QString, QUrl, QFileDialog::Option, QStringList)

- QString getOpenFileName(QWidget *,QString,QString,QString)
- QUrl getSaveFileName(QWidget *,QString, QUrl,QString, QString *, QFileDialog::Option options,QStringList)

99.61 QPainter Class

C++ Reference : <http://doc.qt.io/qt-5/QtPainter.html>

Parameters : void

- QBrush background(void)
- int backgroundMode(void)
- bool begin(QPaintDevice *device)
- void beginNativePainting(void)
- QRect boundingRect(int x, int y, int w, int h, int flags, QString text)
- QBrush brush(void)
- QPoint brushOrigin(void)
- QRectF clipBoundingRect(void)
- QPainterPath clipPath(void)
- QRegion clipRegion(void)
- QTransform combinedTransform(void)
- int compositionMode(void)
- QPaintDevice *device(void)
- QTransform deviceTransform(void)
- void drawArc(int x, int y, int width, int height, int startAngle, int spanAngle)
- void drawChord(int x, int y, int width, int height, int startAngle, int spanAngle)
- void drawConvexPolygon(QPoint * points, int pointCount)
- void drawEllipse(int x, int y, int width, int height)
- void drawGlyphRun(QPointF position, QGlyphRun glyphs)
- void drawImage(int x, int y, QImage image)
- void drawLine(int x1, int y1, int x2, int y2)
- void drawLines(QLine * lines, int lineCount)
- void drawPath(QPainterPath path)
- void drawPicture(int x, int y, QPicture picture)
- void drawPie(int x, int y, int width, int height, int startAngle, int spanAngle)
- void drawPixmap(int x, int y, QPixmap)
- void drawPoints(QPointF * points, int pointCount)
- void drawRect(int x, int y, int width, int height)
- void drawRects(QRectF * rectangles, int rectCount)

- void drawRoundedRect(int x, int y, int w, int h, qreal xRadius, qreal yRadius, Qt::SizeMode mode)
- void drawStaticText(int left, int top, QStaticText staticText)
- void drawText(int x, int y, QString text)
- void drawTiledPixmap(int x, int y, int width, int height, QPixmap pixmap, int sx, int sy)
- bool end(void) # In RingQt use : bool endpaint(void)
- void endNativePainting(void)
- void eraseRect(int x, int y, int width, int height)
- void fillPath(QPainterPath path, QBrush brush)
- void fillRect(int x, int y, int width, int height, QBrush)
- QFont font(void)
- QFontInfo fontInfo(void)
- bool hasClipping(void)
- void initFrom(QWidget *widget)
- bool isActive(void)
- int layoutDirection(void)
- double opacity(void)
- QPaintEngine *paintEngine(void)
- QPen pen(void)
- int renderHints(void)
- void resetTransform(void)
- void restore(void)
- void rotate(qreal angle)
- void save(void)
- void scale(double sx, double sy)
- void setBackground(QBrush brush)
- void setBackgroundMode(Qt::BGMode mode)
- void setBrush(QBrush brush)
- void setBrushOrigin(int x, int y)
- void setClipPath(QPainterPath path, Qt::ClipOperation operation)
- void setClipRect(int x, int y, int width, int height, Qt::ClipOperation operation)
- void setClipRegion(QRegion region, Qt::ClipOperation operation)
- void setClipping(bool enable)
- void setCompositionMode(QPainter::CompositionMode mode)
- void setFont(QFont font)
- void setLayoutDirection(Qt::LayoutDirection direction)
- void setOpacity(qreal opacity)

- void setPen(QPen pen)
- void setRenderHint(QPainter::RenderHint hint, bool on)
- void setTransform(QTransform transform, bool combine)
- void setViewTransformEnabled(bool enable)
- void setViewport(int x, int y, int width, int height)
- void setWindow(int x, int y, int width, int height)
- void setWorldMatrixEnabled(bool enable)
- void setWorldTransform(QTransform matrix, bool combine)
- void shear(double sh, double sv)
- void strokePath(QPainterPath path, QPen pen)
- bool testRenderHint(QPainter::RenderHint hint)
- QTransform transform(void)
- void translate(double dx, double dy)
- bool viewTransformEnabled(void)
- QRect viewport(void)
- QRect window(void)
- bool worldMatrixEnabled(void)
- QTransform worldTransform(void)
- void drawPolygon(QPainter *pObject, Qt::FillRule fillRule)

99.62 QPainter2 Class

C++ Reference : <http://doc.qt.io/qt-5/QPainter2.html>

Parameters : QPaintDevice *

Parent Class : QPainter

99.63 QPicture Class

C++ Reference : <http://doc.qt.io/qt-5/QPicture.html>

Parameters : void

- QRect boundingRect(void)
- const char *data(void)
- bool isNull(void)
- bool load(QString, const char *format) # In RingQt use : bool loadfile(QString, const char *format)
- bool play(QPainter *painter)
- bool save(QString , const char *format)
- void setBoundingRect(QRect)

- `int size(void)`
- `void swap(QPicture)`

99.64 QPen Class

C++ Reference : <http://doc.qt.io/qt-5/QPen.html>

Parameters : void

- `QBrush brush(void)`
- `int capStyle(void)`
- `QColor color(void)`
- `double dashOffset(void)`
- `bool isCosmetic(void)`
- `bool isSolid(void)`
- `int joinStyle(void)`
- `double miterLimit(void)`
- `void setBrush(QBrush)`
- `void setCapStyle(Qt::PenCapStyle style)`
- `void setColor(QColor)`
- `void setCosmetic(bool cosmetic)`
- `void setDashOffset(double offset)`
- `void setJoinStyle(Qt::PenJoinStyle style)`
- `void setMiterLimit(double limit)`
- `void setStyle(Qt::PenStyle style)`
- `void setWidth(int width)`
- `void setWidthF(double width)`
- `int style(void)`
- `void swap(QPen)`
- `int width(void)`
- `double widthF(void)`

99.65 QColor Class

C++ Reference : <http://doc.qt.io/qt-5/QColor.html>

Parameters : void

- `int alpha(void)`
- `double alphaF(void)`
- `int black(void)`

- double blackF(void)
- int blue(void)
- double blueF(void)
- QColor convertTo(QColor::Spec colorSpec)
- int cyan(void)
- double cyanF(void)
- QColor darker(int factor)
- void getCmyk(int *c, int *m, int *y, int *k, int *a)
- void getCmykF(qreal *c, qreal *m, qreal *y, qreal *k, qreal *a)
- void getHsl(int *h, int *s, int *l, int *a)
- void getHslF(qreal *h, qreal *s, qreal *l, qreal *a)
- void getHsv(int *h, int *s, int *v, int *a)
- void getHsvF(qreal *h, qreal *s, qreal *v, qreal *a)
- void getRgb(int *r, int *g, int *b, int *a)
- void getRgbF(qreal *r, qreal *g, qreal *b, qreal *a)
- int green(void)
- double greenF(void)
- int hslHue(void)
- double hslHueF(void)
- int hslSaturation(void)
- double hslSaturationF(void)
- int hsvHue(void)
- double hsvHueF(void)
- int hsvSaturation(void)
- double hsvSaturationF(void)
- int hue(void)
- double hueF(void)
- bool isValid(void)
- QColor lighter(int factor)
- int lightness(void)
- double lightnessF(void)
- int magenta(void)
- double magentaF(void)
- QString name(void)
- int red(void)
- double redF(void)

- `QRgb rgb(void)`
- `QRgb rgba(void)`
- `int saturation(void)`
- `double saturationF(void)`
- `void setAlpha(int alpha)`
- `void setAlphaF(double alpha)`
- `void setBlue(int blue)`
- `void setBlueF(double blue)`
- `void setCmyk(int c, int m, int y, int k, int a)`
- `void setCmykF(double c, double m, double y, double k, double a)`
- `void setGreen(int green)`
- `void setGreenF(double green)`
- `void setHsl(int h, int s, int l, int a)`
- `void setHslF(double h, double s, double l, double a)`
- `void setHsv(int h, int s, int v, int a)`
- `void setHsvF(double h, double s, double v, double a)`
- `void setNamedColor(QString)`
- `void setRed(int red)`
- `void setRedF(double red)`
- `void setRgb(int r, int g, int b, int a)`
- `void setRgbF(double r, double g, double b, double a)`
- `void setRgba(QRgb rgba)`
- `int spec(void)`
- `QColor toCmyk(void)`
- `QColor toHsl(void)`
- `QColor toHsv(void)`
- `QColor toRgb(void)`
- `int value(void)`
- `double valueF(void)`
- `int yellow(void)`
- `double yellowF(void)`
- `QStringList colorNames(void)`
- `QColor fromCmyk(int c, int m, int y, int k, int a)`
- `QColor fromCmykF(double c, double m, double y, double k, double a)`
- `QColor fromHsl(int h, int s, int l, int a)`
- `QColor fromHslF(double h, double s, double l, double a)`

- QColor fromHsv(int h, int s, int v, int a)
- QColor fromHsvF(double h, double s, double v, double a)
- QColor fromRgb(int r, int g, int b, int a)
- QColor fromRgbF(double r, double g, double b, double a)
- QColor fromRgba(QRgb rgba)
- bool isValidColor(QString)

99.66 QPrinter Class

C++ Reference : <http://doc.qt.io/qt-5/QPrinter.html>

Parameters : QPrinter::PrinterMode

- bool abort(void)
- bool collateCopies(void)
- int colorMode(void)
- int copyCount(void)
- QString creator(void)
- QString docName(void)
- int duplex(void)
- bool fontEmbeddingEnabled(void)
- int fromPage(void)
- bool fullPage(void)
- bool isValid(void)
- QString outputFileName(void)
- int outputFormat(void)
- QRectF pageRect(QPrinter::Unit unit)
- QRectF paperRect(QPrinter::Unit unit)
- int paperSource(void)
- QPrintEngine *printEngine(void)
- QString printProgram(void)
- int printRange(void)
- QString printerName(void)
- QString printerSelectionOption(void)
- int printerState(void)
- int resolution(void)
- void setCollateCopies(bool collate)
- void setColorMode(QPrinter::ColorMode newColorMode)

- void setCopyCount(int count)
- void setCreator(QString)
- void setDocName(QString)
- void setDuplex(QPrinter::DuplexMode duplex)
- void setFontEmbeddingEnabled(bool enable)
- void setFromTo(int from, int to)
- void setFullPage(bool fp)
- void setOutputFileName(QString)
- void setOutputFormat(QPrinter::OutputFormat format)
- void setPrintProgram(QString)
- void setPrintRange(QPrinter::PrintRange)
- void setPrinterName(QString)
- void setPrinterSelectionOption(QString)
- void setResolution(int dpi)
- bool supportsMultipleCopies(void)
- int toPage(void)
- bool newPage(void)
- QPaintEngine *paintEngine(void)
- void setPageSizeMM(QSizeF)

99.67 QFont Class

C++ Reference : <http://doc.qt.io/qt-5/QFont.html>

Parameters : QString, int, int, bool

- bool bold(void)
- int capitalization(void)
- QString defaultFamily(void)
- bool exactMatch(void)
- QString family(void)
- bool fixedPitch(void)
- bool fromString(QString)
- int hintingPreference(void)
- bool isCopyOf(QFont)
- bool italic(void)
- bool kerning(void)
- QString key(void)

- QString lastResortFamily(void)
- QString lastResortFont(void)
- double letterSpacing(void)
- int letterSpacingType(void)
- bool overline(void)
- int pixelSize(void)
- int pointSize(void)
- double pointSizeF(void)
- bool rawMode(void)
- QString rawName(void)
- QFont resolve(QFont)
- void setBold(bool enable)
- void setCapitalization(QFont::Capitalization caps)
- void setFamily(QString)
- void setFixedPitch(bool enable)
- void setHintingPreference(QFont::HintingPreference hintingPreference)
- void setItalic(bool enable)
- void setKerning(bool enable)
- void setLetterSpacing(QFont::SpacingType type, double spacing)
- void setOverline(bool enable)
- void setPixelSize(int pixelSize)
- void setPointSize(int pointSize)
- void setPointSizeF(double pointSize)
- void setRawMode(bool enable)
- void setRawName(QString)
- void setStretch(int factor)
- void setStrikeOut(bool enable)
- void setStyle(QFont::Style style)
- void setStyleHint(QFont::StyleHint hint, QFont::StyleStrategy strategy)
- void setStyleName(QString)
- void setStyleStrategy(QFont::StyleStrategy s)
- void setUnderline(bool enable)
- void setWeight(int weight)
- void setWordSpacing(double spacing)
- int stretch(void)
- bool strikeOut(void)

- `int style(void)`
- `int styleHint(void)`
- `QString styleName(void)`
- `int styleStrategy(void)`
- `QString toString(void)`
- `bool underline(void)`
- `int weight(void)`
- `double wordSpacing(void)`
- `void insertSubstitution(QString,QString)`
- `void insertSubstitutions(QString,QStringList)`
- `QString substitute(QString)`
- `QStringList substitutes(QString)`
- `QStringList substitutions(void)`

99.68 QBrush Class

C++ Reference : <http://doc.qt.io/qt-5/QBrush.html>

Parameters : void

- `QColor color(void)`
- `QGradient *gradient(void)`
- `bool isOpaque(void)`
- `QMatrix matrix(void)`
- `void setColor(QColor)`
- `void setMatrix(QMatrix)`
- `void setStyle(Qt::BrushStyle style)`
- `void setTexture(QPixmap)`
- `void setTextureImage(QImage)`
- `void setTransform(QTransform)`
- `int style(void)`
- `void swap(QBrush)`
- `QPixmap texture(void)`
- `QImage textureImage(void)`
- `QTransform transform(void)`

99.69 QByteArray Class

C++ Reference : <http://doc.qt.io/qt-5/QByteArray.html>

Parameters : void

- QByteArray append(const char *str)
- char at(int i)
- int capacity(void)
- void chop(int n)
- void clear(void)
- const char *constData(void)
- bool contains(const char *str)
- int count(const char *str)
- const char *data(void)
- bool endsWith(const char *str)
- QByteArray fill(char ch, int size)
- int indexOf(const char *str, int from)
- QByteArray insert(int i, const char *str, int len)
- bool isEmpty(void)
- bool isNull(void)
- int lastIndexOf(const char *str, int from)
- QByteArray left(int len)
- QByteArray leftJustified(int width, char fill, bool truncate)
- int length(void)
- QByteArray mid(int pos, int len)
- QByteArray prepend(const char *str, int len)
- void push_back(const char *str)
- void push_front(const char *str)
- QByteArray remove(int pos, int len)
- QByteArray repeated(int times)
- QByteArray replace(int pos, int len, const char *after, int alen)
- void reserve(int size)
- void resize(int size)
- QByteArray right(int len)
- QByteArray rightJustified(int width, char fill, bool truncate)
- QByteArray setNum(int n, int base)
- QByteArray setRawData(const char *data, uint size)

- QByteArray simplified(void)
- int size(void)
- void squeeze(void)
- bool startsWith(const char *str)
- void swap(QByteArray other)
- QByteArray toBase64(void)
- double toDouble(bool * ok)
- float toFloat(bool * ok)
- QByteArray toHex(void)
- int toInt(bool *ok, int base)
- long toLong(bool *ok, int base)
- qlonglong toLongLong(bool *ok, int base)
- QByteArray toLower(void)
- QByteArray toPercentEncoding(QByteArray, QByteArray, char percent)
- short toShort(bool *ok, int base)
- int toUInt(bool *ok, int base)
- int toULong(bool *ok, int base)
- int toULongLong(bool * ok, int base)
- int toUShort(bool * ok, int base)
- QByteArray toUpper(void)
- QByteArray trimmed(void)
- void truncate(int pos)
- QByteArray fromBase64(QByteArray)
- QByteArray fromHex(QByteArray)
- QByteArray fromPercentEncoding(QByteArray, char percent)
- QByteArray fromRawData(const char *data, int size)
- QByteArray number(int n, int base)

99.70 QIODevice Class

C++ Reference : <http://doc.qt.io/qt-5/QIODevice.html>

Parameters : void

- QString errorString(void)
- bool getChar(char *c)
- bool isOpen(void)
- bool isReadable(void)

- bool isTextModeEnabled(void)
- bool isWritable(void)
- int openMode(void)
- int peek(char *data, int maxSize)
- int read(char *data, int maxSize)
- int readLine(char *data, int maxSize)
- void ungetChar(char c)
- int write(const char *data, int maxSize)
- bool atEnd(void)
- bool canReadLine(void)
- void close(void)
- bool open(QIODevice::OpenMode flags)
- qint64 pos(void)
- bool seek(qint64 pos)
- qint64 size(void)
- void setaboutToCloseEvent(const char *)
- void setbytesWrittenEvent(const char *)
- void setreadChannelFinishedEvent(const char *)
- void setreadyReadEvent(const char *)
- const char *getaboutToCloseEvent(void)
- const char *getbytesWrittenEvent(void)
- const char *getreadChannelFinishedEvent(void)
- const char *getreadyReadEvent(void)

99.71 QAbstractSocket Class

C++ Reference : <http://doc.qt.io/qt-5/QAbstractSocket.html>

Parameters : void

Parent Class : QIODevice

- void abort(void)
- bool bind(QHostAddress address, int port, QAbstractSocket::BindFlag mode)
- void connectToHost(QString hostName, int port, QIODevice::OpenModeFlag openMode, QAbstractSocket::NetworkLayerProtocol protocol)
- void disconnectFromHost(void)
- int error(void)
- bool flush(void)
- bool isValid(void)

- QHostAddress localAddress(void)
- int localPort(void)
- int pauseMode(void)
- QHostAddress peerAddress(void)
- QString peerName(void)
- int peerPort(void)
- QNetworkProxy proxy(void)
- int readBufferSize(void)
- void resume(void)
- void setPauseMode(QAbstractSocket::PauseMode pauseMode)
- void setProxy(QNetworkProxy networkProxy)
- void setReadBufferSize(int size)
- bool setSocketDescriptor(qintptr socketDescriptor, QAbstractSocket::SocketState socketState, QIODevice::OpenModeFlag openMode)
- void setSocketOption(QAbstractSocket::SocketOption option, QVariant value)
- int *socketDescriptor(void)
- QVariant socketOption(QAbstractSocket::SocketOption option)
- int socketType(void)
- int state(void)
- bool waitForConnected(int msec)
- bool waitForDisconnected(int msec)
- bool atEnd(void)
- int bytesAvailable(void)
- int bytesToWrite(void)
- bool canReadLine(void)
- void close(void)
- bool isSequential(void)
- bool waitForBytesWritten(int msec)
- bool waitForReadyRead(int msec)
- void setconnectedEvent(const char *)
- void setdisconnectedEvent(const char *)
- void seterrorEvent(const char *)
- void sethostFoundEvent(const char *)
- void setproxyAuthenticationRequiredEvent(const char *)
- void setstateChangedEvent(const char *)
- const char *getconnectedEvent(void)

- `const char *getDisconnectedEvent(void)`
- `const char *getErrorEvent(void)`
- `const char *getHostFoundEvent(void)`
- `const char *getProxyAuthenticationRequiredEvent(void)`
- `const char *getStateChangedEvent(void)`

99.72 QNetworkProxy Class

C++ Reference : <http://doc.qt.io/qt-5/QNetworkProxy.html>

Parameters : void

- `int capabilities(void)`
- `bool hasRawHeader(QByteArray headerName)`
- `QVariant header(QNetworkRequest::KnownHeaders header)`
- `QString hostName(void)`
- `bool isCachingProxy(void)`
- `bool isTransparentProxy(void)`
- `QString password(void)`
- `int port(void)`
- `QByteArray rawHeader(QByteArray headerName)`
- `void setCapabilities(QNetworkProxy::Capability capabilities)`
- `void setHeader(QNetworkRequest::KnownHeaders header, QVariant value)`
- `void setHostName(QString hostName)`
- `void setPassword(QString password)`
- `void setPort(int port)`
- `void setRawHeader(QByteArray headerName, QByteArray headerValue)`
- `void setType(QNetworkProxy::ProxyType type)`
- `void setUser(QString user)`
- `void swap(QNetworkProxy other)`
- `int type(void)`
- `QString user(void)`
- `QNetworkProxy applicationProxy(void)`
- `void setApplicationProxy(QNetworkProxy networkProxy)`

99.73 QTcpSocket Class

C++ Reference : <http://doc.qt.io/qt-5/QTcpSocket.html>

Parameters : QObject *

Parent Class : QAbstractSocket

- void setconnectedEvent(const char *)
- void setdisconnectedEvent(const char *)
- void seterrorEvent(const char *)
- void sethostFoundEvent(const char *)
- void setproxyAuthenticationRequiredEvent(const char *)
- void setstateChangedEvent(const char *)
- void setaboutToCloseEvent(const char *)
- void setbytesWrittenEvent(const char *)
- void setreadChannelFinishedEvent(const char *)
- void setreadyReadEvent(const char *)
- const char *getconnectedEvent(void)
- const char *getdisconnectedEvent(void)
- const char *geterrorEvent(void)
- const char *gethostFoundEvent(void)
- const char *getproxyAuthenticationRequiredEvent(void)
- const char *getstateChangedEvent(void)
- const char *getaboutToCloseEvent(void)
- const char *getbytesWrittenEvent(void)
- const char *getreadChannelFinishedEvent(void)
- const char *getreadyReadEvent(void)

99.74 QTcpServer Class

C++ Reference : <http://doc.qt.io/qt-5/QTcpServer.html>

Parameters : QWidget *

- void close(void)
- QString errorString(void)
- bool hasPendingConnections(void)
- bool isListening(void)
- bool listen(QHostAddress, int port)
- int maxPendingConnections(void)
- QTcpSocket *nextPendingConnection(void)

- void pauseAccepting(void)
- QNetworkProxy proxy(void)
- void resumeAccepting(void)
- QHostAddress serverAddress(void)
- int serverError(void)
- int serverPort(void)
- void setMaxPendingConnections(int numConnections)
- void setProxy(QNetworkProxy)
- bool setSocketDescriptor(qintptr socketDescriptor)
- int *socketDescriptor(void)
- bool waitForNewConnection(int msec, bool *timedOut)
- void setacceptErrorEvent(const char *)
- void setnewConnectionEvent(const char *)
- const char *getacceptErrorEvent(void)
- const char *getnewConnectionEvent(void)

99.75 QHostAddress Class

C++ Reference : <http://doc.qt.io/qt-5/QHostAddress.html>

Parameters : void

- void clear(void)
- bool isInSubnet(QHostAddress, int netmask)
- bool isNull(void)
- int protocol(void)
- QString scopeId(void)
- bool setAddress(QString)
- int toIPv4Address(void)
- Q_IPV6ADDR toIPv6Address(void)
- QString toString(void)

99.76 QHostInfo Class

C++ Reference : <http://doc.qt.io/qt-5/QHostInfo.html>

Parameters : void

- int error(void)
- QString errorString(void)
- QString hostName(void)

- int lookupId(void)
- void setError(QHostInfo::HostInfoError error)
- void setErrorString(QString)
- void setHostName(QString)
- void setLookupId(int id)
- void abortHostLookup(int id)
- QHostInfo fromName(QString)
- QString localDomainName(void)
- QString localHostName(void)
- int lookupHost(QString, QObject *receiver, const char *member)

99.77 QFileInfo Class

C++ Reference : <http://doc.qt.io/qt-5/QFileInfo.html>

Parameters : void

- QDir absoluteDir(void)
- QString absoluteFilePath(void)
- QString absolutePath(void)
- QString baseName(void)
- QString bundleName(void)
- bool caching(void)
- QString canonicalFilePath(void)
- QString canonicalPath(void)
- QString completeBaseName(void)
- QString completeSuffix(void)
- QDateTime created(void)
- QDir dir(void)
- bool exists(void)
- QString fileName(void)
- QString filePath(void)
- QString group(void)
- int groupId(void)
- bool isAbsolute(void)
- bool isBundle(void)
- bool isDir(void)
- bool isExecutable(void)

- bool isFile(void)
- bool isHidden(void)
- bool isNativePath(void)
- bool isReadable(void)
- bool isRelative(void)
- bool isRoot(void)
- bool isSymLink(void)
- bool isWritable(void)
- QDateTime lastModified(void)
- QDateTime lastRead(void)
- bool makeAbsolute(void)
- QString owner(void)
- uint ownerId(void)
- QString path(void)
- bool permission(QFileDevice::Permission permissions)
- int permissions(void)
- void refresh(void)
- void setCaching(bool enable)
- void setFile(QString)
- int size(void)
- QString suffix(void)
- void swap(QFileInfo)
- QString symLinkTarget(void)

99.78 QDirModel Class

C++ Reference : <http://doc.qt.io/qt-5/QDirModel.html>

Parameters : void

- QIcon fileIcon(QModelIndex)
- QFileInfo fileInfo(QModelIndex)
- QString fileName(QModelIndex)
- QString filePath(QModelIndex)
- int filter(void)
- QFileIconProvider *iconProvider(void)
- QModelIndex index(QString path, int column)
- bool isDir(QModelIndex)

- bool isReadOnly(void)
- bool lazyChildCount(void)
- QModelIndex mkdir(QModelIndex parent, QString name)
- QStringList nameFilters(void)
- bool remove(QModelIndex index)
- bool resolveSymlinks(void)
- bool rmdir(QModelIndex index)
- void setFilter(QDir::Filter filters)
- void setIconProvider(QFileIconProvider *provider)
- void setLazyChildCount(bool enable)
- void setNameFilters(QStringList filters)
- void setReadOnly(bool enable)
- void setResolveSymlinks(bool enable)
- void setSorting(QDir::SortFlag sort)
- int sorting(void)
- void refresh(QModelIndex parent)

99.79 QFontDialog Class

C++ Reference : <http://doc.qt.io/qt-5/QFontDialog.html>

Parameters : void

Parent Class : QDialog

- QFont currentFont(void)
- void open(QObject *receiver, const char *member)
- int options(void)
- QFont selectedFont(void)
- void setCurrentFont(QFont)
- void setOption(QFontDialog::FontDialogOption option, bool on)
- void setOptions(QFontDialog::FontDialogOption options)
- bool testOption(QFontDialog::FontDialogOption option)
- int getfont(void)

99.80 QDialog Class

C++ Reference : <http://doc.qt.io/qt-5/QDialog.html>

Parameters : QWidget *parent

Parent Class : QWidget

- bool isSizeGripEnabled(void)
- int result(void)
- void setModal(bool modal)
- void setResult(int i)
- void setSizeGripEnabled(bool)
- void accept(void)
- void done(int r) # In RingQt use : void donedialog(int r)
- int exec(void)
- void open(void)
- void reject(void)

99.81 QTextCursor Class

C++ Reference : <http://doc.qt.io/qt-5/QTextCursor.html>

Parameters : void

- int anchor(void)
- bool atBlockEnd(void)
- bool atBlockStart(void)
- bool atEnd(void)
- bool atStart(void)
- void beginEditBlock(void)
- QTextBlock block(void)
- QTextCharFormat blockCharFormat(void)
- QTextBlockFormat blockFormat(void)
- int blockNumber(void)
- QTextCharFormat charFormat(void)
- void clearSelection(void)
- int columnNumber(void)
- QTextList *createList(QTextListFormat)
- QTextFrame *currentFrame(void)
- QTextList *currentList(void)
- QTextTable *currentTable(void)
- void deleteChar(void)
- void deletePreviousChar(void)
- QTextDocument *document(void)
- void endEditBlock(void)

- bool hasComplexSelection(void)
- bool hasSelection(void)
- void insertBlock(void)
- void insertFragment(QTextDocumentFragment)
- QTextFrame *insertFrame(QTextFrameFormat)
- void insertHtml(QString)
- void insertImage(QTextImageFormat)
- QTextList *insertList(QTextListFormat)
- QTextTable *insertTable(int rows, int columns, QTextTableFormat)
- void insertText(QString)
- void insertText_2(QString, QTextCharFormat)
- bool isCopyOf(QTextCursor)
- bool isNull(void)
- void joinPreviousEditBlock(void)
- bool keepPositionOnInsert(void)
- void mergeBlockCharFormat(QTextCharFormat)
- void mergeBlockFormat(QTextBlockFormat)
- void mergeCharFormat(QTextCharFormat)
- bool movePosition(QTextCursor::MoveOperation operation, QTextCursor::MoveMode mode, int n)
- int position(void)
- int positionInBlock(void)
- void removeSelectedText(void)
- void select(QTextCursor::SelectionType selection)
- void selectedTableCells(int *firstRow, int *numRows, int *firstColumn, int *numColumns)
- QString selectedText(void)
- QTextDocumentFragment selection(void)
- int selectionEnd(void)
- int selectionStart(void)
- void setBlockCharFormat(QTextCharFormat)
- void setBlockFormat(QTextBlockFormat)
- void setCharFormat(QTextCharFormat)
- void setKeepPositionOnInsert(bool b)
- void setPosition(int pos, QTextCursor::MoveMode m)
- void setVerticalMovementX(int x)
- void setVisualNavigation(bool b)
- int verticalMovementX(void)

- bool visualNavigation(void)

99.82 QColorDialog Class

C++ Reference : <http://doc.qt.io/qt-5/QColorDialog.html>

Parameters : void

Parent Class : QDialog

- QColor currentColor(void)
- void open(void)
- int options(void)
- QColor selectedColor(void)
- void setCurrentColor(QColor)
- void setOption(QColorDialog::ColorDialogOption option, bool on)
- void setOptions(QColorDialog::ColorDialogOption options)
- bool testOption(QColorDialog::ColorDialogOption option)
- QColor customColor(int index)
- int customCount(void)
- void setCustomColor(int index, int color)
- void setStandardColor(int index, int color)
- int getcolor(void)

99.83 QStringList Class

C++ Reference : <http://doc.qt.io/qt-5/QStringList.html>

Parameters : void

- QString join(QString)
- void sort(void)
- int removeDuplicates(void)
- QStringList filter(QString, Qt::CaseSensitivity)
- QStringList replaceInStrings(QString,QString, Qt::CaseSensitivity)
- void append(QString)
- QString at(int)
- QString back(void)
- void clear(void)
- bool contains(QString)
- int count(void)
- bool empty(void)

- bool endsWith(QString)
- QString first(void)
- QString front(void)
- int indexOf(QString, int)
- void insert(int, QString)
- bool isEmpty(void)
- QString last(void)
- int lastIndexOf(QString,int)
- int length(void)
- void move(int,int)
- void pop_back(void)
- void pop_front(void)
- void prepend(QString)
- void push_back(QString)
- void push_front(QString)
- int removeAll(QString)
- void removeAt(int)
- void removeFirst(void)
- void removeLast(void)
- bool removeOne(QString)
- void replace(int,QString)
- void reserve(int)
- int size(void)
- bool startsWith(QString)
- void swap(int,int)
- QString takeAt(int)
- QString takeFirst(void)
- QString takeLast(void)
- QString value(int)

99.84 QKeySequence Class

C++ Reference : <http://doc.qt.io/qt-5/QKeySequence.html>

Parameters : QString

99.85 QLCDNumber Class

C++ Reference : <http://doc.qt.io/qt-5/QLCDNumber.html>

Parameters : QWidget *

Parent Class : QFrame

- bool checkOverflow(double num)
- int digitCount(void)
- int intValue(void)
- int mode(void)
- int segmentStyle(void)
- void setDigitCount(int numDigits)
- void setMode(QLCDNumber::Mode)
- void setSegmentStyle(QLCDNumber::SegmentStyle)
- bool smallDecimalPoint(void)
- double value(void)
- void display(double)
- void setBinMode(void)
- void setDecMode(void)
- void setHexMode(void)
- void setOctMode(void)
- void setSmallDecimalPoint(bool)

99.86 QInputDialog Class

C++ Reference : <http://doc.qt.io/qt-5/QInputDialog.html>

Parameters : QWidget *

Parent Class : QDialog

- QString cancelButtonText(void)
- QStringList comboBoxItems(void)
- int doubleDecimals(void)
- double doubleMaximum(void)
- double doubleMinimum(void)
- double doubleValue(void)
- int inputMode(void)
- int intMaximum(void)
- int intMinimum(void)
- int intStep(void)

- `int intValue(void)`
- `bool isComboBoxEditable(void)`
- `QString labelText(void)`
- `QString okButtonText(void)`
- `void open(QObject *receiver, const char *member)`
- `int options(void)`
- `void setCancelButtonText(QString)`
- `void setComboBoxEditable(bool editable)`
- `void setComboBoxItems(QStringList)`
- `void setDoubleDecimals(int decimals)`
- `void setDoubleMaximum(double max)`
- `void setDoubleMinimum(double min)`
- `void setDoubleRange(double min, double max)`
- `void setDoubleValue(double value)`
- `void setInputMode(QInputDialog::InputMode mode)`
- `void setIntMaximum(int max)`
- `void setIntMinimum(int min)`
- `void setIntRange(int min, int max)`
- `void setIntStep(int step)`
- `void setIntValue(int value)`
- `void setLabelText(QString)`
- `void setOkButtonText(QString)`
- `void setOption(QInputDialog::InputDialogOption option, bool on)`
- `void setOptions(QInputDialog::InputDialogOption options)`
- `void setTextEchoMode(QLineEdit::EchoMode mode)`
- `void setTextValue(QString)`
- `bool testOption(QInputDialog::InputDialogOption option)`
- `int textEchoMode(void)`
- `QString textValue(void)`
- `double getDouble(QWidget *parent, QString, QString, double value, double min, double max , int decimals, bool *ok, Qt::WindowType flags)`
- `int getInt(QWidget *parent, QString, QString, int value, int min, int max, int step, bool *ok, Qt::WindowType flags)`

99.87 QAllEvents Class

Parameters : QWidget *

Parent Class : QWidget

- void accept(void)
- void ignore(void)
- int getKeyCode(void)
- QString getKeyText(void)
- int getModifiers(void)
- int getx(void)
- int gety(void)
- int getglobalx(void)
- int getglobaly(void)
- int getbutton(void)
- int getbuttons(void)
- void setKeyPressEvent(const char *cStr)
- void setMouseButtonPressEvent(const char *cStr)
- void setMouseButtonReleaseEvent(const char *cStr)
- void setMouseButtonDbClickEvent(const char *cStr)
- void setMouseMoveEvent(const char *cStr)
- void setCloseEvent(const char *cStr)
- void setContextMenuEvent(const char *cStr)
- void setDragEnterEvent(const char *cStr)
- void setDragLeaveEvent(const char *cStr)
- void setDragMoveEvent(const char *cStr)
- void setDropEvent(const char *cStr)
- void setEnterEvent(const char *cStr)
- void setFocusInEvent(const char *cStr)
- void setFocusOutEvent(const char *cStr)
- void setKeyReleaseEvent(const char *cStr)
- void setLeaveEvent(const char *cStr)
- void setNonClientAreaMouseButtonDbClickEvent(const char *cStr)
- void setNonClientAreaMouseButtonPressEvent(const char *cStr)
- void setNonClientAreaMouseButtonReleaseEvent(const char *cStr)
- void setNonClientAreaMouseMoveEvent(const char *cStr)
- void setMoveEvent(const char *cStr)

- void setResizeEvent(const char *cStr)
- void setWindowActivateEvent(const char *cStr)
- void setWindowBlockedEvent(const char *cStr)
- void setWindowDeactivateEvent(const char *cStr)
- void setWindowStateChangeEvent(const char *cStr)
- void setWindowUnblockedEvent(const char *cStr)
- void setPaintEvent(const char *cStr)
- const char *getKeyPressEvent(void)
- const char *getMouseButtonPressEvent(void)
- const char *getMouseButtonReleaseEvent(void)
- const char *getMouseButtonDbClickEvent(void)
- const char *getMouseMoveEvent(void)
- const char *getCloseEvent(void)
- const char *getContextMenuEvent(void)
- const char *getDragEnterEvent(void)
- const char *getDragLeaveEvent(void)
- const char *getDragMoveEvent(void)
- const char *getDropEvent(void)
- const char *getEnterEvent(void)
- const char *getFocusInEvent(void)
- const char *getFocusOutEvent(void)
- const char *getKeyReleaseEvent(void)
- const char *getLeaveEvent(void)
- const char *getNonClientAreaMouseButtonDbClickEvent(void)
- const char *getNonClientAreaMouseButtonPressEvent(void)
- const char *getNonClientAreaMouseButtonReleaseEvent(void)
- const char *getNonClientAreaMouseMoveEvent(void)
- const char *getMoveEvent(void)
- const char *getResizeEvent(void)
- const char *getWindowActivateEvent(void)
- const char *getWindowBlockedEvent(void)
- const char *getWindowDeactivateEvent(void)
- const char *getWindowStateChangeEvent(void)
- const char *getWindowUnblockedEvent(void)
- const char *getPaintEvent(void)
- void setEventOutput(bool x)

- `QObject *getParentObject(void)`
- `QWidget *getParentWidget(void)`
- `void setKeyPressFunc(const char *cStr)`
- `void setMouseButtonPressFunc(const char *cStr)`
- `void setMouseButtonReleaseFunc(const char *cStr)`
- `void setMouseButtonDbClickFunc(const char *cStr)`
- `void setMouseMoveFunc(const char *cStr)`
- `void setCloseFunc(const char *cStr)`
- `void setContextMenuFunc(const char *cStr)`
- `void setDragEnterFunc(const char *cStr)`
- `void setDragLeaveFunc(const char *cStr)`
- `void setDragMoveFunc(const char *cStr)`
- `void setDropFunc(const char *cStr)`
- `void setEnterFunc(const char *cStr)`
- `void setFocusInFunc(const char *cStr)`
- `void setFocusOutFunc(const char *cStr)`
- `void setKeyReleaseFunc(const char *cStr)`
- `void setLeaveFunc(const char *cStr)`
- `void setNonClientAreaMouseButtonDbClickFunc(const char *cStr)`
- `void setNonClientAreaMouseButtonPressFunc(const char *cStr)`
- `void setNonClientAreaMouseButtonReleaseFunc(const char *cStr)`
- `void setNonClientAreaMouseMoveFunc(const char *cStr)`
- `void setMoveFunc(const char *cStr)`
- `void setResizeFunc(const char *cStr)`
- `void setWindowActivateFunc(const char *cStr)`
- `void setWindowBlockedFunc(const char *cStr)`
- `void setWindowDeactivateFunc(const char *cStr)`
- `void setWindowStateChangeFunc(const char *cStr)`
- `void setWindowUnblockedFunc(const char *cStr)`
- `void setPaintFunc(const char *cStr)`
- `const char *getKeyPressFunc(void)`
- `const char *getMouseButtonPressFunc(void)`
- `const char *getMouseButtonReleaseFunc(void)`
- `const char *getMouseButtonDbClickFunc(void)`
- `const char *getMouseMoveFunc(void)`
- `const char *getCloseFunc(void)`

- const char *getContextMenuFunc(void)
- const char *getDragEnterFunc(void)
- const char *getDragLeaveFunc(void)
- const char *getDragMoveFunc(void)
- const char *getDropFunc(void)
- const char *getEnterFunc(void)
- const char *getFocusInFunc(void)
- const char *getFocusOutFunc(void)
- const char *getKeyReleaseFunc(void)
- const char *getLeaveFunc(void)
- const char *getNonClientAreaMouseButtonDbClickFunc(void)
- const char *getNonClientAreaMouseButtonPressFunc(void)
- const char *getNonClientAreaMouseButtonReleaseFunc(void)
- const char *getNonClientAreaMouseMoveFunc(void)
- const char *getMoveFunc(void)
- const char *getResizeFunc(void)
- const char *getWindowActivateFunc(void)
- const char *getWindowBlockedFunc(void)
- const char *getWindowDeactivateFunc(void)
- const char *getWindowStateChangeFunc(void)
- const char *getWindowUnblockedFunc(void)
- const char *getPaintFunc(void)

99.88 QDesktopWidget Class

C++ Reference : <http://doc.qt.io/qt-5/QDesktopWidget.html>

Parameters : void

Parent Class : QWidget

- QRect availableGeometry(int screen)
- bool isVirtualDesktop(void)
- int primaryScreen(void)
- QWidget *screen(int screen)
- int screenCount(void)
- QRect screenGeometry(int screen)
- int screenNumber(QWidget *widget)

99.89 QRect Class

C++ Reference : <http://doc.qt.io/qt-5/QRect.html>

Parameters : void

- void adjust(int dx1, int dy1, int dx2, int dy2)
- QRect adjusted(int dx1, int dy1, int dx2, int dy2)
- int bottom(void)
- QPoint bottomLeft(void)
- QPoint bottomRight(void)
- QPoint center(void)
- bool contains(int x, int y, bool proper)
- void getCoords(int *x1, int *y1, int *x2, int *y2)
- void getRect(int *x, int *y, int *width, int *height)
- int height(void)
- QRect intersected(QRect)
- bool intersects(QRect)
- bool isEmpty(void)
- bool isNull(void)
- bool isValid(void)
- int left(void)
- void moveBottom(int y)
- void moveBottomLeft(QPoint)
- void moveBottomRight(QPoint)
- void moveCenter(QPoint)
- void moveLeft(int x)
- void moveRight(int x)
- void moveTo(int x, int y)
- void moveTop(int y)
- void moveTopLeft(QPoint)
- void moveTopRight(QPoint)
- QRect normalized(void)
- int right(void)
- void setBottom(int y)
- void setBottomLeft(QPoint)
- void setBottomRight(QPoint)
- void setCoords(int x1, int y1, int x2, int y2)

- void setHeight(int height)
- void setLeft(int x)
- void setRect(int x, int y, int width, int height)
- void setRight(int x)
- void setSize(QSize)
- void setTop(int y)
- void setTopLeft(QPoint)
- void setTopRight(QPoint)
- void setWidth(int width)
- void setX(int x)
- void setY(int y)
- QSize size(void)
- int top(void)
- QPoint topLeft(void)
- QPoint topRight(void)
- void translate(int dx, int dy)
- QRect translated(int dx, int dy)
- QRect united(QRect)
- int width(void)
- int x(void)
- int y(void)

99.90 QTextDocument Class

C++ Reference : <http://doc.qt.io/qt-5/QTextDocument.html>

Parameters : void

Parent Class : QObject

- void addResource(int type, QUrl name, QVariant resource)
- void adjustSize(void)
- QVector<QTextFormat> allFormats(void)
- int availableRedoSteps(void)
- int availableUndoSteps(void)
- QTextBlock begin(void)
- int blockCount(void)
- QChar characterAt(int pos)
- int characterCount(void)

- void clearUndoRedoStacks(QTextDocument::Stacks stacksToClear)
- QTextDocument *clone(QObject *parent)
- int defaultCursorMoveStyle(void)
- QFont defaultFont(void)
- QString defaultStyleSheet(void)
- QTextOption defaultTextOption(void)
- QAbstractTextDocumentLayout *documentLayout(void)
- double documentMargin(void)
- void drawContents(QPainter *p, QRectF rect)
- QTextBlock end(void) # In RingQt use : QTextBlock enddoc(void)
- QTextCursor find(QString subString, QTextCursor cursor, QTextDocument::FindFlag options)
- QTextBlock findBlock(int pos)
- QTextBlock findBlockByLineNumber(int lineNumber)
- QTextBlock findBlockByNumber(int blockNumber)
- QTextBlock firstBlock(void)
- double idealWidth(void)
- double indentWidth(void)
- bool isEmpty(void)
- bool isModified(void)
- bool isRedoAvailable(void)
- bool isUndoAvailable(void)
- bool isUndoRedoEnabled(void)
- QTextBlock lastBlock(void)
- int lineCount(void)
- void markContentsDirty(int position, int length)
- int maximumBlockCount(void)
- QString metaInformation(QTextDocument::MetaInformation info)
- QTextObject *object(int objectIndex)
- QTextObject *objectForFormat(QTextFormat f)
- int pageCount(void)
- QSizeF pageSize(void)
- void print(QPrinter *printer)
- void redo(QTextCursor *cursor)
- QVariant resource(int type, QUrl name)
- int revision(void)
- QTextFrame *rootFrame(void)

- void setDefaultCursorMoveStyle(Qt::CursorMoveStyle style)
- void setDefaultFont(QFont font)
- void setDefaultStyleSheet(QString sheet)
- void setDefaultTextOption(QTextOption option)
- void setDocumentLayout(QAbstractTextDocumentLayout * layout)
- void setDocumentMargin(double margin)
- void setHtml(QString html)
- void setIndentWidth(double width)
- void setMaximumBlockCount(int maximum)
- void setMetaInformation(QTextDocument::MetaInformation info, QString string)
- void setPageSize(QSizeF size)
- void setPlainText(QString text)
- void setTextWidth(double width)
- void setUndoRedoEnabled(bool enable)
- void setUseDesignMetrics(bool b)
- QSizeF size(void)
- qreal textWidth(void)
- QString toHtml(QByteArray encoding)
- QString toPlainText(void)
- void undo(QTextCursor *cursor)
- bool useDesignMetrics(void)
- void setModified(bool m)

99.91 QTextBlock Class

C++ Reference : <http://doc.qt.io/qt-5/QTextBlock.html>

Parameters : void

- int blockFormatIndex(void)
- int blockNumber(void)
- QTextCharFormat charFormat(void)
- int charFormatIndex(void)
- void clearLayout(void)
- bool contains(int position)
- QTextDocument *document(void)
- bool isValid(void)
- bool isVisible(void)

- QTextLayout * layout(void)
- int length(void)
- int lineCount(void)
- QTextBlock next(void) # In RingQt use : QTextBlock nextblock(void)
- int position(void)
- QTextBlock previous(void)
- int revision(void)
- void setLineCount(int count)
- void setRevision(int rev)
- void setUserData(QTextBlockUserData * data)
- void setUserState(int state)
- void setVisible(bool visible)
- QString text(void)
- int textDirection(void)
- QTextList * textList(void)
- QTextBlockUserData * userData(void)
- int userState(void)

99.92 QTime Class

C++ Reference : <http://doc.qt.io/qt-5/QTime.html>

Parameters : void

- QTime addMSecs(int ms)
- QTime addSecs(int s)
- int elapsed(void)
- int hour(void)
- bool isNull(void)
- bool isValid(void)
- int minute(void)
- int msec(void)
- int msecsSinceStartOfDay(void)
- int msecsTo(QTime)
- int restart(void)
- int second(void)
- int secsTo(QTime)
- bool setHMS(int h, int m, int s, int ms)

- void start(void)
- QString toString(QString)
- QTime currentTime(void)
- QTime fromMSecsSinceStartOfDay(int msecs)
- QTime fromString(QString,QString)

99.93 QListWidgetItem Class

C++ Reference : <http://doc.qt.io/qt-5/QListWidgetItem.html>

Parameters : void

- QBrush background(void)
- Qt::CheckState checkState(void)
- Qt::ItemFlags flags(void)
- QFont font(void)
- QBrush foreground(void)
- QIcon icon(void)
- bool isHidden(void)
- bool isSelected(void)
- QListWidget *listWidget(void)
- void setBackground(QBrush brush)
- void setCheckState(Qt::CheckState state)
- void setFlags(Qt::ItemFlags flags)
- void setFont(QFont font)
- void setForeground(QBrush brush)
- void setHidden(bool hide)
- void setIcon(QIcon icon)
- void setSelected(bool select)
- void setSizeHint(QSize size)
- void setStatusTip(QString statusTip)
- void setText(QString text)
- void setTextAlignment(int alignment)
- void setToolTip(QString toolTip)
- void setWhatsThis(QString whatsThis)
- QSize sizeHint(void)
- QString statusTip(void)
- QString text(void)

- int textAlignment(void)
- QString toolTip(void)
- int type(void)
- QString whatsThis(void)

99.94 QSystemTrayIcon Class

C++ Reference : <http://doc.qt.io/qt-5/QSystemTrayIcon.html>

Parameters : void

- QMenu *contextMenu(void)
- QRect geometry(void)
- QIcon icon(void)
- bool isVisible(void)
- void setContextMenu(QMenu *menu)
- void setIcon(QIcon)
- void setToolTip(QString)
- QString toolTip(void)
- void hide(void)
- void setVisible(bool visible)
- void show(void)
- void showMessage(QString, QString, QSystemTrayIcon::MessageIcon, int millisecondsTimeoutHint)
- bool isSystemTrayAvailable(void)
- bool supportsMessages(void)

99.95 QDate Class

C++ Reference : <http://doc.qt.io/qt-5/QDate.html>

Parameters : void

- QDate addDays(int ndays)
- QDate addMonths(int nmonths)
- QDate addYears(int nyears)
- int day(void)
- int dayOfWeek(void)
- int dayOfYear(void)
- int daysInMonth(void)
- int daysInYear(void)
- int daysTo(QDate)

- void getDate(int * year, int * month, int * day)
- bool isNull(void)
- bool isValid(void)
- int month(void)
- bool setDate(int year, int month, int day)
- int toJulianDay(void)
- QString toString(QString)
- int weekNumber(int * yearNumber)
- int year(void)
- QDate currentDate(void)
- QDate fromJulianDay(int jd)
- QDate fromString(QString, QString)
- bool isLeapYear(int year)
- QString longDayName(int weekday)
- QString longMonthName(int month)
- QString shortDayName(int weekday)
- QString shortMonthName(int month)

99.96 QTextCodec Class

C++ Reference : <http://doc.qt.io/qt-5/QTextCodec.html>

- QTextCodec *codecForName(const char *name)
- void setCodecForLocale(QTextCodec *c)

99.97 QSqlDatabase Class

C++ Reference : <http://doc.qt.io/qt-5/QSqlDatabase.html>

Parameters : void

- void close(void)
- bool commit(void)
- QString connectOptions(void)
- QString connectionName(void)
- QString databaseName(void)
- QSqlDriver *driver(void)
- QString driverName(void)
- QSqlQuery exec(QString)
- QString hostName(void)

- bool isOpen(void)
- bool isOpenError(void)
- bool isValid(void)
- QSqlError lastError(void)
- QSql::NumericalPrecisionPolicy numericalPrecisionPolicy(void)
- bool open(void)
- QString password(void)
- int port(void)
- QSqlIndex primaryIndex(QString)
- QSqlRecord record(QString)
- bool rollback(void)
- void setConnectOptions(QString)
- void setDatabaseName(QString)
- void setHostName(QString)
- void setNumericalPrecisionPolicy(QSql::NumericalPrecisionPolicy precisionPolicy)
- void setPassword(QString)
- void setPort(int port)
- void setUsername(QString)
- QStringList tables(QSql::TableType type)
- bool transaction(void)
- QString userName(void)
- QSqlDatabase addDatabase(QString)
- QSqlDatabase cloneDatabase(QSqlDatabase, QString)
- QStringList connectionNames(void)
- bool contains(QString)
- QSqlDatabase database(QString , bool)
- QStringList drivers(void)
- bool isDriverAvailable(QString)
- void registerSqlDriver(QString, QSqlDriverCreatorBase *)
- void removeDatabase(QString)

99.98 QSqlDriver Class

C++ Reference : <http://doc.qt.io/qt-5/QSqlDriver.html>

Parameters : void

- QSqlError lastError(void)

- QSql::NumericalPrecisionPolicy numericalPrecisionPolicy(void)
- void setNumericalPrecisionPolicy(QSql::NumericalPrecisionPolicy)

99.99 QSqlQuery Class

C++ Reference : <http://doc.qt.io/qt-5/QSqlQuery.html>

Parameters : void

- void addBindValue(QVariant, QSql::ParamType paramType)
- int at(void)
- void bindValue(QString, QVariant, QSql::ParamType paramType)
- QVariant boundValue(QString)
- void clear(void)
- QSqlDriver * driver(void)
- bool exec(QString)
- bool exec_2(void)
- bool execBatch(QSqlQuery::BatchExecutionMode mode)
- QString executedQuery(void)
- void finish(void)
- bool first(void)
- bool isActive(void)
- bool isForwardOnly(void)
- bool isNull(int field)
- bool isSelected(void)
- bool isValid(void)
- bool last(void)
- QSqlError lastError(void)
- QVariant lastInsertId(void)
- QString lastQuery(void)
- bool next(void) # In RingQt use : bool movenext(void)
- bool nextResult(void)
- int numRowsAffected(void)
- QSql::NumericalPrecisionPolicy numericalPrecisionPolicy(void)
- bool prepare(QString)
- bool previous(void)
- QSqlRecord record(void)
- QSqlResult *result(void)

- `bool seek(int index, bool relative)`
- `void setForwardOnly(bool forward)`
- `void setNumericalPrecisionPolicy(QSql::NumericalPrecisionPolicy precisionPolicy)`
- `int size(void)`
- `QVariant value(int index)`

99.100 QSqlError Class

C++ Reference : <http://doc.qt.io/qt-5/QSqlError.html>

Parameters : `QString`, `QString`, `QSqlError::ErrorType`

- `QString databaseText(void)`
- `QString driverText(void)`
- `bool isValid(void)`
- `int number(void)`
- `void setDatabaseText(QString)`
- `void setDriverText(QString)`
- `void setNumber(int number)`
- `void setType(QSqlError::ErrorType type)`
- `QString text(void)`
- `QSqlError::ErrorType type(void)`

99.101 QSqlIndex Class

C++ Reference : <http://doc.qt.io/qt-5/QSqlIndex.html>

Parameters : `QString`, `QString`

Parent Class : `QSqlRecord`

- `void append(QSqlField, bool)`
- `QString cursorName(void)`
- `bool isDescending(int i)`
- `QString name(void)`
- `void setCursorName(QString)`
- `void setDescending(int i, bool desc)`
- `void setName(QString)`

99.102 QSqlRecord Class

C++ Reference : <http://doc.qt.io/qt-5/QtSqlRecord.html>

Parameters : void

- void append(QSqlField)
- void clear(void)
- void clearValues(void)
- bool contains(QString)
- int count(void)
- QSqlField field(int index)
- QString fieldName(int index)
- int indexOf(QString)
- void insert(int pos, QSqlField)
- bool isEmpty(void)
- bool isGenerated(QString)
- bool isNull(QString)
- void remove(int pos)
- void replace(int pos, QSqlField)
- void setGenerated(QString, bool generated)
- void setNull(int index)
- void setValue(int index, QVariant)
- QVariant value(int index)

99.103 QSqlField Class

C++ Reference : <http://doc.qt.io/qt-5/QtSqlField.html>

Parameters : QString, QVariant::Type

- void clear(void)
- QVariant defaultValue(void)
- bool isAutoValue(void)
- bool isGenerated(void)
- bool isNull(void)
- bool isReadOnly(void)
- bool isValid(void)
- int length(void)
- QString name(void)
- int precision(void)

- RequiredStatus requiredStatus(void)
- void setAutoValue(bool autoVal)
- void setDefaultValue(QVariant)
- void setGenerated(bool gen)
- void setLength(int fieldLength)
- void setName(QString)
- void setPrecision(int precision)
- void setReadOnly(bool readOnly)
- void setRequired(bool required)
- void setRequiredStatus(QSqlField::RequiredStatus required)
- void setType(QVariant::Type type)
- void setValue(QVariant)
- QVariant::Type type(void)
- QVariant value(void)

99.104 QSqlDriverCreatorBase Class

C++ Reference : <http://doc.qt.io/qt-5/QSqlDriverCreatorBase.html>

Parameters : void

99.105 QVariant Class

C++ Reference : <http://doc.qt.io/qt-5/QVariant.html>

Parameters : void

- bool canConvert(int targetTypeId)
- void clear(void)
- bool convert(int targetTypeId)
- bool isNull(void)
- bool isValid(void)
- void swap(QVariant)
- QBitArray toBitArray(void)
- bool toBool(void)
- QByteArray toByteArray(void)
- QChar toChar(void)
- QDate toDate(void)
- QDateTime toDateTime(void)
- double toDouble(bool *ok)

- QEasingCurve toEasingCurve(void)
- float toFloat(bool *ok)
- int toInt(bool *ok)
- QJsonArray toJsonArray(void)
- QJsonDocument toJsonDocument(void)
- QJsonObject toJsonObject(void)
- QJsonValue toJsonValue(void)
- QLine toLine(void)
- QLineF toLineF(void)
- QLocale toLocale(void)
- qlonglong toLongLong(bool *ok)
- QModelIndex toModelIndex(void)
- QPointF toPointF(void)
- qreal toReal(bool *ok)
- QRect toRect(void)
- QRectF toRectF(void)
- QRegExp toRegExp(void)
- QRegularExpression toRegularExpression(void)
- QSize toSize(void)
- QSizeF toSizeF(void)
- QStringList toStringList(void)
- QTime toTime(void)
- uint toUInt(bool *ok)
- qulonglong toULongLong(bool *ok)
- QUrl toUrl(void)
- QUuid toUuid(void)
- QVariant::Type type(void)
- const char *typeName(void)
- int userType(void)
- QString toString(void)

99.106 QNetworkRequest Class

C++ Reference : <http://doc.qt.io/qt-5/QNetworkRequest.html>

Parameters : QUrl

- QVariant attribute(QNetworkRequest::Attribute, QVariant)

- bool hasRawHeader(QByteArray)
- QVariant header(QNetworkRequest::KnownHeaders)
- QObject *originatingObject(void)
- QNetworkRequest::Priority priority(void)
- QByteArray rawHeader(QByteArray)
- void setAttribute(QNetworkRequest::Attribute, QVariant)
- void setHeader(QNetworkRequest::KnownHeaders, QVariant)
- void setOriginatingObject(QObject *object)
- void setPriority(QNetworkRequest::Priority priority)
- void setRawHeader(QByteArray, QByteArray)
- void swap(QNetworkRequest)
- QUrl url(void)

99.107 QNetworkAccessManager Class

C++ Reference : <http://doc.qt.io/qt-5/QNetworkAccessManager.html>

Parameters : QObject *

Parent Class : QObject

- void setfinishedEvent(const char *)
- const char *getfinishedEvent(void)
- QNetworkConfiguration activeConfiguration(void)
- QAbstractNetworkCache *cache(void)
- void clearAccessCache(void)
- QNetworkConfiguration configuration(void)
- void connectToHost(QString, quint16)
- QNetworkReply *deleteResource(QNetworkRequest)
- QNetworkReply *get(QNetworkRequest) # In RingQt use : QNetworkReply *getValue(QNetworkRequest)
- QNetworkReply *head(QNetworkRequest)
- QNetworkAccessManager::NetworkAccessibility networkAccessible(void)
- QNetworkReply *post(QNetworkRequest, QByteArray)
- QNetworkProxy proxy(void)
- QNetworkProxyFactory *proxyFactory(void)
- QNetworkReply *put(QNetworkRequest, QByteArray) # In RingQt use : QNetworkReply *put-value(QNetworkRequest, QByteArray)
- QNetworkReply *sendCustomRequest(QNetworkRequest, QByteArray, QIODevice *)
- void setCache(QAbstractNetworkCache *cache)
- void setConfiguration(QNetworkConfiguration)

- void setCookieJar(QNetworkCookieJar *cookieJar)
- void setNetworkAccessible(QNetworkAccessManager::NetworkAccessibility accessible)
- void setProxy(QNetworkProxy)
- void setProxyFactory(QNetworkProxyFactory *factory)
- QStringList supportedSchemes(void)
- void geteventparameters(void)

99.108 QNetworkReply Class

C++ Reference : <http://doc.qt.io/qt-5/QNetworkReply.html>

Parameters : void

Parent Class : QIODevice

- QVariant attribute(QNetworkRequest::Attribute code)
- QNetworkReply::NetworkError error(void)
- bool hasRawHeader(QByteArray)
- QVariant header(QNetworkRequest::KnownHeaders header)
- bool isFinished(void)
- bool isRunning(void)
- QNetworkAccessManager *manager(void)
- QNetworkAccessManager::Operation operation(void)
- QByteArray rawHeader(QByteArray)
- qint64 readBufferSize(void)
- QNetworkRequest request(void)

99.109 QPainterPath Class

C++ Reference : <http://doc.qt.io/qt-5/QPainterPath.html>

Parameters : void

- void addEllipse(qreal x, qreal y, qreal width, qreal height)
- void addPath(QPainterPath)
- void addPolygon(QPolygonF)
- void addRect(qreal x, qreal y, qreal width, qreal height)
- void addRegion(QRegion)
- void addRoundedRect(qreal x, qreal y, qreal w, qreal h, qreal xRadius, qreal yRadius, Qt::SizeMode mode)
- void addText(qreal x, qreal y, QFont, QString)
- qreal angleAtPercent(qreal t)
- void arcMoveTo(qreal x, qreal y, qreal width, qreal height, qreal angle)

- void arcTo(qreal x, qreal y, qreal width, qreal height, qreal startAngle, qreal sweepLength)
- QRectF boundingRect(void)
- void closeSubpath(void)
- void connectPath(QPainterPath)
- bool contains(QPointF)
- QRectF controlPointRect(void)
- void cubicTo(qreal c1X, qreal c1Y, qreal c2X, qreal c2Y, qreal endPointX, qreal endPointY)
- QPointF currentPosition(void)
- QPainterPath::Element elementAt(int index)
- int elementCount(void)
- Qt::FillRule fillRule(void)
- QPainterPath intersected(QPainterPath)
- bool intersects(QRectF)
- bool isEmpty(void)
- qreal length(void)
- void lineTo(qreal x, qreal y)
- void moveTo(qreal x, qreal y)
- qreal percentAtLength(qreal len)
- QPointF pointAtPercent(qreal t)
- void quadTo(qreal cx, qreal cy, qreal endPointX, qreal endPointY)
- void setElementPositionAt(int index, qreal x, qreal y)
- void setFillRule(Qt::FillRule fillRule)
- QPainterPath simplified(void)
- qreal slopeAtPercent(qreal t)
- QPainterPath subtracted(QPainterPath)
- void swap(QPainterPath)
- QPolygonF toFillPolygon(QTransform)
- QPainterPath toReversed(void)
- void translate(qreal dx, qreal dy)
- QPainterPath translated(qreal dx, qreal dy)
- QPainterPath united(QPainterPath)

99.110 QImage Class

C++ Reference : <http://doc.qt.io/qt-5/QImage.html>

Parameters : void

- bool allGray(void)
- int bitPlaneCount(void)
- uchar *bits(void)
- int byteCount(void)
- int bytesPerLine(void)
- qint64 cacheKey(void)
- QRgb color(int i)
- int colorCount(void)
- const uchar *constBits(void)
- const uchar *constScanLine(int i)
- QImage convertToFormat(QImage::Format format, Qt::ImageConversionFlags flags)
- QImage copy(int x, int y, int width, int height)
- QImage createAlphaMask(Qt::ImageConversionFlags flags)
- QImage createHeuristicMask(bool clipTight)
- QImage createMaskFromColor(QRgb color, Qt::MaskMode mode)
- int depth(void)
- int dotsPerMeterX(void)
- int dotsPerMeterY(void)
- void fill(QColor)
- QImage::Format format(void)
- bool hasAlphaChannel(void)
- int height(void)
- void invertPixels(QImage::InvertMode mode)
- bool isGrayscale(void)
- bool isNull(void)
- bool load(QString, const char *format) # In RingQt use : bool loadimage(QString, const char *format)
- bool loadFromData(QByteArray, const char *format)
- QImage mirrored(bool horizontal, bool vertical)
- QPoint offset(void)
- QRgb pixel(int x, int y)
- int pixelIndex(int x, int y)
- QRect rect(void)
- QImage rgbSwapped(void)
- bool save(QString, const char *format, int quality)
- QImage scaled(int width, int height, Qt::AspectRatioMode aspectRatioMode, Qt::TransformationMode transformMode)

- QImage scaledToHeight(int height, Qt::TransformationMode mode)
- QImage scaledToWidth(int width, Qt::TransformationMode mode)
- uchar *scanLine(int i)
- void setColor(int index, QRgb colorValue)
- void setColorCount(int colorCount)
- void setDotsPerMeterX(int x)
- void setDotsPerMeterY(int y)
- void setOffset(QPoint)
- void setPixel(int x, int y, uint index_or_rgb)
- void setText(QString,QString)
- QSize size(void)
- void swap(QImage)
- QString text(QString)
- QStringList textKeys(void)
- QImage transformed(QMatrix, Qt::TransformationMode mode)
- bool valid(int x, int y)
- int width(void)

99.111 RingCodeHighlighter Class

Parameters : QTextDocument *parent

- void setColors(QColor c1,QColor c2,QColor c3,QColor c4,QColor c5)
- void setKeywordsBold(int nStatus)

99.112 QDomStreamReader Class

C++ Reference : <http://doc.qt.io/qt-5/QDomStreamReader.html>

Parameters : void

- void addData(QByteArray)
- void addData_2(QString)
- void addData_3(const char * data)
- void addExtraNamespaceDeclaration(QDomStreamNamespaceDeclaration)
- void addExtraNamespaceDeclarations(QDomStreamNamespaceDeclarations)
- bool atEnd(void)
- QDomStreamAttributes attributes(void)
- qint64 characterOffset(void)
- void clear(void)

- qint64 columnNumber(void)
- QIODevice *device(void)
- QStringRef documentEncoding(void)
- QStringRef documentVersion(void)
- QStringRef dtdName(void)
- QStringRef dtdPublicId(void)
- QStringRef dtdSystemId(void)
- QDomStreamEntityDeclarations entityDeclarations(void)
- QDomStreamEntityResolver *entityResolver(void)
- Error error(void)
- QString errorString(void)
- bool hasError(void)
- bool isCDATA(void)
- bool isCharacters(void)
- bool isComment(void)
- bool isDTD(void)
- bool isEndDocument(void)
- bool isEndElement(void)
- bool isEntityReference(void)
- bool isProcessingInstruction(void)
- bool isStandaloneDocument(void)
- bool isStartDocument(void)
- bool isStartElement(void)
- bool isWhitespace(void)
- qint64 lineNumber(void)
- QStringRef name(void)
- QDomStreamNamespaceDeclarations namespaceDeclarations(void)
- bool namespaceProcessing(void)
- QStringRef namespaceUri(void)
- QDomStreamNotationDeclarations notationDeclarations(void)
- QStringRef prefix(void)
- QStringRef processingInstructionData(void)
- QStringRef processingInstructionTarget(void)
- QStringRef qualifiedName(void)
- void raiseError(QString)
- QString readElementText(QDomStreamReader::ReadElementTextBehaviour)

- TokenType readNext(void)
- bool readNextStartElement(void)
- void setDevice(QIODevice *device)
- void setEntityResolver(QXmlStreamEntityResolver *resolver)
- void setNamespaceProcessing(bool)
- void skipCurrentElement(void)
- QStringRef text(void)
- QString tokenString(void)
- TokenType tokenType(void)

99.113 QXmlStreamWriter Class

C++ Reference : <http://doc.qt.io/qt-5/QXmlStreamWriter.html>

Parameters : void

- bool autoFormatting(void)
- int autoFormattingIndent(void)
- QTextCodec *codec(void)
- QIODevice *device(void)
- bool hasError(void)
- void setAutoFormatting(bool enable)
- void setAutoFormattingIndent(int spacesOrTabs)
- void setCodec(QTextCodec *codec)
- void setCodec_2(const char *codecName)
- void setDevice(QIODevice *device)
- void writeAttribute(QString, QString, QString)
- void writeAttribute_2(QString, QString)
- void writeAttribute_3(QXmlStreamAttribute)
- void writeAttributes(QXmlStreamAttributes)
- void writeCDATA(QString text)
- void writeCharacters(QString text)
- void writeComment(QString text)
- void writeCurrentToken(QXmlStreamReader reader)
- void writeDTD(QString dtd)
- void writeDefaultNamespace(QString namespaceUri)
- void writeEmptyElement(QString namespaceUri, QString name)
- void writeEmptyElement_2(QString qualifiedName)

- void writeEndDocument(void)
- void writeEndElement(void)
- void writeEntityReference(QString name)
- void writeNamespace(QString namespaceUri, QString prefix)
- void writeProcessingInstruction(QString target, QString data)
- void writeStartDocument(QString version)
- void writeStartDocument_2(QString version, bool standalone)
- void writeStartDocument_3(void)
- void writeStartElement(QString namespaceUri, QString name)
- void writeStartElement_2(QString qualifiedName)
- void writeTextElement(QString namespaceUri, QString name, QString text)
- void writeTextElement_2(QString qualifiedName, QString text)

99.114 QXmlStreamNotationDeclaration Class

C++ Reference : <http://doc.qt.io/qt-5/QXmlStreamNotationDeclaration.html>

Parameters : void

- QStringRef name(void)
- QStringRef publicId(void)
- QStringRef systemId(void)

99.115 QXmlStreamNamespaceDeclaration Class

C++ Reference : <http://doc.qt.io/qt-5/QXmlStreamNamespaceDeclaration.html>

Parameters : void

- QStringRef namespaceUri(void)
- QStringRef prefix(void)

99.116 QXmlStreamEntityResolver Class

C++ Reference : <http://doc.qt.io/qt-5/QXmlStreamEntityResolver.html>

Parameters : void

99.117 QXmlStreamEntityDeclaration Class

C++ Reference : <http://doc.qt.io/qt-5/QXmlStreamEntityDeclaration.html>

Parameters : void

- QStringRef name(void)
- QStringRef notationName(void)
- QStringRef publicId(void)
- QStringRef systemId(void)
- QStringRef value(void)

99.118 QDomStreamAttributes Class

C++ Reference : <http://doc.qt.io/qt-5/QDomStreamAttributes.html>

Parameters : void

- void append(QString namespaceUri, QString name, QString value)
- void append_2(QString qualifiedName, QString value)
- bool hasAttribute(QString qualifiedName)
- bool hasAttribute_2(QLatin1String qualifiedName)
- bool hasAttribute_3(QString namespaceUri, QString name)
- QStringRef value(QString namespaceUri, QString name)
- QStringRef value_2(QString namespaceUri, QLatin1String name)
- QStringRef value_3(QLatin1String namespaceUri, QLatin1String name)
- QStringRef value_4(QString qualifiedName)
- QStringRef value_5(QLatin1String qualifiedName)

99.119 QDomStreamAttribute Class

C++ Reference : <http://doc.qt.io/qt-5/QDomStreamAttribute.html>

Parameters : void

- bool isDefault(void)
- QStringRef name(void)
- QStringRef namespaceUri(void)
- QStringRef prefix(void)
- QStringRef qualifiedName(void)
- QStringRef value(void)

99.120 QThread Class

C++ Reference : <http://doc.qt.io/qt-5/QThread.html>

Parameters : QObject *

Parent Class : QObject

- QAbstractEventDispatcher *eventDispatcher(void)
- void exit(int returnCode) # In RingQt use : void exitfromthread(int returnCode)
- bool isFinished(void)
- bool isInterruptionRequested(void)
- bool isRunning(void)
- QThread::Priority priority(void)
- void requestInterruption(void)
- void setEventDispatcher(QAbstractEventDispatcher *eventDispatcher)
- void setPriority(QThread::Priority priority)
- void setStackSize(uint stackSize)
- uint stackSize(void)
- bool wait(unsigned long time)
- void quit(void)
- void start(QThread::Priority priority)
- void terminate(void)
- QThread *currentThread(void)
- Qt::HANDLE currentThreadId(void)
- int idealThreadCount(void)
- void msleep(unsigned long msecs)
- void sleep(unsigned long secs)
- void usleep(unsigned long usecs)
- void yieldCurrentThread(void)
- void setStartedEvent(const char *)
- void setFinishedEvent(const char *)
- const char *getStartedEvent(void)
- const char *getFinishedEvent(void)

99.121 QThreadPool Class

C++ Reference : <http://doc.qt.io/qt-5/QThreadPool.html>

Parameters : void

Parent Class : QObject

- int activeThreadCount(void)
- void clear(void)
- int expiryTimeout(void)
- int maxThreadCount(void)

- void releaseThread(void)
- void reserveThread(void)
- void setExpiryTimeout(int expiryTimeout)
- void setMaxThreadCount(int maxThreadCount)
- void start(QRunnable * runnable, int priority)
- bool tryStart(QRunnable * runnable)
- bool waitForDone(int msec)
- QThreadPool *globalInstance(void)

99.122 QRegularExpression Class

C++ Reference : <http://doc.qt.io/qt-5/QRegularExpression.html>

Parameters : void

- int captureCount(void)
- QString errorString(void)
- QRegularExpressionMatchIterator globalMatch(QString subject, int offset, QRegularExpression::MatchType matchType, QRegularExpression::MatchOptions matchOptions)
- bool isValid(void)
- QRegularExpressionMatch match(QString subject, int offset, QRegularExpression::MatchType matchType, QRegularExpression::MatchOptions matchOptions)
- QStringList namedCaptureGroups(void)
- QString pattern(void)
- int patternErrorOffset(void)
- QRegularExpression::PatternOptions patternOptions(void)
- void setPattern(QString pattern)
- void setPatternOptions(QRegularExpression::PatternOptions options)
- void swap(QRegularExpression other)

99.123 QRegularExpressionMatch Class

C++ Reference : <http://doc.qt.io/qt-5/QRegularExpressionMatch.html>

Parameters : void

- QString captured(int nth)
- QString captured_2(const QString name)
- int capturedEnd(int nth)
- int capturedEnd_2(const QString name)
- int capturedLength(int nth)

- `int capturedLength_2(const QString name)`
- `QStringRef capturedRef(int nth)`
- `QStringRef capturedRef_2(const QString name)`
- `int capturedStart(int nth)`
- `int capturedStart_2(const QString name)`
- `QStringList capturedTexts(void)`
- `bool hasMatch(void)`
- `bool hasPartialMatch(void)`
- `bool isValid(void)`
- `int lastCapturedIndex(void)`
- `QRegularExpression::MatchOptions matchOptions(void)`
- `QRegularExpression::MatchType matchType(void)`
- `QRegularExpression regularExpression(void)`
- `void swap(QRegularExpressionMatch other)`

99.124 QRegularExpressionMatchIterator Class

C++ Reference : <http://doc.qt.io/qt-5/QRegularExpressionMatchIterator.html>

Parameters : void

- `bool hasNext(void)`
- `bool isValid(void)`
- `QRegularExpression::MatchOptions matchOptions(void)`
- `QRegularExpression::MatchType matchType(void)`
- `QRegularExpressionMatch next(void)` # In RingQt use : `QRegularExpressionMatch nextitem(void)`
- `QRegularExpressionMatch peekNext(void)`
- `QRegularExpression regularExpression(void)`
- `void swap(QRegularExpressionMatchIterator other)`

99.125 QJsonArray Class

C++ Reference : <http://doc.qt.io/qt-5/QJsonArray.html>

Parameters : void

- `void append(QJsonValue value)`
- `QJsonValue at(int i)`
- `bool contains(QJsonValue value)`
- `int count(void)`
- `bool empty(void)`

- `QJsonValue first(void)`
- `void insert(int i, QJsonValue value)`
- `bool isEmpty(void)`
- `QJsonValue last(void)`
- `void pop_back(void)`
- `void pop_front(void)`
- `void prepend(QJsonValue value)`
- `void push_back(QJsonValue value)`
- `void push_front(QJsonValue value)`
- `void removeAt(int i)`
- `void removeFirst(void)`
- `void removeLast(void)`
- `void replace(int i, QJsonValue value)`
- `int size(void)`
- `QJsonValue takeAt(int i)`
- `QVariantList toVariantList(void)`
- `QJsonArray fromStringList(QStringList list)`
- `QJsonArray fromVariantList(QVariantList list)`

99.126 QJsonDocument Class

C++ Reference : <http://doc.qt.io/qt-5/QJsonDocument.html>

Parameters : void

- `QJsonArray array(void)`
- `bool isArray(void)`
- `bool isEmpty(void)`
- `bool isNull(void)`
- `bool isObject(void)`
- `QJsonObject object(void)`
- `const char * rawData(int * size)`
- `void setArray(QJsonArray array)`
- `void setObject(QJsonObject object)`
- `QByteArray toBinaryData(void)`
- `QByteArray toJson(QJsonDocument::JsonFormat format)`
- `QVariant toVariant(void)`
- `QJsonDocument fromBinaryData(QByteArray data, QJsonDocument::DataValidation validation)`

- `QJsonDocument fromJson(QByteArray json, QJsonParseError * error)`
- `QJsonDocument fromRawData(char * data, int size, QJsonDocument::DataValidation validation)`
- `QJsonDocument fromVariant(QVariant variant)`

99.127 QJsonObject Class

C++ Reference : <http://doc.qt.io/qt-5/QJsonObject.html>

Parameters : void

- `bool contains(QString key)`
- `int count(void)`
- `bool empty(void)`
- `bool isEmpty(void)`
- `QStringList keys(void)`
- `int length(void)`
- `void remove(QString key)`
- `int size(void)`
- `QJsonValue take(QString key)`
- `QVariantMap toVariantMap(void)`
- `QJsonValue value(QString key)`
- `QJsonObject fromVariantMap(QVariantMap map)`

99.128 QJsonParseError Class

C++ Reference : <http://doc.qt.io/qt-5/QJsonParseError.html>

Parameters : void

- `QString errorString(void)`

99.129 QJsonValue Class

C++ Reference : <http://doc.qt.io/qt-5/QJsonValue.html>

Parameters : void

- `bool isArray(void)`
- `bool isBool(void)`
- `bool isDouble(void)`
- `bool isNull(void)`
- `bool isObject(void)`
- `bool isString(void)`

- bool isUndefined(void)
- QJsonArray toArray(QJsonArray defaultValue)
- QJsonArray toArray_2(void)
- bool toBool(bool defaultValue)
- double toDouble(double defaultValue)
- int toInt(int defaultValue)
- QJsonObject toObject(QJsonObject defaultValue)
- QJsonObject toObject_2(void)
- QString toString(QString defaultValue)
- QVariant toVariant(void)
- QJsonValue::Type type(void)
- QJsonValue fromVariant(QVariant variant)

99.130 QPlainTextEdit Class

C++ Reference : <http://doc.qt.io/qt-5/QPlainTextEdit.html>

Parameters : QWidget *

Parent Class : QAbstractScrollArea

- QString anchorAt(QPoint pos)
- bool backgroundVisible(void)
- int blockCount(void)
- bool canPaste(void)
- bool centerOnScroll(void)
- QMenu * createStandardContextMenu(void)
- QTextCharFormat currentCharFormat(void)
- QTextCursor cursorForPosition(QPoint pos)
- QRect cursorRect(QTextCursor cursor)
- QRect cursorRect_2(void)
- int cursorWidth(void)
- QTextDocument * document(void)
- QString documentTitle(void)
- void ensureCursorVisible(void)
- QList<QTextEdit::ExtraSelection> extraSelections(void)
- bool find(QString exp, QTextDocument::FindFlags options)
- bool isReadOnly(void)
- bool isUndoRedoEnabled(void)

- `QPlainTextEdit::LineWrapMode lineWrapMode(void)`
- `int maximumBlockCount(void)`
- `void mergeCurrentCharFormat(QTextCharFormat modifier)`
- `void moveCursor(QTextCursor::MoveOperation operation, QTextCursor::MoveMode mode)`
- `bool overwriteMode(void)`
- `void print(QPagedPaintDevice *printer)`
- `void setBackgroundVisible(bool visible)`
- `void setCenterOnScroll(bool enabled)`
- `void setCurrentCharFormat(QTextCharFormat format)`
- `void setCursorWidth(int width)`
- `void setDocument(QTextDocument *document)`
- `void setDocumentTitle(QString title)`
- `void setExtraSelections(QList<QTextEdit::ExtraSelection> selections)`
- `void setLineWrapMode(QPlainTextEdit::LineWrapMode mode)`
- `void setMaximumBlockCount(int maximum)`
- `void setOverwriteMode(bool overwrite)`
- `void setReadOnly(bool ro)`
- `void setTabChangesFocus(bool b)`
- `void setTabStopWidth(int width)`
- `void setTextCursor(QTextCursor cursor)`
- `void setTextInteractionFlags(Qt::TextInteractionFlags flags)`
- `void setUndoRedoEnabled(bool enable)`
- `void setWordWrapMode(QTextOption::WrapMode policy)`
- `bool tabChangesFocus(void)`
- `int tabStopWidth(void)`
- `QTextCursor textCursor(void)`
- `Qt::TextInteractionFlags textInteractionFlags(void)`
- `QString toPlainText(void)`
- `QTextOption::WrapMode wordWrapMode(void)`
- `void appendHtml(QString html)`
- `void appendPlainText(QString text)`
- `void centerCursor(void)`
- `void clear(void)`
- `void copy(void)`
- `void cut(void)`
- `void insertPlainText(QString text)`

- void paste(void)
- void redo(void)
- void selectAll(void)
- void setPlainText(QString text)
- void undo(void)
- void zoomIn(int range)
- void zoomOut(int range)
- void setblockCountChangedEvent(const char *cStr)
- void setcopyAvailableEvent(const char *cStr)
- void setcursorPositionChangedEvent(const char *cStr)
- void setmodificationChangedEvent(const char *cStr)
- void setredoAvailableEvent(const char *cStr)
- void setselectionChangedEvent(const char *cStr)
- void settextChangedEvent(const char *cStr)
- void setundoAvailableEvent(const char *cStr)
- void setupdateRequestEvent(const char *cStr)
- const char *getblockCountChangedEvent(void)
- const char *getcopyAvailableEvent(void)
- const char *getcursorPositionChangedEvent(void)
- const char *getmodificationChangedEvent(void)
- const char *getredoAvailableEvent(void)
- const char *getselectionChangedEvent(void)
- const char *gettextChangedEvent(void)
- const char *getundoAvailableEvent(void)
- const char *getupdateRequestEvent(void)
- void cyanline(void)
- void setactivelinecolor(QColor)

99.131 CodeEditor Class

C++ Reference : <http://doc.qt.io/qt-5/CodeEditor.html>

Parameters : QWidget *

Parent Class : QPlainTextEdit

- void setCompleter(QCompleter *c)
- QCompleter *completer(void)
- void setLineNumbersAreaColor(QColor oColor)

- void setLineNumbersAreaBackColor(QColor oColor)

99.132 QGridLayout Class

C++ Reference : <http://doc.qt.io/qt-5/QGridLayout.html>

Parameters : void

- void addItem(QLayoutItem * item, int row, int column, int rowSpan , int columnSpan , Qt::Alignment alignment)
- void addLayout(QLayout * layout, int row, int column, Qt::Alignment alignment)
- void addLayout_2(QLayout * layout, int row, int column, int rowSpan, int columnSpan, Qt::Alignment alignment)
- void addWidget(QWidget * widget, int row, int column, Qt::Alignment alignment)
- void addWidget_2(QWidget * widget, int fromRow, int fromColumn, int rowSpan, int columnSpan, Qt::Alignment alignment)
- QRect cellRect(int row, int column)
- int columnCount(void)
- int columnMinimumWidth(int column)
- int columnStretch(int column)
- void getItemPosition(int index, int * row, int * column, int * rowSpan, int * columnSpan)
- int horizontalSpacing(void)
- QLayoutItem * itemAtPosition(int row, int column)
- Qt::Corner originCorner(void)
- int rowCount(void)
- int rowMinimumHeight(int row)
- int rowStretch(int row)
- void setColumnMinimumWidth(int column, int minSize)
- void setColumnStretch(int column, int stretch)
- void setHorizontalSpacing(int spacing)
- void setOriginCorner(Qt::Corner corner)
- void setRowMinimumHeight(int row, int minSize)
- void setRowStretch(int row, int stretch)
- void setSpacing(int spacing)
- void setVerticalSpacing(int spacing)
- int spacing(void)
- int verticalSpacing(void)

99.133 QTextCharFormat Class

C++ Reference : <http://doc.qt.io/qt-5/QTextCharFormat.html>

Parameters : void

- QString anchorHref(void)
- QStringList anchorNames(void)
- QFont font(void)
- QFont::Capitalization fontCapitalization(void)
- QString fontFamily(void)
- bool fontFixedPitch(void)
- QFont::HintingPreference fontHintingPreference(void)
- bool fontItalic(void)
- bool fontKerning(void)
- qreal fontLetterSpacing(void)
- QFont::SpacingType fontLetterSpacingType(void)
- bool fontOverline(void)
- qreal fontPointSize(void)
- int fontStretch(void)
- bool fontStrikeOut(void)
- QFont::StyleHint fontStyleHint(void)
- QFont::StyleStrategy fontStyleStrategy(void)
- bool fontUnderline(void)
- int fontWeight(void)
- qreal fontWordSpacing(void)
- bool isAnchor(void)
- bool isValid(void)
- void setAnchor(bool anchor)
- void setAnchorHref(QString value)
- void setAnchorNames(QStringList names)
- void setFontCapitalization(QFont::Capitalization capitalization)
- void setFontFamily(QString family)
- void setFontFixedPitch(bool fixedPitch)
- void setFontHintingPreference(QFont::HintingPreference hintingPreference)
- void setFontItalic(bool italic)
- void setFontKerning(bool enable)
- void setFontLetterSpacing(qreal spacing)

- void setFontLetterSpacingType(QFont::SpacingType letterSpacingType)
- void setFontOverline(bool overline)
- void setFontPointSize(qreal size)
- void setFontStretch(int factor)
- void setFontStrikeOut(bool strikeOut)
- void setFontStyleHint(QFont::StyleHint hint, QFont::StyleStrategy strategy)
- void setFontStyleStrategy(QFont::StyleStrategy strategy)
- void setFontUnderline(bool underline)
- void setFontWeight(int weight)
- void setFontWordSpacing(qreal spacing)
- void setTextOutline(QPen pen)
- void setToolTip(QString text)
- void setUnderlineColor(QColor color)
- void setUnderlineStyle(QTextCharFormat::UnderlineStyle style)
- void setVerticalAlignment(QTextCharFormat::VerticalAlignment alignment)
- QPen textOutline(void)
- QString toolTip(void)
- QColor underlineColor(void)
- QTextCharFormat::UnderlineStyle underlineStyle(void)
- QTextCharFormat::VerticalAlignment verticalAlignment(void)

99.134 QCameraViewfinder Class

C++ Reference : <http://doc.qt.io/qt-5/QCameraViewfinder.html>

Parameters : QWidget *

Parent Class : QVideoWidget

99.135 QGraphicsVideoItem Class

C++ Reference : <http://doc.qt.io/qt-5/QGraphicsVideoItem.html>

Parameters : void

- Qt::AspectRatioMode aspectRatioMode(void)
- QSizeF nativeSize(void)
- QPointF offset(void)
- void setAspectRatioMode(Qt::AspectRatioMode mode)
- void setOffset(QPointF offset)
- void setSize(QSizeF size)

- QSizeF size(void)

99.136 QVideoWidgetControl Class

C++ Reference : <http://doc.qt.io/qt-5/QVideoWidgetControl.html>

Parent Class : QMediaControl

99.137 QCamera Class

C++ Reference : <http://doc.qt.io/qt-5/QCamera.html>

Parameters : void

Parent Class : QMediaObject

- QCamera::CaptureModes captureMode(void)
- QCamera::Error error(void)
- QString errorString(void)
- QCameraExposure * exposure(void)
- QCameraFocus * focus(void)
- QCameraImageProcessing * imageProcessing(void)
- bool isCaptureModeSupported(QCamera::CaptureModes mode)
- QCamera::LockStatus lockStatus(void)
- QCamera::LockStatus lockStatus_2(QCamera::LockType lockType)
- QCamera::LockTypes requestedLocks(void)
- void setViewfinder(QVideoWidget * viewfinder)
- void setViewfinder_2(QGraphicsVideoItem * viewfinder)
- void setViewfinder_3(QAbstractVideoSurface * surface)
- QCamera::State state(void)
- QCamera::Status status(void)
- QCamera::LockTypes supportedLocks(void)
- void load(void)
- void searchAndLock(void)
- void searchAndLock_2(QCamera::LockTypes locks)
- void setCaptureMode(QCamera::CaptureModes mode)
- void start(void)
- void stop(void)
- void unload(void)
- void unlock(void)
- void unlock_2(QCamera::LockTypes locks)

99.138 QCameraImageCapture Class

C++ Reference : <http://doc.qt.io/qt-5/QCameraImageCapture.html>

Parameters : QMediaObject * mediaObject

- QMultimedia::AvailabilityStatus availability(void)
- QVideoFrame::PixelFormat bufferFormat(void)
- QCameraImageCapture::CaptureDestinations captureDestination(void)
- QImageEncoderSettings encodingSettings(void)
- QCameraImageCapture::Error error(void)
- QString errorString(void)
- QString imageCodecDescription(QString codec)
- bool isAvailable(void)
- bool isCaptureDestinationSupported(QCameraImageCapture::CaptureDestinations destination)
- bool isReadyForCapture(void)
- void setBufferFormat(QVideoFrame::PixelFormat format)
- void setCaptureDestination(QCameraImageCapture::CaptureDestinations destination)
- void setEncodingSettings(QImageEncoderSettings settings)
- QList<QVideoFrame::PixelFormat> supportedBufferFormats(void)
- QStringList supportedImageCodecs(void)
- QList<QSize> supportedResolutions(QImageEncoderSettings settings , bool * continuous)
- void cancelCapture(void)
- int capture(QString file)

99.139 QMediaObject Class

C++ Reference : <http://doc.qt.io/qt-5/QMediaObject.html>

Parameters : void

Parent Class : QWidget

- QStringList availableMetaData(void)
- bool isMetaDataAvailable(void)
- QVariant metaData(QString key)
- int notifyInterval(void)
- void setNotifyInterval(int milliseconds)

99.140 QHeaderView Class

C++ Reference : <http://doc.qt.io/qt-5/QHeaderView.html>

Parameters : Qt::Orientation, QWidget *

Parent Class : QAbstractItemView

- bool cascadingSectionResizes(void)
- int count(void)
- Qt::Alignment defaultAlignment(void)
- int defaultSectionSize(void)
- int hiddenSectionCount(void)
- void hideSection(int logicalIndex)
- bool highlightSections(void)
- bool isSectionHidden(int logicalIndex)
- bool isSortIndicatorShown(void)
- int length(void)
- int logicalIndex(int visualIndex)
- int logicalIndexAt(int position)
- int logicalIndexAt_2(int x, int y)
- int logicalIndexAt_3(QPoint pos)
- int maximumSectionSize(void)
- int minimumSectionSize_2(void)
- void moveSection(int from, int to)
- int offset(void)
- Qt::Orientation orientation(void)
- int resizeContentsPrecision(void)
- void resizeSection(int logicalIndex, int size)
- void resizeSections(QHeaderView::ResizeMode mode)
- bool restoreState(QByteArray state)
- QByteArray saveState(void)
- int sectionPosition(int logicalIndex)
- QHeaderView::ResizeMode sectionResizeMode(int logicalIndex)
- int sectionSize(int logicalIndex)
- int sectionSizeHint(int logicalIndex)
- int sectionViewportPosition(int logicalIndex)
- bool sectionsClickable(void)
- bool sectionsHidden(void)

- bool sectionsMovable(void)
- bool sectionsMoved(void)
- void setCascadingSectionResizes(bool enable)
- void setDefaultAlignment(Qt::Alignment alignment)
- void setDefaultSectionSize(int size)
- void setHighlightSections(bool highlight)
- void setMaximumSectionSize(int size)
- void setMinimumSectionSize(int size)
- void setResizeContentsPrecision(int precision)
- void setSectionHidden(int logicalIndex, bool hide)
- void setSectionResizeMode(QHeaderView::ResizeMode mode)
- void setSectionResizeMode_2(int logicalIndex, QHeaderView::ResizeMode mode)
- void setSectionsClickable(bool clickable)
- void setSectionsMovable(bool movable)
- void setSortIndicator(int logicalIndex, Qt::SortOrder order)
- void setSortIndicatorShown(bool show)
- void setStretchLastSection(bool stretch)
- void showSection(int logicalIndex)
- Qt::SortOrder sortIndicatorOrder(void)
- int sortIndicatorSection(void)
- bool stretchLastSection(void)
- int stretchSectionCount(void)
- void swapSections(int first, int second)
- int visualIndex(int logicalIndex)
- int visualIndexAt(int position)
- void headerDataChanged(Qt::Orientation orientation, int logicalFirst, int logicalLast)
- void setOffset(int offset)
- void setOffsetToLastSection(void)
- void setOffsetToSectionPosition(int visualSectionNumber)
- void setgeometriesChangedEvent(const char *)
- void setsectionClickedEvent(const char *)
- void setsectionCountChangedEvent(const char *)
- void setsectionDoubleClickedEvent(const char *)
- void setsectionEnteredEvent(const char *)
- void setsectionHandleDoubleClickedEvent(const char *)
- void setsectionMovedEvent(const char *)

- void setsectionPressedEvent(const char *)
- void setsectionResizedEvent(const char *)
- void setsortIndicatorChangedEvent(const char *)
- const char *getgeometriesChangedEvent(void)
- const char *getsectionClickedEvent(void)
- const char *getsectionCountChangedEvent(void)
- const char *getsectionDoubleClickedEvent(void)
- const char *getsectionEnteredEvent(void)
- const char *getsectionHandleDoubleClickedEvent(void)
- const char *getsectionMovedEvent(void)
- const char *getsectionPressedEvent(void)
- const char *getsectionResizedEvent(void)
- const char *getsortIndicatorChangedEvent(void)
- void geteventparameters(void)

99.141 QFontMetrics Class

C++ Reference : <http://doc.qt.io/qt-5/QFontMetrics.html>

Parameters : QFont

- int ascent(void)
- int averageCharWidth(void)
- QRect boundingRect(QChar ch)
- QRect boundingRect_2(QString text)
- QRect boundingRect_3(int x, int y, int width, int height, int flags, QString text, int tabStops , int * tabArray)
- QRect boundingRect_4(QRect rect, int flags, QString text, int tabStops , int * tabArray)
- int descent(void)
- QString elidedText(QString text, Qt::TextElideMode mode, int width, int flags)
- int height(void)
- bool inFont(QChar ch)
- bool inFontUcs4(uint character)
- int leading(void)
- int leftBearing(QChar ch)
- int lineSpacing(void)
- int lineWidth(void)
- int maxWidth(void)
- int minLeftBearing(void)

- int minRightBearing(void)
- int overlinePos(void)
- int rightBearing(QChar ch)
- QSize size(int flags, QString text, int tabStops , int * tabArray)
- int strikeOutPos(void)
- QRect tightBoundingRect(QString text)
- int underlinePos(void)
- int width(QString text, int len)
- int width_2(QChar ch)
- int xHeight(void)

99.142 QSplashScreen Class

C++ Reference : <http://doc.qt.io/qt-5/QSplashScreen.html>

Parameters : QPixmap

Parent Class : QWidget

- void finish(QWidget *mainWin)
- QPixmap pixmap(void)
- void repaint(void)
- void setPixmap(QPixmap pixmap)
- void clearMessage(void)
- void showMessage(QString message, int alignment ,QColor color)

99.143 QBoxLayout Class

C++ Reference : <http://doc.qt.io/qt-5/QBoxLayout.html>

Parameters : QBoxLayout::Direction dir, QWidget *parent

Parent Class : QLayout

- void addLayout(QLayout * layout, int stretch)
- void addSpacerItem(QSpacerItem * spacerItem)
- void addSpacing(int size)
- void addStretch(int stretch)
- void addStrut(int size)
- void addWidget(QWidget * widget, int stretch , Qt::Alignment alignment)
- QBoxLayout::Direction direction(void)
- void insertLayout(int index, QLayout * layout, int stretch)
- void insertSpacerItem(int index, QSpacerItem * spacerItem)

- void insertSpacing(int index, int size)
- void insertStretch(int index, int stretch)
- void insertWidget(int index, QWidget * widget, int stretch , Qt::Alignment alignment)
- void setDirection(QBoxLayout::Direction direction)
- void setSpacing(int spacing)
- void setStretch(int index, int stretch)
- bool setStretchFactor(QWidget * widget, int stretch)
- bool setStretchFactor_2(QLayout * layout, int stretch)
- int spacing(void)
- int stretch(int index)

99.144 QLayout Class

C++ Reference : <http://doc.qt.io/qt-5/QLayout.html>

Parameters : QWidget *

Parent Class : QObject

- bool activate(void)
- void addWidget(QWidget *w)
- QMargins contentsMargins(void)
- QRect contentsRect(void)
- void getContentsMargins(int *left, int *top, int *right, int *bottom)
- bool isEnabled(void)
- QWidget *menuBar(void)
- QWidget *parentWidget(void)
- void removeItem(QLayoutItem *item)
- void removeWidget(QWidget *widget)
- bool setAlignment(QWidget *w, Qt::Alignment alignment)
- void setAlignment_2(Qt::Alignment alignment)
- bool setAlignment_3(QLayout *l, Qt::Alignment alignment)
- void setContentsMargins(int left, int top, int right, int bottom)
- void setContentsMargins_2(QMargins margins)
- void setEnabled(bool enable)
- void setMenuBar(QWidget *widget)
- void setSizeConstraint(QLayout::SizeConstraint)
- void setSpacing(int)
- QLayout::SizeConstraint sizeConstraint(void)

- `int spacing(void)`
- `void update(void)`
- `QSize closestAcceptableSize(QWidget * widget, QSize size)`

99.145 QLinearGradient Class

C++ Reference : <http://doc.qt.io/qt-5/QLinearGradient.html>

Parameters : void

Parent Class : QGradient

- `QPointF finalStop(void)`
- `void setFinalStop(QPointF stop)`
- `void setFinalStop_2(qreal x,qreal y)`
- `void setStart(QPointF start)`
- `void setStart_2(qreal x,qreal y)`
- `QPointF start(void)`

99.146 QGradient Class

C++ Reference : <http://doc.qt.io/qt-5/QGradient.html>

Parameters : void

- `QGradient::CoordinateMode coordinateMode(void)`
- `void setColorAt(qreal position, QColor color)`
- `void setCoordinateMode(QGradient::CoordinateMode mode)`
- `void setSpread(QGradient::Spread method)`
- `void setStops(QGradientStops stopPoints)`
- `QGradient::Spread spread(void)`
- `QGradientStops stops(void)`
- `QGradient::Type type(void)`

99.147 QPointF Class

C++ Reference : <http://doc.qt.io/qt-5/QPointF.html>

Parameters : void

- `bool isNull(void)`
- `qreal manhattanLength(void)`
- `qreal rx(void)`
- `qreal ry(void)`

- void setX(qreal x)
- void setY(qreal y)
- QPoint toPoint(void)
- qreal x(void)
- qreal y(void)

99.148 QPoint Class

C++ Reference : <http://doc.qt.io/qt-5/QPoint.html>

Parameters : void

- bool isNull(void)
- int manhattanLength(void)
- int rx(void)
- int ry(void)
- void setX(int x)
- void setY(int y)
- int x(void)
- int y(void)

99.149 QScrollArea Class

C++ Reference : <http://doc.qt.io/qt-5/QScrollArea.html>

Parameters : QWidget *parent

Parent Class : QAbstractScrollArea

- Qt::Alignment alignment(void)
- void ensureVisible(int x, int y, int xmargin , int ymargin)
- void ensureWidgetVisible(QWidget *childWidget, int xmargin , int ymargin)
- void setAlignment(Qt::Alignment)
- void setWidget(QWidget *widget)
- void setWidgetResizable(bool resizable)
- QWidget *takeWidget(void)
- QWidget *widget(void)
- bool widgetResizable(void)

99.150 QSplitter Class

C++ Reference : <http://doc.qt.io/qt-5/QSplitter.html>

Parameters : QWidget *parent

Parent Class : QFrame

- void addWidget(QWidget *widget)
- bool childrenCollapsible(void)
- int count(void)
- void getRange(int index, int *min, int *max)
- QSplitterHandle * handle(int index)
- int handleWidth(void)
- int indexOf(QWidget *widget)
- void insertWidget(int index, QWidget *widget)
- bool isCollapsible(int index)
- bool opaqueResize(void)
- Qt::Orientation orientation(void)
- void refresh(void)
- bool restoreState(QByteArray state)
- QByteArray saveState(void)
- void setChildrenCollapsible(bool)
- void setCollapsible(int index, bool collapse)
- void setHandleWidth(int)
- void setOpaqueResize(bool opaque)
- void setOrientation(Qt::Orientation)
- void setSizes(QList<int> list)
- void setStretchFactor(int index, int stretch)
- QList<int> sizes(void)
- QWidget * widget(int index)

99.151 QCompleter Class

C++ Reference : <http://doc.qt.io/qt-5/QCompleter.html>

Parameters : QObject *parent

Parent Class : QObject

- Qt::CaseSensitivity caseSensitivity(void)
- int completionColumn(void)
- int completionCount(void)

- `QCompleter::CompletionMode completionMode(void)`
- `QAbstractItemModel *completionModel(void)`
- `QString completionPrefix(void)`
- `int completionRole(void)`
- `QString currentCompletion(void)`
- `QModelIndex currentIndex(void)`
- `int currentRow(void)`
- `Qt::MatchFlags filterMode(void)`
- `int maxVisibleItems(void)`
- `QAbstractItemModel * model(void)`
- `QCompleter::ModelSorting modelSorting(void)`
- `QAbstractItemView * popup(void)`
- `void setCaseSensitivity(Qt::CaseSensitivity caseSensitivity)`
- `void setCompletionColumn(int column)`
- `void setCompletionMode(QCompleter::CompletionMode mode)`
- `void setCompletionRole(int role)`
- `bool setCurrentRow(int row)`
- `void setFilterMode(Qt::MatchFlags filterMode)`
- `void setMaxVisibleItems(int maxItems)`
- `void setModel(QAbstractItemModel *model)`
- `void setModelSorting(QCompleter::ModelSorting sorting)`
- `void setPopup(QAbstractItemView *popup)`
- `void setWidget(QWidget *widget)`
- `QWidget * widget(void)`
- `bool wrapAround(void)`
- `void complete(QRect rect)`
- `void setCompletionPrefix(QString prefix)`
- `void setWrapAround(bool wrap)`

99.152 QCompleter2 Class

C++ Reference : <http://doc.qt.io/qt-5/QCompleter2.html>

Parameters : `QAbstractItemModel *model`, `QObject *parent`

Parent Class : `QCompleter`

99.153 QCompleter3 Class

C++ Reference : <http://doc.qt.io/qt-5/QCompleter3.html>

Parameters : QStringList list, QObject *parent

Parent Class : QCompleter

99.154 QString2 Class

C++ Reference : <http://doc.qt.io/qt-5/QString2.html>

Parameters : void

- QString append(QString str)
- QStringList split(QString sep, QString::SplitBehavior behavior , Qt::CaseSensitivity cs)
- QStringList split_2(QChar sep, QString::SplitBehavior behavior , Qt::CaseSensitivity cs)
- QStringList split_3(QRegExp rx, QString::SplitBehavior behavior)
- QStringList split_4(QRegularExpression re, QString::SplitBehavior behavior)

99.155 QProcess Class

C++ Reference : <http://doc.qt.io/qt-5/QProcess.html>

Parameters : QObject *

Parent Class : QIODevice

- QStringList arguments(void)
- void closeReadChannel(QProcess::ProcessChannel channel)
- void closeWriteChannel(void)
- QProcess::ProcessError error(void)
- int exitCode(void)
- QProcess::ExitStatus exitStatus(void)
- QProcess::InputChannelMode inputChannelMode(void)
- QProcess::ProcessChannelMode processChannelMode(void)
- QProcessEnvironment processEnvironment(void)
- QString program(void)
- QByteArray readAllStandardError(void)
- QByteArray readAllStandardOutput(void)
- QProcess::ProcessChannel readChannel(void)
- void setArguments(QStringList arguments)
- void setInputChannelMode(QProcess::InputChannelMode mode)
- void setProcessChannelMode(QProcess::ProcessChannelMode mode)

- void setProcessEnvironment(QProcessEnvironment environment)
- void setProgram(QString program)
- void setReadChannel(QProcess::ProcessChannel channel)
- void setStandardErrorFile(QString fileName, QIODevice::OpenMode mode)
- void setStandardInputFile(QString fileName)
- void setStandardOutputFile(QString fileName, QIODevice::OpenMode mode)
- void setStandardOutputProcess(QProcess *destination)
- void setWorkingDirectory(QString dir)
- void start(QString program, QStringList arguments, QIODevice::OpenMode mode)
- void start_2(QString command, QIODevice::OpenMode mode)
- void start_3(QIODevice::OpenMode mode)
- QProcess::ProcessState state(void)
- bool waitForFinished(int msec)
- bool waitForStarted(int msec)
- QString workingDirectory(void)
- void kill(void)
- void terminate(void)
- void setreadyReadStandardErrorEvent(const char *)
- void setreadyReadStandardOutputEvent(const char *)
- const char *getreadyReadStandardErrorEvent(void)
- const char *getreadyReadStandardOutputEvent(void)

99.156 QMdiArea Class

C++ Reference : <http://doc.qt.io/qt-5/QMdiArea.html>

Parameters : QWidget *

Parent Class : QAbstractScrollArea

- QMdiArea::WindowOrder activationOrder(void)
- QMdiSubWindow * activeSubWindow(void)
- QMdiSubWindow * addSubWindow(QWidget * widget, Qt::WindowFlags windowFlags)
- QBrush background(void)
- QMdiSubWindow * currentSubWindow(void)
- bool documentMode(void)
- void removeSubWindow(QWidget * widget)
- void setActivationOrder(QMdiArea::WindowOrder order)
- void setBackground(QBrush background)

- void setDocumentMode(bool enabled)
- void setOption(QMdiArea::AreaOption option, bool on)
- void setTabPosition(QTabWidget::TabPosition position)
- void setTabShape(QTabWidget::TabShape shape)
- void setTabsClosable(bool closable)
- void setTabsMovable(bool movable)
- void setViewMode(QMdiArea::ViewMode mode)
- QList<QMdiSubWindow *> subWindowList(QMdiArea::WindowOrder order)
- QTabWidget::TabPosition tabPosition(void)
- QTabWidget::TabShape tabShape(void)
- bool tabsClosable(void)
- bool tabsMovable(void)
- bool testOption(QMdiArea::AreaOption option)
- QMdiArea::ViewMode viewMode(void)
- void activateNextSubWindow(void)
- void activatePreviousSubWindow(void)
- void cascadeSubWindows(void)
- void closeActiveSubWindow(void)
- void closeAllSubWindows(void)
- void setActiveSubWindow(QMdiSubWindow * window)
- void tileSubWindows(void)

99.157 QMdiSubWindow Class

C++ Reference : <http://doc.qt.io/qt-5/QMdiSubWindow.html>

Parameters : QWidget *

Parent Class : QWidget

- bool isShaded(void)
- int keyboardPageStep(void)
- int keyboardSingleStep(void)
- QMdiArea * mdiArea(void)
- void setKeyboardPageStep(int step)
- void setKeyboardSingleStep(int step)
- void setOption(QMdiSubWindow::SubWindowOption option, bool on)
- void setSystemMenu(QMenu * systemMenu)
- void setWidget(QWidget * widget)

- `QMenu * systemMenu(void)`
- `bool testOption(QMdiSubWindow::SubWindowOption option)`
- `QWidget * widget(void)`
- `void showShaded(void)`
- `void showSystemMenu(void)`

99.158 QCursor Class

C++ Reference : <http://doc.qt.io/qt-5/QCursor.html>

Parameters : void

- `QBitmap *bitmap(void)`
- `QPoint hotSpot(void)`
- `QBitmap *mask(void)`
- `QPixmap pixmap(void)`
- `void setShape(Qt::CursorShape shape)`
- `Qt::CursorShape shape(void)`
- `QPoint pos(void)`
- `QPoint pos_2(QScreen *)`
- `void setPos(int x, int y)`
- `void setPos_2(QScreen *screen, int x, int y)`
- `void setPos_3(QPoint)`
- `void setPos_4(QScreen *screen, QPoint)`

99.159 QListView Class

C++ Reference : <http://doc.qt.io/qt-5/QListView.html>

Parameters : `QWidget *`

Parent Class : `QAbstractItemView`

- `int batchSize(void)`
- `void clearPropertyFlags(void)`
- `QListView::Flow flow(void)`
- `QSize gridSize(void)`
- `bool isRowHidden(int row)`
- `bool isSelectionRectVisible(void)`
- `bool isWrapping(void)`
- `QListView::LayoutMode layoutMode(void)`
- `int modelColumn(void)`

- `QListView::Movement movement(void)`
- `QListView::ResizeMode resizeMode(void)`
- `void setBatchSize(int batchSize)`
- `void setFlow(QListView::Flow flow)`
- `void setGridSize(QSize size)`
- `void setLayoutMode(QListView::LayoutMode mode)`
- `void setModelColumn(int column)`
- `void setMovement(QListView::Movement movement)`
- `void setResizeMode(QListView::ResizeMode mode)`
- `void setRowHidden(int row, bool hide)`
- `void setSelectionRectVisible(bool show)`
- `void setSpacing(int space)`
- `void setUniformItemSizes(bool enable)`
- `void setViewMode(QListView::ViewMode mode)`
- `void setWordWrap(bool on)`
- `void setWrapping(bool enable)`
- `int spacing(void)`
- `bool uniformItemSizes(void)`
- `QListView::ViewMode viewMode(void)`
- `bool wordWrap(void)`

99.160 QAxObject Class

C++ Reference : <http://doc.qt.io/qt-5/QAxObject.html>

Parameters : `QString`

Parent Class : `QAxBase`

99.161 QAxBase Class

C++ Reference : <http://doc.qt.io/qt-5/QAxBase.html>

Parameters : `QWidget *`

Parent Class : `QObject`

- `QVariant asVariant(void)`
- `QString control(void)`
- `void disableClassInfo(void)`
- `void disableEventSink(void)`
- `void disableMetaObject(void)`

- QVariant dynamicCall(char *function)
- QVariant dynamicCall_2(char *function,QString)
- QString generateDocumentation(void)
- bool isNull(void)
- QAxObject * querySubObject(char *name)
- bool setControl(QString)
- QStringList verbs(void)

99.162 QUuid Class

C++ Reference : <http://doc.qt.io/qt-5/QUuid.html>

Parameters : void

- QString toString(void)

99.163 QToolButton Class

C++ Reference : <http://doc.qt.io/qt-5/QToolButton.html>

Parameters : QWidget *

Parent Class : QAbstractButton

- Qt::ArrowType arrowType(void)
- bool autoRaise(void)
- QAction * defaultAction(void)
- QMenu * menu(void)
- QToolButton::ToolButtonPopupMode popupMode(void)
- void setArrowType(Qt::ArrowType type)
- void setAutoRaise(bool enable)
- void setMenu(QMenu * menu)
- void setPopupMode(QToolButton::ToolButtonPopupMode mode)
- Qt::ToolButtonStyle toolButtonStyle(void)
- void setDefaultAction(QAction * action)
- void setToolButtonStyle(Qt::ToolButtonStyle style)
- void showMenu(void)
- void settriggeredEvent(const char *)
- const char *gettriggeredEvent(void)
- void setClickEvent(const char *)
- const char *getClickEvent(void)

99.164 QSerialPort Class

C++ Reference : <http://doc.qt.io/qt-5/QSerialPort.html>

Parameters : QObject *

Parent Class : QIODevice

- qint32 baudRate(QSerialPort::Directions directions)
- bool clear(QSerialPort::Directions directions)
- void clearError(void)
- QSerialPort::DataBits dataBits(void)
- QSerialPort::SerialPortError error(void)
- QSerialPort::FlowControl flowControl(void)
- bool flush(void)
- void *handle(void)
- bool isDataTerminalReady(void)
- bool isRequestToSend(void)
- QSerialPort::Parity parity(void)
- QSerialPort::PinoutSignals pinoutSignals(void)
- QString portName(void)
- qint64 readBufferSize(void)
- bool setBaudRate(qint32 baudRate, QSerialPort::Directions directions)
- bool setBreakEnabled(bool set)
- bool setDataBits(QSerialPort::DataBits dataBits)
- bool setDataTerminalReady(bool set)
- bool setFlowControl(QSerialPort::FlowControl flowControl)
- bool setParity(QSerialPort::Parity parity)
- void setPort(QSerialPortInfo serialPortInfo)
- void setPortName(QString name)
- void setReadBufferSize(qint64 size)
- bool setRequestToSend(bool set)
- bool setStopBits(QSerialPort::StopBits stopBits)
- QSerialPort::StopBits stopBits(void)
- void setbaudRateChangedEvent(const char *)
- void setbreakEnabledChangedEvent(const char *)
- void setdataBitsChangedEvent(const char *)
- void setdataTerminalReadyChangedEvent(const char *)
- void seterrorEvent(const char *)

- void setflowControlChangedEvent(const char *)
- void setparityChangedEvent(const char *)
- void setrequestToSendChangedEvent(const char *)
- void setstopBitsChangedEvent(const char *)
- const char *getbaudRateChangedEvent(void)
- const char *getbreakEnabledChangedEvent(void)
- const char *getdataBitsChangedEvent(void)
- const char *getdataTerminalReadyChangedEvent(void)
- const char *getErrorEvent(void)
- const char *getflowControlChangedEvent(void)
- const char *getparityChangedEvent(void)
- const char *getrequestToSendChangedEvent(void)
- const char *getstopBitsChangedEvent(void)

99.165 QSerialPortInfo Class

C++ Reference : <http://doc.qt.io/qt-5/QSerialPortInfo.html>

Parameters : void

- QString description(void)
- bool hasProductIdentifier(void)
- bool hasVendorIdentifier(void)
- bool isBusy(void)
- bool isNull(void)
- QString manufacturer(void)
- QString portName(void)
- quint16 productIdentifier(void)
- void swap(QSerialPortInfo other)
- QString systemLocation(void)
- quint16 vendorIdentifier(void)

99.166 QStringRef Class

C++ Reference : <http://doc.qt.io/qt-5/QStringRef.html>

Parameters : void

- QStringRef appendTo(QString * string)
- QChar at(int position)
- void clear(void)

- `int compare(QString other, Qt::CaseSensitivity cs)`
- `int compare_2(QStringRef other, Qt::CaseSensitivity cs)`
- `int compare_3(QLatin1String other, Qt::CaseSensitivity cs)`
- `QChar * constData(void)`
- `bool contains(QString str, Qt::CaseSensitivity cs)`
- `bool contains_2(QChar ch, Qt::CaseSensitivity cs)`
- `bool contains_3(QStringRef str, Qt::CaseSensitivity cs)`
- `bool contains_4(QLatin1String str, Qt::CaseSensitivity cs)`
- `int count(void)`
- `int count_2(QString str, Qt::CaseSensitivity cs)`
- `int count_3(QChar ch, Qt::CaseSensitivity cs)`
- `int count_4(QStringRef str, Qt::CaseSensitivity cs)`
- `QChar * data(void)`
- `bool endsWith(QString str, Qt::CaseSensitivity cs)`
- `bool endsWith_2(QChar ch, Qt::CaseSensitivity cs)`
- `bool endsWith_3(QLatin1String str, Qt::CaseSensitivity cs)`
- `bool endsWith_4(QStringRef str, Qt::CaseSensitivity cs)`
- `int indexOf(QString str, int from, Qt::CaseSensitivity cs)`
- `int indexOf_2(QLatin1String str, int from, Qt::CaseSensitivity cs)`
- `int indexOf_3(QChar ch, int from, Qt::CaseSensitivity cs)`
- `int indexOf_4(QStringRef str, int from, Qt::CaseSensitivity cs)`
- `bool isEmpty(void)`
- `bool isNull(void)`
- `int lastIndexOf(QString str, int from, Qt::CaseSensitivity cs)`
- `int lastIndexOf_2(QChar ch, int from, Qt::CaseSensitivity cs)`
- `int lastIndexOf_3(QLatin1String str, int from, Qt::CaseSensitivity cs)`
- `int lastIndexOf_4(QStringRef str, int from, Qt::CaseSensitivity cs)`
- `int length(void)`
- `int localeAwareCompare(QString other)`
- `int localeAwareCompare_2(QStringRef other)`
- `int position(void)`
- `int size(void)`
- `bool startsWith(QString str, Qt::CaseSensitivity cs)`
- `bool startsWith_2(QLatin1String str, Qt::CaseSensitivity cs)`
- `bool startsWith_3(QStringRef str, Qt::CaseSensitivity cs)`
- `bool startsWith_4(QChar ch, Qt::CaseSensitivity cs)`

- QString * string(void)
- QByteArray toLatin1(void)
- QByteArray toLocal8Bit(void)
- QString toString(void)
- QVector<uint> toUcs4(void)
- QByteArray toUtf8(void)
- QChar * unicode(void)
- int compare_4(QStringRef s1, QString s2, Qt::CaseSensitivity cs)
- int compare_5(QStringRef s1, QStringRef s2, Qt::CaseSensitivity cs)
- int compare_6(QStringRef s1, QLatin1String s2, Qt::CaseSensitivity cs)
- int localeAwareCompare_3(QStringRef s1, QString s2)
- int localeAwareCompare_4(QStringRef s1, QStringRef s2)

99.167 QMutex Class

C++ Reference : <http://doc.qt.io/qt-5/QMutex.html>

Parameters : QMutex::RecursionMode

- bool isRecursive(void)
- void lock(void)
- void unlock(void)

99.168 QMutexLocker Class

C++ Reference : <http://doc.qt.io/qt-5/QMutexLocker.html>

Parameters : QMutex *

- QMutex * mutex(void)
- void relock(void)
- void unlock(void)

99.169 QBuffer Class

C++ Reference : <http://doc.qt.io/qt-5/QBuffer.html>

Parameters : QObject *

Parent Class : QIODevice

- QByteArray buffer(void)
- QByteArray data(void)
- void setBuffer(QByteArray *byteArray)

- void setData(QByteArray data)
- void setData_2(char *data, int size)

99.170 QBluetoothAddress Class

C++ Reference : <http://doc.qt.io/qt-5/QBluetoothAddress.html>

Parameters : void

- void clear(void)
- bool isNull(void)
- QString toString(void)
- quint64 toUInt64(void)

99.171 QBluetoothDeviceDiscoveryAgent Class

C++ Reference : <http://doc.qt.io/qt-5/QBluetoothDeviceDiscoveryAgent.html>

Parameters : QObject *

- QList<QBluetoothDeviceInfo> discoveredDevices(void)
- QBluetoothDeviceDiscoveryAgent::Error error(void)
- QString errorString(void)
- QBluetoothDeviceDiscoveryAgent::InquiryType inquiryType(void)
- bool isActive(void)
- void setInquiryType(QBluetoothDeviceDiscoveryAgent::InquiryType type)
- void start(void)
- void stop(void)
- void setCanceledEvent(const char *)
- void setDeviceDiscoveredEvent(const char *)
- void setErrorEvent(const char *)
- void setFinishedEvent(const char *)
- const char *getCanceledEvent(void)
- const char *getDeviceDiscoveredEvent(void)
- const char *getErrorEvent(void)
- const char *getFinishedEvent(void)

99.172 QBluetoothDeviceInfo Class

C++ Reference : <http://doc.qt.io/qt-5/QBluetoothDeviceInfo.html>

Parameters : void

- `QBluetoothAddress address(void)`
- `bool isValid(void)`
- `QBluetoothDeviceInfo::MajorDeviceClass majorDeviceClass(void)`
- `quint8 minorDeviceClass(void)`
- `QString name(void)`
- `qint16 rssi(void)`
- `QBluetoothDeviceInfo::ServiceClasses serviceClasses(void)`
- `QList<QBluetoothUuid> serviceUuids(QBluetoothDeviceInfo::DataCompleteness *completeness)`
- `QBluetoothDeviceInfo::DataCompleteness serviceUuidsCompleteness(void)`
- `void setCached(bool cached)`
- `void setServiceUuids(QList<QBluetoothUuid> uuids, QBluetoothDeviceInfo::DataCompleteness completeness)`

99.173 QBluetoothHostInfo Class

C++ Reference : <http://doc.qt.io/qt-5/QBluetoothHostInfo.html>

Parameters : void

- `QBluetoothAddress address(void)`
- `QString name(void)`
- `void setAddress(QBluetoothAddress address)`
- `void setName(QString name)`

99.174 QBluetoothLocalDevice Class

C++ Reference : <http://doc.qt.io/qt-5/QBluetoothLocalDevice.html>

Parameters : `QObject *`

- `QBluetoothAddress address(void)`

99.175 QDateTime Class

C++ Reference : <http://doc.qt.io/qt-5/QDateTime.html>

Parameters : void

- `QDateTime addDays(int ndays)`
- `QDateTime addMsecs(qint64 msecs)`
- `QDateTime addMonths(int nmonths)`
- `QDateTime addSecs(int s)`
- `QDateTime addYears(int nyears)`

- `QDate date(void)`
- `int daysTo(QDateTime other)`
- `bool isNull(void)`
- `bool isValid(void)`
- `qint64 msecsTo(QDateTime other)`
- `int secsTo(QDateTime other)`
- `void setDate(QDate date)`
- `void setMSecsSinceEpoch(qint64 msecs)`
- `void setTime(QTime time)`
- `void setTimeSpec(Qt::TimeSpec spec)`
- `void setTime_t(uint seconds)`
- `QTime time(void)`
- `Qt::TimeSpec timeSpec(void)`
- `QDateTime toLocalTime(void)`
- `qint64 toMSecsSinceEpoch(void)`
- `QString toString(QString format)`
- `QString toString_2(Qt::DateFormat format)`
- `QDateTime toTimeSpec(Qt::TimeSpec specification)`
- `uint toTime_t(void)`
- `QDateTime toUTC(void)`
- `QDateTime currentDateTime(void)`
- `QDateTime currentDateTimeUtc(void)`
- `qint64 currentMSecsSinceEpoch(void)`
- `QDateTime fromMSecsSinceEpoch(qint64 msecs)`
- `QDateTime fromString(QString string, Qt::DateFormat format)`
- `QDateTime fromString_2(QString string, QString format)`
- `QDateTime fromTime_t(uint seconds)`

99.176 QScreen Class

C++ Reference : <http://doc.qt.io/qt-5/QScreen.html>

- `int angleBetween(Qt::ScreenOrientation a, Qt::ScreenOrientation b)`
- `QRect availableGeometry(void)`
- `QSize availableSize(void)`
- `QRect availableVirtualGeometry(void)`
- `QSize availableVirtualSize(void)`

- `int depth(void)`
- `qreal devicePixelRatio(void)`
- `QRect geometry(void)`
- `QPixmap grabWindow(int window, int x, int y, int width, int height)`
- `QPixmap grabWindow_2(int window)`
- `QPlatformScreen * handle(void)`
- `bool isLandscape(Qt::ScreenOrientation o)`
- `bool isPortrait(Qt::ScreenOrientation o)`
- `qreal logicalDotsPerInch(void)`
- `qreal logicalDotsPerInchX(void)`
- `qreal logicalDotsPerInchY(void)`
- `QRect mapBetween(Qt::ScreenOrientation a, Qt::ScreenOrientation b, QRect rect)`
- `QString name(void)`
- `Qt::ScreenOrientation nativeOrientation(void)`
- `Qt::ScreenOrientation orientation(void)`
- `Qt::ScreenOrientations orientationUpdateMask(void)`
- `qreal physicalDotsPerInch(void)`
- `qreal physicalDotsPerInchX(void)`
- `qreal physicalDotsPerInchY(void)`
- `QSizeF physicalSize(void)`
- `Qt::ScreenOrientation primaryOrientation(void)`
- `qreal refreshRate(void)`
- `void setOrientationUpdateMask(Qt::ScreenOrientations mask)`
- `QSize size(void)`
- `QTransform transformBetween(Qt::ScreenOrientation a, Qt::ScreenOrientation b, QRect target)`

99.177 QWindow Class

C++ Reference : <http://doc.qt.io/qt-5/QWindow.html>

Parameters : `QScreen *`

Parent Class : `QObject`

- `QSize baseSize(void)`
- `Qt::ScreenOrientation contentOrientation(void)`
- `void create(void)`
- `QCursor cursor(void)`
- `void destroy(void)`

- qreal devicePixelRatio(void)
- QString filePath(void)
- Qt::WindowFlags flags(void)
- QObject * focusObject(void)
- QRect frameGeometry(void)
- QMargins frameMargins(void)
- QPoint framePosition(void)
- QRect geometry(void)
- int height(void)
- QIcon icon(void)
- bool isActive(void)
- bool isAncestorOf(QWindow *child, QWindow::AncestorMode mode)
- bool isExposed(void)
- bool isModal(void)
- bool isTopLevel(void)
- bool isVisible(void)
- QPoint mapFromGlobal(QPoint pos)
- QPoint mapToGlobal(QPoint pos)
- QRegion mask(void)
- int maximumHeight(void)
- QSize maximumSize(void)
- int maximumWidth(void)
- int minimumHeight(void)
- QSize minimumSize(void)
- int minimumWidth(void)
- Qt::WindowModality modality(void)
- qreal opacity(void)
- QPoint position(void)
- void reportContentOrientationChange(Qt::ScreenOrientation orientation)
- QSurfaceFormat requestedFormat(void)
- void resize(QSize newSize)
- void resize_2(int w, int h)
- QScreen * screen(void)
- void setBaseSize(QSize size)
- void setCursor(QCursor cursor)
- void setFilePath(QString filePath)

- void setFlags(Qt::WindowFlags flags)
- void setFormat(QSurfaceFormat format)
- void setFramePosition(QPoint point)
- void setGeometry(int posX, int posY, int w, int h)
- void setGeometry_2(QRect rect)
- void setIcon(QIcon icon)
- bool setKeyboardGrabEnabled(bool grab)
- void setMask(QRegion region)
- void setMaximumSize(QSize size)
- void setMinimumSize(QSize size)
- void setModality(Qt::WindowModality modality)
- bool setMouseGrabEnabled(bool grab)
- void setOpacity(qreal level)
- void setParent(QWindow *parent)
- void setPosition(QPoint pt)
- void setPosition_2(int posX, int posY)
- void setScreen(QScreen *newScreen)
- void setSizeIncrement(QSize size)
- void setTransientParent(QWindow *parent)
- void setVisibility(QWindow::Visibility v)
- void setWindowState(Qt::WindowState state)
- QSize sizeIncrement(void)
- QString title(void)
- QWindow * transientParent(void)
- Qt::WindowType type(void)
- void unsetCursor(void)
- QWindow::Visibility visibility(void)
- int width(void)
- WId winId(void)
- Qt::WindowState windowState(void)
- int x(void)
- int y(void)
- void alert(int msec)
- bool close(void)
- void hide(void)
- void lower(void)

- void raise(void)
- void requestActivate(void)
- void setHeight(int arg)
- void setMaximumHeight(int h)
- void setMaximumWidth(int w)
- void setMinimumHeight(int h)
- void setMinimumWidth(int w)
- void setTitle(QString)
- void setVisible(bool visible)
- void setWidth(int arg)
- void setX(int arg)
- void setY(int arg)
- void show(void)
- void showFullScreen(void)
- void showMaximized(void)
- void showMinimized(void)
- void showNormal(void)
- QWindow * fromWinId(WId id)
- void setActiveChangedEvent(const char *)
- void setContentOrientationChangedEvent(const char *)
- void setFocusObjectChangedEvent(const char *)
- void setHeightChangedEvent(const char *)
- void setMaximumHeightChangedEvent(const char *)
- void setMaximumWidthChangedEvent(const char *)
- void setMinimumHeightChangedEvent(const char *)
- void setMinimumWidthChangedEvent(const char *)
- void setModalityChangedEvent(const char *)
- void setOpacityChangedEvent(const char *)
- void setScreenChangedEvent(const char *)
- void setVisibilityChangedEvent(const char *)
- void setVisibleChangedEvent(const char *)
- void setWidthChangedEvent(const char *)
- void setWindowStateChangedEvent(const char *)
- void setWindowTitleChangedEvent(const char *)
- void setXChangedEvent(const char *)
- void setYChangedEvent(const char *)

- const char *getactiveChangedEvent(void)
- const char *getcontentOrientationChangedEvent(void)
- const char *getfocusObjectChangedEvent(void)
- const char *getheightChangedEvent(void)
- const char *getmaximumHeightChangedEvent(void)
- const char *getmaximumWidthChangedEvent(void)
- const char *getminimumHeightChangedEvent(void)
- const char *getminimumWidthChangedEvent(void)
- const char *getmodalityChangedEvent(void)
- const char *getopacityChangedEvent(void)
- const char *getscreenChangedEvent(void)
- const char *getvisibilityChangedEvent(void)
- const char *getvisibleChangedEvent(void)
- const char *getwidthChangedEvent(void)
- const char *getWindowStateChangedEvent(void)
- const char *getWindowTitleChangedEvent(void)
- const char *getXChangedEvent(void)
- const char *getYChangedEvent(void)

99.178 QGuiApplication Class

C++ Reference : <http://doc.qt.io/qt-5/QGuiApplication.html>

Parent Class : QCoreApplication

Parameters : int,char **

- qreal devicePixelRatio(void)
- bool isSavingSession(void)
- bool isSessionRestored(void)
- QString sessionId(void)
- QString sessionKey(void)
- QWindowList allWindows(void)
- QString applicationDisplayName(void)
- Qt::ApplicationState applicationState(void)
- void changeOverrideCursor(QCursor cursor)
- QClipboard * clipboard(void)
- bool desktopSettingsAware(void)
- int exec(void)

- `QObject * focusObject(void)`
- `QWindow * focusWindow(void)`
- `QFont font(void)`
- `QInputMethod * inputMethod(void)`
- `bool isLeftToRight(void)`
- `bool isRightToLeft(void)`
- `Qt::KeyboardModifiers keyboardModifiers(void)`
- `Qt::LayoutDirection layoutDirection(void)`
- `QWindow * modalWindow(void)`
- `Qt::MouseButtons mouseButtons(void)`
- `QCursor * overrideCursor(void)`
- `QPalette palette(void)`
- `QString platformName(void)`
- `QPlatformNativeInterface * platformNativeInterface(void)`
- `QScreen * primaryScreen(void)`
- `Qt::KeyboardModifiers queryKeyboardModifiers(void)`
- `bool quitOnLastWindowClosed(void)`
- `void restoreOverrideCursor(void)`
- `QList<QScreen * > screens(void)`
- `void setApplicationDisplayName(QString name)`
- `void setDesktopSettingsAware(bool on)`
- `void setFont(QFont font)`
- `void setLayoutDirection(Qt::LayoutDirection direction)`
- `void setOverrideCursor(QCursor cursor)`
- `void setPalette(QPalette pal)`
- `void setQuitOnLastWindowClosed(bool quit)`
- `QStyleHints * styleHints(void)`
- `void sync(void)`
- `QWindow * topLevelAt(QPoint pos)`
- `QWindowList topLevelWindows(void)`
- `void setapplicationDisplayNameChangedEvent(const char *)`
- `void setapplicationStateChangedEvent(const char *)`
- `void setcommitDataRequestEvent(const char *)`
- `void setfocusObjectChangedEvent(const char *)`
- `void setfocusWindowChangedEvent(const char *)`
- `void setfontDatabaseChangedEvent(const char *)`

- void setlastWindowClosedEvent(const char *)
- void setlayoutDirectionChangedEvent(const char *)
- void setpaletteChangedEvent(const char *)
- void setprimaryScreenChangedEvent(const char *)
- void setsaveStateRequestEvent(const char *)
- void setscreenAddedEvent(const char *)
- void setscreenRemovedEvent(const char *)
- const char *getApplicationDisplayNameChangedEvent(void)
- const char *getApplicationStateChangedEvent(void)
- const char *getcommitDataRequestEvent(void)
- const char *getfocusObjectChangedEvent(void)
- const char *getfocusWindowChangedEvent(void)
- const char *getfontDatabaseChangedEvent(void)
- const char *getlastWindowClosedEvent(void)
- const char *getlayoutDirectionChangedEvent(void)
- const char *getpaletteChangedEvent(void)
- const char *getprimaryScreenChangedEvent(void)
- const char *getsaveStateRequestEvent(void)
- const char *getscreenAddedEvent(void)
- const char *getscreenRemovedEvent(void)

99.179 QCoreApplication Class

C++ Reference : <http://doc.qt.io/qt-5/QCoreApplication.html>

Parent Class : QObject

- void installNativeEventFilter(QAbstractNativeEventFilter *filterObj)
- void removeNativeEventFilter(QAbstractNativeEventFilter *filterObject)
- void quit(void)
- void addLibraryPath(QString path)
- QString applicationDirPath(void)
- QString applicationFilePath(void)
- QString applicationName(void)
- qint64 applicationPid(void)
- QString applicationVersion(void)
- QStringList arguments(void)
- bool closingDown(void)

- QAbstractEventDispatcher * eventDispatcher(void)
- int exec(void)
- void exit(int returnCode)
- bool installTranslator(QTranslator *translationFile)
- QCoreApplication * instance(void)
- bool isQuitLockEnabled(void)
- QStringList libraryPaths(void)
- QString organizationDomain(void)
- QString organizationName(void)
- void postEvent(QObject *receiver, QEvent *event, int priority)
- void processEvents(QEventLoop::ProcessEventsFlags flags)
- void processEvents_2(QEventLoop::ProcessEventsFlags flags, int maxtime)
- void removeLibraryPath(QString path)
- void removePostedEvents(QObject *receiver, int eventType)
- bool removeTranslator(QTranslator *translationFile)
- bool sendEvent(QObject *receiver, QEvent *event)
- void sendPostedEvents(QObject *receiver, int event_type)
- void setApplicationName(QString application)
- void setApplicationVersion(QString version)
- void setAttribute(Qt::ApplicationAttribute attribute, bool on)
- void setEventDispatcher(QAbstractEventDispatcher *eventDispatcher)
- void setLibraryPaths(QStringList paths)
- void setOrganizationDomain(QString orgDomain)
- void setOrganizationName(QString orgName)
- void setQuitLockEnabled(bool enabled)
- bool startingUp(void)
- bool testAttribute(Qt::ApplicationAttribute attribute)
- QString translate(char *context, char *sourceText, char *disambiguation, int n)

99.180 QTextBrowser Class

C++ Reference : <http://doc.qt.io/qt-5/QTextBrowser.html>

Parameters : QWidget *

Parent Class : QTextEdit

- int backwardHistoryCount(void)
- void clearHistory(void)

- `int forwardHistoryCount(void)`
- `QString historyTitle(int i)`
- `QUrl historyUrl(int i)`
- `bool isBackwardAvailable(void)`
- `bool isForwardAvailable(void)`
- `bool openExternalLinks(void)`
- `bool openLinks(void)`
- `QStringList searchPaths(void)`
- `void setOpenExternalLinks(bool open)`
- `void setOpenLinks(bool open)`
- `void setSearchPaths(QStringList paths)`
- `QUrl source(void)`
- `void setanchorClickedEvent(const char *)`
- `void setbackwardAvailableEvent(const char *)`
- `void setforwardAvailableEvent(const char *)`
- `void sethighlightedEvent(const char *)`
- `void sethistoryChangedEvent(const char *)`
- `void setsourceChangedEvent(const char *)`
- `const char *getanchorClickedEvent(void)`
- `const char *getbackwardAvailableEvent(void)`
- `const char *getforwardAvailableEvent(void)`
- `const char *gethighlightedEvent(void)`
- `const char *gethistoryChangedEvent(void)`
- `const char *getsourceChangedEvent(void)`

99.181 QRegion Class

C++ Reference : <http://doc.qt.io/qt-5/QRegion.html>

Parameters : void

- `QRect boundingRect(void)`
- `bool contains(QPoint p)`
- `bool contains_2(QRect r)`
- `QRegion intersected(QRegion r)`
- `QRegion intersected_2(QRect rect)`
- `bool intersects(QRegion region)`
- `bool intersects_2(QRect rect)`

- `bool isEmpty(void)`
- `bool isNull(void)`
- `int rectCount(void)`
- `QVector<QRect> rects(void)`
- `void setRects(QRect *rects, int number)`
- `QRegion subtracted(QRegion r)`
- `void swap(QRegion other)`
- `void translate(int dx, int dy)`
- `void translate_2(QPoint point)`
- `QRegion translated(int dx, int dy)`
- `QRegion translated_2(QPoint p)`
- `QRegion united(QRegion r)`
- `QRegion united_2(QRect rect)`
- `QRegion xored(QRegion r)`

FREQUENTLY ASKED QUESTIONS (FAQ)

100.1 Why do we need Yet Another Programming Language (YAPL)?

The language comes with better support for Natural Language Programming and Declarative Programming. The innovation comes in supporting these paradigms with new practical techniques on the top of Object-Oriented Programming and Functional Programming. Ring provide the programmers with the tools required to build a Natural Language like Supernova or Declarative Language like REBOL and QML without the need to know anything about (Compilers and Parsing). You get the language constructs ready for use to create domain-specific languages in a fraction of time.

Check the Supernova programming language, In this language you can type (I want window and the window title is hello world.) and it will create a GUI window with “Hello, World!” as the window title. When I created Supernova language in 2010. I discovered that using the Natural Code can be (Like English without limits and we can get the human language power in programming) but to implement that You need a new language that are

1. General Purpose
2. Practical
3. Can create Natural Languages very quickly.

So we can get a System that can increase ease of use and productivity to the maximum level.

Ring is the best language to do that. So I created Ring to achieve this goal.

Supernova was just a test for the idea. To get a near view about what are the Advantages and Disadvantages. After testing the new ideas you provide something practical. So we have Ring after Supernova. A story that maybe similar to having Python after ABC. Python avoid ABC problems but bring ABC advantages. Also Ring learn from Ruby and ROR story. The language power could appears in Frameworks better than direct usage as a general purpose language. Also Ring comes with a clear goal/motivation (Creating a new version of the PWCT Software) something learned from designing the C language to create the Unix Operating System. You have a goal that direct you in each design decision.

You will understand the value of our decisions once you start trying to solve the problem that we will use Ring to solve. Could you enable any one in the world without knowledge about computer programming concepts to create very powerful software? In science the answer is (Visual Programming) and (Natural Programming). In practical we still away from switching to these paradigms without introducing other problems. Ring is designed to solve this problem. It's designed to provide Natural Programming in a practical way. And to create a powerful Visual Programming tool. Ring is designed to be a new world of programming after 10 years of research in Visual Programming and Natural Languages.

The Ring Programming Language (Compiler+VM) is developed 100% using Visual Programming without writing a single line of code. I used my tool (Programming Without Coding Technology) to design everything and get the C code generated for me. Advantages ?

1. More Faster
2. No Syntax Errors
3. Easy to understand and manage the code because the Abstraction level is more higher

4. No Critical Disadvantages because you can control everything as writing your code.

From my experience in using Visual Programming for 10 years and Natural Programming for 5 years I designed Ring to move the knowledge to mainstream programmers by providing a practical language that support these ideas.

I agree that each programmer/developer has the freedom to form his opinions about any software including programming languages. Ring is not an exception but you may miss the idea behind the language. It's innovative and may help you to think different about how to solve your problems. Maybe this is not clear to many programmers because It's practical language and includes many features known to programmers and when a programmer look at a language he/she may think that nothing new because it's familiar. I created Ring to solve problems in a different way. Where I will start programming just by describing the software using new natural interfaces that I will implement later when I move from the design stage to the implementation stage. (I don't determine the time to switch between stages, You are free to use Agile methods). Since Ring is a new language you have 3 options.

1. Just don't care.
2. Think in the future of the language and help us if you understand the idea and want to contribute.
3. Wait and come back again in the future to use it.

Summary:

- Ring is designed based on a need to develop a new version of the PWCT software.

Once we finish PWCT 2.0 we will have good and large software developed using Ring.

- We will push Declarative and Natural paradigms many steps forward. Also in next versions

we have a plan to present a new paradigm for Network Programming and Concurrency. We tested this new paradigm through simple prototypes during the last years and we will integrate it with Ring in Future releases.

100.2 Why Ring is weakly typed?

Because it's more Natural, and Faster and this is important for the language goals. What comes first is what you want. When you type "Print : " + 5 , The String comes first then 5 will be converted to a String. while when you type 5 + "10" The number comes first so "10" will be converted to 10. This help a lot to quickly convert between numbers and strings using the same operator. If you want to prevent conversion (Write code that prevent conversion) In these cases you will notice that what you are writing is less code (And can be removed).

Weakly Typed = Automatic Conversion and *Automatic* is *Good Thing* and is better than *Manual* if you know how to use it correctly.

100.3 What are the advantages to using Ring over Lisp or Smalltalk?

Smalltalk and Lisp are GREAT languages. I like many of the concepts behind them but I'm sure that selecting the right programming language is based on the problem and comes after the problem definition. I have a problem that I want to solve and these GREAT languages are not ideal for this problem so I designed Ring.

When you design a new language, You can learn from the past but you must look forward and live in the Future. What you know about Natural Programming maybe based on the *Old Knowledge* about the power of these paradigms in the practical world and I agree with you but I see another techniques that can be applied to get this to work in practical. What you miss about *Natural Language* is that they are *Context Sensitive* and this means we can use it and think different about how we can express our ideas.

example : I want window contains 3 buttons.

In one sentence I created 4 objects (The window and the three buttons) and added the buttons to the window. The idea of Natural Programming is to get many things done like that.

100.4 Why Ring is largely focussed on UI creation?

Yes UI creation is one of the important things in the language features because it's designed to create a visual programming tool, But the language is a multi-paradigm language where we can select the programming paradigm based on the problem.

100.5 Is Ring some sort of improvement over PHP?

Ring is not designed to replace PHP, Lua and/or Smalltalk. And Ring support for Declarative Programming and Natural Language Programming is very innovative and much better than staying with Procedural, Object-Oriented and Functional Languages. Ring see the future in programming without code (Using Natural Languages) and is designed to support that.

100.6 What are the advantages to using Ring over native C or C++?

Ring provides a better way to mix between different programming paradigms in solving problems.

The different programming paradigms play well together in the same language.

1. It's easy to switch from one programming paradigm to another one because the language constructs use similar syntax for similar concepts.
2. The paradigms are provided to interact and used together in different layers in the software.

for example you can create a game engine using object-oriented programming but write the game code using declarative programming or natural programming and behind the scene your declarative or natural code will use the object-oriented classes.

3. Ring is very productive and natural programming language than C/C++ languages.
4. Ring is a dynamic language. We can generate and execute code during the runtime. We have dynamic typing and weakly typed language for flexibility.
5. The Garbage collector is generational (escape analysis) and also use reference counting. it's very fast and still provide control to the programmer who can delete memory at any time.
6. Ring Compiler and Virtual Machine is just 15,000 lines of ANSI C code that can be compiled and used in any platform.
7. You can use C/C++ libraries and Ring comes with code generator to create wrappers from C functions or C++ classes. so when you need more performance or when you need to use more libraries you can easily do that.

100.7 What is the difference between Ring and Python? And is Ring Open Source?

Yes the language is Free Open Source (MIT license)

In general I like Python and Ruby but I was looking for a language more suitable for creating the next version of the Programming Without Coding Technology (PWCT) software so I started the Ring design.

Some simple changes that matters for my goal are

1. Not case sensitive

2. The list index start from 1
3. You can call functions before definition
4. Don't use Python syntax like (indentation, using self, :, pass & _)
5. Weakly typed (convert automatically between types based on context)
6. The programs follow simple and constant structure (Statements then functions then packages and classes)
7. Using the '=' operator for assignment and for testing values

Critical changes are

1. Small Language : The Ring compiler + Virtual Machine = 15K lines of C code , the other 500K lines are related to libraries and are optional when we go for using the language in C/C++ programs.
2. The Garbage collector : Uses Escape Analysis/Reference counting and give the programmer the ability to determine when to delete memory using the assignment operator
3. Compact Syntax : Ring is not line sensitive, you don't need to write ; or press ENTER to separate between statements
4. Using { } to access the object then using the object attributes and methods directly
5. Natural Programming : It's very easy to create natural interfaces using Ring based on OOP
6. Declarative Programming using Nested Structure

The Ring programming language is designed based on my experience from using many other languages like C, C++, C#, Lua, PHP, Python, Ruby, Harbour, Basic and Supernova And the language comes with innovative features added to achieve the language goal

- Applications programming language.
- Productivity and developing high quality solutions that can scale.
- Small and fast language that can be embedded in C/C++ projects.
- Simple language that can be used in education and introducing Compiler/VM concepts.
- General-Purpose language that can be used for creating domain-specific libraries, frameworks and tools.
- Practical language designed for creating the next version of the Programming Without Coding Technology software.

100.8 What are the advantages to using Ring over Perl, PHP, Python or Ruby?

1. Ring is New and Innovative. The language will let you think different about programming.
2. Ring is Smaller. (Lessons learned from the Lua language)
3. Ring is Simple. (Lessons learned from the BASIC and Clipper/Harbour languages)
4. Ring is more Natural. (Lessons learned from the Supernova language)
5. Ring is more Declarative. (Lessons learned from REBOL and QML languages)
6. Ring Implementation is Transparent, Visual and comes with Rich Features.

100.9 What are the advantages to using Ring over Tcl or Lua?

1. Clean Code (More Natural)
2. More Features (A lot of useful programming paradigms)

100.10 What are the advantages to using Ring over C# or Java?

1. Compact Code (Clean and Natural), More Productivity and Flexibility.
2. Better support for Declarative Programming and Natural Programming

100.11 The documentation says functional programming is supported, but then this happens?

The question was about this code

```
f = func {
  a = 42
  return func { return a }
}

innerF = call f()
call innerF()
```

Output:

```
Using uninitialized variable : a In function _ring_anonymous_func_16601()
```

The Answer:

- It's Anonymous Functions, i.e. Not Closures.
- Many developers asked about supporting Closures and during language development we may add new features that doesn't go against the language goals or sprite.
- You can use classes and objects when you want to merge between the state and functions to provide a clear solution.
- You can use Lists and put the anonymous function inside the List then return the list that contains the state and the function. Pass the list to the function when you use it.
- You can use eval() and substr() to add the variable value directly to the anonymous function before return.
- We protect you from other scopes when you define the function. In Ring we provided the Three Scopes Rule where at each point you have only at maximum three scopes (Global, Object Scope and Local Scope).
- We don't get everything from everywhere to be like others! We don't need to do that. If we will think like that then we will create a very complex language or we will save our time and use other languages.
- When you think about learning or studying a new language concentrate about (What is new?) and (What is better in this language?) to know when to use it. Don't compare a new language just released little months ago with languages started many years ago and expect to find everything that you used to have.
- Each programming language miss features in other languages. The idea is not the Features. it's the sprite and ability behind all of the features together.

100.12 Why the ability to define your own languages Instead of just handing over the syntax so you can parse it using whatever code you like?

It's innovation - You create natural statements without the need to learn about parsing. You just use Classes which is intelligent decision (where later we can mix between classes to support more statements based on the context - We can change and translate the defined statements and many more!). Also the statements are added in Ring World where you can use any Ring statement.

100.13 Why you can specify the number of loops you want to break out of?

The language supports programming in the small and programming in the large. The selection of what features to use is based on what are you going to do. Any programmer can write poorly code in any language if he/she wants to do that. The idea is what must be done in the language design to prevent errors without causing other problems like killing flexibility.

Read some source code in the Linux Kernel and Ruby Implementation for example, You will find good usage for GOTO as a practical example that General Rules are not for All Use Cases and great programmers know when to break the rules. I'm not saying go and use GOTO or saying Ring add things like that. But the ability to break more than one loop and/or the ability to break the loop from sub functions is practical for small programs.

Anyway these are some of the small new things added by the language (Not the big idea).

100.14 Why Ring uses 'See', 'Give', 'But' and 'Ok' Keywords?

See and Give are selected not to be "opposite actions" but to reflect what I want to do as a programmer.

When I want to see something on the screen I use 'See'.

When I want to give some input to the program I use 'Give'.

My selection of "but" and "ok" is based on selecting keywords that can be written quickly.

Also using "but" is easy to remember than elseif/elif/elsif where each language select a different keyword.

In Ring 1.1 and later versions All of this is just an option.

You can use 'Put' and 'Get' instead of 'See' and 'Give'

You can use 'elseif' and 'end' instead of 'But' and 'Ok'

It's your choice. In Ring we have syntax flexibility where we provide more than one style.

Also you can change the language keywords and operators.

Also you can define new natural languages too.

100.15 What is the philosophy behind data types in Ring?

The Ring programming language is designed to be SMALL. The language provides the basic constructs that you need to do anything! One of the goals is to keep the basic constructs simple and small as possible.

Using Lists in Ring you can

- Create Arrays (one data type)
- Create Lists (Mix of data types)
- Create Tree (Nested arrays)
- Use String Index (Looks like Dictionary/Hash Table)

The same principle is applied to Numbers

- You can use the number for int value
- You can use the number for double value
- You can use the number for Boolean value (True/False)

The same principle is applied for Strings

- You can use the string for storing one character
- You can use the string for storing text (one or many lines)
- You can use the string for storing binary data
- You can use the string for storing date
- You can use the string for storing time
- You can use the string for storing NULL values (empty strings)

And we have Object Oriented Support + Operator Overloading where the programmer can define new data types and use them as default types defined by the language

So We have

- A small and simple language that someone can pick in little days
- A fast language that provide primitive types (String ? Number ? List ? Object)
- A flexible language that can be extended using OOP to add new types according to the application domain

100.16 What about the Boolean values in Ring?

You can use true for 1 and false for 0

when you test the result of Boolean expressions in your code.

Just when you print the value using the see command you will see 1 for (true) and 0 for (false)

Why ?

Because Ring contains only 4 types of variables

1. Number
2. String
3. List
4. Object

The first type (Number) is used to represent int, double and Boolean values.

The second type (String) is used to represent char, array of characters, date and time.

The third type (List) is used to represent Arrays of one type, Arrays of more than one type, Hash (Dictionary), Tree, etc.

The object can be an object created from a Ring class (Any Class) or just a C Pointer that we get from calling a C/C++ function/method.

Why ?

The Ring is designed to give the programmer/developer the most simple constructs that can be used to do everything. The programmer/developer can customize the language by creating new classes (and use operator overloading) to get more types that he care about according to the problem domain.

Why ?

Because simple is better, and easy to learn and remember! And this provide flexibility to convert between high level types that can be represented using the same basic type

100.17 What is the goal of including the “Main” function in Ring?

The main function is very important, you need it when you want to write statements that uses local variables instead of the Global scope.

Example:

```
x = 10
myfunc()
See "X value = " + X  # here I expect that x will be (10)
                      # but I will get another value (6) because myfunc() uses x !

Func myfunc
  for x = 1 to 5
    See x + nl
  next
```

Output:

```
1
2
3
4
5
X value = 6
```

Now using the Main function

```
Func Main
  x = 10
  myfunc()
  See "X value = " + X

Func myfunc
  for x = 1 to 5
    See x + nl
  next
```

Output

```
1
2
3
```



```
4
5
X value = 10
```

100.18 Why the list index start from 1 in Ring?

It's about how we count in the real world, when we have three apples in our hand

we say 1 2 3

We don't start from 0

The question must be why the other languages start from 0 ?

The answer is, because this is related to the machine and how we deal with values and memory address.

Example

we have array called myarray[5]

In memory : myarray will have an address

The first item will be stored in that address

The second item will come after that address and so on

Now when we need to point to the first item we need the address of myarray

So we type myarray[0] because myarray + 0 result will still point to the first item

for the second item myarray[1] because myarray + 1 result will point to the second item and so on

In Low Level languages or languages near to the machine it's good to be like this

But for high level language designed for applications it's better to be natural

Example

```
mylist = [1,2,3,4,5]
for x = 1 to len(mylist)
    see x + nl
next
```

In the previous example we start from 1 to the length of the array if the index starts from 0 we will write

```
for x = 0 to len(mylist)-1
```

or remember the for loop in other languages

```
for (x=0 ; x<nMax ; x++ )
```

You will use the < operator !

100.19 Why Ring is not case-sensitive?

1. To be more human-friendly
2. Like Ada, SQL, Pascal, Delphi, Visual Basic, Visual FoxPro, etc.
3. To help in supporting Natural Language Programming.

4. To be able to select your favorite style when writing the language keywords

```
see "lower case!"
```

```
SEE "UPPER case!"
```

```
See "First Letter is UPPER case!"
```

5. To avoid getting error message when writing quick tests then type “variable” instead of “Variable”.
6. To avoid getting error message when you type “Dosomething()” instead of “doSomething()”
7. In Ring, No conflict between Variables, Method Names & Classes Names

We can write person as variable name and Person as class name.

```
person = new Person
class Person
    name address phone
```

100.20 Why the Assignment operator uses Deep Copy?

“Because it’s a poor tradeoff to add complexity for dubious performance gains, a good approach to deep vs. shallow copies is to prefer deep copies until proven otherwise.”

, Steve McConnell, Code Complete

1. It’s more natural, When you use the assignment operator, You expect a deep copy.
2. If you don’t need a deep copy, Just don’t use it!
3. The Ring language is designed to reduce references usage as much as possible.
4. The Ring language is designed to make using references simple and possible in special cases where this make sense.
5. **We have references when this is natural, like passing lists and objects to functions,** creating objects (Like GUI Objects) from a C/C++ library, returning an object stored inside a list.
6. **It is a feature, We can use it to create pure functions. The Value() function in the** stdlib uses this feature to pass lists & objects by value when we need this.
7. When we need references, It’s recommended to create a class that manage sharing lists and objects.
8. It’s more safe at the application level to avoid many logical errors.
9. **In Ring, we start without thinking about the little details and concentrate on the application, You** don’t have to write the type (Dynamic Typing), You don’t have to write explicit conversions between numbers and strings (Weakly Typed) and you don’t have to select between using values or references, You don’t have to write the scope (Lexical Scoping).
10. **In Ring, we have smart garbage collector (Simple & Fast), We can delete the memory directly** at any time using the Assignment operator too. Reducing references usage or using them through managers helps a lot to achieve this goal. by doing this we have full control.
11. **If you want to create references and avoid creating a manager, You can use Object2Pointer() and Pointer2Object() function** But It’s not the Ring way “Sprite” to do things.

100.21 Is there constructor methods in Ring?

When you create new object for example

```
new point
```

1 - Ring will allocate dynamic memory space to be used for the new object attributes that Ring doesn't know anything about them.

2 - Ring will change the current local scope and the current object scope to use the object state created in step (1)

3 - Ring will move the execution to the class Region (After the class name and before any methods)

4 - Any Instructions/Code in the class region will be executed as any Ring code

5 - Control is moved from the class region to the location of (new point) once we reach the end of the class region or we uses a Return command.

So All attributes that added to the object are dynamic attributes, this mean that you can control what attributes will be added through the runtime.

Example:

```
$3D = False
see new point
$3D = True
see new point

class point
    x y
    if not $3D return ok
    z
```

Output:

```
x: NULL
y: NULL
x: NULL
y: NULL
z: NULL
```

You have an option to call init() method directly when you create a new object

This method can do anything with the object attributes as it will be called after creating the object and executing the class region code.

```
p1 = new point3d(100,200,300)
see p1

class point3d
    x y z
    func init p1,p2,p3
        x=p1 y=p2 z=p3
```

100.22 What happens when we create a new object?

1- When you create an object, the class region code will be executed and you will have the object attributes based on the code in that region

2- Ring don't care about the object methods until you start calling a method

3- When you call a method, Ring will check the object class and the class parent (if you are using inheritance) and will collect the methods for you to be used now or later from any object that belong to the same class.

4- Since methods are dynamic and each object get the method from the class, you can after creating objects, add new methods and use it with the object or any object created or will be created from the same class.

Example:

```
o1 = new point {x=10 y=20 z=30}
o2 = new point {x=100 y=200 z =300}

addmethod(o1,"print", func { see x + nl + y + nl + z + nl } )

o1.print()
o2.print()

class point x y z
```

Output:

```
10
20
30
100
200
300
```

100.23 Can we use the attributes by accessing the Getter and Setter methods?

Yes we can, The setter/getter methods are called automatically when you start using the attributes from outside the class Also you can call the methods instead of using the attributes. It's your choice.

Example:

```
o1 = new Developer
o1.name = "Mahmoud" see o1.name + nl
o1 { name = "Gal" see name }
o1 { name = "Bert" see name }

o1.setname("Marino")
see o1.getname()

Class Developer

    name language = "Ring Programming Language"

    func setname value
        see "Message from SetName() Function!" + nl
        name = value + " - " + language

    func getname
        see "Message from GetName() Function!" + nl + nl
        return "Mr. " + name + nl
```

Output

```

Message from SetName() Function!
Message from GetName() Function!

Mr. Mahmoud - Ring Programming Language

Message from SetName() Function!
Message from GetName() Function!

Mr. Gal - Ring Programming Language
Message from SetName() Function!
Message from GetName() Function!

Mr. Bert - Ring Programming Language
Message from SetName() Function!
Message from GetName() Function!

Mr. Marino - Ring Programming Language

```

100.24 Why should a search of global names be made while defining the class attributes?

The question is why we don't avoid conflicts with global variable names when we define the class attributes ?

At first remember that using the optional \$ mark in the global variables names solve the problem. Also using the Main function and avoiding global variables may help.

The Answer:

Ring is a dynamic language

We can in the run-time determine the class attributes (Add/Remove)

We can execute (any code) while defining the class attributes

Example (1)

```

oPerson = new Person
Class Person
    See "Welcome to the Ring language"

```

Example (2)

Customize attributes based on global variable value

```

$debug = true
oPerson = new Person
see oPerson
Class Person
    if $debug  date=date()  time=time()  ok

```

In the previous example when we have the \$debug flag set to true, we will add the Date and Time attributes to the object state.

Example (3)

Store the object index based on global variable

```

$ObjectsCount = 0
oPerson = new Person
see oPerson
oPerson2 = new Person
see oPerson2
Class Person
    $ObjectsCount++
    nIndex = $ObjectsCount

```

Output:

```

nindex: 1.000000
nindex: 2.000000

```

Common Example:

- Connect to the database then get table columns (Using global Variable/Object).
- Create class attributes based on the column names.
- Later when you modify the database - you may don't need to modify your code.

It's flexibility but remember that power comes with great responsibility.

100.25 Why Ring doesn't avoid the conflict between Global Variables and Class Attributes Names?

In this use case we have

- 1 - Global Variable defined without a special mark like \$
- 2 - Class contains Attributes defined using a special syntax (where we type the attribute name directly after the class)
- 3 - The Attributes are defined in the class region that allows writing code and using global variables

If I will accepted your proposal about changing how Ring find variables in the class region I must break one of the previous three features which will lead to more problems that are more important than this problem.

I don't like changing the feature number (1) because I would like to keep Ring code more clean and let the programmer decide when to use \$ or not.

I don't like changing the feature number (2) because I like this feature and I don't like forcing the programmer to type self.attribute

I don't like changing the feature number (3) because it's very important in many applications to access global variables in the class region.

So what was my decision ?

I decided to leave this case for the programmer who will decide what to do to avoid this special case

- 1 - The programmer can avoid using global variables (Better) and can use the Main function (Optional)
- 2 - The programmer can use \$ before the variable name or any mark like **global_** or **g_**
- 3 - The programmer can use self.attribute after the class name to define the attributes

In general, for small programs you can use global variables and functions. For large programs, use classes and objects and small number of global variables or avoid them at all.

100.26 Where can I write a program and execute it?

Run the Ring Notepad where you can write/execute programs.

If you want to run programs using the command line

Add Ring/bin folder to the path then

100.27 How to get the file size using ftell() and fseek() functions?

The next function can be used to get the file size without reading the file!

```
func getFileSize fp
  C_FILESTART = 0
  C_FILEEND = 2
  fseek(fp,0,C_FILEEND)
  nFileSize = ftell(fp)
  fseek(fp,0,C_FILESTART)
  return nFileSize
```

Note: The previous function take the fp (file pointer) as parameter, We can get the fp from opening the file using fopen() function.

```
fp = fopen("filename","r")

see "File Size : " + getFileSize(fp) + nl
```

Another solution (Read the file)

```
see len(read("filename"))
```

100.28 How to get the current source file path?

We can use the next function to get the current source file path then we can add the path variable to the file name

```
cPath = CurrentPath()
func currentpath
  cFileName = filename()
  for x = len(cFileName) to 1 step -1
    if cFileName[x] = "/"
      return left(cFileName,x-1)
  ok
next
return cFileName
```

100.29 What about predefined parameters or optional parameters in functions?

if you want to use predefined parameters or optional parameters Just accept a list that works like hash/dictionary

Example

```

sum([ :a = 1, :b = 2])
sum([ :a = 1 ])
sum([ :b = 2 ])
func sum pList
    if plist[:a] = NULL pList[:a] = 4 ok
    if plist[:b] = NULL pList[:b] = 5 ok
    see pList[:a] + pList[:b] + nl

```

Output

```

3
6
6

```

100.30 How to print keys or values only in List/Dictionary?

If you want to print keys only or values only just select the index of the item (one or two).

Example

```

C_COUNTRY = 1
C_CITY = 2
mylist = [
    :KSA = "Riyadh" ,
    :Egypt = "Cairo"
]

for x in mylist
    see x[C_COUNTRY] + nl
next

for x in mylist
    see x[C_CITY] + nl
next

```

Output

```

ksa
egypt
Riyadh
Cairo

```

100.31 Why I get a strange result when printing nl with lists?

In the next code

```

list = 1:5          # list = [1,2,3,4,5]
see list + nl

```

New Line will be added to the list then the list will be printed, the default print of the lists will print a newline at the end, You added new newline and You have now 2 newlines to be printed.

See <Expr>

The see command just print the final result of the expression, the expression will be evaluated as it


```
nl = char(13) + char(10) # just a variable that you can change to anything !
```

The + is an operator

```
string + string ---> new string
string + number ---> new string
number + number ---> new number
number + string ---> new number
```

list + item —> nothing new will be created but the item will be added to the same list

Exception

number + nl ?> New String

This exception is added to easily print numbers then new line.

No need for this with printing lists because after printing the last item we already get a new line.

100.32 Could you explain the output of the StrCmp() function?

At first remember that you can check strings using ‘=’ operator directly.

```
see strcmp("hello","hello") + nl +
strcmp("abc","bcd") + nl +
strcmp("bcd","abc") + nl
```

if the two strings are the same then it returns 0

abc and bcd aren’t the same. in the second line it returns -1 and in the third line it returns 1

In the second line we compare between “abc” and “bcd”

Not equal because the first letter in “abc” = “a” and the first letter in “bcd” = “b”

So we have “a” != “b” and “a” < “b”

So we get output = -1

In the third line we have “bcd” and “abc”

the first letter in “bcd” is “b” and the first letter in “abc” is “a”

So we have “b” != “a” and “b” > “a”

So we get output = 1

Note: ASCII(“a”) = 97 and ASCII(“b”) = 98 So “a” < “b” because 97 < 98

100.33 How to use many source code files in the project?

Example:

I have the next folder

```
C:\LRing
```

Contains the next files

```
C:\LRing\t1.ring
C:\LRing\mylib.ring
C:\LRing\libs\mylib2.ring
```

The file t1.ring contains the next code

```
load "mylib.ring"
load "libs\mylib2.ring"
myfunc()
test()
```

The file mylib.ring contains the next code

```
func myfunc
    see "message from myfunc"+nl
```

The file libsmplib2.ring contains the next code

```
func test
    see "message from test" + nl
```

from the folder C:LRing

If Ring is not added to the path you can add it or use the next command

```
set path=%path%;c:\ring\bin;
```

Where c:ring is the Ring folder

Now run

```
Ring t1.ring
```

Output

```
message from myfunc
message from test
```

100.34 Why this example use the GetChar() twice?

The GetChar() function accept one character from the keyboard buffer

In this example

```
While True
    See "
        Main Menu
        (1) Say Hello
        (2) Exit
    "
    Option = GetChar()
    GetChar() GetChar() # End of line
    # the previous two lines can be replaced with the next line
    # Give Option

    if Option = 1
        see "Enter your name : " give cName
        see "Hello " + cName
    else
```

```

    ok      bye
End

```

We uses GetChar() Three times

The first time we get the user option

```
Option = GetChar()
```

But in the second and the third times (We accept the new line characters from the buffer)

```
GetChar() GetChar() # End of line
```

Example : when the user select the option number 1 then press ENTER

We have Three Characters

- The first character is : Number 1
- The second character is : CHAR(13)
- The third character is : CHAR(10)

Because Windows uses CHAR(13) and CHAR(10) for each new line (i.e. CR+LF)

100.35 How to use NULL and ISNULL() function?

when we try to use uninitialized variable in the Ring programming language, we get a clear runtime error message

Example

```
See x
```

Output

```
Line 1 Error (R24) : Using uninitialized variable : x
in file tests\seeuninit.ring
```

The same happens when you try to access uninitialized attributes

Example

```
o1 = new point
see o1
see o1.x
class point x y z
```

Output

```
x: NULL
y: NULL
z: NULL

Line 3 Error (R24) : Using uninitialized variable : x
in file tests\seeuninit2.ring
```

if you want to check for the error, just use Try/Catch/End

```

Try
    see x
Catch
    See "Sorry, We can't use x!" + nl
Done

```

Output

```
Sorry, We can't use x!
```

Now we will talk about NULL and ISNULL()

Since we get error message when we deal with uninitialized variables

We can check these errors using Try/Catch/Done, So we uses NULL and ISNULL() for dealing with Strings.

NULL is a variable contains an empty string

ISNULL() is a function that returns true (1) if the input is an empty string or just a string contains “NULL”

This because we need to test these values (empty strings) and strings contains “NULL” that sometimes come from external resource like DBMS.

Example

```

See IsNull(5) + nl +      # print 0
IsNull("hello") + nl +    # print 0
IsNull([1,3,5]) + nl +    # print 0
IsNull("") + nl +         # print 1
IsNull("NULL") + nl +     # print 1

```

100.36 How to print lists that contains objects?

In this example we will see how we can print a list contains objects.

```

aList = [[1,2,3] , new point(1,2,3), new point(1,2,3)]
see "print the list" + nl
see alist
see "print the item (object)" + nl
see alist[2]
class point x y z
    func init p1,p2,p3 x=p1 y=p2 z=p3

```

Output

```

print the list
1
2
3
x: 1.000000
y: 2.000000
z: 3.000000
x: 1.000000
y: 2.000000
z: 3.000000
print the item (object)
x: 1.000000
y: 2.000000
z: 3.000000

```

100.37 How to insert an item to the first position in the list?

To insert an item we can use the `insert(aList,nIndex,Value)` function.

```
aList = 1:5
insert(aList,0,0)
See aList # print numbers from 0 to 5
```

100.38 How to print new lines and other characters?

To print new line we can use the `nl` variable.

```
See "Hello" + nl
```

or we can use multi-line literal as in the next example

```
See "Hello
"
```

if we want to print other characters we can use the `char(nASCII)` function

```
See char(109) + nl + # print m
char(77) # print M
```

100.39 Why we don't use () after the qApp class name?

When we use RingQt to create GUI application, we uses `()` after the class name when we create new objects for example.

```
new QWidget() { setTitle("Hello World") resize(400,400) show() }
```

but before doing that we create an object from the `qApp` class and we don't use `()` after that

```
Load "guilib.ring"
app = new qApp
{
    win=new QWidget()
    {
        setTitle(:test)
        show()
    }
    exec()
}
```

Using `()` after the class name means calling the `init()` method in the class and passing parameters to this method.

If we used `()` while no `init()` method in the class we get the expected error message.

The class `qApp` don't have this method while the other classes have it because they need it to create an object using a function that return a pointer to that object and this pointer will be stored in an attribute called `pObject`, for more information see `ring_qt.ring` file which contains the classes.

100.40 Why the window title bar is going outside the screen?

When we write the next code

```
Load "guilib.ring"
app = new QApplication
{
    win=new QWidget()
    {
        setWindowTitle(:test)
        setGeometry(0,0,200,200)
        show()
    }
    exec()
}
```

I would expect that the window will run at the point (0,0) with (200,200) size but the actual result is that the window title bar is going outside the screen.

This is related to the behavior of Qt framework.

The next code will avoid the problem

```
load "guilib.ring"
new QApplication {
    new QWidget() {
        move(0,0)
        resize(200,200)
        show()
    }
    exec()
}
```

100.41 How to create an array of buttons in GUI applications?

Check the next example:

```
Load "guilib.ring"

App1 = new QApplication {

    win1 = new QWidget() {
        move(0,0)
        resize(500,500)
        new QPushButton(win1)
        {
            setText("OK")
            setClickedEvent("click()")
        }
        btn1 = new QPushButton(win1)
        {
            setGeometry(100,100,100,30)
            setText("Button1")
        }

        btn2 = new QPushButton(win1)
        {
```

```

        setgeometry(200,100,100,30)
        setttext("Button2")
    }

    button = [btn1, btn2]
    show()
}

exec()

}

func click

    button[1] { setttext ("Button3") }
    button[2] { setttext ("Button4") }

```

100.42 How to Close a window then displaying another one?

This example demonstrates how to close a window and show another one

```

Load "guilib.ring"

app=new QApplication
{
    frmBefore=new QWidget()
    {
        setWindowTitle("before!")
        resize(300,320)
        move(200,200)

        button=new QPushButton(frmBefore)
        {
            setText("Close")
            setClickEvent("frmBefore.close() frmMain.show()")
        }

        show()
    }

    frmMain=new QWidget()
    {
        setWindowTitle("After!")
        resize(300,320)
        move(200,200)
    }

    exec()
}

```

100.43 How to create a Modal Window?

This example demonstrates how to create a modal window

```

load "guilib.ring"
app=new QApplication
{
    frmStart=new QWidget()
    {
        setWindowTitle("The First Window")
        resize(300,320)
        move(200,200)

        button=new QPushButton(frmStart)
        {
            setText("Show Modal Window")
            resize(200,30)
            setClickEvent("frmModal.show()")
        }

        new QPushButton(frmStart)
        {
            setText("Close Window")
            move(0,50)
            resize(200,30)
            setClickEvent("frmStart.Close()")
        }

        show()
    }

    frmModal =new QWidget()
    {
        setWindowTitle("Modal Window")
        resize(300,320)
        move(200,200)
        setparent(frmStart)
        setwindowmodality(true)
        setwindowflags(Qt_Dialog)
    }

    exec()
}

```

Related Documents

- <http://doc.qt.io/qt-5/qtwidgets-widgets-windowflags-example.html>
- <http://doc.qt.io/qt-5/qt.html#WindowType-enum>
- <http://doc.qt.io/qt-5/qwindow.html#setParent>
- <http://doc.qt.io/qt-5/qt.html#WindowModality-enum>

100.44 How can I disable maximize button and resize window?

Use the method `setWindowFlags()`

```

Load "guilib.ring"
app1 = new QApplication {
    win1 = new QWidget() {

```



```

        setwindowtitle("First")
        setgeometry(100,100,500,500)

        new qpushbutton(win1) {
            setgeometry(100,100,100,30)
            settext("close")
            setclickevent("app1.quit()")
        }

        new qpushbutton(win1) {
            setgeometry(250,100,100,30)
            settext("Second")
            setclickevent("second()")
        }

        showmaximized()
    }
    exec()
}

func second
    win2 = new qwidget() {
        setwindowtitle("Second")
        setgeometry(100,100,500,500)
        setwindowflags(Qt_dialog)
        show()
    }

```

100.45 How to use SQLite using ODBC?

In Ring 1.1 and later versions we have native support for SQLite, so you don't need to use it through ODBC.

Also we can access SQLite through RingQt.

The answer to your question

```

pODBC = odbc_init()
odbc_connect(pODBC,"DRIVER=SQLite3 ODBC Driver;Database=mydb.db;LongNames=0;"+
    "Timeout=1000;NoTXN=0;SyncPragma=NORMAL;StepAPI=0;")
odbc_execute(pODBC,"create table 'tel' ('ID','NAME','PHONE');")
odbc_execute(pODBC,"insert into 'tel' values ('1','Mahmoud','123456');")
odbc_execute(pODBC,"insert into 'tel' values ('2','Ahmed','123456');")
odbc_execute(pODBC,"insert into 'tel' values ('3','Ibrahim','123456');")
odbc_execute(pODBC,"select * from tel") + nl
nMax = odbc_colcount(pODBC)
See "Columns Count : " + nMax + nl
while odbc_fetch(pODBC)
    See nl
    for x = 1 to nMax
        see odbc_getdata(pODBC,x)
        if x != nMax see " - " ok
    next
end
odbc_disconnect(pODBC)
odbc_close(pODBC)

```

Output:

```
Columns Count : 3
1 - Mahmoud - 123456
2 - Ahmed - 123456
3 - Ibrahim - 123456
```

The program will create the file : mydb.db

Note : when I print the odbc drivers I see the long list that includes

```
SQLite3 ODBC Driver - UsageCount=1
SQLite ODBC Driver - UsageCount=1
SQLite ODBC (UTF-8) Driver - UsageCount=1
```

And I'm using "SQLite3 ODBC Driver".

100.46 Can I connect to dbase/harbour database?

You can connect to any database using ODBC

To connect to xbase files (*.DBF)

```
See "Using DBF Files using ODBC" + nl
pODBC = odbc_init()
See "Connect to database" + nl
odbc_connect(pODBC,"Driver={Microsoft dBase Driver (*.dbf)};" +
             "datasource=dBase Files;DriverID=277")

See "Select data" + nl
odbc_execute(pODBC,"select * from tel.dbf")
nMax = odbc_colcount(pODBC)
See "Columns Count : " + nMax + nl
while odbc_fetch(pODBC)
    See "Row data:" + nl
    for x = 1 to nMax
        see odbc_getdata(pODBC,x) + " - "
    next
end
See "Close database..." + nl
odbc_disconnect(pODBC)
odbc_close(pODBC)
```

Output

```
Using DBF Files using ODBC
Connect to database
Select data
Columns Count : 3
Row data:
Ahmad - Egypt - 234567 - Row data:
Fady - Egypt - 345678 - Row data:
Shady - Egypt - 456789 - Row data:
Mahmoud - Egypt - 123456 - Close database...
```

Also you can connect to a Visual FoxPro database (requires installing Visual FoxPro driver)

```
See "ODBC test 6" + nl
pODBC = odbc_init()
See "Connect to database" + nl
```

```
odbc_connect (pODBC,"Driver={Microsoft Visual FoxPro Driver};"+
    "SourceType=DBC;SourceDB=C:\PWCT19\ssbuild\PWCTDATA\CH1\Data\mydata.dbc;")
See "Select data" + nl
see odbc_execute(pODBC,"select * from t38") + nl
nMax = odbc_colcount(pODBC)
See "Columns Count : " + nMax + nl
while odbc_fetch(pODBC)
    See "Row data:" + nl
    for x = 1 to nMax
        see odbc_getdata(pODBC,x) + " - "
    next
end
See "Close database..." + nl
odbc_disconnect(pODBC)
odbc_close(pODBC)
```

100.47 Why setClickEvent() doesn't see the object methods directly?

setClickEvent(cCode) take a string contains code. The code will be executed when the event happens.

Ring support Many Programming Paradigms like Procedural, OOP, Functional and others.

But when you support many paradigms at the language level you can't know which paradigm will be used so you have two options

1. Provide General Solutions that works with many programming paradigms.
2. Provide Many Specific solutions where each one match a specific paradigm.

setClickEvent() and others belong to (General Solutions that works with many programming paradigms).

You just pass a string of code that will be executed without any care about classes and objects.

This code could be anything like calling a function, calling a method and setting variable value.

Some other languages force you to use OOP and call methods for events. Also some other languages uses anonymous functions that may get parameters like the current object.

Now we have the general solution (not restricted with any paradigm), In the future we may add specific solutions that match specific paradigms (OOP, Functional, Declarative and Natural).

100.48 Why I get Calling Function without definition Error?

Each program follow the next order

1 - Loading Files 2 - Global Variables and Statements 3 - Functions 4 - Packages, Classes and Methods

So what does that mean ?

1. **** No Functions comes After Classes ****
2. **** No command is required to end functions/methods/classes/packages ****

Look at this example

```
See "Hello"
test()
func test
```

```

    see "message from the test function!" + nl
class test

```

In the previous example we have a function called test() so we can call it directly using test()

In the next example, test() will become a method

```

See"Hello"
test()      # runtime error message
class test
    func test # Test() now is a method (not a function)
        see "message from the test method!" + nl

```

The errors comes when you define a method then try calling it directly as a function.

The previous program must be

```

See"Hello"
new test { test() }  # now will call the method
class test
    func test # Test() now is a method (not a function)
        see "message from the test method!" + nl

```

100.49 Can Ring work on Windows XP?

Ring can work on Windows XP and load extensions without problems.

Just be sure that the extension can work on Windows XP and your compiler version support that (modern compilers requires some flags to support XP)

Check this topic <https://blogs.msdn.microsoft.com/vcblog/2012/10/08/windows-xp-targeting-with-c-in-visual-studio-2012/>

For example, We added

```

/link /SUBSYSTEM:CONSOLE, "5.01"

```

To the batch file to support Windows XP

See : <https://github.com/ring-lang/ring/blob/master/src/buildvccomplete.bat>

100.50 How to extend RingQt and add more classes?

You have many options

In general you can extend Ring using C or C++ code

Ring from Ring code you can call C Functions or use C++ Classes & Methods

This chapter in the documentation explains this part in the language <http://ring-lang.sourceforge.net/doc/extension.html>

For example the next code in .c file can be compiled to a DLL file using the Ring library (.lib)

```

#include "ring.h"

RING_FUNC(ring_ringlib_dlfunc)
{

```

```

    printf("Message from dlfunc");
}

RING_API void ringlib_init(RingState *pRingState)
{
    ring_vm_funcregister("dlfunc", ring_ringlib_dlfunc);
}

```

Then from Ring you can load the DLL file using LoadLib() function then call the C function that called dlfunc() as any Ring function.

```

See "Dynamic DLL" + NL
LoadLib("ringlib.dll")
dlfunc()

```

Output

```

Dynamic DLL
Message from dlfunc

```

When you read the documentation you will know about how to get parameters like (strings, numbers, lists and objects)

And how to return a value (any type) from you function.

From experience, when we support a C library or C++ Library

We discovered that a lot of functions share a lot of code

To save our time, and to quickly generate wrappers for C/C++ Libraries to be used in Ring

We have this code generator

<https://github.com/ring-lang/ring/blob/master/extensions/codegen/parsec.ring>

The code generator is just a Ring program < 1200 lines of Ring code

The generator take as input a configuration file contains the C/C++ library information

like Functions Prototype, Classes and Methods, Constants, Enum, Structures and members , etc.

Then the generator will generate

- *.C File for C libraries (to be able to use the library functions)
- *.CPP File for C++ libraries (to be able to use C++ classes and methods)
- *.Ring File (to be able to use C++ classes as Ring classes)
- *.RH file (Constants)

To understand how the generator work check this extension for the Allegro game programming library

<https://github.com/ring-lang/ring/tree/master/extensions/ringallegro>

At first we have the configuration file

<https://github.com/ring-lang/ring/blob/master/extensions/ringallegro/allegro.cf>

To write this file, i just used the Allegro documentation + the Ring code generator rules

Then after executing the generator using this batch file

<https://github.com/ring-lang/ring/blob/master/extensions/ringallegro/gencode.bat>

or using this script

<https://github.com/ring-lang/ring/blob/master/extensions/ringallegro/gencode.sh>

I get the generated source code file

https://github.com/ring-lang/ring/blob/master/extensions/ringallegro/ring_allegro.c

The generated source code file (ring_allegro.c) is around 12,000 Lines of code (12 KLOC)

While the configuration file is less than 1 KLOC

To build the library (create the DLL files)

<https://github.com/ring-lang/ring/blob/master/extensions/ringallegro/buildvc.bat>

Also you can check this extension for the LibSDL Library

<https://github.com/ring-lang/ring/tree/master/extensions/ringsdl>

After this know you should know about

- 1 - Writing the configuration file
- 2 - Using the Code Generator
- 3 - Building your library/extension
- 4 - Using your library/extension from Ring code

Let us move now to you question about Qt

We have RingQt which is just an extension to ring (ringqt.dll)

You don't need to modify Ring.

1. You just need to modify RingQt
2. Or extend Ring with another extension based on Qt (but the same Qt version)

For the first option see the RingQt extension

<https://github.com/ring-lang/ring/tree/master/extensions/ringqt>

Configuration file

<https://github.com/ring-lang/ring/blob/master/extensions/ringqt/qt.cf>

To generate the source code

<https://github.com/ring-lang/ring/blob/master/extensions/ringqt/gencode.bat>

<https://github.com/ring-lang/ring/blob/master/extensions/ringqt/gencode.sh>

<https://github.com/ring-lang/ring/blob/master/extensions/ringqt/gencodeandroid.bat>

To build the DLL/so/Dylib files

<https://github.com/ring-lang/ring/blob/master/extensions/ringqt/buildmingw32.bat>

<https://github.com/ring-lang/ring/blob/master/extensions/ringqt/buildgcc.sh>

<https://github.com/ring-lang/ring/blob/master/extensions/ringqt/buildclang.sh>

Study RingQt

Learn about the options that you have

1. wrapping a Qt class directly
2. Creating a new class then wrapping your new class

For the second option (in the previous two points or in the two points before that)

You will create new classes in C++ code

Then you merge these classes to RingQt or provide special DLL for them (your decision)

If your work is general (will help others) just put it to RingQt.

if your work is special (to specific application) just put it in another extension.

100.51 How to add Combobox and other elements to the cells of a QTableWidgetItem?

Check the next code

```
Load "guilib.ring"
New qApp
{
    win1 = new QMainWindow() {
        setGeometry(100,100,1100,370)
        setWindowTitle("Using QTableWidgetItem")

        Table1 = new QTableWidgetItem (win1) {
            setRowCount(10) setColumnCount(10)
            setGeometry(0,0,800,400)
            setSelectionBehavior(QAbstractItemView_SelectRows)

            for x = 1 to 10
                for y = 1 to 10
                    item1 = new QTableWidgetItem("R"+X+"C"+Y)
                    setItem(x-1,y-1, item1)
                next
            next

            cmb = new QComboBox(Table1) {
                alist = ["one","two","three","four","five"]
                for x in alist addItem(x,0) next
            }

            setCellWidget(5, 5, cmb)
        }

        setCentralWidget(table1)
        show()
    }
    exec()
}
```

100.52 How to perform some manipulations on selected cells in QTableWidgetItem?

Check the next sample

```
Load "guilib.ring"

New qApp {
```

```

win1 = new QMainWindow() {
    setGeometry(100,100,800,600)
    setWindowTitle("Using QTableWidgetItem")
    Table1 = new QTableWidgetItem(win1) {
        setRowCount(10) setColumnCount(10)
        setGeometry(10,10,400,400)
        for x = 1 to 10
            for y = 1 to 10
                item1 = new QTableWidgetItem("10")
                setItem(x-1,y-1,item1)
            next
        next
    }
    btn1 = new QPushButton(win1) {
        setText("Increase")
        setGeometry(510,10,100,30)
        setClickEvent("pClick() ")
    }
    show()
}
exec()
}

func pClick
    for nRow = 0 to Table1.rowcount() - 1
        for nCol = 0 to Table1.columncount() - 1
            Table1.item(nRow,nCol) {
                if isSelected()
                    setText( " " + ( 10 + text() ) )
                ok
            }
        next
    next
next

```

100.53 Which of 3 coding styles are commonly used or recommended by the community?

1. **Just select any style of them but don't mix between the different styles in the same project** or at least in the same context (Implementation, Tests, Scripts, etc)

Note: State the rules in the start of each project and follow it.

2. You can create your style (by changing keywords) - The idea is about customization and freedom.

Note: It's better to change keywords and create new style only for a clear reason like using another natural language (Arabic, French, etc.)

3. **The First style is better (IMHO) for questions, tutorials and small applications/programs (Less than 5,000 LOC)**
Example : Ring Book, Most of Ring Samples and Applications.

4. The Third style is better(IMHO) for large applications and mainstream programmers

Example (Form Designer) : <https://github.com/ring-lang/ring/tree/master/applications/formdesigner>

LANGUAGE REFERENCE

In this chapter we will learn about

- Language keywords
- Language Functions
- Compiler Errors
- Runtime Errors
- Environment Errors
- Language Grammar
- Virtual Machine (VM) Instructions

101.1 Language Keywords

Keywords Count : 49

- again
- and
- but
- bye
- call
- case
- catch
- changingkeyword
- changingoperator
- class
- def
- do
- done
- else
- elseif
- end

- exit
- for
- from
- func
- get
- give
- if
- import
- in
- load
- loadsyntax
- loop
- new
- next
- not
- off
- ok
- on
- or
- other
- package
- private
- put
- return
- see
- step
- switch
- to
- try
- while
- endfunc
- endclass
- endpackage

101.2 Language Functions

Functions Count : 198

```
len() add() del() sysget() clock() lower()
upper() input() ascii() char() date() time()
filename() getchar() system() random() timelist() adddays()
difftdays() version() clocksperssecond() prevfilename() swap() shutdown()
isstring() isnumber() islist() type() isnull() isobject()
hex() dec() number() string() str2hex() hex2str()
str2list() list2str() left() right() trim() copy()
substr() lines() strcmp() eval() raise() assert()
isalnum() isalpha() iscntrl() isdigit() isgraph() islower()
isprint() ispunct() isspace() isupper() isxdigit() locals()
globals() functions() cfunctions() islocal() isglobal() isfunction()
iscfunction() packages() ispackage() classes() isclass() packageclasses()
ispackageclass() classname() objectid() attributes() methods() isattribute()
ismethod() isprivateattribute() isprivatemethod()
addattribute() addmethod() getattribute()
setattribute() mergemethods() packagename() ringvm_fileslist()
ringvm_calllist() ringvm_memorylist()
ringvm_functionslist() ringvm_classeslist() ringvm_packageslist()
ringvm_cfunctionslist() ringvm_settrace() ringvm_tracedata()
ringvm_traceevent() ringvm_tracefunc() ringvm_scopescount()
ringvm_evalinscope() ringvm_passerror() ringvm_hideerrormsg()
ringvm_callfunc() list() find() min() max() insert()
sort() reverse() binarysearch() sin() cos() tan()
asin() acos() atan() atan2() sinh() cosh()
tanh() exp() log() log10() ceil() floor()
fabs() pow() sqrt() unsigned() decimals() murmur3hash()
fopen() fclose() fflush() freopen() tempfile() tempname()
fseek() ftell() rewind() fgetpos() fsetpos() clearerr()
feof() ferror() perror() rename() remove() fgetc()
fgets() fputc() fputs() ungetc() fread() fwrite()
dir() read() write() fexists() int2bytes() float2bytes()
double2bytes() bytes2int() bytes2float()
bytes2double() ismsdos() iswindows()
iswindows64() isunix() ismacosx() islinux() isfreebsd() isandroid()
windowsnl() currentdir() exefilename() chdir() exefolder() loadlib()
closelib() callgc() varptr() intvalue() object2pointer() pointer2object()
nullpointer() space() ptrcmp() ring_state_init()
ring_state_runcode() ring_state_delete()
ring_state_runfile() ring_state_findvar() ring_state_newvar()
ring_state_runobjectfile() ring_state_setvar()
```

101.3 Compiler Errors

- Error (C1) : Error in parameters list, expected identifier
- Error (C2) : Error in class name
- Error (C3) : Unclosed control strucutre, 'ok' is missing
- Error (C4) : Unclosed control strucutre, 'end' is missing
- Error (C5) : Unclosed control strucutre, next is missing
- Error (C6) : Error in function name

- Error (C7) : Error in list items
- Error (C8) : Parentheses ‘)’ is missing
- Error (C9) : Brackets ‘]’ is missing
- Error (C10) : Error in parent class name
- Error (C11) : Error in expression operator
- Error (C12) : No class definition
- Error (C13) : Error in variable name
- Error (C14) : Try/Catch miss the Catch keyword!
- Error (C15) : Try/Catch miss the Done keyword!
- Error (C16) : Error in Switch statement expression!
- Error (C17) : Switch statement without OFF
- Error (C18) : Missing closing brace for the block opened!
- Error (C19) : Numeric Overflow!
- Error (C20) : Error in package name
- Error (C21) : Unclosed control strucutre, ‘again’ is missing
- Error (C22) : Function redefinition, function is already defined!
- Error (C23) : Using ‘(’ after number!
- Error (C24) : The parent class name is identical to the subclass name
- Error (C25) : Trying to access the self reference after the object name”
- Error (C26) : Class redefinition, class is already defined!

101.4 Runtime Errors

- Error (R1) : Cann’t divide by zero !
- Error (R2) : Array Access (Index out of range) !
- Error (R3) : Calling Function without definition !
- Error (R4) : Stack Overflow !
- Error (R5) : Can’t access the list item, Object is not list !
- Error (R6) : Variable is required
- Error (R7) : Can’t assign to a string letter more than one character
- Error (R8) : Variable is not a string
- Error (R9) : Using exit command outside loops
- Error (R10) : Using exit command with number outside the range
- Error (R11) : error in class name, class not found!
- Error (R12) : error in property name, property not found!
- Error (R13) : Object is required

- Error (R14) : Calling Method without definition !
- Error (R15) : error in parent class name, class not found!
- Error (R16) : Using braces to access unknown object !
- Error (R17) : error, using 'Super' without parent class!
- Error (R18) : Numeric Overflow!
- Error (R19) : Calling function with less number of parameters!
- Error (R20) : Calling function with extra number of parameters!
- Error (R21) : Using operator with values of incorrect type
- Error (R22) : Using loop command outside loops
- Error (R23) : Using loop command with number outside the range
- Error (R24) : Using uninitialized variable
- Error (R25) : Error in package name, Package not found!
- Error (R26) : Calling private method from outside the class
- Error (R27) : Using private attribute from outside the class
- Error (R28) : Using bad data type as step value
- Error (R29) : Using bad data type in for loop
- Error (R30) : parent class name is identical to child class name
- Error (R31) : Trying to destroy the object using the self reference
- Error (R32) : The CALL command expect a variable contains string!
- Error (R33) : Bad decimals number (correct range ≥ 0 and ≤ 14) !
- Error (R34) : Variable is required for the assignment operation
- Error (R35) : Can't create/open the file!
- Error (R36) : The column number is not correct! It's greater than the number of columns in the list
- Error (R37) : Sorry, The command is not supported in this context
- Error (R38) : Runtime Error in loading the dynamic library!
- Error (R39) : Error occurred creating unique filename.

101.5 Environment Errors

- Error (E1) : Caught SegFault
- Error (E2) : Out of Memory
- Error (E3) : Deleting scope while no scope!
- Error (E4) : Long VM Instruction!

101.6 Language Grammar

Program \rightarrow {statement}

Statement \rightarrow ‘package’ <Identifier> { ‘.’ <Identifier> } [‘{’ {statement} ‘}’] [‘end’] ‘endpackage’]

Statement \rightarrow ‘class’ <Identifier> [‘from’ ‘:’ ‘|’ <Identifier>] [‘{’ {statement} ‘}’] [‘end’] ‘endclass’]

Statement \rightarrow ‘func’ ‘|’ ‘def’ <Identifier> [ParaList] [‘{’ {statement} ‘}’] [‘end’] ‘endfunc’]

Statement \rightarrow ‘import’ <Identifier> { ‘.’ <Identifier> }

Statement \rightarrow ‘private’

Statement \rightarrow ‘load’ <Literal>

Statement \rightarrow ‘loadsyntax’ <Literal>

Statement \rightarrow ‘changingkeyword’ <OldKeyword> <NewKeyword>

Statement \rightarrow ‘changingoperator’ <OldOperator> <NewOperator>

Statement \rightarrow ‘see’ ‘|’ ‘put’ <Expr>

Statement \rightarrow ‘give’ ‘|’ ‘get’ <Identifier>

Statement \rightarrow ‘if’ <Expr> [‘{’ {statement} [{ ‘but’ ‘|’ ‘elseif’ <Expr> {Statement} }] [‘else’ {Statement}] ‘ok’] ‘end’] ‘|’ }

Statement \rightarrow ‘Switch’ <Expr> [‘{’ { ‘on’ ‘|’ ‘case’ <Expr> {statement} } [‘other’ {Statement}] [‘off’] ‘end’] ‘|’ }

Statement \rightarrow ‘for’ <Identifier> ‘=’ <Expr> ‘to’ <Expr> [‘step’ <Expr>] [‘{’ {Statement} ‘next’] ‘end’] ‘|’ }

Statement \rightarrow ‘for’ <Identifier> ‘in’ <Expr> [‘step’ <Expr>] [‘{’ {statement} ‘next’] ‘end’] ‘|’ }

Statement \rightarrow ‘while’ <Expr> [‘{’ {statement} ‘end’] ‘|’ }

Statement \rightarrow ‘do’ {statement} ‘again’ <Expr>

Statement \rightarrow ‘try’ {statement} [‘{’ ‘catch’ {statement} ‘done’] ‘end’] ‘|’ }

Statement \rightarrow ‘return’ <Expr>

Statement \rightarrow ‘bye’

Statement \rightarrow ‘exit’

Statement \rightarrow ‘loop’

Statement \rightarrow <Expr>

Statement \rightarrow epsilon

ParaList \rightarrow epsilon

ParaList \rightarrow [‘(’ <Identifier> [{ ‘,’ <Identifier> }] [‘)’]]

Expr \rightarrow <LogicNot> [{ ‘and’ ‘|’ ‘or’ <LogicNot> }]

LogicNot \rightarrow [‘not’] <EqualOrNot>

EqualOrNot \rightarrow [‘=’ ‘|’ ‘!’] <Compare>

Compare \rightarrow <BitOrXor> [{ ‘<’ ‘|’ ‘>’ ‘|’ ‘<=’ ‘|’ ‘>=’ <BitOrXor> }]

BitOrXor \rightarrow <BitAnd> [{ ‘|’ ‘|’ ‘^’ <BitAnd> }]

BitAnd \rightarrow <BitShift> [{ ‘&’ <BitShift> }]

BitShift \rightarrow <Arithmetic> [{ ‘<<’ ‘|’ ‘>>’ <Arithmetic> }]

Arithmetic \rightarrow $\langle \text{Term} \rangle [\{ '+' | '-' \} \langle \text{Term} \rangle]$
 Term \rightarrow $\langle \text{Range} \rangle [\{ '*' | '/' | '%' \} \langle \text{Range} \rangle]$
 Range \rightarrow $\langle \text{Factor} \rangle [':' \langle \text{Factor} \rangle]$
 Factor \rightarrow $\langle \text{Identifier} \rangle [\{ \text{Mixer} \} ['=' \langle \text{Expr} \rangle]$
 Factor \rightarrow $\langle \text{Number} \rangle$
 Factor \rightarrow $\langle \text{Literal} \rangle$
 Factor \rightarrow $'.' \langle \text{Identifier} \rangle$
 Factor \rightarrow $'-' \langle \text{Expr} \rangle$
 Factor \rightarrow $'(' \langle \text{Expr} \rangle ')'$
 Factor \rightarrow $\langle \text{List} \rangle$
 Factor \rightarrow $'new' \langle \text{Identifier} \rangle$
 Factor \rightarrow $\langle \text{AnonymousFunction} \rangle$
 Factor \rightarrow $'call' \langle \text{identifier} \rangle ['.' \langle \text{Identifier} \rangle] (' \langle \text{Parameters} \rangle ')'$
 List \rightarrow $'[' [\langle \text{Expr} \rangle [',' \langle \text{Expr} \rangle]] ']'$
 Mixer \rightarrow $\{ '.' \langle \text{Identifier} \rangle \}$
 Mixer \rightarrow $'[' \langle \text{Expr} \rangle ']'$
 Mixer \rightarrow $'(' [\langle \text{Expr} \rangle [',' \langle \text{Expr} \rangle]] ')'$
 Mixer \rightarrow $\{ '{' \{ \text{Statement} \} '}'$
 AnonymousFunction \rightarrow $'func'|'def'| \langle \text{ParaList} \rangle ['{' \{ \text{Statement} \} '}'$

101.7 Virtual Machine (VM) Instructions

Definitions :-

- VM : Virtual Machine
- Stack : VM Stack
- IR : Instruction Register
- PC : Program Counter
- VP : Variable Pointer
- Stack[nSize] : Last Item in the Stack (Last In - First Out)
- VV : Variable Value (We have a Pointer to a variable, And we access this variable value)

(Stack and Variables)

Operation	Description
• ICO_PUSHC	Add string from the IR to the stack
• ICO_PUSHN	Add number from the IR to the stack
• ICO_PUSHV	Replace VP in the stack[nSize] with the variable value
• ICO_LOADADDRESS	Read variable name from the IR, push VP to the stack
• ICO_ASSIGNMENT	Stack[nSize-1] VV = Stack[nSize] VV , POP Stack[nSize]
• ICO_INC	Increment Number in Stack[nSize] by 1
• ICO_LOADAPUSHV	The same as ICO_LOADADDRESS then ICO_PUSHV
• ICO_NEWLINE	Store new line number (debug info)
• ICO_FREESTACK	Remove all items from the stack , nSize = 0
• ICO_FILENAME	Store the source code file name (debug info)
• ICO_FREELOADASCOPE	Free the Scope List of the current Expression

(Jump)

Operation	Description
• ICO_JUMP	Set PC to new value from the IR
• ICO_JUMPZERO	If Stack[nSize] is a number = 0 then Set PC to new value from the IR
• ICO_JUMPFOR	End of for loop
• ICO_JUMPONE	If Stack[nSize] is a number = 1 then Set PC to new value from the IR
• ICO_JUMPZERO2	As ICO_JUMPZERO but add 1 to the stack (required for many 'AND' conditions)
• ICO_JUMPONE2	As ICO_JUMPONE but add 1 to the stack (required for many 'OR' conditions)

(Compare)

Operation	Description
• ICO_LESSEQUAL	If $\text{stack}[\text{nSize}-1] \leq \text{stack}[\text{nSize}]$, POP $\text{stack}[\text{nSize}]$, set $\text{Stack}[\text{nSize}-1] = 1$ else set $\text{Stack}[\text{nSize}-1] = 0$
• ICO_EQUAL	If $\text{stack}[\text{nSize}-1] = \text{stack}[\text{nSize}]$, POP $\text{stack}[\text{nSize}]$, set $\text{Stack}[\text{nSize}-1] = 1$ else set $\text{Stack}[\text{nSize}-1] = 0$
• ICO_LESS	If $\text{stack}[\text{nSize}-1] < \text{stack}[\text{nSize}]$, POP $\text{stack}[\text{nSize}]$, set $\text{Stack}[\text{nSize}-1] = 1$ else set $\text{Stack}[\text{nSize}-1] = 0$
• ICO_GREATER	If $\text{stack}[\text{nSize}-1] > \text{stack}[\text{nSize}]$, POP $\text{stack}[\text{nSize}]$, set $\text{Stack}[\text{nSize}-1] = 1$ else set $\text{Stack}[\text{nSize}-1] = 0$
• ICO_GREATEREQUAL	If $\text{stack}[\text{nSize}-1] \geq \text{stack}[\text{nSize}]$, POP $\text{stack}[\text{nSize}]$, set $\text{Stack}[\text{nSize}-1] = 1$ else set $\text{Stack}[\text{nSize}-1] = 0$
• ICO_NOTEQUAL	If $\text{stack}[\text{nSize}-1] \neq \text{stack}[\text{nSize}]$, POP $\text{stack}[\text{nSize}]$, set $\text{Stack}[\text{nSize}-1] = 1$ else set $\text{Stack}[\text{nSize}-1] = 0$

(Math)

Operation	Description
• ICO_SUM	$\text{Stack}[\text{nSize}-1] = \text{Stack}[\text{nSize}-1] + \text{Stack}[\text{nSize}]$, POP $\text{stack}[\text{nSize}]$
• ICO_SUB	$\text{Stack}[\text{nSize}-1] = \text{Stack}[\text{nSize}-1] - \text{Stack}[\text{nSize}]$, POP $\text{stack}[\text{nSize}]$
• ICO_MUL	$\text{Stack}[\text{nSize}-1] = \text{Stack}[\text{nSize}-1] * \text{Stack}[\text{nSize}]$, POP $\text{stack}[\text{nSize}]$
• ICO_DIV	$\text{Stack}[\text{nSize}-1] = \text{Stack}[\text{nSize}-1] / \text{Stack}[\text{nSize}]$, POP $\text{stack}[\text{nSize}]$
• ICO_MOD	$\text{Stack}[\text{nSize}-1] = \text{Stack}[\text{nSize}-1] \% \text{Stack}[\text{nSize}]$, POP $\text{stack}[\text{nSize}]$
• ICO_NEG	$\text{Stack}[\text{nSize}] = - \text{Stack}[\text{nSize}-1]$
• ICO_PLUSPLUS	$\text{Stack}[\text{nSize}] = \text{Stack}[\text{nSize}] + 1$
• ICO_MINUSMINUS	$\text{Stack}[\text{nSize}] = \text{Stack}[\text{nSize}] - 1$

(Logic)

Operation	Description
• ICO_AND	$\text{Stack}[\text{nSize}-1] = \text{Stack}[\text{nSize}-1] \&\& \text{Stack}[\text{nSize}]$, POP $\text{stack}[\text{nSize}]$
• ICO_OR	$\text{Stack}[\text{nSize}-1] = \text{Stack}[\text{nSize}-1] \parallel \text{Stack}[\text{nSize}]$, POP $\text{stack}[\text{nSize}]$
• ICO_NOT	$\text{Stack}[\text{nSize}] = ! \text{Stack}[\text{nSize}]$

(Lists)

Operation	Description
• ICO_LISTSTART	Start New List in Temp. Memory
• ICO_LISTITEM	Add List Item
• ICO_LISTEND	End List
• ICO_LOADINDEXADDRESS	Stack[nSize-1] = Stack[nSize-1] VV [Stack[nSize]] , POP stack[nSize]

(Functions)

Operation	Description
• ICO_LOADFUNC	Find function
• ICO_CALL	Call function
• ICO_RETURN	Return from function
• ICO_RETNUL	Return NULL from function
• ICO_RETFROMEVAL	Return after eval()
• ICO_RETITEMREF	Return the list item reference - not the value
• ICO_NEWFUNC	Start new function
• ICO_BLOCKFLAG	Flag to determine where to jump later (after ICO_RETURN)
• ICO_FUNCSEX	Start executing function
• ICO_ENDFUNCSEX	End function execution
• ICO_ANONYMOUS	Anonymous function

(User Interface)

Operation	Description
• ICO_PRINT	Print value to the standard output
• ICO_GIVE	Get input from the keyboard

(End Program/Loop)

Operation	Description
• ICO_BYE	End execution of VM
• ICO_EXITMARK	Place to exit to from a loop
• ICO_POPEXITMARK	Remove exit mark
• ICO_EXIT	Break from one loop or more
• ICO_LOOP	Continue to next loop

(For Better Performance)

Operation	Description
• ICO_PUSHP	Push pointer to the stack
• ICO_INCP	Increment variable value using pointer
• ICO_PUSHVP	Push value of variable using variable pointer
• ICO_INCPJUMP	Increment then jump
• ICO_INCPJUMP	Increment using pointer then jump
• ICO_JUMPVARLENUM	Jump if variable value is \leq numeric value
• ICO_JUMPVARPLENUM	Jump if variable value (using pointer) \leq numeric value
• ICO_LOADFUNCP	Push function pointer
• ICO_PUSHLOCAL	Push pointer to local variable
• ICO_INCLPJUMP	Increment value using pointer to local variable then jump
• ICO_JUMPVARLPLENUM	Jump if the variable value (using pointer) \leq numeric value
• ICO_INCPJUMPSTEP1	Increment value using variable pointer then jump (for loop step = 1)
• ICO_JUMPVARPLENUMSTEP1	Increment value using variable pointer then jump (for loop step = 1)

(Try-Catch-Done)

Operation	Description
• ICO_TRY	Start try region
• ICO_DONE	End try region

(Duplicate and Range)

Operation	Description
• ICO_DUPLICATE	Duplicate stack value
• ICO_RANGE	Create list from value to value

(OOP)

Operation	Description
• ICO_NEWOBJ	Create new object, get class name from the IR, push object pointer to the stack.
• ICO_SETSCOPE	Called after creating new object, set the active scope to be the object scope.
• ICO_LOADSUBADDRESS	Get object attribute, push the pointer to the stack.
• ICO_LOADMETHOD	Find object method
• ICO_AFTERCALLMETHOD	Used after calling a method - normal case
• ICO_AFTERCALLMETHOD2	Used after calling a method - second case
• ICO_NEWCLASS	Start new class region
• ICO_BRACESTART	Open brace
• ICO_BRACEEND	End brace
• ICO_IMPORT	Import package
• ICO_PRIVATE	start private attributes region
• ICO_SETPROPERTY	set attribute value - check for setter.
• ICO_CALLCLASSINIT	call call init() method.

(Other)

Operation	Description
• ICO_SETREFERENCE	Copy by reference
• ICO_KILLREFERENCE	Remove reference
• ICO_ASSIGNMENTPOINTER	Determine the left side variable
• ICO_BEFOREEQUAL	Determine operators like += , -= , ... etc

(Bitwise Operators)

Operation	Description
• ICO_BITAND	$\text{Stack}[\text{nSize}-1] = \text{Stack}[\text{nSize}-1] \& \text{Stack}[\text{nSize}]$, POP $\text{stack}[\text{nSize}]$
• ICO_BITOR	$\text{Stack}[\text{nSize}-1] = \text{Stack}[\text{nSize}-1] \mid \text{Stack}[\text{nSize}]$, POP $\text{stack}[\text{nSize}]$
• ICO_BITXOR	$\text{Stack}[\text{nSize}-1] = \text{Stack}[\text{nSize}-1] \wedge \text{Stack}[\text{nSize}]$, POP $\text{stack}[\text{nSize}]$
• ICO_BITNOT	$\text{Stack}[\text{nSize}] = ! \text{Stack}[\text{nSize}]$
• ICO_BITSHL	$\text{Stack}[\text{nSize}-1] = \text{Stack}[\text{nSize}-1] \ll \text{Stack}[\text{nSize}]$, POP $\text{stack}[\text{nSize}]$
• ICO_BITSHR	$\text{Stack}[\text{nSize}-1] = \text{Stack}[\text{nSize}-1] \gg \text{Stack}[\text{nSize}]$, POP $\text{stack}[\text{nSize}]$

(For Step)

Operation	Description
• ICO_STEPNUMBER	Determine step number in for loop
• ICO_POPSTEP	POP step number from steps stack
• ICO_LOADAFIRST	Load the first address of variable name

RESOURCES

In this section you will find resources about the language

102.1 Ring Language Website

For news about the language check the website

<http://ring-lang.net>

<http://ring-lang.sf.net>

102.2 Ring Group

For questions use the Ring Group (English)

<https://groups.google.com/forum/#!forum/ring-lang>

102.3 Contact the Authors

Name : Eng. Mahmoud Samir Fayed

Country : Egypt

Email : msfclipper@yahoo.com

Facebook : <https://facebook.com/mahmoudfayed1986>

LinkedIn : <https://sa.linkedin.com/in/mahmoudfayed1986>

Name : Dr. Atif M. Alamri

Country : Saudi Arabia

Email : atif@ksu.edu.sa

LinkedIn : <https://sa.linkedin.com/in/dr-atif-alamri-8b341747>

Ring Team : <http://ring-lang.sourceforge.net/team.html>

- 3D Cube and Texture
 - Using RingOpenGL and RingAllegro for 3D Graphics, 570
- Access List Items by String Index
 - Lists, 212
- Access Objects Using Braces
 - Object Oriented Programming, 289
- Access String Letters
 - Strings, 214
- Accessing the class attributes from braces inside class methods
 - Scope Rules, 718
- Add Items
 - Lists, 206
- addattribute()
 - Reflection and Meta-programming, 317
- AddDays()
 - Date and Time, 222
- Adding code to the generated code
 - Code Generator, 813
- Adding Hyperlink to QLabel
 - Desktop and Mobile Development, 621
- addmethod()
 - Reflection and Meta-programming, 317
- Analog Clock
 - Applications developed in little hours, 6
- Animate Class
 - Game Engine for 2D Games, 473
- Animate Events
 - Game Engine for 2D Games, 484
- Animation
 - Game Engine for 2D Games, 478
- Animation and Functions
 - Game Engine for 2D Games, 479
- Anonymous and Nested Functions
 - Functional Programming, 302
- Application Class
 - Web Development (CGI Library), 435
- Applications
 - How to contribute?, 142
- Applications developed in little hours
 - Analog Clock, 6
 - Calculator Application, 10
 - FetchStockData Application, 3
 - Fifteen Puzzle Game 2, 4
 - Google API Shortener Application, 5
 - Innovative, 19
 - Introduction, 1
 - Practical, 20
 - Quotes about Ring, 1
 - Samples in this book, 13
 - Squares Puzzle Game, 8
 - TicTacToe Game, 7
 - Video-Music-Player Application, 9
 - Werdy Application, 12
 - Windows StartUp Manager Application, 11
- apppath()
 - Stdlib Functions, 324
- Arithmetic Operators
 - Operators, 173
- Ascii()
 - Data Type, 230
- Assert()
 - Eval() and Debugging, 256
- Assignment Operators
 - Operators, 174
- attributes()
 - Reflection and Meta-programming, 314
- Better Call Command
 - What is new in Ring 1.2?, 121
- Better Code Generator for Extensions
 - What is new in Ring 1.1?, 131
 - What is new in Ring 1.5?, 72
- Better Documentation
 - What is new in Ring 1.1?, 131
- Better Documentation Generator for Extensions
 - What is new in Ring 1.5?, 73
- Better Functions
 - What is new in Ring 1.2?, 118
- Better LoopExit Command
 - What is new in Ring 1.3?, 112
- Better Natural Language Programming Support
 - What is new in Ring 1.1?, 123
- Better Objects Library

- What is new in Ring 1.5?, 59
- Better Quality
 - What is new in Ring 1.2?, 121
 - What is new in Ring 1.5?, 81
- Better Ring For Android
 - What is new in Ring 1.6?, 38
- Better Ring Notepad
 - What is new in Ring 1.2?, 118
 - What is new in Ring 1.3?, 108
 - What is new in Ring 1.5?, 50
 - What is new in Ring 1.6?, 41
- Better RingQt
 - What is new in Ring 1.2?, 118
 - What is new in Ring 1.3?, 104
 - What is new in Ring 1.4?, 99
 - What is new in Ring 1.5?, 57
 - What is new in Ring 1.6?, 43
- Better RingREPL
 - What is new in Ring 1.6?, 44
- Better RingVM
 - What is new in Ring 1.6?, 44
- Better Scripts for building Ring
 - What is new in Ring 1.6?, 40
- Better StdLib
 - What is new in Ring 1.3?, 112
 - What is new in Ring 1.4?, 97
 - What is new in Ring 1.5?, 53
 - What is new in Ring 1.6?, 44
- Better WebLib
 - What is new in Ring 1.4?, 97
 - What is new in Ring 1.5?, 54
- binarydigits()
 - Stdlib Functions, 333
- Bitwise Operators
 - Operators, 174
- BraceError() Method
 - Natural Language Programming, 377
- BraceExprEval Method
 - Natural Language Programming, 376
- BraceStart and BraceEnd Methods
 - Natural Language Programming, 375
- Branching
 - Control Structures, 176
 - Control Structures - Second Style, 183
 - Control Structures - Third Style, 186
- BreakPoint
 - The Trace Library and the Interactive Debugger, 792
- Building From Source Code
 - Building using CMake, 140
 - Building using Fedora Linux, 136
 - Building using MacOS X, 138
 - Building using Microsoft Windows, 133
 - Building using Ubuntu Linux, 135
 - Introduction, 132
- Building Games For Android
 - Building the project, 520
 - Download Requirements and Update the Android SDK, 519
 - Introduction, 518
 - Project Folder, 519
- Building RingQt Applications for Mobile
 - Comments about developing for Android using RingQt, 697
 - Download Requirements, 695
 - Install Qt for Android, 695
 - Introduction, 694
 - Update the Android SDK, 695
 - Using Ring2EXE, 698
- Building standalone console application
 - Distributing Ring Application using Ring2EXE, 762
- Building the Cards Game for Mobile using RingQt
 - Distributing Ring Application using Ring2EXE, 764
- Building the Form Designer for Mobile using RingQt
 - Distributing Ring Application using Ring2EXE, 766
- Building the project
 - Building Games For Android, 520
- Building the Weight History Application for Mobile using RingQt
 - Distributing Ring Application using Ring2EXE, 765
- Building using CMake
 - Building From Source Code, 140
- Building using Fedora Linux
 - Building From Source Code, 136
- Building using MacOS X
 - Building From Source Code, 138
- Building using Microsoft Windows
 - Building From Source Code, 133
- Building using Ubuntu Linux
 - Building From Source Code, 135
- Calculator Application
 - Applications developed in little hours, 10
 - What is new in Ring 1.5?, 49
- Call Functions
 - Functions - First Style, 193
 - Functions - Second Style, 197
 - Functions - Third Style, 201
- callgc()
 - Low Level Functions, 770
- Calling a function sharing the name with a method in the current class
 - Scope Rules for Functions and Methods, 729
- Can I connect to dbase/harbour database?
 - Frequently Asked Questions, 1802
- Can Ring work on Windows XP?
 - Frequently Asked Questions, 1804
- capitalized()
 - Stdlib Functions, 327

- cfunctions()
 - Reflection and Meta-programming, 307
- CGI Support
 - Command Line Options, 757
- Change Focus
 - Desktop and Mobile Development, 656
- Change Language Keywords
 - Syntax Flexibility, 731
- Change Language Operators
 - Syntax Flexibility, 732
- Change the '=' operator to 'is'
 - Natural Language Programming, 373
- Change the Ring Keyword 'And'
 - Natural Language Programming, 371
- Change the Ring Operator '+'
 - Natural Language Programming, 372
- Change: Basic Extensions are separated from RingVM
 - What is new in Ring 1.4?, 92
- changestring()
 - Stdlib Functions, 336
- Char()
 - Data Type, 230
- ChDir() Function
 - System Functions, 252
- Check Character
 - Data Type, 225
- Check Data Type
 - Data Type, 224
- Check Parameters Count
 - Extension, 800
- Check Parameters Type
 - Extension, 801
- Classes and Objects
 - Object Oriented Programming, 287
- Classes and their Methods to use the default events
 - Desktop and Mobile Development, 686
- classes()
 - Reflection and Meta-programming, 311
- classname()
 - Reflection and Meta-programming, 313
- Clean Natural Code
 - Natural Language Programming, 378
- Clearerr()
 - Files, 242
- clock()
 - Date and Time, 220
- Close Window Event
 - RingLibSDL, 465
- ClosPerSecond()
 - Date and Time, 220
- Code Generator
 - Adding code to the generated code, 813
 - Comments in configuration file, 815
 - Configuration file, 811
 - Configuration file for the Allegro library, 816
 - Configuration Files Examples, 829
 - Constants Type, 816
 - Defining Constants, 814
 - Determine Structure Members Types, 814
 - Enum and Numbers, 815
 - Executing code during code generation, 815
 - Filtering using Expressions, 815
 - function prototype, 811
 - Introduction, 810
 - Prefix for Functions Names, 813
 - Qt configuration file, 820
 - Register New Functions, 814
 - Threads Support, 818
 - Using configuration file that wrap C++ library, 820
 - Using the tool, 811
 - Wrap structures, 813
 - Wrapping C++ Classes, 819
- CodeEditor Class
 - RingQt Classes Reference, 1739
- Command Line Options
 - CGI Support, 757
 - Generate Object File, 758
 - Introduction, 743
 - No Run, 757
 - Performance, 757
 - Printing Final Intermediate Code, 752
 - Printing Instruction Operation Code, 757
 - Printing Intermediate Code, 750
 - Printing Rules, 746
 - Printing Tokens, 744
- Comments about developing for Android using RingQt
 - Building RingQt Applications for Mobile, 697
- Comments about evaluation
 - Control Structures - First Style, 181
- Comments in configuration file
 - Code Generator, 815
- Compact Syntax
 - Language Design, 30
- Compiler and Virtual Machine (VM)
 - How to contribute?, 142
- Compiler Errors
 - Reference, 1811
- Composition
 - Object Oriented Programming, 290
- Configuration file
 - Code Generator, 811
- Configuration file for the Allegro library
 - Code Generator, 816
- Configuration Files Examples
 - Code Generator, 829
- Configure the Apache web server
 - Web Development (CGI Library), 387
- Conflict between Class Attributes and Local Variables

- Scope Rules, 715
- Conflict between Global Variables and Class Attributes
 - Frequently Asked Questions, 1790
 - Scope Rules, 714
- Conflict between self inside braces and self in the class region
 - Scope Rules, 721
- Constants Type
 - Code Generator, 816
- Constructor methods in Ring
 - Frequently Asked Questions, 1786
- Contact the Author
 - Resources, 1822
- Context Menu
 - Desktop and Mobile Development, 603
- Control Structures
 - Branching, 176
 - Looping, 187
- Control Structures - First Style
 - Comments about evaluation, 181
 - Do Again Loop, 179
 - Exit, 179
 - Exit from two loops, 179
 - Exit/Loop inside sub functions, 180
 - For In Loop, 178
 - for in to modify lists, 178
 - For Loop, 177
 - Introduction, 175
 - Loop Command, 180
 - Looping, 177
 - Short-circuit evaluation, 180
 - Step Option, 178
 - Switch Statement, 176
 - While Loop, 177
- Control Structures - Second Style
 - Branching, 183
 - Exceptions, 185
 - For In Loop, 185
 - For Loop, 184
 - If Statement, 183
 - Introduction, 182
 - Looping, 184
 - Switch Statement, 183
- Control Structures - Third Style
 - Branching, 186
 - Exceptions, 189
 - For In Loop, 189
 - For Loop, 188
 - If Statement, 186
 - Introduction, 185
 - Switch Statement, 186
 - While Loop, 187
- Conversion
 - Data Type, 229
- Conversion Class
 - Stdlib Classes, 358
- Convert between Numbers and Bytes
 - What is new in Ring 1.4?, 96
- Convert Letters Case
 - Strings, 214
- Cookies
 - Web Development (CGI Library), 399
- Copy Lists
 - Lists, 210
- Copy()
 - Strings, 216
- Could you explain the output of the StrCmp() function?
 - Frequently Asked Questions, 1793
- Create Database
 - MySQL Functions, 271
- Create Executable File
 - Getting Started - First Style, 143
 - Getting Started - Second Style, 146
 - Getting Started - Third Style, 148
- Create Lists
 - Lists, 206
- Create Table and Insert Data
 - MySQL Functions, 272
- Create Window
 - RingLibSDL, 462
- Create Zip File
 - RingZip, 445
- Creating a Class for each Window in GUI applications
 - Scope Rules, 720
- Creating Menubar
 - Desktop and Mobile Development, 600
- Creating more than one window
 - Desktop and Mobile Development, 639
- Creating Reports using the WebLib and the GUILib
 - Desktop and Mobile Development, 691
- Creating StatusBar
 - Desktop and Mobile Development, 605
- Creating the Game Window
 - Game Engine for 2D Games, 473
- Creating the Qt resource file using Folder2qrc
 - Distributing Ring Application using Ring2EXE, 768
- Creating Toolbar
 - Desktop and Mobile Development, 604
- Creating Windows Installer
 - Distributing Ring Application, 760
- CRUD Example using MVC
 - Web Development (CGI Library), 421
- CurrentDir() Function
 - System Functions, 252
- Data Type
 - Ascii(), 230
 - Char(), 230

- Check Character, [225](#)
- Check Data Type, [224](#)
- Conversion, [229](#)
- Dec(), [230](#)
- Hex(), [230](#)
- Hex2str(), [231](#)
- Introduction, [223](#)
- IsAlNum(), [226](#)
- IsAlpha(), [226](#)
- IsCntrl(), [226](#)
- IsDigit(), [227](#)
- IsGraph(), [227](#)
- IsList(), [225](#)
- IsLower(), [227](#)
- IsNULL(), [225](#)
- IsNumber(), [224](#)
- IsPrint(), [227](#)
- IsPunct(), [228](#)
- IsSpace(), [228](#)
- IsString(), [224](#)
- IsUpper(), [228](#)
- IsXdigit(), [228](#)
- Number(), [229](#)
- Str2Hex(), [231](#)
- String(), [229](#)
- Type(), [225](#)
- Database, ModelBase & ControllerBase classes
 - Web Development (CGI Library), [429](#)
- DataType Class
 - Stdlib Classes, [357](#)
- Date and Time
 - AddDays(), [222](#)
 - clock(), [220](#)
 - ClosPerSecond(), [220](#)
 - Date(), [220](#)
 - DiffDays(), [222](#)
 - EpochTime(), [222](#)
 - Introduction, [219](#)
 - Time(), [220](#)
 - TimeList(), [221](#)
- Date()
 - Date and Time, [220](#)
- DateTime Class
 - Stdlib Classes, [352](#)
- dayofweek()
 - Stdlib Functions, [334](#)
- Debug Class
 - Stdlib Classes, [356](#)
- Dec()
 - Data Type, [230](#)
- Decimals()
 - Mathematical Functions, [235](#)
- Declarative Programming
 - Declarative programming on the top of Object-Oriented, [367](#)
 - executing code after the end of object access, [367](#)
 - Introduction, [363](#)
 - More Beatiful Code, [368](#)
 - Objects inside lists, [364](#)
 - Return object by reference, [365](#)
- Declarative programming on the top of Object-Oriented
 - Declarative Programming, [367](#)
- Declare parameters
 - Functions - First Style, [194](#)
 - Functions - Second Style, [198](#)
 - Functions - Third Style, [202](#)
- Decrypt()
 - Security and Internet Functions, [285](#)
- Deep Copy
 - Variables, [171](#)
- Define Declarative Languages
 - Language Design, [33](#)
- Define Functions
 - Functions - First Style, [193](#)
 - Functions - Second Style, [197](#)
 - Functions - Third Style, [201](#)
- Define Natural Statements
 - Language Design, [31](#)
- Defining Commands
 - Using the Natural Library, [382](#)
- Defining commands using classes
 - Using the Natural Library, [385](#)
- Defining Constants
 - Code Generator, [814](#)
- Defining Variables and Variables Access
 - Scope Rules, [712](#)
- Delete Item From List
 - Lists, [207](#)
- Demo Programs
 - Introduction, [256](#)
 - Language Shell, [257](#)
 - Main Menu, [257](#)
 - The Cards Game, [680](#)
- Designed for a clear goal
 - Language Design, [28](#)
- Desktop and Mobile Development
 - Adding Hyperlink to QLabel, [621](#)
 - Change Focus, [656](#)
 - Classes and their Methods to use the default events, [686](#)
 - Context Menu, [603](#)
 - Creating Menubar, [600](#)
 - Creating more than one window, [639](#)
 - Creating Reports using the WebLib and the GUILib, [691](#)
 - Creating StatusBar, [605](#)
 - Creating Toolbar, [604](#)

- Dialog Functions, [646](#)
- Display Image using QLabel, [625](#)
- Display Scaled Image using QLabel, [634](#)
- Drawing using QPainter, [636](#)
- Dynamic Objects, [660](#)
- Inheritance from GUI Classes, [652](#)
- KeyPress and Mouse Move Events, [646](#)
- Menubar and StyleSheet Example, [626](#)
- Methods to use Events with Events Filter, [689](#)
- Movable Label Example, [642](#)
- Moving Objects using the Mouse, [648](#)
- New Classes names - Index Start from 1, [691](#)
- Notepad Application, [665](#)
- Other Widgets Events, [630](#)
- Playing Sound, [640](#)
- Printing using QPrinter, [638](#)
- QLineEdit Events and QMessageBox, [628](#)
- QMessageBox Example, [643](#)
- QVideoWidget and QMediaPlayer, [622](#)
- Regular Expressions, [657](#)
- RingQt Classes and their Qt Documentation, [691](#)
- Rotate Text, [654](#)
- Simple Client and Server Example, [658](#)
- The Cards Game, [680](#)
- The Difference between Qt and RingQt, [690](#)
- The First GUI Application, [590](#)
- Using Layout, [591](#)
- Using QCheckBox, [618](#)
- Using QComboBox Class, [599](#)
- Using QDateEdit, [613](#)
- Using QDesktopWidget Class, [653](#)
- Using QDial, [614](#)
- Using QDockWidget, [606](#)
- Using QFrame, [623](#)
- Using QInputDialog Class, [644](#)
- Using qLCDNumber, [641](#)
- Using QProgressBar, [610](#)
- Using QProgressBar and Timer, [633](#)
- Using QRadioButton and QButtonGroup, [619](#)
- Using QSlider, [611](#)
- Using QSpinBox, [611](#)
- Using QTableWidget, [609](#)
- Using QTabWidget, [607](#)
- Using QTreeView and QFileSystemModel, [597](#)
- Using QTreeWidget and QTreeWidgetItem, [598](#)
- Using QWebView, [617](#)
- Using the QColorDialog Class, [640](#)
- Using the QFileDialog Class, [635](#)
- Using the QListWidget Class, [594](#)
- Using the QTextEdit Class, [593](#)
- Using the QTimer Class, [632](#)
- Weight History Application, [661](#)
- Determine Structure Members Types
 - Code Generator, [814](#)
- Dialog Functions
 - Desktop and Mobile Development, [646](#)
- DiffDays()
 - Date and Time, [222](#)
- Dir()
 - Files, [238](#)
- direxists()
 - Stdlib Functions, [336](#)
- Disable BreakPoints
 - The Trace Library and the Interactive Debugger, [792](#)
- Display Error Message
 - Extension, [801](#)
- Display Image
 - RingLibSDL, [462](#)
- Display Image using QLabel
 - Desktop and Mobile Development, [625](#)
- Display PNG Images
 - RingLibSDL, [463](#)
- Display Scaled Image using QLabel
 - Desktop and Mobile Development, [634](#)
- Display Transparent Images
 - RingLibSDL, [464](#)
- Display Warnings Option
 - What is new in Ring 1.2?, [121](#)
- Distributing Applications and Games for Mobile
 - Distributing Ring Application, [760](#)
- Distributing Applications for Microsoft Windows
 - Distributing Ring Application, [759](#)
- Distributing Applications for Mobile using RingQt
 - Distributing Ring Application using Ring2EXE, [764](#)
- Distributing Ring Application
 - Creating Windows Installer, [760](#)
 - Distributing Applications and Games for Mobile, [760](#)
 - Distributing Applications for Microsoft Windows, [759](#)
 - Introduction, [758](#)
 - Protecting the Source Code, [759](#)
 - Using C/C++ Compiler and Linker, [760](#)
- Distributing Ring Application using Ring2EXE
 - Building standalone console application, [762](#)
 - Building the Cards Game for Mobile using RingQt, [764](#)
 - Building the Form Designer for Mobile using RingQt, [766](#)
 - Building the Weight History Application for Mobile using RingQt, [765](#)
 - Creating the Qt resource file using Folder2qrc, [768](#)
 - Distributing Applications for Mobile using RingQt, [764](#)
 - Distributing RingAllegro Applications, [763](#)
 - Example, [761](#)
 - How Ring2EXE works?, [761](#)
 - Important Information about Ring2EXE, [768](#)

- Introduction, [760](#)
- Options, [762](#)
- Using Ring2EXE, [761](#)
- Distributing RingAllegro Applications
 - Distributing Ring Application using Ring2EXE, [763](#)
- Do Again Loop
 - Control Structures - First Style, [179](#)
- Documentation
 - How to contribute?, [141](#)
- Download Requirements
 - Building RingQt Applications for Mobile, [695](#)
- Download Requirements and Update the Android SDK
 - Building Games For Android, [519](#)
- Download()
 - Security and Internet Functions, [286](#)
- Draw Rectangle
 - RingLibSDL, [463](#)
- Drawing Text
 - Game Engine for 2D Games, [474](#)
- Drawing using QPainter
 - Desktop and Mobile Development, [636](#)
- Drawing using RingOpenGL
 - Using RingOpenGL and RingFreeGLUT for 3D Graphics, [524](#)
- Drawing, Animation and Input
 - Graphics and Game Programming, [448](#)
- Dynamic Attributes
 - Object Oriented Programming, [294](#)
- Dynamic Objects
 - Desktop and Mobile Development, [660](#)
- Dynamic Typing
 - Variables, [171](#)
- Editors Support
 - How to contribute?, [142](#)
- Embedding Ring in Ring
 - Embedding Ring in Ring, [794](#)
 - Embedding Ring in Ring without sharing the State, [795](#)
 - ring_state_setvar(), [796](#)
 - Serial Execution of Programs, [795](#)
- Embedding Ring in Ring without sharing the State
 - Embedding Ring in Ring, [795](#)
 - What is new in Ring 1.3?, [114](#)
- Embedding Ring Language in C/C++ Programs
 - Introduction, [807](#)
 - Ring State, [808](#)
 - Ring State Functions, [808](#)
 - Ring State Variables, [809](#)
- Employee Application
 - What is new in Ring 1.6?, [37](#)
- Encourage Organization
 - Language Design, [30](#)
- Encrypt()
 - Security and Internet Functions, [284](#)
- endswith()
 - Stdlib Functions, [330](#)
- Entering Items
 - Form Designer, [710](#)
- Enum and Numbers
 - Code Generator, [815](#)
- Environment Errors
 - Reference, [1813](#)
- EpochTime()
 - Date and Time, [222](#)
 - Stdlib Functions, [338](#)
- Equality of functions
 - Functional Programming, [303](#)
- Eval()
 - Eval() and Debugging, [254](#)
- Eval() and Debugging
 - Assert(), [256](#)
 - Eval(), [254](#)
 - Introduction, [253](#)
 - Raise(), [255](#)
 - Try/Catch/Done, [254](#)
- evenorodd()
 - Stdlib Functions, [332](#)
- Events Code
 - Form Designer, [706](#)
- Example
 - Distributing Ring Application using Ring2EXE, [761](#)
 - Files, [244](#)
 - Mathematical Functions, [232](#)
 - Natural Language Programming, [370](#)
 - Objects Library for RingQt Application, [699](#)
 - RingMurmurHash Functions Reference, [843](#)
 - Security and Internet Functions, [285](#)
 - System Functions, [249](#)
 - The Type Hints Library, [741](#)
- Example - The Trace Library
 - Low Level Functions, [786](#)
- Example - Using the Trace Functions
 - Low Level Functions, [782](#)
- Example about Sharing Names between Functions and Methods
 - Scope Rules for Functions and Methods, [727](#)
- Exceptions
 - Control Structures - Second Style, [185](#)
 - Control Structures - Third Style, [189](#)
- Execute Program Line by Line
 - The Trace Library and the Interactive Debugger, [791](#)
- Execute Query and Print Result
 - ODBC Functions, [265](#)
- executing code after the end of object access
 - Declarative Programming, [367](#)
- Executing code during code generation
 - Code Generator, [815](#)

- ExeFileName() Function
 - System Functions, 252
- ExeFolder() Function
 - System Functions, 252
- Exit
 - Control Structures - First Style, 179
- Exit from two loops
 - Control Structures - First Style, 179
- Exit/Loop inside sub functions
 - Control Structures - First Style, 180
- Extension
 - Check Parameters Count, 800
 - Check Parameters Type, 801
 - Display Error Message, 801
 - Fopen() and Fclose() Implementation, 802
 - Function Prototype, 802
 - Function Structure, 800
 - Get Parameters Values, 801
 - Introduction, 797
 - Module Organization, 799
 - MySQL_Columns() Implementation, 805
 - Return Value, 801
 - RING API - list Functions, 803
 - RING API - String Functions, 805
 - ring_ext.c, 798
 - ring_ext.h, 798
 - Shared Libraries, 806
 - Sin() Implementation, 802
- Extensions in C/C++
 - How to contribute?, 142
- Extract Zip File
 - RingZip, 445
- Facebook Login
 - RingLibCurl, 441
- factorial()
 - Stdlib Functions, 328
- factors()
 - Stdlib Functions, 332
- Fclose()
 - Files, 239
- Features
 - Introduction, 24
- Feof()
 - Files, 242
- Ferror()
 - Files, 242
- FetchStockData Application
 - Applications developed in little hours, 3
- Fexists()
 - Files, 244
- Fflush()
 - Files, 240
- Fgetc()
 - Files, 243
- Fgetpos()
 - Files, 242
- Fgets()
 - Files, 243
- fibonacci()
 - Stdlib Functions, 329
- Fifteen Puzzle Game 2
 - Applications developed in little hours, 4
- File Class
 - Stdlib Classes, 354
- File Hash
 - Security and Internet Functions, 285
- file2list()
 - Stdlib Functions, 330
- Files
 - Clearerr(), 242
 - Dir(), 238
 - Example, 244
 - Fclose(), 239
 - Feof(), 242
 - Ferror(), 242
 - Fexists(), 244
 - Fflush(), 240
 - Fgetc(), 243
 - Fgetpos(), 242
 - Fgets(), 243
 - Fopen(), 239
 - Fputc(), 243
 - Fputs(), 243
 - Fread(), 244
 - Freopen(), 240
 - Fseek(), 241
 - Fsetpos(), 242
 - Ftell(), 241
 - Fwrite(), 244
 - Introduction, 236
 - Numbers and Bytes, 246
 - Perror(), 242
 - Read File using Read(), 238
 - Remove(), 239
 - Rename(), 239
 - Rewind(), 241
 - Tempfile(), 241
 - Tempname(), 241
 - Ungetc(), 243
 - Write file using Write(), 238
- filter()
 - Stdlib Functions, 326
- Filtering using Expressions
 - Code Generator, 815
- Find SubString
 - Strings, 217
- Find() and List of Objects

- Object Oriented Programming, 295
- First-Class Functions
 - Functional Programming, 301
- First-Class Lists
 - Lists, 211
- Flappy Bird 3000 Game
 - Game Engine for 2D Games, 500
- Fopen()
 - Files, 239
- Fopen() and Fclose() Implementation
 - Extension, 802
- For In Loop
 - Control Structures - First Style, 178
 - Control Structures - Second Style, 185
 - Control Structures - Third Style, 189
- for in to modify lists
 - Control Structures - First Style, 178
- For Loop
 - Control Structures - First Style, 177
 - Control Structures - Second Style, 184
 - Control Structures - Third Style, 188
- Form Designer
 - Entering Items, 710
 - Events Code, 706
 - Introduction, 703
 - Keyboard Shortcuts, 709
 - Menubar Designer, 709
 - More Samples and Tests, 711
 - Running Forms, 706
 - The Designer Windows, 705
 - The Properties, 705
 - Using Layouts, 711
 - What is new in Ring 1.3?, 115
 - Window Flags, 710
- Fputc()
 - Files, 243
- Fputs()
 - Files, 243
- Frames Per Second
 - Using RingOpenGL and RingFreeGLUT for 3D Graphics, 557
- Fread()
 - Files, 244
- Freopen()
 - Files, 240
- Frequently Asked Questions
 - Can I connect to dbase/harbour database?, 1802
 - Can Ring work on Windows XP?, 1804
 - Conflict between Global Variables and Class Attributes, 1790
 - Constructor methods in Ring, 1786
 - Could you explain the output of the StrCmp() function?, 1793
 - Getter and Setter Methods, 1788
 - Goal of including the Main function in Ring, 1784
 - How can I disable maximize button and resize window?, 1800
 - How to add Combobox and other elements to the cells of a QTableWidgetItem?, 1807
 - How to Close a window then displaying another one?, 1799
 - How to create a Modal Window?, 1799
 - How to create an array of buttons in GUI applications?, 1798
 - How to extend RingQt and add more classes?, 1804
 - How to get the current source file path?, 1791
 - How to get the file size using ftell() and fseek() functions?, 1791
 - How to insert an item to the first position in the list?, 1796
 - How to perform some manipulations on selected cells in QTableWidgetItem?, 1807
 - How to print keys or values only in List/Dictionary?, 1792
 - How to print lists that contains objects?, 1796
 - How to print new lines and other characters?, 1797
 - How to use many source code files in the project?, 1793
 - How to use NULL and ISNULL() function?, 1795
 - How to use SQLite using ODBC?, 1801
 - Introduction, 1776
 - Is Ring some sort of improvement over PHP?, 1779
 - List index start from 1, 1785
 - Philosophy behind data types in Ring, 1782
 - Search of global names while defining the class attributes, 1789
 - The documentation says functional programming is supported, but then this happens?, 1781
 - What about predefined parameters or optional parameters in functions?, 1791
 - What about the Boolean values in Ring?, 1783
 - What are the advantages to using Ring over C# or Java?, 1781
 - What are the advantages to using Ring over Lisp or Smalltalk?, 1778
 - What are the advantages to using Ring over native C or C++?, 1779
 - What are the advantages to using Ring over Python and Ruby?, 1780
 - What are the advantages to using Ring over Tcl and Lua?, 1780
 - What happens when we create a new object?, 1787
 - What is the difference between Ring and Python? And is Ring Open Source?, 1779
 - Where can I write a program and execute it?, 1790
 - Which of 3 coding styles are commonly used or recommended by the community?, 1808
 - Why do we need Yet Another Programming Lan-

- guage (YAPL)?, [1777](#)
- Why I get a strange result when printing nl with lists?, [1792](#)
- Why I get Calling Function without definition Error?, [1803](#)
- Why Ring is largely focussed on UI creation?, [1778](#)
- Why Ring is not case-sensitive, [1785](#)
- Why Ring is weakly typed?, [1778](#)
- Why Ring uses 'See', 'Give', 'But' and 'Ok' Keywords?, [1782](#)
- Why setClickEvent() doesn't see the object methods directly?, [1803](#)
- Why the ability to define your own languages Instead of just handing over the syntax so you can parse it using whatever code you like?, [1781](#)
- Why the Assignment operator uses Deep copy?, [1786](#)
- Why the window title bar is going outside the screen?, [1797](#)
- Why this example use the GetChar() twice?, [1794](#)
- Why we don't use () after the qApp class name?, [1797](#)
- Why you can specify the number of loops you want to break out of?, [1782](#)
- Fseek()
 - Files, [241](#)
- Fsetpos()
 - Files, [242](#)
- FSize()
 - Stdlib Functions, [337](#)
- Ftell()
 - Files, [241](#)
- Function Prototype
 - Extension, [802](#)
- function prototype
 - Code Generator, [811](#)
- Function Structure
 - Extension, [800](#)
- Functional Programming
 - Anonymous and Nested Functions, [302](#)
 - Equality of functions, [303](#)
 - First-Class Functions, [301](#)
 - Higher-order Functions, [301](#)
 - Introduction, [299](#)
 - Pure Functions, [300](#)
- Functions - First Style
 - Call Functions, [193](#)
 - Declare parameters, [194](#)
 - Define Functions, [193](#)
 - Introduction, [192](#)
 - Main Function, [194](#)
 - Recursion, [196](#)
 - Return Value, [195](#)
 - Send Parameters, [194](#)
 - Variables Scope, [195](#)
- Functions - Second Style
 - Call Functions, [197](#)
 - Declare parameters, [198](#)
 - Define Functions, [197](#)
 - Introduction, [196](#)
 - Main Function, [198](#)
 - Recursion, [200](#)
 - Return Value, [199](#)
 - Send Parameters, [198](#)
 - Variables Scope, [199](#)
- Functions - Third Style
 - Call Functions, [201](#)
 - Declare parameters, [202](#)
 - Define Functions, [201](#)
 - Introduction, [200](#)
 - Main Function, [202](#)
 - Recursion, [204](#)
 - Return Value, [203](#)
 - Send Parameters, [202](#)
 - Variables Scope, [203](#)
- functions()
 - Reflection and Meta-programming, [307](#)
- Fwrite()
 - Files, [244](#)
- Game Class
 - Game Engine for 2D Games, [471](#)
- Game Engine Classes
 - Game Engine for 2D Games, [470](#)
- Game Engine for 2D Games
 - Animate Class, [473](#)
 - Animate Events, [484](#)
 - Animation, [478](#)
 - Animation and Functions, [479](#)
 - Creating the Game Window, [473](#)
 - Drawing Text, [474](#)
 - Flappy Bird 3000 Game, [500](#)
 - Game Class, [471](#)
 - Game Engine Classes, [470](#)
 - GameObject Class, [471](#)
 - Games Layer, [470](#)
 - Graphics Library Bindings, [469](#)
 - Interface to graphics library, [469](#)
 - Introduction, [468](#)
 - Map, [486](#)
 - Map Class, [473](#)
 - Map Events, [487](#)
 - Moving Text, [475](#)
 - Object and Drawing, [489](#)
 - Playing Sound, [477](#)
 - Project Layers, [469](#)
 - Sound Class, [473](#)
 - Sprite Automatic Movement, [480](#)

- Sprite Class, 472
- Sprite Keypress Event, 481
- Sprite Mouse Event, 482
- Sprite State Event, 483
- Stars Fighter Game, 492
- Super Man 2016 Game, 507
- Text Class, 472
- What is new in Ring 1.1?, 129
- GameObject Class
 - Game Engine for 2D Games, 471
- Games Layer
 - Game Engine for 2D Games, 470
- gcd()
 - Stdlib Functions, 331
- Generate Object File
 - Command Line Options, 758
- Generate/Execute Ring Object Files (*.ringo)
 - What is new in Ring 1.1?, 124
- Generating Pages using Objects
 - Web Development (CGI Library), 411
- Get Active Source File Name
 - System Functions, 251
- Get Command Line Arguments
 - System Functions, 250
- Get List Item
 - Lists, 207
- Get List Size
 - Lists, 207
- Get Number of Characters from position
 - Strings, 217
- Get Parameters Values
 - Extension, 801
- Get Request
 - RingLibCurl, 441
- Get Stock Data From Yahoo
 - RingLibCurl, 443
- Get String Length
 - Strings, 214
- Get Substring from position to end
 - Strings, 217
- getattribute()
 - Reflection and Meta-programming, 319
- GetChar()
 - Getting Input, 191
- getnumber()
 - Stdlib Functions, 324
- getstring()
 - Stdlib Functions, 324
- Getter and Setter Methods
 - Frequently Asked Questions, 1788
- Getting Input
 - GetChar(), 191
 - Getting Started - First Style, 144
 - Getting Started - Second Style, 147
 - Getting Started - Third Style, 149
 - Give Command, 191
 - Input(), 192
 - Introduction, 190
- Getting Started - First Style
 - Create Executable File, 143
 - Getting Input, 144
 - Hello World, 143
 - Introduction, 142
 - Multi-Line literals, 143
 - No Explicit End For Statements, 144
 - Not Case-Sensitive, 143
 - Run the program, 143
 - Using ? to print expression then new line, 144
 - Writing Comments, 145
- Getting Started - Second Style
 - Create Executable File, 146
 - Getting Input, 147
 - Hello World, 146
 - Introduction, 145
 - Multi-Line literals, 146
 - No Explicit End For Statements, 147
 - Not Case-Sensitive, 146
 - Run the program, 146
 - Writing Comments, 147
- Getting Started - Third Style
 - Create Executable File, 148
 - Getting Input, 149
 - Hello World, 148
 - Introduction, 147
 - Multi-Line literals, 149
 - No Explicit End For Statements, 149
 - Not Case-Sensitive, 148
 - Run the program, 148
 - Writing Comments, 149
- Give Command
 - Getting Input, 191
- globals()
 - Reflection and Meta-programming, 306
- Goal of including the Main function in Ring
 - Frequently Asked Questions, 1784
- Google API Shortener Application
 - Applications developed in little hours, 5
- Gradient
 - Web Development (CGI Library), 410
- Graphics and Game Programming
 - Drawing, Animation and Input, 448
 - Introduction, 447
 - Playing Sound, 456
 - Scaling and Rotating Images, 457
 - Threads, 459
 - Transparent Image, 458
 - TrueType Fonts, 455
- Graphics Library Bindings

- Game Engine for 2D Games, 469
- Hash Functions
 - Web Development (CGI Library), 405
- HashTable Class
 - Stdlib Classes, 347
- Hello World
 - Getting Started - First Style, 143
 - Getting Started - Second Style, 146
 - Getting Started - Third Style, 148
- Hello World Program using the Web Library
 - Web Development (CGI Library), 388
- Hex()
 - Data Type, 230
- Hex2str()
 - Data Type, 231
- Higher-order Functions
 - Functional Programming, 301
- History
 - Introduction, 23
 - Natural Language Programming, 370
- How can I disable maximize button and resize window?
 - Frequently Asked Questions, 1800
- How Ring find a functions and methods?
 - Scope Rules for Functions and Methods, 727
- How Ring find the Variable?
 - Scope Rules, 713
- How Ring2EXE works?
 - Distributing Ring Application using Ring2EXE, 761
- How to add Combobox and other elements to the cells of a QWidget?
 - Frequently Asked Questions, 1807
- How to Close a window then displaying another one?
 - Frequently Asked Questions, 1799
- How to contribute?
 - Applications, 142
 - Compiler and Virtual Machine (VM), 142
 - Documentation, 141
 - Editors Support, 142
 - Extensions in C/C++, 142
 - Ideas and suggestions, 142
 - Introduction, 140
 - Libraries in Ring, 142
 - Samples, 141
 - Special thanks to contributors, 141
 - Testing, 141
- How to create a Modal Window?
 - Frequently Asked Questions, 1799
- How to create an array of buttons in GUI applications?
 - Frequently Asked Questions, 1798
- How to extend RingQt and add more classes?
 - Frequently Asked Questions, 1804
- How to get the current source file path?
 - Frequently Asked Questions, 1791
- How to get the file size using ftell() and fseek() functions?
 - Frequently Asked Questions, 1791
- How to insert an item to the first position in the list?
 - Frequently Asked Questions, 1796
- How to perform some manipulations on selected cells in QWidget?
 - Frequently Asked Questions, 1807
- How to print keys or values only in List/Dictionary?
 - Frequently Asked Questions, 1792
- How to print lists that contains objects?
 - Frequently Asked Questions, 1796
- How to print new lines and other characters?
 - Frequently Asked Questions, 1797
- How to use many source code files in the project?
 - Frequently Asked Questions, 1793
- How to use NULL and ISNULL() function?
 - Frequently Asked Questions, 1795
- How to use SQLite using ODBC?
 - Frequently Asked Questions, 1801
- HTML Lists
 - Web Development (CGI Library), 408
- HTML Special Characters
 - Web Development (CGI Library), 404
- HTML Tables
 - Web Development (CGI Library), 409
- HtmlPage Class
 - Web Development (CGI Library), 416, 440
- HTTP Get Example
 - Web Development (CGI Library), 389
- HTTP POST Example
 - Web Development (CGI Library), 394
- Ideas and suggestions
 - How to contribute?, 142
- If Statement
 - Control Structures - Second Style, 183
 - Control Structures - Third Style, 186
- Important Information about Ring2EXE
 - Distributing Ring Application using Ring2EXE, 768
- Inheritance
 - Object Oriented Programming, 293
- Inheritance from GUI Classes
 - Desktop and Mobile Development, 652
- Innovative
 - Applications developed in little hours, 19
- Input()
 - Getting Input, 192
- Insert()
 - Lists, 210
- Install Qt for Android
 - Building RingQt Applications for Mobile, 695
- Interactive Debugger
 - The Trace Library and the Interactive Debugger, 791
- Interface to graphics library

- Game Engine for 2D Games, 469
- Internet Class
 - Stdlib Classes, 363
- Introduction
 - Features, 24
 - History, 23
 - Introduction, 21
 - Motivation, 22
 - Ring and other languages, 23
- Is Ring some sort of improvement over PHP?
 - Frequently Asked Questions, 1779
- IsAlNum()
 - Data Type, 226
- IsAlpha()
 - Data Type, 226
- IsAndroid() Function
 - System Functions, 249
- isattribute()
 - Reflection and Meta-programming, 315
- iscfunction()
 - Reflection and Meta-programming, 309
- isclass()
 - Reflection and Meta-programming, 311
- IsCntrl()
 - Data Type, 226
- IsDigit()
 - Data Type, 227
- IsFreeBSD() Function
 - System Functions, 249
- isfunction()
 - Reflection and Meta-programming, 308
- isglobal()
 - Reflection and Meta-programming, 308
- IsGraph()
 - Data Type, 227
- isleapyear()
 - Stdlib Functions, 333
- IsLinux() Function
 - System Functions, 249
- IsList()
 - Data Type, 225
- islocal()
 - Reflection and Meta-programming, 308
- IsLower()
 - Data Type, 227
- IsMacOSX() Function
 - System Functions, 248
- ismainsourcefile()
 - Stdlib Functions, 336
- ismethod()
 - Reflection and Meta-programming, 316
- IsMSDOS() Function
 - System Functions, 248
- IsNull()
 - Data Type, 225
- IsNumber()
 - Data Type, 224
- isobject()
 - Reflection and Meta-programming, 314
- ispackage()
 - Reflection and Meta-programming, 310
- ispackagesclass()
 - Reflection and Meta-programming, 312
- isprime()
 - Stdlib Functions, 329
- IsPrint()
 - Data Type, 227
- isprivateattribute()
 - Reflection and Meta-programming, 315
- isprivatemethod()
 - Reflection and Meta-programming, 316
- IsPunct()
 - Data Type, 228
- IsSpace()
 - Data Type, 228
- isspecial()
 - Stdlib Functions, 327
- IsString()
 - Data Type, 224
- IsUnix() Function
 - System Functions, 248
- IsUpper()
 - Data Type, 228
- isvowel()
 - Stdlib Functions, 328
- IsWindows() Function
 - System Functions, 248
- IsWindows64() Function
 - System Functions, 248
- IsXdigit()
 - Data Type, 228
- JustFileName()
 - Stdlib Functions, 325
- JustFilePath()
 - Stdlib Functions, 324
- Keyboard Events and Colors
 - Using RingOpenGL and RingFreeGLUT for 3D Graphics, 529
- Keyboard Shortcuts
 - Form Designer, 709
- KeyPress and Mouse Move Events
 - Desktop and Mobile Development, 646
- Language Design
 - Compact Syntax, 30
 - Define Declarative Languages, 33
 - Define Natural Statements, 31

- Designed for a clear goal, 28
- Encourage Organization, 30
- Introduction, 27
- Simple, 28
- Smart Garbage Collector, 36
- Transparent Implementation, 34
- Trying to be natural, 29
- Visual Implementation, 35
- Why Ring?, 28
- Language Functions
 - Reference, 1810
- Language Grammar
 - Reference, 1813
- Language Keywords
 - Reference, 1809
- Language Shell
 - Demo Programs, 257
- lcm()
 - Stdlib Functions, 331
- Left()
 - Strings, 215
- Libraries in Ring
 - How to contribute?, 142
- Library Usage
 - Objects Library for RingQt Application, 699
- linecount()
 - Stdlib Functions, 328
- Lines()
 - Strings, 216
- List Class
 - Stdlib Classes, 344
- List index start from 1
 - Frequently Asked Questions, 1785
- List of changes and new features
 - What is new in Ring 1.1?, 123
 - What is new in Ring 1.2?, 117
 - What is new in Ring 1.3?, 104
 - What is new in Ring 1.4?, 92
 - What is new in Ring 1.5?, 48
 - What is new in Ring 1.6?, 37
- list of functions
 - Mathematical Functions, 232
- list2file()
 - Stdlib Functions, 329
- ListAllFiles() Function
 - Stdlib Functions, 338
- Lists
 - Access List Items by String Index, 212
 - Add Items, 206
 - Copy Lists, 210
 - Create Lists, 206
 - Delete Item From List, 207
 - First-Class Lists, 211
 - Get List Item, 207
 - Get List Size, 207
 - Insert(), 210
 - Introduction, 205
 - Nested Lists, 210
 - Passing Lists to Functions, 211
 - Passing Parameters Using List, 212
 - Reverse(), 209
 - Search, 208
 - Set List Item, 207
 - Sort(), 208
 - Swap Items, 213
 - Using Lists during definition, 211
- Load Syntax Files
 - Syntax Flexibility, 732
- Loading the Trace library
 - The Trace Library and the Interactive Debugger, 790
- locals()
 - Reflection and Meta-programming, 306
- Logical Operators
 - Operators, 173
- Loop Command
 - Control Structures - First Style, 180
- Looping
 - Control Structures, 187
 - Control Structures - First Style, 177
 - Control Structures - Second Style, 184
- Low Level Functions
 - callgc(), 770
 - Example - The Trace Library, 786
 - Example - Using the Trace Functions, 782
 - Introduction, 769
 - nullpointer(), 772
 - object2pointer(), 772
 - pointer2object(), 772
 - ptrcmp(), 773
 - ringvm_callfunc(), 782
 - RingVM_CallList(), 778
 - RingVM_CFunctionsList(), 774
 - RingVM_ClassesList(), 774
 - ringvm_evalinscope(), 781
 - RingVM_FilesList(), 779
 - RingVM_FunctionsList(), 774
 - ringvm_hideerrorMsg(), 781
 - RingVM_MemoryList(), 776
 - RingVM_PackagesList(), 775
 - ringvm_passerror(), 781
 - ringvm_scopescount(), 781
 - ringvm_settrace(), 780
 - ringvm_tracedata(), 780
 - ringvm_traceevent(), 780
 - ringvm_tracefunc(), 781
 - space(), 771
 - varptr(), 771

- Main Function
 - Functions - First Style, [194](#)
 - Functions - Second Style, [198](#)
 - Functions - Third Style, [202](#)
- Main Menu
 - Demo Programs, [257](#)
- Make a Cube using RingOpenGL and RingFreeGLUT
 - Using RingOpenGL and RingFreeGLUT for 3D Graphics, [566](#)
- makedir()
 - Stdlib Functions, [337](#)
- Many Cubes
 - Using RingOpenGL and RingAllegro for 3D Graphics, [574](#)
- Map
 - Game Engine for 2D Games, [486](#)
- Map Class
 - Game Engine for 2D Games, [473](#)
- Map Events
 - Game Engine for 2D Games, [487](#)
- map()
 - Stdlib Functions, [326](#)
- Math Class
 - Stdlib Classes, [349](#)
- Mathematical Functions
 - Decimals(), [235](#)
 - Example, [232](#)
 - Introduction, [231](#)
 - list of functions, [232](#)
 - Random(), [234](#)
 - Unsigned(), [235](#)
 - Using _ in numbers, [236](#)
 - Using f after numbers, [236](#)
- matrixmulti()
 - Stdlib Functions, [333](#)
- matrixtrans()
 - Stdlib Functions, [334](#)
- MD5()
 - Security and Internet Functions, [282](#)
- Menu Events
 - Using RingOpenGL and RingFreeGLUT for 3D Graphics, [542](#)
- Menubar and StyleSheet Example
 - Desktop and Mobile Development, [626](#)
- Menubar Designer
 - Form Designer, [709](#)
- mergemethods()
 - Reflection and Meta-programming, [321](#)
- Methods to use Events with Events Filter
 - Desktop and Mobile Development, [689](#)
- methods()
 - Reflection and Meta-programming, [314](#)
- Misc Operators
 - Operators, [174](#)
- Module Organization
 - Extension, [799](#)
- More Beatiful Code
 - Declarative Programming, [368](#)
- More Samples and Tests
 - Form Designer, [711](#)
- More Syntax Flexibility
 - What is new in Ring 1.5?, [79](#)
- Motivation
 - Introduction, [22](#)
- Mouse Events
 - RingLibSDL, [466](#)
 - Using RingOpenGL and RingFreeGLUT for 3D Graphics, [538](#)
- Movable Label Example
 - Desktop and Mobile Development, [642](#)
- Moving Objects using the Mouse
 - Desktop and Mobile Development, [648](#)
- Moving Text
 - Game Engine for 2D Games, [475](#)
- Multi-Line literals
 - Getting Started - First Style, [143](#)
 - Getting Started - Second Style, [146](#)
 - Getting Started - Third Style, [149](#)
- MurmurHash1 functions
 - RingMurmurHash Functions Reference, [843](#)
- MurmurHash2 functions
 - RingMurmurHash Functions Reference, [843](#)
- MurmurHash3 functions
 - RingMurmurHash Functions Reference, [843](#)
- MySQL Class
 - Stdlib Classes, [360](#)
- MySQL Functions
 - Create Database, [271](#)
 - Create Table and Insert Data, [272](#)
 - Introduction, [269](#)
 - MySQL_AutoCommit(), [277](#)
 - MySQL_Close(), [271](#)
 - MySQL_Columns(), [274](#)
 - MySQL_Commit(), [277](#)
 - MySQL_Connect(), [271](#)
 - MySQL_Error(), [271](#)
 - MySQL_Escape_String(), [276](#)
 - MySQL_Info(), [270](#)
 - MySQL_Init(), [271](#)
 - MySQL_Inser_ID(), [273](#)
 - MySQL_Next_Result(), [274](#)
 - MySQL_Query(), [271](#)
 - MySQL_Result(), [274](#)
 - MySQL_Result2(), [275](#)
 - MySQL_Rollback(), [277](#)
 - Print Query Result, [274](#)
 - Restore Image From The Database, [276](#)
 - Save Image Inside the Database, [276](#)

- Transaction Example, [277](#)
- MySQL_AutoCommit()
 - MySQL Functions, [277](#)
- MySQL_Close()
 - MySQL Functions, [271](#)
- MySQL_Columns()
 - MySQL Functions, [274](#)
- MySQL_Columns() Implementation
 - Extension, [805](#)
- MySQL_Commit()
 - MySQL Functions, [277](#)
- MySQL_Connect()
 - MySQL Functions, [271](#)
- MySQL_Error()
 - MySQL Functions, [271](#)
- MySQL_Escape_String()
 - MySQL Functions, [276](#)
- MySQL_Info()
 - MySQL Functions, [270](#)
- MySQL_Init()
 - MySQL Functions, [271](#)
- MySQL_Inser_ID()
 - MySQL Functions, [273](#)
- MySQL_Next_Result()
 - MySQL Functions, [274](#)
- MySQL_Query()
 - MySQL Functions, [271](#)
- MySQL_Result()
 - MySQL Functions, [274](#)
- MySQL_Result2()
 - MySQL Functions, [275](#)
- MySQL_Rollback()
 - MySQL Functions, [277](#)
- Natural Language Programming
 - BraceError() Method, [377](#)
 - BraceExprEval Method, [376](#)
 - BraceStart and BraceEnd Methods, [375](#)
 - Change the '=' operator to 'is', [373](#)
 - Change the Ring Keyword 'And', [371](#)
 - Change the Ring Operator '+', [372](#)
 - Clean Natural Code, [378](#)
 - Example, [370](#)
 - History, [370](#)
 - Introduction, [369](#)
 - Real Natual Code, [376](#)
 - Using Eval() with our Natural Code, [374](#)
- Natural Library - Demo Program
 - Using the Natural Library, [380](#)
- Nested Lists
 - Lists, [210](#)
- New Classes names - Index Start from 1
 - Desktop and Mobile Development, [691](#)
- New Functions
 - What is new in Ring 1.2?, [117](#)
 - What is new in Ring 1.3?, [113](#)
- New Functions and Changes
 - What is new in Ring 1.1?, [126](#)
- New Style to Ring Notepad
 - What is new in Ring 1.4?, [95](#)
- New Tool: Folder2qrc
 - What is new in Ring 1.6?, [39](#)
- New Tool: Ring2EXE
 - What is new in Ring 1.6?, [38](#)
- newlist()
 - Stdlib Functions, [327](#)
- No Explicit End For Statements
 - Getting Started - First Style, [144](#)
 - Getting Started - Second Style, [147](#)
 - Getting Started - Third Style, [149](#)
- No Run
 - Command Line Options, [757](#)
- Not Case-Sensitive
 - Getting Started - First Style, [143](#)
 - Getting Started - Second Style, [146](#)
 - Getting Started - Third Style, [148](#)
- Notepad Application
 - Desktop and Mobile Development, [665](#)
- nullpointer()
 - Low Level Functions, [772](#)
- Number()
 - Data Type, [229](#)
- Numbers and Bytes
 - Files, [246](#)
- Object and Drawing
 - Game Engine for 2D Games, [489](#)
- Object Library Source Code
 - Objects Library for RingQt Application, [703](#)
- Object Oriented Programming
 - Access Objects Using Braces, [289](#)
 - Classes and Objects, [287](#)
 - Composition, [290](#)
 - Dynamic Attributes, [294](#)
 - Find() and List of Objects, [295](#)
 - Inheritance, [293](#)
 - Introduction, [286](#)
 - Operator Overloading, [292](#)
 - Packages, [294](#)
 - Printing Objects, [295](#)
 - Private Attributes and Methods, [291](#)
 - Setter and Getter, [291](#)
 - Sort() and List of Objects, [296](#)
 - Using Self.Attribute, [298](#)
- object2pointer()
 - Low Level Functions, [772](#)
- objectid()
 - Reflection and Meta-programming, [313](#)

- Objects inside lists
 - Declarative Programming, 364
- Objects Library for RingQt
 - What is new in Ring 1.2?, 119
- Objects Library for RingQt Application
 - Example, 699
 - Introduction, 698
 - Library Usage, 699
 - Object Library Source Code, 703
 - Open_WindowAndLink() Function, 702, 703
 - Open_WindowInPackages() Function, 703
- ODBC Class
 - Stdlib Classes, 359
- ODBC Functions
 - Execute Query and Print Result, 265
 - Introduction, 260
 - odbc_autocommit(), 267
 - odbc_close(), 262
 - odbc_colcount(), 264
 - odbc_columns(), 266
 - odbc_commit(), 267
 - odbc_connect(), 263
 - odbc_datasources(), 262
 - odbc_disconnect(), 264
 - odbc_drivers(), 261
 - odbc_execute(), 264
 - odbc_fetch(), 265
 - odbc_getdata(), 265
 - odbc_init(), 261
 - odbc_rollback(), 267
 - odbc_tables(), 265
 - Open and Close Connection, 264
 - Print List of ODBC Data Sources, 263
 - Print List of ODBC Drivers, 262
 - Save and Restore Images, 268
 - Transactions and Using Commit and Rollback, 267
- odbc_autocommit()
 - ODBC Functions, 267
- odbc_close()
 - ODBC Functions, 262
- odbc_colcount()
 - ODBC Functions, 264
- odbc_columns()
 - ODBC Functions, 266
- odbc_commit()
 - ODBC Functions, 267
- odbc_connect()
 - ODBC Functions, 263
- odbc_datasources()
 - ODBC Functions, 262
- odbc_disconnect()
 - ODBC Functions, 264
- odbc_drivers()
 - ODBC Functions, 261
- odbc_execute()
 - ODBC Functions, 264
- odbc_fetch()
 - ODBC Functions, 265
- odbc_getdata()
 - ODBC Functions, 265
- odbc_init()
 - ODBC Functions, 261
- odbc_rollback()
 - ODBC Functions, 267
- odbc_tables()
 - ODBC Functions, 265
- Open and Close Connection
 - ODBC Functions, 264
- Open_WindowAndLink() Function
 - Objects Library for RingQt Application, 702, 703
- Open_WindowInPackages() Function
 - Objects Library for RingQt Application, 703
- Operator Overloading
 - Object Oriented Programming, 292
- Operators
 - Arithmetic Operators, 173
 - Assignment Operators, 174
 - Bitwise Operators, 174
 - Introduction, 172
 - Logical Operators, 173
 - Misc Operators, 174
 - Operators Precedence, 174
 - Relational Operators, 173
 - Using the Natural Library, 384
- Operators Precedence
 - Operators, 174
- Options
 - Distributing Ring Application using Ring2EXE, 762
- OSCopyFile() Function
 - Stdlib Functions, 339
- OSCopyFolder() Function
 - Stdlib Functions, 339
- OSCreateOpenFolder() Function
 - Stdlib Functions, 339
- OSDeleteFile() Function
 - Stdlib Functions, 340
- OSDeleteFolder() Function
 - Stdlib Functions, 339
- OSRenameFile() Function
 - Stdlib Functions, 340
- Other Widgets Events
 - Desktop and Mobile Development, 630
- packagename()
 - Reflection and Meta-programming, 322
- Packages
 - Object Oriented Programming, 294
- packages()

- Reflection and Meta-programming, 309
- packagesclasses()
 - Reflection and Meta-programming, 312
- Page Class
 - Web Development (CGI Library), 436
- palindrome()
 - Stdlib Functions, 333
- Pass Error
 - The Trace Library and the Interactive Debugger, 791
- Passing Lists to Functions
 - Lists, 211
- Passing Parameters Using List
 - Lists, 212
- Performance
 - Command Line Options, 757
- permutation()
 - Stdlib Functions, 334
- Perror()
 - Files, 242
- Philosophy behind data types in Ring
 - Frequently Asked Questions, 1782
- Play Sound
 - RingLibSDL, 467
- Playing Sound
 - Desktop and Mobile Development, 640
 - Game Engine for 2D Games, 477
 - Graphics and Game Programming, 456
- pointer2object()
 - Low Level Functions, 772
- Post Request
 - RingLibCurl, 441
- Practical
 - Applications developed in little hours, 20
- Prefix for Functions Names
 - Code Generator, 813
- PrevFileName() Function
 - System Functions, 251
- Print files in Zip file
 - RingZip, 445
- Print List of ODBC Data Sources
 - ODBC Functions, 263
- Print List of ODBC Drivers
 - ODBC Functions, 262
- Print Query Result
 - MySQL Functions, 274
- print()
 - Stdlib Functions, 323
- Print2Str() Function
 - Stdlib Functions, 323
- Printing Final Intermediate Code
 - Command Line Options, 752
- Printing Instruction Operation Code
 - Command Line Options, 757
- Printing Intermediate Code
 - Command Line Options, 750
- Printing Objects
 - Object Oriented Programming, 295
- Printing Rules
 - Command Line Options, 746
- Printing Tokens
 - Command Line Options, 744
- Printing using QPrinter
 - Desktop and Mobile Development, 638
- Private Attributes and Methods
 - Object Oriented Programming, 291
- prodlist()
 - Stdlib Functions, 332
- Program Structure
 - Introduction, 204
 - Source Code File Sections, 205
 - Using Many Source Code Files, 205
- Project Folder
 - Building Games For Android, 519
- Project Layers
 - Game Engine for 2D Games, 469
- Protecting the Source Code
 - Distributing Ring Application, 759
- ptrcmp()
 - Low Level Functions, 773
- Pure Functions
 - Functional Programming, 300
- puts()
 - Stdlib Functions, 323
- QAbstractButton Class
 - RingQt Classes Reference, 1670
- QAbstractItemView Class
 - RingQt Classes Reference, 1656
- QAbstractScrollArea Class
 - RingQt Classes Reference, 1655
- QAbstractSlider Class
 - RingQt Classes Reference, 1663
- QAbstractSocket Class
 - RingQt Classes Reference, 1691
- QAbstractSpinBox Class
 - RingQt Classes Reference, 1666
- QAction Class
 - RingQt Classes Reference, 1674
- QAllEvents Class
 - RingQt Classes Reference, 1704
- QApp Class
 - RingQt Classes Reference, 1621
- QAxBase Class
 - RingQt Classes Reference, 1758
- QAxObject Class
 - RingQt Classes Reference, 1758
- QBitmap Class
 - RingQt Classes Reference, 1629

- QBluetoothAddress Class
 - RingQt Classes Reference, 1764
- QBluetoothDeviceDiscoveryAgent Class
 - RingQt Classes Reference, 1764
- QBluetoothDeviceInfo Class
 - RingQt Classes Reference, 1764
- QBluetoothHostInfo Class
 - RingQt Classes Reference, 1765
- QBluetoothLocalDevice Class
 - RingQt Classes Reference, 1765
- QBoxLayout Class
 - RingQt Classes Reference, 1748
- QBrush Class
 - RingQt Classes Reference, 1688
- QBuffer Class
 - RingQt Classes Reference, 1763
- QButtonGroup Class
 - RingQt Classes Reference, 1671
- QByteArray Class
 - RingQt Classes Reference, 1688
- QCamera Class
 - RingQt Classes Reference, 1743
- QCameraImageCapture Class
 - RingQt Classes Reference, 1744
- QCameraViewfinder Class
 - RingQt Classes Reference, 1742
- QCheckBox Class
 - RingQt Classes Reference, 1669
- QColor Class
 - RingQt Classes Reference, 1682
- QColorDialog Class
 - RingQt Classes Reference, 1701
- QComboBox Class
 - RingQt Classes Reference, 1645
- QCompleter Class
 - RingQt Classes Reference, 1752
- QCompleter2 Class
 - RingQt Classes Reference, 1753
- QCompleter3 Class
 - RingQt Classes Reference, 1753
- QCoreApplication Class
 - RingQt Classes Reference, 1773
- QCursor Class
 - RingQt Classes Reference, 1757
- QDate Class
 - RingQt Classes Reference, 1715
- QDateEdit Class
 - RingQt Classes Reference, 1664
- QDateTime Class
 - RingQt Classes Reference, 1765
- QDateTimeEdit Class
 - RingQt Classes Reference, 1664
- QDesktopServices Class
 - RingQt Classes Reference, 1621
- QDesktopWidget Class
 - RingQt Classes Reference, 1708
- QDial Class
 - RingQt Classes Reference, 1667
- QDialog Class
 - RingQt Classes Reference, 1698
- QDir Class
 - RingQt Classes Reference, 1640
- QDirModel Class
 - RingQt Classes Reference, 1697
- QDockWidget Class
 - RingQt Classes Reference, 1651
- QEvent Class
 - RingQt Classes Reference, 1675
- QFileDialog Class
 - RingQt Classes Reference, 1677
- QFileInfo Class
 - RingQt Classes Reference, 1696
- QFileSystemModel Class
 - RingQt Classes Reference, 1640
- QFont Class
 - RingQt Classes Reference, 1686
- QFontDialog Class
 - RingQt Classes Reference, 1698
- QFontMetrics Class
 - RingQt Classes Reference, 1747
- QFrame Class
 - RingQt Classes Reference, 1654
- QFrame2 Class
 - RingQt Classes Reference, 1655
- QFrame3 Class
 - RingQt Classes Reference, 1655
- QGradient Class
 - RingQt Classes Reference, 1750
- QGraphicsVideoItem Class
 - RingQt Classes Reference, 1742
- QGridLayout Class
 - RingQt Classes Reference, 1740
- QGuiApplication Class
 - RingQt Classes Reference, 1771
- QHBoxLayout Class
 - RingQt Classes Reference, 1634
- QHeaderView Class
 - RingQt Classes Reference, 1744
- QHostAddress Class
 - RingQt Classes Reference, 1695
- QHostInfo Class
 - RingQt Classes Reference, 1695
- QIcon Class
 - RingQt Classes Reference, 1631
- QImage Class
 - RingQt Classes Reference, 1725
- QInputDialog Class
 - RingQt Classes Reference, 1703

- QIODevice Class
 - RingQt Classes Reference, 1690
- QJsonArray Class
 - RingQt Classes Reference, 1734
- QJsonDocument Class
 - RingQt Classes Reference, 1735
- QJsonObject Class
 - RingQt Classes Reference, 1736
- QJsonParseError Class
 - RingQt Classes Reference, 1736
- QJsonValue Class
 - RingQt Classes Reference, 1736
- QKeySequence Class
 - RingQt Classes Reference, 1702
- QLabel Class
 - RingQt Classes Reference, 1628
- QLayout Class
 - RingQt Classes Reference, 1749
- QLCDNumber Class
 - RingQt Classes Reference, 1702
- QLinearGradient Class
 - RingQt Classes Reference, 1750
- QLineEdit Class
 - RingQt Classes Reference, 1631
- QLineEdit Events and QMessageBox
 - Desktop and Mobile Development, 628
- QListView Class
 - RingQt Classes Reference, 1757
- QListWidget Class
 - RingQt Classes Reference, 1636
- QListWidgetItem Class
 - RingQt Classes Reference, 1714
- QMainWindow Class
 - RingQt Classes Reference, 1649
- QMdiArea Class
 - RingQt Classes Reference, 1755
- QMdiSubWindow Class
 - RingQt Classes Reference, 1756
- QMediaObject Class
 - RingQt Classes Reference, 1744
- QMediaPlayer Class
 - RingQt Classes Reference, 1672
- QMediaPlaylist Class
 - RingQt Classes Reference, 1672
- QMenu Class
 - RingQt Classes Reference, 1647
- QMenuBar Class
 - RingQt Classes Reference, 1647
- QMessageBox Class
 - RingQt Classes Reference, 1676
- QMessageBox Example
 - Desktop and Mobile Development, 643
- QMutex Class
 - RingQt Classes Reference, 1763
- QMutexLocker Class
 - RingQt Classes Reference, 1763
- QNetworkAccessManager Class
 - RingQt Classes Reference, 1723
- QNetworkProxy Class
 - RingQt Classes Reference, 1693
- QNetworkReply Class
 - RingQt Classes Reference, 1724
- QNetworkRequest Class
 - RingQt Classes Reference, 1722
- QObject Class
 - RingQt Classes Reference, 1621
- QPainter Class
 - RingQt Classes Reference, 1679
- QPainter2 Class
 - RingQt Classes Reference, 1681
- QPainterPath Class
 - RingQt Classes Reference, 1724
- QPen Class
 - RingQt Classes Reference, 1682
- QPicture Class
 - RingQt Classes Reference, 1681
- QPixmap Class
 - RingQt Classes Reference, 1629
- QPixmap2 Class
 - RingQt Classes Reference, 1631
- QPlainTextEdit Class
 - RingQt Classes Reference, 1737
- QPoint Class
 - RingQt Classes Reference, 1751
- QPointF Class
 - RingQt Classes Reference, 1750
- QPrinter Class
 - RingQt Classes Reference, 1685
- QProcess Class
 - RingQt Classes Reference, 1754
- QProgressBar Class
 - RingQt Classes Reference, 1661
- QPushButton Class
 - RingQt Classes Reference, 1629
- QRadioButton Class
 - RingQt Classes Reference, 1671
- QRect Class
 - RingQt Classes Reference, 1708
- QRegion Class
 - RingQt Classes Reference, 1775
- QRegularExpression Class
 - RingQt Classes Reference, 1733
- QRegularExpressionMatch Class
 - RingQt Classes Reference, 1733
- QRegularExpressionMatchIterator Class
 - RingQt Classes Reference, 1734
- QScreen Class
 - RingQt Classes Reference, 1766

- QScrollArea Class
 - RingQt Classes Reference, 1751
- QSerialPort Class
 - RingQt Classes Reference, 1759
- QSerialPortInfo Class
 - RingQt Classes Reference, 1761
- QSize Class
 - RingQt Classes Reference, 1631
- QSlider Class
 - RingQt Classes Reference, 1664
- QSpinBox Class
 - RingQt Classes Reference, 1662
- QSplashScreen Class
 - RingQt Classes Reference, 1748
- QSplitter Class
 - RingQt Classes Reference, 1751
- QSqlDatabase Class
 - RingQt Classes Reference, 1716
- QSqlDriver Class
 - RingQt Classes Reference, 1717
- QSqlDriverCreatorBase Class
 - RingQt Classes Reference, 1721
- QSqlError Class
 - RingQt Classes Reference, 1719
- QSqlField Class
 - RingQt Classes Reference, 1720
- QSqlIndex Class
 - RingQt Classes Reference, 1719
- QSqlQuery Class
 - RingQt Classes Reference, 1718
- QSqlRecord Class
 - RingQt Classes Reference, 1719
- QStatusBar Class
 - RingQt Classes Reference, 1651
- QString2 Class
 - RingQt Classes Reference, 1754
- QStringList Class
 - RingQt Classes Reference, 1701
- QStringRef Class
 - RingQt Classes Reference, 1761
- QSystemTrayIcon Class
 - RingQt Classes Reference, 1715
- Qt Class Convertor
 - What is new in Ring 1.4?, 99
- Qt configuration file
 - Code Generator, 820
- QTableView Class
 - RingQt Classes Reference, 1658
- QTableWidget Class
 - RingQt Classes Reference, 1659
- QTableWidgetItem Class
 - RingQt Classes Reference, 1653
- QTabWidget Class
 - RingQt Classes Reference, 1652
- QTcpServer Class
 - RingQt Classes Reference, 1694
- QTcpSocket Class
 - RingQt Classes Reference, 1693
- QTest Class
 - RingQt Classes Reference, 1621
- QTextBlock Class
 - RingQt Classes Reference, 1712
- QTextBrowser Class
 - RingQt Classes Reference, 1774
- QTextCharFormat Class
 - RingQt Classes Reference, 1740
- QTextCodec Class
 - RingQt Classes Reference, 1716
- QTextCursor Class
 - RingQt Classes Reference, 1699
- QTextDocument Class
 - RingQt Classes Reference, 1710
- QTextEdit Class
 - RingQt Classes Reference, 1634
- QThread Class
 - RingQt Classes Reference, 1731
- QThreadPool Class
 - RingQt Classes Reference, 1732
- QTime Class
 - RingQt Classes Reference, 1713
- QTimer Class
 - RingQt Classes Reference, 1677
- QToolBar Class
 - RingQt Classes Reference, 1648
- QToolButton Class
 - RingQt Classes Reference, 1759
- QTreeView Class
 - RingQt Classes Reference, 1638
- QTreeWidget Class
 - RingQt Classes Reference, 1641
- QTreeWidgetItem Class
 - RingQt Classes Reference, 1644
- Queue Class
 - Stdlib Classes, 346
- Quotes about Ring
 - Applications developed in little hours, 1
- QUrl Class
 - RingQt Classes Reference, 1668
- QUuid Class
 - RingQt Classes Reference, 1759
- QVariant Class
 - RingQt Classes Reference, 1721
- QVBoxLayout Class
 - RingQt Classes Reference, 1633
- QVideoWidget and QMediaPlayer
 - Desktop and Mobile Development, 622
- QVideoWidget Class
 - RingQt Classes Reference, 1673

- QVideoWidgetControl Class
 - RingQt Classes Reference, 1743
- QWebView Class
 - RingQt Classes Reference, 1667
- QWidget Class
 - RingQt Classes Reference, 1622
- QWindow Class
 - RingQt Classes Reference, 1767
- QXmlStreamAttribute Class
 - RingQt Classes Reference, 1731
- QXmlStreamAttributes Class
 - RingQt Classes Reference, 1731
- QXmlStreamEntityDeclaration Class
 - RingQt Classes Reference, 1730
- QXmlStreamEntityResolver Class
 - RingQt Classes Reference, 1730
- QXmlStreamNamespaceDeclaration Class
 - RingQt Classes Reference, 1730
- QXmlStreamNotationDeclaration Class
 - RingQt Classes Reference, 1730
- QXmlStreamReader Class
 - RingQt Classes Reference, 1727
- QXmlStreamWriter Class
 - RingQt Classes Reference, 1729
- Raise()
 - Eval() and Debugging, 255
- RandBytes()
 - Security and Internet Functions, 285
- Random Image
 - Web Development (CGI Library), 407
- Random()
 - Mathematical Functions, 234
- Read File using Read()
 - Files, 238
- readline()
 - Stdlib Functions, 335
- Real Natual Code
 - Natural Language Programming, 376
- Recursion
 - Functions - First Style, 196
 - Functions - Second Style, 200
 - Functions - Third Style, 204
- Reference
 - Compiler Errors, 1811
 - Environment Errors, 1813
 - Introduction, 1808
 - Language Functions, 1810
 - Language Grammar, 1813
 - Language Keywords, 1809
 - Runtime Errors, 1812
 - Virtual Machine Instructions, 1815
- Reflection and Meta-programming
 - addattribute(), 317
 - addmethod(), 317
 - attributes(), 314
 - cfunctions(), 307
 - classes(), 311
 - classname(), 313
 - functions(), 307
 - getattribute(), 319
 - globals(), 306
 - Introduction, 304
 - isattribute(), 315
 - isfunction(), 309
 - isclass(), 311
 - isfunction(), 308
 - isglobal(), 308
 - islocal(), 308
 - ismethod(), 316
 - isobject(), 314
 - ispackage(), 310
 - ispackagesclass(), 312
 - isprivateattribute(), 315
 - isprivatemethod(), 316
 - locals(), 306
 - mergemethods(), 321
 - methods(), 314
 - objectid(), 313
 - packagename(), 322
 - packages(), 309
 - packagesclasses(), 312
 - setattribute(), 320
- Register New Functions
 - Code Generator, 814
- Regular Expressions
 - Desktop and Mobile Development, 657
- Relational Operators
 - Operators, 173
- Remove()
 - Files, 239
- Rename()
 - Files, 239
- Resources
 - Contact the Autor, 1822
 - Introduction, 1821
 - Ring Group, 1822
 - Ring Language Website, 1822
- Restore Image From The Database
 - MySQL Functions, 276
- Return object by reference
 - Declarative Programming, 365
- Return Self by Reference
 - What is new in Ring 1.3?, 113
- Return Value
 - Extension, 801
 - Functions - First Style, 195
 - Functions - Second Style, 199

- Functions - Third Style, [203](#)
- Reverse()
 - Lists, [209](#)
- Rewind()
 - Files, [241](#)
- Right()
 - Strings, [215](#)
- Ring and other languages
 - Introduction, [23](#)
- RING API - list Functions
 - Extension, [803](#)
- RING API - String Functions
 - Extension, [805](#)
- Ring CGI Hello World Program
 - Web Development (CGI Library), [388](#)
- Ring Group
 - Resources, [1822](#)
- Ring Language Website
 - Resources, [1822](#)
- Ring mode for Emacs Editor
 - What is new in Ring 1.3?, [111](#)
- Ring Notepad
 - Introduction, [150](#)
 - Ring Notepad - Creating and running your first console application, [151](#)
 - Ring Notepad - Creating and running your first game, [157](#)
 - Ring Notepad - Creating and running your first GUI application, [154](#)
 - Ring Notepad - Creating and running your first Web application, [155](#)
 - Ring Notepad - Main Window, [151](#)
 - The Browser Menu, [162](#)
 - The Distribute Menu, [163](#)
 - The Edit Menu, [159](#)
 - The File Menu, [159](#)
 - The Help Menu, [163](#)
 - The Main File in the Project, [158](#)
 - The Program Menu, [162](#)
 - The Tools Menu, [163](#)
 - The View Menu, [160](#)
- Ring Notepad - Creating and running your first console application
 - Ring Notepad, [151](#)
- Ring Notepad - Creating and running your first game
 - Ring Notepad, [157](#)
- Ring Notepad - Creating and running your first GUI application
 - Ring Notepad, [154](#)
- Ring Notepad - Creating and running your first Web application
 - Ring Notepad, [155](#)
- Ring Notepad - Main Window
 - Ring Notepad, [151](#)
- Ring State
 - Embedding Ring Language in C/C++ Programs, [808](#)
- Ring State Functions
 - Embedding Ring Language in C/C++ Programs, [808](#)
- Ring State Variables
 - Embedding Ring Language in C/C++ Programs, [809](#)
- Ring VM - Tracing Functions
 - What is new in Ring 1.5?, [73](#)
- ring_ext.c
 - Extension, [798](#)
- ring_ext.h
 - Extension, [798](#)
- ring_state_setvar()
 - Embedding Ring in Ring, [796](#)
- RingAllegro Functions Reference
 - Introduction, [844](#)
- RingCodeHighlighter Class
 - RingQt Classes Reference, [1727](#)
- RingConsoleColors Extension
 - What is new in Ring 1.6?, [40](#)
- RingConsoleColors Functions Reference
 - Introduction, [840](#)
- RingFreeGLUT Extension
 - What is new in Ring 1.5?, [59](#)
- RingFreeGLUT Functions Reference
 - Introduction, [879](#)
- RingLibCurl
 - Facebook Login, [441](#)
 - Get Request, [441](#)
 - Get Stock Data From Yahoo, [443](#)
 - Introduction, [440](#)
 - Post Request, [441](#)
 - Save output to string, [442](#)
 - What is new in Ring 1.2?, [120](#)
- RingLibCurl Functions Reference
 - Introduction, [830](#)
- RingLibSDL
 - Close Window Event, [465](#)
 - Create Window, [462](#)
 - Display Image, [462](#)
 - Display PNG Images, [463](#)
 - Display Transparent Images, [464](#)
 - Draw Rectangle, [463](#)
 - Introduction, [461](#)
 - Mouse Events, [466](#)
 - Play Sound, [467](#)
 - Switch Between Two Images, [462](#)
 - Use TTF Fonts, [464](#)
 - What is new in Ring 1.1?, [129](#)
- RingLibSDL Functions Reference
 - Introduction, [863](#)
- RingLibZip Functions Reference
 - Introduction, [839](#)
- RingMurmurHash Extension

- What is new in Ring 1.6?, [40](#)
- RingMurmurHash Functions Reference
 - Example, [843](#)
 - Introduction, [842](#)
 - MurmurHash1 functions, [843](#)
 - MurmurHash2 functions, [843](#)
 - MurmurHash3 functions, [843](#)
- RingOpenGL (OpenGL 1.1) Functions Reference
 - Introduction, [888](#)
- RingOpenGL (OpenGL 1.2) Functions Reference
 - Introduction, [913](#)
- RingOpenGL (OpenGL 1.3) Functions Reference
 - Introduction, [939](#)
- RingOpenGL (OpenGL 1.4) Functions Reference
 - Introduction, [968](#)
- RingOpenGL (OpenGL 1.5) Functions Reference
 - Introduction, [998](#)
- RingOpenGL (OpenGL 2.0) Functions Reference
 - Introduction, [1029](#)
- RingOpenGL (OpenGL 2.1) Functions Reference
 - Introduction, [1063](#)
- RingOpenGL (OpenGL 3.0) Functions Reference
 - Introduction, [1105](#)
- RingOpenGL (OpenGL 3.1) Functions Reference
 - Introduction, [1150](#)
- RingOpenGL (OpenGL 3.2) Functions Reference
 - Introduction, [1196](#)
- RingOpenGL (OpenGL 3.3) Functions Reference
 - Introduction, [1243](#)
- RingOpenGL (OpenGL 4.0) Functions Reference
 - Introduction, [1290](#)
- RingOpenGL (OpenGL 4.1) Functions Reference
 - Introduction, [1337](#)
- RingOpenGL (OpenGL 4.2) Functions Reference
 - Introduction, [1384](#)
- RingOpenGL (OpenGL 4.3) Functions Reference
 - Introduction, [1431](#)
- RingOpenGL (OpenGL 4.4) Functions Reference
 - Introduction, [1478](#)
- RingOpenGL (OpenGL 4.5) Functions Reference
 - Introduction, [1525](#)
- RingOpenGL (OpenGL 4.6) Functions Reference
 - Introduction, [1572](#)
- RingOpenGL Extension
 - What is new in Ring 1.5?, [68](#)
- RingQt Classes and their Qt Documentation
 - Desktop and Mobile Development, [691](#)
- RingQt Classes Reference
 - CodeEditor Class, [1739](#)
 - Introduction, [1620](#)
 - QAbstractButton Class, [1670](#)
 - QAbstractItemView Class, [1656](#)
 - QAbstractScrollArea Class, [1655](#)
 - QAbstractSlider Class, [1663](#)
 - QAbstractSocket Class, [1691](#)
 - QAbstractSpinBox Class, [1666](#)
 - QAction Class, [1674](#)
 - QAllEvents Class, [1704](#)
 - QApp Class, [1621](#)
 - QAxBase Class, [1758](#)
 - QAxObject Class, [1758](#)
 - QBitmap Class, [1629](#)
 - QBluetoothAddress Class, [1764](#)
 - QBluetoothDeviceDiscoveryAgent Class, [1764](#)
 - QBluetoothDeviceInfo Class, [1764](#)
 - QBluetoothHostInfo Class, [1765](#)
 - QBluetoothLocalDevice Class, [1765](#)
 - QBoxLayout Class, [1748](#)
 - QBrush Class, [1688](#)
 - QBuffer Class, [1763](#)
 - QButtonGroup Class, [1671](#)
 - QByteArray Class, [1688](#)
 - QCamera Class, [1743](#)
 - QCameraImageCapture Class, [1744](#)
 - QCameraViewfinder Class, [1742](#)
 - QCheckBox Class, [1669](#)
 - QColor Class, [1682](#)
 - QColorDialog Class, [1701](#)
 - QComboBox Class, [1645](#)
 - QCompleter Class, [1752](#)
 - QCompleter2 Class, [1753](#)
 - QCompleter3 Class, [1753](#)
 - QCoreApplication Class, [1773](#)
 - QCursor Class, [1757](#)
 - QDate Class, [1715](#)
 - QDateEdit Class, [1664](#)
 - QDateTime Class, [1765](#)
 - QDateTimeEdit Class, [1664](#)
 - QDesktopServices Class, [1621](#)
 - QDesktopWidget Class, [1708](#)
 - QDial Class, [1667](#)
 - QDialog Class, [1698](#)
 - QDir Class, [1640](#)
 - QDirModel Class, [1697](#)
 - QDockWidget Class, [1651](#)
 - QEvent Class, [1675](#)
 - QFileDialog Class, [1677](#)
 - QFileInfo Class, [1696](#)
 - QFileSystemModel Class, [1640](#)
 - QFont Class, [1686](#)
 - QFontDialog Class, [1698](#)
 - QFontMetrics Class, [1747](#)
 - QFrame Class, [1654](#)
 - QFrame2 Class, [1655](#)
 - QFrame3 Class, [1655](#)
 - QGradient Class, [1750](#)
 - QGraphicsVideoItem Class, [1742](#)
 - QGridLayout Class, [1740](#)

QGuiApplication Class, 1771
QHBoxLayout Class, 1634
QHeaderView Class, 1744
QHostAddress Class, 1695
QHostInfo Class, 1695
QIcon Class, 1631
QImage Class, 1725
QInputDialog Class, 1703
QIODevice Class, 1690
QJsonArray Class, 1734
QJsonDocument Class, 1735
QJsonObject Class, 1736
QJsonParseError Class, 1736
QJsonValue Class, 1736
QKeySequence Class, 1702
QLabel Class, 1628
QLayout Class, 1749
QLCDNumber Class, 1702
QLinearGradient Class, 1750
QLineEdit Class, 1631
QListView Class, 1757
QListWidget Class, 1636
QListWidgetItem Class, 1714
QMainWindow Class, 1649
QMdiArea Class, 1755
QMdiSubWindow Class, 1756
QMediaObject Class, 1744
QMediaPlayer Class, 1672
QMediaPlaylist Class, 1672
QMenu Class, 1647
QMenuBar Class, 1647
QMessageBox Class, 1676
QMutex Class, 1763
QMutexLocker Class, 1763
QNetworkAccessManager Class, 1723
QNetworkProxy Class, 1693
QNetworkReply Class, 1724
QNetworkRequest Class, 1722
QObject Class, 1621
QPainter Class, 1679
QPainter2 Class, 1681
QPainterPath Class, 1724
QPen Class, 1682
QPicture Class, 1681
QPixmap Class, 1629
QPixmap2 Class, 1631
QPlainTextEdit Class, 1737
QPoint Class, 1751
QPointF Class, 1750
QPrinter Class, 1685
QProcess Class, 1754
QProgressBar Class, 1661
QPushButton Class, 1629
QRadioButton Class, 1671
QRect Class, 1708
QRegion Class, 1775
QRegularExpression Class, 1733
QRegularExpressionMatch Class, 1733
QRegularExpressionMatchIterator Class, 1734
QScreen Class, 1766
QScrollArea Class, 1751
QSerialPort Class, 1759
QSerialPortInfo Class, 1761
QSize Class, 1631
QSlider Class, 1664
QSpinBox Class, 1662
QSplashScreen Class, 1748
QSplitter Class, 1751
QSqlDatabase Class, 1716
QSqlDriver Class, 1717
QSqlDriverCreatorBase Class, 1721
QSqlError Class, 1719
QSqlField Class, 1720
QSqlIndex Class, 1719
QSqlQuery Class, 1718
QSqlRecord Class, 1719
QStatusBar Class, 1651
QString2 Class, 1754
QStringList Class, 1701
QStringRef Class, 1761
QSystemTrayIcon Class, 1715
QTableView Class, 1658
QTableWidget Class, 1659
QTableWidgetItem Class, 1653
QTabWidget Class, 1652
QTcpServer Class, 1694
QTcpSocket Class, 1693
QTest Class, 1621
QTextBlock Class, 1712
QTextBrowser Class, 1774
QTextCharFormat Class, 1740
QTextCodec Class, 1716
QTextCursor Class, 1699
QTextDocument Class, 1710
QTextEdit Class, 1634
QThread Class, 1731
QThreadPool Class, 1732
QTime Class, 1713
QTimer Class, 1677
QToolBar Class, 1648
QToolButton Class, 1759
QTreeView Class, 1638
QTreeWidget Class, 1641
QTreeWidgetItem Class, 1644
QUrl Class, 1668
QUuid Class, 1759
QVariant Class, 1721
QVBoxLayout Class, 1633

- QVideoWidget Class, 1673
- QVideoWidgetControl Class, 1743
- QWebView Class, 1667
- QWidget Class, 1622
- QWindow Class, 1767
- QXmlStreamAttribute Class, 1731
- QXmlStreamAttributes Class, 1731
- QXmlStreamEntityDeclaration Class, 1730
- QXmlStreamEntityResolver Class, 1730
- QXmlStreamNamespaceDeclaration Class, 1730
- QXmlStreamNotationDeclaration Class, 1730
- QXmlStreamReader Class, 1727
- QXmlStreamWriter Class, 1729
- RingCodeHighlighter Class, 1727
- RingREPL
 - What is new in Ring 1.4?, 95
- RingSQLite
 - What is new in Ring 1.1?, 130
- ringvm_callfunc()
 - Low Level Functions, 782
- RingVM_CallList()
 - Low Level Functions, 778
- RingVM_CFunctionsList()
 - Low Level Functions, 774
- RingVM_ClassesList()
 - Low Level Functions, 774
- ringvm_evalinscope()
 - Low Level Functions, 781
- RingVM_FilesList()
 - Low Level Functions, 779
- RingVM_FunctionsList()
 - Low Level Functions, 774
- ringvm_hideerrorMsg()
 - Low Level Functions, 781
- RingVM_MemoryList()
 - Low Level Functions, 776
- RingVM_PackagesList()
 - Low Level Functions, 775
- ringvm_passerror()
 - Low Level Functions, 781
- ringvm_scopescount()
 - Low Level Functions, 781
- ringvm_settrace()
 - Low Level Functions, 780
- ringvm_tracedata()
 - Low Level Functions, 780
- ringvm_traceevent()
 - Low Level Functions, 780
- ringvm_tracefunc()
 - Low Level Functions, 781
- RingZip
 - Create Zip File, 445
 - Extract Zip File, 445
 - Introduction, 444
 - Print files in Zip file, 445
 - Using RingZip Classes, 445
 - Zip Class Reference, 447
 - ZipEntry Class Reference, 447
- RingZip Library
 - What is new in Ring 1.3?, 114
- Rotate Text
 - Desktop and Mobile Development, 654
- Rules
 - The Type Hints Library, 742
- Run the program
 - Getting Started - First Style, 143
 - Getting Started - Second Style, 146
 - Getting Started - Third Style, 148
- Running Forms
 - Form Designer, 706
- Runtime Errors
 - Reference, 1812
- Samples
 - How to contribute?, 141
- Samples in this book
 - Applications developed in little hours, 13
- Samples Source (Authors)
 - Using RingOpenGL and RingFreeGLUT for 3D Graphics, 522
- Save and Restore Images
 - ODBC Functions, 268
- Save Image Inside the Database
 - MySQL Functions, 276
- Save output to string
 - RingLibCurl, 442
- Scaling and Rotating Images
 - Graphics and Game Programming, 457
- Scope Rules
 - Accessing the class attributes from braces inside class methods, 718
 - Conflict between Class Attributes and Local Variables, 715
 - Conflict between Global Variables and Class Attributes, 714
 - Conflict between self inside braces and self in the class region, 721
 - Creating a Class for each Window in GUI applications, 720
 - Defining Variables and Variables Access, 712
 - How Ring find the Variable?, 713
 - Introduction, 711
 - Summary of Scope Rules, 725
 - The Self Object, 713
 - Three Scopes, 712
 - Using Braces to access objects inside Class Methods, 716

- Using braces to escape from the current object scope, 724
- Using Object.Attribute, 713
- Scope Rules for Functions and Methods
 - Calling a function sharing the name with a method in the current class, 729
 - Example about Sharing Names between Functions and Methods, 727
 - How Ring find a functions and methods?, 727
 - Introduction, 726
- ScriptFunctions Class
 - Web Development (CGI Library), 438
- Search
 - Lists, 208
- Search of global names while defining the class attributes
 - Frequently Asked Questions, 1789
- Security and Internet Functions
 - Decrypt(), 285
 - Download(), 286
 - Encrypt(), 284
 - Example, 285
 - File Hash, 285
 - Introduction, 281
 - MD5(), 282
 - RandBytes(), 285
 - SendEmail(), 286
 - SHA1(), 283
 - SHA224(), 284
 - SHA256(), 283
 - SHA384(), 284
 - SHA512(), 283
- Security Class
 - Stdlib Classes, 362
- Send Parameters
 - Functions - First Style, 194
 - Functions - Second Style, 198
 - Functions - Third Style, 202
- SendEmail()
 - Security and Internet Functions, 286
- Serial Execution of Programs
 - Embedding Ring in Ring, 795
- Set List Item
 - Lists, 207
- setattrattribute()
 - Reflection and Meta-programming, 320
- Setter and Getter
 - Object Oriented Programming, 291
- SHA1()
 - Security and Internet Functions, 283
- SHA224()
 - Security and Internet Functions, 284
- SHA256()
 - Security and Internet Functions, 283
- SHA384()
 - Security and Internet Functions, 284
- SHA512()
 - Security and Internet Functions, 283
- Shared Libraries
 - Extension, 806
- Short-circuit evaluation
 - Control Structures - First Style, 180
- Shutdown() Function
 - System Functions, 253
- sign()
 - Stdlib Functions, 329
- Simple
 - Language Design, 28
- Simple Client and Server Example
 - Desktop and Mobile Development, 658
- Sin() Implementation
 - Extension, 802
- Single: Desktop and Mobile Development
 - Introduction, 589
- sleep()
 - Stdlib Functions, 336
- Smart Garbage Collector
 - Language Design, 36
- Sort()
 - Lists, 208
- Sort() and List of Objects
 - Object Oriented Programming, 296
- Sound Class
 - Game Engine for 2D Games, 473
- Source Code File Sections
 - Program Structure, 205
- space()
 - Low Level Functions, 771
- Special thanks to contributors
 - How to contribute?, 141
- split()
 - Stdlib Functions, 326
- splitmany()
 - Stdlib Functions, 326
- Sprite Automatic Movement
 - Game Engine for 2D Games, 480
- Sprite Class
 - Game Engine for 2D Games, 472
- Sprite Keypress Event
 - Game Engine for 2D Games, 481
- Sprite Mouse Event
 - Game Engine for 2D Games, 482
- Sprite State Event
 - Game Engine for 2D Games, 483
- SQLite
 - Introduction, 278
 - sqlite_close(), 279
 - sqlite_execute(), 279
 - sqlite_init(), 279

- sqlite_open(), 279
- SQLite Class
 - Stdlib Classes, 361
- sqlite_close()
 - SQLite, 279
- sqlite_execute()
 - SQLite, 279
- sqlite_init()
 - SQLite, 279
- sqlite_open()
 - SQLite, 279
- Squares Puzzle Game
 - Applications developed in little hours, 8
- Stack Class
 - Stdlib Classes, 346
- Stars Fighter Game
 - Game Engine for 2D Games, 492
- startswith()
 - Stdlib Functions, 330
- StdBase Class
 - Stdlib Classes, 341
- Stdlib Classes
 - Conversion Class, 358
 - DataType Class, 357
 - DateTime Class, 352
 - Debug Class, 356
 - File Class, 354
 - HashTable Class, 347
 - Internet Class, 363
 - Introduction, 340
 - List Class, 344
 - Math Class, 349
 - MySQL Class, 360
 - ODBC Class, 359
 - Queue Class, 346
 - Security Class, 362
 - SQLite Class, 361
 - Stack Class, 346
 - StdBase Class, 341
 - String Class, 342
 - System Class, 355
 - Tree Class, 348
- Stdlib Functions
 - apppath(), 324
 - binarydigits(), 333
 - capitalized(), 327
 - changestring(), 336
 - dayofweek(), 334
 - direxists(), 336
 - endswith(), 330
 - EpochTime(), 338
 - evenorodd(), 332
 - factorial(), 328
 - factors(), 332
 - fibonacci(), 329
 - file2list(), 330
 - filter(), 326
 - FSize(), 337
 - gcd(), 331
 - getnumber(), 324
 - getstring(), 324
 - Introduction, 322
 - isleapyear(), 333
 - ismainsourcefile(), 336
 - isprime(), 329
 - isspecial(), 327
 - isvowel(), 328
 - JustFileName(), 325
 - JustFilePath(), 324
 - lcm(), 331
 - linecount(), 328
 - list2file(), 329
 - ListAllFiles() Function, 338
 - makedir(), 337
 - map(), 326
 - matrixmulti(), 333
 - matrixtrans(), 334
 - newlist(), 327
 - OSCopyFile() Function, 339
 - OSCopyFolder() Function, 339
 - OSCreateOpenFolder() Function, 339
 - OSDeleteFile() Function, 340
 - OSDeleteFolder() Function, 339
 - OSRenameFile() Function, 340
 - palindrome(), 333
 - permutation(), 334
 - print(), 323
 - Print2Str() Function, 323
 - prodlist(), 332
 - puts(), 323
 - readline(), 335
 - sign(), 329
 - sleep(), 336
 - split(), 326
 - splitmany(), 326
 - startswith(), 330
 - substring(), 335
 - sumlist(), 331
 - SystemCmd() Function, 338
 - SystemSilent() Function, 339
 - times(), 325
 - TrimAll(), 337
 - TrimLeft(), 337
 - TrimRight(), 338
 - value(), 325
- StdLib functions and classes written in Ring
 - What is new in Ring 1.1?, 127
- Step Option

- Control Structures - First Style, 178
- Str2Hex()
 - Data Type, 231
- str2list() and list2str()
 - Strings, 218
- strcmp()
 - Strings, 218
- String Class
 - Stdlib Classes, 342
- String Literals
 - Strings, 214
- String()
 - Data Type, 229
- Strings
 - Access String Letters, 214
 - Convert Letters Case, 214
 - Copy(), 216
 - Find SubString, 217
 - Get Number of Characters from position, 217
 - Get String Length, 214
 - Get Substring from position to end, 217
 - Introduction, 213
 - Left(), 215
 - Lines(), 216
 - Right(), 215
 - str2list() and list2str(), 218
 - strcmp(), 218
 - String Literals, 214
 - Substr(), 216
 - Transform Substring To Another Substring, 217
 - Trim(), 216
- StyleFunctions Class
 - Web Development (CGI Library), 438
- Substr()
 - Strings, 216
- substring()
 - Stdlib Functions, 335
- sumlist()
 - Stdlib Functions, 331
- Summary of Scope Rules
 - Scope Rules, 725
- Super Man 2016 Game
 - Game Engine for 2D Games, 507
- Swap Items
 - Lists, 213
- Switch Between Two Images
 - RingLibSDL, 462
- Switch Statement
 - Control Structures - First Style, 176
 - Control Structures - Second Style, 183
 - Control Structures - Third Style, 186
- Syntax Flexibility
 - Change Language Keywords, 731
 - Change Language Operators, 732
- Introduction, 730
- Load Syntax Files, 732
- Using 'case' as 'on' in switch statements, 738
- Using 'def' as 'func' in functions/methods definition, 738
- Using 'end' keyword after Packages/Classes/Functions, 739
- Using 'endpackage'/'endclass'/'endfunc' keywords after Packages/Classes/Functions, 739
- Using 'put' and 'get' as 'see' and 'give', 738
- Using () around the function parameters, 732
- in the start of the variable name, 735
- Using braces { } in Packages/Classes/Functions, 739
- Using braces to start and end different control structures, 737
- Using Semi-colon after and between statements, 734
- Using the 'else' keyword as 'other' in switch statement, 735
- Using the 'elseif' keyword as 'but' in if statement, 735
- Using the 'end' keyword in different control structures, 736
- What is new in Ring 1.1?, 124
- SysGet() Function
 - System Functions, 248
- System Class
 - Stdlib Classes, 355
- System Functions
 - ChDir() Function, 252
 - CurrentDir() Function, 252
 - Example, 249
 - ExeFileName() Function, 252
 - ExeFolder() Function, 252
 - Get Active Source File Name, 251
 - Get Command Line Arguments, 250
 - Introduction, 246
 - IsAndroid() Function, 249
 - IsFreeBSD() Function, 249
 - IsLinux() Function, 249
 - IsMacOSX() Function, 248
 - IsMSDOS() Function, 248
 - IsUnix() Function, 248
 - IsWindows() Function, 248
 - IsWindows64() Function, 248
 - PrevFileName() Function, 251
 - Shutdown() Function, 253
 - SysGet() Function, 248
 - System() Function, 247
 - Version() Function, 252
 - Windowsnl() Function, 250
- System() Function
 - System Functions, 247
- SystemCmd() Function
 - Stdlib Functions, 338

- SystemSilent() Function
 - Stdlib Functions, 339
- Tempfile()
 - Files, 241
- Templates
 - Web Development (CGI Library), 402
- Tempname()
 - Files, 241
- Testing
 - How to contribute?, 141
- Text Class
 - Game Engine for 2D Games, 472
- The Browser Menu
 - Ring Notepad, 162
- The Camera
 - Using RingOpenGL and RingFreeGLUT for 3D Graphics, 532
- The Cards Game
 - Demo Programs, 680
 - Desktop and Mobile Development, 680
- The Designer Windows
 - Form Designer, 705
- The Difference between Qt and RingQt
 - Desktop and Mobile Development, 690
- The Distribute Menu
 - Ring Notepad, 163
- The documentation says functional programming is supported, but then this happens?
 - Frequently Asked Questions, 1781
- The Edit Menu
 - Ring Notepad, 159
- The File Menu
 - Ring Notepad, 159
- The First GUI Application
 - Desktop and Mobile Development, 590
- The First Triangle
 - Using RingOpenGL and RingFreeGLUT for 3D Graphics, 525
- The First Window using RingFreeGLUT
 - Using RingOpenGL and RingFreeGLUT for 3D Graphics, 523
- The Help Menu
 - Ring Notepad, 163
- The Main File in the Project
 - Ring Notepad, 158
- The Natural Library
 - What is new in Ring 1.4?, 93
- The Program Menu
 - Ring Notepad, 162
- The Properties
 - Form Designer, 705
- The Self Object
 - Scope Rules, 713
- The Tools Menu
 - Ring Notepad, 163
- The Trace Library and the Interactive Debugger
 - BreakPoint, 792
 - Disable BreakPoints, 792
 - Execute Program Line by Line, 791
 - Interactive Debugger, 791
 - Introduction, 789
 - Loading the Trace library, 790
 - Pass Error, 791
 - Trace All Events, 790
 - Trace control flow between functions, 790
 - Using the Interactive Debugger, 792
- The Type Hints Library
 - Example, 741
 - Introduction to the Type Hints Library, 740
 - Rules, 742
 - User Types, 741
 - Using Types inside Code, 742
 - Why Type Hints?, 741
- The View Menu
 - Ring Notepad, 160
- Threads
 - Graphics and Game Programming, 459
- Threads Support
 - Code Generator, 818
- Three Scopes
 - Scope Rules, 712
- TicTacToe 3D Game
 - Using RingOpenGL and RingAllegro for 3D Graphics, 579
- TicTacToe Game
 - Applications developed in little hours, 7
- Time()
 - Date and Time, 220
- TimeList()
 - Date and Time, 221
- times()
 - Stdlib Functions, 325
- Trace All Events
 - The Trace Library and the Interactive Debugger, 790
- Trace control flow between functions
 - The Trace Library and the Interactive Debugger, 790
- Trace Library and Interactive Debugger
 - What is new in Ring 1.5?, 77
- Transaction Example
 - MySQL Functions, 277
- Transactions and Using Commit and Rollback
 - ODBC Functions, 267
- Transform Substring To Another Substring
 - Strings, 217
- Transparent Image
 - Graphics and Game Programming, 458
- Transparent Implementation

- Language Design, 34
- Tree Class
 - Stdlib Classes, 348
- Triangle Rotation
 - Using RingOpenGL and RingFreeGLUT for 3D Graphics, 528
- Trim()
 - Strings, 216
- TrimAll()
 - Stdlib Functions, 337
- TrimLeft()
 - Stdlib Functions, 337
- TrimRight()
 - Stdlib Functions, 338
- TrueType Fonts
 - Graphics and Game Programming, 455
- Try/Catch/Finally
 - Eval() and Debugging, 254
- Trying to be natural
 - Language Design, 29
- Type Hints Library
 - What is new in Ring 1.5?, 80
- Type()
 - Data Type, 225
- Ungetc()
 - Files, 243
- Unsigned()
 - Mathematical Functions, 235
- Update the Android SDK
 - Building RingQt Applications for Mobile, 695
- Upload Files
 - Web Development (CGI Library), 396
- URL Encode
 - Web Development (CGI Library), 401
- Use TTF Fonts
 - RingLibSDL, 464
- User Types
 - The Type Hints Library, 741
- Users registration and Login
 - Web Development (CGI Library), 423
- Using '<' and ':' operators as 'from' keyword
 - What is new in Ring 1.3?, 113
- Using 'case' as 'on' in switch statements
 - Syntax Flexibility, 738
- Using 'def' as 'func' in functions/methods definition
 - Syntax Flexibility, 738
- Using 'end' keyword after Packages/Classes/Functions
 - Syntax Flexibility, 739
- Using 'endpackage'/'endclass'/'endfunc' keywords after Packages/Classes/Functions
 - Syntax Flexibility, 739
- Using 'put' and 'get' as 'see' and 'give'
 - Syntax Flexibility, 738
- Using () around the function parameters
 - Syntax Flexibility, 732
- Using ? to print expression then new line
 - Getting Started - First Style, 144
 - What is new in Ring 1.6?, 46
- in the start of the variable name
 - Syntax Flexibility, 735
- Using && and || operators
 - What is new in Ring 1.6?, 45
- Using _ in numbers
 - Mathematical Functions, 236
- Using Atom
 - Using Other Code Editors, 167
- Using Bootstrap Library using Functions
 - Web Development (CGI Library), 417
- Using Bootstrap Library using Objects
 - Web Development (CGI Library), 418
- Using braces { } in Packages/Classes/Functions
 - Syntax Flexibility, 739
- Using Braces to access objects inside Class Methods
 - Scope Rules, 716
- Using braces to escape from the current object scope
 - Scope Rules, 724
- Using braces to start and end different control structures
 - Syntax Flexibility, 737
- Using C/C++ Compiler and Linker
 - Distributing Ring Application, 760
- Using configuration file that wrap C++ library
 - Code Generator, 820
- Using CR as Carriage return
 - What is new in Ring 1.6?, 45
- Using Emacs Editor
 - Using Other Code Editors, 169
- Using Eval() with our Natural Code
 - Natural Language Programming, 374
- Using f after numbers
 - Mathematical Functions, 236
- Using Fonts
 - Using RingOpenGL and RingFreeGLUT for 3D Graphics, 549
- Using Geany
 - Using Other Code Editors, 166
- Using Layout
 - Desktop and Mobile Development, 591
- Using Layouts
 - Form Designer, 711
- Using Lists during definition
 - Lists, 211
- Using Many Source Code Files
 - Program Structure, 205
- Using Notepad++
 - Using Other Code Editors, 165
- Using NULL instead of NULLPointer()
 - What is new in Ring 1.2?, 121

- Using Object.Attribute
 - Scope Rules, [713](#)
- Using Other Code Editors
 - Introduction, [164](#)
 - Using Atom, [167](#)
 - Using Emacs Editor, [169](#)
 - Using Geany, [166](#)
 - Using Notepad++, [165](#)
 - Using Sublime Text 2, [168](#)
 - Using Visual Studio IDE, [169](#)
- Using QCheckBox
 - Desktop and Mobile Development, [618](#)
- Using QComboBox Class
 - Desktop and Mobile Development, [599](#)
- Using QDateEdit
 - Desktop and Mobile Development, [613](#)
- Using QDesktopWidget Class
 - Desktop and Mobile Development, [653](#)
- Using QDial
 - Desktop and Mobile Development, [614](#)
- Using QDockWidget
 - Desktop and Mobile Development, [606](#)
- Using QFrame
 - Desktop and Mobile Development, [623](#)
- Using QInputDialog Class
 - Desktop and Mobile Development, [644](#)
- Using qLCDNumber
 - Desktop and Mobile Development, [641](#)
- Using QProgressBar
 - Desktop and Mobile Development, [610](#)
- Using QProgressBar and Timer
 - Desktop and Mobile Development, [633](#)
- Using QRadioButton and QButtonGroup
 - Desktop and Mobile Development, [619](#)
- Using QSlider
 - Desktop and Mobile Development, [611](#)
- Using QSpinBox
 - Desktop and Mobile Development, [611](#)
- Using QTableWidget
 - Desktop and Mobile Development, [609](#)
- Using QTabWidget
 - Desktop and Mobile Development, [607](#)
- Using QTreeView and QFileSystemModel
 - Desktop and Mobile Development, [597](#)
- Using QTreeWidgetItem and QTreeWidgetItem
 - Desktop and Mobile Development, [598](#)
- Using QWebView
 - Desktop and Mobile Development, [617](#)
- Using Ring2EXE
 - Building RingQt Applications for Mobile, [698](#)
 - Distributing Ring Application using Ring2EXE, [761](#)
- Using RingOpenGL and RingAllegro for 3D Graphics
 - 3D Cube and Texture, [570](#)
 - Introduction, [569](#)
 - Many Cubes, [574](#)
 - TicTacToe 3D Game, [579](#)
- Using RingOpenGL and RingFreeGLUT for 3D Graphics
 - Drawing using RingOpenGL, [524](#)
 - Frames Per Second, [557](#)
 - Introduction, [521](#)
 - Keyboard Events and Colors, [529](#)
 - Make a Cube using RingOpenGL and RingFreeGLUT, [566](#)
 - Menu Events, [542](#)
 - Mouse Events, [538](#)
 - Samples Source (Authors), [522](#)
 - The Camera, [532](#)
 - The First Triangle, [525](#)
 - The First Window using RingFreeGLUT, [523](#)
 - Triangle Rotation, [528](#)
 - Using Fonts, [549](#)
 - What is RingFreeGLUT?, [523](#)
 - What is RingOpenGL?, [522](#)
 - Window Resize Event, [526](#)
- Using RingZip Classes
 - RingZip, [445](#)
- Using Self.Attribute
 - Object Oriented Programming, [298](#)
- Using Self.Attribute in the Class Region to define new attributes
 - What is new in Ring 1.1?, [131](#)
- Using Semi-colon after and between statements
 - Syntax Flexibility, [734](#)
- Using Sublime Text 2
 - Using Other Code Editors, [168](#)
- Using Tab instead of char(9)
 - What is new in Ring 1.6?, [44](#)
- Using the
 - operator as not
 - What is new in Ring 1.6?, [45](#)
- Using the 'else' keyword as 'other' in switch statement
 - Syntax Flexibility, [735](#)
- Using the 'elseif' keyword as 'but' in if statement
 - Syntax Flexibility, [735](#)
- Using the 'end' keyword in different control structures
 - Syntax Flexibility, [736](#)
- Using the Interactive Debugger
 - The Trace Library and the Interactive Debugger, [792](#)
- Using the Natural Library
 - Defining Commands, [382](#)
 - Defining commands using classes, [385](#)
 - Introduction, [379](#)
 - Natural Library - Demo Program, [380](#)
 - Operators, [384](#)
- Using the QColorDialog Class
 - Desktop and Mobile Development, [640](#)
- Using the QFileDialog Class
 - Desktop and Mobile Development, [635](#)

- Using the QListWidget Class
 - Desktop and Mobile Development, 594
- Using the QTextEdit Class
 - Desktop and Mobile Development, 593
- Using the QTimer Class
 - Desktop and Mobile Development, 632
- Using the tool
 - Code Generator, 811
- Using This.Attribute in nested Braces inside the Class
 - Methods
 - What is new in Ring 1.1?, 131
- Using Types inside Code
 - The Type Hints Library, 742
- Using Visual Studio IDE
 - Using Other Code Editors, 169
- value()
 - Stdlib Functions, 325
- Variables
 - Deep Copy, 171
 - Dynamic Typing, 171
 - Introduction, 170
 - Weakly Typed, 172
- Variables Scope
 - Functions - First Style, 195
 - Functions - Second Style, 199
 - Functions - Third Style, 203
- varptr()
 - Low Level Functions, 771
- Version() Function
 - System Functions, 252
- Video-Music-Player Application
 - Applications developed in little hours, 9
 - What is new in Ring 1.5?, 48
- Virtual Machine Instructions
 - Reference, 1815
- Visual Implementation
 - Language Design, 35
- Weakly Typed
 - Variables, 172
- Web Development (CGI Library)
 - Application Class, 435
 - Configure the Apache web server, 387
 - Cookies, 399
 - CRUD Example using MVC, 421
 - Database, ModelBase & ControllerBase classes, 429
 - Generating Pages using Objects, 411
 - Gradient, 410
 - Hash Functions, 405
 - Hello World Program using the Web Library, 388
 - HTML Lists, 408
 - HTML Special Characters, 404
 - HTML Tables, 409
 - HtmlPage Class, 416, 440
 - HTTP Get Example, 389
 - HTTP POST Example, 394
 - Introduction, 386
 - Page Class, 436
 - Random Image, 407
 - Ring CGI Hello World Program, 388
 - ScriptFunctions Class, 438
 - StyleFunctions Class, 438
 - Templates, 402
 - Upload Files, 396
 - URL Encode, 401
 - Users registration and Login, 423
 - Using Bootstrap Library using Functions, 417
 - Using Bootstrap Library using Objects, 418
 - Web Library Features, 389
 - WebLib API, 434
 - WebPage Class, 439
- Web Library Features
 - Web Development (CGI Library), 389
- WebLib API
 - Web Development (CGI Library), 434
- WebPage Class
 - Web Development (CGI Library), 439
- Weight History Application
 - Desktop and Mobile Development, 661
- Werdy Application
 - Applications developed in little hours, 12
- What about predefined parameters or optional parameters in functions?
 - Frequently Asked Questions, 1791
- What about the Boolean values in Ring?
 - Frequently Asked Questions, 1783
- What are the advantages to using Ring over C# or Java?
 - Frequently Asked Questions, 1781
- What are the advantages to using Ring over Lisp or Smalltalk?
 - Frequently Asked Questions, 1778
- What are the advantages to using Ring over native C or C++?
 - Frequently Asked Questions, 1779
- What are the advantages to using Ring over Python and Ruby?
 - Frequently Asked Questions, 1780
- What are the advantages to using Ring over Tcl and Lua?
 - Frequently Asked Questions, 1780
- What happens when we create a new object?
 - Frequently Asked Questions, 1787
- What is new in Ring 1.1?
 - Better Code Generator for Extensions, 131
 - Better Documentation, 131
 - Better Natural Language Programming Support, 123
 - Game Engine for 2D Games, 129
 - Generate/Execute Ring Object Files (*.ringo), 124

- Introduction, 122
- List of changes and new features, 123
- New Functions and Changes, 126
- RingLibSDL, 129
- RingSQLite, 130
- StdLib functions and classes written in Ring, 127
- Syntax Flexibility, 124
- Using Self.Attribute in the Class Region to define new attributes, 131
- Using This.Attribute in nested Braces inside the Class Methods, 131
- What is new in Ring 1.2?
 - Better Call Command, 121
 - Better Functions, 118
 - Better Quality, 121
 - Better Ring Notepad, 118
 - Better RingQt, 118
 - Display Warnings Option, 121
 - Introduction, 116
 - List of changes and new features, 117
 - New Functions, 117
 - Objects Library for RingQt, 119
 - RingLibCurl, 120
 - Using NULL instead of NULLPointer(), 121
- What is new in Ring 1.3?
 - Better LoopExit Command, 112
 - Better Ring Notepad, 108
 - Better RingQt, 104
 - Better StdLib, 112
 - Embedding Ring in Ring without sharing the State, 114
 - Form Designer, 115
 - Introduction, 103
 - List of changes and new features, 104
 - New Functions, 113
 - Return Self by Reference, 113
 - Ring mode for Emacs Editor, 111
 - RingZip Library, 114
 - Using '<' and ':' operators as 'from' keyword, 113
- What is new in Ring 1.4.1?
 - What is new in Ring 1.4?, 100
- What is new in Ring 1.4?
 - Better RingQt, 99
 - Better StdLib, 97
 - Better WebLib, 97
 - Change: Basic Extensions are separated from RingVM, 92
 - Convert between Numbers and Bytes, 96
 - Introduction, 91
 - List of changes and new features, 92
 - New Style to Ring Notepad, 95
 - Qt Class Convertor, 99
 - RingREPL, 95
 - The Natural Library, 93
 - What is new in Ring 1.4.1?, 100
 - What is new in Ring 1.5.1?
 - What is new in Ring 1.5?, 81
 - What is new in Ring 1.5.2?
 - What is new in Ring 1.5?, 86
 - What is new in Ring 1.5.3?
 - What is new in Ring 1.5?, 87
 - What is new in Ring 1.5.4?
 - What is new in Ring 1.5?, 90
 - What is new in Ring 1.5?
 - Better Code Generator for Extensions, 72
 - Better Documentation Generator for Extensions, 73
 - Better Objects Library, 59
 - Better Quality, 81
 - Better Ring Notepad, 50
 - Better RingQt, 57
 - Better StdLib, 53
 - Better WebLib, 54
 - Calculator Application, 49
 - Introduction, 47
 - List of changes and new features, 48
 - More Syntax Flexibility, 79
 - Ring VM - Tracing Functions, 73
 - RingFreeGLUT Extension, 59
 - RingOpenGL Extension, 68
 - Trace Library and Interactive Debugger, 77
 - Type Hints Library, 80
 - Video-Music-Player Application, 48
 - What is new in Ring 1.5.1?, 81
 - What is new in Ring 1.5.2?, 86
 - What is new in Ring 1.5.3?, 87
 - What is new in Ring 1.5.4?, 90
 - Windows StartUp Manager Application, 49
 - What is new in Ring 1.6?
 - Better Ring For Android, 38
 - Better Ring Notepad, 41
 - Better RingQt, 43
 - Better RingREPL, 44
 - Better RingVM, 44
 - Better Scripts for building Ring, 40
 - Better StdLib, 44
 - Employee Application, 37
 - Introduction, 36
 - List of changes and new features, 37
 - New Tool: Folder2qrc, 39
 - New Tool: Ring2EXE, 38
 - RingConsoleColors Extension, 40
 - RingMurmurHash Extension, 40
 - Using ? to print expression then new line, 46
 - Using && and || operators, 45
 - Using CR as Carriage return, 45
 - Using Tab instead of char(9), 44
 - Using the operator as not, 45

- What is RingFreeGLUT?
 - Using RingOpenGL and RingFreeGLUT for 3D Graphics, [523](#)
- What is RingOpenGL?
 - Using RingOpenGL and RingFreeGLUT for 3D Graphics, [522](#)
- What is the difference between Ring and Python? And is Ring Open Source?
 - Frequently Asked Questions, [1779](#)
- Where can I write a program and execute it?
 - Frequently Asked Questions, [1790](#)
- Which of 3 coding styles are commonly used or recommended by the community?
 - Frequently Asked Questions, [1808](#)
- While Loop
 - Control Structures - First Style, [177](#)
 - Control Structures - Third Style, [187](#)
- Why do we need Yet Another Programming Language (YAPL)?
 - Frequently Asked Questions, [1777](#)
- Why I get a strange result when printing nl with lists?
 - Frequently Asked Questions, [1792](#)
- Why I get Calling Function without definition Error?
 - Frequently Asked Questions, [1803](#)
- Why Ring is largely focussed on UI creation?
 - Frequently Asked Questions, [1778](#)
- Why Ring is not case-sensitive
 - Frequently Asked Questions, [1785](#)
- Why Ring is weakly typed?
 - Frequently Asked Questions, [1778](#)
- Why Ring uses 'See', 'Give', 'But' and 'Ok' Keywords?
 - Frequently Asked Questions, [1782](#)
- Why Ring?
 - Language Design, [28](#)
- Why setClickEvent() doesn't see the object methods directly?
 - Frequently Asked Questions, [1803](#)
- Why the ability to define your own languages Instead of just handing over the syntax so you can parse it using whatever code you like?
 - Frequently Asked Questions, [1781](#)
- Why the Assignment operator uses Deep copy?
 - Frequently Asked Questions, [1786](#)
- Why the window title bar is going outside the screen?
 - Frequently Asked Questions, [1797](#)
- Why this example use the GetChar() twice?
 - Frequently Asked Questions, [1794](#)
- Why Type Hints?
 - The Type Hints Library, [741](#)
- Why we don't use () after the qApp class name?
 - Frequently Asked Questions, [1797](#)
- Why you can specify the number of loops you want to break out of?
 - Frequently Asked Questions, [1782](#)
- Window Flags
 - Form Designer, [710](#)
- Window Resize Event
 - Using RingOpenGL and RingFreeGLUT for 3D Graphics, [526](#)
- Windows StartUp Manager Application
 - Applications developed in little hours, [11](#)
 - What is new in Ring 1.5?, [49](#)
- Windowsnl() Function
 - System Functions, [250](#)
- Wrap structures
 - Code Generator, [813](#)
- Wrapping C++ Classes
 - Code Generator, [819](#)
- Write file using Write()
 - Files, [238](#)
- Writing Comments
 - Getting Started - First Style, [145](#)
 - Getting Started - Second Style, [147](#)
 - Getting Started - Third Style, [149](#)
- Zip Class Reference
 - RingZip, [447](#)
- ZipEntry Class Reference
 - RingZip, [447](#)