Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

1 ALIGN BOTTOM

1. REDIRECT Alignment modes

2 ALIGN BOTTOM LEFT

1. REDIRECT Alignment modes

3 ALIGN BOTTOM RIGHT

1. REDIRECT Alignment modes

4 ALIGN CENTER

1. REDIRECT Alignment modes

5 ALIGN CENTER LEFT

1. REDIRECT Alignment modes

6 ALIGN CENTER RIGHT

1. REDIRECT Alignment modes

7 ALIGN TOP

1. REDIRECT Alignment modes

8 ALIGN TOP LEFT

1. REDIRECT Alignment modes

9 ALIGN TOP RIGHT

1. REDIRECT Alignment modes

10 ALL SOUND

ALL_SOUND can be passed to stop_wav() to stop all sound playing.

Constant	- *Value*	- *Description*
ALL_SOUND	- -1	- Stop all sound playing.

11 ALL TEXT

ALL_TEXT can be passed to delete_text() to delete all dynamically written texts on the screen.

Constant	- *Value*	- *Description*
ALL_TEXT	- 0	- Delete all texts dynamically written on the screen.

12 B ABLEND

1. REDIRECT blit_flags

13 B ALPHA

1. REDIRECT blit_flags

14 B HMIRROR

1. REDIRECT blit_flags

15 B NOCOLORKEY

1. REDIRECT blit_flags

16 B SBLEND

1. REDIRECT blit_flags

17 B TRANSLUCENT

1. REDIRECT blit_flags

18 B VMIRROR

1. REDIRECT blit_flags

19 BACKGROUND

1. REDIRECT Predefined_graphcodes

20 BLUR 3X3

1. REDIRECT blur

21 BLUR 5X5

1. REDIRECT blur

22 BLUR 5X5 MAP

1. REDIRECT blur

23 BLUR NORMAL

1. REDIRECT blur

24 C 0

1. REDIRECT Coordinatenumber flags

25 C 1

1. REDIRECT Coordinatenumber flags

26 C 2

1. REDIRECT Coordinatenumber flags

27 C 3

1. REDIRECT Coordinatenumber flags

28 C 4

1. REDIRECT Coordinatenumber flags

29 C 5

1. REDIRECT Coordinatenumber flags

30 C 6

1. REDIRECT Coordinatenumber flags

31 C 7

1. REDIRECT Coordinatenumber flags

32 C 8

1. REDIRECT Coordinatenumber flags

33 C 9

1. REDIRECT Coordinatenumber flags

34 C M7

1. REDIRECT Coordinatetype_modes

35 C SCREEN

1. REDIRECT Coordinatetype_modes

36 C SCROLL

1. REDIRECT Coordinatetype_modes

37 CD ERROR

1. REDIRECT CD_statuscodes

38 CD PAUSED

1. REDIRECT CD_statuscodes

39 CD PLAYING

1. REDIRECT CD_statuscodes

40 CD STOPPED

1. REDIRECT CD_statuscodes

41 CD TRAYEMPTY

1. REDIRECT CD_statuscodes

42 COMPLETE DUMP

1. REDIRECT dump_modes

43 COMPLETE RESTORE

1. REDIRECT restore_modes

44 CRYPT 3DES

1. REDIRECT crypt_modes

45 CRYPT DES

1. REDIRECT crypt_modes

46 CRYPT NONE

1. REDIRECT crypt_modes

47 DOUBLE BUFFER

1. REDIRECT render_flags

48 FALSE

48.1 Definition

INT FALSE

False is a constant integer, equal to the value 0. It is used to state that something is false and not true.

Checking whether a variable is false, is the same as checking if it's zero.

48.2 Example

```
Program example;
Private
    int b = false;
Begin

    // comparison with the constant FALSE
    if(b == false)
        say("b was FALSE! so b==0");
    else
        say("b was not FALSE! so b!=0");
    end

    // checking the integer itself
    if(!b)
        say("b was false! so b==0");
    else
        say("b was true! so b!=0");
    end

    Loop
        frame;
    End

End
```

49 G ANIMATION SPEED

1. REDIRECT graphical_infotypes

50 G ANIMATION STEP

1. REDIRECT graphical_infotypes

51 G ANIMATION STEPS

1. REDIRECT graphical_infotypes

52 G CENTER X

1. REDIRECT graphical_infotypes

53 G CENTER Y

1. REDIRECT graphical_infotypes

54 G DEPTH

1. REDIRECT graphical_infotypes

55 G FRAMES

1. REDIRECT graphical_infotypes

56 G HEIGHT

1. REDIRECT graphical_infotypes

57 G PITCH

1. REDIRECT graphical_infotypes

58 G WIDE

1. REDIRECT graphical_infotypes

59 G WIDTH

1. REDIRECT graphical_infotypes

60 G X CENTER

1. REDIRECT graphical_infotypes

61 HW SURFACE

1. REDIRECT render_flags

62 JOY HAT CENTERED

1. REDIRECT Joystick constants

63 JOY HAT DOWN

1. REDIRECT Joystick constants

64 JOY HAT LEFT

1. REDIRECT Joystick constants

65 JOY HAT LEFTDOWN

1. REDIRECT Joystick constants

66 JOY HAT LEFTUP

1. REDIRECT Joystick constants

67 JOY HAT RIGHT

1. REDIRECT Joystick constants

68 JOY HAT RIGHTDOWN

1. REDIRECT Joystick constants

69 JOY HAT RIGHTUP

1. REDIRECT Joystick constants

70 JOY HAT UP

1. REDIRECT Joystick constants

71 M1024X768

1. REDIRECT resolution_modes

72 M1280X1024

1. REDIRECT resolution_modes

73 M320X200

1. REDIRECT resolution_modes

74 M320X240

1. REDIRECT resolution_modes

75 M320X400

1. REDIRECT resolution_modes

76 M360X240

1. REDIRECT resolution_modes

77 M376X282

1. REDIRECT resolution_modes

78 M400X300

1. REDIRECT resolution_modes

79 M512X384

1. REDIRECT resolution_modes

80 M640X400

1. REDIRECT resolution_modes

81 M640X480

1. REDIRECT resolution_modes

82 M800X600

1. REDIRECT resolution_modes

83 MAX BYTE

83.1 Definition

CHAR MAX_BYTE = 255

The maximum integer value reachable with a byte (unsigned char).

See also MIN_BYTE.

Template:Moduledocbox

84 MAX CHAR

84.1 Definition

CHAR MAX_CHAR = 255

The maximum integer value reachable with a char (byte).

See also MIN_CHAR.

Template:Moduledocbox

85 MAX DWORD

85.1 Definition

DWORD MAX_DWORD = 4294967295

The maximum integer value reachable with a dword (unsigned integer).

See also MIN_DWORD.

Template:Moduledocbox

86 MAX INT

86.1 Definition

INT MAX_INT = 2147483647

The maximum integer value reachable with an int.

See also MIN_INT.

Template:Moduledocbox

87 MAX SBYTE

87.1 Definition

CHAR MAX_SBYTE = 127

The maximum integer value reachable with a signed byte.

See also MIN_SBYTE.

Template:Moduledocbox

88 MAX SHORT

88.1 Definition

SHORT MAX_SHORT = 32767

The maximum integer value reachable with a short (signed word).

See also MIN_SHORT.

Template:Moduledocbox

89 MAX WORD

89.1 Definition

WORD MAX_WORD = 65535

The maximum integer value reachable with a word (unsigned short).

See also MIN_WORD.

Template:Moduledocbox

90 MIN BYTE

90.1 Definition

CHAR MIN_BYTE = 0

The minimum integer value reachable with a byte (unsigned char).

See also MAX_BYTE.

Template:Moduledocbox

91 MIN CHAR

91.1 Definition

CHAR MIN_CHAR = 0

The minimum integer value reachable with a char (byte).

See also MAX_CHAR.

Template:Moduledocbox

92 MIN DWORD

92.1 Definition

DWORD MIN_DWORD = 0

The minimum integer value reachable with a dword (unsigned integer).

See also MAX_DWORD.

Template:Moduledocbox

93 MIN INT

93.1 Definition

INT MIN_INT = -2147483648

The minimum integer value reachable with an int.

See also MAX_INT.

Template:Moduledocbox

94 MIN SBYTE

94.1 Definition

CHAR MIN_SBYTE = -128

The minimum integer value reachable with a signed byte.

See also MAX_SBYTE.

Template:Moduledocbox

95 MIN SHORT

95.1 Definition

SHORT MIN_SHORT = -32768

The minimum integer value reachable with a short (signed word).

See also MAX_SHORT.

Template:Moduledocbox

96 MIN WORD

96.1 Definition

WORD MIN_WORD = 0

The minimum integer value reachable with a word (unsigned short).

See also MAX_WORD.

Template:Moduledocbox

97 MODE 16BITS

1. REDIRECT graph_modes

98 MODE 16BPP

1. REDIRECT graph_modes

99 MODE 2XSCALE

1. REDIRECT render_flags

100 MODE 8BITS

1. REDIRECT graph_modes

101 MODE 8BPP

1. REDIRECT graph_modes

102 MODE DOUBLEBUFFER

1. REDIRECT render_flags

103 MODE FRAMELESS

1. REDIRECT render_flags

104 MODE FULLSCREEN

1. REDIRECT render_flags

105 MODE HARDWARE

1. REDIRECT render_flags

106 MODE MODAL

1. REDIRECT render_flags

107 MODE MONO

1. REDIRECT sound_modes

108 MODE STEREO

1. REDIRECT sound_modes

109 MODE WAITVSYNC

1. REDIRECT render_flags

110 MODE WINDOW

1. REDIRECT render_flags

111 NO RESTORE

1. REDIRECT restore_modes

112 NULL

112.1 Description

NULL is a pointer, pointing to nothing. In programming it points to 0x00000000, or just plain 0.

The following check is the same as checking if a pointer variable is NULL:

```
if(my_pointer)
end
```

In which my_pointer is a pointer variable.

Constant	*Value*	*Description*
NULL	0	0x00000000, pointing to nothing.

113 O RDWR

1. REDIRECT readwrite_modes

114 O READ

1. REDIRECT readwrite_modes

115 O READWRITE

1. REDIRECT readwrite_modes

116 O WRITE

1. REDIRECT readwrite_modes

117 O ZREAD

1. REDIRECT readwrite_modes

118 O ZWRITE

1. REDIRECT readwrite_modes

119 OS BEOS

1. REDIRECT OS_codes

120 OS BSD

1. REDIRECT OS_codes

121 OS DC

1. REDIRECT OS_codes

122 OS GP2X

1. REDIRECT OS_codes

123 OS GP32

1. REDIRECT OS_codes

124 OS LINUX

1. REDIRECT OS_codes

125 OS MACOS

1. REDIRECT OS_codes

126 OS WIN32

1. REDIRECT OS_codes

127 PARTIAL DUMP

1. REDIRECT dump_modes

128 PARTIAL RESTORE

1. REDIRECT restore_modes

129 PF NODIAG

1. REDIRECT pathfind_flags

130 PF REVERSE

1. REDIRECT pathfind_flags

131 PI

131.1 Definition

INT PI == 180000

The equivalent to the mathematical pi (3.14159265...), in thousandths of a degree. It can be used to define angles, like assigning one to the local variable angle, or passing one to the function xadvance().

131.2 Example

```
angle = PI/2; // 90000 (90 degrees)
xadvance(-PI/4,10); // -45000 (-45 degrees)
```

132 S DFL

1. REDIRECT signal actions

133 S FREEZE

1. REDIRECT signals

134 S FREEZE FORCE

1. REDIRECT signals

135 S FREEZE TREE

1. REDIRECT signals

136 S FREEZE TREE FORCE

1. REDIRECT signals

137 S IGN

1. REDIRECT signal actions

138 S KILL

1. REDIRECT signals

139 S KILL FORCE

1. REDIRECT signals

140 S KILL TREE

1. REDIRECT signals

141 S KILL TREE FORCE

1. REDIRECT signals

142 S SLEEP

1. REDIRECT signals

143 S SLEEP FORCE

1. REDIRECT signals

144 S SLEEP TREE

1. REDIRECT signals

145 S SLEEP TREE FORCE

1. REDIRECT signals

146 S WAKEUP

1. REDIRECT signals

147 S WAKEUP FORCE

1. REDIRECT signals

148 S WAKEUP TREE

1. REDIRECT signals

149 S WAKEUP TREE FORCE

1. REDIRECT signals

150 SCALE HQ2X

1. REDIRECT scale_modes

151 SCALE NOFILTER

1. REDIRECT scale_modes

152 SCALE NONE

1. REDIRECT scale_modes

153 SCALE NORMAL2X

1. REDIRECT scale_modes

154 SCALE SCALE2X

1. REDIRECT scale_modes

155 SCALE SCANLINE2X

1. REDIRECT scale_modes

156 SCREEN

1. REDIRECT Predefined_graphcodes

157 SEEK CUR

1. REDIRECT seek_modes

158 SEEK END

1. REDIRECT seek_modes

159 SEEK SET

1. REDIRECT seek_modes

160 STATUS DEAD

1. REDIRECT Process_statuscodes

161 STATUS FROZEN

1. REDIRECT Process_statuscodes

162 STATUS KILLED

1. REDIRECT Process_statuscodes

163 STATUS RUNNING

1. REDIRECT Process_statuscodes

164 STATUS SLEEPING

1. REDIRECT Process_statuscodes

165 STATUS WAITING

1. REDIRECT Process_statuscodes

166 TRUE

166.1 Definition

INT TRUE

True is a constant integer, equal to the value 1. It is used to state that something is true and not false.

Checking whether a variable is true, is the same as checking if it's nonzero.

166.2 Example

```
Program example;
Private
    int b = true;
Begin

    // comparison with the constant TRUE
    if(b == true)
        say("b was TRUE! so b==1");
    else
        say("b was not TRUE! so b!=1");
    end

    // checking the integer itself
    if(b)
        say("b was true! so b!=0");
    else
        say("b was false! so b==0");
    end

    Loop
        frame;
    End

End
```

167 \ 0

1. REDIRECT Scancodes

168 \ 1

1. REDIRECT Scancodes

169 \ 2

1. REDIRECT Scancodes

170 \ 3

1. REDIRECT Scancodes

171 \ 4

1. REDIRECT Scancodes

172 \ 5

1. REDIRECT Scancodes

173 \ 6

1. REDIRECT Scancodes

174 \ 7

1. REDIRECT Scancodes

175 \ 8

1. REDIRECT Scancodes

176 \ 9

1. REDIRECT Scancodes

177 \ A

1. REDIRECT Scancodes

178 \ ALT

1. REDIRECT Scancodes

179 \ APOSTROPHE

1. REDIRECT Scancodes

1. REDIRECT Scancodes

181 \ B

1. REDIRECT Scancodes

182 \ BACKSLASH

1. REDIRECT Scancodes

183 \ BACKSPACE

1. REDIRECT Scancodes

184 \ C

1. REDIRECT Scancodes

185 \ C ASTERISK

1. REDIRECT Scancodes

186 \ C BACKSLASH

1. REDIRECT Scancodes

187 \ C CENTER

1. REDIRECT Scancodes

188 \ C DEL

1. REDIRECT Scancodes

189 \ C DOWN

1. REDIRECT Scancodes

190 \ C END

1. REDIRECT Scancodes

191 \ C ENTER

1. REDIRECT Scancodes

192 \ C HOME

1. REDIRECT Scancodes

193 \ C INS

1. REDIRECT Scancodes

194 \ C LEFT

1. REDIRECT Scancodes

195 \ C MINUS

1. REDIRECT Scancodes

196 \ C PGDN

1. REDIRECT Scancodes

197 \ C PGUP

1. REDIRECT Scancodes

198 \ C PLUS

1. REDIRECT Scancodes

199 \ C RIGHT

1. REDIRECT Scancodes

200 \ C UP

1. REDIRECT Scancodes

201 \ CAPS LOCK

1. REDIRECT Scancodes

202 \ COMMA

1. REDIRECT Scancodes

203 \ CONTROL

1. REDIRECT Scancodes

204 \ D

1. REDIRECT Scancodes

205 \ DEL

1. REDIRECT Scancodes

206 \ DOWN

1. REDIRECT Scancodes

207 \ E

1. REDIRECT Scancodes

208 \ END

1. REDIRECT Scancodes

209 \ ENTER

1. REDIRECT Scancodes

210 \ EQUALS

1. REDIRECT Scancodes

211 \ ESC

1. REDIRECT Scancodes

212 \ F

1. REDIRECT Scancodes

213 \ F1

1. REDIRECT Scancodes

214 \ F10

1. REDIRECT Scancodes

215 \ F11

1. REDIRECT Scancodes

216 \ F12

1. REDIRECT Scancodes

217 \ F2

1. REDIRECT Scancodes

218 \ F3

1. REDIRECT Scancodes

219 \ F4

1. REDIRECT Scancodes

220 \ F5

1. REDIRECT Scancodes

221 \ F6

1. REDIRECT Scancodes

222 \ F7

1. REDIRECT Scancodes

223 \ F8

1. REDIRECT Scancodes

224 \ F9

1. REDIRECT Scancodes

225 \ G

1. REDIRECT Scancodes

226 \ GREATER

1. REDIRECT Scancodes

227 \ H

1. REDIRECT Scancodes

228 \ HOME

1. REDIRECT Scancodes

229 \ I

1. REDIRECT Scancodes

230 \ INS

1. REDIRECT Scancodes

231 \ J

1. REDIRECT Scancodes

232 \ K

1. REDIRECT Scancodes

233 \ L

1. REDIRECT Scancodes

234 \ L ALT

1. REDIRECT Scancodes

235 \ L BRACHET

1. REDIRECT Scancodes

236 \ L CONTROL

1. REDIRECT Scancodes

237 \ L SHIFT

1. REDIRECT Scancodes

238 \ L WINDOWS

1. REDIRECT Scancodes

239 \ LEFT

1. REDIRECT Scancodes

240 \ LESS

1. REDIRECT Scancodes

241 \ M

1. REDIRECT Scancodes

242 \ MENU

1. REDIRECT Scancodes

243 \ MINUS

1. REDIRECT Scancodes

244 \ N

1. REDIRECT Scancodes

245 \ NUM LOCK

1. REDIRECT Scancodes

246 \ O

1. REDIRECT Scancodes

247 \ P

1. REDIRECT Scancodes

248 \ P NOWAIT

1. REDIRECT Exec#Notes

249 \ P WAIT

1. REDIRECT Exec#Notes

250 \ PGDN

1. REDIRECT Scancodes

251 \ PGUP

1. REDIRECT Scancodes

252 \ PLUS

1. REDIRECT Scancodes

253 \ POINT

1. REDIRECT Scancodes

254 \ PRN SCR

1. REDIRECT Scancodes

255 \ Q

1. REDIRECT Scancodes

256 \ R

1. REDIRECT Scancodes

257 \ R ALT

1. REDIRECT Scancodes

258 \ R BRACHET

1. REDIRECT Scancodes

259 \ R CONTROL

1. REDIRECT Scancodes

260 \ R SHIFT

1. REDIRECT Scancodes

261 \ R WINDOWS

1. REDIRECT Scancodes

262 \ RIGHT

1. REDIRECT Scancodes

263 \ S

1. REDIRECT Scancodes

264 \ SCROLL LOCK

1. REDIRECT Scancodes

265 \ SEMICOLON

1. REDIRECT Scancodes

266 \ SLASH

1. REDIRECT Scancodes

267 \ SPACE

1. REDIRECT Scancodes

268 \ T

1. REDIRECT Scancodes

269 \ TAB

1. REDIRECT Scancodes

270 \ U

1. REDIRECT Scancodes

271 \ UP

1. REDIRECT Scancodes

272 \ V

1. REDIRECT Scancodes

273 \ W

1. REDIRECT Scancodes

274 \ WAVE

1. REDIRECT Scancodes

275 \ X

1. REDIRECT Scancodes

276 \ Y

1. REDIRECT Scancodes

277 \ Z

1. REDIRECT Scancodes