

Constants

12
%1-10
0&12
3&14FE
0.1A1F
#CONSTANT name value
#INCLUDE "filename"

Operators

Arithmetic
String
Relational
Bitwise
Boolean

([^], mod, %, /, +, -)
(+)
(<, <=, >, >=, =, <>)
(<<, >>, &&, ||, ~, ~-)
(AND, OR, NOT)

Variable Declarations

variableName
variableName#
variableName\$
variableName AS INTEGER
variableName AS FLOAT
variableName AS BOOLEAN
variableName AS BYTE
variableName AS WORD
variableName AS DWORD
variableName AS DOUBLE INTEGER
variableName AS DOUBLE FLOAT
TYPE typeName
field

ENDTYPE
variableName AS typeName
GLOBAL variable (AS type)
DIM name(V)
UNDIM name(0)

Core Statements

END
READ variableName
DATA value, value, value...
RESTORE
IF condition
(ELSE statement(s)1
(ELSE statement(s)2
SELECT value
CASE v1 statement(s)
ENDCASE
CASE v2 statement(s)
ENDCASE
CASE DEFAULT statement(s)
ENDCASE
ENDESELECT

FOR v = start TO finish (STEP s)

statement(s)
NEXT v
REPEAT
UNTIL condition
WHILE condition
ENDWHILE
DO
statement(s)

LOOP
EXIT
REM text
.text
REMS*
REMS*
REMS*
REMS*

Keyboard

CLEAR BUFFER
CONTROLKEY(j)integer
DOWNKEY(\$)string
ESCAPEKEY(j)integer
INPUT ("prompt"), variableName
INKEY(\$)string
KEYSTATE(scanCode)integer
LEFTKEY(j)integer
RIGHTKEY(j)integer
SCANCODE(j)integer
SHIFTCODE(j)integer
SUSPEND FOR KEY
UPKEY(j)integer
WAIT KEY

Text Output

CENTER TEXT x, y, string
CLS (colour)
INK lb
PRINT value [value...|...|]
RGB(c,g,b)
RGBG(c,g,b)integer
RGBB(c,g,b)integer
SET CURSOR x,y
SET TEXT FONT font
SET TEXT OPAQUE
SET TEXT TO BOLD
SET TEXT TO ITALIC
SET TEXT TO NORMAL
SET TEXT Z-POINTS
SET TEXT TRANSPARENT
TEXT x,y
TEXT BACKGROUND TYPE(j)integer
TEXT FONT(\$)string
TEXT HEIGHT(string)integer
TEXT SIZE(j)integer
TEXT STYLE(j)integer
TEXT WIDTH(\$string)integer

Routines

FUNCTION name (param list)
ENDFUNCTION (val)
EXITFUNCTION (val)
label:
RETURN
GOSUB label

System

GET DATES(\$)string
GET TIME(\$)string
TIMER()integer

Strings

ASC(c:char)integer
BIN\$(v:integer)string
LEN\$(v:integer)string
LEFT\$(v:integer,string)
LEN\$(string)integer/string
LOWER\$(string)string
MID\$(s:string, v:integer)string
RIGHT\$(s:string, v:integer)string
SPACE\$(v:integer)string
STR\$(v:number)string
UPPER\$(s:string)string
VAL\$(s:string)number

Bitmaps

BITMAP DEPTH(hno)integer
BITMAP EXIST(hno)integer
BITMAP HEIGHT(hno)integer
BITMAP MIRRORED(hno)integer
BITMAP WIDTH(hno)integer
BLUR BITMAP bno, blur
COPY BITMAP b1,x1,y1,x2,y2,b2,x3,y3,x4,y4
CREATE BITMAP bno, w, h
CURRENT BITMAP(j)integer
DELETE BITMAP bno
FLIP BITMAP bno
LOAD BITMAP file, bno
MIRROR BITMAP bno
SET CURRENT BITMAP bno

Images

DELETE IMAGE iho
GET IMAGE iho, x1, y1, x2, y2
IMAGE EXIST(iho)integer
LOAD IMAGE file, iho
PASTE IMAGE iho, X, Y
SAVE IMAGE file, iho
SET IMAGE COLORKEY red, green, blue

Maths

ABS(v)real
ACOS(v)real
ASIN(v)real
ATAN(v)real
COS(v)real
EXP(v)real
HCOS(v)real
HSIN(v)real
HTAN(v)real
SIN(v)real
TAN(v)real

2D Vectors

ADD VECTOR2 vld3, vld, vld2
COPY VECTOR2 vld2, vld
DELETE VECTOR2(vld)integer
DIVIDE VECTOR2(vld)div
DOT PRODUCT VECTOR2(vld, vld2)real
IS EQUAL VECTOR2(vld, vld2)real
LENGTH VECTOR2(vld)real
MAXIMIZE VECTOR2 vld3, vld, vld2
MINIMIZE VECTOR2 vld3, vld, vld2
MULTIPLY VECTOR2 vld, vld2
SCALE VECTOR2 vld2, vld, mult
SET VECTOR2 vld, vx, vy
SQUARED LENGTH VECTOR2(vld)real
SUBTRACT VECTOR2 vld3, vld, vld2
X VECTOR2(vld)real
Y VECTOR2(vld)real

Sprites

BACKDROP OFF
BACKDROP ON
CLONE SPRITE sno1, sno2
COLOR BACKDROP colour
CREATE ANIMATED SPRITE sno, file, r, c, iho
DELETE SPRITE sno
FLIP SPRITE sno
HIDE ALL SPRITES
HIDE SPRITE sno
MIRROR SPRITE sno
MOVE SPRITE sno, vel
OFFSET SPRITE sno, xoff, yoff
PASTE SPRITE sno, x, y
PLAY SPRITE sno
ROTATE SPRITE sno, angle
SCALE SPRITE sno, scale
SET SPRITE sno, backsave, trans
SET SPRITE ALPHA sno, alpha
SET SPRITE DIFFUSE sno, r, g, b
SET SPRITE FRAME sno, iho
SET SPRITE IMAGE sno, iho
SET SPRITE PRIORITY sno, priority
SET SPRITE TEXTURE COORD sno, vertex, U, V
SHOW ALL SPRITES
SHOW SPRITE sno
SIZE SPRITE sno, szx, ysz
SPRITE sno, x, y, iho
SPRITE ALPHA(sno)integer
SPRITE ANGLE(sno)float
SPRITE BL\$(sno)integer
SPRITE BLD\$(sno)integer
SPRITE EXIST(sno)integer
SPRITE FRAME(sno)integer
SPRITE GREEN(sno)integer
SPRITE HEIGHT(sno, sno2)integer
SPRITE HIT(sno)integer
SPRITE FLIPPED(sno)integer
SPRITE MIRRORED(sno)integer
SPRITE OFFSET X(sno)integer
SPRITE OFFSET Y(sno)integer
SPRITE RED(sno)integer
SPRITE SCALE X(sno)integer
SPRITE SCALE Y(sno)integer
SPRITE ID(sno)integer
SPRITE VISIBLE(sno)integer
SPRITE X(sno)integer
STRETCH SPRITE sno, xscale, yscale

Animation

ANIMATION EXIST(ano)integer
ANIMATION HEIGHT(ano)integer
ANIMATION LOOPING(ano)integer
ANIMATION PLAYING(ano)integer
ANIMATION POSITION Y(ano)integer
ANIMATION POSITION X(ano)integer
ANIMATION SPEED(ano)integer
ANIMATION VOLUME(ano)integer
ANIMATION WIDTH(ano)integer
DELETE ANIMATION ano
LOAD ANIMATION file, ano
LOAD DVD ANIMATION ano
LOOP ANIMATION ano
PAUSE ANIMATION ano
PLAY ANIMATION ano, x1, y1, x2, y2
PLAY ANIMATION ano
RESUME ANIMATION ano
SET ANIMATION SPEED ano, spd
SET ANIMATION VOLUME ano, vol
SET DVD CHAPTER ano, iho, cro
STOP ANIMATION ano
TOTAL DVD CHAPTERS(ano, tro)integer

Adds vld and vld2 storming result in vld3
Copies vld to vld2
Deletes vector vld and returns 1 if successful
Divides vector vld by div
Returns vld/vld2, + vld, vld2,
Returns the length of vector vld
Returns a 2D vector with vld, returns 1 if okay
Assigns the min values from vld and vld2 to vld3
Assigns the min values from vld and vld2 to vld3
Multiplies vld by mult
Returns the length of vector vld
Multiplies vld by mult and assigns result to vld2
Assigns real values vx, vy to vector vld
Returns the square of the length of vector vld
Subtracts vld2 from vld storming result in vld3
Returns the x value of vector vld
Returns the y value of vector vld

Activates the blue backdrop
Returns to yellow backdrop
Activates specified backdrop
Sprite sno2 created as copy of sno1
Sets backdrop colour
Sets backdrop colour
Sprite sno from file (v by c frames)image iho
Deletes sprite sno from RAM
Flips sprite sno
Hides all sprites
Hides sprite sno
Mirrors sprite sno
Moves sprite sno vel pixels
Moves sprite sno to origin xoff, yoff pixels
Moves sprite sno to (x,y)
Plays sprite sno frames
Plays sprite sno in both directions
Scales sprite sno in both directions
Saves sprite background(0,1), sprite transparent(0,1)
Whole sprite transparency (0 to 255 - opaque)
Sets the strength of colours (0 - off, .255 - full on)
Frame frame in animated sprite sno is displayed.
Loads a new image, iho, into sprite sno
Sets order in which sno is drawn (prty 1 = last)
Specifies how the image is mapped onto sprite
Makes sprite sno visible
Sets the dimensions of sno (szx and ysz in pixels)
Places sprite sno containing image iho at (x,y)
Returns the current alpha setting for sprite sno
Returns angle of sprite sno
Returns lue diffuse setting for sno
Returns 1 if sprite sno is visible
Returns 1 if sprite sno is visible
Returns green diffuse setting for sno
Returns frame diffusing setting for sno
Returns height in pixels of sprite sno
Returns ID of sprite hit by sno, if none, 0, returned
Returns 1 if sprite sno flipped
Returns image currently being used by sprite sno
Returns 1 if sprite sno mirrored
Returns x offset of sprite sno's origin
Returns x offset of sprite sno's origin
Returns red diffuse setting for sno
Returns scaling factor of sprite sno's width
Returns scaling factor of sprite sno's height
Returns 1 if sprite sno is visible
Returns width in pixels of sprite sno
Returns 1 if sprite sno is visible
Returns y ordinate of sprite sno's origin
Sets dimensions of sno as a % of current size

Deletes image iho from RAM
Creates image iho from bitmap area (x1,y1) to (x2,y2)
Returns 1 if image iho exists
Loads file iho into RAM and ihs it as iho
Pastes image iho with top-left at (x,y)
Saves image iho in file
Sets the transparency colour of all images
Returns absolute value of v
Returns the arc-cosine of v
Returns the arc-tangent of v
Returns the arc-sine of v
Returns the cosine of angle v
Returns e
Returns the hyperbolic cosine of v
Returns the hyperbolic sine of v
Returns the sine of v
Returns the square root of v
Returns the tangent of v

Deletes image iho from RAM
Creates image iho from bitmap area (x1,y1) to (x2,y2)
Returns 1 if image iho exists
Loads file iho into RAM and ihs it as iho
Pastes image iho with top-left at (x,y)
Saves image iho in file
Sets the transparency colour of all images
Returns absolute value of v
Returns the arc-cosine of v
Returns the arc-tangent of v
Returns the arc-sine of v
Returns the cosine of angle v
Returns e
Returns the hyperbolic cosine of v
Returns the hyperbolic sine of v
Returns the sine of v
Returns the square root of v
Returns the tangent of v

DarkBASIC Professional Statements

for the book
Hands On DarkBASIC Pro
volume 1

Book titles available:

- Hands On Pascal
- Hands On C++
- Hands On Java
- Hands On XHTML
- Hands On DarkBASIC Pro Volumes 1 and 2

Mouse			
CHANGE MOUSE cno	(681)		
HIDE MOUSE	(680)		
MOUSECLICK(i)integer	(678)		
MOUSEMOVE(X)integer	(684)		
MOUSEMOVE(Y)integer	(684)		
MOUSEMOVE(Z)integer	(683)		
MOUSEX()integer	(685)		
MUSEY()integer	(685)		
POSITION MOUSE x,y	(681)		
SHOW MOUSE	(680)		
SUSPEND FOR MOUSE	(678)		
WAIT MOUSE	(678)		
Miscellaneous			
RANDOMIZE seed	(93)		
RND(max)integer	(93)		
SLEEP m	(143)		
WAIT m	(143)		
Display			
CONTROL DISPLAY MODE	(367)		
PERFORM CHECKLIST FOR GRAPHICS CARDS (c1,c2,c3,c4)	(163)		
PERFORM CHECKLIST FOR DISPLAY MODES	(370)		
SCREEN DEPTH(i)integer	(364)		
SCREEN FPS(i)integer	(368)		
SCREEN HEIGHT(i)integer	(368)		
SCREEN INVALID(i)integer	(100)		
SCREEN VALID(i)integer	(369)		
SCREEN TYPE(i)integer	(371)		
SCREEN WIDTH(i)integer	(100)		
SET DISPLAY MODE width, height, bits	(64)		
SET GAMMA r, g, b	(372)		
SET GRAPHICS CARD cardname	(370)		
Window Display			
SHOW WINDOW	(376)		
HIDE WINDOW	(375)		
MAXIMIZE WINDOW	-		
MINIMIZE WINDOW	-		
RESTORE WINDOW	-		
SET WINDOW ON	(373)		
SET WINDOW OFF	-		
SET WINDOW SIZE width, height	(373)		
SET WINDOW TITLE title	(374)		
SET WINDOW LAYOUT style, bar, icon	(374)		
SET WINDOW POSITION x,y	(373)		
Files			
APPNAMES(i)string	(382)		
CLIP	(381)		
CLOSE FILE fn	(384)		
COPY FILE fn1,fn2	(384)		
DELETE DIRECTORY fold	(384)		
DELETE FILE fn	(385)		
DIR	(385)		
DRIVELIST	(380)		
EXECUTE FILE fn, cmd, path, (waitfig)	(388)		
FILE EXIST (fn)integer	(387)		
FILE SIZE (fn)integer	(392)		
FIND FIRST	(389)		
FIND NEXT	(389)		
GET DIRS()string	(381)		
GET FILE CREATIONS()string	(390)		
GET FILE DATES()string	(389)		
GET FILE NAMES()string	(389)		
GET FILE SIZES()string	(390)		
LINK ARRAY E()integer	(389)		
LINK ARRAY F()integer	(389)		
MAKE DIRECTORY fold	(383)		
MAKE FILE fn	(407)		
MOVE FILE fn1,fn2	(387)		
OPEN TO READ fn, fn	(393)		
OPEN TO WRITE fn, fn	(393)		
PATH EXIST(fold)integer	(383)		
PERFORM CHECKLIST FOR DRIVES	(410)		
PERFORM CHECKLIST FOR FILES	(410)		
READ type, fn, var	(398)		
READ BYTE FROM FILE (fn, post)integer	(401)		
READ DIRBLOCK fn, fold	(406)		
READ FILE fn, var	(399)		
READ FILEBLOCK fn, fn	(405)		
RENAME FILE fn1, fn2	(405)		
SET ARRAY fn, aname(i)	(408)		
SKIP BYTES fn, bytes	(400)		
WINDERS()string	(392)		
WRITE type, fn, var	(394)		
WRITE BYTE TO FILE fn, post, byte	(402)		
WRITE DIRBLOCK fn, fold	(404)		
WRITE FILE fn, v	(397)		
WRITE FILEBLOCK fn, fn	(403)		
Drawing			
BOX x1,y1,x2,y2 {c1,c2,c3,c4}	(163)		
CIRCLE x,y, rad	(164)		
DOT x,y	(160)		
ELLIPSE x,v,h,r	(165)		
LINE x1,y1,x2,y2	(162)		
POINT (x,y)integer	(161)		
Checklists			
CHECKLIST QUANTITY(i)integer	(364)		
CHECKLIST STRINGS(v)integer,string	(365)		
CHECKLIST VALUE A(v)integer,integer	(366)		
CHECKLIST VALUE B(v)integer,integer	(366)		
CHECKLIST VALUE C(v)integer,integer	(366)		
CHECKLIST VALUE D(v)integer,integer	(366)		
EMPTY CHECKLIST	(366)		
PERFORM CHECKLIST FOR CONTROL DEVICES	(716)		
PERFORM CHECKLIST FOR DISPLAY MODES	(364)		
PERFORM CHECKLIST FOR DRIVES	(410)		
PERFORM CHECKLIST FOR FILES	(410)		
PERFORM CHECKLIST FOR GRAPHIC CARDS	(370)		
Lists			
ARRAY COUNT name(i)	(309)		
ARRAY DELETE ELEMENT name(i),p	(310)		
ARRAY INDEX TO BOTTOM name(i)	(314)		
ARRAY INDEX TO TOP name(i)	(314)		
ARRAY INSERT AT BOTTOM name(i),qty	(306)		
ARRAY INSERT AT ELEMENT name(i),p	(308)		
ARRAY INSERT AT TOP name(i),qty	(308)		
EMPTY ARRAY name(i)	(310)		
NEXT ARRAY INDEX name(i)	(311)		
PREVIOUS ARRAY INDEX name(i)	(314)		
Queues			
ADD TO QUEUE name(i)	(318)		
ARRAY INDEX TO QUEUE name(i)	(319)		
REMOVE FROM QUEUE name(i)	(319)		
Stacks			
ADD TO STACK name(i)	(320)		
ARRAY INDEX TO STACK name(i)	(321)		
REMOVE FROM STACK	(321)		
Joystick			
FORCE ANGLE v, angle, time	(727)		
FORCE AUTO CENTER OFF	(728)		
FORCE AUTO CENTER ON	(728)		
FORCE CHAINSAW mag, time	(729)		
FORCE DOWN mag	(726)		
FORCE IMPACT mag, time	(731)		
FORCE LEFT mag	(726)		
FORCE NO EFFECT	(728)		
FORCE RIGHT mag	(726)		
FORCE STOP mag, time	(728)		
FORCE UP mag	(726)		
FORCE WATER EFFECT mag, time	(728)		
JOYSTICK UP()integer	(717)		
JOYSTICK DOWN()integer	(717)		
JOYSTICK LEFT()integer	(717)		
JOYSTICK RIGHT()integer	(717)		
JOYSTICK X()integer	(718)		
JOYSTICK Y()integer	(718)		
JOYSTICK Z()integer	(718)		
JOYSTICK FIRE A()integer	(721)		
JOYSTICK FIRE B()integer	(721)		
JOYSTICK FIRE D()integer	(721)		
JOYSTICK FIRE Xb()integer	(721)		
JOYSTICK FIRE Yb()integer	(722)		
JOYSTICK SLIDER A()integer	(723)		
JOYSTICK SLIDER C()integer	(723)		
JOYSTICK SLIDER D()integer	(723)		
JOYSTICK TWIST X()integer	(723)		
JOYSTICK TWIST Y()integer	(723)		
JOYSTICK TWIST Z()integer	(723)		
JOYSTICK HAT ANGLE(h)integer	(724)		
PERFORM CHECKLIST FOR CONTROL DEVICES	(716)		
Control Devices			
ANGLE	(163)		
ANGLE (x,y)	(164)		
ANGLE (x,y) radius, rad	(165)		
ANGLE (x,y) horz., rad = hr; vert rad = vr	(162)		
ANGLE (x,y)	(161)		
ANGLE (x,y) radius, rad	(364)		
ANGLE (x,y) horz., rad = hr; vert rad = vr	(365)		
ANGLE (x,y)	(366)		
ANGLE (x,y) horz., rad = hr; vert rad = vr	(366)		
ANGLE (x,y)	(366)		
ANGLE (x,y) horz., rad = hr; vert rad = vr	(366)		
ANGLE (x,y)	(716)		
ANGLE (x,y) horz., rad = hr; vert rad = vr	(717)		
ANGLE (x,y)	(717)		
ANGLE (x,y) horz., rad = hr; vert rad = vr	(718)		
ANGLE (x,y)	(718)		
ANGLE (x,y) horz., rad = hr; vert rad = vr	(718)		
ANGLE (x,y)	(721)		
ANGLE (x,y) horz., rad = hr; vert rad = vr	(721)		
ANGLE (x,y)	(721)		
ANGLE (x,y) horz., rad = hr; vert rad = vr	(722)		
ANGLE (x,y)	(723)		
ANGLE (x,y) horz., rad = hr; vert rad = vr	(723)		
ANGLE (x,y)	(723)		
ANGLE (x,y) horz., rad = hr; vert rad = vr	(723)		
ANGLE (x,y)	(724)		
ANGLE (x,y) horz., rad = hr; vert rad = vr	(724)		