

AGK 2 BASIC Statements and Functions

(ver 0.1)

(from Hands On AGK BASIC 2    www.digital-skills.co.uk)

Comments

Comments are added to aid human readability and document aspects of the program.

rem	075	Remainder of line is comment
//	075	Remainder of line is comment
`	075	Remainder of line is comment
remstart ... remend	075	Lines between the two terms are comments

Constants

Any fixed value used within a program is known as a constant or literal. A name can be assigned to a constant. The value assigned to a named constant cannot be modified later in the program code.

<b>Declaring named constants:</b>		
#constant name [=] value	103	Assigns a name to a constant \$ and # not allowed

Single quotes strings	114	Allows escape characters to be included in string
-----------------------	-----	---

Variables

Variables may be declared explicitly or used without declaration. Undeclared variables must have a special final symbol if storing real or string values. Declared variables do not require any special symbol.

<b>Undeclared Variable name endings:</b>	
#            float	100
\$            string	100
no symbol   integer	099

<b>Declared Variables:</b>		
name as type	102	Gives the variable's name and type

<b>Recognised types:</b>	
integer	102
float	102
string	102

Core Statements

var = val (assignment)	106	Assigns value <i>va</i> / to variable <i>var</i>
continue	174	Skips the remainder of the current iteration of a loop
do...loop	176	Infinite loop structure
exit	172	Loop structure exit
for...next	165	Counting loop structure
if...endif	138	Condition-dependant action
if...else...endif	146	Alternative condition-dependant actions
if...elseif...else..endif	150	Multi-way selection structure
if...then...else	147	Alternative if structure
repeat...until	164	Exit-controlled loop structure
select...endselect	150	Multi-way selection structure
variable = value	106	Assigns a value to a variable
while...endwhile	162	Entrance-controlled loop structure

Arithmetic, Logic and String Operators and Functions

+ - * / ^	107	Addition, subtraction, division, multiplication, power
+	113	Join strings
and	142	Combines Boolean expressions
dec var [,v1]	111	Subtracts <i>v1</i> to <i>var</i> . If <i>v1</i> omitted, subtracts 1
inc var [,v1]	111	Adds <i>v1</i> to <i>var</i> . If <i>v1</i> omitted, adds 1
Mod(v1,v2):int	112	Returns the integer remainder after dividing <i>v1</i> by <i>v2</i>
not	145	Returns the complement of a Boolean expression
or	142	Combines Boolean expressions

IDE Commands

#include "file"	127	Includes source file in the program code.
#option_explicit	102	Forces all variables used to be declared explicitly

Processor Control

Sleep(m)	120	Causes the program to halt for <i>m</i> milliseconds
----------	-----	--

Device Input

Because AGK programs run on many platforms most (but not all) input commands work with non-specific devices. For example, the pointer commands reads from mouse when running on a desktop but from the touch screen when used on a smart device.

<b>Pointer Commands</b>		
GetPointerPressed():int	217	Returns 1 if mouse pressed or screen touched, else 0
GetPointerReleased():int	217	Returns 1 if pointer released, else 0
GetPointerState():int	218	Returns 1 if pointer currently pressed, else 0
GetPointerX():float	218	Returns the x coordinate of the pointer
GetPointerY():float	218	Returns the y coordinate of the pointer

Dialog Boxes

A dialog box is a message that appears within a separate window. The window will remain until the OK button within the window is pressed.

Message(m\$)	082	Displays <i>m\$</i> in a separate window.
--------------	-----	---

Display Output

The output statements here are used to display values and draw basic geometric shapes. Output produced is not persistent, which is to say that the output statements must be repeated between calls to Syncn().

DrawBox(x1#,y1#,x2#,y2#,c1,c2,c3,c4,f)	196	Draws a rectangle (x1#,y1#) is top-left, (x2#,y2#) is bottom-right. Colours c1,c2,c3,c4 from each corner. f = 1: filled box; f=0: outline box
DrawEllipse(x#,y#,xr#,yr#,c1,c2,f)	196	Draws an ellipse. Centre (x#,y#). c1:top fill colour. c2: bottom fill colour. f = 1: filled ellipse; f=0: outline ellipse
DrawLine(x1#,y1#,x2#,y2#,r,g,b)	195	Draws line from (x1#,y1#) to (x2#,y2#) using colour (r,g,b)
DrawLine(x1#,y1#,x2#,y2#,c1,c2)	195	Draws line from (x1#,y1#) to (x2#,y2#) change colour from c1 to c2
GetColorBlue(c):int	194	Returns the blue component of colour value c
GetColorGreen(c):int	194	Returns the green component of colour value c
GetColorRed(c):int	194	Returns the red component of colour value c
MakeColor(r,g,b):int	194	Returns colour value constructed from r,g and b (used in Draw...() functions)
Print(v)	076	Sends v to back buffer adding a newline character(use Syncn() to display)
PrintC(v)	077	Sends v to back buffer (no newline)
SetPrintColor(r,g,b,tr)	078	Sets the colour used when displaying text to (r,g,b) with tr transparency.
SetPrintSize(sz)	080	Sets size of text to sz
SetPrintSpacing(sp)	080	Sets spacing between characters to sz
Syncn()	117	Updates the contents of the screen (and other tasks)

Images

Images can be PNG, BMP or JPG. Images are displayed by being linked to a sprite after having been loaded. Images must be stored in the project's media folder. Images, like other resources, are assigned an ID. No two images can be assigned the same ID. IDs values can be selected by the programmer or by the program.

DeleteAllImages()	215	Deletes all images
DeletImage(id)	215	Deletes image <i>id</i>
LoadImage(id,file)	204	Loads image from <i>file</i> and assigns it the ID <i>id</i>
LoadImage(file):int	204	Loads image from <i>file</i> ; returns an allocated ID

Random Numbers

These commands allow a pseudo random number within a specified range to be generated.

Random([s,f])	122	Generates random value in range s to f. If parameters omitted, generates a number in the range 1 to 65,535
Random2([s,f])	124	Generates random number in range s to f. If parameters omitted, generates a number in the range -2,147,483,648 to 2,147,483,647
RandomSign(v):int	125	Returns v in positive form 50% of time, else returns -v
SetRandomSeed(s)	123	Sets seed for Random() to s
SetRandomSeed2(s)	124	Sets seed for Random2() to s

Sprites

A sprite is a visual element which can be moved, resized, rotated and made invisible. A sprite normally displays a previously loaded image. The image can be sprite into several frames to create an animated sprite.

CloneSprite(id,idorg)	213	Creates sprite <i>id</i> as an exact copy of sprite <i>idorg</i>
CreateSprite(id,img)	206	Creates sprite with ID <i>id</i> from image with ID <i>img</i>
CreateSprite(img):int	206	Creates sprite from image <i>img</i> ; returns the ID allocated
DeleteAllSprites()	215	Deletes all sprites
DeleteSprite(id)	215	Deletes sprite <i>id</i>
GetSpriteDepth(id):int	213	Returns the layer of sprite <i>id</i>
GetSpriteHit(x#,y#)	220	Returns ID of sprite under (x#,y#)
GetSpriteVisible(id):int	214	Returns 1 if sprite <i>id</i> visible, else 0
GetSpriteX()	210	Returns the x coordinate of sprite
GetSpriteY()	210	Returns y coordinate of sprite
SetSortCreated(f)	211	Ensures (f=1) sprites on layer drawn in creation order
SetSpriteDepth(id,d)	211	Sets sprite layer (default 10)
SetSpritePosition(id,x#,y#)	209	Positions sprite <i>id</i> top-left corner at (x#,y#)
SetSpriteSize(id,w#,h#)	208	Sizes sprite <i>id</i> to width <i>w#</i> and height <i>h#</i> (use a -1 value for one dimension to have it automatically calculated)
SetSpriteVisible(id,f)	214	Sets sprite <i>id</i> 's visibility(f=1:visible; f=0:invisible)

Text

A text resources allows text to be placed on the screen in the same manner as a sprite. Text, size, colour, position and rotation can be specified. Text is produced from a default image. This image can be replaced for all or specific text allowing a different font to be used.

CreateText(id,str)	222	Creates text with ID <i>id</i> containing the string <i>str</i>
CreateText(str):int	222	Creates text containing <i>str</i> ; returns the ID generated
DeleteText(id)	225	Deletes text <i>id</i>
DeleteAllText()	225	Deletes all text resources
GetTextVisible(id):int	225	Returns the visibility of sprite <i>id</i> (1:visible; 0:invisible)
SetTextColor(id,r,g,b,op)	223	Set colour for text <i>id</i> to (r,g,b) with opacity set to <i>op</i>
SetTextPosition(id,x#,y#)	223	Set text <i>id</i> 's top-left corner to (x#,y#)
SetTextSize(id,sz#)	224	Sets text <i>id</i> 's height to sz
SetTextString(id,str)	224	Sets text <i>id</i> 's contents to <i>str</i>
SetTextVisible(id,f)	225	Sets visibility of text <i>id</i> (f=1: visible; f=0:invisible)

Time and Date Functions

GetMilliSeconds():int	119	Returns time app has been running in milliseconds
GetSeconds():int	120	Returns time app has been running to nearest second
ResetTimer()	119	Resets the app timer to zero
Timer():float	115	Returns the time app has been running in seconds

Window and Screen Setup

When run on a desktop machine, an AGK app will normally run within a window (though it can be made full screen).

On a mobile device the app will take up the maximum possible area of the screen. In order to handle screens of varying resolutions, screen positions are either defined using the default percentage system or virtual pixels. The visual elements appearing on a screen can be made to rotate in synchrony with a mobile device.

ClearScreen()	081	Clears screen to the colour specified by SetClearColor()
GetDeviceHeight()	193	Returns the window height (screen height on mobile)
GetDeviceWidth()	193	Returns the window width (screen width on mobile)
GetMaxDeviceHeight()	192	Returns the height of screen in pixels
GetMaxDeviceWidth()	192	Returns the width of screen in pixels
SetBorderColor(r,g,b)	089	Sets fill colour for unused areas of screen where the device being used as different aspect ratio to that given
SetClearColor(r,g,b)	081	Sets the colour to be used when the screen is cleared
SetDisplayAspect(r#)	089	Sets the assumed width to height ratio of the screen.
SetOrientationAllowed(p,ip,ll,rl)	085	Adjust orientation of image for selected orientations 1: allowed, 0: not allowed. Order portrait, inverted portrait left landscape, right landscape (mobile devices)
SetWindowTitle(m\$)	084	Text <i>m\$</i> appears in window title bar (PC and Mac)
SetVirtualResolution(w,h)	088	Sets the assumed resolution to w pixels wide by h pixels high. The true resolution may be different.

