Blassic: keywords.

Last revision: 12-feb-2005

a b c d e f g h i j k l m n o p q r s t u v w x y z

Square brackets represent that the part surrounded is optional.

Several elements separated by | means that you may choose one of the elements.

ABS
: Function. Returns the absolute value of the parameter.
`PRINT ABS (-1)`

ACOS
: Function. Returns the arc cosine of the parameter.

ACOSH
: Function. Returns the hyperbolic arc cosine of the parameter.

AFTER
: See SYMBOL AFTER.

ALLOC_MEMORY
: Function: `ALLOC_MEMORY ( n )` Allocates a block of memory of size n and return his address. The block of memory is readable, writable and executable if the operating system allows it. See the sample program machinecode.bas for an example of use.

AND
: Operator. Does a integer or logical (depending on the state of the system flags) binary and between its arguments.

APPEND
: See OPEN.

AS
: See OPEN, SOCKET, FIELD and NAME.

ASC
: Function. Returns the code of the first char of the argument, 0 if the string is empty. The name comes from the ASCII character set, but the code can be outside the ascii range (0-127).

ASIN
: Function. Returns the arc sine value of the parameter.

ASINH
: Function. Returns the hyperbolic arc sine of the parameter.

AT
: PRINT modifier. `PRINT ... AT y, x ...` Place the cursor in the x column of the line y. Contrary to the other cursor movement instructions, the upper left of the screen is 0, 0.

ATAN
: Function. Returns the arc tangent of the arguemnt.

ATAN2
: Function. `ATAN (y, x)` returns the arc between the x axis and the line (0, 0) - (x,y).

ATANH
: Function. Returns the hyperbolic arc tangent of the argument.

AUTO
: `AUTO [initial number] [, increment]` Enter the auto numbering mode. The default values are 10 and 10 but can be modified via the system vars AutoInit and AutoInc or with the -a command line option.

BEEP

Makes a beep.

BIN$

Function. `BIN$ (number [,c] )` Returns a string of binary digits with the value of number, if the c argument is present and the length of the result is shorter than c is filled with zeroes.

BINARY

Mode specification for OPEN

BREAK

See `ON BREAK`.

BRIGTH

Not documented yet.

CHAIN

`CHAIN filaname$ [ , line ]` Loads a program in memory and begin execution from the beginning or from *line*. Variables are not cleared.

`CHAIN MERGE filename$ [ , line ] [ , DELETE range ]` Merge a program with the present in memory and begins executing the resulting program from the beginning or from *line*. If the DELETE part is present the range of lines is deleted before the merge, *range* can be a single line or in the form initial-final where initial and final are line numbers and can be omitted.

CHDIR

Changes current directory.

CHR$

Function. Returns a string of lenght 1 with the char that has the argument as code.

CINT

Function. Returns the argument rounded to integer.

CIRCLE

`CIRCLE (x, y), r, [ color ], [ initial ], [ final ] [ , e ]` Draws a circumference centered in *(x, y)* with radius *r*, setting the graphics pen to *color* beginning in angle 0 or in *initial* and ending in 2 * PI radians or in *final*. If *e* is present an ellipse is drawed instead, if *e* is greater than 1 the x axis is divided by *e*, else the y axis is multiplied by it.

CLEAR

`CLEAR` Clear all variables and arrays, close all opened files, erase user defined functions, set trigonometric mode to radians, forget pending loops.

`CLEAR INK` Restore all inks to its default values.

`CLEAR INPUT` Discards all key and mouse button events actually buffered.

CLOSE

Not documented yet.

CLS

In text mode: `CLS` Clear the screen.

In graphics mode `CLS [ #window ]` Clear the text window specified (0 by default).

See also GRAPHICS CLS.

COLOR

Not documented yet.

CONT

Continue execution previously stopped by a STOP instruction or a break.

COPYCHR$

Function. `COPYCHR$ (#w)` return the character present at the current cursor position in window w, or the empty string if the character can't be recognized.

COS

Function. Returns the cosine value or the argument.

COSH

Function. Returns the hyperbolic cosine value of the argument.

CVD

Not documented yet.

CVDMBF

Not documented yet.

CVI

Not documented yet.

CVL

Not documented yet.

CVS

Not documented yet.

CVSMBF

Not documented yet.

DATA

Not documented yet.

DATE$

Function without arguments. Returns a string with the current date in the format mm-dd-yyyy.

DEC$

Function `DEC$ ( number, format)` Returns a string representing the *number* using the specified *format*, that is a format specifier like the used in PRINT USING limited to numeric formats that represent a single number.

DEF

`DEF DBL | INT | REAL | SNG | STR letter-specifier.` Define that the variables beginning by a letter specified in the *letter-specifier* are of type: DBL, REAL, SNG ---> floating point, INT ---> integer, STR ---> string. The various types of floating point are admitted for compatibility with other Basics.
The *letter-specifier* is a comma separated list of single letter of ranges of letters.

`DEF FN` Not documented yet.

DEFDBL

Same as DEF REAL

DEFINT

Same as DEF INT

DEFREAL

Same as DEF REAL

DEFSNG

Same as DEF REAL

DEFSTR

Same as DEF STR

DELETE

Not documented yet.

DEG

Set the trigonometric mode to degrees.

DELIMITER

Not documented yet.

DIM

Not documented yet.

DRAW

Not documented yet.

DRAWARC
: Not documented yet.

DRAWR
: Not documented yet.

EDIT
: Not documented yet.

ELSE
: Not documented yet.

END
: Not documented yet.

ENVIRON
: Not documented yet.

ENVIRON$
: Not documented yet.

EOF
: Not documented yet.

ERASE
: Not documented yet.

ERL
: Function without parameters. Returns the program line where the last error was generated, used normally inside an ON ERROR GOTO routine.

ERR
: Function without parameters. Returns the code of the last error generated, used normally inside an ON ERROR GOTO routine.

ERROR
: `ERROR n` Forces the error number n, stopping the program or jumping to the position established by an ON ERROR GOTO instruction. For a list of the error numbers used by blassic refer to the error.h file in the sources.

EXIT
: `EXIT [n]` Quit blassic returning to the operating system. The n value, 0 if not specified, is used as the return value of the blassic command (the ERRORLEVEL value in Windows).

EXP
: Function. Returns the exponential of the argument, that is, e (the base of natural logarithms) raised to that power.

FIELD
: Not documented yet.

FIELD APPEND
: Not documented yet.

FIELD CLEAR
: Not documented yet.

FILES
: `FILES` List files in current directory.
`FILES [ #channel , ] "mask"` List files in current directory that matches mask. The wildcards allowed are operating system dependant. The listing is directed to the channel specified, 0 by default, that can be a window or a file.

FIND_FIRST_NOT_OF
: Function. As FIND_FIRST_OF but with characters that are not in char$.

FIND_FIRST_OF
: Funtion. `FIND_FIRST_OF (searched$, char$)` Returns the first position of searched that contains a character of char$, or 0 if none is found.

FIND_LAST_NOT_OF
: Function. As FIND_LAST_OF, but with characters thar are not in char$.

FIND_LAST_OF

Fuction. As FIND_FIRST_OF but returns the last position.

FINDFIRST$

Function. `FINDFIRST$ (mask$)` Returns the first file that matches wiht mask$ or the empty string if there is no match . The wildcards allowed and his interpertation are operating system dependant.

FINDNEXT$

Function whitout parameters. Returns the next file that matches with the mask$ used in the las FINDFISRT$ executed, or the empty string if there is no more matches.

FIX

Not documented yet.

FN

Not documented yet.

FOR

Not documented yet.

FREE_MEMORY

Deallocates the memory at the address sppecified. The addres must have been obtained by a previous ALLOC_MEMORY function, otherwise the result is undefined.

FREEFILE

Function. Returns the lower free channel number.

GET

Not documented yet.

GOSUB

`GOSUB line_specification` Calls the subroutine located at line_specification, that can be a line number or a valid LABEL. The control will return to the actual point when a RETURN is executed. See also ON GOSUB.

GOTO

`GOTO line_specification` Transfer the control to the position line_specification, tnat can be a line number or a valid LABEL. See also ON GOTO.

GRAPHICS

Not documented yet.

HEX$

Function. `HEX$ (number [ , width] )` Returns the hexadecimal representation of number as string. If width is specified and is too great the result will be filed with leading zeros, if is too small the result is NOT shortened.

IF

Not documented yet.

IF_DEBUG

`IF_DEBUG n` If debug level is greater or equal than n the rest of the line is executed, otherwise is ignored.
See the --debug command line option and the DebugLevel system var.

INK

`INK n` Same as `PEN n`

`INK n, color [, second]` Define the ink number n as the color, from the tabla of Amstrad CPC compatible colors. The optional second parameter is accepted for compatibility with the CPC but is ignored.

`INK n, r, g, b` Define the ink number n as the color from the rgb componentes, r, g and b between 0 and 255.

PRINT modifier: `INK n` Same as standalone instruction, but the efects remains only until the actual PRINT ends.

INKEY

Not documented yet.

INKEY$

Not documented yet.

INPUT

Not documented yet.

INPUT$
: Not documented yet.

INSTR
: Function. `INSTR ( [start_pos , ] searched_string$, string_to_find$)` Returns the position of the first occurrence of sting_to_find$ in searched_string$, or 0 if not present.The position at wich to start can be specified as start_pos.

INT
: Not documented yet.

INVERSE
: Not documented yet.

KILL
: Delete a file.

LABEL
: `LABEL identifier` Define a label in the program, then instructions that normally uses a line number as parameter, such as GOTO, can use the indetifier instead. LABEL must be the first instruction in a line, otherwise is ignored.

LCASE$
: Same as LOWER$

LEFT$
: Function. `LEFT$ (string$, n)` Returns the first n charaters from string$, or string$ if n is greater than LEN(string$).

LEN
: Function. `LEN (string$)` return the length of the string.
Note for C programmers: The total length of the string is returned, including possible NUL characters contained.

LET
: Not documented yet.

LINE
: Not documented yet.

LIST
: Not documented yet.

LLIST
: Same as LIST but the default channel is directed to the printer output.

LOAD
: `LOAD "program"` Load a blassic program. If program is not found program.blc and program.bas are tried.
`LOAD "filename", var$` Load a file into a string var.
`LOAD "filename", address [, size]` Load a file in the address of memory specified. If size is given loads only that size, otherwise the entire file.

LOC
: Function. Returns the current position in the channel number specified, 0 if no read or write operation has been done yet, that must be a file opened in random mode.

LOCAL
: Not documented yet.

LOCATE
: `LOCATE [ #channel , ] a, b` Move the cursor to the position specified with the coordinates a, b. The screen origin is 1, 1. Channel must be a graphics window in graphics mode or the console output in text mode, if omitted channel 0 is used. The order of the coordinates is dependant of the system variable Flags1, by default Microsoft style (row, col) is used.

LOF
: Function. Returns the length of the file associated with the channel number specified.

LOG

Function. Returns the natural logarithm of the argument.

LOG10

Function. Returns the decimal logarithm of the argument.

LOWER$

Function. Returns the string argument converted to lower case.

LPRINT

Same as PRINT but the default channel is directed to the printer output.

LSET

Not documented yet.

LTRIM$

Function. Left trim. Returns the string argument without possible initial speces.

MASK

Not documented yet.

MAX

Function. `MAX (list of numeric expresssions)` Returns the maximum of the list.

MID$

Function: `MID$ (string$, pos [ , length ] )` Returns the substring of string$ that begins at pos and has length characters or to the end of string$ if length is not specified or is too long.

Instruction: `MID$ (string$, pos [, length] )= stringexpr$` Replaces the substring of string$ specified as in MID$ function with stringexpr$.

MIN

Function. `MIN (list of numeric expressions)` Returns the minimum value of the list.

MKD$

Not documented yet.

MKDIR

Create a directory.

MKDMBF$

Not documented yet.

MKI$

Not documented yet.

MKL$

Not documented yet.

MKS$

Not documented yet.

MKSMBF$

Not documented yet.

MOD

Operator. `a MOD b` returns the remainder of a / b.

MODE

Not documented yet.

MOVE

Not documented yet.

MOVER

Not documented yet.

NAME

`NAME file1$ AS file2$` Renames the file named file1$ as file2$.

NEW

Not documented yet.

NEXT

Not documented yet.

NOT

Operator. Return the binary not of the argument taken as integer.

OCT$

Function. `OCT$ (number [ , width ] )` Returns the octal representation of number as string. If width is specified and is too great the result will be filled with leading zeros, if is too samll the result is NOT shortened.

ON

See ON BREAK, ON ERROR GOTO, ON ... GOTO and ON ... GOSUB.

ON BREAK

`ON BREAK CONT` Cancel the effect of ctrl-c of stopping the program.

`ON BREAK GOSUB n` When ctrl-c is pressed the program whill jump to the subroutine specified. n can be a line number or a label.

`ON BREAK STOP` Disables the effect of ON BREAK CONT and ON BREAK GOSUB, ctrl-c will interrupt the program.

ON ERROR GOTO

`ON ERROR GOTO n` Jumps to the specified line when an error is detected in the program. n can be a line number or a label. See RESUME, ERR, ERL.

ON ... GOTO

Not documented yet.

ON ... GOSUB

Not documented yet.

OPEN

Not documented yet.

OR

Operator. Does a integer or logical (depending on the state of the system flags) binary or between its arguments.

ORIGIN

Not documented yet.

OSFAMILY$

Function without parameters. Returns the type of the operating system under which Blassic is used. Actually only two familys are defined, "windows" and "unix". The value will be unix under Windows if the version od Blassic was compiled with Cygwin.

OSNAME$

Function without parameters. Returns the name of the operating system under wich Blassic is used. The value will be "Windows" under Windows (except if compiled under Cygwin) and the value returned from a *uname* system call otherwise.

OUTPUT

Not documented yet.

PAINT

Not documented yet.

PAPER

Not documented yet.

PAUSE

Not documented yet.

PEEK

Not documented yet.

PEEK16

Not documented yet.

PEEK32

Not documented yet.

PEN
: Not documented yet.

PI
: Function without parameters. Returns the value of the pi number.

PLEASE
: Undocumented instruction, please do not use.

PLOT
: Not documented yet.

PLOTR
: Not documented yet.

POKE
: Not documented yet.

POKE16
: Not documented yet.

POKE32
: Not documented yet.

POP
: Discard the last value of the GOSUB/RETURN stack. Variables defined as LOCAL are restored to his previous values.

POPEN
: Not documented yet.

POS
: Function. `POS (#channel)` return the horizontal position of the cursos in that channel. Only screen channels are allowed.

PRINT
: Not documented yet.

PROGRAMARG$
: Function: `PROGRAMARG$ (n)` returns the n command line parameter with blassic is calles, excluded the blassic command and the program name, or the empty string if there is no sufficient parameters.
  Instruction: `PROGRAMARG$ arg1$ [, args$ ....]` Establishes the return values of the PROGRAMARG$ function to the args specified. Useful when debugging.

PROGRAMPTR
: Funtion without parameters. Returns the address of the program memory area, or 0 if there is no program. See the autoview.bas and automod.bas sample programs.

PULL
: Drops the las element of the REPEAT / UNTIL stack.

PULL FOR
: Drops the las element of the FOR / NEXT stack.

PULL GOSUB
: Same as POP.

PULL REPEAT
: Same as PULL.

PULL WHILE
: Drops the las element of the WHILE / WEND stack.

PUT
: Not documented yet.

RAD
: Set trigonometric mode to radians.

RANDOM
: Not documented yet.

RANDOMIZE
: Not documented yet.

READ
: Not documented yet.

REAL
: See DEF REAL.

REGEXP_INSTR
: Function: `REGEXP_INSTR ( [ pos , ] searched$, regexp$ [ , flags] )` Returns the position of the first match of the regexp$ regular expression in searched$, or 0 if no match. If pos is specified the regexp matching begins at that position. For the values of the flag parameter see REGEXP_REPLACE$.

REGEXP_REPLACE$
: Function: `REGEXP_REPLACE$ ( [ pos, ] searched$, regexp$, rep [, flags] )` Returns the string searched$ with the matches of the regular expression regexp$ replaced by the rep parameter, that can be a string or a FN. In the former case, the $n contained in the string are replaced by the subexpressions matched, in the later the FN function is called with the subexpressions as arguments. The bits on the flag affect the result in the follwing mode: bit 0 sets nocase mode, bit 1 set nobegin mode (the string is treated as no located as the begin in a line), bit 2 set noend mode (the string is treated as no located at the end of a line), bit 3 sets the newline mode (the 'any charater' wildcards not match a newline character) and the other bits are reserved for future use, must be 0. If pos is specified the regexp matching begins at that position.

REM
: Remark. The rest of the line is ignored.

RENUM
: Not documented yet.

REPEAT
: Not documented yet.

RESTORE
: Not documented yet.

RESUME
: `RESUME [NEXT | n]` On an ON ERROR GOTO routine, returns to the statement that produced the error, to the next statement, or to the postion specified by n, that can be a number line or a label.

RETURN
: Continues execition after the point of the last GOSUB. Variables defined with LOCAL are restored to his previous values.

RIGHT$
: Function. `RIGHT$ (string$, n)` Returns the last n charaters from string$, or string$ if n is greater than LEN(string$).

RINSTR
: Function. Like INSTR but from end ot string to beginning.

RMDIR
: Not documented yet.

RND
: Not documented yet.

ROUND
: Not documented yet.

RSET
: Not documented yet.

RTRIM$
: Function. Right trim. Returns the strig argument without possible final spaces.

RUN

Not documented yet.

SAVE

`SAVE "program"` Saves the program in Blassic binary format. The recommended extension is .blc, but is not added automatically.

`SAVE "program", A` Saves the program in text format. The recommended extension is .bas but is not added automatically.

`SAVE "filename", B, var$` Save the content of the string var$ into filename.

`SAVE "filename", B, address, length` Save length bytes of memory beginning at address.

SCREEN$

Not documented yet.

SCROLL

`SCROLL [#window ,] [n]` Scroll the graphics window specified by window, 0 by default, the number of lines specified by n, 1 by default. If n is positive scrolls up and place the cursor at the beginning of the last line, if negative scrolls down and place the cursor at the beginning of the screen.

SET_TITLE

Set the title of the window. In graphics mode, the title of the graphics window is affected. In text mode, in windows the console window is affected, in unix if the terminal type is xterm, aixterm, dtterm, iris-ansi, sun-cmd or hpterm the title of the terminal window is affected, otherwise the instruction has no effect.

SGN

Function. Returns -1, 0 or 1 if the argument is negative, 0 or positive, respectively.

SHELL

`SHELL command$` Executes command$ as a shell command.

SIN

Function. Returns the sine value or the argument.

SINH

Function. Returns the hyperbolic sine value or the argument.

SOCKET

`SOCKET "host", port AS #channel` Open a socket and connect it to the host specified using the tcp port number, and associates it to the input/output channel.

SPC

PRINT modifier. Prints the number of spaces indicated by the arguemnt.

SPACE$

Funtion. Returns a string composed of the number of spaces indicated by the argument.

SQR

Function. Returns the square root of the argument.

STEP

Not documented yet.

STOP

Stops execution. See CONT.

STR

See DEF STR.

STR$

Not documented yet.

STRING$

Not documented yet.

SWAP

Not documented yet.

SYMBOL

`SYMBOL n, b1, b2, b3, b4, b5, b6, b7, b8` Define the char number n with 8 bytes of data, each bit

corresponding to one pixel. b1 to b8 can be omitted, 0 is applied in that case.

SYMBOL AFTER

`SYMBOL AFTER n` Set the number of permissible user defined characters. Characteres with values from n to 255 become definables. With 0 all chars are definables, with 256 none. All characters are reset to their default values.

SYNCHRONIZE

Not documented yet.

SYSTEM

`SYSTEM` Quits blassic returning to the operating system.

SYSVARPTR

Function whitout parameters. Returns the address of the system vars memory area. For information on the system vars refer to the sysvar.h file in the sources.

TAB

PRINT modifier. `TAB n` Prints spaces until the horizontal postition of printing reaches n. If n is greater than the current position, before doing this spaces are printed until reached the end of the line if the current channel is a grpahic window, then a line end is emitted..

TAG

Not documented yet.

TAGOFF

Not documented yet.

TAN

Not documented yet.

TANH

Not documented yet.

TEST

Function. `TEST (x, y)` moves the graphic cursor to x, y and returns the ink number corresponding to the color of that point.

TESTR

Function. Same as TEST but the cursor movement is relative to the current position.

THEN

Not documented yet.

TIME

Not documented yet.

TIME$

Not documented yet.

TO

Not documented yet.

TRIM$

Function. Returns the string argument without possible initial and final spaces.

TROFF

Set trace mode to off. See TRON.

TRON

Set trace mode to on and stablishes the type of trace.

`TRON [ LINE ] [ #channel ]`

When the trace mode is on, information is printed about the current line before executing it. Whithou LINE the information printed is `[line-number]` (The [ ] are literal in this case); with LINE specified the entire line is printed. The information is printed in the channel specified, 0 if omitted.

UCASE$

Same as LOWER$

UNTIL

Not documented yet.

UPPER$

Function. Returns the string argument converted to upper case.

USING

Not documented yet.

USR

Not documented yet.

VAL

Function. Returns the value of the first characters of the string argument that are a valid number according to the syntax of Blassic (0 if nome), or the entire string evaluated as a Blassic numeric expression, depending of the value of the system variable TypeOfVal.

VAL$

Function. Returns the value of the string argument evaluated as a Blassic string expression.

VARPTR

Not documented yet.

VPOS

Function. `VPOS (#chanel)` return the vertical position of the cursor in channel. Only screen channels in graphics mode are allowed.

WEND

Not documented yet.

WIDTH

`WIDTH LPRINT n [, m]` Stablishes the width of the printer output, when the width is exceeded a line separator is emitted (see the PrinterLine system variable. A value of 0 means unlimitted width. The value m if specified set the left margin for the printer in chars.

WHILE

Not documented yet.

WINDOW

Not documented yet.

WINDOW SWAP

Not documented yet.

WRITE

Not documented yet.

XMOUSE

Function without paramaters. Returns the x coordinate of the mouse position (only in graphics mode).

XOR

Operator. Does a integer binary xor (exclusive or) between its arguments.

XPOS

Function without parameters. Returns the x coordinate of the graphics cursor.

YMOUSE

Function without parameters. Returns the y coordinate of the mouse position (only in graphics mode).

YPOS

Function without parameters. Returns the y coordinate of the graphics cursor.

ZONE

`ZONE value` Stablishes *value* as the width of the zones used by the , PRINT delimiter. A value of 0 means 8 in the graphics window and that a tab character is used instead of spaces in printer and in text mode.

ZX_PLOT

Mimics the PLOT instruction of the ZX-81 Basic.

ZX_UNPLOT

Mimics the UNPLOT instruction of the ZX-81 Basic.

That's all folks!