CLIBASIC Language Manual

Manual symbols:

{} = choose from
[] = optional
| = choice separator
/ = continuous choice separator
… = continue

Syntax:

- Commands are separated by newlines or colons.
- Command, function, and variable names are not case sensitive.
- Variable names are allowed to contain A-Z, 0-9, #, $, %, !, ?, @, and _.
- Variables must have at least one non numeric valid character to be a valid name.
- Variables can be addressed as an array by appending a [, the element index, and a] to the end of the variable name.
- Function names are allowed to contain the same characters as variable names.
- Functions can be called by adding (, a comma separated list of arguments, and a) to the end of a function name.
- Arguments are separated by commas.
- Adding a comma with nothing after it will count as a argument but the command/function will be aware that the argument is empty.
- Strings must have one " to begin and another " to end.
- Strings can only be added together.
- Numbers can include . And 0-9.
- Logic compares must have blocks of a value, a =, >, <, >=, <=, =>, =<, or <>, and another value. These blocks must be separated by a &, or |.

Commands:

CALL FILENAME$	Opens and runs FILENAME$ in the current session. Called programs will retain the same arguments as the parent.
{CHDIR|CD} DIR$	Changes the current directory to DIR$.
CLS [COLOR]	Clears the screen with optional color.
COLOR {FGC [, BGC] | [FGC], BGC}	Sets the foreground color to FGC and the background color to BGC.
DEL {VAR$|VAR}	Deletes the variable VAR$ or VAR.
DIM {VAR$\|VAR}, MAX, {INIT$ | INIT}	Makes an array with the max index being MAX and the initial value for each element being INIT/INIT$.
FILES	List the files and directories in the current directory.
EXEC PROGRAM$ [, {ARG$/ARG}...]	Runs PROGRAM$ and passes the remaining arguments to PROGRAM$.
EXIT [CODE]	Exits with CODE (or 0 if CODE is not supplied).
LOCATE {X, [Y] | [X], Y}	Moves the cursor to X, Y.
PRINT [{STRING$/NUMBER} {,/;} ...]	Prints text on the screen, ‘;’ means print without newline and ‘,’ means print tab.
PUT [{STRING$/NUMBER} ...]	Puts STRING$ or NUMBER on the terminal.
QUIT [CODE]	Refer to ‘EXIT’.
RESETTIMER	Resets the timer.
RUN FILENAME$ [, {ARG|ARG$}...]	Runs FILENAME$ in a new session and passes ARG/ARG$ to it.
{{SET|LET} {VAR$\|VAR}, {STRING$| NUMBER}|{VAR$\|VAR} = {STRING$| NUMBER}}	Sets the variable VAR$ or VAR to STRING$ or NUMBER.
SH COMMAND$	Runs COMMAND$ in sh on Linux and Command Prompt on Windows
{SRAND|SRND} SEED	Seeds the random number generator with SEED.
WAIT SEC	Waits for SEC seconds.

WAITMS MSEC	Waits for MSEC milliseconds.
WAITUS USEC	Waits for USEC microseconds.
_AUTOCMDHIST	Enables automatic history saving (saves to '.clibasic_history' to the user's home directory, remove this file to disable this feature).
_LOADCMDHIST FILENAME$	Loads the command history from FILENAME$.
_PROMPT STRING$	Sets the prompt string to solve to STRING$.
_PROMPTTAB WIDTH	Sets the prompt tab width to WIDTH.
_RESETTITLE	Resets the terminal title.
_SAVECMDHIST FILENAME$	Saves the command history to FILENAME$.
_SHATTRIB {ATTRIB$\|ATTRIB}, {VALUE$|VALUE}	Sets the 'SH' attribute ATTRIB$ or ATTRIB to VALUE$ or VALUE.
_TITLE STRING$	Sets the terminal title to STRING$.
_TXTATTRIB {ATTRIB$\|ATTRIB}, {VALUE$, VALUE}	Sets the text attribute ATTRIB$ or ATTRIB to VALUE$ or VALUE. Available attributes: 0/"RESET" 1/"BOLD" 2/"ITALIC" 3/"UNDERLINE" 4/"DBL_UNDERLINE"/"DOUBLE_UNDERLINE" 5/"SQG_UNDERLINE"/"SQUIGGLY_UNDERLINE" 6/"STRIKETROUGH" 7/"OVERLINE" 8/"DIM" 9/"BLINK" 10/"HIDDEN" 11/"REVERSE" 12/"UNDERLINE_COLOR" 13/"FGC" 14/"BGC" 15/"TRUECOLOR"/"TRUE_COLOR"/"24BIT_COLOR"
_TXTLOCK	Stops the keyboard from echoing on the terminal.
_TXTUNLOCK	Undoes the effect of '_TXTLOCK'.

Functions:

ASC(STRING$ [, POSITION])	Returns the ASCII code of character POSITION (starting at and defaulting if not specified to zero) of STRING$.
BASENAME$(FILENAME$)	Returns the file name out of the file path provided by FILENAME$.
BGC()	Returns the current background color.
{CHDIR|CD}(DIR$)	Attempts to change the current directory to DIR$ and returns 0 on success and an error code on failure (the error code is taken directly from the C variable 'errno' set by the C 'chdir()' function).
CHR$(CODE)	Returns ASCII character CODE.
CHRAT$(STRING$, POSITION)	Returns the character at POSITION of STRING$.
CWD$()	Returns the current working directory.
CINT(NUMBER)	Returns NUMBER rounded.
COS(NUMBER)	Returns the cosine of NUMBER.
COSH(NUMBER)	Returns the hyperbolic cosine of NUMBER.
CURX()	Returns the X position of the cursor.
CURY()	Returns the Y position of the cursor.
DIRNAME$(FILENAME$)	Returns the directory name out of the file path provided by FILENAME$.
EXEC(PROGRAM$ [, {ARG$/ARG}...])	Runs PROGRAM$, passes the remaining arguments to PROGRAM$, then returns the exit code of the program or 127 if running PROGRAM$ failed.
EXEC$(PROGRAM$ [, {ARG$/ARG}...])	Runs PROGRAM$, passes the remaining arguments to PROGRAM$, then returns the output of PROGRAM$.
EXP(NUMBER)	Returns the exponent of NUMBER.
FILES$()	Returns a list of the files in the directory.
FGC()	Returns the current foreground color.

HEIGHT()	Returns the height of the terminal.
HEX$(NUMBER)	Returns the hexadecimal version of NUMBER.
INKEY$()	Returns a character from the terminal.
INPUT$(PROMPT$)	Returns a string after prompting for PROMPT$ (asks "?: " if PROMPT$ is not supplied).
INT(NUMBER)	Returns NUMBER rounded down.
LCASE$(STRING$)	Returns the lower-case version of STRING$.
LEN(STRING$)	Returns the length of STRING$.
LIMIT(NUMBER, {MAX | MIN, [MAX]})	Returns NUMBER trimmed to MIN and/or MAX.
LINE$(LINE, STRING$)	Returns line LINE (starting at zero) of STRING$.
LINES(STRING$)	Returns the line count of STRING$.
LOG(NUMBER)	Returns the natural logarithm of NUMBER.
LOG10(NUMBER)	Returns the common logarithm if NUMBER.
MOD(NUMBER)	Returns the modulus of NUMBER.
OCT$(NUMBER)	Returns the octal version of NUMBER.
PAD(STRING$/NUMBER, WIDTH [, CHAR$])	Returns STRING$/NUMBER padded to WIDTH using CHAR$. CHAR$ must contain one character, if CHAR$ is not provided then ' ' is used for STRING$ and '0' is used for NUMBER.
PI()	Returns Pi.
{RAND|RND}({MAX | MIN, MAX})	Returns a random number from MIN (0 if MIN is not supplied) to MAX.
RGB(RED, GREEN, BLUE)	Returns a 24-bit color code from separate red, green, and blue values.
SH(COMMAND$)	Runs COMMAND$ in the command line or shell and returns the exit status.
SH$(COMMAND$)	Runs COMMAND$ in the command line or shell and returns the text the command outputs.
SIN(NUMBER)	Returns the sine of NUMBER.

SINH(NUMBER)	Returns the hyperbolic sine of NUMBER.
SNIP$(STRING$, {TO | {[FROM], TO | FROM [, TO]}})	Returns the part of STRING$ defined by FROM through TO.
STR$(NUMBER)	Returns NUMBER as a string.
TAN(NUMBER)	Returns the tangent of NUMBER.
TANH(NUMBER)	Returns the hyperbolic tangent of NUMBER.
TIME()	Returns the current time in seconds.
TIMEMS()	Returns the current time in milliseconds.
TIMEUS()	Returns the current time in microseconds.
TIMER()	Returns the timer value in seconds.
TIMERMS()	Returns the timer value in milliseconds.
TIMERUS()	Returns the timer value in microseconds.
UCASE$(STRING$)	Returns the upper-case version of STRING$.
VAL(STRING$ [, TYPE])	Returns the numeric value of STRING$, TYPE is what type the number is (0 = DEC, 1 = HEX, 2 = OCT, etc/not supplied = Auto (sscanf auto-detect)).
WIDTH()	Returns the width of the terminal.
_ARGC()	Returns how many arguments where passed to the program
_ARG$([N])	Returns argument N or all arguments except #0 if N is not provided. Argument 0 is the full/real path to the program file.
_BITS$()	Returns the executable bit format.
_ENV$(STRING$)	Returns the content of the environment variable defined by STRING$.
_ENVSET(STRING$)	Returns 1 if the environment variable defined by STRING$ is set and 0 otherwise.
_HOME$()	Returns the path to the user’s home directory.
_OS$()	Returns the current operating system name.
_PROMPT$()	Returns the prompt string.

_STARTCMD$()	Returns the full/real path to the command used to start CLIBASIC.
_TXTLOCK()	Returns 1 if the text lock is in effect and 0 otherwise.
_VER$()	Returns the CLIBASIC version.

Logic Commands:

DO	Begins a DO block.
DOWHILE CONDITION	Begins a DO block while CONDITION is true.
ELSE	Inverts an IF command.
ENDIF	Ends an IF block.
FOR VAR, INIT, CONDITION, I	Begins a FOR block, sets VAR to INIT and loops while adding I to VAR while CONDITION is true.
IF CONDITION	Begins an IF block and runs commands if CONDITION is true.
LOOP	Jumps to the beginning of a DO/LOOP block.
LOOPWHILE CONDITION	Jumps to the beginning of a DO/LOOP block if CONDITION is true.
NEXT	Jumps to the beginning of a FOR block.
REM	Comments out one command.

Symbols:

?	Shortcut to PRINT.
{'|#}	Comment until the end of the line.

Comparing:

=	Equal to
<>	Not equal to
>	Greater than
<	Less than
{>=|=>}	Greater than or equal to
{<=|=<}	Less than or equal to
&	And
|	Or