

# **CLIBASIC Language Manual**

## Commands:

<code>{CHDIR CD} DIR\$</code>	Changes the current directory to <code>DIR\$</code> .
<code>CLS [COLOR]</code>	Clears the screen with optional color.
<code>COLOR {FGC, [BGC] [FGC], BGC}</code>	Sets the foreground color to <code>FGC</code> and the background color to <code>BGC</code> .
<code>DEL {VAR\$ VAR}</code>	Deletes the variable <code>VAR\$</code> or <code>VAR</code> .
<code>EXIT [CODE]</code>	Exits with <code>CODE</code> (or 0 if <code>CODE</code> is undefined).
<code>LOCATE {X, [Y] [X], Y}</code>	Moves the cursor to <code>X, Y</code> .
<code>PRINT [{STRING\$ NUMBER} {, ;} ...]</code>	Prints text on the screen, ‘;’ means print without newline and ‘,’ means print tab.
<code>PUT [{STRING\$ NUMBER} ...]</code>	Puts <code>STRING\$</code> or <code>NUMBER</code> on the terminal.
<code>QUIT [CODE]</code>	Refer to ‘EXIT’.
<code>RESETTIMER</code>	Resets the timer.
<code>RUN FILENAME\$</code>	Opens and runs <code>FILENAME\$</code> .
<code>{{SET LET} {VAR\$ VAR}, {STRING\$  NUMBER} {VAR\$ VAR} = {STRING\$  NUMBER}}</code>	Sets the variable <code>VAR\$</code> or <code>VAR</code> to <code>STRING\$</code> or <code>NUMBER</code> .
<code>{SH EXEC} COMMAND\$</code>	Runs <code>COMMAND\$</code> in <code>sh</code> on Linux and Command Prompt on Windows
<code>{SRAND SRND} SEED</code>	Seeds the random number generator with <code>SEED</code> .
<code>WAIT SEC</code>	Waits for <code>SEC</code> seconds.
<code>WAITMS MSEC</code>	Waits for <code>MSEC</code> milliseconds.
<code>WAITUS USEC</code>	Waits for <code>USEC</code> microseconds.
<code>_AUTOHIST</code>	Enables automatic history saving (saves to ‘.clibasic_history’ to the user’s home directory, remove this file to disable this feature).
<code>_DEBUGOFF</code>	Disables the printing of debug text.
<code>_DEBUGON</code>	Enables the printing of debug text.
<code>_LOADCMDHIST FILENAME\$</code>	Loads the command history from <code>FILENAME\$</code> .
<code>_PROMPT STRING\$</code>	Sets the prompt string to solve to <code>STRING\$</code> .

<code>_SAVECMDHIST FILENAME\$</code>	Saves the command history to <code>FILENAME\$</code> .
<code>_SHATTRIB {ATTRIB\$ ATTRIB}, {VALUE\$ VALUE}</code>	Sets the 'SH' attribute <code>ATTRIB\$</code> or <code>ATTRIB</code> to <code>VALUE\$</code> or <code>VALUE</code> .
<code>_TXTATTRIB {ATTRIB\$ ATTRIB}, {VALUE\$, VALUE}</code>	Sets the text attribute <code>ATTRIB\$</code> or <code>ATTRIB</code> to <code>VALUE\$</code> or <code>VALUE</code> .
<code>_TXTLOCK</code>	Stops the keyboard from echoing on the terminal.
<code>_TXTUNLOCK</code>	Undoes the effect of ' <code>_TXTLOCK</code> '.

## Functions:

<code>ASC (STRING\$, [POSITION])</code>	Returns the ASCII code of character <code>POSITION</code> (or 0 if <code>POSITION</code> is not defined) of <code>STRING\$</code> .
<code>BGC ()</code>	Returns the current background color.
<code>CHDIR (DIR\$)</code>	Attempts to change the current directory to <code>DIR\$</code> 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).
<code>CHR\$ (CODE)</code>	Returns ASCII character <code>CODE</code> .
<code>CWD\$ ()</code>	Returns the current working directory.
<code>CINT (NUMBER)</code>	Returns <code>NUMBER</code> rounded.
<code>COS (NUMBER)</code>	Returns the cosine of <code>NUMBER</code> .
<code>COSH (NUMBER)</code>	Returns the hyperbolic cosine of <code>NUMBER</code> .
<code>CURX ()</code>	Returns the X position of the cursor.
<code>CURY ()</code>	Returns the Y position of the cursor.
<code>EXP (NUMBER)</code>	Returns the exponent of <code>NUMBER</code> .
<code>FGC ()</code>	Returns the current foreground color.
<code>INKEY\$ ()</code>	Returns a character from the terminal.
<code>INPUT\$ (PROMPT\$)</code>	Returns a string after prompting for <code>PROMPT\$</code> (asks "?: " if <code>PROMPT\$</code> is not defined).
<code>INT (NUMBER)</code>	Returns <code>NUMBER</code> rounded down.

LCASE\$(STRING\$)	Returns the lower-case version of STRING\$.
LEN (STRING\$)	Returns the length of STRING\$.
LOG (NUMBER)	Returns the natural logarithm of NUMBER.
LOG10 (NUMBER)	Returns the common logarithm of NUMBER.
MOD (NUMBER)	Returns the modulus of NUMBER.
PI ()	Returns Pi.
{RAND RND} ({MAX MIN, MAX})	Returns a random number from MIN (0 if MIN is undefined) to MAX.
{SH EXEC} (COMMAND\$)	Runs COMMAND\$ in sh on Linux and Command Prompt on Windows and returns the exit status.
SIN (NUMBER)	Returns the sine of NUMBER.
SINH (NUMBER)	Returns the hyperbolic sine of NUMBER.
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\$)	Returns the numeric value of STRING\$.
_BITS\$ ()	Returns the executable bit format.
_DEBUG ()	Returns 1 if in debug mode and 0 otherwise.
_HOME\$ ()	Returns the path to the user's home directory.
_OS\$ ()	Returns the current operating system name.

<code>_PROMPT\$()</code>	Returns the prompt string.
<code>_TXTLOCK()</code>	Returns 1 if the text lock is in effect and 0 otherwise.
<code>_VER\$()</code>	Returns the CLIBASIC version.

## Logic Commands:

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

## Symbols:

<code>?</code>	Shortcut to <code>PRINT</code> .
<code>'</code>	Comment until the end of the line.