

BASIC commands

ABS	Return the absolute value of a numeric expression
AND	Perform a logical bitwise AND operation
ASC	Return the ASCII code of the specified character in a string
ASM	Identify an assembly-language statement
ATN	Return the arctangent of an argument
BEEP	Generate a beep using the speaker
BIN\$	Return a string that is the binary (base 2) representation of a value
BIT	Set/reset the value of a particular bit in an integer-class variable
CALL	Invoke a procedure (Sub or Function)
CBYT	Convert a value to a Byte data type
CDBL	Convert a value to a Double-precision data type
CDWD	Convert a value to a Double-word data type
CHR\$	Convert one or more ASCII codes into ASCII character(s)
CINT	Convert a value to a Integer data type
CLNG	Convert a value to a Long-integer data type
COS	Return the cosine of an argument
CSNG	Convert a value to a Single-precision data type
DATA	Declare an array of constants
DATE\$	Set or retrieve the system date
DEFBYT	Declare the default variable type to be Byte
DEFDBL	Declare the default variable type to be Double-precision
DEFDWD	Declare the default variable type to be Double-word
DEFINT	Declare the default variable type to be Integer
DEFLNG	Declare the default variable type to be Long-integer
DEFSNG	Declare the default variable type to be Single-precision
DEFSTR	Declare the default variable type to be String
DEFWRD	Declare the default variable type to be Word
DIM	Declare and dimension arrays, scalar variables, and pointers
DIR\$	Return a filename that matches the given mask
DISKFREE	Return the amount of available space on a disk, in bytes
DISKSIZE	Return the total amount of space on a disk, in bytes
DO/LOOP	Define a group of program statements that are executed repetitively
EOF	Return the end-of-file status of a file
EXIT	Transfer program execution out of a block structure
EXP	Return a number raised to a power of e (inverse natural logarithm)
FIX	Truncate a floating-point number to an integer
FOR/NEXT	Define a loop of program statements controlled by a counter
FORMAT\$	Format numeric data according to a string mask expression
FUNCTION	Define the start of a Function block
GET	Read a record from a random-access file
GET\$	Read a string from a file opened in binary mode
GLOBAL	Declare global (shared) variables between Subs and Functions
GOSUB	Invoke a local subroutine
GOTO	Transfer program execution to the statement identified by a label
HEX\$	Return a hexadecimal (base 16) string representation of an argument

IF	Test a condition and execute one or more program statements
IF/END IF	Create a IF/THEN/ELSE block with multiple lines and conditions
INPUT#	Load variables with data from a sequential file
INSTR	Search a string for the first occurrence of a character or string
INT	Convert a numeric expression to an integer-class value
KILL	Delete a disk file
LCASE\$	Return a lowercase version of a string argument
LEFT\$	Return the left-most n characters of a string
LEN	Return the logical length of a variable
LET	Assign a numeric variable or string
LINE INPUT#	Read line(s) from a sequential file into a string variable or string array
LOC	Determine the current seek position in an open disk file
LOCAL	Declare local variables in a Sub or Function
LOF	Return the length of an open disk file
LOG	Return the natural (base e) logarithm of an argument
LSET	Left-align a string within the space of another string
LSET\$	Return a string containing a left-justified (padded) string
LTRIM\$	Return a string with leading characters or strings removed
MID\$	Return a portion of a string
NOT	Logical bitwise NOT operation
OCT\$	Return a string that is a octal (base 8) representation of a value
ON ERROR	Specify an error handling routine; enable/ disable trapping
ON GOSUB	Call one of several subroutines based on a numeric expression
ON GOTO	Send program flow to one of several labels based on a value
OPEN	Prepare a file or device for reading or writing
OR	Logical bitwise OR arithmetic operation
PEEK	Return the byte at a specific memory location
PEEK\$	Return a sequence of bytes starting at a specific memory location
POKE	Store a byte at a specific memory location
POKE\$	Store a sequence of bytes starting at a specific memory location
PRINT#	Write a string or a complete array to a sequential file
PUT	Write a record to a random-access file or a variable to a binary file
PUT\$	Write a string to a file opened in binary mode
RANDOMIZE	Seed the random number generator
READ\$	Retrieve string data from a local DATA list
REDIM	Declare dynamic arrays, allocate, deallocate, or reallocate memory
REM	Indicate the remainder of a line of source code is a remark or comment
RESET	Set a variable, array element or an entire array to zero
RESUME	Continue execution after error handling with ON ERROR GOTO
RETURN	Return from a subroutine (GOSUB) to its caller
RIGHT\$	Return the rightmost n characters of a string
RND	Return a random number
RSET	Right justify a string into the space of a string variable
RSET\$	Return a string containing a right-justified (padded) string
RTRIM\$	Return a copy of a string with trailing characters or strings removed
SEEK	Set the position in a file for the next input or output operation
SELECT CASE	Control program flow based on the value of an expression
SGN	Return the sign of a numeric expression

SHIFT	Shift the bits in an integer-class variable
SIN	Return the sine of an argument
SPACE\$	Return a string consisting of a specified number of spaces
SQR	Return the square root of an argument
STATIC	Declare static variables inside of a Sub or Function
STR\$	Return the string representation of a number in printable form
STRING\$	Return a string with multiple copies of the specified character
SUB/END SUB	Define a Sub (procedure) block
TAN	Return the tangent of an argument
TIME\$	Read and/or set the system time
TIMER	Return the number of seconds that have elapsed since midnight
UCASE\$	Return an all-uppercase (capitalized) version of a string
USING\$	Format one or more string/numeric expressions using a mask string
VAL	Return the numeric equivalent of a string argument
VARPTR	Return the 32-bit address of a variable or string handle
WHILE/WEND	Define a block of program statements that are executed repeatedly
WRITE#	Output data to a sequential file in a delimited format
XOR	Perform a logical or a bitwise Exclusive-OR operation