

APPLE® COMMANDS

A Applesoft I Integer Basic	D DOS 3.3 P ProDOS™	f File Name	A\$ String X Variable	m,n,i,j Integers x,y,z Real numbers
--------------------------------	------------------------	-------------	--------------------------	--

- A I **ABS(x)** Absolute (positive) value of x
- A I **AND** Logical "and" in an IF statement
- D P **APPEND f** Add data to a sequential text file
- A I **ASC("A")** ASCII value of a character
- A I **ASC(A\$)** ASCII of string's first character
- A I **AT** See DRAW, XDRAW, HLIN and VLIN.
- A **ATN(x)** Arctangent of x in radians
- I **AUTO n,m** Start auto-line numbering
- D P **BLOAD f** Load binary file f
- D P **BRUN f** Load and Run binary program f
- D P **BSAVE f,A,n,Lm** Save data; Address n, Length m
- A I **CALL n** Branch to machine language routine at n
- P **CAT** Display 40-column ProDOS disk contents
- D P **CATALOG** Display disk's contents
- D P **CHAIN f** Run file f without clearing variables
- A **CHR\$(n)** Character whose ASCII value is n
- A **CLEAR** Clear all variables and strings
- D P **CLOSE f** Stop reading or writing a text file
- I **CLR** Clear all variables and strings
- A I **COLOR=n** Set lo-res color to n (0-15)
- I **CON** Continue an Integer program
- A **CONT** Continue an Applesoft program
- A **COS(x)** Cosine of x in radians
- P **CREATE f** Create a directory file
- A **DATA A\$,x,y,z** Strings and values to be READ
- A **DEF FN A(X)=f(x)** Define a function
- A I **DEL n,m** Delete program lines n through m
- D P **DELETE f** Delete file f from a disk
- A I **DIM X(n)** Dimension a numerical array
- A I **DIM A\$(n)** Dimension a string-array or string
- A **DRAW n AT i,j** Draw hi-res shape n at i,j
- I **DSP X** Display variable values and line numbers
- A I **END** Stop a program with no message (see STOP)
- D P **EXEC f** Execute text file f
- A **EXP(x)** e (2.718289) to the xth power
- A **FLASH** Set flashing screen output (40-columns)
- P **FLUSH** Write buffer to disk without closing file
- A **FN** See DEF FN
- A I **FOR X=n TO m** Let X=n, X=n+1... until X=m
- D **FP** Clear memory; switch to Applesoft Basic
- P **FRE** Free all available memory (garbage collection)
- A **FRE(0)** Amount of free memory available
- A **GET A\$** Wait for one-character user input
- A **GET X** Wait for one-number user input
- A I **GOSUB n** Branch to subroutine at line n
- I **GOSUB X** Branch to subroutine at line X
- A I **GOTO n** Branch to line n
- I **GOTO X** Branch to line X
- A I **GR** View and clear lo-res page 1
- A **HCOLOR=n** Set hi-res color to n (0-7)
- A **HGR** View and clear upper hi-res page 1
- A **HGR2** View and clear full hi-res page 2
- A I **HIMEM: n** Set highest memory address available
- A I **HLIN n,m AT j** Draw a horizontal lo-res line
- A **HOME** Clear text screen to black
- A **HPOINT i,j** Plot a hi-res point
- A **HPOINT i,j TO n,m** Draw a hi-res line
- A **HTAB n** Position cursor at horizontal position n
- A I **IF...THEN...** Logical "if" true, "then" execute
- A I D P **IN#n** Take input from slot n
- D **INIT f** Erase and format a disk
- A I **INPUT X (or A\$)** Wait for user input
- I **INPUT "ABC", A\$ (or X)** Print & wait for input
- A **INPUT "ABC"; A\$ (or X)** Print & wait for input
- D **INT** Switch to Integer Basic; clear memory
- A **INT(RND(1)*n)** Random integer 0 to n-1
- A **INT(x)** Integer value of x
- A **INVERSE** Set black-on-white text output
- A **LEFT\$(A\$,n)** Left n characters of a string
- A I **LEN(A\$)** Number of characters in a string
- A I **LET X=Y** Set X equal to Y ("LET" is optional)
- A I **LIST** List program from beginning
- A **LIST-n** List to line n
- A **LIST n-** List from line n
- A **LIST n-m** List lines n through m
- A I **LIST n,m** List lines n through m
- A I **LOAD** Load a program from tape
- D P **LOAD f** Load a file from disk
- D P **LOCK f** Protect a file from alteration
- A **LOG(x)** Natural logarithm of x
- A I **LOMEM: n** Set start-of-variables location
- I **MAN** Cancel AUTO
- D **MAXFILES n** Reserve n file buffers (1-16)
- A **MID\$(A\$,n,m)** m characters of A\$, starting at n
- I **A\$(n,m)** Characters n through m of a string
- I **m MOD n** Remainder of m divided by n
- D **MON C,I,O** Display disk functions
- A I **NEW** Delete current program from memory
- A **NEXT** Define bottom of a FOR-NEXT loop
- A I **NEXT X** Define bottom of a FOR-NEXT loop
- D **NOMON C,I,O** Cancel MON
- A **NORMAL** Set normal white-on-black text output
- A I **NOT** Logical "not" in an IF statement
- A I **NOTRACE** Cancel TRACE
- A **ON X GOSUB n,m...** GOSUB Xth line number
- A **ON X GOTO n,m** Branch to Xth line number
- A **ONERR GOTO n** Branch to line n if error occurs
- D P **OPEN f** Begin READ or WRITE of a text file
- A I **OR** Logical "or" in an IF statement
- A I **PDL(n)** Value (0-255) of paddle n (0-3)
- A I **PEEK(n)** Memory value at location n
- A I **PLOT i,j** Plot a lo-res dot
- A I **POKE n,m** Set location n to value m
- A I **POP** Cancel most recent GOSUB
- A **POS(0)** Horizontal cursor position
- D P **POSITION f** Locate READ or WRITE in file
- A I D P **PR#n** Send output to slot n
- P **PREFIX f** Change default directory
- A I **PRINT** Skip a text line
- A I **PRINT "ABC"** Print characters within quotes
- A I **PRINT X** Print value of variable X
- A **READ A\$** Read a DATA string
- A **READ X** Read a DATA value
- D P **READ f** Initiate reading a disk text file
- A **RECALL X** Retrieve array from tape
- A I **REM** Programmer's remark follows
- D P **RENAME f1,f2** Rename a file on disk
- A **RESTORE** Set pointer to first DATA element
- P **RESTORE f** Retrieve strings & variables from file f
- A **RESUME** Continue program where error occurred
- A I **RETURN** Branch back to statement after GOSUB
- A **RIGHT\$(A\$,n)** Last n characters of a string
- A **RND(0)** Repeat last random number
- A **RND(1)** Random number (0 to 0.999999999)
- A **RND(n)+1** Random integer between 1 and n
- I **ROT=n** Set rotation of a shape to n (0-64)
- A I **RUN** Execute program from beginning
- A I **RUN n** Execute program from line n
- D P **RUN f** Load and execute program from disk
- A I **SAVE** Save program to tape
- D P **SAVE f** Save program to disk
- A **SCALE=n** Set scale of DRAW or XDRAW
- A I **SCRN(i,j)** Lo-res screen color at point i,j
- A **SGN(X)** Sign (+1, -1 or 0) of X
- A **SHLOAD** Load shape table from tape
- A **SIN(x)** Sine of x in radians
- A **SPC(n)** n spaces in a PRINT statement
- A **SPEED=n** Character output rate (0-255)
- A **SQR(x)** Square root of x
- A I **STEP n** Increment-size in a FOR-NEXT loop
- A **STOP** Halt program and print line number
- P **STORE f** Store variables & strings as file f
- A **STORE X** Store array on tape
- A **STR\$(x)** String equivalent of value x
- A **TAB(n)** Position cursor in a PRINT statement
- I **TAB n** Position cursor at horizontal position n
- A **TAN(x)** Tangent of x in radians
- A I **TEXT** Switch to text mode; cancel windows
- A I **THEN** Logical "then" in an IF statement
- A I **TO** See FOR and HPOINT.
- A I **TRACE** Print line numbers as executed
- D P **UNLOCK f** Cancel LOCK
- A **USR(x)** Pass x to a machine subroutine
- A **VAL(A\$)** Numeric value of a string
- D P **VERIFY f** Verify a file on disk
- A I **VLIN n,m AT i** Draw a vertical lo-res line
- A I **VTAB n** Move cursor to text line n
- A **WAIT i,j,k** Insert conditional pause
- D P **WRITE f** Initiate writing to a disk text file
- A **XDRAW n AT i,j** DRAW in opposite color
- A **XPLOT** (Unused Applesoft reserved word)
- I **#** Not equal to
- P **-f** Execute file f, regardless of type
- A **?** Same as PRINT

CONTROL AND ESCAPE COMMANDS:

- A I **control-C** Stop an Applesoft or Integer program
- D P **control-D** CHR\$(4)—DOS command character
- A I **control-G** Beep the speaker
- A I **control-H** Left-arrow (backspace)
- A I **control-I** Tab (IIe/IIc software-controlled)
- A I **control-J** Down-arrow (line feed)
- A I **control-K** Up-arrow (IIe/IIc software-controlled)
- A I **control-M** Carriage return
- A I **control-U** Right-arrow (forward space)
- A I **control-X** Cancel line being typed
- A I **control-[** Escape (see esc commands below)
- A I **esc-@** Clear text screen; leave no prompt
- A I **esc-A** Move cursor one space right
- A I **esc-B** Move cursor one space left
- A I **esc-C** Move cursor one space down
- A I **esc-D** Move cursor one space up
- A I **esc-↑** (or esc-I) Move cursor up; recursive
- A I **esc-←** (or esc-J) Move cursor left; recursive
- A I **esc-→** (or esc-K) Move cursor right; recursive
- A I **esc-↓** (or esc-M) Move cursor down; recursive



Beagle Bros
Micro Software Inc.