

APPLE II COMMANDS

APPLE II is a registered trademark of the Apple Computer Company.

-] > **ABS(x)** Absolute (positive) value of x
-] > **AND** Logical "and" in IF statement
- D **APPEND f** Add data to end of sequential file
-] > **ASC("A")** ASC value of character in quotes
-] > **ASC(A\$)** ASC value of 1st character of string
-] > **AT** See DRAW, XDRAW, HLIN, VLIN
-] **ATN(x)** Arctangent of x (radians)
- > **AUTO n,m** Automatic line numbering

- D **BLOAD f** Load binary file
- D **BRUN f** Load & run binary file
- D **BSAVE f,An,Lm** Save binary program

-] > **CALL n** Branch to machine subroutine at n
-] > **CALL -1998** Clear full lo-res screen to black
-] > **CALL -1994** Clear upper lo-res screen to black
-] > **CALL -958** Clear screen from cursor to bottom
-] > **CALL -936** Clear text window
-] > **CALL -912** Scroll text up one line
-] > **CALL -868** Clear line from cursor to right
-] > **CALL -151** Enter monitor
-] **CALL 62450** Clear hi-res screen to black
-] **CALL 62454** Clear hi-res screen to color
- D **CHAIN f** Run file; variables not cleared
-] **CHR\$(n)** Character whose ASC = n
-] **CLEAR** Reset all variables to zero
- D **CLOSE** Close text file
- > **CLR** Reset all variables to zero
-] > **COLOR = n** Lo-res color (0-15)
- > **CON** Continue program
-] **CONT** Continue program
-] > **ctrl-C** Stop program
- D **ctrl-D** Execute command in print statement
-] > **ctrl-G** Beep speaker
-] > **ctrl-X** Cancel line being typed
-] **COS(x)** Cosine of x (radians)

-] **DATA x,y,z . . .** Data line to be read
-] **DEF FN A(X) = f(x)** Define substitute function
-] > **DEL n,m** Delete lines n through m

>: INTEGER BASIC f: File Name m,n,i,j: Integer
]: APPLESOFT A\$: String x,y,z: Real No.
D: D.O.S. X: Variable

Consult your Apple manuals for detailed descriptions and instructions.

-] **INT(RND(1)*n)** Random integer 1 to n
-] **INT(x)** Integer of x
-] **INVERSE** Set reverse text output
-] > **INPUT X (or A\$)** Wait for user input & return key
- > **INPUT "AB",X (or A\$)** Print AB & get input
-] **INPUT "AB";X (or A\$)** Print AB & get input

-] **LEFT\$(A\$,n)** Left n characters of A\$
-] > **LEN(A\$)** Number of characters in A\$
-] > **LET X = Y** X=Y (LET is optional)
-] > **LIST** List entire program
-] **LIST-n** List program to line n
-] **LIST n-** List lines n through end
-] **LIST n-m** List lines n through m
-] > **LIST n,m** List lines n through m
-] > **LOAD** Load program from tape
- D **LOAD f** Load file from disk
- D **LOCK f** Protect file from alteration
-] **LOG(x)** Natural logarithm of x
-] > **LOMEM:n** Set lowest memory available

- > **MAN** Cancel AUTO
- D **MAXFILES i** Reserve file buffers (1-16)
-] **MID\$(A\$,n,m)** m letters of A\$, starting with # n
- > **A\$(n,m)** Characters n through m of A\$
- > **m MOD n** Remainder of m divided by n
- D **MON C,I,O** Display disk executions

-] > **NEW** Delete current program
-] **NEXT** Define bottom of FOR . . . NEXT loop
-] > **NEXT X** Define bottom of FOR . . . NEXT loop
- D **NOMON C,I,O** Cancel MON
-] **NORMAL** Set normal text output
-] > **NOT** Logical "not" in IF statement

-] > **POKE -16302,0** Graphics & text to full graphics
-] > **POKE -16301,0** Full graphics to graphics & text
-] > **POKE -16300,0** Page 2 to page 1
-] > **POKE -16299,0** Page 1 to page 2
-] > **POKE -16298,0** Lo-res switch
-] > **POKE -16297,0** Hi-res switch
-] > **POKE 32,n** Set text window left edge (0-39)
-] > **POKE 33,n** Set text window width (1-40)
-] > **POKE 34,n** Set text window top edge (0-23)
-] > **POKE 35,n** Set text window bottom (0-23)
-] > **POKE 36,n** Move cursor to horizontal position n
-] > **POKE 37,n** Move cursor to vertical position n
-] > **POKE 50,63** Set reverse text output
-] > **POKE 50,127** Set flashing A-Z text output
-] > **POKE 50,255** Set normal text output
-] **POP** Cancel GOSUB
-] **POS (0)** Horizontal cursor position (0-39)
- D **POSITION f** Position READ or WRITE in file
-] > **PR#n** Send output to slot # n
-] > **PRINT "ABC"** Print characters in quotes
-] > **PRINT X** Print value of variable X

-] **READ X (or A\$)** Assign values from DATA
- D **READ f** Get input from text file
-] **RECALL X** Retrieve array from tape
-] > **REM** Remark; ignored by program
- D **RENAME f1,f2** Rename file on disk
-] **RESTORE** Reset pointer to 1st DATA element
-] **RESUME** Resume program where error occurred
-] > **RETURN** Branch to statement following GOSUB
-] **RIGHT\$(A\$,n)** Last n characters of string
-] **RND(0)** Repeat last random number
-] **RND(1)** Random number (0 to 0.999999999)
- > **RND(n) + 1** Random integer between 1 & n
-] **ROT = n** Set rotation of shape to n
-] > **RUN** Execute program from lowest line number
-] > **RUN n** Execute program from line n
- D **RUN f** Load & execute file from disk

-] > **SAVE** Save program to tape

D]> **DELETE f** Delete file from disk
]> **DIM X(n) or A\$(m)** Dimension array or string
] **DRAW n AT i,j** Draw hi-res shape at i,j
]> **DSP X** Display X values & line numbers

]> **END** Stop program (no message)
]> **esc-A** Move cursor one space right
]> **esc-B** Move cursor one space left
]> **esc-C** Move cursor one space down
]> **esc-D** Move cursor one space up
D] **EXEC f** Execute file retaining memory
] **EXP(x)** e (2.718289) to the xth power

] **FLASH** Set flashing screen output
]> **FOR X = n TO m** Set X = n, X = n + 1 ... until X = m
D] **FP** Switch to Applesoft BASIC
] **FRE(0)** Amount of memory available

] **GET A\$** Wait for one character input
]> **GOSUB n** Branch to subroutine at line n
]> **GOTO n** Branch to line n
]> **GOTO X or GOSUB X** Branch to line X
]> **GR** Switch to lo-res graphics

] **HCOLOR = n** Hi-res color (0-7)
] **HGR** Hi-res page 1 + text
] **HGR2** Hi-res page 2
]> **HIMEM: n** Set highest memory available
]> **HLIN n,m AT j** Horizontal lo-res line
] **HOME** Clear text screen
] **HPlot i,j** Plot hi-res dot
] **HPlot i,j TO n,m** Hi-res line from i,j to n,m
] **HTAB n** Cursor to horizontal tab n (1-40)

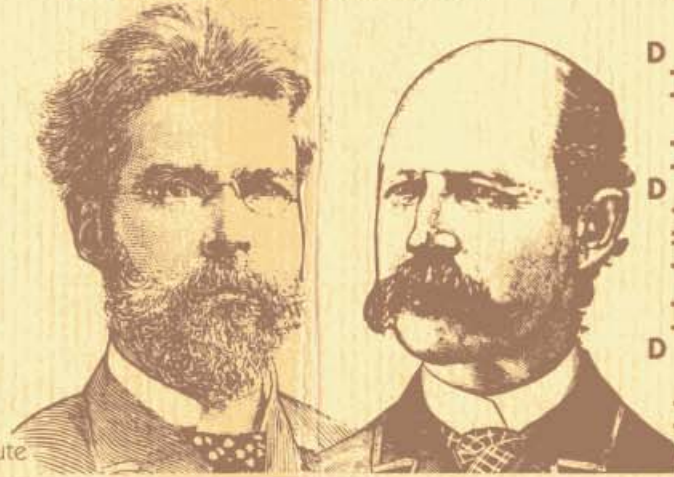
]> **IF ... THEN ...** Logical "if" true, "then" execute
]> **IN #n** Take input from slot n

D] **INIT f** Initialize disk
D] **INT** Switch to Integer BASIC

]> **NOT** Logical "not" in IF statement
]> **NOTRACE** Cancel TRACE

] **ON X GOSUB n,m ...** To subroutine at Xth no.
] **ON X GOTO n,m ...** Branch to Xth number
] **ONERR GOTO n** Branch to n if error occurs
D] **OPEN f** Open text file
]> **OR** Logical "or" in IF statement

]> **PDL(n)** Value (0-255) of paddle n (0-3)
]> **PEEK(n)** Memory contents of location n
]> **PEEK(n-16287)** Read paddle # n button (0-3)
]> **PEEK(-16384)** ASC value of key pressed
]> **PEEK(-16336)** Click speaker
]> **PEEK(36)** Horizontal cursor position (0-39)
]> **PEEK(37)** Vertical cursor position (0-23)
] **PEEK(222)** Error code
]> **PLOT i,j** Plot lo-res dot
]> **POKE n,m** Set memory at n to value m
]> **POKE -16368,0** Reset keyboard reader to zero
]> **POKE -16304,0** Text to graphics
]> **POKE -16303,0** Graphics to text



D]> **SAVE f** Save program to tape
] **SCALE = n** Scale for DRAW or XDRAW
]> **SCRN(i,j)** Lo-res screen color at i,j
]> **SGN(X)** Sign (+1, -1 or 0) of X
] **SHLOAD** Load shape table from tape
] **SIN(x)** Sine of x (radians)
] **SPC(n)** n spaces in PRINT statement
] **SPEED = n** Character output rate (0-255)
] **SQR(x)** Square root of x
]> **STEP n** Size of step in FOR ... NEXT loop
] **STOP** Stop program & print line number
] **STORE X** Store array on tape
] **STR\$(x)** String of value x

] **TAB(n)** Horizontal position in PRINT statement
]> **TAB n** Cursor to horizontal position n
] **TAN(x)** Tangent of x (radians)
]> **TEXT** Switch to text mode
]> **THEN** Logical "then" in IF statement
]> **TO** See FOR X= and HPlot
]> **TRACE** Print line numbers during execution

D] **UNLOCK f** Cancel LOCK
] **USR(x)** Pass x to machine subroutine

] **VAL(A\$)** Numeric value for string
D] **VERIFY f** Verify file on disk
]> **VLIN n,m AT i** Vertical lo-res line
]> **VTAB n** Move cursor to vertical position n

] **WAIT i,j,k** Insert conditional pause
D] **WRITE f** Write to text file

] **XDRAW n AT i,j** Erase hi-res shape

]># Not equal to
]><> Not equal to
]>< Not equal to



©1980, BEAGLE BROS. Micro Software

BEAGLE BROS. sells games for the Apple II. Send \$1 (refundable with purchase) for our informative 36 page TIP BOOK and CATALOG to: BEAGLE BROS., 4315 Sierra Vista, San Diego, CA 92103