

## TI-99/4A BASIC Reference Card



**C:** COMMAND **F:** FUNCTION **S:** STATEMENT

**ABS** (*numeric-expression*)  
returns absolute value. **F**

**ASC** (*string-expression*)  
returns the ASCII code of the first character  
of the string expression. **F**

**ATN** (*radian-expression*)  
returns trigonometric Arctangent. **F**

**BREAK** [*line-list*]  
causes program to halt when encountered or  
optionally when lines listed are  
encountered. **C,S**

**BYE**  
closes open files and leaves TI BASIC. **C**

**CALL CHAR** (*character-code, pattern-identifier*)  
redefines specified ASCII code using 16  
character HEX coded string. **C,S**

**CALL CLEAR**  
places the space character (ASCII 32) in all  
screen positions. **C,S**

**CALL COLOR** (*character-set, foreground-color,  
background-color*)  
specifies foreground and background colors  
of all characters in the specified set. **C,S**

**CALL GCHAR** (*row, column, numeric-variable*)  
returns the ASCII code of the character  
located at specified row (1-24) and column  
(1-32). **C,S**

**CALL HCHAR** (*row, column, character-code  
[,repetitions]*)  
places ASCII character at specified row  
(1-24) and column (1-32) and optionally  
repeats it horizontally. **C,S**

**CALL JOYST** (*key-unit, x-return, y-return*)  
inputs data based on the x (-4 0 4) and y  
(-4, 0, 4) position of the specified key-unit  
(1-4). **C,S**

**CALL KEY** (*key-unit, return-variable,  
status-variable*)  
assigns ASCII code of key pressed on  
specified key-unit (0-5) to return-variable. **C,S**  
Status information:  
1 = new key pressed  
-1 = same key pressed  
0 = no key pressed

**CALL SCREEN** (*color-code*)  
changes screen color. **C,S**

**CALL SOUND** (*duration, freq1, volume1  
[,freq2, volume2] [,freq3, volume3] [,freq4, volume4]*)  
controls up to three tone and one noise  
generators. Tone and noise parameters can  
occur in any order. Negative duration causes  
immediate sound update. **C,S**  
*duration:* 1 thru 4250 ms.  
-4250 thru -1 ms.  
*frequency:* 110 thru 44733 Hz  
(cycles/sec.) for tone  
-1 thru 8 for noise  
*volume:* 0 (loudest) thru 30 (softest)

**CALL VCHAR** (*row, column, character-code  
[,repetitions]*)  
places ASCII character at specified row  
(1-24) and column (1-32) and optionally  
repeats it vertically. **C,S**

**CHRS** (*numeric-expression*)  
returns the string character corresponding to the ASCII code. **F**

**CLOSE** #*file-number* | **DELETE** |  
discontinues association between a file and a program and optionally erases a file. **C,S**

**{ CONTINUE |  
CON }**  
resumes execution after breakpoint has been encountered. **C**

**COS** (*radial-expression*)  
returns trigonometric cosine. **F**

**DATA** *data-list*  
stores numeric and string constant data in a program. **S**

**DEF** *function-name* (*parameter*) = *expression*  
associates user-defined numeric or string expression with function name. **S**

**DELETE** { *file-name* }  
          { *program-name* }  
removes program or file from computer's filing system. **C**

**DIM** { *array-name* (*integer1*, *integer2*,  
*integer3*) }  
dimensions the listed arrays as specified by integers. **C,S**

**DISPLAY** | *print-list* |  
(see **PRINT** statement) **C,S**

**{ EDIT** *line-number* }  
**line-number** { | } }  
displays a line for editing. **C**

**END**  
terminates program execution. **C,S**

**EOF** (*numeric-expression*)  
returns the end-of-file condition of the specified file. **F**  
0: not end-of-file  
1: logical end-of-file  
-1: physical end-of-file

**EXP** (*numeric-expression*)  
returns exponential value ( $e^x$ ) of the argument. **F**

**FOR** *control-variable* = *initial-value* **TO** *limit*  
**STEP** *increment*  
repeats execution of statements between **FOR** and **NEXT** until the control-variable exceeds the limit. **STEP** default is one. **S**

**GOSUB** *line-number*  
transfers control to a subroutine at specified line-number until **RETURN** encountered. **S**

**GOTO** *line-number*  
unconditionally branches to specified line-number. **S**

**IF** { *relational-expression* } **THEN** *line-number1*  
          { *numeric-expression* }

**ELSE** *line-number2*  
transfers control to line-number1 if the relational expression is true or the numeric expression is not equal to zero. If false or equal to zero, control passes to the next statement or optionally to line-number2. **S**

**INPUT** | *input-prompt* | *variable-list*  
suspends program execution until data is entered from the keyboard. The optional input-prompt indicates data to be entered. **S**

**INPUT** #*file-number* | **REC** *record-number* |  
*variable-list*  
assigns data from specified file to the listed variables. Records are read sequentially unless optional **REC** clause is used. **S**

**INT** (*numeric-expression*)  
returns greatest integer less than or equal to the argument. **F**

**LEN** (*string-expression*)  
returns the number of characters in the string expression. **F**

**[LET]** { *numeric-variable* = *numeric-expression* }  
          { *string-variable* = *string-expression* }  
assigns the value of an expression to the specified variable. **C,S**

**LIST** { *line-number* }  
          { *line-number1* *line-number2* }  
          { *line-number* }  
          { *line-number* }  
sequentially displays program statements and optionally a single line number or all lines between the specified line numbers. **C**

**LOG** (*numeric-expression*)  
returns natural logarithm. **F**

**NEW**  
clears memory and screen and prepares computer for new program. **C**

**NEXT** *control-variable* (See **FOR** statement.) **S**

{ **NUMBER** } { *initial-line* | *increment* }  
{ **NUM** }  
automatically generates sequenced line numbers starting at 100 in increments of 10. Optionally, an initial line and/or increment may be specified. **C**

**OLD** *file-name*  
loads a program from a mass storage device into the computer's memory. **C**

**ON** *numeric-expression* **GOSUB** *line-number-list*  
transfers control to the subroutine with a beginning line number in the position corresponding to the value of the numeric expression. **S**

**ON** *numeric-expression* **GOTO** *line-number-list*  
unconditionally branches to line number in the position corresponding to the value of the expression. **S**

**OPEN** #*file-number*:*file-name* | *file-organization* |  
| *file-type* | | *open-mode* | | *record-type* |  
prepares a program to use specified file. **C,S**  
*file-number*: 0-255  
*file-organization*: SEQUENTIAL or  
RELATIVE  
*file-type*: DISPLAY or INTERNAL  
*open-mode*: INPUT, OUTPUT, UPDATE, or  
APPEND  
*record-type*: FIXED or VARIABLE

**OPTION BASE** { 0 }  
                  { 1 }  
sets the lowest allowable subscript of arrays to zero or one. Default is zero. **S**

**POS** (*string1*, *string2*, *numeric-expression*)  
returns the position of *string2* in *string1*. Search begins at position specified by numeric expression. Returns zero if no match is found. **F**

**PRINT** { | *print-list* | }  
          { #*file-number* | **REC** *record-number* | }  
          { : *print-list* }  
outputs to the display screen and optionally to an external file or device. The **REC** clause directs output to the specified record number. **C,S**

**RANDOMIZE** [*seed*]  
resets random number generator to an unpredictable sequence. With optional *seed* (numeric-expression), the sequence is repeatable. **C,S**

**READ** *variable-list*  
assigns number and string constants in DATA statements to variables listed. **S**

**REM** indicates internal program documentation. **C,S**

**{ RESEQUENCE }** *{initial-line|,increment|*  
**{ RES }** renumbers program statements starting at 100 in increments of 10. Optionally an initial line number and/or increment may be specified. **C**

**RESTORE** *{#file-number|,REC record-number|*  
*{line-number}* indicates the record or line from which data will be read. If no options, the beginning of a file or first data statement will be read next. **C,S**

**RETURN** transfers program control from subroutine to statement following corresponding GOSUB or ON . . . GOSUB statement. **S**

**RND** generates a pseudo-random number greater than or equal to zero and less than one. **F**

**RUN** *{line-number}* starts program execution at the lowest numbered statement or optionally at the specified line number. **C**

**SAVE** *file-name* places a copy of current program on the specified device. **C**

**SEGS** *(string-expression,position,length)* returns a substring beginning in the specified position with specified length. **F**

**SGN** *(numeric-expression)* returns 1 if argument is positive, 0 if argument equals zero, and -1 if argument is negative. **F**

**SIN** *(radian-expression)* returns the trigonometric sine. **F**

**SQR** *(numeric-expression)* returns square root. **F**

**STOP** terminates program execution. **C,S**

**STR\$** *(numeric-expression)* converts the value of the argument into a string. **F**

**TAB** *(numeric-expression)* controls column position of the output from a PRIN I or DISPLAY statement. **F**

**TAN** *(radian-expression)* returns the trigonometric tangent. **F**

**TRACE** lists line numbers of statements before they are executed. **C,S**

**UNBREAK** *{line-list}* removes all breakpoints or optionally those from lines listed. **C,S**

**UNTRACE** cancels the TRACE command. **C,S**

**VAL** *(string-expression)* converts a string representation of a number into a numeric constant. **F**

**Numeric Operators:** +, -, \*, /, ^

**String Operators:** &

**Relational Operators:** >, <, =, >=, <=, <>

**Numeric Range:**  
-1E-128 to -9.999999999999999E+127  
zero  
1E-128 to 9.999999999999999E+127

### CHARACTER SETS

Set	ASCII Codes	Set	ASCII Codes
1	32-39	0	06-103
2	40-47	10	104-111
3	48-55	11	112-119
4	56-63	12	120-127
5	64-71	13	128-135
6	72-79	14	136-143
7	80-87	15	144-151
8	88-95	16	152-159

### COLOR CODES

Value	Color	Value	Color
1	Transparent	9	Medium Red
2	Black	10	Light Red
3	Medium Green	11	Dark Yellow
4	Light Green	12	Light Yellow
5	Dark Blue	13	Dark Green
6	Light Blue	14	Magenta
7	Dark Red	15	Gray
8	Cyan	16	White

### FUNCTION KEY CODES

	Codes		Function Name	Function Key
	TI-99/4 & BASIC Modes	Pascal Mode		
1	129		AID	FCTN 7
2	130		CLEAR	FCTN 4
3	131		DELeTe	FCTN 1
4	132		INSert	FCTN 2
5	133		QUIT	FCTN =
6	134		REDO	FCTN 8
7	135		ERASE	FCTN 3
8	136		LEFT arrow	FCTN S
9	137		RIGHT arrow	FCTN D
10	138		DOWN arrow	FCTN X
11	139		UP arrow	FCTN E
12	140		PROG D	FCTN 6
13	141		ENTER	ENTER
14	142		BEGIN	FCTN 5
15	143		BACK	FCTN 9

### CONTROL KEY CODES

Codes			Press
BASIC Mode	Pascal Mode	Mnemonic Code	
129	1	SOH	CONTROL A
130	2	STX	CONTROL B
131	3	ETX	CONTROL C
132	4	EOT	CONTROL D
133	5	ENQ	CONTROL E
134	6	ACK	CONTROL F
135	7	BEL	CONTROL G
136	8	BS	CONTROL H
137	9	HT	CONTROL I
138	10	LF	CONTROL J
139	11	VT	CONTROL K
140	12	FF	CONTROL L
141	13	CR	CONTROL M
142	14	SO	CONTROL N
143	15	SI	CONTROL O
144	16	DLF	CONTROL P
145	17	DC1	CONTROL Q
146	18	DC2	CONTROL R
147	19	DC3	CONTROL S
148	20	DC4	CONTROL T
149	21	NAK	CONTROL U
150	22	SYN	CONTROL V
151	23	ETB	CONTROL W
152	24	CAN	CONTROL X
153	25	EM	CONTROL Y
154	26	SUB	CONTROL Z
155	27	ESC	CONTROL .
156	28	FS	CONTROL :
157	29	GS	CONTROL =
158	30	RS	CONTROL 8
159	31	US	CONTROL 9

CHARACTER CODES		ASCII CODE	CHAR	ASCII CODE	CHAR
ASCII CODE	CHARACTER	71	G	112	P
		72	H	113	Q
32	(space)	73	I	114	R
33	! (exclamation point)	74	J	115	S
34	" (quote)	75	K	116	T
35	# (number or pound sign)	76	L	117	U
36	\$ (dollar)	77	M	118	V
37	% (percent)	78	N	119	W
38	& (ampersand)	79	O	120	X
39	' (apostrophe)	80	P	121	Y
40	( (open parenthesis)	81	Q	122	Z
41	) (close parenthesis)	82	R	123	{ (left brace)
42	* (asterisk)	83	S	124	(pipe)
43	+ (plus)	84	T	125	} (right brace)
44	, (comma)	85	U	126	~ (tilde)
45	- (minus)	86	V	127	DEL (appears on screen as a blank.)
46	.	87	W		
47	/ (slant)	88	X		
48	0	89	Y		128-159 (user defined)
49	1	90	Z		
50	2	91			(open bracket)
51	3	92	\		(reverse slant)
52	4	93	}		(close bracket)
53	5	94	^		(exponentiation)
54	6	95	_		(line)
55	7	96	`		(grave)
56	8	97	A		
57	9	98	B		
58	:	99	C		
59	;	100	D		
60	<	101	E		
61	=	102	F		
62	>	103	G		
63	?	104	H		
64	@ (at sign)	105	I		
65	A	106	J		
66	B	107	K		
67	C	108	L		
68	D	109	M		
69	E	110	N		
70	F	111	O		