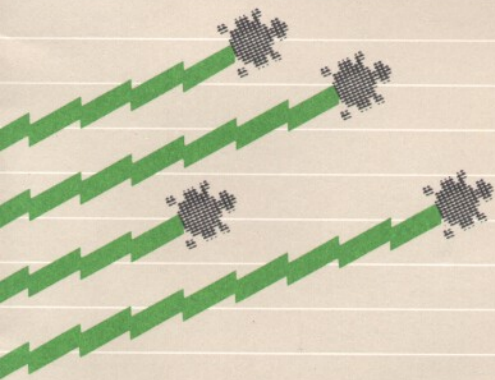


# ATARI LOGO

---

## Quick Reference Guide



---

## Table of Contents

Getting Started	2
Description of Inputs	3
Input Words	3
ATARI Logo Primitives	4
1 Turtle Graphics	4
2 Words and Lists	6
3 Variables	6
4 Arithmetic Operations	7
5 Defining and Editing Procedures	7
6 Flow of Control and Conditionals	8
7 Logical Operations	8
8 The Outside World	9
9 Workspace Management	10
10 Files	10
11 Special Primitives	11
12 Special Keys	11
Index	13

---

## Getting Started

1. With the computer off, turn on your TV set or monitor. If you have one, turn on your ATARI Disk Drive and wait for the busy light to go off. If you are not using a disk drive, skip to step 3.
2. Insert the ATARI Master Diskette in the disk drive and close the disk drive door. You may also use a data diskette if it contains DOS (Disk Operating System) files.
3. Insert the ATARI Logo Cartridge into the console's cartridge slot and turn the computer on.

The ? (question mark) is the prompt symbol. When ? is on the screen, you can type something. The █ is the cursor. It shows where the next character you type will appear.



---

## Description of Inputs

All of the words that we use in describing the inputs to the Logo primitives are explained below.

### Input Words

<i>byte</i>	A unit of data used by the computer. An integer from 0 through 255.
<i>character, char</i>	Letters of the alphabet, numbers, and punctuation marks.
<i>colornumber</i>	An integer from 0 through 127.
<i>condnumber</i>	An integer from 0 through 21.
<i>degrees</i>	Degrees of an angle. A real number between -9999.9999 and 9999.9999. The command REPEAT can be used to exceed this limit.
<i>device</i>	A device name. "C: is cassette, "D: is Disk, and "P: is Printer. The " (quote mark) and : (colon) are required at all times.
<i>distance</i>	A number from -9999.9999 through 9999.9999. The command REPEAT can be used to exceed this limit.
<i>duration</i>	An integer from 0 through 255.
<i>filename</i>	A file name.
<i>freq</i>	An integer from 14 through 64,000 in Hz.
<i>inputs</i>	Words with colons in front. Used in conjunction with TO.
<i>instructionlist</i>	A list of procedures that Logo can execute.
<i>joysticknumber</i>	An integer from 0 through 3.
<i>list</i>	Information enclosed in [ ] brackets.
<i>n, a, b, x, y</i>	A number.
<i>name</i>	A word naming a procedure or a variable.
<i>namelist</i>	A list of names.
<i>object, obj</i>	A Logo object (a word, a list or a number).
<i>paddlenumber</i>	An integer from 0 through 7.
<i>pennumber</i>	An integer from 0 through 2.
<i>position, pos</i>	A list of two numbers giving the coordinates of the turtle or the <i>cursor</i> .
<i>pred</i>	A predicate, which is an operation that outputs either the word TRUE or the word FALSE.
<i>shapenumber</i>	An integer from 0 through 15.
<i>shapespec</i>	A list of 16 numbers representing the shape grid.
<i>turtlenumber</i>	An integer from 0 through 3.
<i>voice</i>	An integer, either 0 or 1.
<i>volume</i>	An integer from 0 through 15.
<i>word</i>	A sequence of characters (not including a space).

---

## ATARI Logo Primitives

Note: A number sign (#) indicates a procedure which can take any number of inputs; if you give it other than the number indicated, you must enclose the entire expression in parentheses. An asterisk (\*) indicates an editing command which works both inside and outside the editor. The procedures that output TRUE under certain conditions would output FALSE when the conditions are not met.

---

### Turtle Graphics

<b>ASK</b> <i>turtlenumber list</i>	Asks the <i>turtlenumber(s)</i> to run the instructions in <i>list</i> .
<b>BACK, BK</b> <i>distance</i>	Moves turtle <i>distance</i> steps back.
<b>BG</b>	Outputs number representing background color.
<b>CLEAN</b>	Erases graphics screen without affecting turtle's state.
<b>COLOR</b>	Outputs number representing the current turtle(s) color.
<b>CS</b>	Erases screen, moves turtle to position [0 0]. Sets heading to 0.
<b>EACH</b> <i>list</i>	Makes each turtle separately run the commands in <i>list</i> .
<b>EDSH</b> <i>shapenumber</i>	Starts the ATARI Logo editor, displaying the shape of <i>shapenumber</i> requested.
<b>FORWARD, FD</b> <i>distance</i>	Moves turtle <i>distance</i> steps forward.
<b>GETSH</b> <i>shapenumber</i>	Returns a list of 16 numbers; these numbers correspond to bits in the shape.
<b>HEADING</b>	Outputs current turtle's heading.
<b>HOME</b>	Moves current turtle(s) to [0 0] and sets heading to 0.
<b>HT</b>	Makes current turtle(s) invisible.
<b>LEFT, LT</b> <i>degrees</i>	Turns turtle <i>degrees</i> left (counter-clockwise).
<b>PC</b> <i>pennumber</i>	Outputs number representing pen color of <i>pennumber</i> .
<b>PE</b>	Puts pen eraser down.
<b>PEN</b>	Outputs pen state (PD, PU, PE or PX).
<b>PENDOWN, PD</b>	Puts turtle's pen down.
<b>PENUP, PU</b>	Raises turtle's pen.
<b>PN</b>	Outputs the pen number (0, 1 or 2) currently being used.
<b>POS</b>	Outputs coordinates of turtle's position.



<b>PUTSH</b> <i>shapenumber shap espec</i>	Gives <i>shapenumber</i> the form of <i>shap espec</i> .
<b>PX</b>	Puts reversing pen down.
<b>RIGHT, RT</b> <i>degrees</i>	Turns turtle <i>degrees</i> right (clockwise).
<b>SETBG</b> <i>colornumber</i>	Sets background to <i>colornumber</i> (0 - 127).
<b>SETC</b> <i>colornumber</i>	Sets the current turtle(s) to <i>colornumber</i> .
<b>SETH</b> <i>degrees</i>	Sets current turtle's heading to <i>degrees</i> .
<b>SETPC</b> <i>pennumber colornumber</i>	Sets <i>pennumber</i> (0, 1 or 2) to <i>colornumber</i> (0 - 127).
<b>SETPN</b> <i>pennumber</i>	Sets the pen to <i>pennumber</i> (0, 1 or 2).
<b>SETPOS</b> <i>position</i>	Moves turtle to <i>position</i> .
<b>SETSH</b> <i>shapenumber</i>	Sets shape of current turtle to <i>shapenumber</i> .
<b>SETSP</b> <i>speed</i>	Sets the current turtle's <i>speed</i> , a number from - 199 through 199.
<b>SETX</b> <i>x</i>	Moves turtle horizontally to x-coordinate <i>x</i> .
<b>SETY</b> <i>y</i>	Moves turtle vertically to y-coordinate <i>y</i> .
<b>SHAPE</b>	Outputs number representing shape of the current turtle.
<b>SHOWNP</b>	Outputs TRUE if turtle is shown.
<b>SPEED</b>	Outputs current turtle's speed.
<b>ST</b>	Makes the turtle(s) visible.
<b>TELL</b> <i>turtlenumber(s)</i>	Addresses all following commands to <i>turtlenumber(s)</i> .
<b>WHO</b>	Outputs number of current turtle.
<b>WINDOW</b>	Makes graphics screen a window of an expanded turtle field. The screen is cleared.
<b>WRAP</b>	Makes turtle field wrap around edges of screen. The screen is cleared.
<b>XCOR</b>	Outputs x-coordinate of turtle's position.
<b>YCOR</b>	Outputs y-coordinate of turtle's position.

## 2

### Words and Lists

---

<b>ASCII</b> <i>character</i>	Outputs ASCII code for <i>character</i> .
<b>BUTFIRST</b> , <b>BF</b> <i>obj</i>	Outputs all but first element of <i>obj</i> .
<b>BUTLAST</b> , <b>BL</b> <i>obj</i>	Outputs all but last element of <i>obj</i> .
<b>CHAR</b> <i>n</i>	Outputs character whose ASCII code is <i>n</i> .
<b>COUNT</b> <i>obj</i>	Outputs the number of elements in <i>obj</i> .
<b>EMPTY</b> <i>obj</i>	Outputs <b>TRUE</b> if <i>obj</i> is empty.
<b>EQUALP</b> <i>obj1 obj2</i>	Outputs <b>TRUE</b> if its inputs are equal.
<b>FIRST</b> <i>obj</i>	Outputs first element of <i>obj</i> .
<b>FPUT</b> <i>obj list</i>	Outputs list formed by putting <i>obj</i> on front of <i>list</i> .
<b>LAST</b> <i>obj</i>	Outputs last element of <i>obj</i> .
<b>LIST</b> <i>obj1 obj2</i>	Outputs list of its inputs.
<b>LISTP</b> <i>obj</i>	Outputs <b>TRUE</b> if <i>obj</i> is a list.
<b>LPUT</b> <i>obj list</i>	Outputs list formed by putting <i>obj</i> on end of <i>list</i> .
<b>MEMBERP</b> <i>obj list</i>	Outputs <b>TRUE</b> if <i>obj</i> is an element in <i>list</i> .
<b>NUMBERP</b> <i>obj</i>	Outputs <b>TRUE</b> if <i>obj</i> is a number.
<b># SE</b> <i>obj1 obj2</i>	Outputs list of its inputs (if words) or the members of its inputs (if lists).
<b># WORD</b> <i>word1 word2</i>	Outputs word made up of its inputs.
<b>WORDP</b> <i>object</i>	Outputs <b>TRUE</b> if <i>object</i> is a word.
<i>obj1 = obj2</i>	Outputs <b>TRUE</b> if <i>obj1</i> is equal to <i>obj2</i> .

## 3

### Variables

---

<b>MAKE</b> <i>name obj</i>	Makes <i>name</i> refer to <i>obj</i> .
<b>NAMEP</b> <i>name</i>	Outputs <b>TRUE</b> if <i>name</i> has a value.
<b>THING</b> <i>name</i>	Outputs object referred to by <i>name</i> .



## 4

### Arithmetic Operations

---

<b>COS</b> $n$	Outputs cosine of $n$ degrees.
<b>INT</b> $n$	Outputs the integer portion of $n$ .
<b># PRODUCT</b> $a$ $b$	Outputs product of its inputs.
<b>RANDOM</b> $n$	Outputs random integer between 0 and $n - 1$ .
<b>REMAINDER</b> $a$ $b$	Outputs remainder of $a$ divided by $b$ .
<b>RERANDOM</b>	Makes <b>RANDOM</b> behave reproducibly.
<b>ROUND</b> $n$	Outputs $n$ rounded off to nearest integer.
<b>SIN</b> $n$	Outputs sine of $n$ degrees.
<b>SQRT</b> $n$	Outputs square root of $n$ .
<b>SUM</b> $a$ $b$	Outputs sum of its inputs.
$a + b$	Outputs $a$ plus $b$ .
$a - b$	Outputs $a$ minus $b$ .
$a * b$	Outputs $a$ times $b$ .
$a / b$	Outputs $a$ divided by $b$ .
$a < b$	Outputs <b>TRUE</b> if $a$ is less than $b$ .
$a > b$	Outputs <b>TRUE</b> if $a$ is greater than $b$ .
$a = b$	Outputs <b>TRUE</b> if $a$ is equal to $b$ .

## 5

### Defining and Editing Procedures

---

<b>EDIT, ED</b> $name(s)$	Starts Logo editor with named procedure(s).
<b>EDNS</b>	Starts Logo editor with all variables in the workspace.
<b>END</b>	Ends the procedure definition started out by <b>TO</b> .
<b>TO</b> $name$ ( $inputs$ )	Begins defining procedure $name$ .

# 6

## Flow of Control and Conditionals

---

<b>COND</b> <i>condnumber</i>	Outputs <b>TRUE</b> if that particular condition specified by <i>condnumber</i> is occurring.
<b>IF</b> <i>pred list1 (list2)</i>	If <i>pred</i> is <b>TRUE</b> , runs <i>list1</i> , otherwise <i>list2</i> .
<b>OUTPUT, OP</b> <i>obj</i>	Returns control to caller, with <i>obj</i> as output.
<b>OVER</b> <i>turtlenumber pennumber</i>	Outputs number symbolizing collision between <i>turtlenumber</i> and <i>pennumber</i> .
<b>REPEAT</b> <i>n list</i>	Runs <i>list</i> <i>n</i> times.
<b>RUN</b> <i>list</i>	Runs <i>list</i> ; outputs what <i>list</i> outputs.
<b>STOP</b>	Stops procedure and returns control to caller.
<b>TOUCHING</b> <i>turtlenumber1 turtlenumber2</i>	Outputs number symbolizing collision between <i>turtlenumber1</i> and <i>turtlenumber2</i> .
<b>WAIT</b> <i>n</i>	Pauses for <i>n</i> 60ths of a second.
<b>WHEN</b> <i>condnumber list</i>	Sets up <b>WHEN</b> demon so whenever condition <i>condnumber</i> occurs, <i>list</i> is run.
<b>WHEN</b> <i>condnumber []</i>	Clears (stops) <b>WHEN</b> demon for <i>condnumber</i> .

# 7

## Logical Operations

---

# <b>AND</b> <i>pred1 pred2</i>	Outputs <b>TRUE</b> if all its inputs are <b>TRUE</b> .
<b>FALSE</b>	Outputs the word <b>FALSE</b> . Special input for <b>AND</b> , <b>IF</b> , <b>NOT</b> and <b>OR</b> .
<b>NOT</b> <i>pred</i>	Outputs <b>TRUE</b> if <i>pred</i> is <b>FALSE</b> .
# <b>OR</b> <i>pred1 pred2</i>	Outputs <b>TRUE</b> if any of its inputs are <b>TRUE</b> .
<b>TRUE</b>	Outputs the word <b>TRUE</b> . Special input for <b>AND</b> , <b>IF</b> , <b>NOT</b> and <b>OR</b> .



## The Outside World

<b>CT</b>	Clears text section of screen.
<b>FS</b>	Devotes entire screen to graphics.
<b>JOY</b> <i>joysticknumber</i>	Outputs current position of <i>joysticknumber</i> .
<b>JOYB</b> <i>joysticknumber</i>	Outputs <b>TRUE</b> if the button on <i>joysticknumber</i> is pressed.
<b>KEYP</b>	Outputs <b>TRUE</b> if a key has been typed but not yet read.
<b>PADDLE</b> <i>paddlenumber</i>	Outputs rotation on dial of <i>paddlenumber</i> .
<b>PADDLEB</b> <i>paddlenumber</i>	Outputs <b>TRUE</b> if the button is pressed on <i>paddlenumber</i> .
<b># PRINT, PR</b> <i>obj</i>	Prints <i>obj</i> followed by carriage return (strips off outer brackets of lists).
<b>RC</b>	Outputs character read by the current device (default is keyboard). Waits if necessary.
<b>RL</b>	Outputs line read by current device (default is keyboard). Waits if necessary.
<b>SETCURSOR</b> <i>position</i>	Puts cursor at <i>position</i> .
<b>SETENV</b> <i>voice duration</i>	Sets envelope of <i>voice</i> for <b>TOOT</b> so volume reduces by one unit every <i>duration</i> .
<b>SHOW</b> <i>obj</i>	Prints <i>obj</i> followed by <b>RETURN</b> with brackets for list.
<b>SS (CTRL S)</b>	Splits screen: top for graphics, bottom for text.
<b>TOOT</b> <i>voice freq volume duration</i>	Produces sound on <i>voice</i> of frequency <i>freq</i> and <i>volume</i> for a given <i>duration</i> .
<b>TS (CTRL T)</b>	Devotes entire screen to text.
<b># TYPE</b> <i>obj</i>	Prints <i>obj</i> leaving <i>cursor</i> at the end of the printed line.

## Workspace Management

<b>ERALL</b>	Erases everything from the workspace. Frees up all nodes.
<b>ERASE, ER</b> <i>name(s)</i>	Erases all named procedure(s).
<b>ERN</b> <i>name(s)</i>	Erases all named variables.
<b>ERNS</b>	Erases all variables from the workspace.
<b>ERPS</b>	Erases all procedures from the workspace.
<b>NODES</b>	Outputs number of free nodes.
<b>PO</b> <i>name(s)</i>	Prints definitions of named procedures.
<b>POALL</b>	Prints all definitions of procedures and names (variables).
<b>POD</b> <i>condnumber</i>	Prints <b>WHEN</b> demon <i>condnumber</i> currently in action.
<b>PODS</b>	Prints out all active <b>WHEN</b> demons.
<b>PONS</b>	Prints names and values of all variables.
<b>POPS</b>	Prints definitions of all procedures.
<b>POTS</b>	Prints title lines of all procedures.
<b>RECYCLE</b>	Performs a garbage collection.

## 10

### Files

<b>CATALOG</b> <i>device:</i>	Displays names of all files on diskette. On a cassette, prints all the procedure definitions and names in the file.
<b>ERF</b> <i>device:filename</i>	Erases <i>filename</i> from <i>device</i> .
<b>LOAD</b> <i>device:filename</i>	Loads file called <i>filename</i> from the <i>device</i> into the computer.
<b>SAVE</b> <i>device:filename</i>	Saves workspace onto the <i>device</i> . If <i>device</i> is a printer, all procedures are printed.
<b>SETREAD</b> <i>device:filename</i>	Sets the <i>device:filename</i> from which data will be read by <b>RC</b> and <b>RL</b> .
<b>SETREAD</b> [ ]	Closes the file that was opened with <b>SETREAD</b> .
<b>SETWRITE</b> <i>device:filename</i>	Starts the process of sending to <i>filename</i> on the <i>device</i> a copy of all the characters displayed on the screen.
<b>SETWRITE</b> [ ]	Closes the file that was opened with <b>SETWRITE</b> .



## 11

### Special Primitives

---

<b>.CALL</b> <i>n</i>	Transfers control to a machine language subroutine starting at address <i>n</i> (decimal).
<b>.DEPOSIT</b> <i>n</i> <i>byte</i>	Writes <i>byte</i> into address <i>n</i> (decimal).
<b>.EXAMINE</b> <i>n</i>	Outputs contents of address <i>n</i> (decimal).
<b>.PRIMITIVES</b>	Prints the list of Logo primitives.
<b>.SETSCR</b> <i>n</i>	Sets aspect ratio to <i>n</i> .

## 12

### Special Keys

---

<b>ATARI Key</b> (↵)	After this key is pressed, all characters typed appear in reverse video on the screen.
<b>Reverse Video Key</b> (◻)	
* <b>BREAK</b>	Aborts whatever Logo is doing. If editing, changes made in the edit buffer will be ignored. Clears the line currently being typed at the top level.
* <b>CTRL</b> →	Moves the <i>cursor</i> one position to the right.
* <b>CTRL</b> ←	Moves the <i>cursor</i> one position to the left.
<b>CTRL</b> ↑	Moves the <i>cursor</i> up to the previous line.
<b>CTRL</b> ↓	Moves the <i>cursor</i> down to the next line.
* <b>CTRL 1</b>	Makes Logo stop scrolling until <b>CTRL 1</b> is typed again.
* <b>CTRL A</b>	Moves the <i>cursor</i> to the beginning of the current line.
* <b>CTRL CLEAR</b>	Deletes text from the <i>cursor</i> position to the end of the current line.
* <b>CTRL DELETE BACK S</b>	Erases the character at the <i>cursor</i> position.
* <b>CTRL E</b>	Moves the <i>cursor</i> to the end of the current line.
<b>CTRL F</b>	Devotes full screen to graphics.
<b>CTRL INSERT</b>	Opens a new line at the position of the <i>cursor</i> .
<b>CTRL S</b>	Splits screen: top for graphics, bottom for text.
<b>CTRL T</b>	Devotes entire screen to text.
<b>CTRL V</b>	Scrolls screen to next page in editor.
<b>CTRL W</b>	Scrolls screen back to previous page in editor.
<b>CTRL X</b>	Moves the <i>cursor</i> to beginning of editor.

* <b>CTRL Y</b>	In the editor, inserts the contents of the delete buffer. Outside the editor, inserts the last command typed.
<b>CTRL Z</b>	Moves the <i>cursor</i> to end of editor.
* <b>DELETE BACK S</b>	Erases the character to the left of the <i>cursor</i> .
<b>ESC</b>	Completes editing and exits to top level.
<b>F1, F2, F3, F4</b>	<i>Cursor</i> control keys that can be programmed.
* <b>RETURN</b>	Completes the line and puts the <i>cursor</i> at the beginning of the next line.
* <b>SHIFT DELETE BACK S</b>	Deletes text from the <i>cursor</i> position to the end of the current line.
<b>SHIFT INSERT</b>	Opens a new line at the position of the <i>cursor</i> .
<b>\ (Backslash)</b>	Tells Logo to interpret the character that follows it literally as a <i>character</i> , rather than keeping some special meaning it might have. You have to backslash [ , ] , ( , ) , + , - , / , = , < , > , and itself.
<b>SYSTEM RESET</b>	Reboots Logo, erasing the memory space.



## Index

<b>A</b>					
AND	8	EMPTY	6	OR	8
ASCII	6	END	7	OUTPUT, OP	8
ASK	4	EQUALP	6	OVER	8
ATARI Key	11	ERALL	10	<b>P</b>	
<b>B</b>		ERASE, ER	10	PADDLE	9
BACK, BK	4	ERF	10	PADDLEB	9
backslash	12	ERN	10	paddlenumber	3
BG	4	ERNS	10	PC	4
BREAK	11	ERPS	10	PE	4
BUTFIRST, BF	6	ESC	12	PEN	4
BUTLAST, BL	6	.EXAMINE	11	PENDOWN, PD	4
byte	3	<b>F</b>		penumber	3
<b>C</b>		F1, F2, F3, F4	12	PENUP, PU	4
.CALL	11	FALSE	8	PN	4
CATALOG	10	filename	3	PO	10
CHAR	6	FIRST	6	POALL	10
character	3	FORWARD, FD	4	POD	10
CLEAN	4	FPUT	6	PODS	10
COLOR	4	frequency, freq	3	PONS	10
colnumber	3	FS (CTRL F)	9	POPS	10
COND	8	<b>G</b>		pos	3
condnumber	3	GETSH	4	POS	4
COS	7	<b>H</b>		position	3
COUNT	6	HEADING	4	POTS	10
CS	4	HOME	4	pred	3
CT	9	HT	4	.PRIMITIVES	11
CTRL 1	11	<b>I</b>		PRINT, PR	9
CTRL ←	11	IF	8	PRODUCT	7
CTRL →	11	inputs	3	PUTSH	5
CTRL ↑	11	instructionlist	3	PX	5
CTRL ↓	11	INT	7	<b>R</b>	
CTRL CLEAR	11	<b>J</b>		RANDOM	7
CTRL DELETE BACK S	11	JOY	9	RC	9
CTRL INSERT	11	JOYB	9	RECYCLE	10
CTRL A	11	joysticknumber	3	REMAINDER	7
CTRL E	11	<b>K</b>		REPEAT	8
CTRL F	11	KEYP	9	RERANDOM	7
CTRL S	11	<b>L</b>		RETURN	12
CTRL T	11	LAST	6	REVERSE VIDEO Key	11
CTRL V	11	LEFT	4	RIGHT	5
CTRL W	11	list	3	RL	9
CTRL X	11	LIST	6	ROUND	7
CTRL Y	12	LISTP	6	RUN	8
CTRL Z	12	LOAD	10	<b>S</b>	
<b>D</b>		LPUT	6	SAVE "D:	10
degrees	3	<b>M</b>		SAVE "P:	10
DELETE BACK S	12	MAKE	6	SE	6
.DEPOSIT	11	MEMBERP	6	SETBG	5
device	3	<b>N</b>		SETC	5
distance	3	name	3	SETCURSOR	9
duration	3	namelist	3	SETENV	9
<b>E</b>		NAMEP	6	SETH	5
EACH	4	NODES	10	SETPC	5
EDIT, ED	7	NOT	8	SETPN	5
EDNS	7	NUMBERP	6	SETPOS	5
EDSH	4	<b>O</b>		SETREAD	10
		object	3	SETREAD []	10

---

.SETSCR	11
SETSH	5
SETSP	5
SETWRITE	10
SETWRITE []	10
SETX	5
SETY	5
SHAPE	5
shapenumber	3
shapespec	3
SHIFT DELETE BACK S	12
SHIFT INSERT	12
SHOW	9
SHOWNP	5
SIN	7
SPEED	5
SQRT	7
SS (CTRL S)	9
ST	5
STOP	8
SUM	7
SYSTEM RESET	12
<b>T</b>	
TELL	5
THING	6
TO	7
TOOT	9
TOUCHING	8
TRUE	8
TS (CTRL T)	9
turtlenumber	3
TYPE	9
<b>V</b>	
voice	3
volume	3
<b>W</b>	
WAIT	8
WHEN	8
WHEN []	8
WHO	5
WINDOW	5
word	3
WORD	6
WORDP	6
WRAP	5
<b>X</b>	
XCOR	5
<b>Y</b>	
YCOR	5
a + b	7
a - b	7
a * b	7
a / b	7
a < b	7
a > b	7
a = b	7
obj1 = obj2	6



Every effort has been made to ensure the accuracy of the product documentation in this manual. However, because we are constantly improving and updating our computer software and documentation, Logo Computer Systems, Inc. is unable to guarantee the accuracy of printed material after the date of publication and disclaims liability for changes, errors or omissions.

No reproduction of this document or any portion of its contents is allowed without the specific written permission of Logo Computer Systems, Inc.

© 1983 Logo Computer Systems, Inc.  
All rights reserved.



Logo Computer  
Systems, Inc.  
9960 Cote de Liesse  
Lachine, Quebec  
Canada H8T 1A1

