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 1 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

byte
   A unit of data used by the computer. An integer from 0 through 255.

calendar
   Letters of the alphabet, numbers, and punctuation marks.

colormnumber
   An integer from 0 through 127.

colomnumber
   An integer from 0 through 21.

degrees
   Degrees of an angle. A real number between -9999.9999 and 9999.9999. The command REPEAT can be used to exceed this limit.

device
   A device name. "C: is cassette, "D: is Disk, and "P: is Printer. The " (quote mark) and : (colon) are required at all times.

distance
   A number from -9999.9999 through 9999.9999. The command REPEAT can be used to exceed this limit.

duration
   An integer from 0 through 255.

duration
   A file name.

duration
   An integer from 1 through 64,000 in Hz.

duration
   Words with colons in front. Used in conjunction with TO.

duration
   A list of procedures that Logo can execute.

duration
   An integer from 0 through 3.

duration
   Information enclosed in [ ] brackets.

duration
   A number.

duration
   A word naming a procedure or a variable.

duration
   A list of names.

duration
   A Logo object (a word, a list or a number).

duration
   An integer from 0 through 7.

duration
   An integer from 0 through 2.

duration
   A list of two numbers giving the coordinates of the turtle or the cursor.

duration
   A predicate, which is an operation that outputs either the word TRUE or the word FALSE.

shape
   An integer from 0 through 15.

shape
   A list of 16 numbers representing the shape grid.

turtle
   An integer from 0 through 3.

turtle
   An integer, either 0 or 1.

turtle
   An integer from 0 through 15.

voice
   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

**ASK turtlenumber list**
Asks the turtlenumber(s) to run the instructions in list.

**BACK, BK distance**
Moves turtle distance steps back.

**BG**
Outputs number representing background color.

**CLEAN**
Erases graphics screen without affecting turtle's state.

**COLOR**
Outputs number representing the current turtle(s) color.

**CS**
Erases screen, moves turtle to position [0 0]. Sets heading to 0.

**EACH list**
Makes each turtle separately run the commands in list.

**EDSH shapenumber**
Starts the ATARI Logo editor, displaying the shape of shapenumber requested.

**FORWARD, FD distance**
Moves turtle distance steps forward.

**GETSH shapenumber**
Returns a list of 16 numbers; these numbers correspond to bits in the shape.

**HEADING**
Outputs current turtle's heading.

**HOME**
Moves current turtle(s) to [0 0] and sets heading to 0.

**HT**
Makes current turtle(s) invisible.

**LEFT, LT degrees**
Turns turtle degrees left (counter-clockwise).

**PC penumber**
Outputs number representing pen color of penumber.

**PE**
Puts pen eraser down.

**PEN**
Outputs pen state (PD, PU, PE or PX).

**PENDOWN, PD**
Puts turtle's pen down.

**PENUP, PU**
Raises turtle's pen.

**PN**
Outputs the pen number (0, 1 or 2) currently being used.

**POS**
Outputs coordinates of turtle's position.

**PUTSH shapenumber shapespec**
Gives shapenumber the form of shapespec.

**PX**
Puts reversing pen down.

**RIGHT, RT degrees**
Turns turtle degrees right (clockwise).

**SETBG colornumber**
Sets background to colornumber (0 - 127).

**SETC colornumber**
Sets the current turtle(s) to colornumber.

**SETH degrees**
Sets current turtle's heading to degrees.

**SETPC penumber colornumber**
Sets pennumber (0, 1 or 2) to colornumber (0 - 127).

**SETPN penumber**
Sets the pen to pennumber (0, 1 or 2).

**SETPOS position**
Moves turtle to position.

**SETSH shapenumber**
Sets shape of current turtle to shapenumber.

**SETSP speed**
Sets the current turtle's speed, a number from -199 through 199.

**SETX x**
Moves turtle horizontally to x-coordinate x.

**SETY y**
Moves turtle vertically to y-coordinate y.

**SHAPE**
Outputs number representing shape of the current turtle.

**SHOWNP**
Outputs TRUE if turtle is shown.

**SPEED**
Outputs current turtle's speed.

**ST**
Makes the turtle(s) visible.

**TELL turtlenumber(s)**
Addresses all following commands to turtlenumber(s).

**WHO**
Outputs number of current turtle.

**WINDOW**
Makes graphics screen a window of an expanded turtle field. The screen is cleared.

**WRAP**
Makes turtle field wrap around edges of screen. The screen is cleared.

**XCOR**
Outputs x-coordinate of turtle's position.

**YCOR**
Outputs y-coordinate of turtle's position.
### Words and Lists

- **ASCII** character: Outputs ASCII code for character.
- **BUTFIRST**, **BF** obj: Outputs all but first element of obj.
- **BUTLAST**, **BL** obj: Outputs all but last element of obj.
- **CHAR** n: Outputs character whose ASCII code is n.
- **COUNT** obj: Outputs the number of elements in obj.
- **EMPTYP** obj: Outputs TRUE if obj is empty.
- **EQUALP** obj1 obj2: Outputs TRUE if its inputs are equal.
- **FIRST** obj: Outputs first element of obj.
- **FPUT** obj list: Outputs list formed by putting obj on front of list.
- **LAST** obj: Outputs last element of obj.
- **LIST** obj1 obj2: Outputs list formed by putting obj on end of list.
- **LISTP** obj: Outputs TRUE if obj is a list.
- **LPUT** obj list: Outputs list formed by putting obj on end of list.
- **MEMBERP** obj list: Outputs TRUE if obj is an element in list.
- **NUMBERP** obj: Outputs TRUE if obj is a number.
- **# WORD** word1 word2: Outputs word made up of its inputs.
- **WORDP** object: Outputs TRUE if object is a word.
- **obj1 = obj2**: Outputs TRUE if obj1 is equal to obj2.

### Variables

- **MAKE** name obj: Makes name refer to obj.
- **NAMEP** name: Outputs TRUE if name has a value.
- **THING** name: Outputs object referred to by name.

### Arithmetic Operations

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

### Defining and Editing Procedures

- **EDIT**, **ED** name(s): Starts Logo editor with named procedure(s).
- **EDNS**: Starts Logo editor with all variables in the workspace.
- **END**: Ends the procedure definition started out by **TO**.
- **TO** name (inputs): Begins defining procedure name.
Flow of Control and Conditionals

**COND** connumber
- Outputs TRUE if that particular condition specified by connumber is occurring.

**IF** pred list1 (list2)
- If pred is TRUE, runs list1, otherwise list2.

**OUTPUT, OP** obj
- Returns control to caller, with obj as output.

**OVER** turtlenumber pennumber
- Outputs number symbolizing collision between turtlenumber and pennumber.

**REPEAT** n list
- Runs list n times.

**RUN** list
- Runs list; outputs what list outputs.

**STOP**
- Stops procedure and returns control to caller.

**TOUCHING** turtlenumber1 turtlenumber2
- Outputs number symbolizing collision between turtlenumber1 and turtlenumber2.

**WAIT** n
- Pauses for n 60ths of a second.

**WHEN** connumber list
- Sets up WHEN demon so whenever condition connumber occurs, list is run.

**WHEN** connumber [ ]
- Clears (stops) WHEN demon for connumber.

Logical Operations

**# AND** pred1 pred2
- Outputs TRUE if all its inputs are TRUE.

**FALSE**
- Outputs the word FALSE. Special input for AND, IF, NOT and OR.

**NOT** pred
- Outputs TRUE if pred is FALSE.

**# OR** pred1 pred2
- Outputs TRUE if any of its inputs are TRUE.

**TRUE**
- Outputs the word TRUE. Special input for AND, IF, NOT and OR.

The Outside World

**CT**
- Clears text section of screen.

**FS**
- Devotes entire screen to graphics.

**JOY** joysticknumber
- Outputs current position of joysticknumber.

**JOYB** joysticknumber
- Outputs TRUE if the button on joysticknumber is pressed.

**KEYP**
- Outputs TRUE if a key has been typed but not yet read.

**PADDLE** paddlenumber
- Outputs rotation on dial of paddlenumber.

**PADDLEB** paddlenumber
- Outputs TRUE if the button is pressed on paddlenumber.

**# PRINT, PR** obj
- Prints obj followed by carriage return (strips off outer brackets of lists).

**RC**
- Outputs character read by the current device (default is keyboard). Waits if necessary.

**RL**
- Outputs line read by current device (default is keyboard). Waits if necessary.

**SETCURSOR** position
- Puts cursor at position.

**SETENV** voice duration
- Sets envelope of voice for TOOT so volume reduces by one unit every duration.

**SHOW** obj
- Prints obj followed by RETURN with brackets for list.

**SS (CTRL S)**
- Splits screen: top for graphics, bottom for text.

**TOOT** voice freq volume duration
- Produces sound on voice of frequency freq and volume for a given duration.

**TS (CTRL T)**
- Devotes entire screen to text.

**# TYPE** obj
- Prints obj leaving cursor at the end of the printed line.
Workspace Management

ERALL
Erases everything from the workspace. Frees up all nodes.

ERASE, ER name(s)
Erases all named procedure(s).

ERN name(s)
Erases all named variables.

ERNS
Erases all variables from the workspace.

ERPS
Erases all procedures from the workspace.

NODES
Outputs number of free nodes.

PO name(s)
Prints definitions of named procedures.

POALL
Prints all definitions of procedures and names (variables).

POD condnumber
Prints WHEN demon condnumber currently in action.

PODS
Prints out all active WHEN demons.

PONS
Prints names and values of all variables.

POPS
Prints definitions of all procedures.

POTS
Prints title lines of all procedures.

RECYCLE
Performs a garbage collection.

Files

CATALOG device:
Displays names of all files on diskette. On a cassette, prints all the procedure definitions and names in the file.

ERF device:filename
Erases filename from device.

LOAD device:filename
Loads file called filename from the device into the computer.

SAVE device:filename
Saves workspace onto the device. If device is a printer, all procedures are printed.

SETREAD device:filename
Sets the device:filename from which data will be read by RC and RL.

SETREAD []
Closes the file that was opened with SETREAD.

SETWRITE device:filename
Starts the process of sending to filename on the device a copy of all the characters displayed on the screen.

SETWRITE []
Closes the file that was opened with SETWRITE.

Special Primitives

.CALL n
Transfers control to a machine language subroutine starting at address n (decimal).

.DEPOSIT n byte
Writes byte into address n (decimal).

.EXAMINE n
Outputs contents of address n (decimal).

.PRIMITIVES
Prints the list of Logo primitives.

.SETSCR n
Sets aspect ratio to n.

Special Keys

ATARI Key ( )
Reverse Video Key ( )

Break
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.

CTRL →
Moves the cursor one position to the right.

CTRL ←
Moves the cursor one position to the left.

CTRL ↑
Moves the cursor up to the previous line.

CTRL ↓
Moves the cursor down to the next line.

CTRL 1
Makes Logo stop scrolling until CTRL 1 is typed again.

CTRL A
Moves the cursor to the beginning of the current line.

CTRL CLEAR
Deletes text from the cursor position to the end of the current line.

CTRL INSERT
Opens a new line at the position of the cursor.

CTRL S
Splits screen: top for graphics, bottom for text.

CTRL T
Devotes entire screen to text.

CTRL V
Scrolls screen to next page in editor.

CTRL W
Scrolls screen back to previous page in editor.

CTRL X
Moves the cursor to beginning of editor.
CTRL Y
In the editor, inserts the contents of the delete buffer. Outside the editor, inserts the last command typed.

CTRL Z
Moves the cursor to end of editor.

* DELETE BACK S
Erases the character to the left of the cursor.

ESC
Completes editing and exits to top level.

F1, F2, F3, F4
Cursor control keys that can be programmed.

* RETURN
Completes the line and puts the cursor at the beginning of the next line.

* SHIFT DELETE BACK S
Deletes text from the cursor position to the end of the current line.

SHIFT INSERT
Opens a new line at the position of the cursor.

\ (Backslash)
Tells Logo to interpret the character that follows it literally as a character, rather than keeping some special meaning it might have. You have to backslash [ ], ( ), +, -, /, =, <, >, and itself.

SYSTEM RESET
Reboots Logo, erasing the memory space.
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