

# The LOGO Notebook

## Part I: Introduction and Basics

by Ron H. Mitchell

**EDITOR'S NOTE:** This is the first of a series of articles by Ron Mitchell, Editor of *The ADAM User Friendly Group newsletter*, covering *SmartLOGO* and the many aspects of programming in *SmartLOGO*.

Most people I've talked to figure that LOGO is mostly for kids. All this stuff about moving turtles and making ADAM "TOOT" out sounds seems somewhat beneath us adults. Not so, I'm here to tell you. While it's true that this language is one that young folks will easily gravitate to, it's also true that LOGO is a language that quite literally grows with you, until you achieve even more control over ADAM's many features than you can using SmartBASIC.

So let's get down to work. Each issue we'll cover some of the basics. Feel free to experiment, and by all means, read the LOGO manual. Unlike the SmartBASIC manual it does an excellent job of explaining all aspects of it's subject. You'll also find that most books on LOGO available in the library and in various bookstores will describe procedures and commands that you can use with ADAM's implementation of the language. So there's lot's of instructional material available. The only command I've found so far in a book that doesn't work is the command "LOCAL" used in Apple's version. This command designates a given variable for use only within a particular procedure. Outside that procedure it has no value. ADAM's LOGO will do the same thing but employs a slightly different means. We'll get to that later.

First a quick outline. The normal method employed by most texts is to introduce first the concept of "turtle graphics" and allow you to make use of LOGO's powerful graphics handling capability right off the top. What we're going to do here is slant our presentation so as to ease you from SmartBASIC to LOGO. We'll therefore be dealing with some of the less spectacular commands first, those which permit you to do some of the same things you've been doing in BASIC. But don't worry, it won't be long before we're into the pretty stuff.

For the balance of this article, we'll cover the following:

Terminology  
Immediate and Edit Modes  
Controlling Screen Output

So let's get at it. Load up your LOGO and read on.

### TERMINOLOGY

What follows is a short version of chapter 2 of the ADAM LOGO reference manual. You really need to give chapter 2 a thorough reading, for it lays the foundations in concept for subsequent material. Being somewhat of a novice myself, I've found that a complete understanding of the difference between a procedure, a primitive, a command, an operation, and a few other elements of LOGO grammar can help us to avoid troubles later on.

**1) PROCEDURE:** A procedure is like a small program or subroutine in SmartBASIC. It consists of series of LOGO commands or operations to be performed when it is invoked. A procedure has a name which can be anything you like except a word already used to call one of LOGO's primitives. You activate or call a procedure simply by typing in it's name, followed by <RETURN>.

**2) COMMANDS AND OPERATIONS:** These two terms are used to describe the two different types of LOGO primitives. Each one is either a command or an operation.

A LOGO command, like a SmartBASIC command, instructs ADAM to do something, and provides the required code to enable ADAM to carry out the task. An operation is a procedure that outputs a value to another procedure. Some LOGO procedures can be either commands or operations depending on the context in which they are used. You'll note that the manual, in discussing each of the LOGO primitives clearly identifies which are which. It's important to note the difference.

**3) PRIMITIVE:** A LOGO primitive is one of the commands or operations that is part of the language. You use these primitives to build your own procedures. Primitives are like reserved words in SmartBASIC.

**4) OBJECT:** An object is a list of LOGO instructions being either primitives or procedures you have defined. It is usually enclosed in square brackets [ ]. LOGO commands act upon or process an object or set of instructions. There are three types of objects: words, lists and numbers.

**5) WORD:** A word is simply a sequence of characters (no spaces allowed).

**6) LIST:** [Information enclosed in square brackets]

**7) INPUT:** Some LOGO commands and operations and eventually, the procedures you define will require inputs. Inputs are parameters which you pass to a procedure when calling it. Examples will follow later.

### IMMEDIATE AND EDIT MODES

As with SmartBASIC, there are two possible modes of operation once you've loaded the LOGO interpreter; immediate and edit.

Those of you who already have LOGO have no doubt tried just entering the various LOGO commands directly from the keyboard. Provided you know the command, it's syntax and the inputs it requires, you've no doubt done some rather startling things that would be impossible in SmartBASIC. For example, try this:

```
CS
HT
PU
SETPOS [-110 90] RT 90
PD
FD 220
```

And you find that a white straight line has been drawn across the top of the screen.

Now I said that we'd not get into turtle graphics right away, but this was just to illustrate the point.

What we did was to first clear the screen [CLEARSCREEN]. Then we turned the turtle off [HIDETURTLE]. We then issued a PENUP command to ensure that the turtle wouldn't put an unwanted line between his home position and the top left hand corner of the screen. After that we set the turtle's position at coordinates screen top left, put his pen down again, turned him right 90 degrees and sent him off to the other side a the screen drawing a line behind him.

All of that was done in the immediate mode, that is as we issued each command it was carried out.

# The LOGO Notebook

## Part I: Introduction and Basics

by Ron H. Mitchell

You'll note that we used the shortform of most of the commands. CS, for example, is short for CLEARSCREEN. Some commands do not have short forms, and the manual clearly states which do and which do not. We could also have written ourselves a procedure which would have accomplished the same result:

```
TO LINE
CS
HT
PU
SETPOS [-110 90]
FIT 90
PD
FD 220
END
```

At which point LOGO hesitates momentarily while it reviews your input, then confirms.

### **LINE DEFINED**

So in order to enter the edit mode, simple type <TO> followed by the name of the procedure you're defining. Enter your instructions and then hit <SmartKEY V1 >, which exits you from the edit mode.

While you're In the edit mode, you'll note that the question mark prompt before the flashing square cursor disappears. That's simply to serve as a reminder that your in the edit mode and that your instructions will form part of the procedure you're defining.

### **CONTROLLING SCREEN OUTPUT**

Now we're going to skip around a little. These are the commands I've found useful so far in my forays into LOGO when it comes to putting things on the screen.

**PRINT** or **PR**. Similar to the PRINT statement in BASIC. You can type: PR "RON or PR [RON] or PRINT [LEFT SQUARE BRACKET RIGHT HAND WIDGET]; and LOGO will faithfully print what is attar the quote or within the square brackets.

**CLEARSCREEN** or **CS**: Does just what it says. Clears the screen of both text and graphics.

**CLEARTEXT** or **CT**: Now here's a neat one. In LOGO you can have both text and graphics on the same screen at the same time. You can also clear off one or the other independently. CT clears text only.

**CLEARGRAPHICS** or **CG**: Removes graphics without affecting text.

**SETTEXT**: If you want to make LOGO imitate the GR or HGR modes of SmartBASIC, you can with this command set the top most line on which text will be printed. A <SETTEXT 20> would put all text on lines 20 to 24, leaving the rest of the screen for graphics. This command takes an input from 0 to 23. Unlike SmartBASIC, however, you can still print text in the graphics area with the SETCURSOR command.

**SETCURSOR**: Does what it says; ie. sets the cursor position to the coordinates in the list which follows. The syntax is: SETCURSOR [X Y]; Where X can be any value

from 0 to 28 and Y is a value from 0 to 23.

Note: Don't forget the square brackets or you'll be informed that:

### **SETCURSOR DOESN'T LIKE 12 AS INPUT**

Took me a while to figure that out.

These are the essential screen control commands. For reference in this discussion, I've used the ADAM LOGO manual, chapter 2, 9 and 10. You'll want to give these a thorough reading and then experiment a little for yourself. Next time we'll cover colors and variable assignments. We'll also design a LOGO procedure that uses what we've covered here and what we'll cover in the next article.

*--This article originally appeared in the December 1991 issue of NIAD and has been republished with permission from the original author, Ron Mitchell.*