THE POWER OF LOGO

Logo's developers designed it to be a full artificial intelligence language. It is described as having "no threshold"—preschool children can use it—and "no ceiling"—computer scientists can also use it for their work. To teach Logo without trivializing it, teachers and parents need to know some of the powerful ideas embedded in it so that their teaching provides windows to Logo's power.

We have identified twenty powerful ideas embedded in the Logo language, and we present them here in their simplest forms. As you read through the book and use Logo yourself, you will be able to expand and augment these descriptions with your own.

1. *The turtle:* The turtle is neither male nor female, young nor old. In many cultures it is a symbol of luck or wisdom. From Aesop we know that it is slow but sure, that it keeps moving and wins the race!

The turtle is a character to identify with, an object to think with. By identifying ourselves with the turtle, we can use our existing knowledge of how we move our own bodies around in space to move the turtle across the graphics screen. By teaching the turtle, we learn to program in Logo.

2. *Playing turtle, playing computer:* A common way to develop a design in turtle graphics is to play turtle. Pretend to be the turtle yourself, and walk yourself through a path on the floor that corresponds to the path you want the turtle on the screen to follow. While walking, observe the number of steps you take, the direction in which you are heading, the sequence of the instructions you are giving yourself to follow, and how you decide when to stop. Then translate the description into a set of Logo turtle commands.

Playing turtle is also a strategy for fixing a program that behaves unexpectedly or that you decide to change. The same concept is useful for debugging nongraphics programs. Play computer—put yourself in the computer's place and carry out each command in order, until you see where your procedure went wrong.

3. *The Total Turtle Trip Theorem:* Every time the turtle follows a series of commands and returns to its exact starting position, it must turn through an exact multiple of 360 degrees. With this theorem you can draw any regular shape, such as a square (4 × 90), a triangle (3 × 120), or a "circle" (36 × 10). And even if you're drawing a less regular shape such as a house, flower, or rocket, this theorem can be a powerful aid in designing and debugging any turtle graphics project.

4. *Drawing:* All people draw. Drawing is a natural expression of ideas. We use it to reflect the world we live in and record the events we participate in. Entering Logo by drawing with the turtle lets us extend our interest in drawing while learning to use a powerful new tool. It becomes a bridge between our familiar forms of expression and a new form of expression involving a computer.

5. *Exploration and discovery:* Books, movies, and television provide avenues for vicarious exploration and discovery. Logo provides a vehicle for genuine discovery. All Logo users are explorers, and share in the excitement of discovery. When exploring with Logo you can discover how to make a square or a circle, as part of a rich domain of mathematical possibilities. The key to success is the willingness to try ideas, free from excessive evaluation and preplanning.

6. *Theory building and problem solving:* We learn naturally by setting ourselves problems. Watch a baby learn to use a spoon for eating pudding. She wants to eat the pudding, and her theory is that she needs a spoon. She sticks the spoon into the pudding upside down, and the pudding slides off the spoon. After a number of trials, she will probably revise her theory to keep the spoon bowl-side-up. Then she gets to eat the pudding.

Logo learners make up theories about how to move the turtle, how far to turn it, or how to write a stop rule—always in the context of a particular task or problem. By testing their theories and observing what happens, they can make the necessary revisions and accomplish their objectives. By making descriptions and playing turtle, they develop strategies for assessing the discrepancies between their expectations and their results. Eventually they develop theories that work, or move on to more tractable problems.

7. *Describing and defining:* A Logo procedure is a description—in Logo commands—of something you want the turtle and/or the computer to do. When you want to use a particular sequence of commands more than once, you can use that sequence of commands to define a new command. Now you can use the new definition in the next level of exploration.

Description allows you to be in control, to make the computer do what you want it to. It is also a powerful tool for debugging. It is often said that if you describe a problem you can solve it.

8. Debugging: The process of finding, identifying, and eliminating bugs is at the heart of any Logo learning experience. Bugs are errors in a computer program, or discrepancies between what your program accomplishes and what you expected your program to do. A bug is not a mistake to be avoided. Rather, it is an opportunity to confront discrepancies in your own knowledge, to challenge your ideas of how to do something and replace them with something more accurate and effective.

Because making errors is such a common element in everyone's computer experience, the developers of Logo took special pains to create specific, useful messages that appear on the screen whenever Logo encounters a command that it can't carry out, or a piece of data that has no meaning. For example, Terrapin Logo has twenty-nine different error messages to guide you in finding, interpreting, and correcting your errors.

9. Procedures: We live most of our lives using procedures. Logo procedures—small recipes for the computer—are expressions of our thoughts and ideas. Ideally, each procedure does only one, clearly defined job. Procedures can be combined—just like building blocks—to create something more complex.

Some procedures are tools to be used over and over again—to draw a circle, for example, or to move the turtle over. Procedures can be thought of as Logo's helpers, and procedures can call other procedures as their helpers. The one procedure that bosses all the others is called a *superprocedure*. Those that are bossed are called *subprocedures*. A *procedure tree* is a map of who is in charge, who calls whom, and the sequence in which things happen.

People put procedures together by *top-down* planning—deciding in advance the procedures that will be needed to solve a particular problem; by *bottom-up* programming—building up procedure after procedure as you go along; and by *middle-out* problem solving—using a combination of the other two methods.

10. Naming: Naming anything is a powerful experience. When you name a boat, you have made it your own in a personal way. When you name a dog, you can make it come when you call.

In Logo you can give names to pictures, procedures, variables (data), and files. Clear naming schemes allow you to remember and locate your information and to reuse your previous work. These are essential to making full use of Logo as a tool for both problem solving and creative expression.

11. *Extensibility:* There is a saying among Logo users: "If Logo doesn't already do it, teach it how." Creating your own commands allows you to extend the language in any direction you wish. If you find typing difficult, create one-letter abbreviations. Or teach the computer to make circles, or to walk in random patterns. Or teach it to count down and blast off, or to sort the elements of a list. Each new procedure, each new word defined, becomes part of your own personal extension of Logo. Eventually you may be able to create Logo procedures that create other procedures, or even modify themselves.

12. *Variables:* A variable is a piece of information with a name. Logo procedures can use variables as inputs, allowing you to vary the size of a square, a house, or a flower; the rotation angle of a polygon; or the number of times a message is to be printed. The operations of addition, subtraction, multiplication, division, and comparison can be performed on variables with numerical values. Variables using words and lists can be combined into larger ones or chopped down into smaller ones. Variables let you create mathematical functions, compare answers to quizzes, and create interactive games.

13. *Repeating by iteration:* Logo's REPEAT command allows you to repeat a procedure for drawing a house in order to create a row of identical houses, or to put several lines of triangles together to form a tesselation. Repetition of a pattern, and finding the number of repeats necessary to produce a desired result, holds the key to discovering a number of mathematical relationships.

14. *Recursion:* Logo procedures routinely call on other procedures as helpers. When a procedure calls on another copy of *itself*—a procedure with the same name and same instructions—as a helper, the process is called *recursion*.

For example, one way to make a circle with the turtle is to repeat the instructions, FORWARD 1 RIGHT 1, 360 times. Another way is to create a procedure, CIRCLE, which moves the turtle FORWARD 1 RIGHT 1, and then calls on a helper called CIRCLE to finish the job. The next CIRCLE procedure moves and turns the turtle, and calls *another* CIRCLE procedure, and so on. The procedure looks like this:

```
TO CIRCLE
  FORWARD 1
  RIGHT 1
  CIRCLE
END
```

The key idea is that the instructions for a complex job can be quite simple. Pretend that you are a recursive procedure. Just do the simplest part of the job yourself, and get a helper to do the rest. And your helper has the same instructions you do! (It's like having each person in turn leave the dinner table with their own dishes. They wash them, dry them, put them away, and then call the next person at the table to do the same!) There are many complex tasks in data processing, and fascinating mathematical problems for which this approach is the simplest, most elegant, and most powerful. Logo provides an opportunity to experience recursion in its simplest form (so-called *tail recursion*, in which the recursive procedure call is the last one in the procedure), and to grow in understanding by using recursion in more complex forms later on.

15. *Conditionals:* Logo procedures use conditional commands to make decisions and choices by testing conditions and evaluating information. Does a certain variable have a value of 0? If so, stop the procedure. Did the user type a response of "yes"? If so, play the game again. And so on.

16. *Interaction between user and computer:* Logo learning is an interactive process. You begin by entering commands and observing the screen to see what happened. Did the computer do what you expected? (You can be sure that it did what you *told* it to!) If not, you can try again. If you type an incorrect command, Logo prints an error message, explaining why it could not carry out your instruction. Children say "the computer talked to me!"

Logo commands such as REQUEST (or READLIST) let you create interactive programs. REQUEST waits until a user types something. Then it can use what has been typed to create a variable, add new information to a list, or compare the response to a set of "correct answers."

17. *Information processing:* Logo stores information (data) in two forms—as words, and as lists. Lists can be lists of words, or lists of lists, which can be nested within other lists, nested within other lists, and so forth. Logo's information-processing commands allow you to take lists and words apart and put them back together, and to compare two pieces of information. With numerical data, Logo can add, subtract, multiply, divide, and compare.

Logo procedures can accept *inputs* from a user, and produce *outputs* in the form of words or actions on the screen. Procedures can also get their inputs from other procedures, and can output information to other procedures.

These capabilities allow you to design information-processing programs: quizzes, games, database programs. They also allow the construction of artificial intelligence programs, which learn new information or make logical inferences from a set of rules and facts.

18. *Thinking about thinking, learning about learning:* A key part of learning Logo is looking at your own thought processes and those of others. Logo is designed to make your thinking visible on the computer screen. Consider the process of estimating angles. If you don't turn the turtle enough, the result is immediately obvious. You can see it's not far enough. If you develop an effective strategy, such as turning the turtle in ever-decreasing amounts until it is pointing in the right direction—RIGHT 110 (too far), LEFT 30 (too far the other way), RIGHT 15 (too far again), LEFT 5 (just right!)—you can observe your strategy, as well as its results. Then you may choose to use the same strategy again.

19. *Microworld:* It is a small, safe, well-defined learning environment, which includes intriguing phenomena, powerful ideas and makes some connection to the rest of a learner's world. Learning in a microworld occurs by open-ended exploration, not by focused instruction. Logo itself is a mathematical microworld for learning geometric concepts. By creating shapes and patterns, repeating and combining them, a learner can discover and use important geometric principles. A teacher becomes a helper and a guide, rather than the sole source of information and instruction.

A microworld permits the user to learn by *being* a mathematician, a poet, an artist, rather than by *learning about* those subjects. Seymour Papert has described learning in a microworld as more like getting to know a person than like learning a set of facts about that person.

20. *Modes of computer use:* Logo has two main modes of use: *direct* or *command mode,* in which the computer carries out your commands immediately, and *edit mode,* in which your commands are named and stored ready to be used later, if called by name. (No actions occur in edit mode.) Understanding both modes and being able to move between them flexibly is one key to using the full power of Logo.