

A diamond-shaped title box with a double border, containing the text "TI LOGO Lingo". The diamond is centered and flanked by horizontal bars with a black and white checkered pattern.

TI LOGO Lingo

SETCOLOR *:word*

SETCOLOR *number*

Gives a color to the Turtle, a sprite or a tile.

SETHEADING *:word*

SETHEADING *number*

Gives the Turtle or a sprite the direction in which to move.

SETSPEED *number*

Gives a sprite a speed.

Sprite

A graphic that has the capability of motion.

In order to be seen, a sprite must be given the attributes of COLOR and SHAPE. There are 32 sprites available in TI LOGO.

State of the Pen

The four conditions which can apply to the Turtle's pen — PENDOWN, PENERASE, PENUP, AND PENREVERSE.

Subprocedure

A procedure that is called by another procedure.

Superprocedure

A procedure that calls other procedures (subprocedures).

SX *number*

SY *number*

Used to position sprites on the display on imaginary x- and y-coordinate planes.

TELL *listener*

Indicates who is the current listener.

THAW

Restarts all the sprites on the display that are stopped by the command FREEZE.

TO *name*

Used to teach the computer.

Turtle

A triangular sprite that has the ability to move, draw, and create geometrical designs.

Variable

A name to which a value, a word, or a list is assigned by means of the CALL or MAKE command or by including it in the name of a procedure.

WAIT *number*

Causes the computer to pause for the specified number times 1/60th of a second.

Work space

The memory area used to design shapes or characters, give commands, or enter procedures; everything that exists in the Random Access Memory (RAM) of the computer after it has been turned on.

Wrapping

A process by which the Turtle or sprites encircle the display, appearing on the screen again on the opposite side.

XCOR

Returns the x-coordinate of the active sprite or Turtle.

YCOR

Returns the y-coordinate of the active sprite or Turtle.

YOURNUMBER

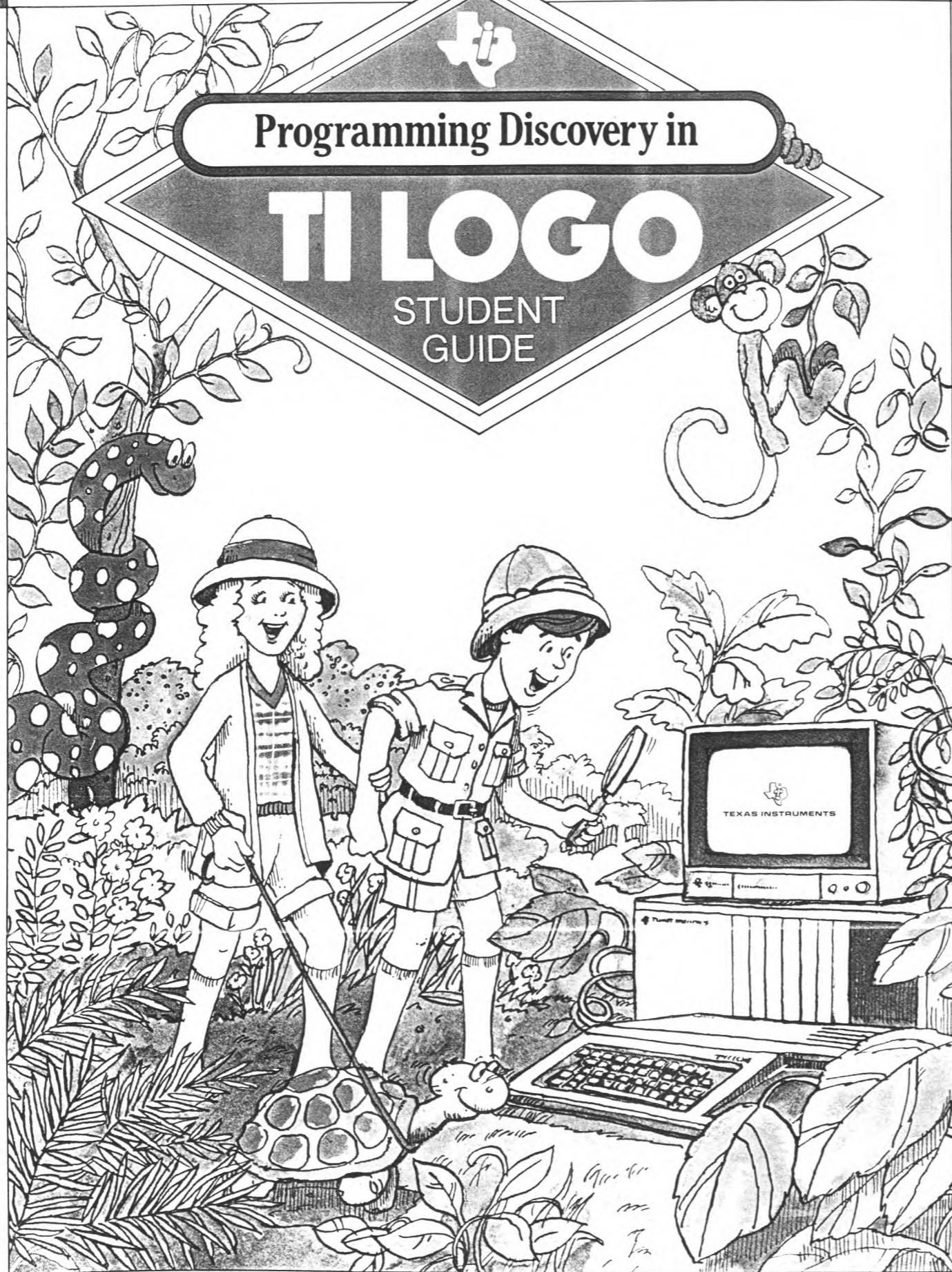
Returns the number of the current sprite.



Programming Discovery in

TI LOGO

STUDENT
GUIDE



Computer Advantage Club



Programming Discovery in

TI LOGO

STUDENT
GUIDE

TI LOGO can help
You discover a lot about
Yourself...



- ... that you are creative
- ... that you are a problem solver
- ... that you can learn to program

Have a great time discovering with TI LOGO!

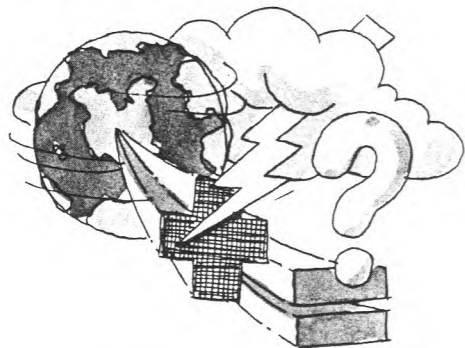
Note to Parents

If you're like most parents, when you enroll your child in a course, you're interested in what your child is going to learn and why. Perhaps you have a few questions about TI LOGO. Here are some we have anticipated.

What is LOGO?

LOGO is called a high-level programming language. This means that an individual can communicate with the computer using easy-to-understand, everyday words.

LOGO was designed by Seymour Papert and staff members of the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology. The development of LOGO is an interesting story told in Papert's book, **Mindstorms**. His ideas about how children learn are an essential ingredient in how the language was developed.



At the heart of LOGO is the idea of creating computer-based environments. In these environments, the learning of mathematics and other subject areas can take place in a natural way — through exploration, trial and error, and discovery. With LOGO, children create their own learning environment and are in control of what the computer does.



What is TI LOGO?

TI LOGO is the result of a cooperative effort between MIT and Texas Instruments to adapt the LOGO computer language to a low-cost microcomputer system — the TI Home Computer.

There are two modes in TI LOGO through which children "teach" the computer: the Turtle mode and the Sprite mode. In the Turtle mode, children can use a triangular shape, called the Turtle, to draw geometric figures and designs and experiment with lines and proportions.

In the Sprite mode, a world of animated shapes appears. The children create these shapes or select a shape that the computer already

knows — a plane, truck, rocket, ball, or box. Once the sprites have shape, they can be given other features — color, speed, heading, and direction.

What are the benefits of TI LOGO?

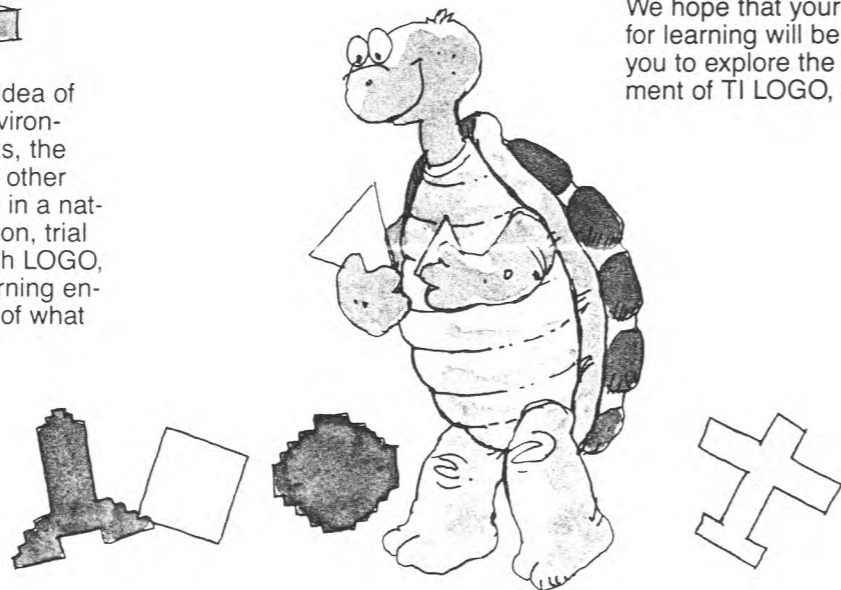
Programming with the TI LOGO language encourages children to:

- develop problem-solving and logical-thinking skills.
- realize self-direction and creativity.
- learn a wide range of math skills — especially in geometry.
- use the computer as a tool for communication and practical problem solving.

What can my child expect to learn in this course?

This course is a first-level programming course. In the time allotted, your child will be introduced to basic TI LOGO programming skills. But because TI LOGO is a "language," the terminology and concepts take time for assimilation and exploration, and there is much more to learn and explore beyond this level.

We hope that your child's enthusiasm for learning will be an incentive for you to explore the learning environment of TI LOGO, too.



VOL. I NO. 1

TI LOGO DAILY NEWS

FREE COPY

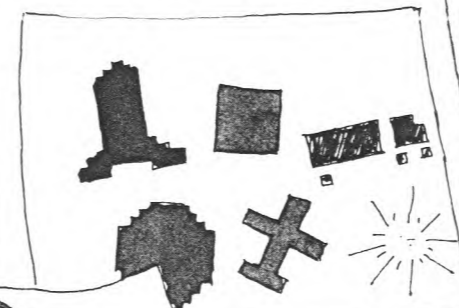
TI LOGO INTRODUCES THE TURTLE & SPRITES

~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~



Sprites are Invisible

~~~~~  
~~~~~  
~~~~~



## TI LOGO Computer language Great for Kids

LOGO IS LOTS OF FUN!

~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~



## Kids Create Fantastic Turtle Designs

~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~

# Key Computer Keys



The TI-99/4A keyboard looks a lot like a typewriter keyboard, doesn't it? It has letter and number keys, keys with special symbols, SHIFT keys, and a SPACE BAR. But it also has other keys used in TI LOGO — the FCTN key and the ENTER key. You can do powerful things using this keyboard — but you need to know what keys are used with TI LOGO and how to use them.

## SHIFT

Use a SHIFT key to type the top symbol on any key with two symbols. To do this, hold down a SHIFT key at the same time you press a key with two symbols. What would you press to type a dollar sign? Answer: Hold down a SHIFT key and then press the 4 key.

## ENTER

Use this key when you're through telling the computer what you want it to do.

## FCTN

(FCTN is an abbreviation for "function.") You can use the function key with other keys in three ways:

1. To type a symbol found on the front of some of the keys like this one:
2. To move the cursor using the arrow keys — but only in the Edit mode. To

do this by pressing the key while holding down the FCTN key. How would you type a question mark? Answer: Try it. Just hold down the FCTN key and press the I key.

do this, hold down the FCTN key and press the needed arrow key.

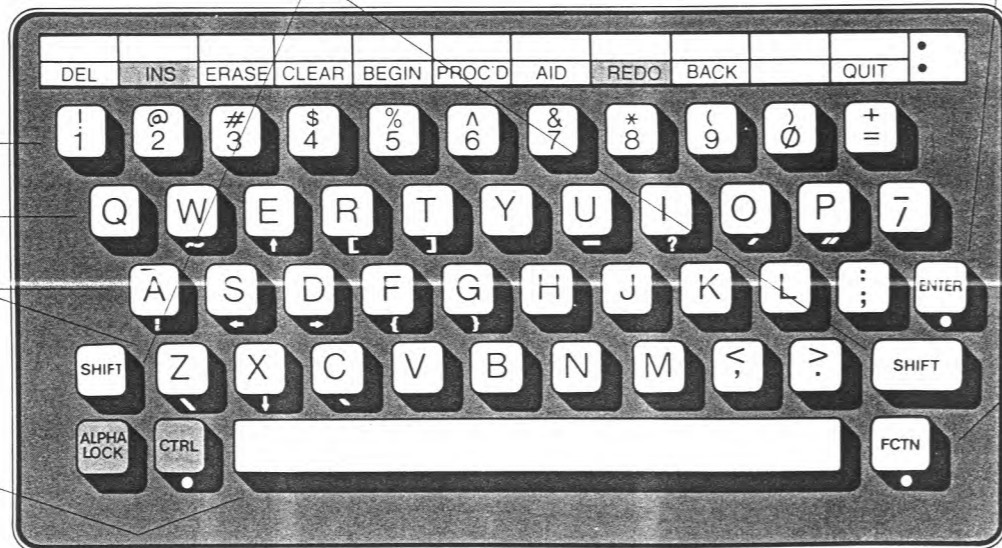
3. To use the functions listed on the insert strip at the top of the keyboard. Just hold down the FCTN key while you press the key directly under the name of the function.

## Letter and Number Keys

Use these keys to type letters, words, and numbers used in TI LOGO.

## Space Bar

Press this key (long bar at the bottom of the keyboard) to leave spaces when you type.



The Keyboard

The chart at the right shows you the modes in which these functions can be used.

| FUNCTION | TURTLE MODE | SPRITE MODE | EDIT MODE |
|----------|-------------|-------------|-----------|
| BEGIN    |             |             | ✓         |
| PROC'D   |             |             | ✓         |
| ERASE    | ✓           | ✓           | ✓         |
| CLEAR    |             |             | ✓         |
| DEL      |             |             | ✓         |
| BACK     | ✓           | ✓           | ✓         |
| AID      | ✓           | ✓           | ✓         |
| QUIT     | ✓           | ✓           | ✓         |

REDO, INS, ALPHA LOCK, and CTRL are not used in TI LOGO.

## The Turtle Mode

What kind of turtle has no legs and can go faster than a rabbit?

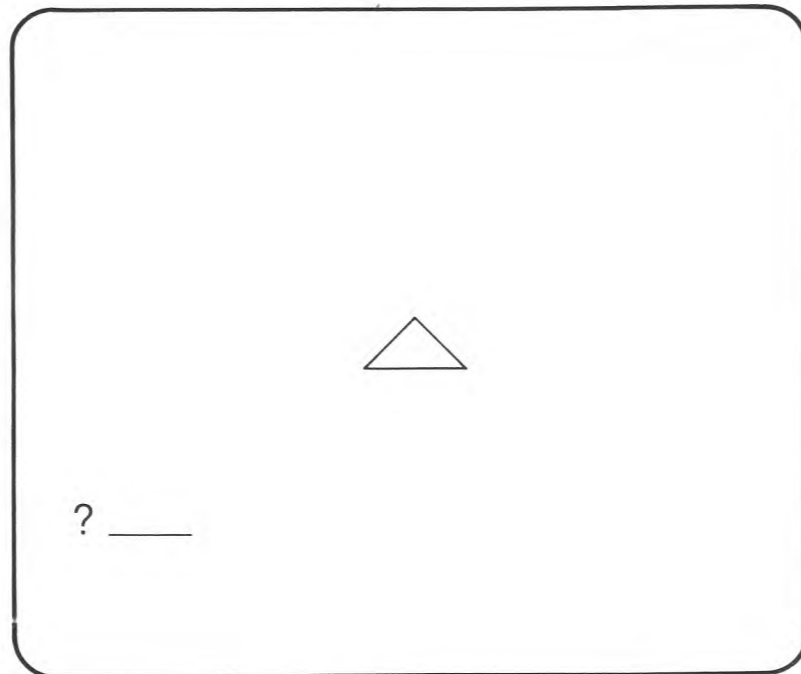


A hare-less turtle?



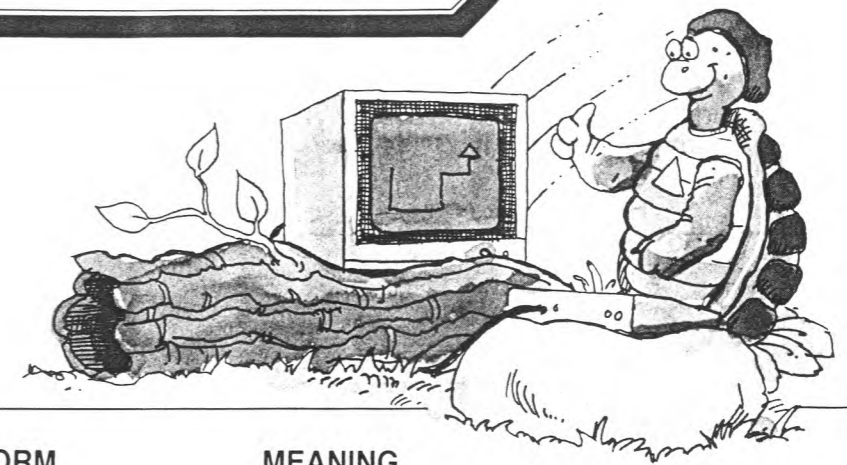
**NO,  
THE TI LOGO TURTLE!**

The Turtle in TI LOGO is a triangle. It appears in the middle of your computer screen at the command, TELL TURTLE. With the Turtle, you can create many imaginative designs. Here's what the Turtle screen looks like after you enter, TELL TURTLE.



## Turtle Commands

Do you have a design you would like the Turtle to draw? Use these commands to tell it what to do.



| COMMAND        | SHORT FORM     | MEANING                                                                               |
|----------------|----------------|---------------------------------------------------------------------------------------|
| TELL TURTLE    | No short form  | Use this command to enter the Turtle mode.                                            |
| FORWARD number | FD number      | Moves the Turtle forward in the direction it is pointing.                             |
| BACK number    | BK number      | Moves the Turtle back the number of Turtle steps you select.                          |
| RIGHT number   | RT number      | Turns the Turtle to the right.                                                        |
| LEFT number    | LT number      | Turns the Turtle to the left the number of steps you choose.                          |
| PENUP          | PU             | Moves the Turtle across the screen without drawing a line.                            |
| PENDOWN        | PD             | Tells the Turtle to place its pen down and to be ready to draw.                       |
| PENERASE       | PE             | Tells the Turtle to erase any lines it passes over.                                   |
| PENREVERSE     | PR             | Erases lines it travels over and draws a new line where one hasn't been drawn before. |
| HIDETURTLE     | HT             | Makes the Turtle disappear but still allows it to draw.                               |
| SHOWTURTLE     | ST             | Makes the Turtle reappear after it is hidden by the command HIDETURTLE.               |
| CLEARSCREEN    | CS             | Clears the screen of both the Turtle designs and the words.                           |
| NOTURTLE       | No short form. | Leaves the Turtle mode.                                                               |

## My Turtle Design

Create a Turtle design! Here's a place for you to write the steps for that special design.

---

---

---

---

---

---

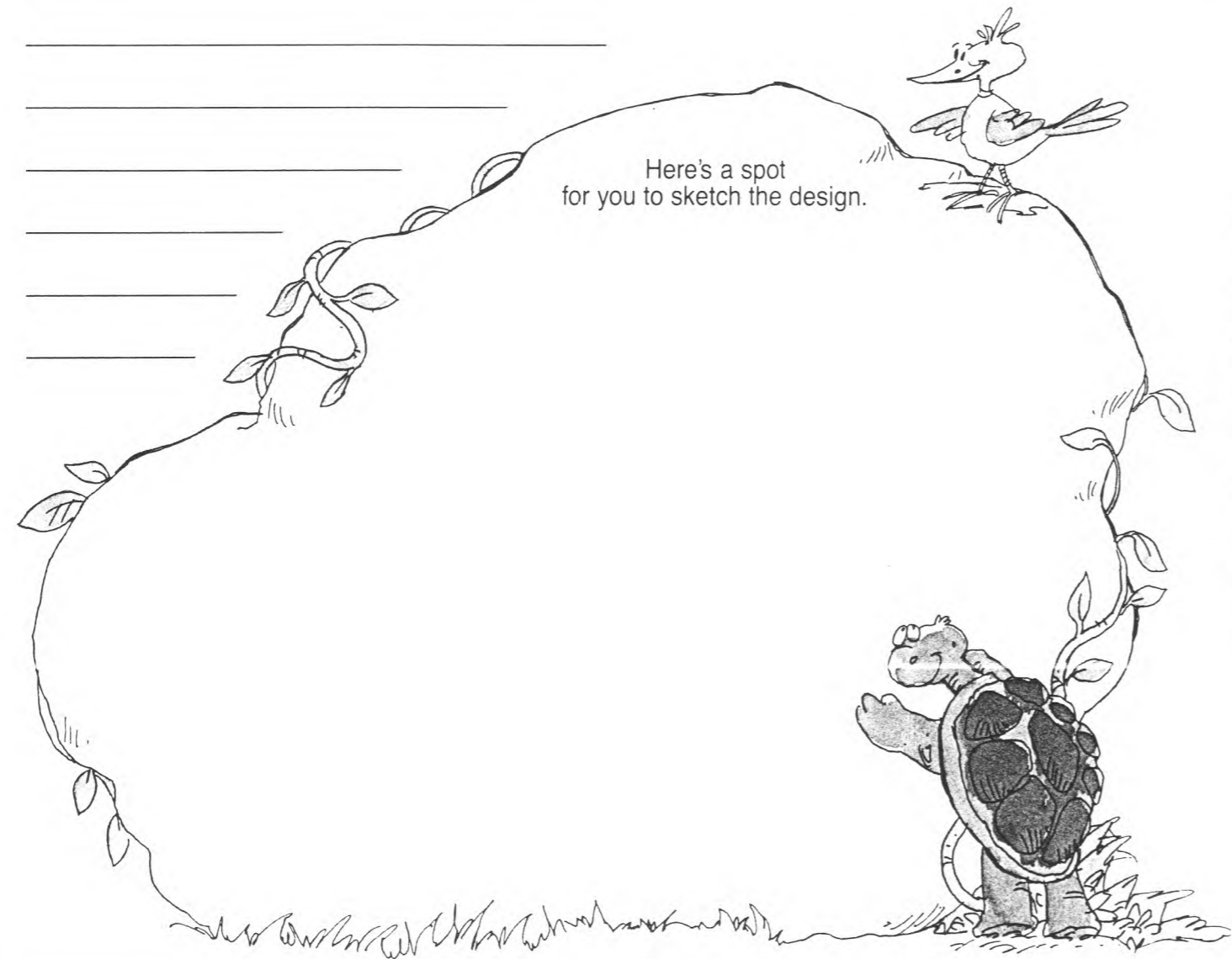
---

---

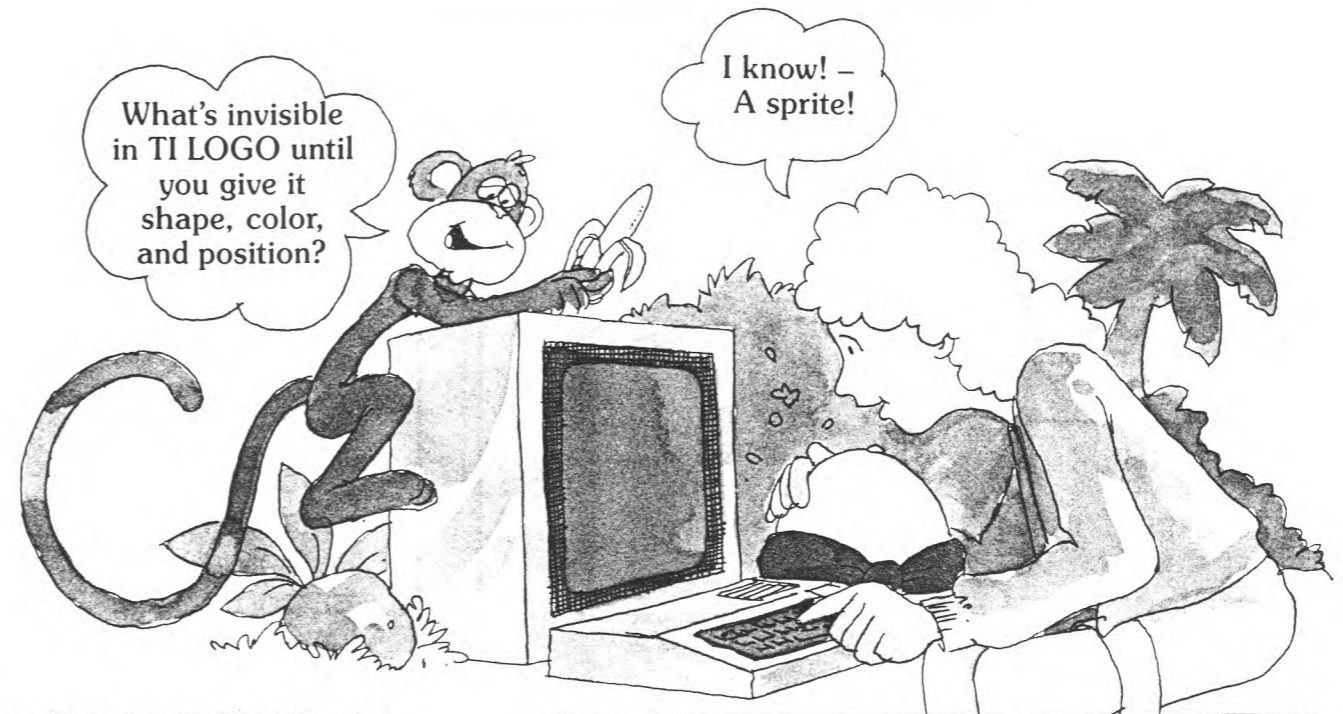
---

---

Here's a spot for you to sketch the design.



## The Sprite Mode



A sprite is an invisible character that can be seen only when you give it the attributes of shape, color, and position. You can also give a sprite direction and speed.

When you first turn on the computer and choose TI LOGO, WELCOME TO TI LOGO appears just above the cursor, and you are in the Sprite mode.

If you have been working with the Turtle and decide you would like to work only with sprites, type NOTURTLE and then press ENTER. The computer screen looks like this, with the cursor blinking in the top left-hand corner:

? \_

# Sprite Commands

Here are sample sprite commands for you to try. Remember, press ENTER after each command.

Use these commands to:






Get a sprite to listen to you.

TELL SPRITE 1

Put a sprite in the middle of the screen.

HOME

Give a sprite shape.

|                                                                                   |               |
|-----------------------------------------------------------------------------------|---------------|
|  | CARRY :PLANE  |
|  | CARRY :TRUCK  |
|  | CARRY :ROCKET |
|  | CARRY :BALL   |
|  | CARRY :BOX    |

Give a sprite color.

```

SETCOLOR :CLEAR
SETCOLOR :BLACK
SETCOLOR :GREEN
SETCOLOR :LIME
SETCOLOR :BLUE
SETCOLOR :SKY
SETCOLOR :RED
SETCOLOR :CYAN
SETCOLOR :RUST
SETCOLOR :ORANGE
SETCOLOR :YELLOW
SETCOLOR :LEMON
SETCOLOR :OLIVE
SETCOLOR :PURPLE
SETCOLOR :GRAY
SETCOLOR :WHITE
    
```

Give a sprite speed.

```

SETSPEED 15
SETSPEED 72
SETSPEED 100
SETSPEED 0
    
```

Give a sprite heading.

```

SETHEADING :NORTH
SETHEADING :EAST
SETHEADING :SOUTH
SETHEADING :WEST
    
```

Here are some shortcuts that make sprite commands even easier!

Get a sprite to listen to you.

You can call a sprite by its number. There are 32 sprites (numbers 0-31).

TELL 1






Or choose your own sprite.

Put a sprite in the middle of the screen.

HOME

Give a sprite shape.

You can use numbers instead of words for the shapes. When you do this, you don't need dots (:). In addition to the five predefined shapes in TI LOGO, there are 21 blank grids on which you can make shapes using the MAKESHAPE command.

|                                                                                     |         |
|-------------------------------------------------------------------------------------|---------|
|  | CARRY 1 |
|  | CARRY 2 |
|  | CARRY 3 |
|  | CARRY 4 |
|  | CARRY 5 |

Give a sprite color.

You can use a short form for the SETCOLOR command — SC. And you have a choice of using a number or a word for a color.

|            |       |
|------------|-------|
| SC :CLEAR  | SC 0  |
| SC :BLACK  | SC 1  |
| SC :GREEN  | SC 2  |
| SC :LIME   | SC 3  |
| SC :BLUE   | SC 4  |
| SC :SKY    | SC 5  |
| SC :RED    | SC 6  |
| SC :CYAN   | SC 7  |
| SC :RUST   | SC 8  |
| SC :ORANGE | SC 9  |
| SC :YELLOW | SC 10 |
| SC :LEMON  | SC 11 |
| SC :OLIVE  | SC 12 |
| SC :PURPLE | SC 13 |
| SC :GRAY   | SC 14 |
| SC :WHITE  | SC 15 |

Give a sprite speed.

You can use the short form for SETSPEED — SS. Not only can you tell a sprite to move forward a certain speed, but with a negative number you can tell the sprite to move backward a certain speed. Here are sample commands. Can you discover the speed limits in TI LOGO?

```

SS -20
SS 100
SS -90
SS 33
    
```

Or choose your own speed.

Give a sprite heading.

You can use the short form for SETHEADING — SH. You can give a sprite a heading with numbers as well as words.

|        |           |
|--------|-----------|
| SH 0   | SH :NORTH |
| SH 90  | SH :EAST  |
| SH 180 | SH :SOUTH |
| SH 270 | SH :WEST  |
| SH 45  | No words  |
| SH 318 | No words  |

Or choose your own heading.



# SET HEADING

What did the sea captain give his crew when he wanted them to "get lost"?



A crew-cut?

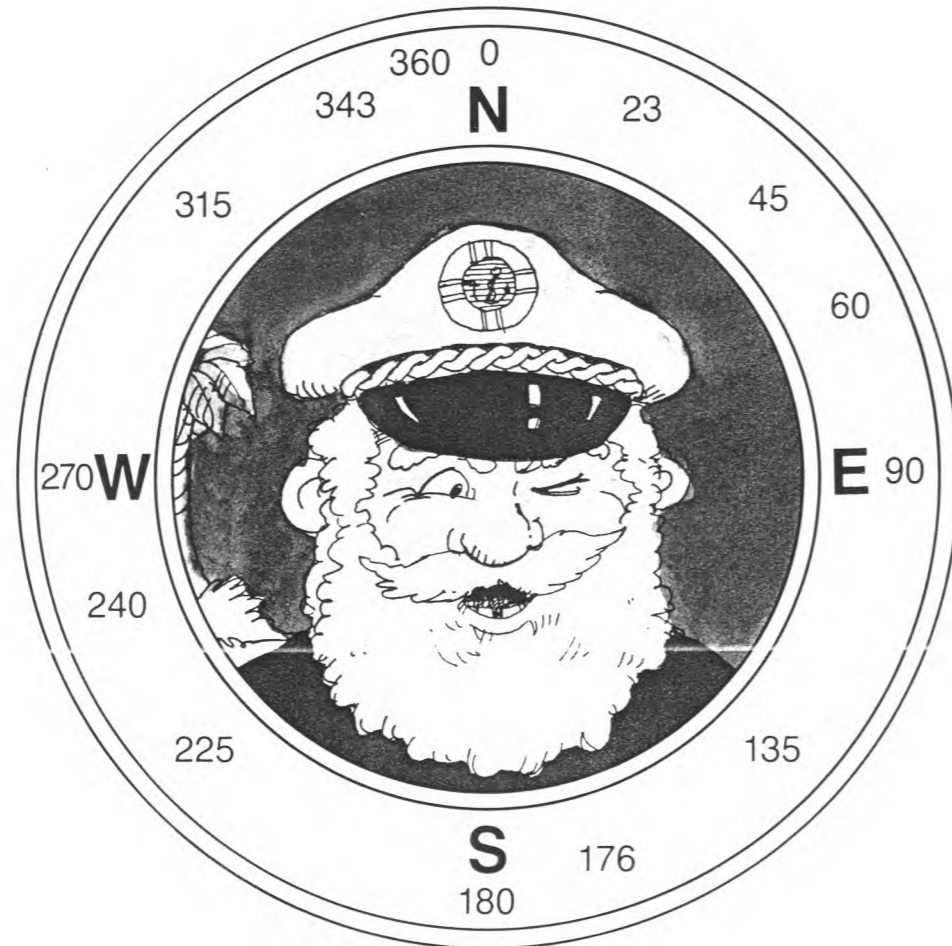


Uh, uh.  
A broken  
compass.



You can tell sprites which way to go with the same directions you find on a compass. Think of the points of the compass as being

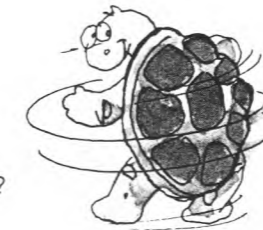
located on the computer screen, with NORTH or 0 always at the top of the screen.



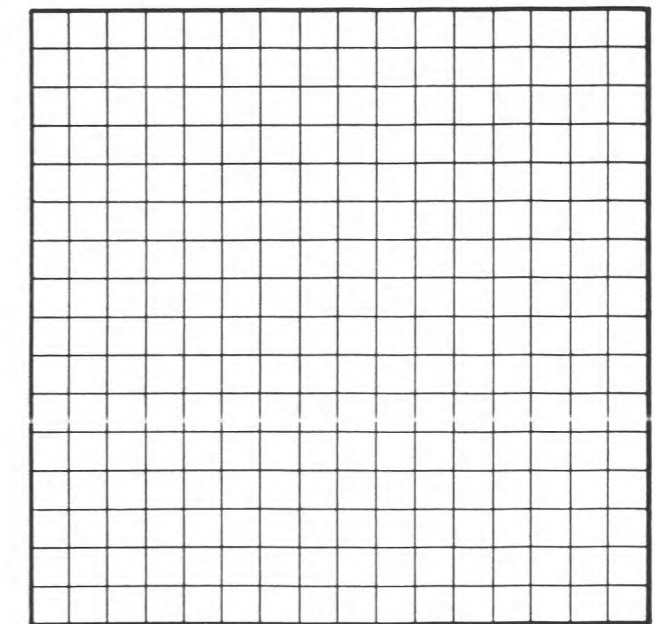
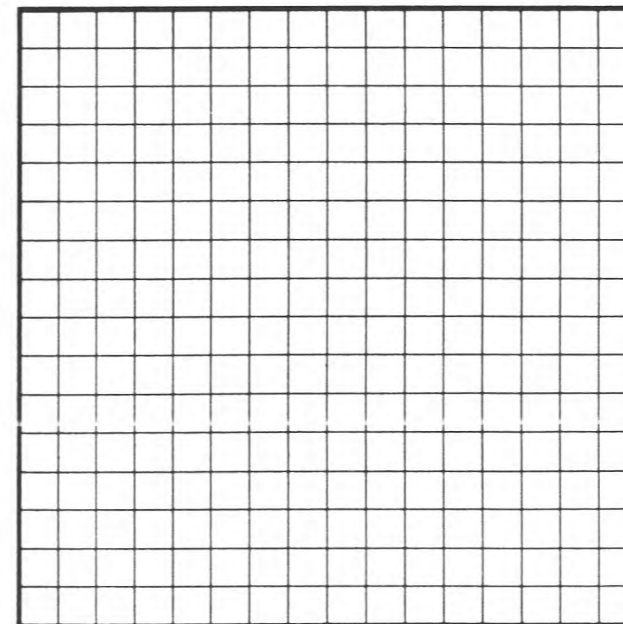
# MAKESHAPE Command

With the MAKESHAPE command you can make shapes for sprites to carry. In addition to the grids for the five shapes the computer already knows, there are 21 blank grids on which you can create shapes.

Here are two grids you can use to plan shapes of your own. Darken the squares that make the design you would like. Below each grid is a blank on which you can write the number of the grid you use when you work on the computer.



Here's a tongue twister to repeat five times as fast as you can — What kind of shape would a makeshape make if a makeshape could make a shape?





# My Sprite Procedure

Here's a special place to write down any procedure you would like to keep ...

---

---

---

---

---

---

---

---

---

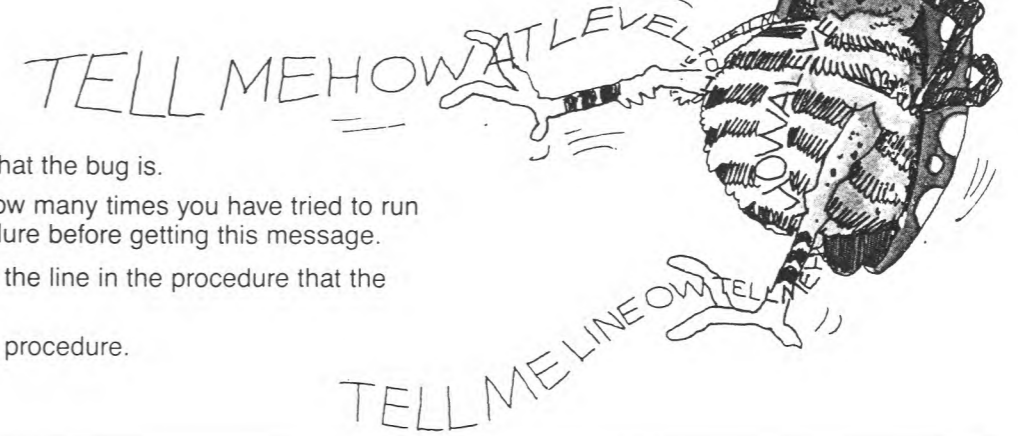
---

... and a place to sketch what happens on the screen in your procedure.

# Get the Message

TI LOGO is programmed to give you helpful messages when you don't give enough information or the right kind of information.

For example, if a line in your procedure has a bug in it, you see the message:



- TELL ME HOW TO ... tells you what the bug is.
- AT LEVEL ... tells you how many times you have tried to run this procedure before getting this message.
- LINE ... shows you the line in the procedure that the bug is on.
- OF ... names the procedure.

Here are other messages you may see:

## Procedure Messages

| Message                     | Explanation                                                         |
|-----------------------------|---------------------------------------------------------------------|
| ELSE IS OUT OF PLACE        | In an IF ... THEN ... ELSE command, ELSE is out of place.           |
| ... MUST BE IN A PROCEDURE  | Information must appear in a procedure.                             |
| PROCEDURE NOT BEING DEFINED | END appears as the last command without TO being the first command. |
| THEN IS OUT OF PLACE        | In an IF ... THEN ... ELSE command, THEN is out of place.           |
| TOO MANY SUBLISTS           | Too many brackets in one procedure.                                 |

## General Messages

| Message                   | Explanation                                                                    |
|---------------------------|--------------------------------------------------------------------------------|
| CAN'T                     | Computer can't do what you told it to do.                                      |
| CHOKED!                   | Computer is out of memory and must be turned off and then on to continue work. |
| DOESN'T LIKE ... AS INPUT | Computer doesn't like a number, word, or list.                                 |

| Message                     | Explanation                                                              |
|-----------------------------|--------------------------------------------------------------------------|
| ... HAS NO VALUE            | No value was given to something that needed a value.                     |
| OUT OF SPACE                | Computer is out of memory space.                                         |
| TELL ME MORE                | Computer needs a number or a variable to go with the command.            |
| TELL ME WHAT TO DO WITH ... | Computer understands instructions but doesn't know what to do with them. |

## Turtle Messages

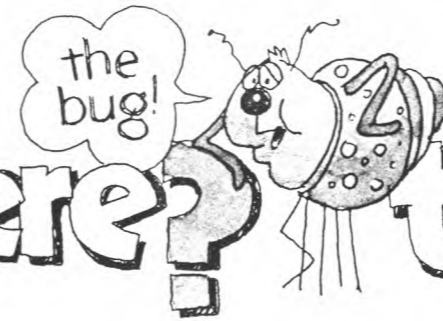
| Message    | Explanation                                                         |
|------------|---------------------------------------------------------------------|
| OUT OF INK | Turtle has no more tiles to draw on. To continue, clear the screen. |

## Symbol Messages

| Message             | Explanation             |
|---------------------|-------------------------|
| MISMATCHED BRACKETS | Brackets need to match. |

# Knock! knock! who's there? the bug who?

The bug in my procedure!



When you write a procedure, sometimes it doesn't work like you planned. To change the procedure, you need to edit, or "debug," it. To do this, type the command TO, a space, the name of the procedure, and press ENTER. You are in the Teaching or Edit mode. The screen turns green and looks like this:

**Teaching mode**

```
TO (PROCEDURE-NAME)
END
```

**Edit mode**

```
TO BOX
REPEAT 3 [FD 30 RT 90]
END
```



This procedure has a bug. See if you can debug it.

The following keys are used in the Teaching or Edit mode. When you use the FCTN key, remember to hold it down while you press the key with the function you need.

**These commands help you move the cursor from one place to another.**

- |             |                                                                                                                                       |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------|
| ENTER       | When the cursor is at the end of a line, pressing this key gives you a blank line and moves the cursor to the beginning of that line. |
| FCTN BEGIN  | Moves the cursor to the beginning of the line.                                                                                        |
| FCTN PROC'D | Moves the cursor to the end of the line.                                                                                              |
| FCTN ↑      | Moves the cursor up one line.                                                                                                         |
| FCTN ←      | Moves the cursor one space to the left.                                                                                               |
| FCTN →      | Moves the cursor one space to the right.                                                                                              |
| FCTN ↓      | Moves the cursor down one line.                                                                                                       |

**These commands help you change what you have written.**

- |             |                                                                                                                                                                                                                                                        |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FCTN ERASE  | <ul style="list-style-type: none"> <li>Erases the character or space one space to the left of the cursor.</li> <li>If the cursor is under the first character of a line, ERASE moves the entire line up, adding it to the end of that line.</li> </ul> |
| FCTN CLEAR  | Removes the character or space above the cursor and everything to the right of the cursor.                                                                                                                                                             |
| FCTN DELETE | <ul style="list-style-type: none"> <li>Deletes the character or space above the cursor.</li> <li>If the cursor is at the end of a line, DELETE moves the next line up, adding it to the end of that line.</li> </ul>                                   |

**This command takes you out of the Teaching or Edit mode.**

- |           |                                             |
|-----------|---------------------------------------------|
| FCTN BACK | Takes you out of the Teaching or Edit mode. |
|-----------|---------------------------------------------|

Have fun trying out all these keys in the Teaching or Edit mode!



## Disappearing Sprites Example

Here are the commands that make all the sprites disappear.

```
TELL :ALL
SS O
HOME
SC O
SH O
FD 97
```

You can write a procedure using these commands. The procedure is often called VANISH, and people use it a lot when they are programming. Simply type TO, a space, and VANISH. Then press ENTER and enter the commands. Remember to press BACK.



## Procedures

Here's an example of a Turtle procedure. Remember that you need to be in the Turtle mode to write the procedure.

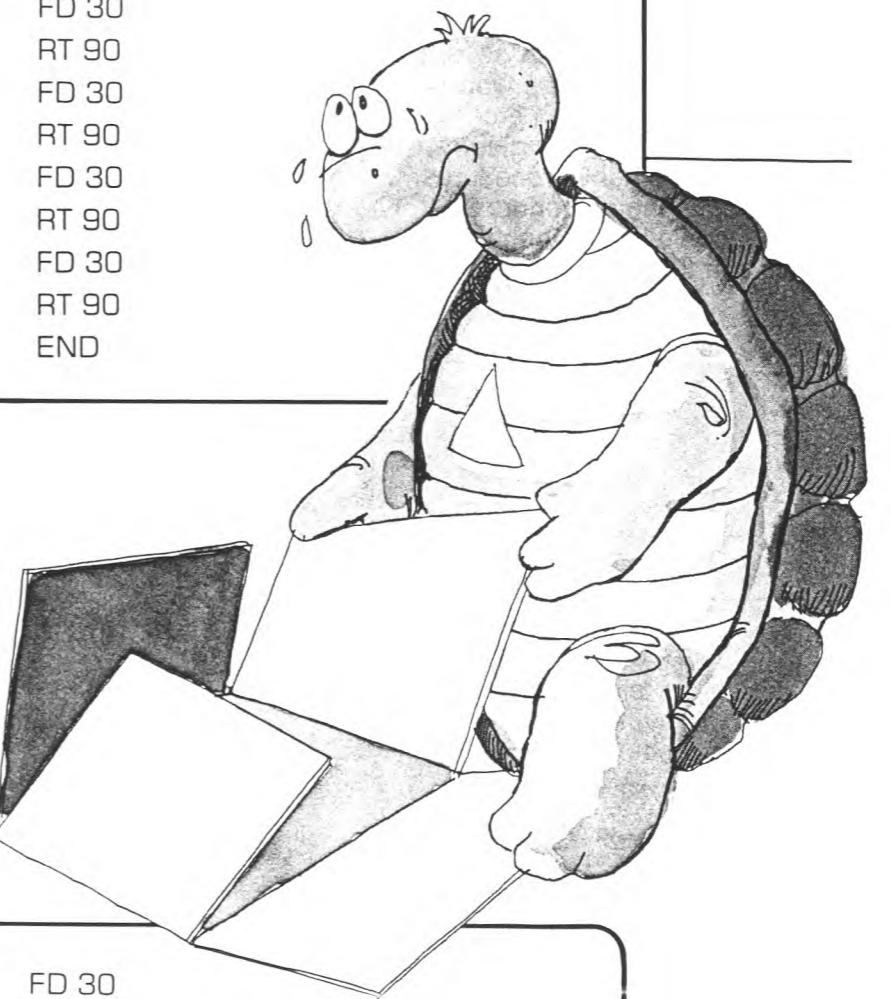
You can make this procedure simpler to write. Notice that to make the box you repeat the commands FD 30 RT 90 four times. Try editing the box procedure using the REPEAT command. Notice that the steps to be repeated are in brackets.

```
TO BOX2
REPEAT 4 [FD 30 RT 90]
END
```

You can make a procedure part of a larger procedure. In other words, it can become what is called a subprocedure. The procedure BOX2 is used three times as a subprocedure of the procedure BLOCKS. Try it out!

Try writing a procedure of your own that has subprocedures.

```
TO BOX1
FD 30
RT 90
FD 30
RT 90
FD 30
RT 90
FD 30
RT 90
END
```



```
FD 30
LT 90
SC :GREEN
BOX2
RT 90
FD 30
LT 90
FD 30
SC :PURPLE
BOX2
END
```

## Variables

Sometimes you might want to draw the same shape more than once but in different sizes. For example, if you want to draw several triangles of different sizes, you could do this by writing a procedure for each size triangle. Instead of writing many procedures, you can write one procedure that uses a variable for the length of the sides of the triangle.

Here is an example of a Turtle procedure that draws a triangle:

```
TO SAMPLE
  FD 30
  RT 120
  FD 30
  RT 120
  FD 30
  END
```

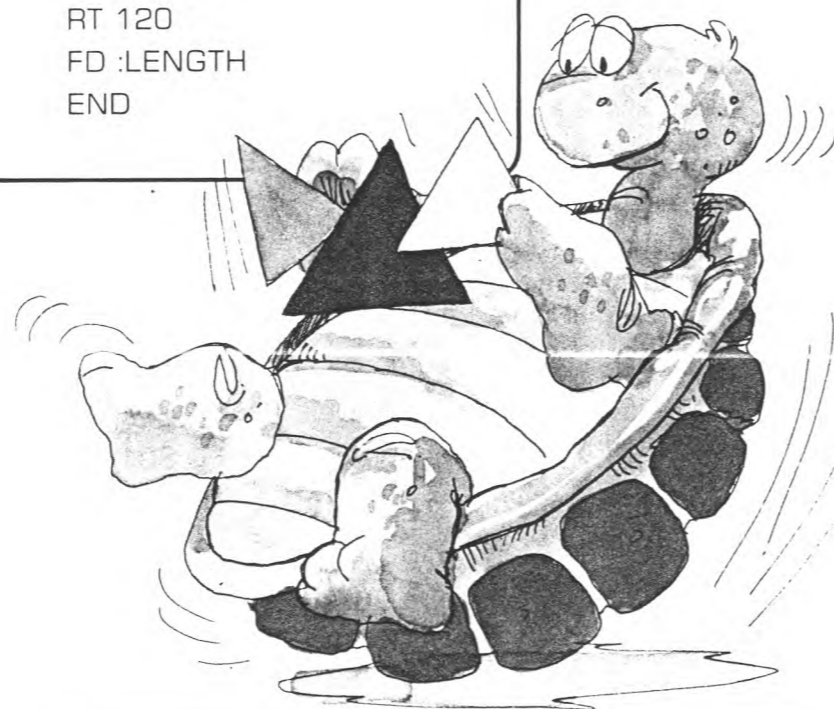
Here is a Turtle procedure that includes a variable — LENGTH.

```
TO TRIANGLE :LENGTH
  FD :LENGTH
  RT 120
  FD :LENGTH
  RT 120
  FD :LENGTH
  END
```

Now when you tell the computer, TRIANGLE, you must also tell it how long the side will be like this:

```
TRIANGLE 16
```

Notice that a variable is just a name you can use instead of a number. Try making larger and smaller triangles. Can you write a procedure for a box using a variable?



## Comments in Procedure Statements

It's good programming form to write comments about the statements in your procedure. Comments are little notes to yourself. They do not affect your procedure. They help you or someone else who uses your procedure to tell at a glance what a statement in your procedure does.

To write a comment, use the semicolon (;) after any statement about which you wish to comment. Take a look at the comment in the TRIANGLE procedure.

```
TO TRIANGLE :LENGTH
  FD :LENGTH ; WITH THIS VARIABLE,
  ANY NUMBER CAN BE GIVEN TO CHANGE THE
  LENGTH OF THE SIDE OF THE TRIANGLE
  RT 120
  FD :LENGTH
  RT 120
  FD :LENGTH
  END
```



## Toolbox Procedures

### Recursion

Recursion is the process of making a procedure repeat itself. You can make recursion happen by writing the name of the procedure as a statement line in the procedure. Here's an example.

First, write this Turtle procedure, called SNOWFLAKE. Remember that you need to be in the Turtle mode to write the procedure.

```
TO SNOWFLAKE
TELL TURTLE
SC :WHITE
RT 90
FD 20
REPEAT 2 [RT 90 FD 10]
RT 90
FD 20
END
```

Then edit SNOWFLAKE by adding SNOWFLAKE before END. Now you've done it. You've made recursion happen! The SNOWFLAKE procedure repeats itself. (Press FCTN BACK to stop the procedure.)

Then, to make more interesting things happen on the screen, follow these steps:

1. Edit SNOWFLAKE again by adding LT 30 before SNOWFLAKE.

2. Write a BOX procedure:

```
TO BOX
REPEAT 4 [FD 10 RT 90]
END
```



3. Now edit SNOWFLAKE by adding BOX before SNOWFLAKE.

Your final SNOWFLAKE procedure should look like this:

```
TO SNOWFLAKE
SC :WHITE
RT 90
FD 20
REPEAT 2 [RT 90 FD 10]
RT 90
FD 20
LT 30
BOX
SNOWFLAKE
END
```

Can you make your snowflake two different sizes and/or two different colors?

## Toolbox Procedures

### Conditionals

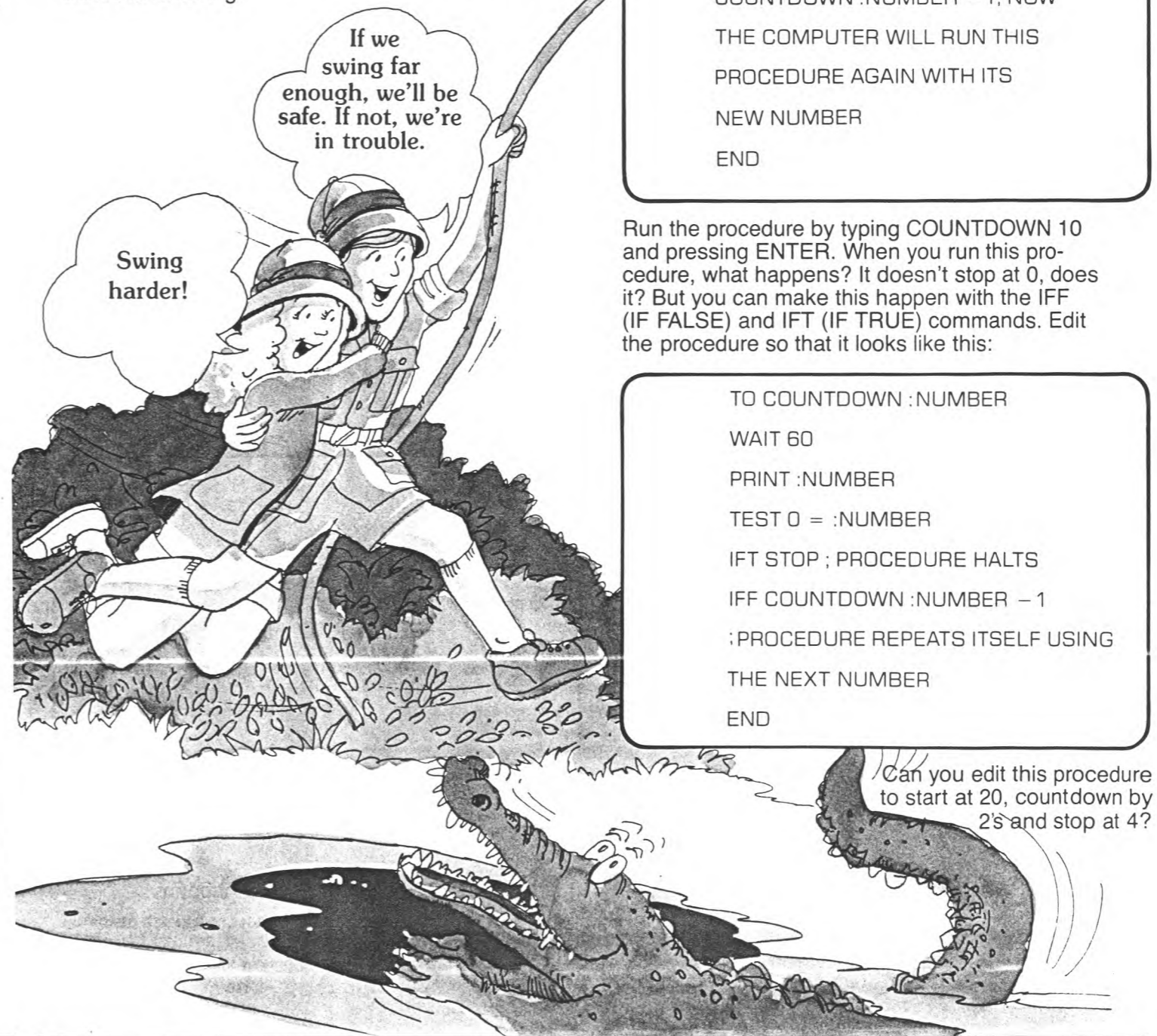
When you write a procedure, you may want something to happen at one time and something else at another time.

Let's say you want to write a procedure that counts backwards from 10 and stops at 0. So you write the following:

```
TO COUNTDOWN :NUMBER
WAIT 60
PRINT :NUMBER
COUNTDOWN :NUMBER - 1; NOW
THE COMPUTER WILL RUN THIS
PROCEDURE AGAIN WITH ITS
NEW NUMBER
END
```

Run the procedure by typing COUNTDOWN 10 and pressing ENTER. When you run this procedure, what happens? It doesn't stop at 0, does it? But you can make this happen with the IFF (IF FALSE) and IFT (IF TRUE) commands. Edit the procedure so that it looks like this:

```
TO COUNTDOWN :NUMBER
WAIT 60
PRINT :NUMBER
TEST 0 = :NUMBER
IFT STOP ; PROCEDURE HALTS
IFF COUNTDOWN :NUMBER - 1
:PROCEDURE REPEATS ITSELF USING
THE NEXT NUMBER
END
```



## Beep and NobEEP

When you want the computer to make a tone, you can use the BEEP and NOBEEP commands. WAIT (and a number) is used with BEEP to set the length of the tone. Try this:

```
TO HONK
BEEP
WAIT 15
NOBEEP
WAIT 15
END
```

Now, write this procedure that includes the HONK procedure:

```
TO PEEP
REPEAT 10 [HONK]
END
```

Enter PEEP and see what happens.



```
1. REPEAT
2. R
3. O
4. G
5. R
6. A
7. M
8. M
9. I
10. N
11. G
12. D
13. I
14. S
15. C
16. O
17. V
18. E
19. R
20. Y
```

## Instructions:

Read the definitions. Select the word on the list that fits the puzzle spaces. The first one is done for you.

## List of terms

|          |             |
|----------|-------------|
| BYE      | BUG         |
| WAIT     | HIDETURTLE  |
| NOBEEP   | CS          |
| MAKESHAP | RECURSION   |
| PENERASE | SPRITE      |
| FD       | COMMENTS    |
| RIGHT    | REPEAT      |
| SETCOLOR | PENREVERSE  |
| THAW     | COORDINATES |
| ENTER    | PROCEDURE   |

## Definitions

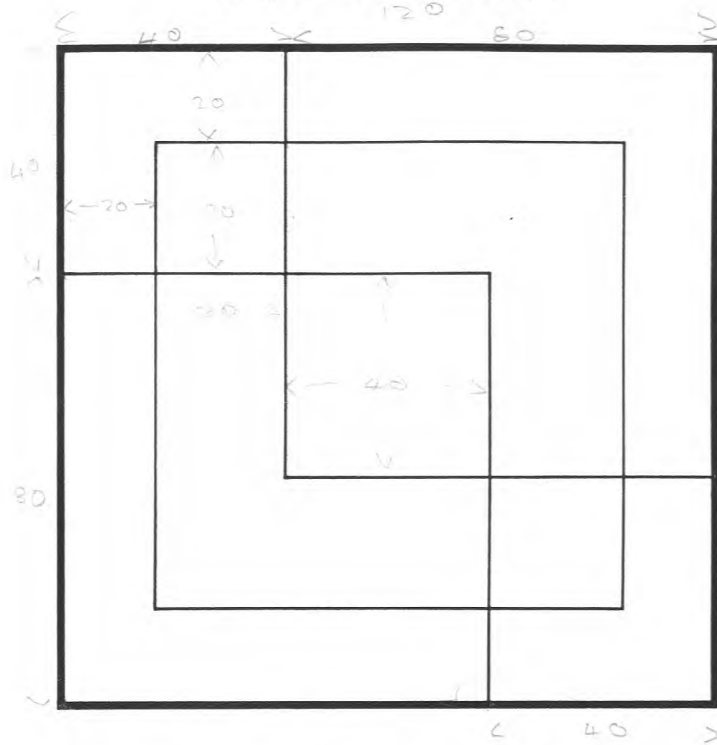
- Use this command, rather than retyping every step.
- Makes the Turtle's pen ready to erase a line.
- Turns off the tone started by a BEEP command.
- Turns the Turtle or a sprite in this direction.
- Causes the procedure to repeat until you press BACK.
- Restarts the sprites after they've been stopped by a FREEZE command.
- Used to change or create a sprite shape.
- Notes in a procedure that tell you what a line in the procedure does.
- Makes the Turtle disappear but still allows it to draw.
- Press this key after writing a line in a procedure.
- Prevents your procedure from operating correctly.
- Represents the short form for the FORWARD command.
- Causes the computer to pause for the period of time that you specify.
- Has these attributes: shape, color, speed, location and heading.
- Represents the short form for the CLEARSCREEN command.
- Used to give a particular location on the screen.
- Erases any line the Turtle travels over and draws a new one where none exists.
- Assigns a color to the Turtle or sprite.
- Gives the computer a set of instructions or steps to follow.
- Leaves TI LOGO and closes all open files and resets the computer.

## ANSWERS

```
1. REPEAT
2. PENERASE
3. NOBEEP
4. RIGHT
5. RECURSION
6. THAW
7. MAKESHAP
8. COMMENTS
9. HIDETURTLE
10. ENTER
11. BUG
12. FD
13. WAIT
14. SPRITE
15. CS
16. COORDINATES
17. PENREVERSE
18. SETCOLOR
19. PROCEDURE
20. BYE
```

# How Many Boxes?

How many boxes can you find in this design?



Try writing a Turtle procedure for this design. Keep a record of your procedure here.

---

---

---

---

---

---

---

---

---

---

# Onward with Sprites

You've learned to change the attributes — shape, color, speed, and heading — of a sprite by using commands. There is another way to make these changes! You can do it by writing a procedure that lets you press keys on the keyboard.

To write this kind of procedure, you need to use the operation READCHAR. With READCHAR, you can tell the computer to wait for a special keystroke. When the special key is pressed, the sprite's attributes are changed immediately.

Before you write a procedure using READCHAR, you need to choose a sprite and give it shape, color, and position by entering these commands:

```
TELL 1
CARRY :ROCKET
SC :GREEN
HOME
```

The procedure MOVE shows you how to use READCHAR with different keys on the computer. Now type this procedure:

```
TO MOVE
CALL READCHAR "KEY
IF :KEY = "G THEN SS 40
IF :KEY = "H THEN SS 0
IF :KEY = "P THEN SC :PURPLE
IF :KEY = "R THEN SC :RED
IF :KEY = "B THEN SC :BLUE
IF :KEY = "Y THEN SC :YELLOW
IF :KEY = "Q THEN STOP
MOVE
END
```

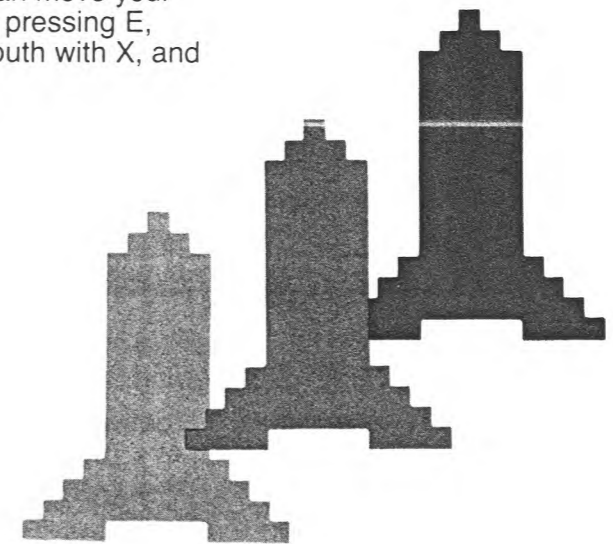
The CALL READCHAR "KEY statement allows you to let the variable, KEY, be equal to whatever key you want to press. For example, to change the green rocket's color to purple in the procedure MOVE, you decide what key you want to press for a change of color, purple. You use the IF...THEN conditional to specify the key and change the color of the sprite. A statement that would do this is IF :KEY = "P THEN SC :PURPLE.

Choose the sprite and enter the procedure. Next, start the procedure, press the P key, and see the purple rocket. Try pressing some of the other keys listed in the procedure and see what happens.

More activities with READCHAR:

1. Would you like to move a sprite shape around the screen by pressing keys? Write a procedure that uses the arrow keys to direct the heading of a sprite carrying a ball. (Hint: Use the letters E, S, D, and X for the arrow keys. In this way, you can move your sprite North by pressing E, East with D, South with X, and West with S.)

2. Try designing four different shapes with MAKESHAPE. Each shape could be controlled by a different arrow key. Hint: IF KEY = "E THEN SH :NORTH CARRY 6 would be one line of the procedure that would carry a special shape (6) North.



## Short Forms For Commands

Here is a list of those primitives\* that have short forms. The computer is programmed to accept both forms, so you can use either. (Those primitives that are not listed do not have short forms.)

| Primitive       | Short Form | Primitive  | Short Form |
|-----------------|------------|------------|------------|
| BACK            | BK         | PENREVERSE | PR         |
| CLEARSCREEN     | CS         | PENUP      | PU         |
| COLORBACKGROUND | CB         | RIGHT      | RT         |
| FORWARD         | FD         | SETCOLOR   | SC         |
| HIDETURTLE      | HT         | SETHEADING | SH         |
| LEFT            | LT         | SETSPEED   | SS         |
| MAKESHape       | MS         | SHOWTURTLE | ST         |
| PENDOWN         | PD         | YOURNUMBER | YN         |
| PENERASE        | PE         |            |            |

\*Primitives and Procedures

The things the computer already knows are called **primitives**. All the commands, the characters on the keyboard, and the five shapes for sprites are primitives. What you teach the computer is called a **procedure**. A procedure is made up of commands. It can have many commands and be a long procedure. Or, it can be short and have only a few commands.

## DEFINITIONS

### Attributes

Characteristics given to the Turtle and the sprites. These include color, shape, direction, speed, and screen position.

### BACK *number*

Moves the Turtle or a sprite backward the number of steps indicated.

### BEEP

Tells the computer to produce a tone.

### Bug

Something in a procedure that prevents correct operation or that results in an action you don't want.

### CARRY *:word*

### CARRY *number*

Defines or changes the shape of a sprite.

### CLEARSCREEN

Clears the display of words and Turtle drawings. Does not erase sprites.

### COLORBACKGROUND *:word*

### COLORBACKGROUND *:number*

Changes the background screen to the color indicated.

### Command

A primitive that tells the computer to perform predefined sets of instructions (subprocedures).

### Comments (;)

A descriptive remark typed at the end of a statement and separated by a semicolon (;).

### Debugging

Editing a procedure to make the computer perform a desired action.

### Dots (:)

The LOGO name for a colon (:). The colon always appears in front of a variable name that has already been assigned a value.

### EACH *[command]*

Tells the computer to do something to each of the sprites in the order they appear in the list.

### EDIT *procedure-name*

Tells the computer to enter the Edit mode so that changes can be made in the procedure.

### END

Defines the end of a procedure.

### ERASE *procedure-name*

Allows you to erase a procedure from the computer's memory.

### FORWARD *number*

Moves the Turtle or sprite forward the number of steps indicated.

### FREEZE

Stops the motion of all sprites.

### HIDETURTLE

Makes the Turtle disappear but still allows it to draw.

### HOME

Makes the active Turtle or sprite go to the middle of the display.

### IFT *action*

### IFF *action*

Used with TEST to determine the action to be taken when a condition is true or false.

### LEFT *number*

Tells the Turtle or a sprite to turn the number of steps indicated.

### MAKESHape *number*

Lets you design shapes for a sprite or a number of sprites to carry.

### Operation

A primitive that instructs the computer to perform a specific function which returns a value; an operation must be used in conjunction with a command.

### NOBEEP

Turns off the tone started by the command BEEP.

### NOTURTLE

Used to leave the Turtle mode.

### PENDOWN

Tells the Turtle to place its pen down and be ready to draw.

### PENERASE

Tells the Turtle to erase any lines it passes over.

### PENREVERSE

Erases any line it travels over and draws a new line where none exists.

### PENUP

Tells the Turtle to pick up its pen so it can be moved around the display without drawing a line.

### Primitives

Words and symbols that are predefined in TI LOGO as commands, operations, graphics, or names.

### PRINT *[list]*

### PRINT *number*

### PRINT *:word*

Tells the computer to print a list, word, number, etc.

### Procedure

A series of statements, including commands, operations, and their parameters that "teaches" (or programs) the computer to perform a desired action.

### READCHAR

Causes the computer to wait for a key to be pressed on the keyboard. This allows you to include keystrokes from the keyboard as part of a procedure.

### RECALL

Retrieves procedures, names, shapes and tiles from a disk or a cassette tape.

### Recursion

The process of making a procedure call itself by including its name as the last statement before the END statement. As a result, the procedure continues to repeat itself until you stop it by pressing BACK.

### REPEAT *number [list]*

Tells the computer to repeat a sequence of operations.

### RIGHT *number*

Tells the Turtle or a sprite to turn right the number of steps indicated.

### SAVE

Used when you want to save any procedures or names in your workspace.