

Leone Learning Systems, Inc.  
*Wonder. Create. Grow.*

Leone Learning Systems, Inc.    Phone 847 951 0127  
237 Custer Ave  
Evanston, IL 60202            Fax 847 733 8812  
Email [tj@leonelearningsystems.com](mailto:tj@leonelearningsystems.com)

---

# MSWLogo

---

An introduction and quick reference guide

**TJ Leone**  
**October 2004**

---



## Introduction

This guide was written to introduce you to features that are special to Microsoft Windows Logo (also known as MSWLogo). It does not assume that you have used Logo before, but it doesn't try to teach you Logo, either. You'll need to look elsewhere to learn details of the language.

You should look at this document while you have MSWLogo running on your computer. If you don't understand all the examples on your first reading of this document, that's OK. The purpose of your first reading is to get comfortable enough with the MSWLogo environment so that you can write and run example procedures. You can start writing procedures of your own later on, when you understand the Logo language better.

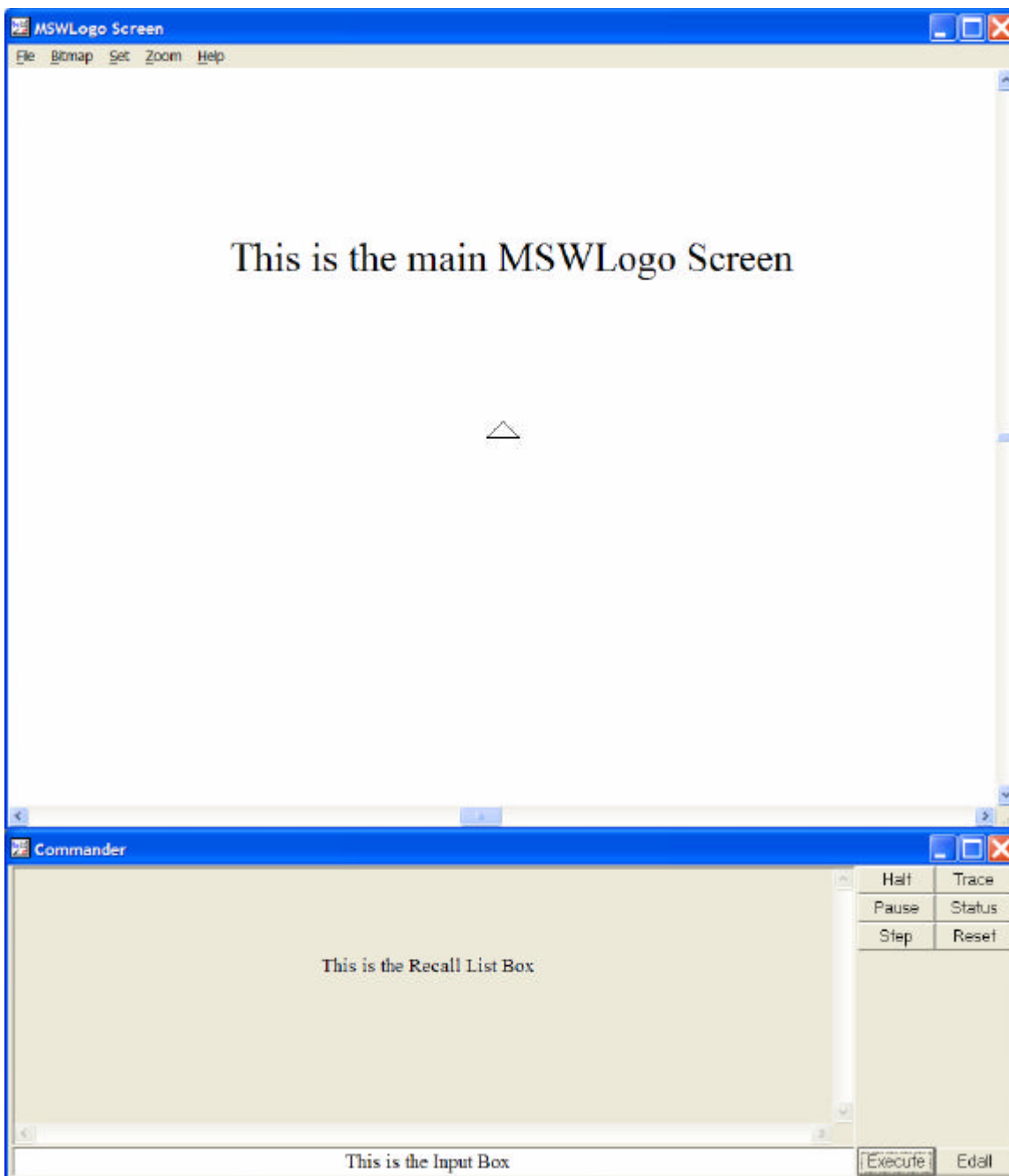
As you learn more about Logo, you can use this document as a reference. There are some nice features of MSWLogo that you might not appreciate on a first reading, such as the Step and Trace functions.

MSWLogo was built on top of Brian Harvey's implementation of Logo called Berkeley Logo. Brian's web site (<http://www.cs.berkeley.edu/~bh/>) is a great source of information on Logo, especially the free downloadable PDFs for the three volumes of the second edition of *Computer Science Logo Style*.

•  
•  
•  
•  
•  
•  
•

## The MSWLogo Screen

When you start up your Microsoft Windows Logo application, you'll see screens that look kind of like the ones in the picture below. In the picture, I added labels for the main MSWLogo Screen, the Recall List Box, and the Input Box:





## Entering Commands

Click in the Input Box and type a command. For example, try typing

```
fd 100
```

Then press the Enter key (or click on the Execute button).

Notice that *you cannot type into the Recall List Box*. This area simply displays commands that you have already typed.

Try some other commands, like

```
rt 90  
fd 60
```

## Reset

To erase everything on the screen and bring the turtle back to the center of the screen, click on the Reset button (the button to the right of the Step button and below the Status button).

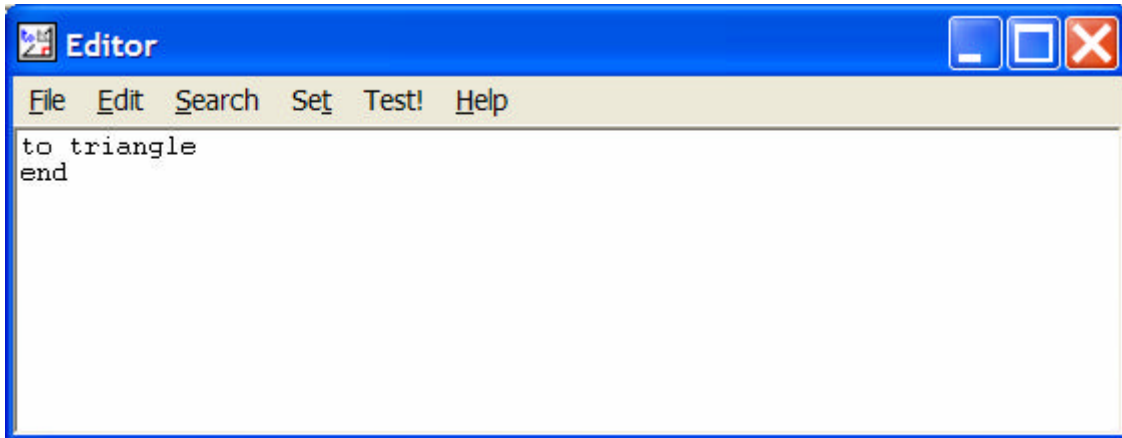
⋮

## Writing Procedures

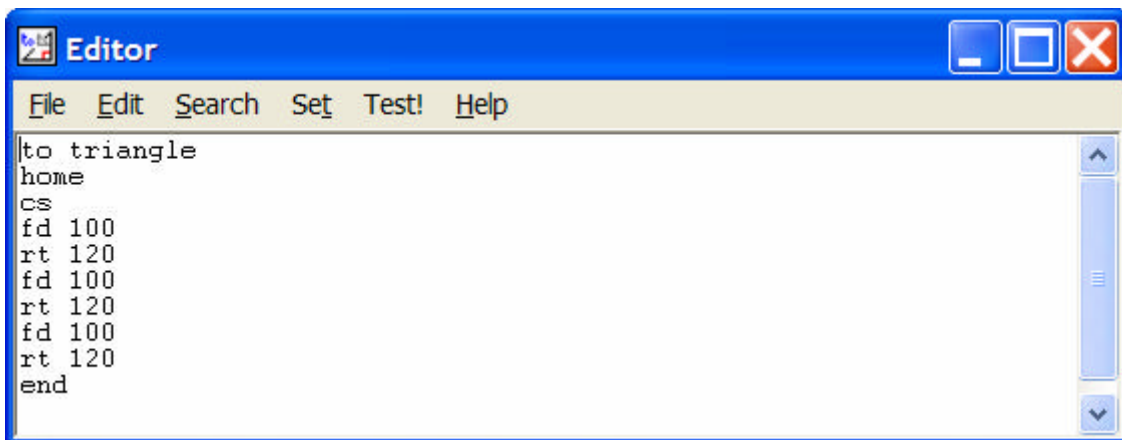
Click in the Input Box and type

```
edit "triangle
```

The edit window will appear:



Add lines to your procedure so that the completed procedure looks like this:



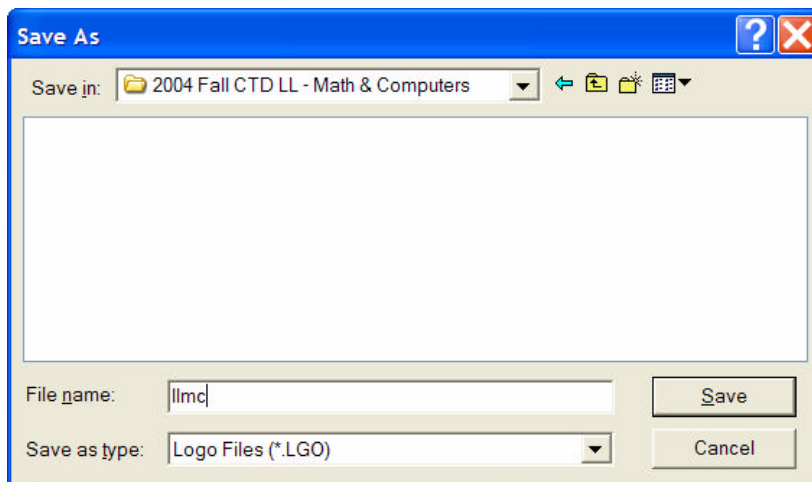
Close the editor window by selecting File/Save and Exit.

Now run your procedure by typing "triangle" into the input box and pressing the Enter key.

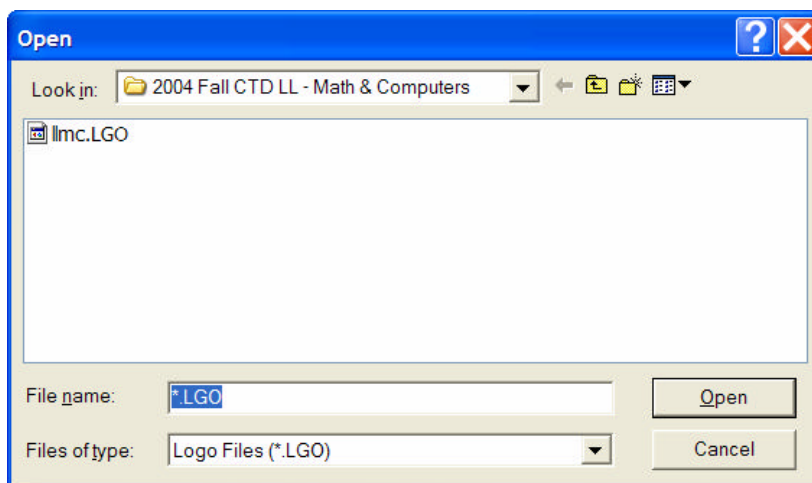
•  
•  
•  
•  
•  
•

## Saving your work

The “Save and Exit” item in the Editor window’s “File” menu only makes your procedures available for the current session. To save the procedures to your hard disk, you need to go to the main MSWLogo Screen and select File/Save. When you do, you will see a window like this one:



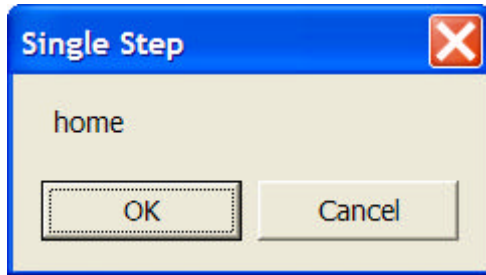
Make sure you remember which folder you saved your file in, and what name you gave to the file. In the example above, the file is “llmc”, and the file is saved to the “2004 Fall CTD LL – Math & Computers” folder. When you start up MSWLogo again in a later session, you can open your file by choosing File/Load... from the MSWLogo Screen:



⋮

## Stepping through Procedures

Try running the triangle procedure again. This time, click on the word “triangle” in the Recall List Box. Notice that the word “triangle” appears in the Input Box. Now press the Step button so that the label of the button changes to UnStep. Next, press the Execute button. You will see a dialog box:



Notice that “home” is the first command in the triangle procedure. Every time you press “OK”, you execute a command and “step” to the next command.

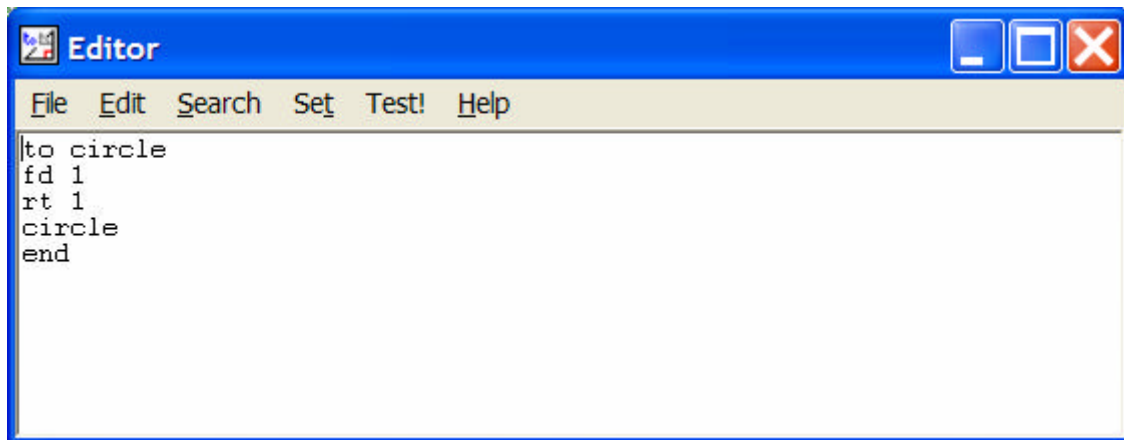
⋮

## Halting Procedures

Click in the Input Box and type

```
edit "circle
```

Then press the Enter key or click on the Execute button. Write the following procedure in the Editor window:



From the File menu of the Editor window, choose Save and Exit. After that, don't forget to also save from the file menu of the MSWLogo Screen. You'll notice that you didn't get a Save dialog box this time. That's because you already gave your file a name. MSWLogo saves your procedures to your hard disk with this new name.

Don't worry that you didn't see the triangle procedure in the editor window. It's still there in the background. If you like, you can see it again by typing edit "triangle into the Input Box. Or see the section below on the Edall button.

Type the following in your Input Box to move the turtle to the center of screen and clear the screen:

```
home cs
```

Then press the Enter key or click on the Execute button. Now execute the circle procedure by typing

```
circle
```

in the Input Box and pressing Enter.

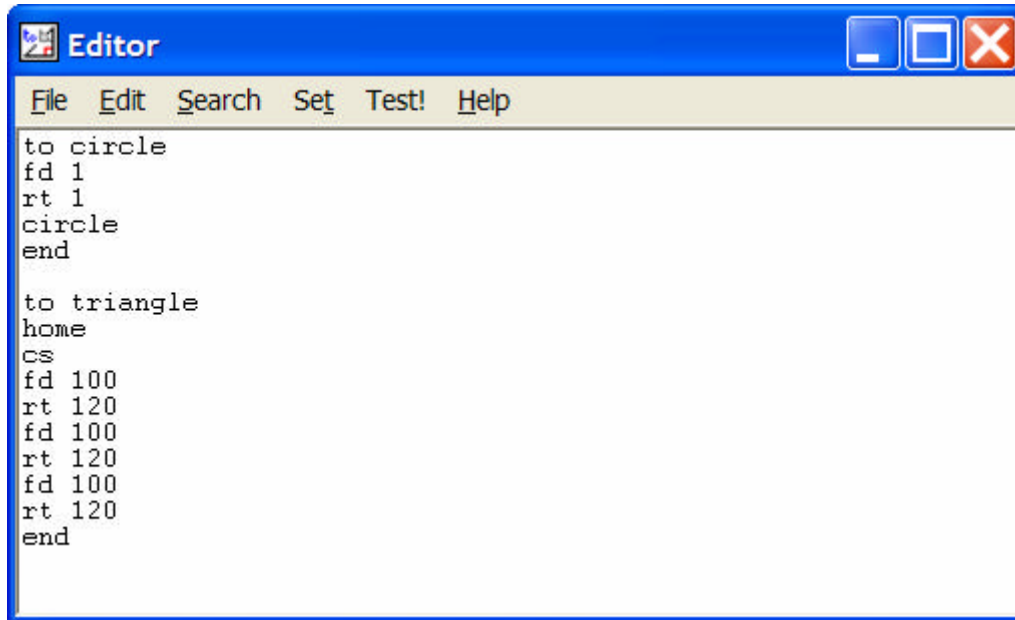
To stop the turtle, press the Halt button.



⋮

## Using Edall

Click on the Edall button, which is located to the right of the Execute button. This brings up an Editor button that allows you to see all of your procedures at once:



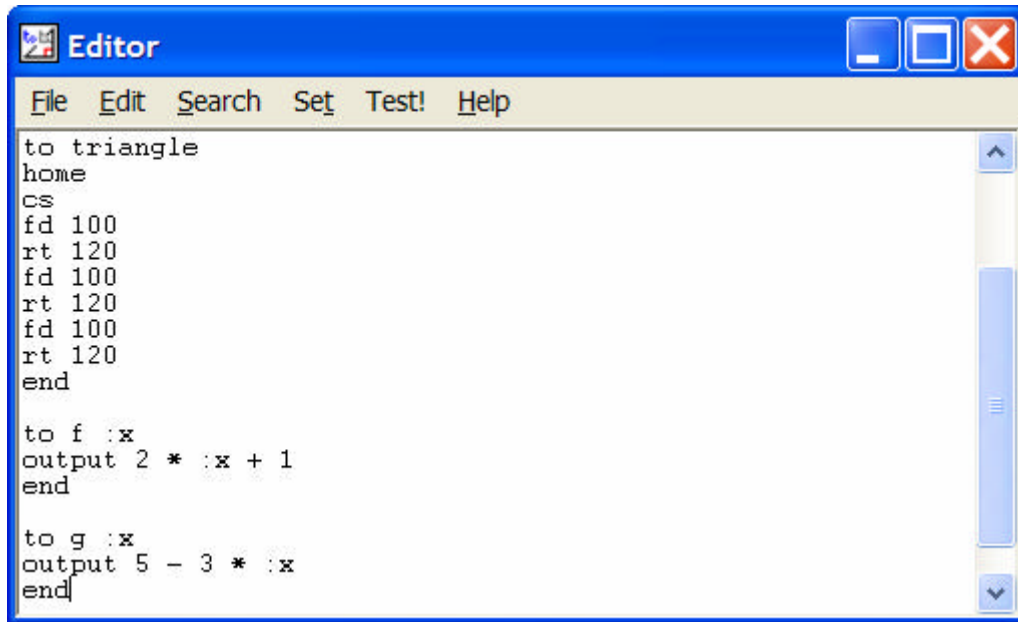
```
to circle
fd 1
rt 1
circle
end

to triangle
home
cs
fd 100
rt 120
fd 100
rt 120
fd 100
rt 120
end
```

⋮

## Tracing Procedure Calls

If you don't have your Editor window open, click on the Edall button to open it. Now add the procedures `f` and `g` as show below:



```
to triangle
home
cs
fd 100
rt 120
fd 100
rt 120
fd 100
rt 120
end

to f :x
output 2 * :x + 1
end

to g :x
output 5 - 3 * :x
end
```

Be careful with spaces. There should be a space between the `f` and the `:x`, but no space between the `:` and the `x`. Also make sure you have spaces between words and numbers, and spaces on either side of the `*` and `+` operators. Choose File/Save and Exit from the Editor window to save the new procedures. Don't forget to also File/Save from the MSWLogo Screen.

In the Input Box, type

```
show f 1
```

and press the Enter key.

In the Recall List Box, you should see:

```
show f 1
3
```

What's happening?

⋮

## Tracing Procedure Calls (cont'd)

Press the Trace button so that the label changes to UnTrace. Execute `show f 1` again (remember, you can click on a line in the Recall List Box to have that line appear in the Input Box). Now you should see this:

```
show f 1
      ( f 1 )
      f outputs 3
3
```

Why do you suppose `f` outputs 3? Remember, the output of the `f` procedure is

```
2 * :x + 1
```

Try using other inputs to the `f` function. For example, try:

```
show f 10
show f 2
show f 5
```

Can you see what's happening? If not, don't worry about it at this point. We have plenty of time to get a good understanding of functions like this one.

If the outputs make sense to you, try to predict outputs for `g` and then execute

```
show g 1
show g 2
```

If you feel comfortable with both the `f` and `g` functions, try combining them:

```
show f g 3
show f f 5
```

Make sure you Trace the procedure calls (i.e., make sure the label on the Trace button is UnTrace).



## The Author

TJ Leone owns and operates Leone Learning Systems, Inc., a private corporation that offers tutoring and educational software. He has a BA in Math and an MS in Computer Science, both from the City College of New York. He spent two years in graduate studies in education and computer science at Northwestern University, and six years developing educational software there. He is a former Montessori teacher and currently teaches gifted children on a part time basis at the Center for Talent Development at Northwestern University in addition to his tutoring and software development work. His web site is <http://www.leonelearningsystems.com>