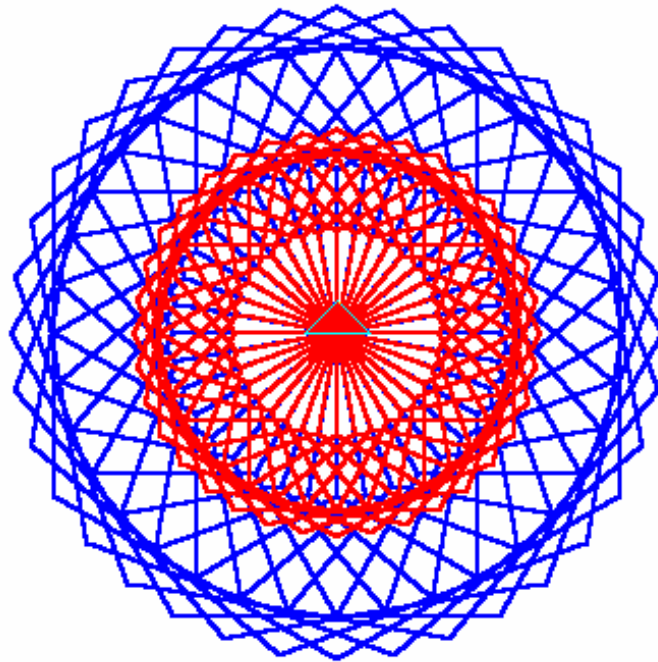


Logo Commands for MSW Logo



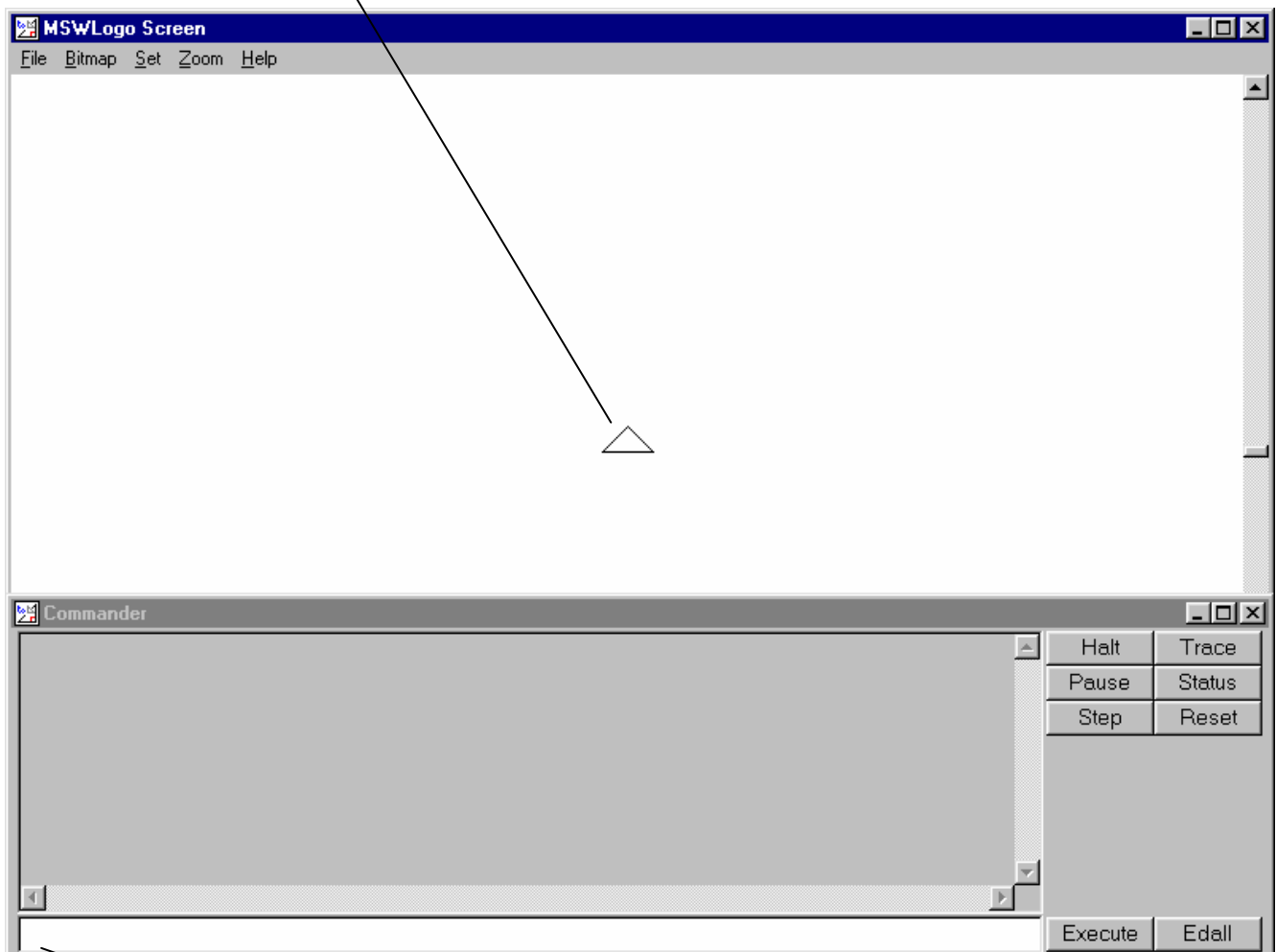
Newcastle 
City Council

I.C.T. Centre

Getting Started

On loading MSW Logo, this is the screen that you will see. The initial commands are typed into the bottom left window and when the return key is pressed, the turtle (arrowhead) will react to the instruction. See the next page for the main commands.

Turtle or arrowhead



Type command here

fd Moves turtle forward.

bk Moves turtle backwards.

lt Turns the turtle to the left.

rt Turns the turtle to the right.

pu Means pen up (no line is drawn)

pd Means pen down (line is drawn)

cs Clears the screen.

fill fills the shape with colour and can be added at the end of a command(as long as the turtle is inside the shape).

Pen colour from set menu lets you choose the pen colour.

Flood colour from set menu lets you choose the fill colours.

To create a repeat command type :

```
repeat 4 [fd 100 rt 90]
```

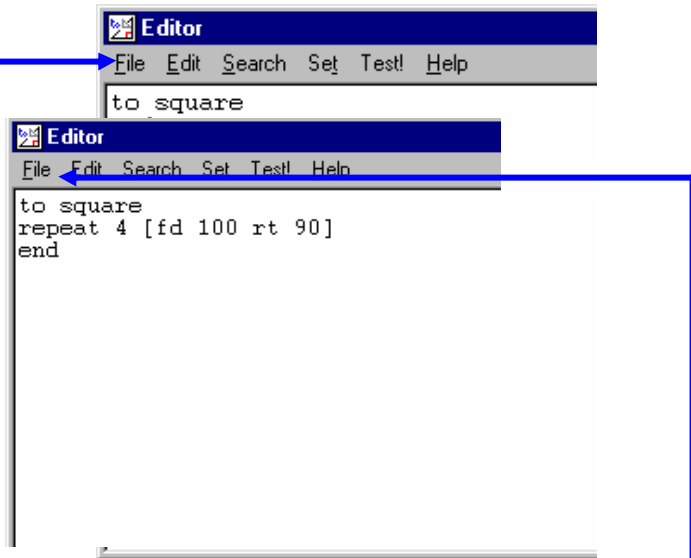
n.b. always leave a space between command and measurement.

Creating procedures

To create a procedure for a square:

Type `edit "square` and press `return`. (don't forget the speech marks before square).

A window opens



Between `to square` and `end` type in your commands.

Go to `File` and select `save` and `quit`.

Your procedure has now been set.

Type in `square` and `return`. The computer will run the procedure automatically.

You can save the logo file including all the created procedures.

Further procedures

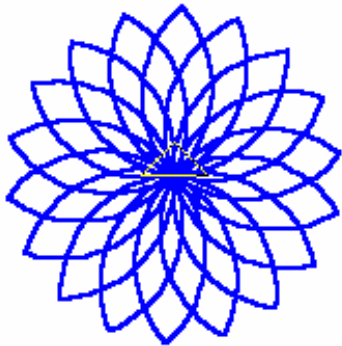
Have fun with this one;

to doodle

```
setpc random 7 fd random 150 rt random 360
```

```
doodle
```

```
end
```



The flower above was created in three stages. The first stage was:

To curve

```
Repeat 18 [fd 5 rt 5]
```

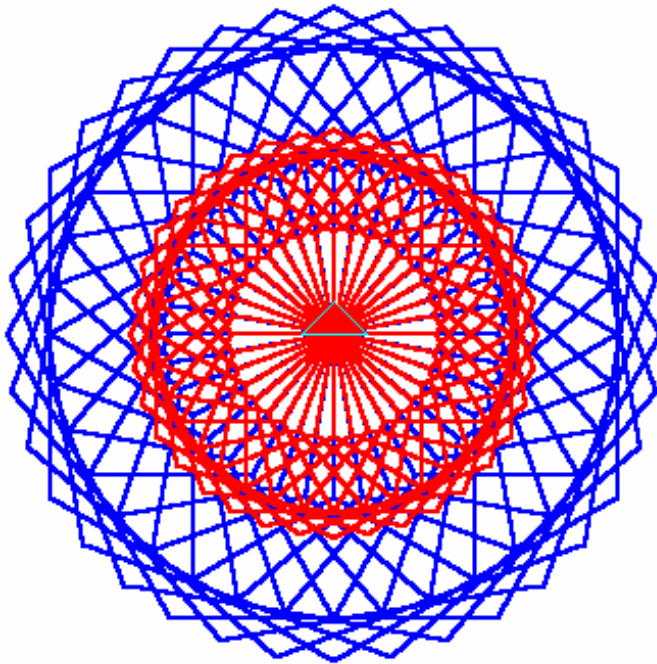
```
End
```

This gives you the side of a petal. Write a procedure to give a petal and then use it to produce a flower. Design flowers with different numbers of petals, different sizes and different colours.

Use your program for a square and try this one:

```
to pat  
repeat 36[square rt 10]  
end
```

NB. The word square might not be your name. Just change it accordingly.



This pattern was made with hexagons! Try making shapes with different numbers of sides and the put them in 'pat'

Visit the Logo website at

<http://www.softronix.com/logo.html>

The usual definition "**to square**" is of the form: **repeat 4[fd 20 rt 90]** This will draw a square with each side 20 units long.

A more flexible approach is to write the procedure so that a **value can be passed** to it at "**run time**". This is done using the colon (":") after the procedure name in the "to" line.

```
to square :sides  
repeat 4 [fd :sides rt 90]  
end
```

The procedure is called using the procedure name followed by a space and then the value to be passed. eg **SQUARE 50** will draw a square of sides 50. **square 20** will draw a square of sides 20 and so on.

MSW Logo uses the "**label**" command to write text to the screen. eg **LABEL "Hello** will write the word Hello at the turtle position.

If you want several words to appear you can use an 'underscore' such as: **LABEL "This_is_several_words**, or use the 'list' feature by putting the text in square brackets. eg **LABEL [This will write several words]**

Using Logo for Modelling.

Write a procedure to print out how many rolls of a simulated die are needed before a 6 appears.

To Rollto6

```
Make "count 0
Make "die 0
Until [:die = 6]
  [Make "die 1 + Random 6
   Make "count :count + 1]
Print :count
```

End

Rolling a die

How many rolls of a fair six-sided die do you think will be needed, on average, before the first 6 appears?

How many rolls might be needed for:

- a four-sided die before the first 4 appears?
- a twelve-sided die before the first 12 appears?

How might this generalise for an n -sided die?

The challenge is to write some procedures in a language such as *MSW LOGO*, or that of a graphic calculator, to simulate the process.

To RolltoN :n

```
Make "count 0
Make "die 0
Until [:die = :n]
  [Make "die 1 + Random :n
   Make "count :count + 1]
Print :count
```

End

Additional challenge

Write a procedure *RollExperiment*, using *RolltoN*.

It will need the arguments:

- $:n$ to specify the type of die, and
- $:times$ to specify how many times to simulate the roll to $:n$.

•It should output the mean number of rolls needed.

To Rollexperiment :n :times

```
To RollExperiment :n :times
  Make "sum 0
  Repeat :times [RolltoN :n
  Make "sum :sum + :count]
  Make "mean :sum / :times
  Print :mean
```

End