



DIVIDE AND RULE

In this instalment of our LOGO course, we introduce a number of turtle graphics procedures that produce interesting shapes by the use of recursion. Some of these shapes have very strange properties indeed, as they were the inventions of mathematicians intent on demonstrating geometric paradoxes.

Our first program is designed to draw 'tree' shapes. To start with, we can simply draw a trunk with a right and a left branch. The branches may then be formed in exactly the same way (although they will be smaller), with a central main stem and right and left twigs. If this process is continued, a tree shape will gradually be built up. This is a good example of the way recursion can be used in LOGO.

Our procedure for drawing such a 'binary' tree requires two inputs: one for the length of the trunk and the other the 'level' number. The length of the branches is halved at each level away from the trunk.

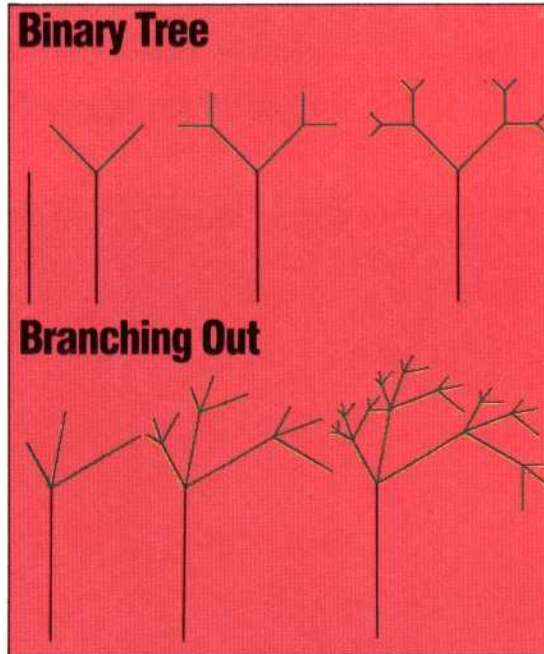
```
TO BRANCH :LENGTH :LEVEL
  IF :LEVEL = 0 THEN STOP
  FD :LENGTH
  LT 45
  BRANCH (:LENGTH / 2) (:LEVEL - 1)
  RT 90
  BRANCH (:LENGTH / 2) (:LEVEL - 1)
  LT 45
  BK :LENGTH
END
```

Notice that the procedure is 'state transparent'. This is important as otherwise the 'state' of the turtle (its position and heading) would be changed each time the procedure calls itself, making it impossible to continue the drawing.

It must be admitted that this procedure produces an unrealistic tree — to make it more interesting, the procedure can be modified in various ways. Here's a version that draws three branches, each of a different length, at each level:

```
TO BRANCH1 :LENGTH :LEVEL
  IF :LEVEL = 0 THEN STOP
  FD :LENGTH
  LT 30
  BRANCH1 (:LENGTH / 3) (:LEVEL - 1)
  RT 40
  BRANCH1 (:LENGTH / 2) (:LEVEL - 1)
  RT 50
  BRANCH1 (:LENGTH / 1.5) (:LEVEL - 1)
  LT 60
  BK :LENGTH
END
```

Try other modifications to produce more lifelike trees.



CHEQUERED POLYGONS

The following procedure draws a square, divides it into four, then divides each part into four, and so on:

```
TO BOARD :LENGTH :LEVEL
  IF :LEVEL = 0 THEN REPEAT 4 [FD :LENGTH RT 90]
  STOP
  BOARD (:LENGTH / 2) (:LEVEL - 1)
  FD (:LENGTH / 2)
  BOARD (:LENGTH / 2) (:LEVEL - 1)
  RT 90
  FD (:LENGTH / 2)
  LT 90
  BOARD (:LENGTH / 2) (:LEVEL - 1)
  BK (:LENGTH / 2)
  BOARD (:LENGTH / 2) (:LEVEL - 1)
  LT 90
  FD (:LENGTH / 2)
  RT 90
END
```

Write a similar procedure that splits a triangle up into four smaller triangles, then splits each of these up into four, and so on.

SNOWFLAKES

First draw an equilateral triangle — one whose sides are all of the same length. Now divide each side into three equal parts and draw a new equilateral triangle on the centre section. Rub out the common lines, then repeat this sequence for