

Chapter 7. Polygons, Circles, Stars and Stuff



“Now it’s time for the magic!”

“Magic?” asked Morf. “What do you mean, magic? You’ve never talked about Logo magic before.”

“We’ve talked about shapes, and how you can put two or more together to make a picture. We’ve talked about writing procedures and adding variables and things to those procedures. But this is all pretty ordinary stuff. Now it’s time to discover some of the magic — not just Logo magic, but math magic too.”

What’s the relationship between a square and a circle, for example? Is there a relationship? And why have it? What rules fit for the square, the circle, and other shapes? How can you prove that those rules are true?

There’s a lot to look at here, more than we can cover in just one chapter. However, we’ll tell you where to explore to discover all sorts of new things.

Playing With Polygons

You’ve worked with three, four, and six-sided polygons so far. By the way, can you write a procedure for a two-sided polygon?

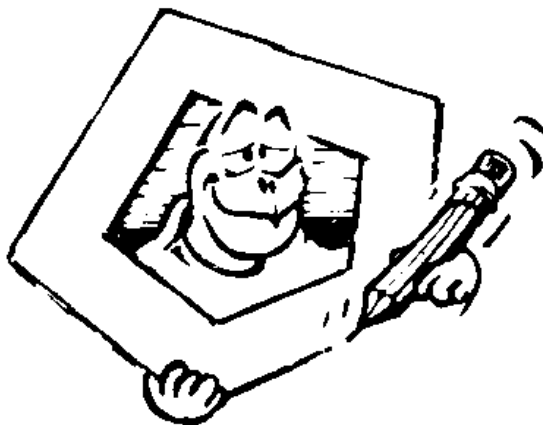
You had better say NO. Polygons are closed shapes with at least three sides.

Polygons, Circles, Stars and Stuff

Seems to me that we missed one above. Three, four, and six-sided polygons? What about a five-sided polygon? What do you call that?

That's a pentagon, just like that big building near Washington, D.C.

Pentagon Power



Pentagon REPEAT 5 [FD :SIDE RT ____]

How do you know how far the turtle should turn for the pentagon?

You can try a few numbers to see what works. But before you waste a lot of time, let's see if there is something to be learned from the other shape procedures you already know.

Look at the triangle. 3 repeats times 120 turns = _____

How about the square? 4 repeats times 90 turns = _____

Polygons, Circles, Stars and Stuff

The hexagon? 6 repeats times 60 turns = _____

The rectangle? 2 repeats times 90 + 90 = _____

What goes in the blanks? I bet it's 360 turns. (There's that number again.) So here's what you have:

For a triangle 3 repeats * 120 turns = 360

Another way of writing that is $360/3$ repeats = 120 turns. Isn't that right? If you did that for a square and hexagon, what would you have?

For a square: $360/4$ repeats = 90 turns

For a hexagon: $360/6$ repeats = 60 turns

So, if you want to write a procedure for a pentagon, how about this?

```
REPEAT 5 [FD :SIDE RT 360/5]
```

Try it. What happens?

Any Number of Sides

If you can do that for a 5-sided shape, bet you can do that for any number of sides. Try that, too!

```
REPEAT 3 [FD :SIDE RT 360/3]
```

```
REPEAT 4 [FD :SIDE RT 360/4]
```

```
REPEAT 6 [FD :SIDE RT 360/6]
```

```
REPEAT 9 [FD :SIDE RT 360/9]
```

```
REPEAT 15 [FD :SIDE RT 360/15]
```

Polygons, Circles, Stars and Stuff

What happens as the number of repeats gets bigger? What does the shape look like?

Actually, before you can tell what it looks like, the shapes begin to go off the screen. When this happens, you can fix it by taking fewer steps.

```
REPEAT 30 [FD 15 RT 360/30]
```

```
REPEAT 60 [FD 10 RT 360/60]
```

```
REPEAT 90 [FD 5 RT 360/90]
```

```
REPEAT 180 [FD 2 RT 360/180]
```

The important news is that the larger the number of repeats, the more the shape begins to look like a circle. If that's true, what would the procedure for a circle be?

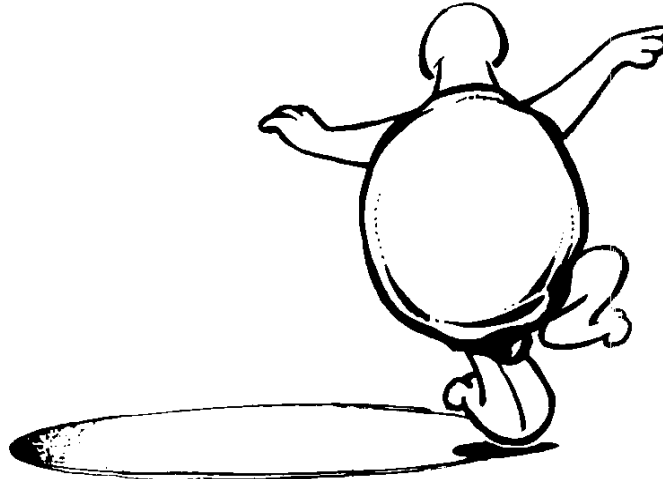
Playing With Circles

With all you've learned about the number 360, this one should be easy. How about this?

```
TO CIRCLE  
REPEAT 360 [FD 2 RT 360/360]  
END
```

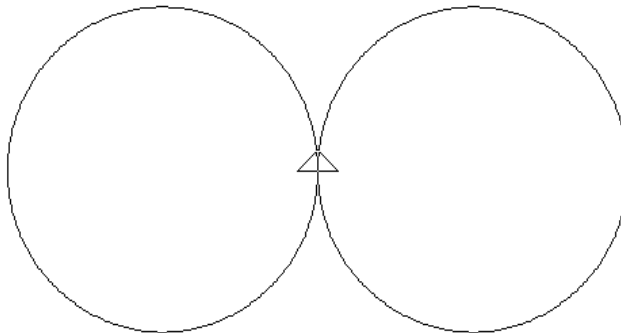
That's the same as:

```
TO CIRCLE  
REPEAT 360 [FD 2 RT 1]  
END
```



Of course, circles don't have to go to the right. Why not try a circle to the left?

```
TO CIRCLES
REPEAT 360 [FD 2 LT 1]
REPEAT 360 [FD 2 RT 1]
END
```



Gee, it looks like cross-eyes, doesn't it?

**A Circle
Challenge**

How would you put two smaller circles inside these big ones? Solving this challenge is like solving any problem.

1. Start with what you know.

You know that the turtle just drew two large circles by first going FORWARD 2 steps, then RIGHT or LEFT 1 turtle turn 360 times.

Polygons, Circles, Stars and Stuff

2. Now, what do you need to know?

You need to know how to change the procedure so the circles will be smaller.

OK, what can you change?

- You can change the number of repeats.
- You can change the number of turtle steps.
- You can change the number of turtle turns.

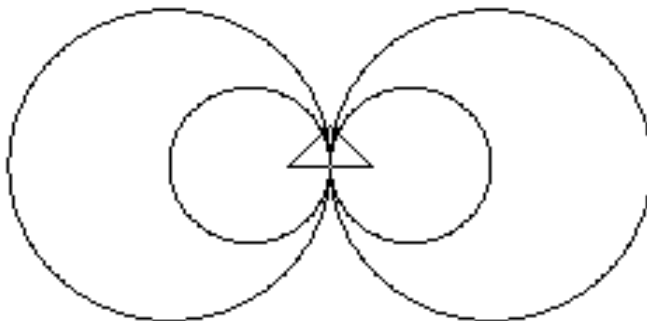
You have already learned that to draw a circle, you tell the turtle to

```
REPEAT 360 [FD :SIDE RT or LT 1]
```

So, to draw a smaller circle, it seems as if you can make the number of steps smaller. So try this:

```
REPEAT 360 [FD 1 RT 1]
```

It looks all right, doesn't it? Next draw a circle to the left.



Now, that really does look like cross-eyes.

Why not create some other drawings using circles and the other shapes you've been using? Draw them on paper or in your journal. Then draw them on the screen.

More About Circles

What else do you know about drawing a circle?

You know that the number of repeats times the number of turtle turns equals 360. You used 360 times 1. What happens if you change that to 180 times 2? Would that make a smaller circle or a larger one? Go ahead. Try it out.

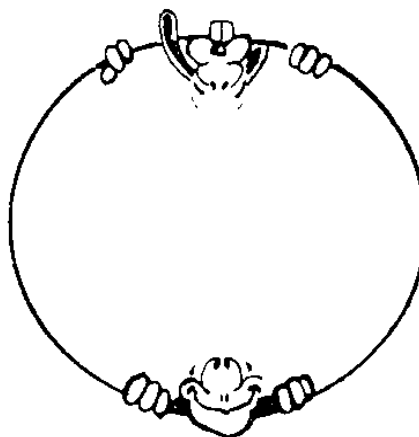
Earlier, you started with the familiar shapes of square, triangle, rectangle, and hexagon, and increased the number of REPEATS and the number of turns. Now let's turn this around and go the other way — just for the fun of it.

REPEAT 360 [FD 2 RT 1]

REPEAT 180 [FD 2 RT 2]

REPEAT 120 [FD 2 RT 3]

Now make up shapes of your own. It soon stops looking like a circle



Defining a Circle

Just what is a circle, really? You made up some rules about other shapes. So what is the rule for circles?

If you ask a math teacher, you'll find that a circle is a closed curve where every point on the curve is the same distance from a fixed point. On the Logo screen, you can use HOME as the fixed point. (That fixed point is really the center.) Then a circle is a closed curve where every point on that curve is the same distance from HOME.

Make sense? Sure! Now, can you write a procedure to draw a circle like this?

Let's make HOME our fixed point, the center of our circle. Next you have to draw a line so that every point on that line is the same distance from HOME.

How can you do that?

You'll have to pick the pen up, go out to where you want your circle to be, draw a point, come back, turn a little bit, go out the same distance, draw another point, come back, turn a little, and so on.

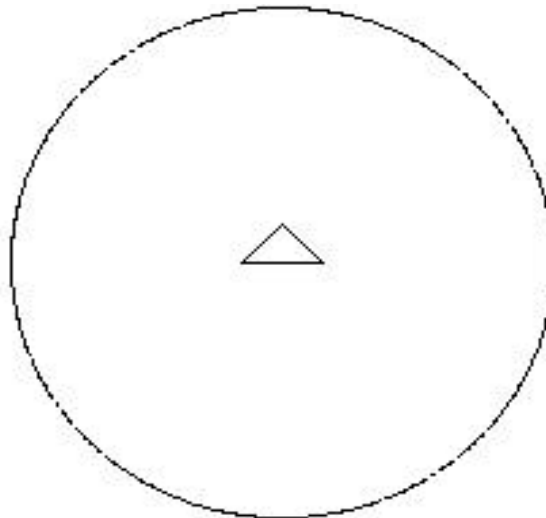
Before you start drawing the line, let's first write a procedure to draw a point.

```
TO POINT  
PD RT 90 FD 1  
BK 1 LT 90 PU  
END
```

See what that does? Now let's put this procedure to good use.


```
TO CIRCLE :SIDE  
HOME CS PU  
REPEAT 360 [FD :SIDE POINT BK :SIDE RT 1]  
END
```

The procedure starts from HOME with a clear screen. Now, rather than go FD 1 and RT 1 like you did before, the turtle goes FD :SIDE, draws a point, goes back HOME, and turns RT 1 turtle turn, goes out, draws a point, comes back, and starts all over again. You end up drawing a line where every point on that line is the same distance from the center.



What do you call that distance from the center of the circle, the fixed point, to the edge?

That's the radius of the circle.

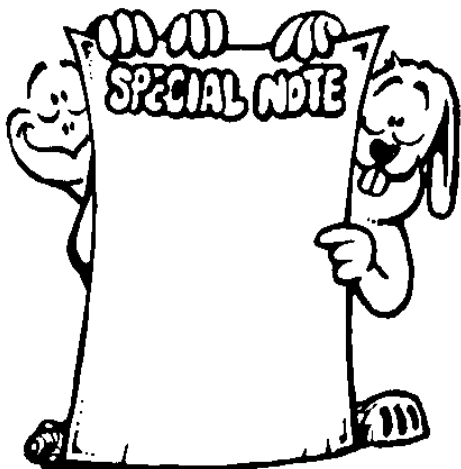
What do you call that edge, the line around the fixed point? Right, that's the circumference of the circle.

Here's another one. What do you call the distance from one side of the circle to the other, the line that goes through the center of the circle?

Give yourself a Gold Star if you said that's the diameter of the circle.

One more: what's the relationship between the diameter of the circle and the circumference? Hmmm? That's a tough one, isn't it?

The diameter times Pi equals the circumference. But what's Pi?



SPECIAL NOTE: Pi is the 16th letter in the Greek alphabet. It is also one of those strange mathematical numbers that will keep your computer busy for hours. They call it a “transcendental” number because it has no set value. The computer will keep on dividing for hours. Pi will never come out even.

PICALC.LGO is a procedure on the CD that comes with this book that calculates Pi to as many decimal places as you want. Load the procedure. Then type PICALC and the number of decimal places to which you want Pi calculated. For example, PICALC 50.

Here's Pi calculated to 50 places.

3.14159265358979323846264338327950288419716939937
5105

It takes some time for your computer to figure that one out. Why not try an even larger number and see how long that takes?

The Rule of 360

Now, do you remember all that you learned about that number 360? It keeps popping up doesn't it. Well, there's another rule in there somewhere. Actually, there are two.

- It seems that when you add up the angles you need to draw any closed shape that is made from straight lines, the answer is 360.
- Another interesting thing is that when the turtle travels through 360 degrees, she always ends up where she started.

Try it. Add the same number to both blanks in this procedure.

```
REPEAT _____ [FD :SIDE RT 360 / _____]
```

Take a look at these commands again. How would you write one procedure that would draw all of them?

```
Triangle: REPEAT 3 [FD :SIDE RT 360 / 3]
```

```
Square: REPEAT 4 [FD :SIDE RT 360 / 4]
```

```
Pentagon: REPEAT 5 [FD :SIDE RT 360 / 5]
```

```
Hexagon: REPEAT 6 [FD :SIDE RT 360 / 6]
```

```
Octagon: REPEAT 8 [FD :SIDE RT 360 / 8]
```

In these procedures you use the variable :SIDE for the length of one side of the shape.

How about this:

Polygons, Circles, Stars and Stuff

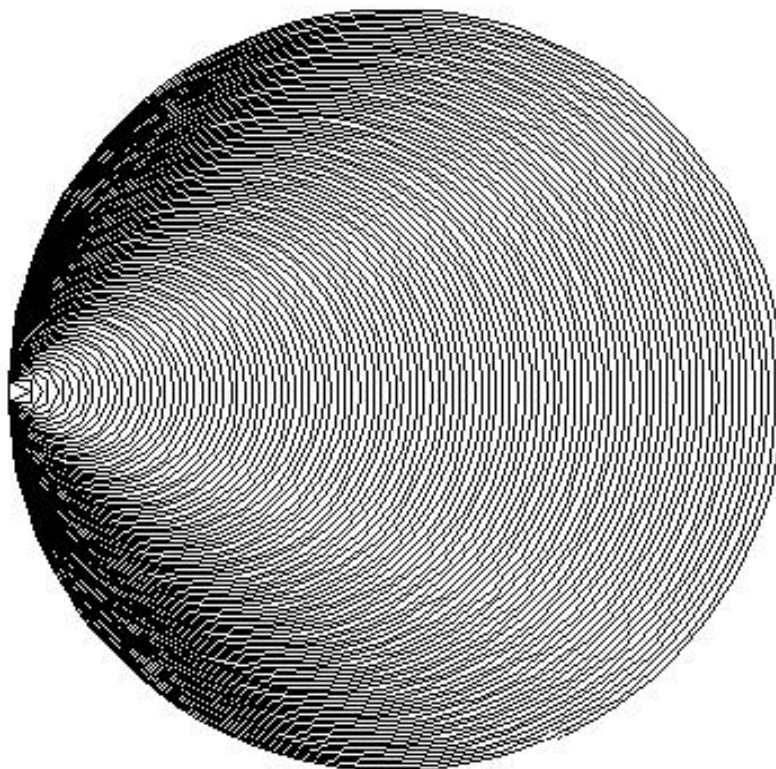
```
TO POLYGON :SIDE :REPEATS  
REPEAT :REPEATS [FD :SIDE RT 360/ :REPEATS]  
END
```

Once you've played with this procedure a bit, write one to draw all of the shapes up to a circle.

```
TO POLYGONS :SIDE :REPEATS  
IF :REPEATS = 360 [STOP]  
REPEAT :REPEATS [FD :SIDE RT 360 / :REPEATS]  
POLYGONS :SIDE :REPEATS + 1  
END
```

You can make the :SIDE anything you want. But what would you start :REPEATS at? Think about the definition of a polygon. That will tell you.

Here's what the procedure looks like when it's run. We had to stop it before it got to big for the screen.

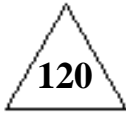







**Rule of 360
Worksheet**

Fill in the correct numbers in the following chart and draw a picture of what happens on the screen. A few have been filled in to get you started.

The numbers at the top of each column in the table are the number of repeats. The numbers down the left column are the number of turtle turns.

For the first row across the top of the table, the number at the top of each column equals the number of sides for the shape.

Number of Turtle Turns (TT)	Number of Repeats						
	3	4	5	6	8	10	12
TT * 1							
TT * 2							
TT * 3	360						
TT * 4		360					
TT * 5			360				

More Playing With Polygons

Do you remember how we got started developing shapes? The first thing you did was create a corner.

```
TO CORNER  
FD 100 RT 90  
END
```

Then you used the corner to make a square. But what if you changed that procedure? What if you added variables?

```
TO CORNER :SIDE :ANGLE  
FD :SIDE RT :ANGLE  
END
```

Now you can make just about any shape. What's this one?

```
REPEAT 3 [CORNER 100 120]
```

Or this one?

```
REPEAT5 [CORNER 100 72]
```

Since you've already used CORNER before, let's find another name. Since this procedure is used to draw polygons, why not call it POLY?

```
TO POLY :SIDE :ANGLE  
FD :SIDE RT :ANGLE  
END
```

Now let's see what you can do with POLY. The procedures shown on the next page can be found on the CD that came with this book as POLYS.LGO.

For one thing, you can improve on your “Any Shape” procedure.

```
TO POLYGON :SIDE :ANGLE
REPEAT 360/:ANGLE [POLY :SIDE :ANGLE]
END
```

Here’s a few other variations you’ll find in the POLYS.LGO file.

```
TO DUOPOLY :SIDE1 :SIDE2 :ANGLE
REPEAT 50 ~
[
  POLY :SIDE1 :ANGLE
  POLY :SIDE2 :ANGLE
]
END
```

```
TO RANDPOLY :SIDE :ANGLE
REPEAT 100
[
  IFELSE (RANDOM 10) = 0 [PD] [PU]
  POLY :SIDE :ANGLE
]
END
```

Random and Rerandom

There’s another new command for you — RANDOM. RANDOM picks an integer between 0 and the input you give it. If you don’t what an integer is, it’s a whole number — a number without any fractions or decimals after it.

15 is an integer or whole number. It doesn’t include any parts of a number such as a fraction (1-1/2) or decimal (1.5). You’ll read more about integers and whole numbers in *The Great Math Adventure* chapter.

To see what RANDOM does, try this:

Polygons, Circles, Stars and Stuff

```
REPEAT 10 [SHOW RANDOM 10]
```

What happened? You got a list of 10 numbers selected randomly from the numbers 0 - 9. Now try it again. And again.

You get a different list each time, right? But what if you want to use that same list again and again? That's where RERANDOM comes in.

RERANDOM takes what they call a **seed** number. This seed number identifies a particular sequence of random numbers. Take a look.

When you want to select a sequence of random numbers to use over and over again, use the RERANDOM command followed by the RANDOM command. Take a look.

```
(RERANDOM 15)  
REPEAT 2 [SHOW RANDOM 5]  
4  
2
```

To repeat that sequence, type

```
(RERANDOM 15)  
REPEAT 2 [SHOW RANDOM 5]  
4  
2
```

If you want to add another sequence of numbers randomly selected from 0 - 4, just change the seed number.

```
(RERANDOM 25)  
REPEAT 2 [SHOW RANDOM 5]  
1
```


0

```
(RERANDOM 25)
REPEAT 2 [SHOW RANDOM 5]
1
0
```

To demonstrate the value of RERANDOM, try this one again.

```
(RERANDOM 15)
REPEAT 2 [SHOW RANDOM 5]
```

What happened?

There's another thing to consider. What if you want to select a random number but not zero? How about this.

```
SHOW (RANDOM :INPUT) + 1
```

Some versions of Logo pick a number between 1 and your input, rather than from 0. So if you ever want Logo to pick 0 as the input, use

```
(RANDOM :INPUT) - 1
```

Dressing Up Poly

These procedures are rather ordinary by themselves. To dress them up a bit, why not add a small turn at the end of each list. For example:

```
TO DUOPOLY :SIDE1 :SIDE2 :ANGLE
REPEAT 50
[
  POLY :SIDE1 :ANGLE
  POLY :SIDE2 :ANGLE
  RT 10
```

Polygons, Circles, Stars and Stuff

```
]
END
or
```

```
TO RANDPOLY :SIDE :ANGLE
REPEAT 100 ~
[
  IFELSE (RANDOM 10) = 0 [PD] [PU]~
  POLY :SIDE :ANGLE RT 10
]
END
```

Play around with different numbers to see just what these procedures can do.

Here's a set of procedures that draws with dots. What type of variations can you dream up for this procedure?

```
TO POLYSPI :ANG
MAKE "SIDE 0
REPEAT 100 ~
[
  POLY :SIDE :ANGLE
  PD FD 1 BK 1 PU
  MAKE "SIDE :SIDE + 2
]
END
```

Finally, what can you do with this inward spiral?

```
TO INSPI :SIDE :ANGLE :INC.
FD :SIDE RT :ANGLE
INSPI :SIDE :ANGLE + :INC :INC
END
```

If you really want a challenge, try digging into this procedure. Can you predict what different values for the variables will product?

If this seems too easy, try this one.

```
TO INSPI_1 :SIDE :ANGLE :INC
REPEAT 360 * :ANGLE ~
  [
  FD :SIDE
  RT REPCOUNT + (2 * :INC - :ANGLE)/ ~
    (2 * :ANGLE)
  ]
END
```

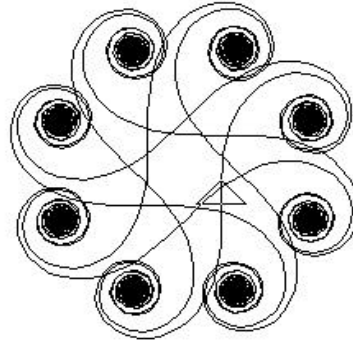
There's another new command REPCOUNT. It outputs the number of repeats called for by the REPEAT command. Here's a very simple demonstration of what REPCOUNT does.

```
REPEAT 3 [(PR "|THIS IS REPCOUNT| REPCOUNT)]

THIS IS REPCOUNT 1
THIS IS REPCOUNT 2
THIS IS REPCOUNT 3
```

As shown in the INSPI_1 procedure, you can use the output of REPCOUNT to calculate other values.

You can spend a very long time studying the mathematics of the INSP and INSPI_1 figures. Here's some inputs to get you started.



This figure was created using inputs of 8 8 5. Why not try

INSPI_1 10 7 3

INSPI_1 10 10 7

INSPI_1 10 12 5

Have fun with these!

Arcs, Circles, Hoops, and Things

By now, you should have a pretty good idea of how to work with shapes such as the square, rectangle, triangle, circle, and such.

Now it's time to start making Logo jump through some hoops! In the rest of this book, you'll find there's a lot more to turtle graphics than just drawing shapes.



You already know that you can draw a circle using an instruction like this one.

Polygons, Circles, Stars and Stuff

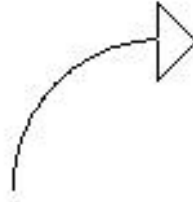
REPEAT 360 [FD 1 RT 1]

How would you draw a curve — just part of a circle?

If there are 360 degrees in a circle, part of a circle would be something less than that. So let's try

REPEAT 90 [FD 1 RT 1]

Hmmm?



Think about this. You already know that RT or LT 180 makes the turtle turn around and go the opposite direction. So what would happen if she “did a 180” here?

REPEAT 4 [REPEAT 90 [FD 1 RT 1] RT 180]

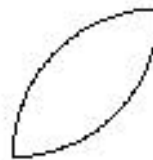
That's interesting. But what happens when you try this one?

REPEAT 6 [REPEAT 60 [FD 1 RT 1] RT 240]

Does this one remind you of a snowflake?

REPEAT 2 [REPEAT 90 [FD 1 RT 1] RT 90]

I don't know about you, but that looks like a leaf to me.



What would happen if you just continued around the circle?

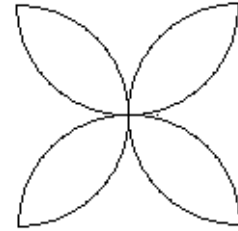
Here's the leaf:

Polygons, Circles, Stars and Stuff

```
TO LEAF  
REPEAT 2 [REPEAT 90 [FD 1 RT 1] RT 90]  
END
```

Now try this:

```
TO LEAVES  
REPEAT 4 [LEAF LT 90]  
END
```



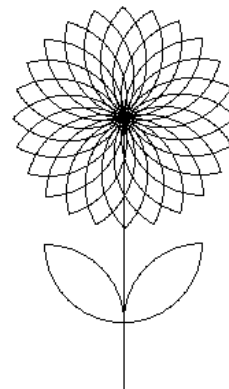
You know, with a little bit of work, that could be all sorts of things. Why not see what you can do with this shape before you move along? Maybe it could be a butterfly.

Or maybe a Indian headdress. That's made from feathers, right?



How about a chrysanthemum?

How about making
a whole flower bed
using different sizes of
arcs for different sizes
of leaves?



What other types of
flowers can you draw?

How about three-dimensional flowers?

```
TO DAHLIA
CS HT CLEARPALETTE SETPENSIZE [2 2]
REPEAT 16 ~
  [
    SETPC (LIST REPCOUNT*256/16 0 0)
    REPEAT 8 [RT 45 REPEAT 4
      [
        REPEAT 90 [FD 2 RT 2] RT 90]
      ]
    RT .4
  ]
PR [ISN'T IT PRETTY!]
END
```

There's another new command here that you can read about in the MSW Logo on-line help file — CLEARPALETTE.

Arc and Circle Tools

There are lots of times when you want to use ARC and CIRCLE procedures. So we made up a set of tools you can use at any time. These are on the CD that came with this book as ARCTOOLS.LGO. Just load them into your own procedures and you're ready to go.

Before you do that, however, there are a few things to note.

Polygons, Circles, Stars and Stuff

ARCL and ARCR start from the edge and draw arcs to the left and right. CIRCLEL and CIRCLER also start from the edge and draw circles to the left and right. The variable :RADIUS tells the computer how big to make the arc or circle. Do you remember, the radius of a circle is the distance from the center to the outside rim or circumference?

The variable :DEGREES tells the computer how big to make the arc. ARCR 100 60 tells the turtle to draw an arc that is 60 degrees long and with a radius of 100.

What is that number 3.14159? That's Pi, remember. Some people use 3.14. Some use 3.1416. In some versions of Logo, you can use the Logo primitive, Pi.

```
TO ARCL :RADIUS :DEGREES
LOCAL "AMT
MAKE "AMT 2 * :RADIUS * 3.14159 / 36
REPEAT :DEGREES / 10 [LT 5 FD :AMT LT 5]
END
```

```
TO ARCR :RADIUS :DEGREES
LOCAL "AMT
MAKE "AMT 2 * :RADIUS * 3.14159 / 36
REPEAT :DEGREES / 10 [RT 5 FD :AMT RT 5]
END
```

```
TO CIRCLEL :RADIUS
LOCAL "AMT
MAKE "AMT 2 * :RADIUS * 3.14159 / 36
REPEAT 36 [LT 5 FD :AMT LT 5]
END
```

```
TO CIRCLER :RADIUS
LOCAL "AMT
```



```
MAKE "AMT 2 * :RADIUS * 3.14159 / 36
REPEAT 36 [RT 5 FD :AMT RT 5]
END
```

What's that :AMT variable?

That's part of the mathematical formula for drawing a circle or part of a circle. It calculates the size of the circle based on the size of the radius you give it.

Have some fun with these procedures. They are useful tools to keep handy.

There's a lot more to do with polygons and things, especially when you start digging into recursion. But first, let's take a look at turtle positions.

ARC, ARC2, CIRCLE, and CIRCLE2

Now that you've learned the hard way to develop arc and circle tools, it's time to tell you about the easy way. There are tools in the Logolib directory that will draw arcs and circles from either the center of the circle or from the edge.

ARC and CIRCLE draw from the center. ARC2 and CIRCLE2 draw from the edge.

The ARC procedures take two inputs:

ARC <angle><radius>

The CIRCLE procedures only take one input, the radius. Why not try these commands?

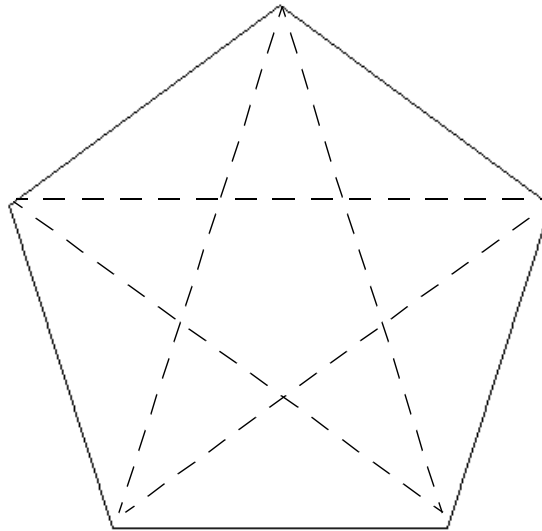
Rabbit Trail 20. Star Gazing



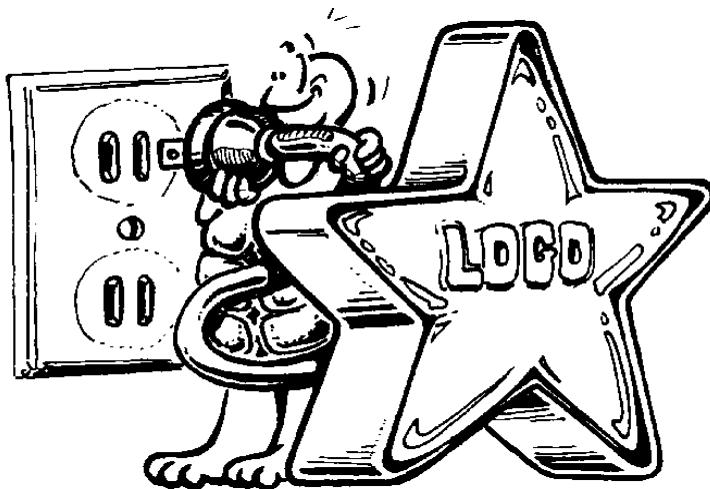
Draw a big pentagon on the screen. Do you remember how to do that?

REPEAT 5 [FD 100 RT 360 / 5]

Now print the screen. Take a pencil and a ruler and draw a diagonal line to each corner. In this way, every corner is connected to every other corner by a line.



What do you see inside the pentagon? I see a big star; how about you?

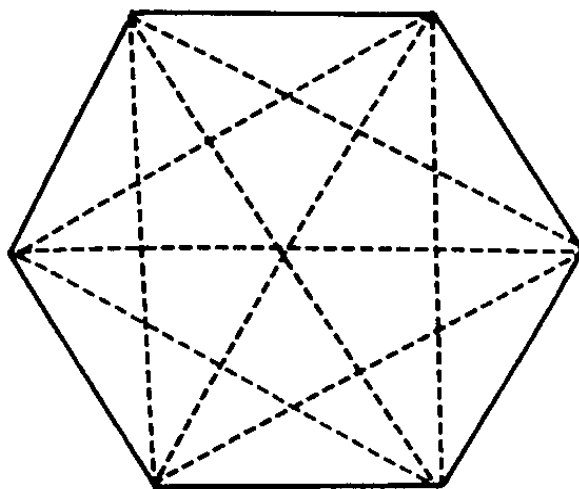


Polygons, Circles, Stars and Stuff

Now draw a big hexagon on the screen.

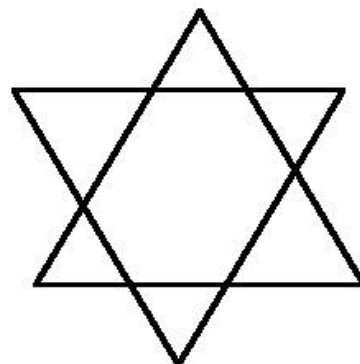
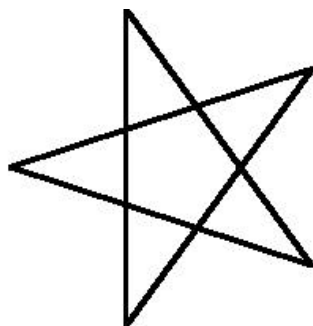
REPEAT 6 [FD 100 RT 360 / 6]

Draw diagonal lines to connect all the corners. What do you see? There's a Star of David, right? That's a six-pointed star.



Try this with an octagon. Now that's a lot of lines to draw.

How would you draw these stars on the screen? Do the drawings you did on paper help you figure it out?



Polygons, Circles, Stars and Stuff

You drew a shape first. Then you drew diagonals to each of the corners. So let's explore the shapes first and see what can be learned there.

Take a look at the pentagon.

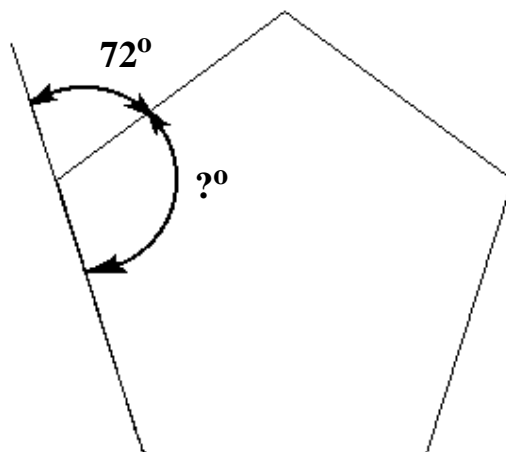
To draw the pentagon shape below, the turtle first turns LT 18, then goes FD 100 RT $360 / 5$ (that's the same as RT 72).

Why turn LT 18?

$$90 - 72 = 18.$$

You should be able to figure that out by looking at the drawing below.

The angle outside the pentagon is 72. What's the angle inside the pentagon?



Let's find out. This is going to get a bit complicated, so hang in there. Just follow along at the computer.

Clear the screen and draw another pentagon like the one above. Make sure you can see the turtle.

Then type:

```
FD 100
```

The turtle has now moved up the first side of the pentagon, right? Now what's the command to make the turtle *turn around* and head back to HOME? It's not BACK 100. The turtle doesn't turn around using the BACK command.

The command to turn around is RT or LT 180, isn't it? That turns you in the opposite direction from where you were headed. Go ahead, try it.

```
RT 180 FD 100
```

Now where's the turtle?

Type HOME and see what happens.

All the turtle does is turn around. She's back home. So let's try something else. Turn LT 18 and head up to left side of the Pentagon.

```
FD 100 RT 72
```

Now the turtle is set to move up the second side of the pentagon.

Now the big question? How far do you have to turn to send the turtle HOME?

That's not that hard, really. You turned 180 the first time when you wanted to face home, when you did not turn RT 72. So what's $180 - 72$? I get 108. What do you get? Try it:

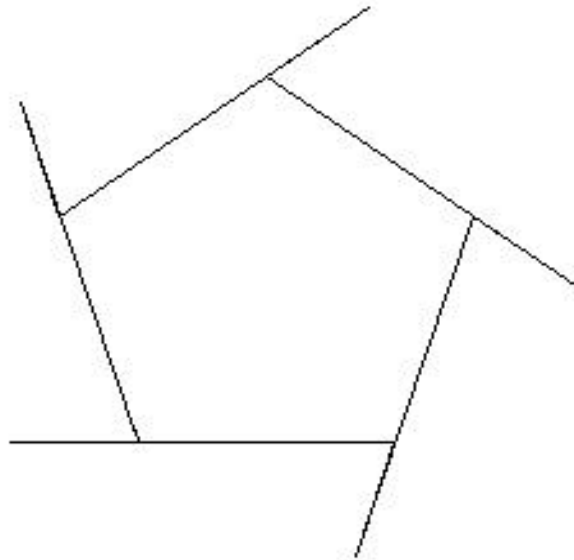
Polygons, Circles, Stars and Stuff

RT 108 FD 100 HOME

There you are! Ernestine is right where she was before.

Believe it or not, you're actually getting somewhere. Don't believe me? Well, try this:

LT 18 REPEAT 5 [FD 150 BK 50 RT 72]



Wow! That's a pretty weird pentagon! But it shows you what you've been doing. It clearly shows the five 72-degree angles Ernestine turns at each corner.

This highlights something that a lot of people get confused about, even Logy and Morf.

When we think about how the angles the turtle moves through to make a polygon, we usually



think about the angles inside the shape.

Well, don't you?

Compare the angles inside a shape with the angles outside a shape.

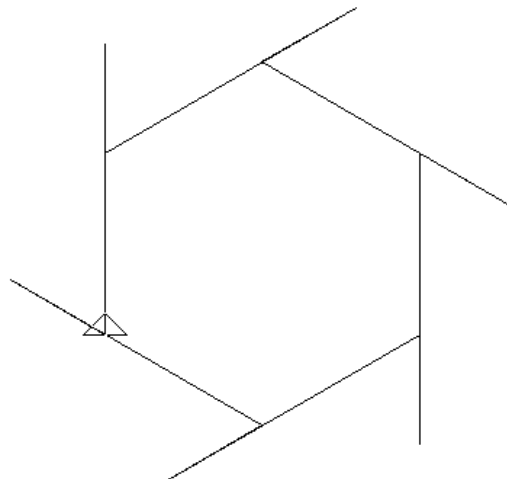
- The angles inside a triangle: 180°
- The angles inside a rectangle (or square): 360°
- The angles inside a pentagon: 540°
- The angles inside a hexagon: 720°

You can figure what the other shapes total yourself. Now what about the outside angles of each shape? What do they total?

Is that a familiar number?

Now where were we. Try the same thing with the hexagon that you did with the pentagon.

REPEAT 6 [FD 150 BK 50 RT 60]



When drawing a hexagon, Ernestine goes FD 150, BACK 50, and turns RT 60 on the outside. That leaves an angle of what on the inside?

Polygons, Circles, Stars and Stuff

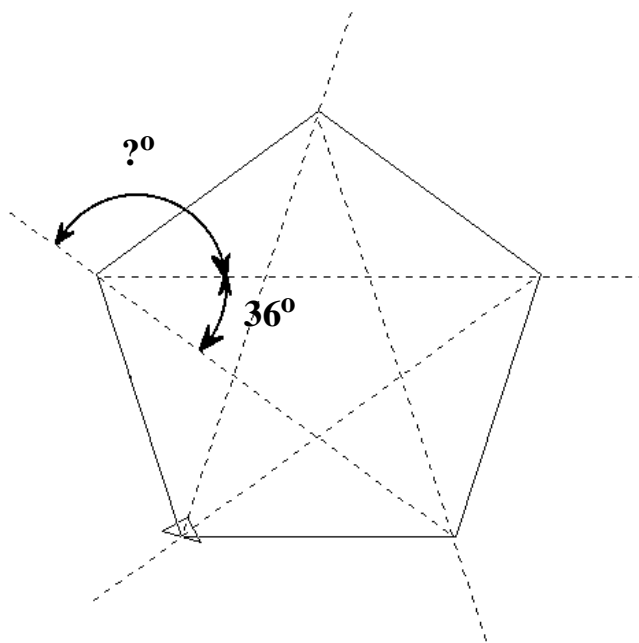
If you said 120, let's make you a big Gold Star!

Look at the drawing of the pentagon with the diagonals below. How many angles do you see at each corner? I see three, how about you?

Are all the angles the same?

It sure looks that way. So if that inside angle is 108, what are each of the three angles? That's 108 divided by 3 or 36, right?

Now, take your pencil and ruler and draw along any one of the diagonal lines to make it go out beyond the pentagon — like Ernestine did when she went FD 150 BK 50. Here you have the large angle outside the star and the small angle inside. You already know the small angle is 36. So what's the big angle?



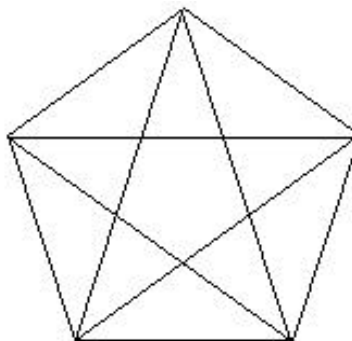
That's $180 - 36$ or 144.

Now you're *really* getting somewhere! So what angle do you need to draw a star?

REPEAT 5 [FD 162 RT _____]

Now add this one:

LT 36 REPEAT 5 [FD 100 RT 72]



How about that! There's the star inside the pentagon, just like the picture you drew on paper. But how'd Ernestine know the distance was 162? Well, that's a lesson in trigonometry that you can read about in *The Great Math Adventure* chapter.

Why not see what you can do with the Star of David now? If you look closely, you'll see there are two triangles in the star.

Here's a hint. If the side of the hexagon is 100, try 172 for the side of the triangle.

TO PENTAGON
REPEAT 5 [FD 100 RT 72]
END

TO STAR
REPEAT 5 [FD 162 RT 144]
END

Polygons, Circles, Stars and Stuff

```
TO STAR.IN.PENT  
RT 18 STAR  
LT 36 PENTAGON  
END
```

```
TO HEXAGON  
REPEAT 6 [FD 100 RT 60]  
END
```

```
TO STAR.OF.DAVID  
RT 30 TRI PU  
RT 90 FD 100 PD  
LT 150 TRI  
END
```

```
TO TRI  
REPEAT 3 [FD 172 RT 120]  
END
```

```
TO STAR.IN.HEX  
STAR.OF.DAVID  
LT 30 HEXAGON  
END
```

Something Else to Think About

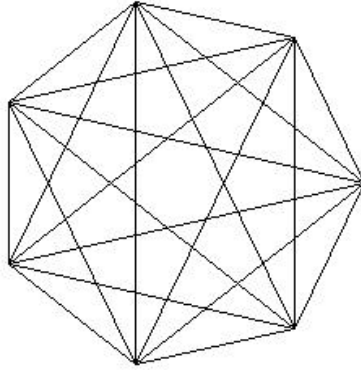
How would you draw a seven-pointed star?

A seven-sided polygon is easy!

```
REPEAT 7 [FD :SIDE RT 360 / 7]
```

Now turn that into a procedure for drawing a seven-pointed star.

Polygons, Circles, Stars and Stuff



Here's a picture of a seven-sided shape with all of its corner connected.

Can you see a seven-pointed star in this picture? How would you draw such a shape?

It's really pretty easy. However, to understand it, let's back up a bit.

The procedure to draw any polygon is

```
REPEAT :REPEATS ~  
[FD :SIZE RT 360 / :REPEATS ]
```

The `:REPEATS` variable stands for the number of sides. Since you're going to be working with Stars now, let's change that to `:POINTS`.

```
REPEAT :POINTS ~  
[FD :SIZE RT 360 / :POINTS ]
```

Now let's take a look at the total of the angles in some of the shapes this procedure draws.

- The angles inside a triangle: 180°
- The angles inside a rectangle (or square): 360°
- The angles inside a pentagon: 540°
- The angles inside a hexagon: 720°
- The angles inside a septagon (7 sides): 900°
- The angles inside an octagon: 1080°
- The angles inside a nanogon (9 sides): 1260°

Look at the triangle and rectangle. To draw each of these shapes, the turtle turns through 360 degrees. Remember, that's one of the rules for a polygon.

Polygons, Circles, Stars and Stuff

To draw a five-pointed star, the turtle actually turns around twice. It starts facing the top of the screen. After five turns, it has moved through 360 degrees twice. To see this clearly, try this:

```
REPEAT 5 [FD 100 RT 144 WAIT 60]
```

If that's true, what would happen if you wrote a procedure like this:

```
TO STAR  
REPEAT 5 [FD 100 RT 360 / 5 * 2]  
END
```

Would this work for a Star of David?

```
TO STAR.OF.DAVID  
REPEAT 6 [FD 100 RT 360 / 6 * 2]  
END
```

No, it won't work. Here's a challenge that will help explain the confusion. Try to draw a six-pointed star without lifting your pencil from the paper and without ever drawing over the same line again.

The Star of David is unique. It's actually two separate triangles linked together. It has been the symbol of Judaism for about 3000 years.

What about a seven-pointed star?

```
REPEAT 7 [FD 100 RT 360 / 7 * 2]
```

What's the difference between this star and the one created by this command:

```
REPEAT 7 [FD 100 RT 360 * 3 / 7]
```

Now see what you can do with 8, 9, 10, 11 points — and other star shapes. If you need help, try this procedure.

```
TO STARS :POINTS :SIZE :MULT  
REPEAT :POINTS [FD :SIZE RT ~  
  360/:POINTS * :MULT]  
END
```

What's that :MULT variable? Take another look at the inside angles.

To draw a pentagon: REPEAT 5 [FD 50 RT 360/5]
The angles inside the polygon total 720 ($360 * 2$).

To draw a hexagon: REPEAT 6 [FD 50 RT 360/6]
The angles inside the polygon total 720 ($360 * 2$).

To draw an octagon: REPEAT 8 [FD 50 RT 360/8] The
angles inside an octagon total 1080 ($360 * 3$).

To draw a 10-sided shape: REPEAT 10 [FD 50 RT 360/
10] Angles total 1440 ($360 * 4$).

To draw a 12-sided polygon: REPEAT 12 [FD 50 RT
360/12] Angles inside the shape total 1800 degrees ($360 * 5$).

To draw a 360-sided shape: REPEAT 360 [FD 50 RT
360/360] Angles inside total 64,440 ($360 * 179$).

Polygons, Circles, Stars and Stuff

Do some experimenting with different numbers to see what happens. Among other things, you'll find you have to change the Rule of 360 a little bit.

Before we said

“Another interesting thing is that when the turtle travels through 360 degrees, she always ends up where she started.”

Now we have to change that to read

“Another interesting thing is that when the turtle travels through 360 degrees, or any multiple of 360, she always ends up where she started.”

You can also say that whenever the turtle travels through any closed shape, when travels through 360 degrees or a multiple of 360 degrees.

Is that really true? What do you think? Have you proven that or not?

From the Center Again

Remember when you had to draw a rectangle around the center? How would you do that for other polygons? How would you find the center of a nine-sided polygon, for example?

Before you start worrying about nine-side shapes, let's start with the simple ones. When you drew a rectangle around the center, you started at the center and went `FD Width / 2, LT 90, FD Length / 2`. This put you in the lower left corner from which you could draw your rectangle.

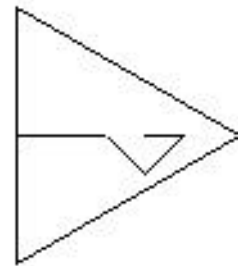
Polygons, Circles, Stars and Stuff

To find the center of a rectangle or a square, you start at a corner and go FD Length / 2, RT (or LT) 90, FD Width / 2. It might be easier to read if it's written like this:

```
TO CENTER :SIDE
REPEAT 4 [FD :SIDE RT 90]
REPEAT 2 [FD :SIDE / 2 RT 90]
END
```

Now let's try this on a triangle.

```
TO CENTER :SIDE
REPEAT 3 [FD :SIDE RT 120]
REPEAT 2 [FD :SIDE / 2 RT 90]
END
```



OK, the line doesn't end in the middle of the triangle. But from what you learned about triangles, you know that the center has to be somewhere along that line. What do you have to do to find that point?

First, let's write a procedure to draw that line that goes through the center.

```
TO FWD :SIDE
FD :SIDE/2
RT 90
FD 100 BK 100
LT 90
FD :SIDE/2
END
```

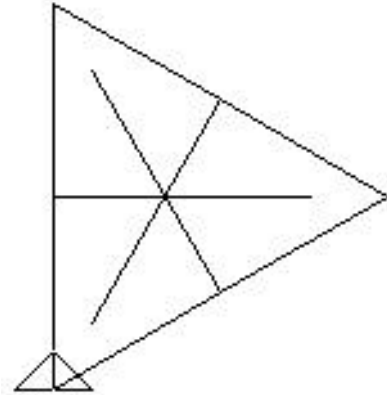
Now let's see what happens when you put that into your shape procedures.

Polygons, Circles, Stars and Stuff

```
TO CENTER :SIDE  
REPEAT 3 [FWD :SIDE RT 120]  
END
```

How about that?
It's seem as if we've
stumbled onto
something here.

Would this work
for other polygons?
Try it out.

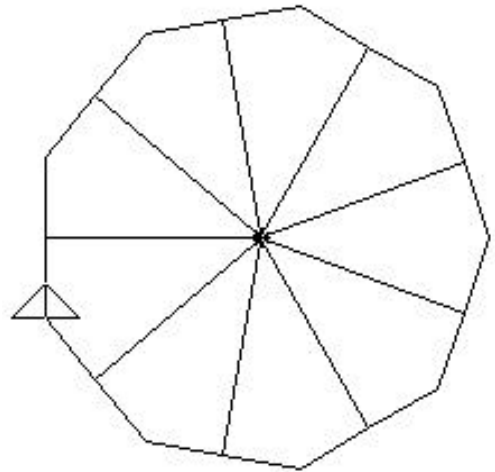


```
TO POLY :REP :SIDE  
REPEAT :REP [FWD :SIDE RT 360 / :REP]  
END
```

Now for the acid
test, the nine-sided
polygon.

Works for me.

Your next
challenge is to figure
out how to measure the
distance to the center.



There's much more to working with polygons, circles,
stars, and stuff in coming chapters. But before you get there,
there's some work to do with Turtle Positions and Coordinates.



Polygons, Circles, Stars and Stuff