



# STITCH IN TIME

**We begin a short series of investigations into the use of LOGO in creating geometric patterns. Here, we show you how the language can be used to draw 'cycloids' — shapes based around circles.**

Earlier in the course, we gave a simple method for drawing a circle using LOGO:

```
TO CIRCLE
  REPEAT 360 [FORWARD 1 RIGHT 1]
END
```

This will give a fairly good approximation of a circle. However, the drawing is painfully slow, although it can be speeded up a little by hiding the turtle. If this procedure doesn't look like a circle on your screen then you need to reset the aspect ratio — you should keep experimenting until you get a circle rather than an ellipse.

Of course, CIRCLE does not actually draw a circle. It draws a 360-sided polygon, but for most purposes this is a good enough approximation. Indeed, for many purposes a polygon with 60, or even 30, sides is quite adequate — and it is much quicker to draw. In this project, our circles will be either 60- or 120-sided polygons, but you can vary this if you like. Larger numbers will give finer detail, smaller numbers will give quicker drawing.

First of all, let's consider what a *cycloid* actually is. Imagine a circle rolling along a straight line. Mark a point on the circumference of your imaginary circle, and then trace out the path this point takes as the circle rotates. The resultant path is what is known as a 'cycloid'. We will use this definition of a cycloid to help us build a program to draw one.

As a first approximation of the cycloid, we'll take 'snapshots' after each 6° turn of a circle rotating along a line across the screen. As the circle turns 6°, it moves ( $2 \times \pi \times \text{radius} \div 60$ ) units forward along the line. So the x co-ordinate of the centre of the circle will have increased by this amount (the y co-ordinate will, of course, remain unaltered). At the same time, the heading of the line joining the 'drawing point' to the centre of the circle will have increased by 6°.

The strategy used in the program involves four simple tasks:

1. move the circle's centre;
2. put the turtle at the centre;
3. point it in the right direction;
4. move it forward by the length of the radius.

This takes the turtle to the next position of the drawing point. We draw a dot on the screen at this point and then repeat the whole process.

The SETSCREEN procedure is the first procedural call from CYCLOID: this deals with a few minor details necessary for the display. SETSCREEN's major tasks are to set the aspect ratio (you will need a different value from ours) and select NOWRAP mode, so that the program stops when the curve goes off the screen.

```
TO MOVECENTRE
  MAKE "XCENT :XCENT + :STEP
END
```

```
TO DOT
  PD
  FORWARD 1
  BACK 1
  PU
END
```

If, instead of taking a point on the circumference of the generating circle, we trace the path made by a point inside the circle, then we get what is known as a *curtate cycloid*. If we take a point outside the circle, but attached to it, we get yet another kind of cycloid — a *prolate cycloid*. To observe these effects we can modify CYCLOID to take an input representing the distance of the drawing point from the circumference. Positive values give curtate cycloids, negative values give prolate cycloids.

```
TO CYCLOID
  SETSCREEN
  MAKE "ANGLESTEP 6
  MAKE "PI 3.14
  MAKE "RADIUS 15
  MAKE "CIRCUMFERENCE 2 * :PI * :RADIUS
  MAKE "STEP :CIRCUMFERENCE / (360 /
  :ANGLESTEP)
  MAKE "XCENT (- 150)
  CYC 0
END
```

```
TO SETSCREEN
  .ASPECT 0.93
  NOWRAP
  DRAW
  PENUP
  HT
END
```

```
TO CYC :ANG
  MOVECENTRE
  SETXY :XCENT 0
  SETH :ANG
  FORWARD :RADIUS
  DOT
  CYC :ANG + :ANGLESTEP
END
```

