# Scratch Art 05

Inspired by the works of TurtleArt - an extension of Logo and the work done by Seymour Papert - we have created a series of activities focused on the creation of art using a simple programming structure. Scratch is a simplified graphical programming environment that mimics a lot of the features of TurtleArt. We can draw lines, change colors, shades, and pen widths -- all to create great geometric art.
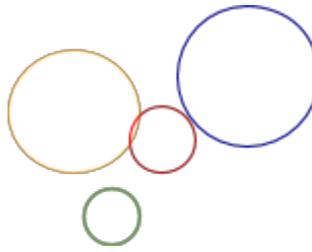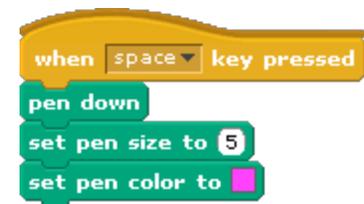
SCRATCH
SKILL REQUIREMENTS
DIFFICULTY 1-5

ELECTRICAL PROTOTYPING

ROBOTICS

SOLDERING

PROGRAMMING

DIY

MATERIALS LIST

● Computer   ● Scratch   ● PicoBoard (optional)

STEP 1: Using the pen
One of the best features in Scratch is the pen. You can set the pen color, size, and position (up or down) with just a few blocks. String together these blocks to get us started.

STEP 2: Advanced shapes -- Curves and Arcs...
In the last activity, we drew regular polygons which have equal length sides and equal angles. What shape do you think this will draw?

STEP 3: Let's draw!
String together these blocks and press the [space] key to run your script. Did you get a circle? How big is the circle? What is the circle's *circumference*?
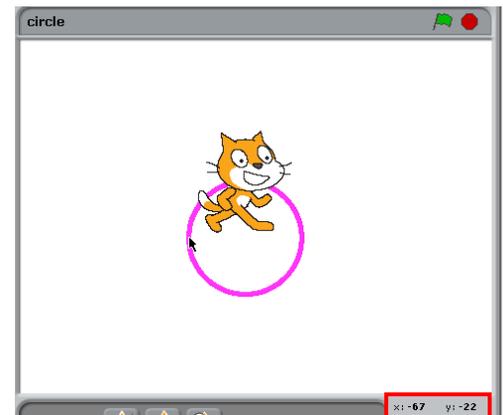
Use the mouse cursor to hover over two points on the circle that are roughly across each other. These define the *diameter* of the circle. Use the distance formula to calculate the diameter.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

First point: (,   )

Second point: (,   )

Distance = _____

STEP 4: Change the step size to 0.5
Change the move block to move 0.5 steps. What is the circumference of the circle now? What happens to the diameter?

STEP 5: Change the step size to 2
Change the move block to move 2 steps. What is the circumference of the circle now? What happens to the diameter?
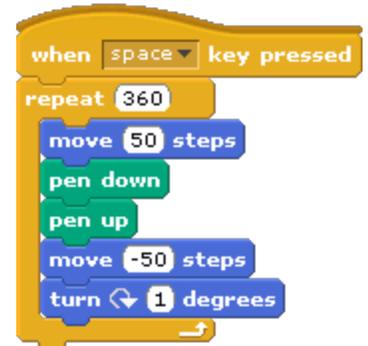

STEP 6: All in the relationship...

How does the circumference compare to the diameter in each of the examples above?
Do you notice that the circumference is always about 3x the length of the diameter? If
we compare all circles of all sizes, and we were really careful with our measurements,
we would find that this relationship is approximately = 3.1415··· or, what
mathematicians call π .

```
          3.14159265358979323846264338327
      5028841971693993751058209749445923
   0781640628620899862803482534211706
  9821    48086          5132
 823      06647          09384
46        09550          58223
17        25359          4081
          2848           1117
          4502           8410
          2701           9385
          21105          55964
          46229          48954
          9303           81964
          4288           10975
          66593          34461
          284756         48233
          78678          31652       71
          2019091        456485      66
          9234603        48610454326648
          2133936        0726024914127
          3724587        00660631558
          817488         152092096
```

STEP 8: There's always more than one way

In most programming exercises, there is more than one way to solve a problem. We
usually call any solution an *algorithm*. Are there other ways of drawing a circle?

Note: Your images will draw faster if you use [hide] to hide the character.

```
when space key pressed
repeat 360
    move 50 steps
    pen down
    pen up
    move -50 steps
    turn ↻ 1 degrees
```

TAKING IT FURTHER

Let's make art...

- On a piece of scratch paper, map out a drawing using standard geometric
  shapes. Use the **go to x: ( ) y: ( )** block and the **pen up** block to move the sprite
  around. Can you draw a car? What about a smiley face?

  ```
  go to x: 0 y: 0
  ```
  ```
  pen up
  ```

- What other shape combinations can you make?