**Basic Drawing in Processing**

**Carson Foster**

Drawing Basic Shapes

The Point

 The most basic ‘shape’ that you can draw is a point. Yes, a single point. Processing’s syntax for drawing a point is quite simple:



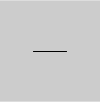
The two arguments represent the x and y coordinates of the point.



In this example, the point function creates a single point in the middle of the screen. The mechanism pictured here is quite useful: after the size function is used, the variable names width and height can be used to refer to the width and height of the drawing space, respectively.

The Line

From the humble point, we now arrive at the line: a connection between two points. In Processing, the syntax to draw a line is the same as the previously discussed representation: the coordinates for the start point, and then the coordinates for the end point, as you can see here:





In the following example, a line is drawn in the center third of the drawing space.



The Triangle

The line had two points, so can you make a guess as to what sort of shape we’ll be discussing now? That’s right, very observant of you. The triangle is our next shape, and is represented as you would expect:



These three points define the triangle, in the way you would imagine.



As you can see, this command draws a nice isosceles triangle in the middle of the drawing area.

The Square

As the next step in the logical progression of shapes, we next look at the syntax for drawing a square. The three necessary arguments to draw a square in Processing are the coordinates of the upper left-hand point of the square, and then a number that is both the width and height of the square.



In the following example, we’ll draw a square at the bottom right of the screen, and its area will be one fourth of the sketch’s area.

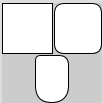


The Rectangle

The more general form of the square, the rectangle is the next shape we’ll cover. The rectangle has a couple of different sets of arguments you can use. All of them include the basic representation of the rectangle that we discussed: the coordinates of the top-left corner, and then the width and the height. The first set of arguments is just these. The other two specify a radius/radii for drawing rounded rectangles. The second set specifies a single radius that it applies to all corners. The last set of arguments needs the 4 regular arguments, and 4 arguments representing the separate radii of each corner, starting at the top left and moving clockwise.



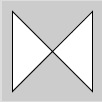
As you can see, there are a couple options to choose from when drawing a rectangle. The following example will show all of them.



I’ve spaced the rectangles out a little bit to make sure that their outlines are visible. Take note of how the different corner radii appear.

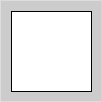
The Quadrilateral

Next, we’ll discuss the more general form of the rectangle: the quadrilateral. As you now definitely suspect, a quadrilateral is specified by four points in Processing. These points must be provided in clockwise or counterclockwise order.



The same points can construct different shapes if they are provided in a different order:



This creates a bowtie shape, while the following will create a rectangle:



arc

circle

ellipse

Bezier

vertex

Color, Stroke, and Fill

text

Transformations

text