

**Processing:****Start simple**

Processing tutorials #1, #2, #3  
 Add color to everything - it makes it fun  
 Rectangles & circles  
 Rainbow color wheel

**More difficult**

Put them together into a drawing  
 Equilateral polygon, 5 pointed star

**Difficulties:** Transferring to Processing from NetLogo was just a matter of learning the new syntax. Since many of the concepts were started in Scratch, it was a matter of transferring the knowledge and adding more difficult programs. Students needed to develop the code from the concepts. This was difficult, but also very rewarding. They often would have the code written, but had a semicolon in the wrong place, which would cause it to not run properly. Having them trouble shoot each others code was very helpful.

WarmUp & Assignment handouts

**Project to work on: Spinning Color Wheel**

This program will create a wheel of the rainbow colors. When the mouse button is pressed, the wheel will spin. The wheel will be drawn as lines from the center of the canvas using the trig function  $x = r * \cos\theta$  and  $y = r * \sin\theta$ . Since Processing uses radians, the 360° of the circle will need to be converted using the *radians()* function. You will also need to set the colorMode to HSB.

Programming steps:

Global variables:

integer: *radius, angle*  
 float: *x, y* for the center of the canvas

*setup()*

Create a canvas 500 x 500  
 Set the radius to 150  
 Set the colorMode(HSB, 360, 100, 100) a different color for each of the 360°, saturation and brightness will be 0 to 100%  
 Set a background color from the color pallet, black works great  
 To center the wheel, set to ½ the width, and y to ½ the height

*draw()*

Set the variable *angle* to zero  
 Draw a line from the center out the correct radius.  
 Now change the color of the line to be the value of *angle* with a saturation and balance of 100.  
 Put this in a *for* loop 360 times, so it will now draw all the colors of the rainbow.

To have the wheel spin

add a global integer *rotate*  
 In *setup()* set it to zero  
 In *draw()*

Use an *if* statement to test *if(mousePressed)*, then increase *rotate* by 20. This can be changed, but it is a good starting point.  
 Add *rotate* to *angle* to change the location of the line which causes the wheel to spin.