Rotate Around a Shape's Center

11 min

As we have seen in the previous exercise, the `rotate()` function on its own is not enough to rotate a shape around its center. When we `rotate()` an element, it often moves off the screen because the whole p5.js canvas is being rotated. In order to rotate a shape around its center, you first need to `translate()` the shape to be in the middle of the canvas.

```
// Move the origin to the center of the canvas
translate(width / 2, height / 2);
```

Then, you need to rotate your shape to the desired radian value. You can use the `radians()` function to convert the value passed to the `rotate()` function is in radians. In the example below, we use the p5.js `frameCount` variable to get the shape to rotate incrementally.

```
// Rotate by frameCount radians
rotate(radians(frameCount));
```

Lastly, you need to draw the shape and set its origin point to be at the `CENTER` of the shape.

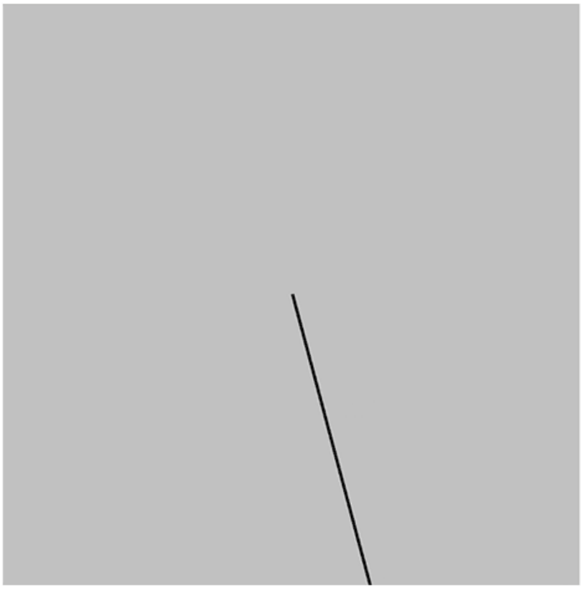
```
// Set the origin point to be the rectangle's center
rectMode(CENTER);
rect (0, 0, 1, height);
```

In order to rotate a shape around its top-left corner, you can set the mode to `CORNER` instead of `CENTER`. Since `CORNER` is the default origin point for drawing rectangles, you don't need to explicitly call the `rectMode()` function. The two code examples below would produce the same result—a rectangle rotating around its top-left corner.

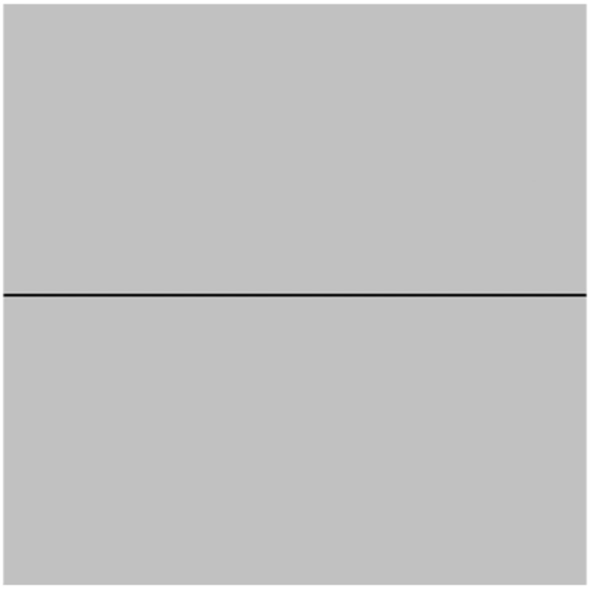
```
// Rotates from rectangle's top-left corner
rectMode(CORNER);
rotate(radians(frameCount));
rect (0, 0, 1, height);

// Also rotates from rectangle's top-left corner because rectMode()
is CORNER by default
rotate(radians(frameCount));
rect (0, 0, 1, height);
```

The diagram below shows the difference between rotating a shape with a `CORNER` versus a `CENTER` origin point.



Rotation From Corner



Rotation From Center

Remember that the values for the transformation functions are accumulative. For example, the values passed into the first `rotate()` function will be added to any of the `rotate()` functions that are called afterward.

Instructions

1.

Below the `fill()` function, specify the origin points of the rectangles to be the `CENTER`.

Hint

You can use the `rectMode()` function to set how the x and y positions of a rectangle's origin point is interpreted—either `CENTER` or `CORNER`.

2.

After the `rectMode()` function, translate the rectangles `width / 2` to the right and `height / 2` to the bottom. This will move your rectangles to the center of the canvas.

Hint

Use the `translate()` function to translate a shape. To translate a shape 100 pixels right and 150 pixels down, your code will look like this:

```
translate(100, 150);
```

3.

Below the `translate()` function, rotate your rectangles by a quarter pi (π).

Hint

To rotate a shape, use the `rotate()` function. Remember that you have to spell out the radian amount in all caps, like `HALF_PI`.

To rotate a shape by `HALF_PI`, your code will look like this:

```
rotate(HALF_PI);
```

4.

Rotate the second rectangle by `frameCount / 2` in radians. Remember that you can convert a value to be in radians using the `radians()` function.

Now you should see the rectangle with a blue outline rotating from its center!
Hint
The `radians()` function converts a degree value to be in radians. This means that when the `frameCount` is at 180, the radian amount is 3.14 (π)—a half-circle rotation.

sketch.js

```
function setup() {
  createCanvas(windowWidth, windowHeight);
}

function draw() {
  background(0, 15);
  stroke(0, 0, 255);
  fill(255, 0, 0, 50);

  // TODO: Specify the rectangle's origin point to be center
  rectMode(CENTER);

  // TODO: Translate rectangle to the center of canvas
  translate(width / 2, height / 2);

  // TODO: Rotate rectangle by a quarter pi
  rotate(QUARTER_PI);
  rect(0, 0, 450, 450); // First rectangle

  stroke(0, 0, 255);
  fill(255, 0);

  // TODO: Rotate rectangle by frameCount / 2 in radians
  rotate(radians(frameCount / 2));
  rect(0, 0, 450, 450); // Second rectangle
}
```