

Using Combinations of Transform Functions

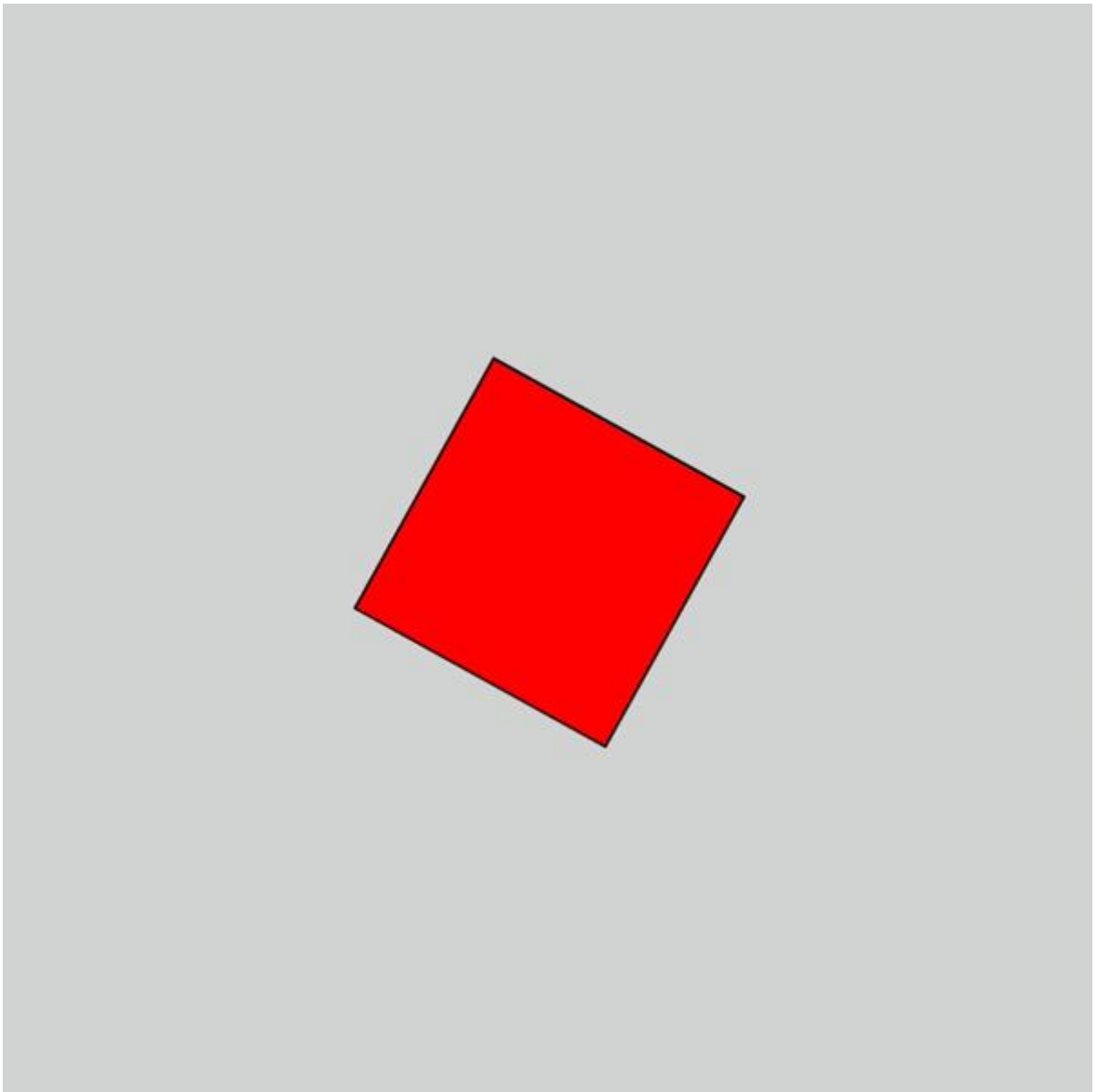
8 min

Interesting and complex sketches can be created when you combine the `translate()`, `rotate()`, and `scale()` functions.

When you apply multiple transformations, the order makes a big difference. A `translate()` followed by a `rotate()` then a `scale()` function will not create the same results if the order of operations were changed.

Look at the code below to see how each transformation functions are pieced together to create the animation of a rectangle, placed in the middle of the canvas, rotating from its center.

```
rectMode(CENTER);  
translate(width/2, height/2);  
rotate(radians(frameCount));  
  
fill(255,0,0);  
rect(0, 0, 100, 100);
```



Now, let's change the order of operations. This time, we call the `translate()` function before the `rotate()` function. Everything else will remain the same, but notice how the resulting animation is wildly different.

```
rectMode(CENTER);  
rotate(radians(frameCount));  
translate(width/2, height/2);  
  
fill(255,0,0);  
rect(0, 0, 100, 100);
```



As you can see that the rectangle rotates from the upper left corner of the canvas in this new version. The rectangle is still translated to the center of the canvas, but only after the `rotate()` function has been applied. The rectangle leaves the canvas for a time because the rotation is happening at the default origin of the canvas at its top-left corner.

Instructions

1. After the first `push()` function, translate the first rectangle with a red outline to be in the center of the canvas.

Calculate the x and y positions for the `translate()` function relative to the `width` and `height` variables.

Hint

You can use the `translate()` function to translate the canvas as follows:

```
translate(x, y);
```

where `x` is the number of pixels to be moved along the x-axis, and `y` is the number of pixels to be moved along the y-axis.

2.

Below the `translate()` function, write a line to rotate the rectangle by `frameCount` in radians. Notice how the red rectangle is rotating at the center of the canvas.

Hint

The syntax to rotate a shape is:

```
rotate(angleValue);
```

In order to convert a degree or numeric value into radians you can use the `radians()` function.

3.

Below the second `push()` function for the blue rectangle, rotate the rectangle by the `frameCount` in radians.

Hint

The syntax to rotate a shape is:

```
rotate(angleValue);
```

Make sure to use the `radians()` function to convert the `frameCount` value to radians.

4.

Below the `rotate()` function you just added, translate the second rectangle to be in the center of the canvas.

Notice how this blue rectangle is rotating from the top-left corner and only appears on the canvas once in a while.

Hint

Make sure to calculate the x and y positions for the `translate()` function relative to the `width` and `height` variables.

sketch.js

```
function setup() {  
  createCanvas(windowWidth, windowHeight);  
  frameRate(12);  
}  
  
function draw() {  
  background(0, 25);  
  rectMode(CENTER);  
  
  // Red Rectangle: Translate then rotate  
  push();  
  
  // TODO: Translate to the center of canvas  
  translate(width/2, height/2);  
  
  // TODO: Rotate by frameCount  
  rotate(radians(frameCount));  
  
  noFill();  
  stroke(255, 0, 0);  
  rect(0, 0, windowWidth, windowHeight);  
  pop();  
  
  // Blue Rectangle: Rotate then translate  
  push();  
  
  // TODO: Rotate by frameCount  
  rotate(radians(frameCount));  
  
  // TODO: Translate to the center of canvas  
  translate(width/2, height/2);  
  
  noFill();  
  stroke(0, 0, 255);  
  rect(0, 0, windowWidth, windowHeight);  
  pop();  
}
```