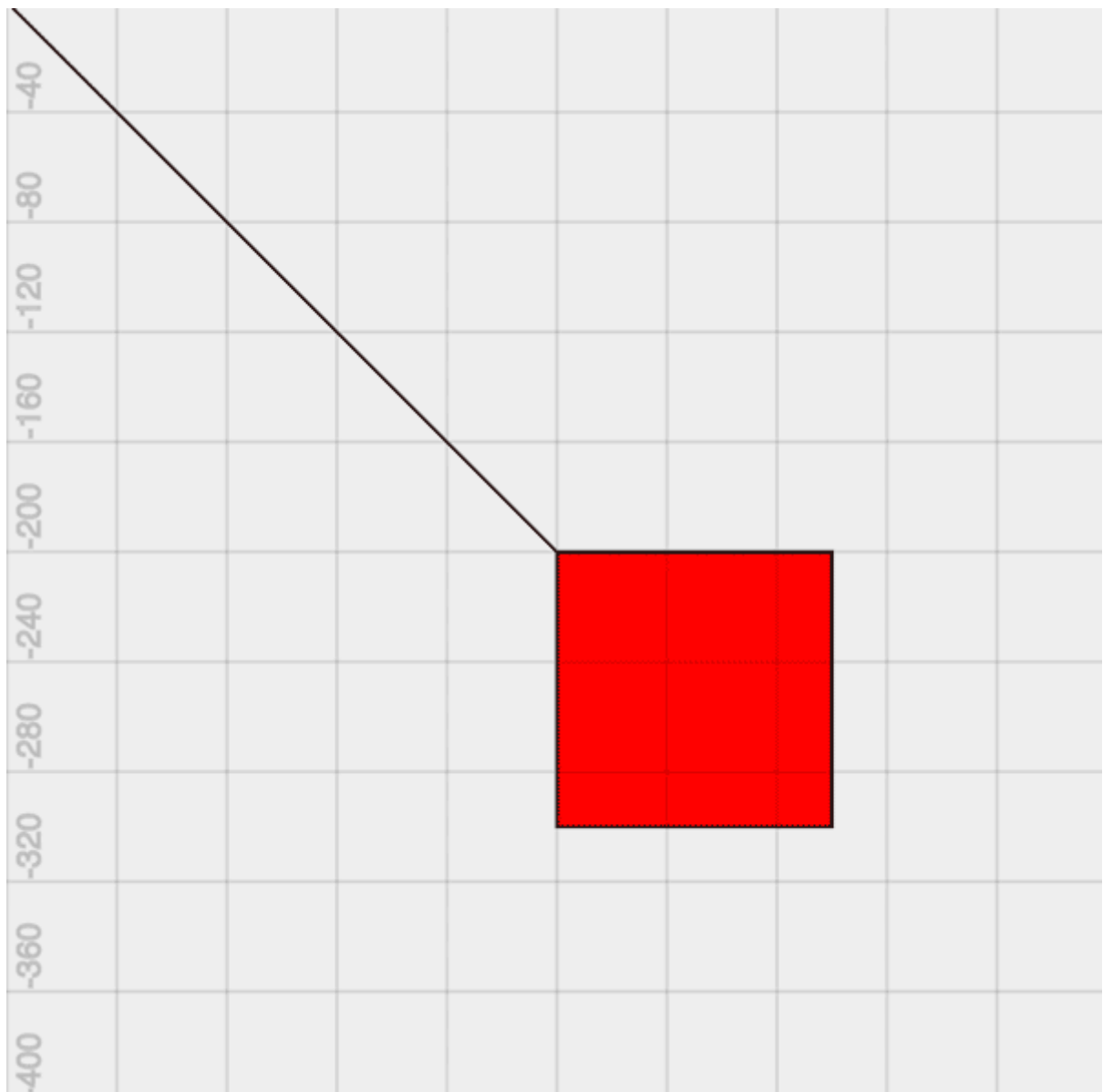


Rotate

9 min

Rotating an element in p5.js means rotating the canvas from its top-left corner at (0, 0). Therefore, all the elements drawn on the coordinate system rotates together with the canvas.

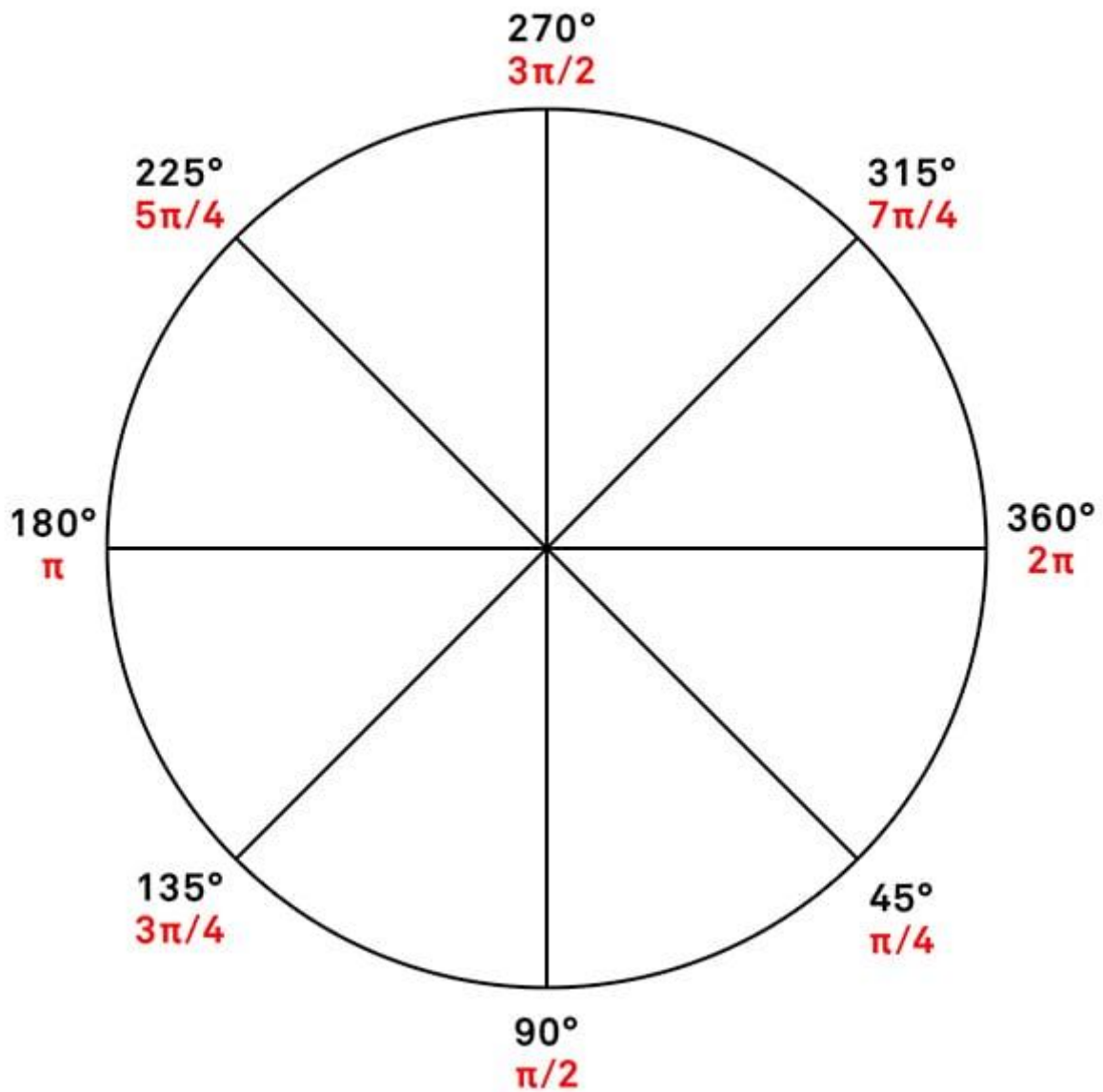
The diagram below shows that the coordinate system is being rotated, along with the rectangle drawn on the canvas. The red rectangle shows the original location, while the gray rectangle shows its positions when rotated. Notice how the gray square goes off the canvas. This happens because the shape is situated on the rotating p5.js coordinate system.



You can rotate the canvas using the `rotate()` function:

```
rotate(angle);
```

The `rotate()` function takes one argument—the angle of rotation represented by the `angle` variable in the above code. By default, the angle of rotation is interpreted as a [radian](#) value—a unit of measuring angles. A single rotation, or a full circle, has 360° in degrees or 2π (6.28) in radians. A half rotation has a 180° in degrees or π (3.14) in radians.



p5.js provides representations of pi (π) values, which you can use by spelling out the words like below:

- π = `PI`
- $\pi / 2$ = `HALF_PI`
- $\pi / 4$ = `QUARTER_PI`
- 2π = `TWO_PI`

The code below shows how you would pass in the `PI`, `HALF_PI`, `QUARTER_PI` or `TWO_PI` value to the `rotate()` function:

```
rotate(HALF_PI);
```

You can also convert a degree value into a radian value by calling the `radians()` function. For example, `radians(180)` would equate to 3.14, the equivalent to one π . It is possible to use the `radians()` function inside the `rotate()` function.

```
rotate(radians(180));
```

Alternatively, you can specify the unit of measurement of angles by calling the `angleMode()` function. Similar to how you set the `rectMode()` and `ellipseMode()` to change the shape's origin, the `angleMode()` determines if the value inside the `rotate()` function is interpreted as either degrees or radians.

You can set the mode to `DEGREES` by writing:

```
angleMode(DEGREES);
```

You can also specify the mode to be `RADIANS`, but keep in mind that the default mode for angles and the `rotate()` function is already `RADIANS`.

```
angleMode(RADIANS);
```

Instructions

1.

Below the `fill()` function, rotate the canvas by a quarter pi (π).

Hint

To rotate a shape, use the `rotate()` function like below:

```
rotate(angleAmount);
```

where `angleAmount` is the amount of angle to rotate the canvas by.

Remember, you need to spell out the word for π as `PI`.

2.

Below the `rotate()` function you just wrote, specify the angle mode to be `DEGREES`.

Notice how changing the angle mode to degrees, alters the rotation amount of the elements.

Hint

The `angleMode()` function is used to specify whether the `rotate()` function should be interpreted as `DEGREES` or `RADIANS`.

3.

Underneath the `angleMode()` function, rotate the ellipses' `for` loop by 30 degrees.

Hint

To rotate a function by 60 degrees (given `DEGREES` angle mode), your code will look like this:

```
rotate(60);
```

4.

Right above the second `for` loop, rotate the rectangle pattern by `frameCount * 10` radians.

Be sure to use the `radians()` function to convert the value into radians.

Hint

Remember, you can call the `radians()` function to convert a degree value into a radian value inside the `rotate()` function.

To convert 60 degrees into its radian value, you can write:

```
radians(60); // returns 1.0471975511965976
```

You may notice that since the angle mode was previously set to `DEGREES`, the converted radian value is actually used as a degree value! What the `radians()` function does here is to slow down the `frameCount * 10` value for the rotation of the blue grid of rectangles.

sketch.js

```
function setup() {  
  createCanvas(windowWidth, windowHeight);  
}  
  
function draw() {  
  background(0);  
  fill(0);  
  
  // TODO: Rotate the ellipse pattern by a quarter pi  
  rotate(QUARTER_PI);  
  
  // TODO: Specify the angle mode to be degrees  
  angleMode(DEGREES);  
  
  // TODO: Rotate by 30 degrees  
  rotate(30);  
  
  for (let i = 0; i < width * 2 ; i += 75) {  
    for (let j = 0; j < height * 2 ; j += 75) {  
      fill(255, 0, 0, 180);  
      ellipse(i, j, 50, 50);  
    }  
  }  
  
  // TODO: Rotate by frameCount * 10  
  
  rotate(radians(frameCount * 10));  
  for (let i = 0; i < width * 2 ; i += 75) {  
    for (let j = 0; j < height * 2; j += 75) {  
      fill(0, 0, 255, 180);  
      rect(i, j, 50, 50);  
    }  
  }  
}
```