

# JavaScript Rotate

**Summary:** in this tutorial, you'll learn how to use the JavaScript `rotate()` method to rotate drawing objects.

## Introduction to JavaScript rotate() canvas API

The `rotate()` is a method of the 2D drawing context. The `rotate()` method allows you to rotate a drawing object on the canvas.

Here is the syntax of the `rotate()` method:

```
ctx.rotate(angle)
```

The `rotate()` method accepts a rotation angle in radians.

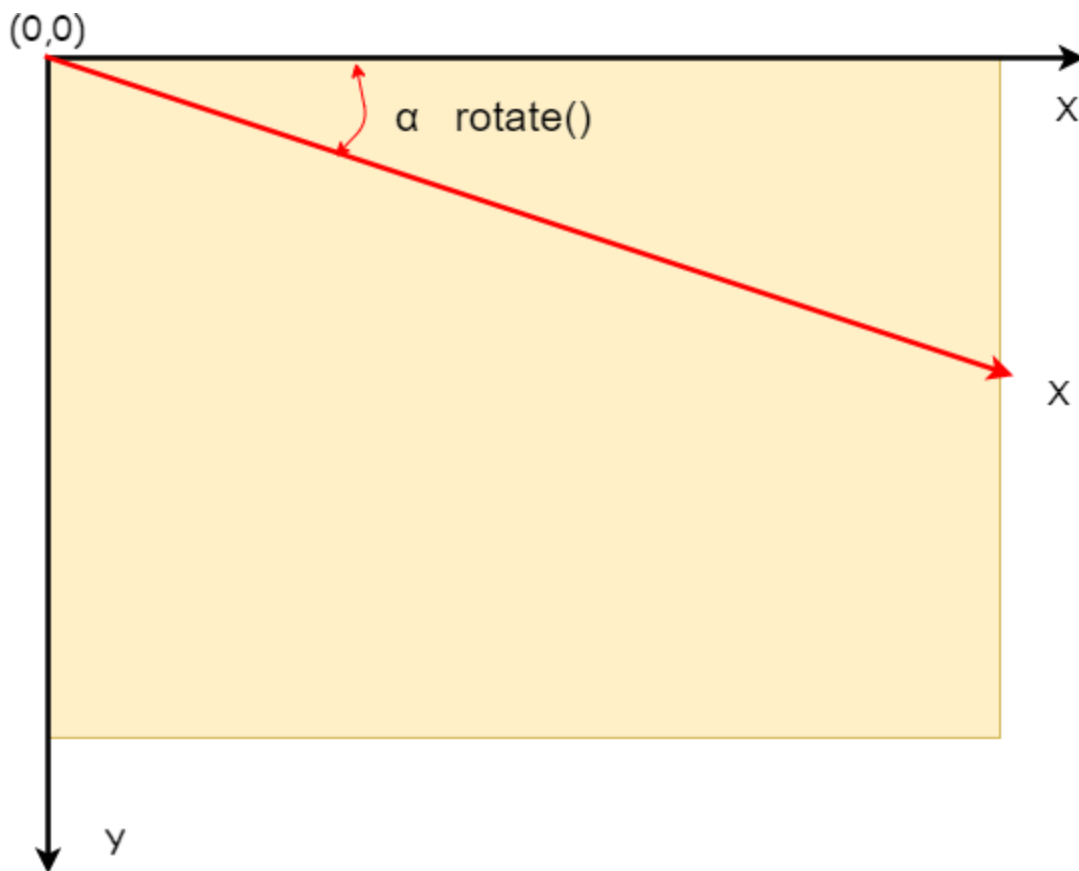
If the angle is positive, the rotation is clockwise. In case the angle is negative, the rotation is counterclockwise.

To convert a degree to a radian, you use the following formula:

```
degree * Math.PI / 180
```

When adding a rotation, the `rotate()` method uses the canvas origin as the rotation center point.

The following picture illustrates the rotation:



If you want to change the rotation center point, you need to move the origin of the canvas using the `translate()` method.

## JavaScript rotate() example

The following example draws a red rectangle starting from the center of the canvas. It then translates the origin of the canvas to the canvas' center and draws the second rectangle with a rotation of 45 degrees:

### HTML

```
<canvas id="canvas" height="300" width="500">
</canvas>
```

### JavaScript

```
const canvas = document.querySelector('#canvas');

if (canvas.getContext) {
```

```
// rectangle's width and height
const width = 150,
      height = 20;

// canvas center X and Y
const centerX = canvas.width / 2,
      centerY = canvas.height / 2;

const ctx = canvas.getContext('2d');

// a red rectangle
ctx.fillStyle = 'red';
ctx.fillRect(centerX, centerY, width, height);

// move the origin to the canvas' center
ctx.translate(centerX, centerY);

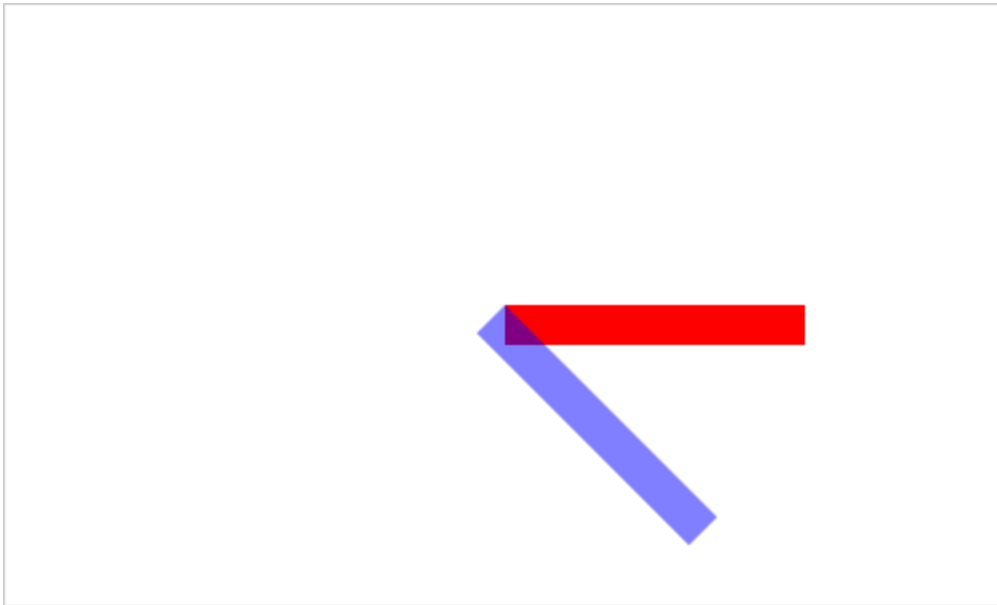
// add 45 degrees rotation
ctx.rotate(45 * Math.PI / 180);

// draw the second rectangle
ctx.fillStyle = 'rgba(0,0,255,0.5)';
ctx.fillRect(0, 0, width, height);

}
```

[Here is the demo link.](#)

Output:



## Summary

- Use JavaScript `rotate()` method to rotate a drawing object on a canvas.