

# rect()

Draws a rectangle.

A rectangle is a four-sided shape defined by the `x`, `y`, `w`, and `h` parameters. `x` and `y` set the location of its top-left corner. `w` sets its width and `h` sets its height. Every angle in the rectangle measures 90°. See [rectMode\(\)](#) for other ways to define rectangles.

The version of `rect()` with five parameters creates a rounded rectangle. The fifth parameter sets the radius for all four corners.

The version of `rect()` with eight parameters also creates a rounded rectangle. Each of the last four parameters set the radius of a corner. The radii start with the top-left corner and move clockwise around the rectangle. If any of these parameters are omitted, they are set to the value of the last radius that was set.

## Examples



```
function setup() {
  createCanvas(100, 100);

  background(200);

  rect(30, 20, 55, 55);

  describe('A white square with a black outline on a gray
  canvas.');
}
```

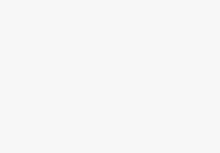


```
function setup() {
  createCanvas(100, 100);

  background(200);

  rect(30, 20, 55, 40);

  describe('A white rectangle with a black outline on a gray
  canvas.');
}
```



```
function setup() {
  createCanvas(100, 100);

  background(200);

  // Give all corners a radius of 20.
  rect(30, 20, 55, 50, 20);

  describe('A white rectangle with a black outline and round
  edges on a gray canvas.');
}
```



```
function setup() {
  createCanvas(100, 100);

  background(200);

  // Give each corner a unique radius.
  rect(30, 20, 55, 50, 20, 15, 10, 5);

  describe('A white rectangle with a black outline and round
  edges of different radii.');
}
```

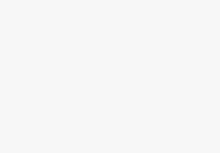


```
function setup() {
  createCanvas(100, 100, WEBGL);

  background(200);

  rect(-20, -30, 55, 55);

  describe('A white square with a black outline on a gray
  canvas.');
}
```



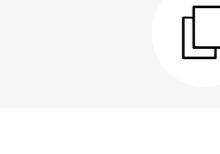
```
function setup() {
  createCanvas(100, 100, WEBGL);

  describe('A white square spins around on gray canvas.');
}

function draw() {
  background(200);

  // Rotate around the y-axis.
  rotateY(frameCount * 0.01);

  // Draw the rectangle.
  rect(-20, -30, 55, 55);
}
```



## Syntax

```
rect(x, y, w, [h], [tl], [tr], [br], [bl])
```



```
rect(x, y, w, h, [detailX], [detailY])
```



## Parameters

<code>x</code>	Number: x-coordinate of the rectangle.
<code>y</code>	Number: y-coordinate of the rectangle.
<code>w</code>	Number: width of the rectangle.
<code>h</code>	Number: height of the rectangle.
<code>tl</code>	Number: optional radius of top-left corner.
<code>tr</code>	Number: optional radius of top-right corner.
<code>br</code>	Number: optional radius of bottom-right corner.
<code>bl</code>	Number: optional radius of bottom-left corner.
<code>detailX</code>	Integer: number of segments in the x-direction (for WebGL mode).
<code>detailY</code>	Integer: number of segments in the y-direction (for WebGL mode).

This page is generated from the comments in [src/core/shape/2d\\_primitives.js](#). Please feel free to edit it and submit a pull request!

## Related References

[arc](#)

Draws an arc.

[circle](#)

Draws a circle.

[ellipse](#)

Draws an ellipse (oval).

[line](#)

Draws a straight line between two points.



p5.js

Resources

Reference

Tutorials

Examples

Contribute

Community

About

Start Coding

Donate

Information

Download

Contact

Copyright

Privacy Policy

Terms of Use

Socials

GitHub ↗

Instagram ↗

X ↗

YouTube ↗

Discord ↗

Forum ↗

Donate Today! Support p5.js and the Processing Foundation.

