

[Start Coding](#)[Donate](#)

Reference

2D Primitives

[arc\(\)](#)
[circle\(\)](#)
[ellipse\(\)](#)
[line\(\)](#)
[point\(\)](#)
[quad\(\)](#)
[rect\(\)](#)
[square\(\)](#)
[triangle\(\)](#)

Shape

Color

Typography

Image

Transform

Environment

Reference > quad()

quad()

Draws a quadrilateral (four-sided shape).

Quadrilaterals include rectangles, squares, rhombuses, and trapezoids. The first pair of parameters (`x1, y1`) sets the quad's first point. The next three pairs of parameters set the coordinates for its next three points (`x2, y2`), (`x3, y3`), and (`x4, y4`). Points should be added in either clockwise or counter-clockwise order.

The version of `quad()` with twelve parameters allows the quad to be drawn in 3D space. Doing so requires adding the `WEBGL` argument to `createCanvas()`.

The thirteenth and fourteenth parameters are optional. In WebGL mode, they set the number of segments used to draw the quadrilateral in the x- and y-directions. They're both 2 by default.

Examples



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

  background(200);

  quad(20, 20, 80, 20, 80, 20, 20, 80);

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



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

  background(200);

  quad(20, 30, 80, 30, 80, 70, 20, 70);

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



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

  background(200);

  quad(50, 62, 86, 50, 50, 38, 14, 50);

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



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

  background(200);

  quad(20, 50, 80, 30, 80, 70, 20, 70);

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



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

  background(200);

  quad(-30, -30, 30, -30, 30, 30, -30, 30);

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

function draw() {
  background(200);

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

  // Draw the quad.
  quad(-30, -30, 0, 30, -30, 0, 30, 30, 20, -30, 30, -20);
}
```

Syntax

```
quad(x1, y1, x2, y2, x3, y3, x4, y4, [detailX], [detailY])
```

```
quad(x1, y1, z1, x2, y2, z2, x3, y3, z3, x4, y4, z4, [detailX], [detailY])
```

Parameters

<code>x1</code>	Number: the x-coordinate of the first point.
<code>y1</code>	Number: the y-coordinate of the first point.
<code>x2</code>	Number: the x-coordinate of the second point.
<code>y2</code>	Number: the y-coordinate of the second point.
<code>x3</code>	Number: the x-coordinate of the third point.
<code>y3</code>	Number: the y-coordinate of the third point.
<code>x4</code>	Number: the x-coordinate of the fourth point.
<code>y4</code>	Number: the y-coordinate of the fourth point.
<code>detailX</code>	Integer: number of segments in the x-direction.
<code>detailY</code>	Integer: number of segments in the y-direction.
<code>z1</code>	Number: the z-coordinate of the first point.
<code>z2</code>	Number: the z-coordinate of the second point.
<code>z3</code>	Number: the z-coordinate of the third point.
<code>z4</code>	Number: the z-coordinate of the fourth point.

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.



<img alt="Attribution-NonCommercial-ShareAlike license logo" data-bbox="160 1951 250