

arc()

Draws an arc.

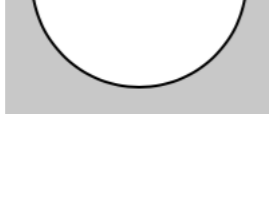
An arc is a section of an ellipse defined by the `x`, `y`, `w`, and `h` parameters. `x` and `y` set the location of the arc's center. `w` and `h` set the arc's width and height. See `ellipse()` and `ellipseMode()` for more details.

The fifth and sixth parameters, `start` and `stop`, set the angles between which to draw the arc. Arcs are always drawn clockwise from `start` to `stop`. The fifth and sixth parameters, `start` and `stop`, set the angles between which to draw the arc. By default, angles are given in radians, but if `angleMode(DEGREES)` is set, the function interprets the values in degrees.

The seventh parameter, `mode`, is optional. It determines the arc's fill style. The fill modes are a semi-circle (`OPEN`), a closed semi-circle (`CHORD`), or a closed pie segment (`PIE`).

The eighth parameter, `detail`, is also optional. It determines how many vertices are used to draw the arc in WebGL mode. The default value is 25.

Examples

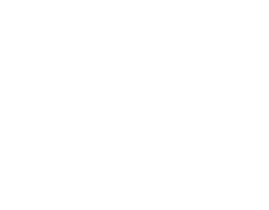


```
function setup() {
  createCanvas(100, 100);

  background(200);

  arc(50, 50, 80, 80, 0, PI + HALF_PI);

  describe('A white circle on a gray canvas. The top-right quarter of the circle is missing.');
```



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

  background(200);

  arc(50, 50, 80, 40, 0, PI + HALF_PI);

  describe('A white ellipse on a gray canvas. The top-right quarter of the ellipse is missing.');
```



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

  background(200);

  // Bottom-right.
  arc(50, 55, 50, 50, 0, HALF_PI);

  noFill();

  // Bottom-left.
  arc(50, 55, 60, 60, HALF_PI, PI);

  // Top-left.
  arc(50, 55, 70, 70, PI, PI + QUARTER_PI);

  // Top-right.
  arc(50, 55, 80, 80, PI + QUARTER_PI, TWO_PI);

  describe(
    'A shattered outline of an circle with a quarter of a white circle at the bottom-right.'
  );
}
```



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

  background(200);

  // Default fill mode.
  arc(50, 50, 80, 80, 0, PI + QUARTER_PI);

  describe('A white circle with the top-right third missing. The bottom is outlined in black.');
```



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

  background(200);

  // OPEN fill mode.
  arc(50, 50, 80, 80, 0, PI + QUARTER_PI, OPEN);

  describe(
    'A white circle missing a section from the top-right. The bottom is outlined in black.'
  );
}
```

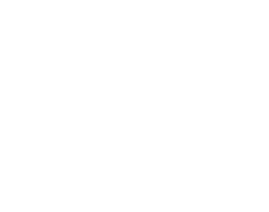


```
function setup() {
  createCanvas(100, 100);

  background(200);

  // CHORD fill mode.
  arc(50, 50, 80, 80, 0, PI + QUARTER_PI, CHORD);

  describe('A white circle with a black outline missing a section from the top-right.');
```



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

  background(200);

  // PIE fill mode.
  arc(50, 50, 80, 80, 0, PI + QUARTER_PI, PIE);

  describe('A white circle with a black outline. The top-right third is missing.');
```



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

  background(200);

  // PIE fill mode.
  arc(0, 0, 80, 80, 0, PI + QUARTER_PI, PIE);

  describe('A white circle with a black outline. The top-right third is missing.');
```



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

  background(200);

  // PIE fill mode with 5 vertices.
  arc(0, 0, 80, 80, 0, PI + QUARTER_PI, PIE, 5);

  describe('A white circle with a black outline. The top-right third is missing.');
```



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

  describe('A yellow circle on a black background. The circle opens and closes its mouth.');
```

```
function draw() {
  background(0);

  // Style the arc.
  noStroke();
  fill(255, 255, 0);

  // Update start and stop angles.
  let biteSize = PI / 16;
  let startAngle = biteSize * sin(frameCount * 0.1) + biteSize;
  let endAngle = TWO_PI - startAngle;

  // Draw the arc.
  arc(50, 50, 80, 80, startAngle, endAngle, PIE);
}
```

Syntax

```
arc(x, y, w, h, start, stop, [mode], [detail])
```

Parameters

<code>x</code>	Number: x-coordinate of the arc's ellipse.
<code>y</code>	Number: y-coordinate of the arc's ellipse.
<code>w</code>	Number: width of the arc's ellipse by default.
<code>h</code>	Number: height of the arc's ellipse by default.
<code>start</code>	Number: angle to start the arc, specified in radians.
<code>stop</code>	Number: angle to stop the arc, specified in radians.
<code>mode</code>	Constant: optional parameter to determine the way of drawing the arc. either CHORD, PIE, or OPEN.
<code>detail</code>	Integer: optional parameter for WebGL mode only. This is to specify the number of vertices that makes up the perimeter of the arc. Default value is 25. Won't draw a stroke for a detail of more than 50.

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	Information	Socials
	Reference Tutorials Examples Contribute Community About Start Coding Donate	Download Contact Copyright Privacy Policy Terms of Use	GitHub ↗ Instagram ↗ X ↗ YouTube ↗ Discord ↗ Forum ↗

