

createCanvas()

Creates a canvas element on the web page.

`createCanvas()` creates the main drawing canvas for a sketch. It should only be called once at the beginning of `setup()`. Calling `createCanvas()` more than once causes unpredictable behavior.

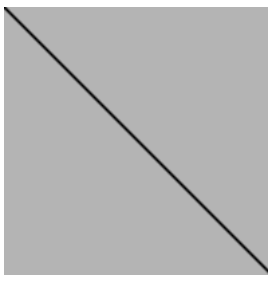
The first two parameters, `width` and `height`, are optional. They set the dimensions of the canvas and the values of the `width` and `height` system variables. For example, calling `createCanvas(900, 500)` creates a canvas that's 900×500 pixels. By default, `width` and `height` are both 100.

The third parameter is also optional. If either of the constants `P2D` or `WEBGL` is passed, as in `createCanvas(900, 500, WEBGL)`, then it will set the sketch's rendering mode. If an existing `HTMLCanvasElement` is passed, as in `createCanvas(900, 500, myCanvas)`, then it will be used by the sketch.

The fourth parameter is also optional. If an existing `HTMLCanvasElement` is passed, as in `createCanvas(900, 500, WEBGL, myCanvas)`, then it will be used by the sketch.

Note: In WebGL mode, the canvas will use a WebGL2 context if it's supported by the browser. Check the `webglVersion` system variable to check what version is being used, or call `setAttributes({ version: 1 })` to create a WebGL1 context.

Examples



▶

■

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

  background(180);

  // Draw a diagonal line.
  line(0, 0, width, height);

  describe('A diagonal line drawn from top-left to bottom-right on a gray background.');
```

▶

■

```
function setup() {
  createCanvas(100, 50);

  background(180);

  // Draw a diagonal line.
  line(0, 0, width, height);

  describe('A diagonal line drawn from top-left to bottom-right on a gray background.');
```

▶

■

```
// Use WebGL mode.

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

  background(180);

  // Draw a diagonal line.
  line(-width / 2, -height / 2, width / 2, height / 2);

  describe('A diagonal line drawn from top-left to bottom-right on a gray background.');
```

▶

■

```
function setup() {
  // Create a p5.Renderer object.
  let cnv = createCanvas(50, 50);

  // Position the canvas.
  cnv.position(10, 20);

  background(180);

  // Draw a diagonal line.
  line(0, 0, width, height);

  describe('A diagonal line drawn from top-left to bottom-right on a gray background.');
```

Syntax

```
createCanvas([width], [height], [renderer], [canvas])
```

```
createCanvas([width], [height], [canvas])
```

Parameters

<code>width</code>	Number: width of the canvas. Defaults to 100.
<code>height</code>	Number: height of the canvas. Defaults to 100.
<code>renderer</code>	Constant: either <code>P2D</code> or <code>WEBGL</code> . Defaults to <code>P2D</code> .
<code>canvas</code>	<code>HTMLCanvasElement</code> : existing canvas element that should be used for the sketch.

Returns

`p5.Renderer`: new `p5.Renderer` that holds the canvas.

This page is generated from the comments in `src/core/rendering.js`. Please feel free to edit it and submit a pull request!

Related References

createFramebuffer Creates a new <code>p5.Framebuffer</code> object with the same WebGL context as the graphics buffer.	remove Removes the graphics buffer from the web page.	reset Resets the graphics buffer's transformations and lighting.	blendMode Sets the way colors blend when added to the canvas.
--	---	--	---