

point()

Draws a single point in space.

A point's default width is one pixel. To color a point, use the `stroke()` function. To change its width, use the `strokeWeight()` function. A point can't be filled, so the `fill()` function won't affect the point's color.

The version of `point()` with two parameters allows the point's location to be set with its x- and y-coordinates, as in `point(10, 20)`.

The version of `point()` with three parameters allows the point to be drawn in 3D space with x-, y-, and z-coordinates, as in `point(10, 20, 30)`. Doing so requires adding the `WEBGL` argument to `createCanvas()`.

The version of `point()` with one parameter allows the point's location to be set with a `p5.Vector` object.

Examples

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

  background(200);

  // Top-left.
  point(30, 20);

  // Top-right.
  point(85, 20);

  // Bottom-right.
  point(85, 75);

  // Bottom-left.
  point(30, 75);

  describe(
    'Four small, black points drawn on a gray canvas. The
    points form the corners of a square.'
  );
}
```

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

  background(200);

  // Top-left.
  point(30, 20);

  // Top-right.
  point(70, 20);

  // Style the next points.
  stroke('purple');
  strokeWeight(10);

  // Bottom-right.
  point(70, 80);

  // Bottom-left.
  point(30, 80);

  describe(
    'Four points drawn on a gray canvas. Two are black and two
    are purple. The points form the corners of a square.'
  );
}
```

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

  background(200);

  // Top-left.
  let a = createVector(30, 20);
  point(a);

  // Top-right.
  let b = createVector(70, 20);
  point(b);

  // Bottom-right.
  let c = createVector(70, 80);
  point(c);

  // Bottom-left.
  let d = createVector(30, 80);
  point(d);

  describe(
    'Four small, black points drawn on a gray canvas. The
    points form the corners of a square.'
  );
}

function draw() {
  background(200);

  // Style the points.
  stroke('purple');
  strokeWeight(10);

  // Top-left.
  point(-20, -30);

  // Bottom-right.
  point(20, 30);
}
```

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

  describe('Two purple points drawn on a gray canvas. The
    scene spins slowly.');
}

function draw() {
  background(200);

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

  // Style the points.
  stroke('purple');
  strokeWeight(10);

  // Top-left.
  point(-20, -30, 0);

  // Bottom-right.
  point(20, 30, -50);
}
```

Syntax

`point(x, y, [z])`

Parameters

- x Number: the x-coordinate.
- y Number: the y-coordinate.
- z Number: the z-coordinate (for WebGL mode).
- coordinateVector `p5.Vector`: the coordinate vector.

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.



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

GitHub ↗ Instagram ↗ X ↗ YouTube ↗ Discord ↗ Forum ↗

Processing Foundation ↗