

div()

Divides a vector's `x`, `y`, and `z` components.

`div()` can use separate numbers, as in `v.div(1, 2, 3)`, another p5.Vector object, as in `v.div(v2)`, or an array of numbers, as in `v.div([1, 2, 3])`.

If only one value is provided, as in `v.div(2)`, then all the components will be divided by 2. If a value isn't provided for a component, it won't change. For example, `v.div(4, 5)` divides `v.x` by 4, `v.y` by 5, and `v.z` by 1. Calling `div()` with no arguments, as in `v.div()`, has no effect.

The static version of `div()`, as in `p5.Vector.div(v, 2)`, returns a new p5.Vector object and doesn't change the originals.

Examples

```

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

  background(200);

  // Style the points.
  strokeWeight(5);

  // Center.
  let p = createVector(50, 50);
  point(p);

  // Top-left.
  // Divide p.x / 2 and p.y / 2
  p.div(2);
  point(p);

  describe('Two black dots drawn on a gray square. One dot is
in the top left corner and the other is in the center.');
}

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

  background(200);

  // Style the points.
  strokeWeight(5);

  // Bottom-right.
  let p = createVector(50, 75);
  point(p);

  // Top-left.
  // Divide p.x / 2 and p.y / 3
  p.div(2, 3);
  point(p);

  describe('Two black dots drawn on a gray square. One dot is
in the top left corner and the other is in the bottom
center.');
}

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

  background(200);

  // Style the points.
  strokeWeight(5);

  // Bottom-right.
  let p = createVector(50, 75);
  point(p);

  // Top-left.
  // Divide p.x / 2 and p.y / 3
  let arr = [2, 3];
  p.div(arr);
  point(p);

  describe('Two black dots drawn on a gray square. One dot is
in the top left corner and the other is in the bottom
center.');
}

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

  background(200);

  // Style the points.
  strokeWeight(5);

  // Bottom-right.
  let p = createVector(50, 75);
  point(p);

  // Top-left.
  // Divide p.x / 2 and p.y / 3
  let p2 = createVector(2, 3);
  p.div(p2);
  point(p);

  describe('Two black dots drawn on a gray square. One dot is
in the top left corner and the other is in the bottom
center.');
}

▶ function draw() {
  background(200);

  let origin = createVector(0, 0);

  // Draw the red arrow.
  let v1 = createVector(50, 50);
  drawArrow(origin, v1, 'red');

  // Draw the blue arrow.
  let v2 = p5.Vector.div(v1, 2);
  drawArrow(origin, v2, 'blue');

  describe('Two arrows extending from the top
left corner. The blue arrow is half the length
of the red arrow.');
}

function drawArrow(base, vec, myColor) {
  push();
  stroke(myColor);
  strokeWeight(3);
  fill(myColor);
  translate(base.x, base.y);
  line(0, 0, vec.x, vec.y);
  rotate(vec.heading());
  let arrowSize = 7;
  translate(vec.mag() - arrowSize, 0);
}

```

Syntax

`div(n)`

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

`div(arr)`

`div(v)`

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

`div(v, n, [target])`

`div(v0, v1, [target])`

`div(v0, arr, [target])`

`n` Number: The number to divide the vector by

`x` Number: number to divide with the x component of the vector.

`y` Number: number to divide with the y component of the vector.

`z` Number: number to divide with the z component of the vector.

`arr` Number[]: array to divide the components of the vector by.

`v` p5.Vector: vector to divide the components of the original vector by.

`target` p5.Vector: The vector to receive the result

`v0` p5.Vector:

`v1` p5.Vector:

`array` Returns the vector's components as an array of numbers.

`clampToZero` Replaces the components of a p5.Vector that are very close to zero with zero.

This page is generated from the comments in `src/math/p5.Vector.js`. Please feel free to edit it and submit a pull request!

Related References

`add` Adds to a vector's x, y, and z components.

`angleBetween` Calculates the angle between two vectors.

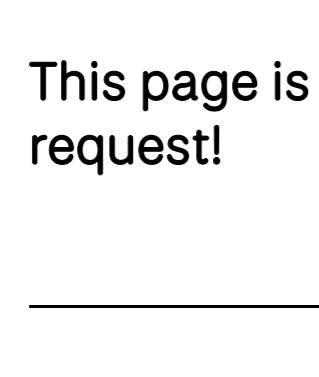
`array` Returns the vector's components as an array of numbers.

`clampToZero` Replaces the components of a p5.Vector that are very close to zero with zero.

[Reference](#) [Tutorials](#) [Examples](#) [Contribute](#) [Community](#) [About](#) [Start Coding](#) [Donate](#)

[GitHub ↗](#) [Instagram ↗](#) [X ↗](#) [YouTube ↗](#) [Discord ↗](#) [Forum ↗](#)

[Download](#) [Contact](#) [Copyright](#) [Privacy Policy](#) [Terms of Use](#)

 [Donate Today!](#) Support p5.js and the Processing Foundation.

[Donate Today!](#) Support p5.js and the Processing Foundation.