

random()

Returns a random number or a random element from an array.

`random()` follows uniform distribution, which means that all outcomes are equally likely. When `random()` is used to generate numbers, all numbers in the output range are equally likely to be returned. When `random()` is used to select elements from an array, all elements are equally likely to be chosen.

By default, `random()` produces different results each time a sketch runs. The `randomSeed()` function can be used to generate the same sequence of numbers or choices each time a sketch runs.

The version of `random()` with no parameters returns a random number from 0 up to but not including 1.

The version of `random()` with one parameter works one of two ways. If the argument passed is a number, `random()` returns a random number from 0 up to but not including the number. For example, calling `random(5)` returns values between 0 and 5. If the argument passed is an array, `random()` returns a random element from that array. For example, calling `random(['lion', 'tiger', 'bear'])` returns either a lion, tiger, or bear emoji.

The version of `random()` with two parameters returns a random number from a given range. The arguments passed set the range's lower and upper bounds. For example, calling `random(-5, 10.2)` returns values from -5 up to but not including 10.2.

Examples

```

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

  background(200);

  // Get random coordinates between 0 and 100.
  let x = random(0, 100);
  let y = random(0, 100);

  // Draw a point.
  strokeWeight(5);
  point(x, y);

  describe('A black dot appears in a random position on a gray square.');
}

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

  background(200);

  // Get random coordinates between 0 and 100.
  let x = random(100);
  let y = random(100);

  // Draw the point.
  strokeWeight(5);
  point(x, y);

  describe('A black dot appears in a random position on a gray square.');
}

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

  background(200);

  // Create an array of emoji strings.
  let animals = ['lion', 'tiger', 'bear'];

  // Choose a random element from the array.
  let choice = random(animals);

  // Style the text.
  textAlign(CENTER);
  textSize(20);

  // Display the emoji.
  text(choice, 50, 50);

  describe('An animal face is displayed at random. Either a lion, tiger, or bear.');
}

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

  // Slow the frame rate.
  frameRate(5);

  describe('A black dot moves around randomly on a gray square.');
}

function draw() {
  background(200);

  // Get random coordinates between 0 and 100.
  let x = random(100);
  let y = random(100);

  // Draw the point.
  strokeWeight(5);
  point(x, y);
}

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

  // Slow the frame rate.
  frameRate(5);

  describe('A black dot moves around randomly in the middle of a gray square.');
}

function draw() {
  background(200);

  // Get random coordinates between 45 and 55.
  let x = random(45, 55);
  let y = random(45, 55);

  // Draw the point.
  strokeWeight(5);
  point(x, y);
}

▶ ■
let x = 50;
let y = 50;

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

  background(200);

  describe('A black dot moves around randomly leaving a trail.');
}

function draw() {
  // Update x and y randomly.
  x += random(-1, 1);
  y += random(-1, 1);

  // Draw the point.
  point(x, y);
}

```

Syntax

`random([min], [max])`

Parameters

<code>min</code>	Number: lower bound (inclusive).
<code>max</code>	Number: upper bound (exclusive).
<code>choices</code>	Array: array to choose from.

Returns

Number: random number.

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

Related References

`random`

Returns a random number or a random element from an array.

`randomGaussian`

Returns a random number fitting a Gaussian, or normal, distribution.

`randomSeed`

Sets the seed value for the `random()` and `randomGaussian()` functions.

