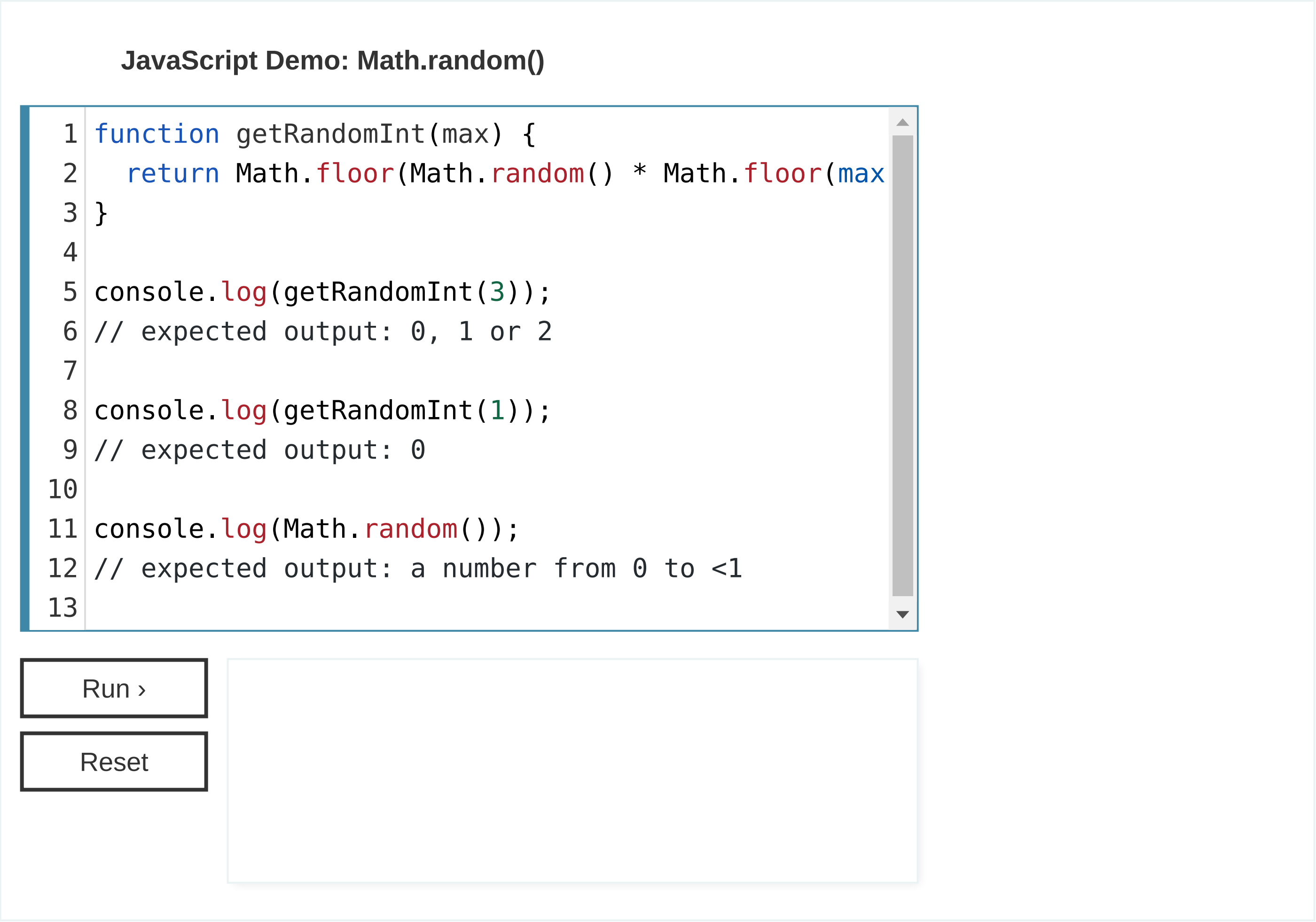
MDN web docs moz://a

Sign in

English ▾

Math.random()

The **`Math.random()`** function returns a floating-point, pseudo-random number in the range 0 to less than 1 (inclusive of 0, but not 1) with approximately uniform distribution over that range — which you can then scale to your desired range. The implementation selects the initial seed to the random number generation algorithm; it cannot be chosen or reset by the user.



JavaScript Demo: Math.random()

```
function getRandomInt(max) {
  return Math.floor(Math.random() * Math.floor(max
}

console.log(getRandomInt(3));
// expected output: 0, 1 or 2

console.log(getRandomInt(1));
// expected output: 0

console.log(Math.random());
// expected output: a number from 0 to <1

```

> `Math.random()` *does not* provide cryptographically secure random numbers. Do not use them for anything related to security. Use the Web Crypto API instead, and more precisely the `window.crypto.getRandomValues()` method.

Syntax

```
Math.random()
```

Return value

A floating-point, pseudo-random number between `0` (inclusive) and 1 (exclusive).

Examples

Note that as numbers in JavaScript are IEEE 754 floating point numbers with round-to-nearest-even behavior, the ranges claimed for the functions below (excluding the one for `Math.random()` itself) aren't exact. If extremely large bounds are chosen (2^{53} or higher), it's possible in *extremely* rare cases to calculate the usually-excluded upper bound.

Getting a random number between 0 (inclusive) and 1 (exclusive)

```
function getRandom() {
  return Math.random();
}
```

Getting a random number between two values

This example returns a random number between the specified values. The returned value is no lower than (and may possibly equal) `min`, and is less than (and not equal) `max`.

```
function getRandomArbitrary(min, max) {
  return Math.random() * (max - min) + min;
}
```

Getting a random integer between two values

This example returns a random *integer* between the specified values. The value is no lower than `min` (or the next integer greater than `min` if `min` isn't an integer), and is less than (but not equal to) `max`.

```
function getRandomInt(min, max) {
  min = Math.ceil(min);
  max = Math.floor(max);
  return Math.floor(Math.random() * (max - min) + min); //The maximum is exclusi
}
```

> It might be tempting to use `Math.round()` to accomplish that, but doing so would cause your random numbers to follow a non-uniform distribution, which may not be acceptable for your needs.

Getting a random integer between two values, inclusive

While the `getRandomInt()` function above is inclusive at the minimum, it's exclusive at the maximum. What if you need the results to be inclusive at both the minimum and the maximum? The `getRandomIntInclusive()` function below accomplishes that.

```
function getRandomIntInclusive(min, max) {
  min = Math.ceil(min);
  max = Math.floor(max);
  return Math.floor(Math.random() * (max - min + 1) + min); //The maximum is inc
}
```

Specifications

Specification

Specification
ECMAScript (ECMA-262) The definition of 'Math.random' in that specification.

Browser compatibility

Update compatibility data on GitHub

	random
Chrome	1
Edge	12
Firefox	1
IE	3
Opera	3
Safari	1
WebView Android	1
Chrome Android	18
Firefox Android	4
Opera Android	10.1
Safari iOS	1
Samsung Internet Android	1.0
nodejs	0.1.100

What are we missing?

..	Full support

See also

- `window.crypto.getRandomValues()`

Last modified: Nov 1, 2020, by MDN contributors

Related Topics

Standard built-in objects

`Math`

Properties

`Math.E`

`Math.LN10`

`Math.LN2`

`Math.LOG10E`

`Math.LOG2E`

`Math.PI`

`Math.SQRT1_2`

`Math.SQRT2`

Methods

`Math.abs()`

`Math.acos()`

`Math.acosh()`

`Math.asin()`

`Math.asinh()`

`Math.atan()`

`Math.atan2()`

`Math.atanh()`

- `Math.cbrt()`
- `Math.ceil()`
- `Math.clz32()`
- `Math.cos()`
- `Math.cosh()`
- `Math.exp()`
- `Math.expm1()`
- `Math.floor()`
- `Math.fround()`
- `Math.hypot()`
- `Math.imul()`
- `Math.log()`
- `Math.log10()`
- `Math.log1p()`
- `Math.log2()`
- `Math.max()`
- `Math.min()`
- `Math.pow()`
- *`Math.random()`*
- `Math.round()`
- `Math.sign()`
- `Math.sin()`
- `Math.sinh()`
- `Math.sqrt()`
- `Math.tan()`
- `Math.tanh()`
- `Math.trunc()`

Inheritance:

`Object`

Properties

`Object.prototype.__proto__`

`Object.prototype.constructor`

Methods

`Object.prototype.__defineGetter__()`

`Object.prototype.__defineSetter__()`

`Object.prototype.__lookupGetter__()`

`Object.prototype.__lookupSetter__()`

`Object.prototype.hasOwnProperty()`

`Object.prototype.isPrototypeOf()`

`Object.prototype.propertyIsEnumerable()`

`Object.prototype.toLocaleString()`

`Object.prototype.toSource()`

`Object.prototype.toString()`

`Object.prototype.valueOf()`

`Object.setPrototypeOf()`