



Math.min()

English ▼

The static function **Math.min()** returns the lowest-valued number passed into it, or **NaN** if any parameter isn't a number and can't be converted into one.

JavaScript Demo: Math.min()

```
console.log(Math.min(2, 3, 1));  
//2 expected output: 1  
3  
console.log(Math.min(-2, -3, -1));  
//5 expected output: -3  
6  
const array1 = [2, 3, 1];  
8  
console.log(Math.min(...array1));  
/10 expected output: 1  
11
```

Run >

Reset

Syntax

```
Math.min([value1[, value2[, ...]]])
```



Parameters

`value1, value2, ...`

Zero or more numbers among which the lowest value will be selected and returned.

Return value

The smallest of the given numbers. If any one or more of the parameters cannot be converted into a number, `NaN` is returned. The result is `Infinity` if no parameters are provided.

Description

Because `min()` is a static method of `Math`, you always use it as `Math.min()`, rather than as a method of a `Math` object you created (`Math` is not a constructor).

If no arguments are given, the result is `Infinity`.

If at least one of arguments cannot be converted to a number, the result is `NaN`.

Examples

Using `Math.min()`



This finds the min of `x` and `y` and assigns it to `z`:

```
1 | var x = 10, y = -20;  
2 | var z = Math.min(x, y);
```

Clipping a value with `Math.min()`

`Math.min()` is often used to clip a value so that it is always less than or equal to a boundary.

For instance, this



```
1 | var x = f(foo);
2 |
3 | if (x > boundary) {
4 |     x = boundary;
5 | }
```

may be written as this

```
1 | var x = Math.min(f(foo), boundary);
```

`Math.max()` can be used in a similar way to clip a value at the other end.

Specifications

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

Browser compatibility

[Update compatibility data on GitHub](#)

min	
Chrome	1
Edge	12
Firefox	1
IE	3
Opera	Yes
Safari	1
WebView Android	1
Chrome Android	18
Firefox Android	4

Opera Android	Yes
Safari iOS	1
Samsung Internet Android	1.0
nodejs	Yes

What are we missing?

Full support

See also

- `Math.max()`

 Last modified: Jan 15, 2020, by MDN contributors

Syntax
Description
Examples
Specifications
Browser compatibility
See also

Related Topics

Standard built-in objects

Math

Properties

- `Math.E`
- `Math.LN10`
- `Math.LN2`
- `Math.LOG10E`
- `Math.LOG2E`

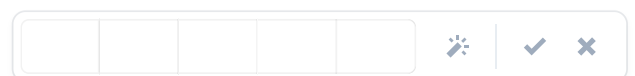






`Math.PI``Math.SQRT1_2``Math.SQRT2`

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.__count__  
⚠️🗑️ Object.prototype.__noSuchMethod__  
⚠️🗑️ Object.prototype.__parent__  
🗨️ 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.unwatch()  
  
Object.prototype.valueOf()  
  
🗨️🗑️ Object.prototype.watch()  
  
Object.setPrototypeOf()
```



Learn the best of web development.

Get the latest and greatest from MDN delivered straight to your inbox.

you@example.com

Sign up now