

[Technologies ▼](#)[References & Guides ▼](#)[Feedback ▼](#)

Object.values()

[English ▼](#)

The **Object.values()** method returns an array of a given object's own enumerable property values, in the same order as that provided by a `for...in` loop (the difference being that a `for-in` loop enumerates properties in the prototype chain as well).

JavaScript Demo: Object.values()

```
1 const object1 = {  
2   a: 'somestring',  
3   b: 42,  
4   c: false  
5 };  
6  
7 console.log(Object.keys(object1));  
8 // expected output: Array ["somestring", 42, false]  
9
```

Run ›Reset

> Array ["a", "b", "c"]

Syntax

`Object.values(obj)`

Parameters

obj

The object whose enumerable own property values are to be returned.

Return value

An array containing the given object's own enumerable property values.

Description

`Object.values()` returns an array whose elements are the enumerable property values found on the object. The ordering of the properties is the same as that given by looping over the property values of the object manually.

Examples



```
1  var obj = { foo: 'bar', baz: 42 };
2  console.log(Object.values(obj)); // ['bar', 42]
3
4  // array like object
```

```
5 var obj = { 0: 'a', 1: 'b', 2: 'c' };
6 console.log(Object.values(obj)); // ['a', 'b', 'c']
7
8 // array like object with random key ordering
9 // when we use numeric keys, the value returned in a numerical order according to the keys
10 var an_obj = { 100: 'a', 2: 'b', 7: 'c' };
11 console.log(Object.values(an_obj)); // ['b', 'c', 'a']
12
13 // getFoo is property which isn't enumerable
14 var my_obj = Object.create({}, { getFoo: { value: function() { return this.foo; } } });
15 my_obj.foo = 'bar';
16 console.log(Object.values(my_obj)); // ['bar']
17
18 // non-object argument will be coerced to an object
19 console.log(Object.values('foo')); // ['f', 'o', 'o']
```

Polyfill

To add compatible `Object.values` support in older environments that do not natively support it, you can find a Polyfill in the [tc39/proposal-object-values-entries](https://github.com/tc39/proposal-object-values-entries) or in the [es-shims/Object.values](https://github.com/es-shims/Object.values) repositories.

Specifications

Specification	Status	Comment
ECMAScript Latest Draft (ECMA-262) The definition of 'Object.values' in that specification.	 D Draft	
ECMAScript 2017 (ECMA-262) The definition of 'Object.values' in that specification.	 ST Standard	Initial definition.

Browser compatibility

[Update compatibility data on GitHub](#)

values	
Chrome	54
Edge	14
Firefox	47
IE	No
Opera	41
Safari	10.1
WebView Android	54
Chrome Android	54
Firefox Android	47

Opera Android	41
Safari iOS	10.3
Samsung Internet Android	6.0
nodejs	7.0.0

What are we missing?



Full support



No support

User must explicitly enable this feature.

See also

- Enumerability and ownership of properties
- `Object.keys()`
- `Object.entries()`
- `Object.prototype.propertyIsEnumerable()`
- `Object.create()`
- `Object.getOwnPropertyNames()`

🕒 Mar 18, 2019, by MDN contributors

Syntax

Description

Examples

Polyfill

Specifications

Browser compatibility

See also

Related Topics

Standard built-in objects

Object

Properties

`Object.prototype`

`Object.prototype.__count__`

`Object.prototype.__noSuchMethod__`

`Object.prototype.__parent__`

`Object.prototype.__proto__`

`Object.prototype.constructor`

Methods

`Object.assign()`
`Object.create()`
`Object.defineProperties()`
`Object.defineProperty()`
`Object.entries()`
`Object.freeze()`
`Object.fromEntries()`
`Object.getNotifier()`
`Object.getOwnPropertyDescriptor()`
`Object.getOwnPropertyDescriptors()`
`Object.getOwnPropertyNames()`
`Object.getOwnPropertySymbols()`
`Object.getPrototypeOf()`
`Object.is()`
`Object.isExtensible()`
`Object.isFrozen()`
`Object.isSealed()`
`Object.keys()`
`Object.observe()`
`Object.preventExtensions()`
`Object.prototype.__defineGetter__()`
`Object.prototype.__defineSetter__()`
`Object.prototype.__lookupGetter__()`
`Object.prototype.__lookupSetter__()`

`Object.prototype.eval()`

`Object.prototype.hasOwnProperty()`

`Object.prototype.isPrototypeOf()`

`Object.prototype.propertyIsEnumerable()`

`Object.prototype.toLocaleString()`

`Object.prototype.toString()`

`Object.prototype.toString()`

`Object.prototype.unwatch()`

`Object.prototype.valueOf()`

`Object.prototype.watch()`

`Object.seal()`

`Object.setPrototypeOf()`

`Object.unobserve()`

`Object.values()`

Inheritance:

Function

Properties

`Function.arguments`

`Function.arity`

`Function.caller`

`Function.displayName`

`Function.length``Function.name``Function.prototype`

Methods

`Function.prototype.apply()``Function.prototype.bind()``Function.prototype.call()``Function.prototype.isGenerator()``Function.prototype.toSource()``Function.prototype.toString()`

Learn the best of web development

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

Sign up now

