

# Object.values()

**Object.values()** 方法返回一个给定对象自己的所有可枚举属性值的数组，值的顺序与使用 **for...in** 循环的顺序相同 ( 区别在于 **for-in** 循环枚举原型链中的属性 )。

## 语法

```
Object.values(obj)
```

## 参数

**obj**

被返回可枚举属性值的对象。

## 返回值

一个包含对象自身的所有可枚举属性键值的数组。

## 描述

**Object.values()** 返回一个数组，其元素是在对象上找到的枚举属性值。属性的顺序与通过手动循环对象的属性值所给出的顺序相同。

## 示例

```
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  var an_obj = { 100: 'a', 2: 'b', 7: 'c' };
10 console.log(Object.values(an_obj)); // ['b', 'c', 'a']
11
12 // getFoo is property which isn't enumerable
```

```
13 var my_obj = Object.create({}, { getFoo: { value: function() { return thi
14 my_obj.foo = "bar";
15 console.log(Object.values(my_obj)); // ['bar']
16
17 // non-object argument will be coerced to an object
18 console.log(Object.values("foo")); // ['f', 'o', 'o']
```

## Polyfill

如果要 `Object.values` 兼容不支持它的旧环境，可在 [tc39/proposal-object-values-entries](#) 或 [es-shims/Object.values](#) 中找到 Polyfill。


## 规范

Specification	Status	Comment
<a href="#">ECMAScript Latest Draft (ECMA-262)</a> Object.values	<b>LS</b> Living Standard	Initial definition.
<a href="#">ECMAScript 2017 (ECMA-262)</a> Object.values	<b>ST</b> Standard	

## 浏览器兼容

	Desktop						Mobile
Feature	Chrome	Edge	Firefox	Internet Explorer	Opera	Safari	
Basic Support	54	(Yes)	47	No	(Yes)	10.1	

## 相关链接

- [Enumerability and ownership of properties](#)
- [Object.keys\(\)](#)
- [Object.entries\(\)](#) 
- [Object.prototype.propertyIsEnumerable\(\)](#)
- [Object.create\(\)](#)
- [Object.getOwnPropertyNames\(\)](#)

