

Object.entries()

Object.entries() 方法返回一个给定对象自身可枚举属性的键值对数组，其排列与使用 **for...in** 循环遍历该对象时返回的顺序一致（区别在于 **for-in** 循环也枚举原型链中的属性）。

语法

```
Object.entries(obj)
```

参数

obj

可以返回其可枚举属性的键值对的对象。

返回值

给定对象自身可枚举属性的键值对数组。

描述

Object.entries() 返回一个数组，其元素是与直接在 **object** 上找到的可枚举属性键值对相对应的数组。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  const obj = { foo: 'bar', baz: 42 };
2  console.log(Object.entries(obj)); // [ ['foo', 'bar'], ['baz', 42] ]
3
4  // array like object
5  const obj = { 0: 'a', 1: 'b', 2: 'c' };
6  console.log(Object.entries(obj)); // [ ['0', 'a'], ['1', 'b'], ['2', 'c'] ]
7
8  // array like object with random key ordering
9  const anObj = { 100: 'a', 2: 'b', 7: 'c' };
10 console.log(Object.entries(anObj)); // [ ['2', 'b'], ['7', 'c'], ['100',
11
12 // getFoo is property which isn't enumerable
```

```
13  const myObj = Object.create({}, { getFoo: { value() { return this.foo; }
14  myObj.foo = 'bar';
15  console.log(Object.entries(myObj)); // [ ['foo', 'bar'] ]
16
17  // non-object argument will be coerced to an object
18  console.log(Object.entries('foo')); // [ ['0', 'f'], ['1', 'o'], ['2', 'o'] ]
19
20  // iterate through key-value gracefully
21  const obj = { a: 5, b: 7, c: 9 };
22  for (const [key, value] of Object.entries(obj)) {
23    console.log(`${key} ${value}`); // "a 5", "b 7", "c 9"
24  }
25
26  // Or, using array extras
27  Object.entries(obj).forEach(([key, value]) => {
28    console.log(`${key} ${value}`); // "a 5", "b 7", "c 9"
29  });
```

将 Object 转换为 Map

`new Map()` 构造函数接受一个可迭代的 `entries`。借助 `Object.entries` 方法你可以很容易的将 `Object` 转换为 `Map`：

```
1  var obj = { foo: "bar", baz: 42 };
2  var map = new Map(Object.entries(obj));
3  console.log(map); // Map { foo: "bar", baz: 42 }
```

Polyfill

要在较旧环境中添加兼容的 `Object.entries` 支持，你可以在 [tc39/proposal-object-values-entries](https://github.com/tc39/proposal-object-values-entries) 中找到 `Object.entries` 的示例（如果你不需要任何对 IE 的支持），在 [es-shims/Object.entries](https://github.com/es-shims/Object.entries) 资料库中的一个 polyfill，或者你可以使用下面列出的简易 polyfill。

```
if (!Object.entries)
  Object.entries = function( obj ){
    var ownProps = Object.keys( obj ),
        i = ownProps.length,
        resArray = new Array(i); // preallocate the Array
    while (i--){
      resArray[i] = [ownProps[i], obj[ownProps[i]]];
    }
  }
```

```
    return resArray;
  };
}
```

对于上述 polyfill 代码片段，如果你需要 IE <9 的支持，那么你还需要一个 `Object.keys` polyfill（如 `Object.keys` 页面上的）。

规范

Specification	Status	Comment
ECMAScript Latest Draft (ECMA-262) Object.entries	<div><div>LS</div>Living Standard</div>	Initial definition.
ECMAScript 2017 (ECMA-262) Object.entries	<div><div>ST</div>Standard</div>	

浏览器兼容

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

相关链接

- 属性的可枚举性和所有权
- `Object.keys()`
- `Object.values()` 
- `Object.prototype.propertyIsEnumerable()`
- `Object.create()`
- `Object.getOwnPropertyNames()`

