

Array.prototype.some()

概述

some() 方法测试数组中的某些元素是否通过由提供的函数实现的测试。

```
const isBiggerThan10 = (element, index, array) => {  
  return element > 10;  
}  
  
[2, 5, 8, 1, 4].some(isBiggerThan10);  
// false  
  
[12, 5, 8, 1, 4].some(isBiggerThan10);  
// true
```

语法

```
arr.some(callback[, thisArg])
```

参数

callback

用来测试每个元素的函数。

thisArg

执行 callback 时使用的 **this** 值。

描述

some 为数组中的每一个元素执行一次 **callback** 函数，直到找到一个使得 **callback** 返回一个“真值”（即可转换为布尔值 **true** 的值）。如果找到了这样一个值，**some** 将会立即返回 **true**。否则，**some** 返回 **false**。**callback** 只会在那些“有值”的索引上被调用，不会在那些被删除或从未被赋值的索引上调用。

callback 被调用时传入三个参数：元素的值，元素的索引，被遍历的数组。

如果为 `some` 提供了一个 `thisArg` 参数，将会把它传给被调用的 `callback`，作为 `this` 值。否则，在非严格模式下将会是全局对象，严格模式下是 `undefined`。

`some` 被调用时不会改变数组。

`some` 遍历的元素的范围在第一次调用 `callback` 时就已经确定了。在调用 `some` 后被添加到数组中的值不会被 `callback` 访问到。如果数组中存在且还未被访问到的元素被 `callback` 改变了，则其传递给 `callback` 的值是 `some` 访问到它那一刻的值。

示例

例子：测试数组元素的值

下面的例子检测在数组中是否有元素大于 10。

```
1 function isBigEnough(element, index, array) {
2   return (element >= 10);
3 }
4 var passed = [2, 5, 8, 1, 4].some(isBigEnough);
5 // passed is false
6 passed = [12, 5, 8, 1, 4].some(isBigEnough);
7 // passed is true
```

兼容旧环境 (Polyfill)

在第 5 版时，`some` 被添加进 ECMA-262 标准；这样导致某些实现环境可能不支持它。你可以把下面的代码插入到脚本的开头来解决此问题，从而允许在那些没有原生支持它的实现环境中使用它。该算法是 ECMA-262 第 5 版中指定的算法，假定 `Object` 和 `TypeError` 拥有他们的初始值，且 `fun.call` 等价于 `Function.prototype.call`。

```
1 if (!Array.prototype.some)
2 {
3   Array.prototype.some = function(fun /*, thisArg */)
4   {
5     'use strict';
6
7     if (this === void 0 || this === null)
8       throw new TypeError();
9
10    var t = Object(this);
11    var len = t.length >>> 0;
```

```
12     if (typeof fun !== 'function')
13         throw new TypeError();
14
15     var thisArg = arguments.length >= 2 ? arguments[1] : void 0;
16     for (var i = 0; i < len; i++)
17     {
18         if (i in t && fun.call(thisArg, t[i], i, t))
19             return true;
20     }
21
22     return false;
23 };
24 }
```

规范

Specification	Status	Comment
ECMAScript 5.1 (ECMA-262) Array.prototype.some	<div>ST</div> Standard	Initial definition. Implemented in JavaScript 1.6
ECMAScript 2015 (6th Edition, ECMA-262) Array.prototype.some	<div>ST</div> Standard	

浏览器兼容性

	Desktop	Mobile			
Feature	Chrome	Firefox (Gecko)	Internet Explorer	Opera	Safari
Basic support	(Yes)	1.5 (1.8)	9	(Yes)	(Yes)

相关链接

- Array.prototype.forEach()
- Array.prototype.every()

