

Array.prototype.indexOf()

indexOf() 方法返回在数组中可以找到一个给定元素的第一个索引，如果不存在，则返回-1。

📌 注意：对于String方法，请参阅 [String.prototype.indexOf\(\)](#)。

```
1 let a = [2, 9, 7, 8, 9];
2 a.indexOf(2); // 0
3 a.indexOf(6); // -1
4 a.indexOf(7); // 2
5 a.indexOf(8); // 3
6 a.indexOf(9); // 1
7
8 if (a.indexOf(3) === -1) {
9     // element doesn't exist in array
10 }
```

语法

```
arr.indexOf(searchElement)
arr.indexOf(searchElement[, fromIndex = 0])
```

参数

searchElement

要查找的元素

fromIndex

开始查找的位置。如果该索引值大于或等于数组长度，意味着不会在数组里查找，返回-1。如果参数中提供的索引值是一个负值，则将其作为数组末尾的一个抵消，即-1表示从最后一个元素开始查找，-2表示从倒数第二个元素开始查找，以此类推。注意：如果参数中提供的索引值是一个负值，仍然从前向后查询数组。如果抵消后的索引值仍小于0，则整个数组都将会被查询。其默认值为0。

返回值

首个被找到的元素在数组中的索引位置; 若没有找到则返回 -1

描述

`indexOf` 使用`strict equality` (无论是 `===`, 还是 `triple-equals`操作符都基于同样的方法)进行判断 `searchElement`与 数组中包含的元素之间的关系。

示例

使用indexOf

以下例子使用 `indexOf`方法确定多个值在数组中的位置。

```
1 | var array = [2, 5, 9];
2 | array.indexOf(2);      // 0
3 | array.indexOf(7);      // -1
4 | array.indexOf(9, 2);   // 2
5 | array.indexOf(2, -1);  // -1
6 | array.indexOf(2, -3);  // 0
```

找出指定元素出现的所有位置

```
1 | var indices = [];
2 | var array = ['a', 'b', 'a', 'c', 'a', 'd'];
3 | var element = 'a';
4 | var idx = array.indexOf(element);
5 | while (idx !== -1) {
6 |     indices.push(idx);
7 |     idx = array.indexOf(element, idx + 1);
8 | }
9 | console.log(indices);
10| // [0, 2, 4]
```

判断一个元素是否在数组里，不在则更新数组

```
1 | function updateVegetablesCollection (veggies, veggie) {
2 |     if (veggies.indexOf(veggie) === -1) {
3 |         veggies.push(veggie);
4 |         console.log('New veggies collection is : ' + veggies);
5 |     } else if (veggies.indexOf(veggie) > -1) {
6 |         console.log(veggie + ' already exists in the veggies collection.')
7 |     }
8 | }
```

```
8   }
9
10  var veggies = ['potato', 'tomato', 'chillies', 'green-pepper'];
11
12  // New veggies collection is : potato,tomato,chillies,green-papper,spinach
13  updateVegetablesCollection(veggies, 'spinach');
14  // spinach already exists in the veggies collection.
15  updateVegetablesCollection(veggies, 'spinach');
```

Polyfill

`indexOf` 在 ECMA-262 标准的第5版中被加入，但并非所有的浏览器都支持该方法。你可以在编写 scripts 时，在其开头使用以下代码，它能够允许你在没有本地支持的情况下使用 `indexOf` 方法。该算法符合 ECMA-262 第5版其中一项规定，即假定 `TypeError` 和 `Math.abs` 呈现它们原有的价值。

```
1  // Production steps of ECMA-262, Edition 5, 15.4.4.14
2  // Reference: http://es5.github.io/#x15.4.4.14
3  if (!Array.prototype.indexOf) {
4      Array.prototype.indexOf = function(searchElement, fromIndex) {
5
6          var k;
7
8          // 1. Let 0 be the result of calling ToObject passing
9          //    the this value as the argument.
10         if (this == null) {
11             throw new TypeError('"this" is null or not defined');
12         }
13
14         var 0 = Object(this);
15
16         // 2. Let lenValue be the result of calling the Get
17         //    internal method of 0 with the argument "length".
18         // 3. Let len be ToUint32(lenValue).
19         var len = 0.length >>> 0;
20
21         // 4. If len is 0, return -1.
22         if (len === 0) {
23             return -1;
24         }
25
26         // 5. If argument fromIndex was passed let n be
```

```
27 //   ToInteger(fromIndex); else let n be 0.
28 var n = +fromIndex || 0;
29
30 if (Math.abs(n) === Infinity) {
31   n = 0;
32 }
33
34 // 6. If n >= len, return -1.
35 if (n >= len) {
36   return -1;
37 }
38
39 // 7. If n >= 0, then Let k be n.
40 // 8. Else, n<0, Let k be len - abs(n).
41 //   If k is less than 0, then let k be 0.
42 k = Math.max(n >= 0 ? n : len - Math.abs(n), 0);
43
44 // 9. Repeat, while k < len
45 while (k < len) {
46   // a. Let Pk be ToString(k).
47   //   This is implicit for LHS operands of the in operator
48   // b. Let kPresent be the result of calling the
49   //   HasProperty internal method of O with argument Pk.
50   //   This step can be combined with c
51   // c. If kPresent is true, then
52   //   i. Let elementK be the result of calling the Get
53   //      internal method of O with the argument ToString(k).
54   //   ii. Let same be the result of applying the
55   //       Strict Equality Comparison Algorithm to
56   //       searchElement and elementK.
57   //   iii. If same is true, return k.
58   if (k in O && O[k] === searchElement) {
59     return k;
60   }
61   k++;
62 }
63 return -1;
64 };
65 }
```

规范

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

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.lastIndexOf\(\)](#)
- [TypedArray.prototype.indexOf\(\)](#)

