Array.prototype.lastIndexOf()

概述

`lastIndexOf()` 方法返回指定元素（也即有效的 JavaScript 值或变量）在数组中的最后一个的索引，如果不存在则返回 -1。从数组的后面向前查找，从 `fromIndex` 处开始。

语法

```
arr.lastIndexOf(searchElement[, fromIndex = arr.length - 1])
```

参数

`searchElement`
被查找的元素。

`fromIndex`
从此位置开始逆向查找。默认为数组的长度减 1，即整个数组都被查找。如果该值大于或等于数组的长度，则整个数组会被查找。如果为负值，将其视为从数组末尾向前的偏移。即使该值为负，数组仍然会被从后向前查找。如果该值为负时，其绝对值大于数组长度，则方法返回 -1，即数组不会被查找。

描述

`lastIndexOf` 使用严格相等（strict equality，即 `===`）比较 `searchElement` 和数组中的元素。

示例

例子：使用 lastIndexOf

下例使用 `lastIndexOf` 定位数组中的值。

```
var array = [2, 5, 9, 2];
var index = array.lastIndexOf(2);
// index is 3
index = array.lastIndexOf(7);
```

```
// index is -1
index = array.lastIndexOf(2, 3);
// index is 3
index = array.lastIndexOf(2, 2);
// index is 0
index = array.lastIndexOf(2, -2);
// index is 0
index = array.lastIndexOf(2, -1);
// index is 3
```

例子：查找所有元素

下例使用 `lastIndexOf` 查找到一个元素在数组中所有的索引（下标），并使用 `push` 将所有添加到另一个数组中。

```
var indices = [];
var array = ['a', 'b', 'a', 'c', 'a', 'd'];
var element = 'a';
var idx = array.lastIndexOf(element);

while (idx != -1) {
  indices.push(idx);
  idx = (idx > 0 ? array.lastIndexOf(element, idx - 1) : -1);
}

console.log(indices);
// [4, 2, 0];
```

Note that we have to handle the case `idx == 0` separately here because the element will always be found regardless of the `fromIndex` parameter if it is the first element of the array. This is different from the `indexOf` method.

注意，必须单独处理 `idx == 0` 时的情况，因为如果元素是数组中的第一个元素，则总会被查找，忽略了 `fromIndex` 参数。这点和 `indexOf` 方法不同。（译注：个人觉得这句话解释有问题，`idx == 0` 时，`array.lastIndexOf(element, idx - 1)` 会从最后一个元素向前查找，这样就重复查找，且死循环了，所以要做一个判断，而且已经查找到第一个元素了，就该结束了）。

兼容旧环境（Polyfill）

`lastIndexOf` 在 ECMA-262 标准第 5 版被添加。因此它在不兼容该标准的浏览器中可能不被支持。你可以把下面代码添加到脚本中来使那些没有实现该方法的实现环境支持该方法。该算法是被 ECMA-

262 第 5 版指定的。假定 `Object` 、 `TypeError` 、 `Number` 、 `Math.floor` 、 `Math.abs` ，以及 `Math.min` 拥有其初始值。

```
if (!Array.prototype.lastIndexOf) {
  Array.prototype.lastIndexOf = function(searchElement /*, fromIndex*/) {
    'use strict';

    if (this === void 0 || this === null) {
      throw new TypeError();
    }

    var n, k,
        t = Object(this),
        len = t.length >>> 0;
    if (len === 0) {
      return -1;
    }

    n = len - 1;
    if (arguments.length > 1) {
      n = Number(arguments[1]);
      if (n != n) {
        n = 0;
      }
      else if (n != 0 && n != (1 / 0) && n != -(1 / 0)) {
        n = (n > 0 || -1) * Math.floor(Math.abs(n));
      }
    }

    for (k = n >= 0
          ? Math.min(n, len - 1)
          : len - Math.abs(n); k >= 0; k--) {
      if (k in t && t[k] === searchElement) {
        return k;
      }
    }
    return -1;
  };
}
```

另外，该实现是为了绝对兼容 Firefox 和 the SpiderMonkey JavaScript 引擎中的 `lastIndexOf`，包括了几种临界情况。如果你要在实际应用中使用该实现，可以忽略这些临界情况，从而简化 `fromIndex` 的计算。

规范

规范版本	规范状态	备注
ECMAScript 5.1 (ECMA-262) Array.prototype.lastIndexOf	ST Standard	JavaScript 1.6 中首次定义
ECMAScript 2015 (6th Edition, ECMA-262) Array.prototype.lastIndexOf	ST Standard	

浏览器兼容性

Desktop Mobile

Feature	Chrome	Firefox (Gecko)	Internet Explorer	Opera	Safari
Basic support	(Yes)	(Yes)	9	(Yes)	(Yes)

相关链接

- `Array.prototype.indexOf()`
- `TypedArray.prototype.lastIndexOf()`