Array.prototype.indexOf()

The **`indexOf()`** method returns the first index at which a given element can be found in the array, or -1 if it is not present.

Syntax

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

Parameters

`searchElement`
Element to locate in the array.

`fromIndex`
The index to start the search at. If the index is greater than or equal to the array's length, -1 is returned, which means the array will not be searched. If the provided index value is a negative number, it is taken as the offset from the end of the array. Note: if the provided index is negative, the array is still searched from front to back. If the calculated index is less than 0, then the whole array will be searched. Default: 0 (entire array is searched).

Description

`indexOf()` compares `searchElement` to elements of the Array using strict equality (the same method used by the ===, or triple-equals, operator).

Examples

Using `indexOf()`

The following example uses `indexOf()` to locate values in an array.

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

Finding all the occurrences of an element

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

Finding if an element exists in the array or not and updating the array

```
function updateVegetablesCollection (veggies, veggie) {
    if (veggies.indexOf(veggie) === -1) {
        veggies.push(veggie);
        console.log('New veggies collection is : ' + veggies);
    } else if (veggies.indexOf(veggie) > -1) {
```

```
        console.log(veggie + ' already exists in the veggies collection.');
    }
}

var veggies = ['potato', 'tomato', 'chillies', 'green-pepper'];

updateVegetablesCollection(veggies, 'spinach'); // New veggies collection is : potato,tomato,chillies,g
updateVegetablesCollection(veggies, 'spinach'); // spinach already exists in the veggies collection.
```

Polyfill

`indexOf()` was added to the ECMA-262 standard in the 5th edition; as such it may not be present in all browsers. You can work around this by utilizing the following code at the beginning of your scripts. This will allow you to use `indexOf()` when there is still no native support. This algorithm matches the one specified in ECMA-262, 5th edition, assuming `TypeError` and `Math.abs()` have their original values.

```
// Production steps of ECMA-262, Edition 5, 15.4.4.14
// Reference: http://es5.github.io/#x15.4.4.14
if (!Array.prototype.indexOf) {
  Array.prototype.indexOf = function(searchElement, fromIndex) {

    var k;

    // 1. Let O be the result of calling ToObject passing
    //    the this value as the argument.
    if (this == null) {
      throw new TypeError('"this" is null or not defined');
    }

    var O = Object(this);

    // 2. Let lenValue be the result of calling the Get
    //    internal method of O with the argument "length".
    // 3. Let len be ToUint32(lenValue).
    var len = O.length >>> 0;

    // 4. If len is 0, return -1.
    if (len === 0) {
      return -1;
    }

    // 5. If argument fromIndex was passed let n be
    //    ToInteger(fromIndex); else let n be 0.
    var n = +fromIndex || 0;

    if (Math.abs(n) === Infinity) {
      n = 0;
    }

    // 6. If n >= len, return -1.
    if (n >= len) {
      return -1;
    }

    // 7. If n >= 0, then Let k be n.
    // 8. Else, n<0, Let k be len - abs(n).
    //    If k is less than 0, then let k be 0.
    k = Math.max(n >= 0 ? n : len - Math.abs(n), 0);

    // 9. Repeat, while k < len
    while (k < len) {
      // a. Let Pk be ToString(k).
      //   This is implicit for LHS operands of the in operator
      // b. Let kPresent be the result of calling the
      //    HasProperty internal method of O with argument Pk.
      //   This step can be combined with c
      // c. If kPresent is true, then
      //    i.  Let elementK be the result of calling the Get
```

```
      //        internal method of O with the argument ToString(k).
      //   ii.  Let same be the result of applying the
      //        Strict Equality Comparison Algorithm to
      //        searchElement and elementK.
      //  iii.  If same is true, return k.
      if (k in O && O[k] === searchElement) {
        return k;
      }
      k++;
    }
    return -1;
  };
}
```

Specifications

Specification	Status	Comment
ECMAScript 5.1 (ECMA-262) The definition of 'Array.prototype.indexOf' in that specification.	ST Standard	Initial definition. Implemented in JavaScript 1.6.
ECMAScript 2015 (6th Edition, ECMA-262) The definition of 'Array.prototype.indexOf' in that specification.	ST Standard	

Browser compatibility

Desktop Mobile

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

See also

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