Array.prototype.splice()

The **`splice()`** method changes the content of an array by removing existing elements and/or adding new elements.

Syntax

```
array.splice(start, deleteCount[, item1[, item2[, ...]]])
```

Parameters

`start`

Index at which to start changing the array. If greater than the length of the array, actual starting index will be set to the length of the array. If negative, will begin that many elements from the end.

`deleteCount`

An integer indicating the number of old array elements to remove. If `deleteCount` is 0, no elements are removed. In this case, you should specify at least one new element. If `deleteCount` is greater than the number of elements left in the array starting at `start`, then all of the elements through the end of the array will be deleted.

`itemN`

The element to add to the array. If you don't specify any elements, `splice()` will only remove elements from the array.

Returns

An array containing the deleted elements. If only one element is removed, an array of one element is returned. If no elements are removed, an empty array is returned.

Description

If you specify a different number of elements to insert than the number you're removing, the array will have a different length at the end of the call.

Examples

Using `splice()`

The following script illustrates the use of `splice()`:

```
var myFish = ['angel', 'clown', 'mandarin', 'surgeon'];

// removes 0 elements from index 2, and inserts 'drum'
var removed = myFish.splice(2, 0, 'drum');
// myFish is ['angel', 'clown', 'drum', 'mandarin', 'surgeon']
// removed is [], no elements removed

// removes 1 element from index 3
removed = myFish.splice(3, 1);
// myFish is ['angel', 'clown', 'drum', 'surgeon']
// removed is ['mandarin']

// removes 1 element from index 2, and inserts 'trumpet'
removed = myFish.splice(2, 1, 'trumpet');
// myFish is ['angel', 'clown', 'trumpet', 'surgeon']
// removed is ['drum']

// removes 2 elements from index 0, and inserts 'parrot', 'anemone' and 'blue'
removed = myFish.splice(0, 2, 'parrot', 'anemone', 'blue');
// myFish is ['parrot', 'anemone', 'blue', 'trumpet', 'surgeon']
// removed is ['angel', 'clown']
```

```
22
23 // removes 2 elements from index 3
24 removed = myFish.splice(3, myFish.length);
25 // myFish is ['parrot', 'anemone', 'blue']
26 // removed is ['trumpet', 'surgeon']
```

Specifications

Specification	Status	Comment
ECMAScript 3rd Edition (ECMA-262)	Standard	Initial definition. Implemented in JavaScript 1.2.
ECMAScript 5.1 (ECMA-262) The definition of 'Array.prototype.splice' in that specification.	Standard	
ECMAScript 2015 (6th Edition, ECMA-262) The definition of 'Array.prototype.splice' in that specification.	Standard	

Browser compatibility

Desktop Mobile

Feature	Chrome	Firefox (Gecko)	Internet Explorer	Opera	Safari
Basic support	1.0	1.0 (1.7 or earlier)	5.5	(Yes)	(Yes)

Backward compatibility

In JavaScript 1.2 the `splice()` method returns the element removed, if only one element is removed (`deleteCount` parameter is 1); otherwise, the method returns an array containing the removed elements.

> **Note:** The last browser to use JavaScript 1.2 was Netscape Navigator 4, so you can depend on `splice()` always returning an array. This is the case, when a JavaScript object has a `length` property and a `splice()` method, `console.log()` treats it as an Array-like object. Checking with an `instanceof Array` on it returns `false`.

See also

- `push()` / `pop()` — add/remove elements from the end of the array
- `unshift()` / `shift()` — add/remove elements from the beginning of the array
- `concat()` — returns a new array comprised of this array joined with other array(s) and/or value(s)