String.prototype.split()

Share: Twitter Facebook Google+

The **`split()`** method splits a `String` object into an array of strings by separating the string into substrings.

Syntax

```
str.split([separator[, limit]])
```

Parameters

`separator`

Optional. Specifies the character(s) to use for separating the string. The `separator` is treated as a string or a regular expression. If `separator` is omitted, the array returned contains one element consisting of the entire string. If `separator` is an empty string, `str` is converted to an array of characters.

`limit`

Optional. Integer specifying a limit on the number of splits to be found. The `split()` method still splits on every match of `separator`, but it truncates the returned array to at most `limit` elements.

Description

The `split()` method returns the new array.

When found, `separator` is removed from the string and the substrings are returned in an array. If `separator` is not found or is omitted, the array contains one element consisting of the entire string. If `separator` is an empty string, `str` is converted to an array of characters.

If `separator` is a regular expression that contains capturing parentheses, then each time `separator` is matched, the results (including any undefined results) of the capturing parentheses are spliced into the output array. However, not all browsers support this capability.

> **Note:** When the string is empty, `split()` returns an array containing one empty string, rather than an empty array. If the string and separator are both empty strings, an empty array is returned.

Examples

Using `split()`

The following example defines a function that splits a string into an array of strings using the specified separator. After splitting the string, the function displays messages indicating the original string (before the split), the separator used, the number of elements in the array, and the individual array elements.

```
function splitString(stringToSplit, separator) {
  var arrayOfStrings = stringToSplit.split(separator);

  console.log('The original string is: "' + stringToSplit + '"');
  console.log('The separator is: "' + separator + '"');
  console.log('The array has ' + arrayOfStrings.length + ' elements: ' + arrayOfStrings.join(' / '));
}

var tempestString = 'Oh brave new world that has such people in it.';
var monthString = 'Jan,Feb,Mar,Apr,May,Jun,Jul,Aug,Sep,Oct,Nov,Dec';

var space = ' ';
var comma = ',';

splitString(tempestString, space);
splitString(tempestString);
splitString(monthString, comma);
```

This example produces the following output:

```
The original string is: "Oh brave new world that has such people in it."
The separator is: " "
The array has 10 elements: Oh / brave / new / world / that / has / such / people / in / it.

The original string is: "Oh brave new world that has such people in it."
The separator is: "undefined"
The array has 1 elements: Oh brave new world that has such people in it.

The original string is: "Jan,Feb,Mar,Apr,May,Jun,Jul,Aug,Sep,Oct,Nov,Dec"
The separator is: ","
The array has 12 elements: Jan / Feb / Mar / Apr / May / Jun / Jul / Aug / Sep / Oct / Nov / Dec
```

Removing spaces from a string

In the following example, `split()` looks for 0 or more spaces followed by a semicolon followed by 0 or more spaces and, when found, removes the spaces from the string. `nameList` is the array returned as a result of `split()`.

```
var names = 'Harry Trump ;Fred Barney; Helen Rigby ; Bill Abel ;Chris Hand ';

console.log(names);

var re = /\s*;\s*/;
var nameList = names.split(re);

console.log(nameList);
```

This logs two lines; the first line logs the original string, and the second line logs the resulting array.

```
Harry Trump ;Fred Barney; Helen Rigby ; Bill Abel ;Chris Hand
Harry Trump,Fred Barney,Helen Rigby,Bill Abel,Chris Hand
```

Returning a limited number of splits

In the following example, `split()` looks for 0 or more spaces in a string and returns the first 3 splits that it finds.

```
var myString = 'Hello World. How are you doing?';
var splits = myString.split(' ', 3);

console.log(splits);
```

This script displays the following:

```
Hello,World.,How
```

Capturing parentheses

If `separator` contains capturing parentheses, matched results are returned in the array.

```
var myString = 'Hello 1 word. Sentence number 2.';
var splits = myString.split(/(\d)/);

console.log(splits);
```

This script displays the following:

```
Hello ,1, word. Sentence number ,2,.
```

Reversing a String using `split()`

```
var str = 'asdfghjkl';
var strReverse = str.split('').reverse().join(''); // 'lkjhgfdsa'
// split() returns an array on which reverse() and join() can be applied
```

Bonus: use === operator to test if the original string was palindrome.

Specifications

Specification	Status	Comment
ECMAScript 3rd Edition (ECMA-262)	ST Standard	Initial definition. Implemented in JavaScript 1.1.
ECMAScript 5.1 (ECMA-262) The definition of 'String.prototype.split' in that specification.	ST Standard	
ECMAScript 2015 (6th Edition, ECMA-262) The definition of 'String.prototype.split' in that specification.	ST Standard	

Browser compatibility

Desktop Mobile

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

See also

- `String.prototype.charAt()`
- `String.prototype.indexOf()`
- `String.prototype.lastIndexOf()`
- `Array.prototype.join()`