JavaScript Array Methods

Let's see the list of JavaScript array methods with their description.

|  |  |
| --- | --- |
| **Methods** | **Description** |
| [concat()](https://www.javatpoint.com/javascript-array-concat-method) | It returns a new array object that contains two or more merged arrays. |
| [copywithin()](https://www.javatpoint.com/javascript-array-copywithin-method) | It copies the part of the given array with its own elements and returns the modified array. |
| [findIndex()](https://www.javatpoint.com/javascript-array-findindex-method) | It returns the index value of the first element in the given array that satisfies the specified condition. |
| [forEach()](https://www.javatpoint.com/javascript-array-foreach-method) | It invokes the provided function once for each element of an array. |
| [includes()](https://www.javatpoint.com/javascript-array-includes-method) | It checks whether the given array contains the specified element. |
| [indexOf()](https://www.javatpoint.com/javascript-array-indexof-method) | It searches the specified element in the given array and returns the index of the first match. |
| [isArray()](https://www.javatpoint.com/javascript-array-isarray-method) | It tests if the passed value ia an array. |
| [join()](https://www.javatpoint.com/javascript-array-join-method) | It joins the elements of an array as a string. |
| [keys()](https://www.javatpoint.com/javascript-array-keys-method) | It creates an iterator object that contains only the keys of the array, then loops through these keys. |
| [lastIndexOf()](https://www.javatpoint.com/javascript-array-lastindexof-method) | It searches the specified element in the given array and returns the index of the last match. |
| [reverse()](https://www.javatpoint.com/javascript-array-reverse-method) | It reverses the elements of given array. |
| [some()](https://www.javatpoint.com/javascript-array-some-method) | It determines if any element of the array passes the test of the implemented function. |
| [shift()](https://www.javatpoint.com/javascript-array-shift-method) | It removes and returns the first element of an array. |
| [slice()](https://www.javatpoint.com/javascript-array-slice-method) | It returns a new array containing the copy of the part of the given array. |
| [sort()](https://www.javatpoint.com/javascript-array-sort-method) | It returns the element of the given array in a sorted order. |
| [toString()](https://www.javatpoint.com/javascript-array-tostring-method) | It converts the elements of a specified array into string form, without affecting the original array. |

# JavaScript Array concat() Method

The JavaScript array concat() method combines two or more arrays and returns a new string. This method doesn't make any change in the original array.

## Syntax

The concat() method is represented by the following syntax:

1. array.concat(arr1,arr2,....,arrn)

## Parameter

**arr1,arr2,....,arrn** - It represent the arrays to be combined.

<!DOCTYPE html>

<html>

<body>

<script>

var arr1=["C","C++","Python"];

var arr2=["Java","JavaScript","Android"];

var result=arr1.concat(arr2);

document.writeln(result);

</script>

</body>

</html>

### Example 2

Here, we will print the combination of three arrays.

<!DOCTYPE html>

<html>

<body>

<script>

var arr1=["C","C++","Python"];

var arr2=["Java","JavaScript","Android"];

var arr3=["Ruby","Kotlin"];

var result=arr1.concat(arr2,arr3);

document.writeln(result);

</script>

</body>

</html>

# JavaScript Array copyWithin() method

The JavaScript array copyWithin() method copies the part of the given array with its own elements and returns the modified array. This method doesn't change the length of the modified array.

## Syntax

The copyWithin() method is represented by the following syntax:

1. array.copyWithin(target, start, end)

## Parameter

**target** - The position where the copied element takes place.

**tart** - It is optional. It represents the index from where the method starts copying elements. By default, it is 0.

**end** - It is optional. It represents the index at which elements stops copying. By default, it is array.length-1.

## Return

The modified array.

## JavaScript Array copyWithin() method example

Let's see some examples of copyWithin() method.

### Example 1

Here, we will pass the target, start and end index with the method.

<html>

<body>

<script>

var arr=["AngularJS","Node.js","JQuery",Bootstrap]

// place at 0th position, the element between 1st and 2nd position.

var result=arr.copyWithin(0,1,2);

document.writeln(result);

</script>

</body>

</html>

# javaScript Array findIndex() method

The JavaScript array findIndex() method returns the index of first element of the given array that satisfies the provided function condition. It returns -1, if no element satisfies the condition.

## Syntax

The findIndex() method is represented by the following syntax:

1. array.findIndex(callback(value,index,arr),thisArg)

## Parameter

**callback** - It represents the function that executes each element.

**value** - The current element of an array.

**index** - It is optional. The index of current element.

**arr** - It is optional. The array on which findIndex() method operated.

**thisArg** - It is optional. The value to use as this while executing callback.

## Return

The index of first element of the array that satisfies the function condition.

## JavaScript Array findIndex() method example

Let's see some examples of findIndex() method.

### Example 1

Let's see a simple example of findIndex() method.

<!DOCTYPE html>

<html>

<body>

<script>

var arr=[5,22,19,25,34];

var result=arr.findIndex(x=>x>20);

document.writeln(result)

</script>

</body>

</html>

# javaScript Array forEach() method

The JavaScript array forEach() method is used to invoke the specified function once for each array element.

## Syntax

The forEach() method is represented by the following syntax:

1. array.forEach(callback(currentvalue,index,arr),thisArg)

## Parameter

**callback** - It represents the function that test the condition.

**currentvalue** - The current element of array.

**index** - It is optional. The index of current element.

**arr** - It is optional. The array on which forEach() operated.

**thisArg** - It is optional. The value to use as this while executing callback.

## Return

undefined

## JavaScript Array forEach() method example

Let's see some examples of forEach() method.

### Example 1

Here, we will print the array elements using forEach().

<!DOCTYPE html>

<html>

<body>

<script>

var arr = ['C', 'C++', 'Python'];

arr.forEach(function(fetch) {

document.writeln(fetch);

});

</script>

</body>

</html>