JS JavaScript TUTORIAL

Array.prototype.concat()

Summary: in this tutorial, you will learn how to use the JavaScript Array `concat()` method to merge two or more arrays into a single array.

Introduction to the JavaScript Array concat() method

The Array `concat()` method allows you to merge elements of two or more arrays into a single array.

Here's the syntax of the `concat()` method:

```
const newArray = array1.concat(array2, array3, ...)
```

In this syntax:

- `array2` , `array3` , … are the arrays you want to merge elements with the `array1` .

The `concat()` method returns a new array that contains the elements of all the input arrays: `array1` , `array2` , `array3` , and so on. More importantly, it *does not modify* the original array.

JavaScript Array concat() method examples

Let's take some examples of using the `concat()` method to merge elements of arrays.

Basic JavaScript Array concat() method examples

The following example uses the `concat()` method to merge two arrays of numbers:

```
let odds = [1, 3, 5];
let evens = [2, 4, 6];

let results = odds.concat(evens);
```

```
console.log({ results });
```

Output:

```
{ results: [ 1, 3, 5, 2, 4, 6 ] }
```

How it works.

First, define two arrays of numbers:

```
let odds = [1, 3, 5];
let evens = [2, 4, 6];
```

Second, return a new array that contains elements of the odds and evens array:

```
let results = odds.concat(evens);
```

We call the `concat()` method of the `odds` array method to merge elements of the two arrays.

Third, display the `results` array in the console:

```
console.log({ results });
```

Similarly, you can call the `concat()` method on an empty array denoted by (`[]`):

```
let odds = [1, 3, 5];
let evens = [2, 4, 6];

let results = [].concat(odds, evens);

console.log({ results });
```

Output:

```
{ results: [ 1, 3, 5, 2, 4, 6 ] }
```

Merging three arrays

The `concat()` method allows you to merge more than two arrays as shown in the following example:

```
let upper = ['A', 'B', 'C'];
let lower = ['a', 'b', 'c'];
let digits = [1, 2, 3];
let alphanumerics = upper.concat(lower, digits);

console.log({ alphanumerics });
```

Output:

```
{ alphanumerics: ['A', 'B', 'C', 'a',  'b', 'c', 1,   2,  3] }
```

In this example, we merge the three arrays: `upper`, `lower`, and `digits` into a single array.

Copying an array

The `concat()` method copies an array if you don't pass any argument. For example:

```
let colors = ['red', 'green', 'blue'];
let rgb = colors.concat();

console.log({ rgb });
```

Output:

```
{ rgb: [ 'red', 'green', 'blue' ] }
```

Appending elements to an array

If you pass values, which are not arrays, into the `concat()` method, it will append each value to the end of the result array:

```
let rgb = ['red', 'green', 'blue'];
let moreColors = rgb.concat('yellow', 'magento');

console.log({ moreColors });
```

Output:

```
{ moreColors: [ 'red', 'green', 'blue', 'yellow', 'magento' ] }
```

Merging arrays using the spread operator

Starting from ES6, you can use the spread operator to merge multiple arrays as follows:

```
let odds = [1, 3, 5];
let evens = [2, 4, 6];

let results = [...odds, ...evens];

console.log({ results });
```

Output:

```
{ results: [ 1, 3, 5, 2, 4, 6 ] }
```

Summary

- Use the `concat()` method to return an array that contains elements of input arrays.