

# Array.prototype.concat()

**concat()** 方法用于合并两个或多个数组。此方法不会更改现有数组，而是返回一个新数组。

```
1  var arr1 = ['a', 'b', 'c'];
2  var arr2 = ['d', 'e', 'f'];
3
4  var arr3 = arr1.concat(arr2);
5
6  // arr3 is a new array [ "a", "b", "c", "d", "e", "f" ]
```

## 语法

```
var new_array = old_array.concat(value1[, value2[, ...[, valueN]]])
```

## 参数

**valueN**

将数组和/或值连接成新数组。详情请参阅下文描述。

## 返回值

新的 [Array](#) 实例。


## 描述

**concat** 方法创建一个新的数组，它由被调用的对象中的元素组成，每个参数的顺序依次是该参数的元素（如果参数是数组）或参数本身（如果参数不是数组）。它不会递归到嵌套数组参数中。

**concat** 方法不会改变 **this** 或任何作为参数提供的数组，而是返回一个浅拷贝，它包含与原始数组相结合的相同元素的副本。原始数组的元素将复制到新数组中，如下所示：

- 对象引用（而不是实际对象）：**concat** 将对象引用复制到新数组中。原始数组和新数组都引用相同的对象。也就是说，如果引用的对象被修改，则更改对于新数组和原始数组都是可见的。这包括也是数组的数组参数的元素。

- 数据类型如字符串，数字和布尔（不是 `String`，`Number` 和 `Boolean` 对象）：`concat` 将字符串和数字的值复制到新数组中。

 **注意：**数组/值在连接时保持不变。此外，对于新数组的任何操作（仅当元素不是对象引用时）都不会对原始数组产生影响，反之亦然。

## 示例

### 连接两个数组

以下代码将两个数组合并为一个新数组：

```
1  var alpha = ['a', 'b', 'c'];
2  var numeric = [1, 2, 3];
3
4  alpha.concat(numeric);
5  // result in ['a', 'b', 'c', 1, 2, 3]
```

### 连接三个数组

以下代码将三个数组合并为一个新数组：

```
1  var num1 = [1, 2, 3],
2      num2 = [4, 5, 6],
3      num3 = [7, 8, 9];
4
5  var nums = num1.concat(num2, num3);
6
7  console.log(nums);
8  // results in [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

### 将值连接到数组

以下代码将三个值连接到数组：

```
1  var alpha = ['a', 'b', 'c'];
2
3  var alphaNumeric = alpha.concat(1, [2, 3]);
4
5
```

```
6 | console.log(alphaNumeric);  
   // results in ['a', 'b', 'c', 1, 2, 3]
```

## 合并嵌套数组

以下代码合并数组并保留引用：

```
1 | var num1 = [[1]];  
2 | var num2 = [2, [3]];  
3 |  
4 | var nums = num1.concat(num2);  
5 |  
6 | console.log(nums);  
7 | // results in [[1], 2, [3]]  
8 |  
9 | // modify the first element of num1  
10 | num1[0].push(4);  
11 |  
12 | console.log(nums);  
13 | // results in [[1, 4], 2, [3]]
```

## 规范

| Specification   | Status                                  | Comment  |
|---|---|--|
| <a href="#">ECMAScript 3rd Edition (ECMA-262)</a>                                 | <div><div></div>STStandard</div>        | Initial definition. Implemented in JavaScript 1.2. |
| <a href="#">ECMAScript 5.1 (ECMA-262)</a><br>Array.prototype.concat               | <div><div></div>STStandard</div>        |  |
| <a href="#">ECMAScript 2015 (6th Edition, ECMA-262)</a><br>Array.prototype.concat | <div><div></div>STStandard</div>        |  |
| <a href="#">ECMAScript Latest Draft (ECMA-262)</a><br>Array.prototype.concat      | <div><div></div>LSLiving Standard</div> |  |

## 浏览器支持

## 相关链接

- [push](#) / [pop](#) — add/remove elements from the end of the array
- [unshift](#) / [shift](#) — add/remove elements from the beginning of the array
- [splice](#) — add/remove elements from the specified location of the array
- [String.prototype.concat\(\)](#)
- [Symbol.isConcatSpreadable](#) – control flattening.