

操作方法

ECMAScript 为操作已经包含在数组中的项提供了很多方法。其中，`concat()` 方法可以基于当前数组中的所有项创建一个新数组。具体来说，这个方法会先创建当前数组一个副本，然后将接收到的参数添加到这个副本的末尾，最后返回新构建的数组。在没有给 `concat()` 方法传递参数的情况下，它只是复制当前数组并返回副本。如果传递给 `concat()` 方法的是一或多个数组，则该方法会将这些数组中的每一项都添加到结果数组中。如果传递的值不是数组，这些值就会被简单地添加到结果数组的末尾。下面来看一个例子。

```
var colors = ["red", "green", "blue"];
var colors2 = colors.concat("yellow", ["black", "brown"]);

alert(colors);      //red,green,blue
alert(colors2);     //red,green,blue,yellow,black,brown
```

下一个方法是 `slice()`，它能够基于当前数组中的一或多个项创建一个新数组。`slice()` 方法可以接受一或两个参数，即要返回项的起始和结束位置。在只有一个参数的情况下，`slice()` 方法返回从该参数指定位置开始到当前数组末尾的所有项。如果有两个参数，该方法返回起始和结束位置之间的项——但不包括结束位置的项。注意，`slice()` 方法不会影响原始数组。请看下面的例子。

```
var colors = ["red", "green", "blue", "yellow", "purple"];
var colors2 = colors.slice(1);
var colors3 = colors.slice(1,4);

alert(colors2);     //green,blue,yellow,purple
alert(colors3);     //green,blue,yellow
```



如果 `slice()` 方法的参数中有一个负数，则用数组长度加上该数来确定相应的位置。例如，在一个包含 5 项的数组上调用 `slice(-2, -1)` 与调用 `slice(3, 4)` 得到的结果相同。如果结束位置小于起始位置，则返回空数组。

下面我们来介绍 `splice()` 方法，这个方法恐怕要算是最强大的数组方法了，它有很多种用法。`splice()` 的主要用途是向数组的中部插入项，但使用这种方法的方式则有如下 3 种。

- **删除**：可以删除任意数量的项，只需指定 2 个参数：要删除的第一项的位置和要删除的项数。例如，`splice(0,2)` 会删除数组中的前两项。
- **插入**：可以向指定位置插入任意数量的项，只需提供 3 个参数：起始位置、0（要删除的项数）和要插入的项。如果要插入多个项，可以再传入第四、第五，以至任意多个项。例如，`splice(2,0,"red","green")` 会从当前数组的位置 2 开始插入字符串 "red" 和 "green"。
- **替换**：可以向指定位置插入任意数量的项，且同时删除任意数量的项，只需指定 3 个参数：起始位置、要删除的项数和要插入的任意数量的项。插入的项数不必与删除的项数相等。例如，`splice(2,1,"red","green")` 会删除当前数组位置 2 的项，然后再从位置 2 开始插入字符串 "red" 和 "green"。

`splice()` 方法始终都会返回一个数组，该数组中包含从原始数组中删除的项（如果没有删除任何项，则返回一个空数组）。下面的代码展示了上述 3 种使用 `splice()` 方法的方式。



```
var colors = ["red", "green", "blue"];
var removed = colors.splice(0,1);           // 删除第一项
alert(colors);      // green,blue
alert(removed);     // red, 返回的数组中只包含一项

removed = colors.splice(1, 0, "yellow", "orange"); // 从位置 1 开始插入两项
alert(colors);      // green,yellow,orange,blue
alert(removed);     // 返回的是一个空数组

removed = colors.splice(1, 1, "red", "purple"); // 插入两项，删除一项
alert(colors);      // green,red,purple,orange,blue
alert(removed);     // yellow, 返回的数组中只包含一项
```

ArrayTypeSpliceExample01.htm