How to combine JavaScript arrays

The concat method can be used to combine two arrays. This method returns a new array and the original arrays are not changed in any way.

If you've never had a need to combine two JavaScript arrays, you can pretty much count on having to, one day. And if you've ever entertained thoughts of rolling your own solution to this problem, I'd recommend against it. So now's a good time to talk about the array.prototype.concat method, which provides a simple way to combine two JavaScript arrays. One thing that's important to know about the concat method is that it does not change the original arrays. In other words, a new array is created and it will consist of the two specified arrays. But keep in mind that if either array contains one or more objects, those objects are passed by reference in JavaScript. So, even though the original arrays are not changed, if you make a change to any of the objects in the new array, that change will be reflected in the original (source) array.

Try it yourself !

In the above example, click the **Result** tab. The first call to the concat method is used to combine two arrays: **["a", "b", "c"]** and **["d", "e", "f"]**. The result is a new array: **["a","b","c","d","e","f"]**. It is important to note that in *both* cases, the original array is not changed, which you can see from the second and third **console.dir** statements. Click the **Result** tab in order to see the results of each **console.dir** statement.

We can also pass a single value to the concat method in this way: `myArray1.concat("hello")` . When we do that, a new array is returned with the single value we have provided as the last element. The effect is very similar to using the Array **push()** method. The main difference is that the **push()** method *changes* the original array and returns the new length of that array, whereas the **concat** method does *not* change the original array and returns a new array . In a similar manner, we can also pass multiple values to the **concat** method: `myArray1.concat("hello", "goodbye")` . This returns a new array with the two values added to the end: **["a","b","c","hello","goodbye"].**

Summary

So, here you've had a look at what a handy tool the array.prototype.concat method is. It not only allows you to combine two JavaScript arrays, it also allows you to add elements to the new array at the same time. But here are two helpful and important things to keep in mind: the original arrays are not changed, and any changes to objects that exist in the new array will be reflected in the original array that it came from.