Array Methods to Manipulate Strings

In this lesson we will study various ways to manipulate strings using array methods in JavaScript.

WE'LL COVER THE FOLLOWING

Arrays have a number of methods that can be used to manipulate existing arrays.

The `concat()` and `slice()` methods

Listing 7-18 demonstrates two of them, `concat()`, and `slice()`.

Listing 7-18: Exercise-07-18/index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Concat and slice</title>
  <script>
    var fruits = ["apple", "banana"];
    var fruit2 = fruits.concat(["orange", "lemon"],
      "pear");
    console.log(fruit2.toString());

    var fruit3 = fruit2.slice(2, 4);
    console.log(fruit3.toString());
  </script>
</head>
<body>
  Listing 7-18: View the console output
</body>
</html>
```

This code snippet produces this output:

This code snippet produces this output:

Console

```
apple,banana,orange,lemon,pear
orange,lemon
```

The `concat()` method returns a new array comprised of this array joined with other arrays and values. In this listing, `fruit2` is assembled from the original contents of fruits and from the items in the `["orange", "lemon"]` array, plus "pear," as shown in the first line of the output. It's important to know that `concat()` does not change the array it is invoked on, instead, it retrieves a new, comprised array.

The `slice()` method returns an extracted section of an array without altering the original array. It accepts two arguments; an index to start extraction (first argument), and an index to end extraction (second argument). The extraction does not include the element at the end index. The `slice(2, 4)` call in the code extracts the third and fourth element from the array (second line of the output), but not the fifth.

The powerful `splice()` method

There is a more powerful method, `splice()`, that can be used to delete items from the array, to replace existing items, or even to insert new items, depending on the arguments passed to the method.

Listing 7-19 demonstrates these operations.

Listing 7-19: Exercise-07-19/index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Splice</title>
  <script>
    var nums = ["one", "two", "three"];

    // Insert a new element
    var removed = nums.splice(2, 0, "hey!");
    console.log(nums.toString());

    // Remove elements
    removed = nums.splice(2, 2);
    console.log(nums.toString());
```

```
    // Replace elements
    removed = nums.splice(0, 1, 2, 3, 4);
    console.log(nums.toString());
  </script>
</head>
<body>
  Listing 7-19: View the console output
</body>
</html>
```

This code creates this output:

console

```
one,two,hey!,three
one,two
2,3,4,two
```

The `splice()` methods combines two operations. It removes a section of the array and inserts a sequence of new elements. The removed section is returned by the method. The first argument marks the index at which to start changing the array.

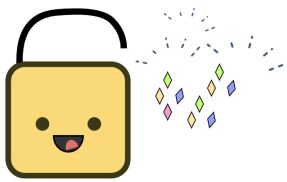
The second integer argument indicates the number of old array elements to remove. All other arguments are considered elements to add to the array at the starting index specified in the first argument. The first invocation of `splice()` did not remove any elements because its second argument was set to zero.

However, it added “hey!” after the second item, as shown in the first line of the output. Because the second `splice()` call did not pass any elements to insert, it simply removed two elements starting from **index 2** (see the second line of the output).

The third call replaced the first element with three new elements (2, 3, and 4), because it removed one element at index 0, and then inserted 2, 3, and 4 to the array, as shown by the last output line.

Achievement unlocked!

Congratulations! You’ve learned how to manipulate strings in JavaScript using various array methods.



Great work! Give yourself a round of applause! :)

In the *next lesson*, we will look at array methods that can be used to manipulate element ordering.