**Array Manipulation in JavaScript –**

[David Nene](https://medium.com/@davidnene?source=post_page-----acf0d2bc4843--------------------------------) Jun 29, 2022

There are a number of ways to manipulate arrays in JavaScript. We shall discuss some of them here.

A required prerequisite is to have basic knowledge of arrays and JavaScript.

In this article, we shall discuss:

*1. Creating an array.*

*2. Adding elements to an array.*

*3. Removing elements from an array.*

*4. Finding an element in an array using its index.*

*5. Finding the index of an element in an array.*

*6. Replacing an element of an array using its index.*

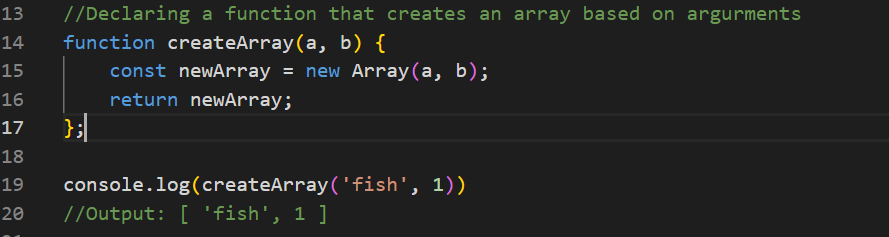
*7. Declaring a function that returns common elements in two arrays.*

**1. Creating an array**

**Literal notation:** We can create an array by declaring a variable and assigning values in square brackets (separated by commas) to the variable.  
**const array1 = ['water', 'onions', 'pepper', 20, 22];  
console.log(array1);  
//Output: ['water', 'onions', 'pepper', 20, 22]**

**Inbuilt constructor:**We can also create an array using a constructor as follows:  
**const array2 = new Array('Soup', 'onions', 10);  
console.log(array2);  
//Output: ['soup', 'onions', 10]**

We can also declare a function that takes in arguments and returns an array comprised of the arguments as follows:



**2. Adding elements to an array**

We can add elements to an array either before the initial element or after the last element.  
**Before the initial element using unshift()**  
**array1.unshift('heater');  
console.log(array1);  
//Output: ['heater', 'water', 'onions', 'pepper', 20, 22]**

**After the last element using push()  
array1.push('oil');  
console.log(array1);  
//Output:['heater', 'water', 'onions', 'pepper', 20, 22, 'oil']**

**3. Removing elements from an array**

We can remove an element from an array in various ways.

**Removing the initial element using shift()  
console.log(array1);  
//Output:['heater', 'water', 'onions', 'pepper', 20, 22, 'oil']**

**array1.shift(); //note we do not pass any argument into the function console.log(array1);  
//Output: ['water', 'onions', 'pepper', 20, 22, 'oil']  
//This can be used in queues data structures (FIFO)**

**Removing the last element using pop()  
console.log(array1);  
//Output: ['water', 'onions', 'pepper', 20, 22, 'oil']**

**array1.pop(); //note we do not pass any argument into the function  
console.log(array1);  
//Output: ['water', 'onions', 'pepper', 20, 22]  
//This can be used in stacks data structures (LIFO)**

**Removing an element by its index using splice()**  
Splice is a special function that can remove, add or replace an element in an array.  
In our case, let’s see how we can use it to remove an element in a specific index. Splice will return an array of the removed element(s).  
Note indices in an array start to count from zero. Fox example index zero of **array1**above is **water**index one is **onions** and so forth.

Let’s remove the element at index 2 of **array1**which is **pepper**.  
**console.log(array1);  
//Output: ['water', 'onions', 'pepper', 20, 22]**

**let removedElements = array1.splice(2, 1);  
//the first arguement(2) specificies the array index  
//the second argument specifies how many elements to be removed from the //specified index.  
//In our case, we just want to remove one element which is at the //specified index**

**console.log(array1);  
//Output: ['water', 'onions', 20, 22]**

**console.log(removedelement);  
//Output: ['pepper']  
//Try passing 2 to the second argument of splice and see what happens**

*We shall see more on how to replace values using splice in part 6  
Additional: Check out how to add elements using splice()*[*here*](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/splice)

**4. Getting an element of an array by its index**

**console.log(array1);  
//Output: ['water', 'onions', 20, 22]  
console.log(array1[1])  
//Output: onions  
//Here we passed index 1. Note we used square brackets**

**console.log(array1[-1]  
//Output: 22  
//[-1] returns the last element of the array**

**Getting a range of elements using slice():**We can use slice to get a range of elements by their indices.  
**console.log(array1);  
//Output: ['water', 'onions', 20, 22]**

**console.log(array1.slice(0,2)  
//Output: ['water', 'onions']  
//The first argument(0) specifies the start point  
//The second argument(2) secifies the end point  
//You might be suprised why index 2 is not printed out  
//This is because the end point is not usually returned**

**console.log(array1.slice(0,1)  
//Output: ['water'] //The endpoint is not returned**

**5. Getting the index of an element in an array using indexOf()**

IndexOf() receives the element as an argument and returns its index. In case the element does not exist, it will return -1.  
**console.log(array1);  
//Output: ['water', 'onions', 20, 22]  
console.log(array1.indexOf('onions');  
Output: 1**

**console.log(array1.indexOf('soup');  
//Output: -1  
//soup does not exist in array1 above**

**6. Replacing an element of an array using its index**

**Literal notation:**  
**console.log(array1);  
//Output: ['water', 'onions', 20, 22]**

**//Lets replace onions with pot in the array1 above**

**array1[1] = 'pot'; //Note we passed the array index to be replaced  
console.log(array1);  
//Output: ['water', 'pot', 20, 22]**

**Inbuilt function: splice()**Remember that splice is a special function that can do various manipulations. In this case, we shall use it to replace an element in an array. It will return the value that has been replaced.  
**console.log(array1);  
//Output: ['water', 'pot', 20, 22]  
//We shall replace 22 to 25**

**let replaced = array1.splice(3, 1, 25)**

**console.log(array1)  
//Output: ['water', 'pot', 20, 25]  
console.log(replaced)  
//Output: 22**

**//the first argument(3) is the specified index to be replaced  
//the second argument(1) specifies the number of elements to be removed  
//at the specified index (in our case it's one element hence we pass 1)  
//the third argument is the element to be added at the specified index**

Note: splice is a destructive function. This means it manipulates the original array. This is not advisable since it might lead to unwanted changes and more debugging time. To curb this, we usually create a copy of the array first, then perform our manipulation.

**console.log(array1);  
//Output: ['water', 'pot', 20, 25]**

**newArray1 = [...array1]; //this creates a copy of array1  
console.log(newArray1);  
//Output: ['water', 'pot', 20, 25]**

**let replaced1 = newArray1.splice(3, 1, 30);**

**console.log(newArray1);  
//Output: ['water', 'pot', 20, 30]  
console.log(array1);  
//Output: ['water', 'pot', 20, 25]  
//array1 elements are still intact**

**7. Declaring a function that returns common elements in two arrays**

Let's create a simple function that loops through two arrays and returns an array of elements that are common.

Text

Description automatically generated