

# JS Assignment

- **Due** May 30th, 2020 by 10:00:00AM
- **Points** 20
- **Submitting:** Submit your REPL.it link via Google Classroom

## Goal:

To improve your understanding of JavaScript syntax, functions, and higher-order functions, and to better understand JavaScript and Array methods.

## Assignment:

The following (12) are some of the most popular and used methods in functional languages:

- forEach
- Map
- Filter
- Some (Any)
- Every
- Reduce
- Includes
- indexOf
- Push
- lastIndexOf
- Object.keys()
- Object.values()

For this assignment, you will recreate these methods using JavaScript functions. Make sure to carefully understand what each method is designed to do, and DO NOT use any of the respective native JS methods to implement your solutions. Also, keep in mind that these methods do not (except for push()) mutate the input array. **We highly suggest utilizing Mozilla Developer Network (MDN) Web Docs to understand how each method works and what arguments they take. Also look into Array.prototype.pop(), Array.prototype.shift(), and other methods for your own enrichment.**

### **forEach()**

Without using the native “Array.prototype.forEach” method of JavaScript, compose a function titled “myEach” that will take in an array of elements and executes any callback function (provided by you) on each of those elements.

### **map()**

Without using the native “Array.prototype.map” method of JavaScript, compose a function titled “myMap” that will take in an array of elements and executes a callback function (provided by you) on each of those elements.

### **filter()**

Without using the native “Array.prototype.filter” method of JavaScript, compose a function titled “myFilter” that will take in an array of elements and executes a callback function (provided by you) on each of those elements.

### **some() (aka any())**

Without using the native “Array.prototype.some” method of JavaScript, compose a function titled “mySome” that will take in an array of elements and executes a callback function (provided by you) on each of those elements.

### **every()**

Without using the native “Array.prototype.every” method of JavaScript, compose a function titled “myEvery” that will take in an array of elements and executes a callback function (provided by you) on each of those elements.

### **reduce()**

Without using the native “Array.prototype.reduce” method of JavaScript, compose a function titled “myReduce” that will take in an array of elements and executes a callback function (provided by you) on each of those elements.

### **includes()**

Without using the native “Array.prototype.includes” method of JavaScript, compose a function titled “myIncludes” that will take in an array of elements and indicates whether or not a target element is contained within the input array. This returns a boolean.

### **indexOf()**

Without using the native “Array.prototype.indexOf” method of JavaScript, compose a function titled “myIndexOf” that will take in an array of elements and returns the index of the first encounter of a target element (if it is found) or -1 if that element does not exist within the input array.

### **push()**

Without using the native “Array.prototype.push” method of JavaScript, compose a function titled “myPush” that will take in an array of elements as well as an elementToAdd and append that element to the end of the array.

### **lastIndexOf()**

Without using the native “Array.prototype.lastIndexOf” method of JavaScript, compose a function titled “myUnshift” (typo) that will take in an array of elements and returns the index of the last encounter of a target element (if it is found) or -1 if that element does not exist within the input array.

### **Object.keys()**

Without using the native “Object.keys()” method of JavaScript, compose a function titled “grabKeys” that will take in an object and return all of the keys of the key:value pairs of that object. Modify the Object class, appending a static method to the Object class., appending a static method to the Object class. Leverage the “for...in” loop.

### **Object.values()**

Without using the native “Object.values()” method of JavaScript, compose a function titled “grabValues” that will take in an object and return all of the values of the key:value pairs of that object. Modify the Object class, appending a static method to the Object class. Leverage the “for...in” loop.

## Miscellaneous Problems (4)

From [Chapter 4 of Eloquent JavaScript](#)

- Sum of a Range
- Reversing an Array
- A List
- Deep Comparison

### Miscellaneous Problems (1)

**1:** Given an array `nums`, write a function to move all 0's to the end of it while maintaining the relative order of the non-zero elements.

```
function moveZeros (nums) {  
  
}
```

Example:

Input: [0,1,0,3,12]

Output: [1,3,12,0,0]

Note:

You must do this in-place without making a copy of the array.