JS Assignment

* **Due** Jan 4 by 10am
* **Points** 12
* **Submitting** website URL(repl.it, github repo, gist) or file upload)

**Goal:**

To strengthen your understanding of JavaScript syntax, functions, and Higher-Order functions, and to understand Array methods.

**Assignment:**

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

* forEach
* Map
* Filter
* Some (Any)
* Every
* Reduce

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 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.push(), Array.prototype.pop(), Array.prototype.shift(), Array.prototype.unshift(), Array.prototype.slice(), Array.prototype.splice() methods as well.**

**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 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 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 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 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 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 on each of those elements.

**BONUS**

For those that have finished the homework before leaving today, or that want some more practice and challenge with JavaScript, please refer to the exercises at the bottom of [Chapter 4 of Eloquent JavaScript](http://eloquentjavascript.net/04_data.html). They are as follows:

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

These problems may dive into topics not covered in class, and are technically considered bonus work. If you found the graded problems easy and/or finished them before leaving today, it is **strongly suggested** that you at least attempt to finish the EJS problems. You can include them in the same gist/repl as the original problems or create a separate gist/repl for submission.