9/30/2020

CS303 Object Oriented and Functional Programming in JavaScript

**Part 0: Array method practice:** Write map, filter, find, include, and reduce functions as described below for these two arrays. Write Mocha tests for each.

```
const numArray = [5,0, 7, 77,  -20, 300, 51, 2]
 const peopleArray = [{name: "Sam", age: 15}, {name: "William", age: 6}, {name: "Lucy", age: 13}, {name: "Barney", age: 80}]
```

map:
- double numbers
- double age

filter:
- filter all even numbers
- filter all age > 10
- find even, include even
- find age > 10, include age > 10

reduce
- find sum of numbers
- find average of numbers
- find max of numbers
- find max for ages
- use a chain of map filter reduce to find the average age of people with even number ages
- use a chain of map filter reduce to find the average age of people with odd number ages

Example of a nice answer from the quiz this morning: (Nothing to do here—just FYI.)

```
abc.map(element => element.age).reduce((max, value) => max>value, 0);
```

**W2D3 Recursion**

**Part I:** Complete the following tasks from The JavaScript Language book. Try to answer them before looking at the solutions. You should work on each one for at least 10 – 20 minutes before looking at the solution. Write your own code for each of these in VSCode before looking at the answers.

In your github README file include a statement about whether you were able to complete all of the exercises on your own and whether you have questions or would like to hear an explanation on any of the solutions from the book. Also comment on something that was interesting or that you learned from the lab. Finally please note whether you had sufficient time to complete the lab and about how long it took.

Chapter: Advanced working with functions

Section: Recursion and stack

- Sum all numbers till the given one
- Calculate factorial
- EC: Fibonacci numbers
- Output a single-linked list
- EC: Output a single-linked list in reverse order

**Part II:** Write a recursive function that will print “name: value” to the console for every node in the following tree data structure. Then write an iterative function that also does this. The following is the output you should have for the recursive version. The iterative version should be similar, but it is not necessary for every line to be in the same order.

body: null
div: null
label: Name
input: this was typed by a user
p: This is text in the a paragraph

```
let node3 = {
   name: "p",
   value: "This is text in the a paragraph",
   children: null
};

let node4 = {
   name: "label",
   value: "Name",
   children: null
};

let node5 = {
   name: "input",
   value: "this was typed by a user",
   children: null
};

let node2 = {
   name: "div",
   value: null,
   children: [node4, node5]
};
```

```
let node1 = {
   name: "body",
   children:  [node2, node3],
   value: null,
};
```

GRADING RUBRIC

- 8 points.  Good status report including:
    - description of which exercises are complete or not complete,
    - Link to Github website page,
    - time spent,
    - problems encountered,
    - outstanding questions,
    - any noteworthy insights
- 9 points
    - Status report as above and most of assignments completed
- 10 points
    - Above and almost everything complete with neat code and output
- 7 points or less
    - Weak, incomplete status report