CMPS 356 Enterprise Application Development
Lab 4 – JavaScript Fundamentals

Objective

The objective of this lab is to practice the following JavaScript topics.

- ✓ **Control Structures**
- ✓ **Functional Programming and Higher Order Functions**
  - **Arrow function**: allows shorter syntax for writing **functions**.
  - **Array functions** (.map, .reduce, .filter, flat,.splice, .sort…)
  - **Spread operator**

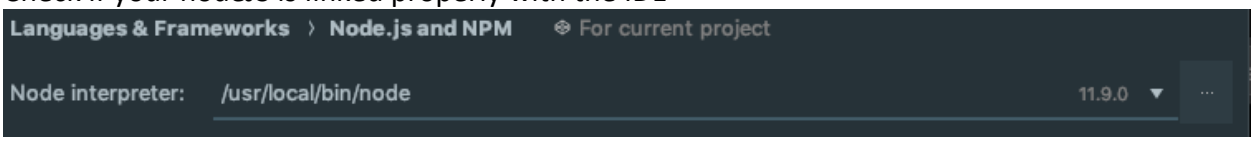
Overview

This Lab has two parts:

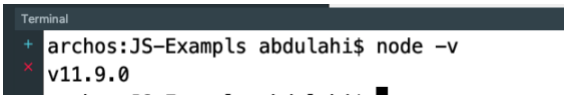
- ❖ **PART A**: Has some warm up exercise to help you practice control structures, arrow functions and array functions
- ❖ **PART B:** assessment on processing array and arrow functions

Preparation

1. If you have not installed nodeJS in your laptop then visit this site and install the appropriate nodejs for your platform https://nodejs.org/en/download/
2. Go to
   - ➢ File | Settings | Languages and Frameworks | Node.js and NPM for Windows and Linux
   - ➢ WebStorm | Preferences | Languages and Frameworks | Node.js and NPM for macOS
3. Check if your nodeJS is linked properly with the IDE



4. Press OK and restart your IDE
5. Now in the terminal type “node -v” you should see the version of the nodejs you have installed



6. If the above command did not work for you, then you will need to do the following.
    a. **You need to Add C:\Program Files\nodejs to your PATH environment variable. To do this follow these steps:**
    b. **Use the global Search Charm to search "Environment Variables"**
    c. **Click "Edit system environment variables"**
    d. **Click "Environment Variables" in the dialog.**
    e. **In the "System Variables" box, search for Path and edit it to include C:\Program Files\nodejs. Make sure it is separated from any other paths by a ;.**
    f. **You will have to restart any currently-opened command prompts before it will take effect.**

PART A – Warm up JS exercises

1. Create a JavaScript file named app.js inside the Lab 4-JS folder
2. Check the following tutorial on **ECMAScript6 -** https://www.w3schools.com/js/js_es6.asp
3. Create an array that contains the following numbers [2,4,18,28,9,5,6,7,8,9] and name it **dataPoints**;
4. Implement the following functions and use the one of the Console functions (.log , .error , .trace , .table , .time , .endTime) to display your results.
    a. Display all the elements in the dataPoints array by using For..of and forEach
    b. Remove the first two elements from the dataPoints array
    c. Remove the last two elements from the dataPoints array
    d. Add 10 and 12 to the dataPoints array
    e. Delete the largest number from the dataPoints array
    f. Sort the elements in the dataPoints array in both ascending and descending orders
    g. Create a second array named newPoints which contains the following values 55,66,77,88,99.
    h. Combine the two arrays into a single array. You can add the newPoints array to the dataPoints array
    i. Find the sum of all the elements inside dataPoints array
    j. Find the maximum and the minimum numbers in the array
    k. Extract all the numbers that are greater than 15 and find their sum. You should write everything as one single statement.

5. **let *matrix*** = [[2, 3], [34, 89], [55, 101, 34], [34, 89, 34, 99]];

   Use the above array and Implement and test the following functions:

   - **flatten:** gets a matrix (i.e., array of arrays) and returns a single dimensional flat array.
   - **max**: gets an array and returns its maximum value.
   - **sort:** gets an array and returns a sorted array in descending order (from big to small).
   - **square:** gets an array and returns an array with squared values.
   - **average:** gets an array and returns its average.
   - **removeDuplicate:** gets an array and returns an array without duplicate elements.

   Use the following matrix to test your work.

Expected output:

```
Original array:
[ [ 2, 3 ], [ 34, 89 ], [ 55, 101, 34 ], [ 34, 89, 34, 99 ] ]

Flattened:
[ 2, 3, 34, 89, 55, 101, 34, 34, 89, 34, 99 ]

Max value:
101

Sorted in descending order:
[ 101, 99, 89, 89, 55, 34, 34, 34, 34, 3, 2 ]

Without duplicate elements:
[ 101, 99, 89, 55, 34, 3, 2 ]

Sum of unique elements:
574

Square of unique elements:
10201 9801 7921 7921 3025 1156 1156 1156 1156 9 4
```

Java Script Best Practices : https://www.w3schools.com/js/js_best_practices.asp