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**

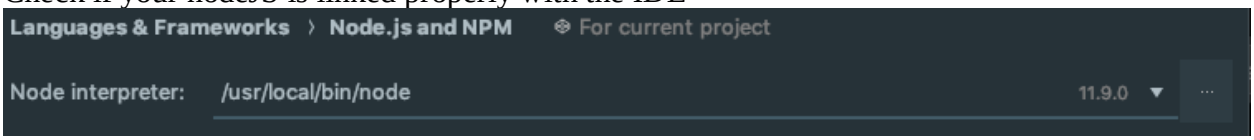
Overview

This Lab has two parts:

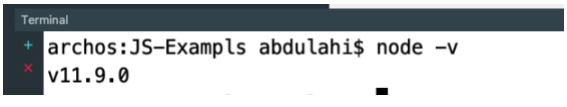
- ❖ **PART A**: Has some warm up exercise to help you practice control structures, arrow functions and array functions
- ❖ **PART B:** assessment on processing array and arrow functions

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



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
```

PART B – Assessment

Control Structures, Arrays and Functions

Due Date 1 Day before the Lab at 11:59PM

1. Using a ***While*** loop, write a JavaScript program that displays odd numbers from 1 to 100.
2. Rewrite the first program using a ***For*** loop.
3. Consider the following array declaration: *let cars = ["Saab", "Volvo", "BMW"];*
    - Add "**Toyota**" to the end of the array
    - Add "**Mercedes**" to the beginning of the array.
    - Create a ***displayArray*** function that takes an array as argument and print the array elements individually using a ***For-of*** loop. Call the function to display the cars array.
    - Sort the array alphabetically and print it again.
4. Write a function named it ***avgArray*** to find and return the average of an integer array. Use it in your code.
5. Write and test a function named **dayOfWeek**. It takes a day number and return the day name (e.g., *dayOfWeek(1)* return Monday). Tip: use a switch to implement this function.

6. Write and test function named **drawTriangle** to draw a Triangle. For example if you call *drawTriangle(5)* you get the following output:

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Tip: use a nested loop and array **push** and **join** functions.

7. Consider the following variables:

```
let string = 'Hello', int = 254;
let float = 25.4;
let arr = [1, 2, 3];
let object = {course: 'JS', part: 1};
let func = function(){return;};
let nullValue = null;
let undefinedValue;
let boolean = true;
```

Add these variables to an array named ***variables*** then loop through the array using for-of loop to display the value of each variable and its data type. The output should be as follows:

```
Hello is string
254 is number
25.4 is number
1,2,3 is object
[object Object] is object
function (){return;} is function
null is object
undefined is undefined
true is boolean
```

8. Write a function named **greet** that takes three parameters **firstName**, **lastName** and **age** then returns the following greeting: *Hello, my name is …. …. I am …. years old*.

The function should replace …. with the function parameters using a **string template literal**.

If you test the function using **greet('Ali', 'Faleh' , 10)**; then you should get back: Hello, my name is Ali Faleh I am 10 years old.

9. Use a **map** to store then display the following data:

Key	Value
1	'Monday'
2	'Tuesday'
3	'Wednesday'
4	'Thursday'
5	'Friday'
6	'Saturday'
7	'Sunday'

10. Consider the following array:

```
let colors = ["white", "blue", "yellow", "black", "red", "green"]
```

Using array destructuring assign the first 2 elements to firstColor and secondColor variables and assign the remaining elements to otherColors variable. Display the values of these 3 variables.

11. Use the **spread** operator to produce an array named **seasons** by concatenating the following 2 arrays and adding extra elements "Cool!" and "Super Hot Summer!" as seasons.

Input arrays:

```
let cold = ['autumn', 'winter'];
let warm = ['spring', 'summer'];
```

Output seasons array:

```
["Cool!", 'autumn', 'winter', 'spring', 'summer', "Super Hot Summer!"]
```

12. Call the `Math.max()` function and pass the **nums** array to it. Do not forget to use **spread** operator to convert the array to multiple arguments expected by the max function.

```
let nums = [1, 2, 3, 4, 45, 5, 6]
```

13. Consider the following the `square` function and **nums** array.

a. Use the square function along with the array **map** function to square the elements of **nums** array and assigned the results to **squaredNums**. Then Display **squaredNums**.

b. Enhance the implementation done in previous question by removing the square function and using anonymous **arrow** function.

```
function square(x) {
    return x * x;
}
let numbers = [1, 2, 3, 4, 4, 5, 6, 7, 8, 9, 10];
```

14. Consider the following array:

```
let nums = [1, 2, 3, 4, 5, 6, 7, 8];
```

- Use **filter** then **map** array functions to filter even numbers then square them. Assign the results to a variable named **squaredEvenNums** then display it. The output should be:

  ```
  squaredEvenNums: [ 4, 16, 36, 64 ]
  ```

- Use the **reduce** array function to compute the of **sums** array. The output should be:

  ```
  Sum of array elements:  36
  ```

You must use arrow functions in this exercise.

15. Square and sum the elements of this array using arrow functions and in 1 line of code. Then find the average of the array.

```
let nums = [2, 4, 5];
```

16. Sort the in ascending, descending order [8,7,6,5,4,3,2,1]

Note that you must use **JavaScript features and capabilities** such as **arrow functions**, array functions **(.map, .reduce, .filter, .splice, .sort…) and spread operator**.

After you complete the lab, fill in the ***Lab3-TestingDoc-Grading-Sheet.docx*** and save it inside ***Lab3-JavScript*** folder. Sync your repository to push your work to Github.