JavaScript
AWP- Lab Work 5 – JS Essential Review Part 3

Name: ______________________________ **Mark: _____/100**

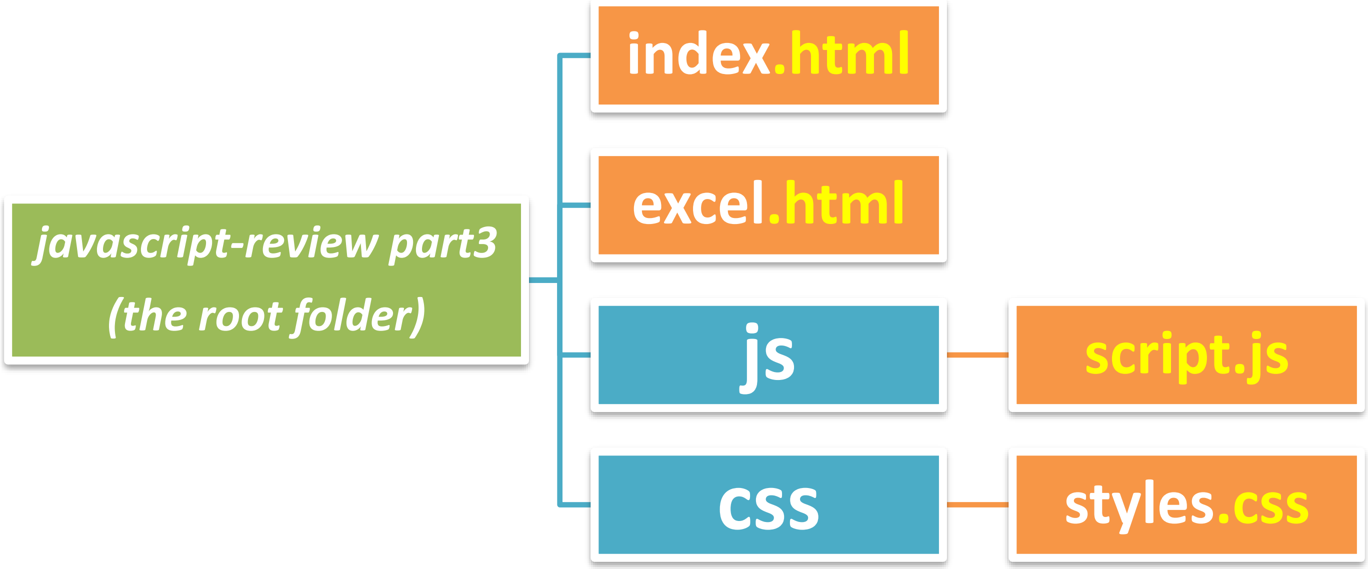
Important Notes:

- Use your Team group to work on this lab-work. Each team must work individually. For example, team1 members can work together they CANNOT work with or assist another member from a different team
- Although this lab-work is a team based, but each student must submit his/her work individually
- The lab-work should be **submitted (uploaded) before the due date** in order not to lose marks
- Follow the **exact naming rules** for all your variables as explained in the instructions
- The lab-work is intensively loaded with all the needed steps, hints, and instructions to solve the entire problem, but as we know, in programming we can use different logic or ways of coding to achieve the same task! So please feel free if you like/prefer to use other different ways or methods etc.
- At the end of the lab-work and after answering all the questions, you will end up by a folder:
  - **"JS-Lab-work"** or **"JavaScript-Lab-work"** which contains the following three files:
    - **index.html**
    - **excel.html**
    - **main.js (inside "js" folder)**
    - **styles.css (inside "css" folder)**
- **You can use Gitpod, your local Microsoft VS code, or any other IDE you prefer in all the cases the main folder of the lab-work (the entire lab-work code) has to:**
  - **Be one of your GitHub Repo.**
  - Be submitted to the lab work section on canvas:
  - Has to have Readme file that contains a link to the live demo

NOTE: Make this repo to be a GitHub pages website to this lab not your main GitHub website! For more information refer to my Zoom recording about Git/GitHub or just review my PDF file

(5 Marks)

Lab-work Folder Structure:



Lab-work Contents:

This lab-work will be a demonstration of your skills in using HTML form elements, CSS, and some of the JavaScript main topics that include:

- Defining different variables with local and global scopes
- Working with some JavaScript built-in (native) functions
- Creating custom functions and Debugging methods
- Parsing the DOM
- Form handling and validation
- Looping and Conditional statements
- Events handlers and Events listeners
- Formulas, and Math Object

A simple website of two pages only for handling the user input using HTML form elements and creating custom functions to imitate/mimic MS Excel commonly used functions.

Lab-work Instructions:

- Create **two** html pages using HTML5: **index.html** and **excel.html**
    - **index.html** contains an HTML form that needs to be validated by JavaScript when submitted
    - **excel.html** contains a simple form to let the user enter their numeric values then you will use excel four common functions to find: the **average**, **maximum**, **minimum**, and **total**

- Create an **external JavaScript** file: **script.js**
    - The file should be placed inside a **sub-folder named “js”**
    - All your JavaScript code must be placed in this file
    - The file must be linked to the two html files *(for sure)*

    Note: You should place the script link at the end of all the HTML elements just before the closing body tag for the two reasons that we discussed in our lectures (refer to my code examples)

- Create an **external Style Sheet** file: **styles.css**
    - The file should be placed inside a **sub-folder named “css”**
    - All your CSS rules must be placed in this file
    - The file must be linked to the two html files *(for sure)*
    - You can use any styling or any CSS rules you like to style your 2 HTML pages but remember that this lab-work is mainly about JavaScript, so it’s better to finish the lab-work JS requirements first then add the styles later

- The two HTML files, JS file, CSS file, plus the two subfolders have to be inside the lab-work **main (root folder)**

Please refer to the diagram on the first page for more clarification about the lab-work folder structure

The two HTML pages contents: *index.html and excel.html*

Create a simple navigation menu using **HTML5 element <nav> with <ul> and <a>** to enable the user easily navigate/switch between the two pages:

- Link name (title): **Home** ➔ Hyperlink to **index.html** file
- Link name (title): **Excel** ➔ Hyperlink to **excel.html** file

Both HTML pages (index and excel) should have the following template (or similar valid HTML5 boilerplate). Please feel free to use any other element(s) you like to present the result for the end user:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Full Review-Your Full Name</title>
    <!-- [[ code: the link to your external CSS file ]] -->
</head>
<body>
    <header>
        <nav>
            <ul>
                <li><a href="index.html">Home</a></li>
                <li><a href="excel.html">Excel</a></li>
            </ul>
        </nav>
    </header>

    <main>
        <form id="main-form" name="mainform">
            <!-- [[ code: the required form elements based on each page instructions ]] -->
        </form>

        <div id="output">
            <!-- here is where the user should see the result (output) -->
        </div>
    </main>

    <footer>
        <p>
          ILAC & EFREI Partnership- HTML, CSS, and JavaScript Main Lab-work - Your Full Name
        </p>
    </footer>
<!-- [[ code: the script link to your external JavaScript file ]] -->
</body>
</html>
```

(10 Marks)

Page 1 (Home Page): index.html

1. You will create **a web form** to ask the user to enter the following information using the full appropriate HTML5 form controls (elements/fields) to represent each item (label) in the list below:
    - **First Name →** input field of type text
    - **Last Name →** input field of type text
    - **Email Address →** input field of type email
    - **Address →** input field of type text
    - **City →** input field of type text
    - **Province →** input field of type text
    - **Radio buttons** with the following options (one of these options should be selected as default, usually the first one):

        ● **Premium Membership**
        ○ **Standard Membership**
        ○ **Basic Membership**

    - **Send button** to **call (trigger)** a **function named "userForm()",** which you will have to create to take/receive all the form information (user's input)
    - **Cancel button** to reset the form elements
2. The **userForm() function** will take/receive all these information using appropriate DOM methods to print them all into the same page *(below the form element)* using DOM also. This function has to be linked to a click event in the send button by using **(Even Handler ☹)** or **(Event listener 😉)**. The result which is the output of this function will be something like this:

    Full Name: Martin Smith
    Email: martinsmith2018@yahoo.ca
    Address: 123 Yonge St.
    Toronto, On
    Membership: Standard

    Display the word "**Premium**" if the user checked the Premium Membership radio button
    Or: **Standard** if the Standard Membership is checked
    Or: **Basic** for the Basic Membership is checked

Notes:

- You can use any kind of layout or HTML elements you prefer like table, div, p, span, etc. and any style. within (inside) the document.write() method in order to view the user information in nice readable format (up to you).
- Notice that you need to **concatenate** first and last name with a space in between to display both in one line as a full name

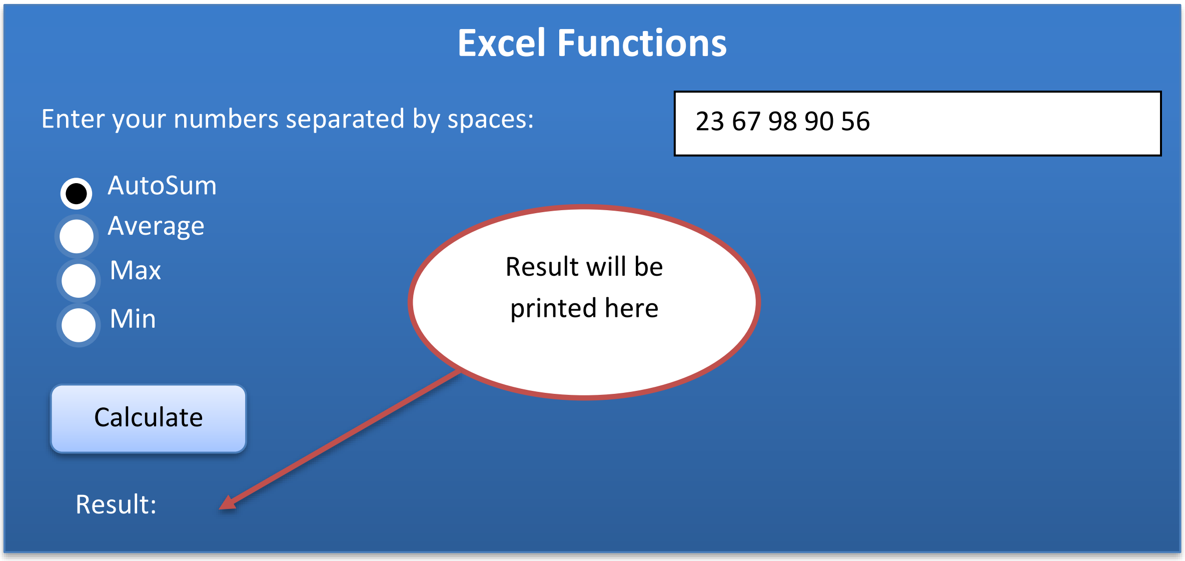
Hints:

- To have all these contents printed in your page, you need to have placeholder(s) in your HTML page like div or span or any element you prefer in order to be targeted using the DOM methods of JavaScript
- You can save each value (user input) into a separate variable (or you can save all of them into a single variable using array data structure)
- After clicking the "submit button" and finishing your code, you can use the "**innerHTML" or "innerText" property** with (getElementsById or ByTagName or ByClassName) to place the result

(40 Marks)

Page 2 (Excel Page): excel.html

1. Design a web form following the wireframe guidelines below. Notice that the image below is just a sample so you can use any styles like colour, font, layout, width, or borders plus any HTML Tags that you prefer:



Excel Functions

Enter your numbers separated by spaces: 23 67 98 90 56

● AutoSum
○ Average
○ Max
○ Min

Calculate

Result:

3. User can type their numbers with a space between each number (**as a string with the space character as a delimiter or space separated numeric values**) inside an input box (You need to notify the user to enter the numbers with a space between every two numbers as shown in the diagram)

4. Then the user can choose only one of the basic excel functions (radio buttons) to do the required calculation

Note: AutoSum radio button should be always selected as default (as a radio button)

5. When the button "**Calculate**" is clicked, a **function** named **myExcelFuns()** will be loaded (triggered). you will need to add:

- **either "onClick event"** to the form button "Calculate" [The classical way ☹]
- **or** adding/attaching **"event listener"** to the submit button [The modern way 😊]

Note: For the button "Calculate", you can use type="button" or type="submit", please refer to my lecture for more information

6. **Inside myExcelFuns() function body (block)** you will have to do the following:

a. Get the user input **(string of numbers and spaces)** and save it into a **variable** named "**numberStr**"

Hint:

You can give the **input textbox** an **id** like: **id="numbers"** to be used by JavaScript to get the values of the input box. Then you can use the DOM: **document.getElementById("numbers ").value** where "**numbers**" is the value of the id attribute that you had given to the input

b. Check first that the user has inserted some values by scanning the variable **numberStr**, if this variable is empty/null you will need to alert the user to type their numbers else you can continue with the rest of your code

Hint:

You can use **"alert()"** the JavaScript built-in function with any descriptive text to notify the user

The logic for this task:

if user didn't insert any value (completely empty) ➔ output/alert a warning/error message

else ➔ continue writing your code to do the required calculation based on the user selection, ***which means that all your logic (the main part of you code) will be inside the "else block"*** as shown below:

```
If (user didn't input any value) {
        Output/alert: your warning message
} else {
        Your code which is the main part of the answer goes here (starting from the next point c)
}
```

c. Make sure to **remove all the extra spaces before and after** the user input (Remove whitespace from both sides of the string) using the appropriate **JavaScript built-in method** for string object

Hint:

To **remove any extra spaces before and after** the input string, you can use the JavaScript built-in method **trim()** for string object

d. Convert the string user input that is saved in "**numberStr"** into an **array** of numbers named "**numberArr**" using the appropriate **JavaScript built-in method** for array object

Hint:

To **convert a string into an array**, you can use the JavaScript built-in method **split()** for string object. Notice that the delimiter parameter for this function should by only a space based on the user's input

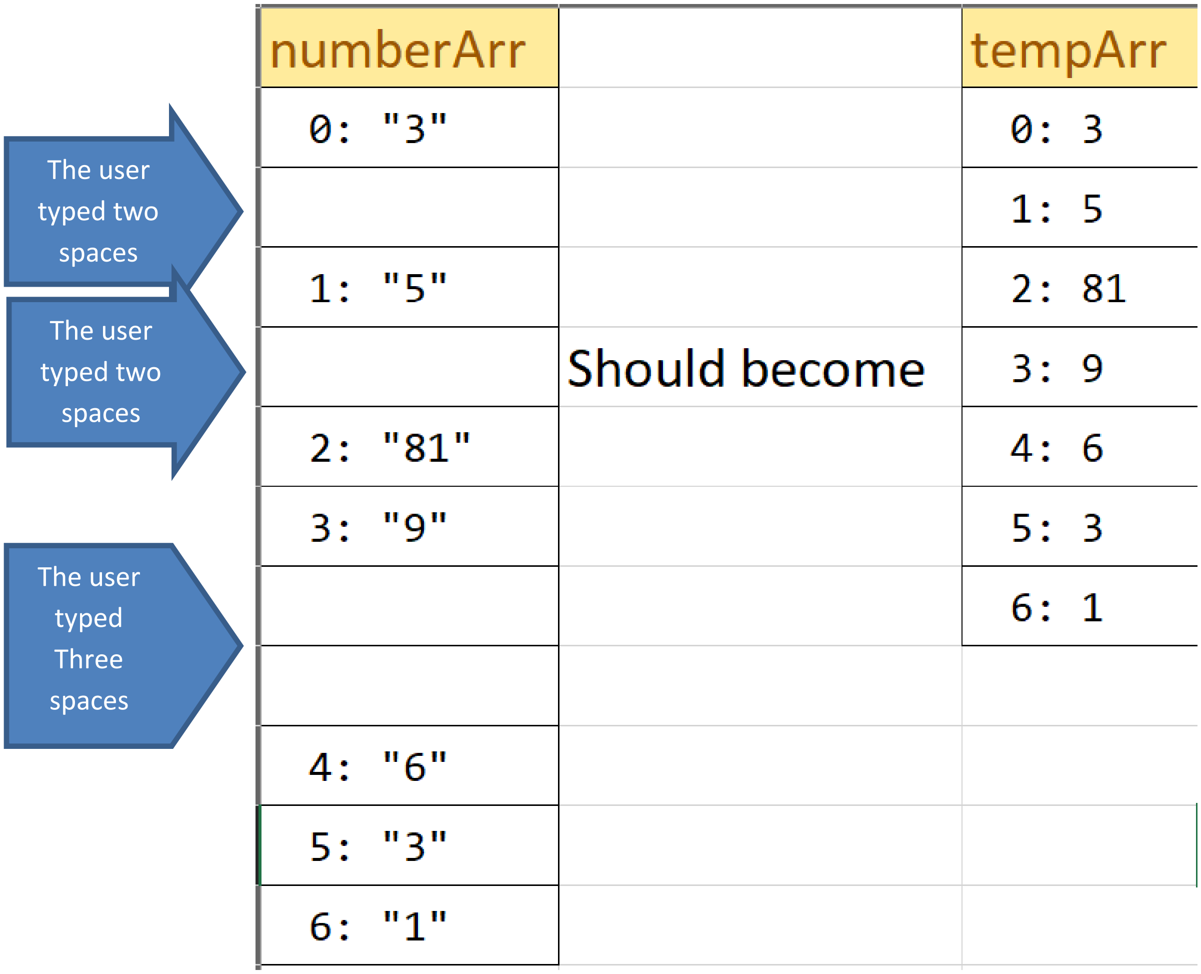
e. To run your formulas against the array, it's strongly recommended to create a new array on the fly out of the original "numberArr" using a loop with the following listed conditions. You can pick any name you prefer for the new generated array like **"finalNumericArray" This new array must contain only numbers as numeric data types and no spaces:**

- Remove any spaces in between the numbers in case if there is any (this can happen if the user type more than one space between any two numbers
- Converting each numeric value of string data type in "numberArr" to be number data type

This step is very important to make sure that the final array "**finalNumuericArray**" contains numbers only and with no spaces. In such case **you can find the total, the max, the min, and the length of the array easier.**

Hints:

Notice in the image below the "numberArr" has some spaces in between numbers. Although we used split() method of string object to convert this string into an array, **but this function works if there is only one space as a delimiter between every two numbers.** If the user typed more than one space, **split() method** will consider the first space as the delimiter and the next ones as element of the new generated array.



The user typed two spaces

The user typed two spaces

The user typed Three spaces

numberArr		tempArr
0: "3"		0: 3
		1: 5
1: "5"		2: 81
	Should become	3: 9
2: "81"		4: 6
3: "9"		5: 3
		6: 1
4: "6"		
5: "3"		
6: "1"		

To avoid this problem, you can use a for loop to loop through the array with if condition to create the new array with numbers only (You can refer to my in-class examples of using DOM for this step):

If the current element of the array is a Number AND NOT empty/space ➔ Push it to the new array

By the end of this loop => we will have the new generated array without any spaces and all the values are numbers with numeric data type (Please refer to my code examples for more clarification)

f. You will use **if condition** to **scan/check** which **radio button** was selected/checked and based on this selection you will do the required calculations

g. Use **if else blocks** to do the **required calculation** based on the selected **radio button** as explained:
If (the AutoSum radio is checked) ➔ Do the calculation to find **the total** of all the numbers
else if (the Average radio is checked) ➔ Do the calculation to find **the average** of the numbers
else if (the Max radio is checked) ➔ Do the calculation to find the **maximum** number
else ➔ Do the calculation to find the **minimum** number

h. The returned value or the final calculation result for each if condition has to be saved into a **global variable** based on "myExcelFuns()" function (it's a **local variable** based on the entire JS file) named "**result**", so you will need to declare this empty variable first, then use it inside each if condition and else block

Notes:

- Please notice that the reason behind creating/declaring one variable named "result", for example, before staring your if/else block is to used to receive the final answer for any calculation so you can use only one single statement "getElementById()" to print/output the final result in one line of code
- You can just declare this variable without any initial value ("undefined"), or you can set its initial value to 0. In both cases its value should be changed through one of the conditions

i. At the end of all the if else blocks (4 blocks), use the getElementById() method to print your result (the value of the result variable) into your specific HTML element (based on your choice)

Hints:

- There is more than one way to get the selected radio button option, one of them, for example, is to give each radio button control an "**id**" in your HTML code as shown below:

```
<input type="radio" name="excel" id="sum">AutoSum
<input type="radio" name="excel" id="avg">Average
<input type="radio" name="excel" id="max">Maximum
<input type="radio" name="excel" id="min">Minimum
```

- To check which option (radio button) was selected, you can use the DOM with if condition:
  (Please refer to our class example for more clarification: forms lesson folder ➔ dom form html files):
  - **if** (**document.getElementById("sum ").checked**) then ➔ **result** = the returned value from the calculation*
    - calculation* ➔ write the full code for finding the total of all the numbers in the array **"numberArr"**
    - You can use the same variable "result" to represent the total container variable or you can create a new variable named "total" then at the end save the value of total into our variable "result"
    - Please refer to our code examples about shopping list to review how to find the total of all numeric values in any array.
  - **else if** (**document.getElementById("avg ").checked**) then ➔ **result** = the returned value from the calculation
    - Notice here you will need to use the same logic as the previous option which is to find the total for all the numeric values in the array first
    - Then use the same variable "result" to get the result of the average formula (which is the total of the number divided by how many number)

  - **else if** (**document.getElementById("max ").checked**) then ➔ **result** = the maximum number in the array
  - **else if** (**document.getElementById("min ").checked**) then (Or just else)➔ **result** = the minimum number in the array

- As mentioned above, **inside each if condition**, you need to save the returned value (the final answer) from the calculation into a variable named "**result**". It's recommended to create the **"result"** variable before the if/else conditions (without any value) then use this variable inside each condition

Hints:

To find the maximum and the minimum number using any from the following:

- Use **max()/min()** methods for the Built-in-in JavaScript **Math constructor**
  Hint: Visit: **https://developer.mozilla.org** and search for these **two methods of the Math constructor**:
  - **Math.max()**
  - **Math.min()**
- Use the **array method "sort()"** that we covered in our lecture (working with array's methods). This method will sort the array ascendingly (A-Z) but it works fine with text values **ONLY**, not numbers. To solve the issue with numbers you will need to use an anonymous function for comparing two values (a and b) with the sort() method to sort all the numeric values ascendingly. The value in index 0 will be the minimum and the value in the last index will be the maximum (the array's length – 1 ➔ the last index), or you can chain the two functions: **sort() then reverse()** together so the maximum value will be on index 0
  Hint: Visit: **https://developer.mozilla.org** and search for the method **sort() to learn about how can you sort an array of numeric values as it requires another piece of code to be added (You will need to read the article and the code example)**
- Use any other way or logic of coding you prefer (the idea of this lab-work is to learn and practice JS)

Notes:

- You can try to use the **modern way** of adding the event to our html code using **"Event Listener"** instead of **"Event handler"**, as mentioned in the instructions, it's up to you.
- You might need to use the syntax return false at the end of your function to prevent the page from being reloaded and hiding the result but it depends on your code and which approach you want to follow

(45 Marks)

My Code Links:

- DOM ➔ https://github.com/anmarjarjees/javascript-class/tree/master/lesson09-dom
- Events ➔ https://github.com/anmarjarjees/javascript-class/tree/master/lesson10-events
- From Interaction ➔ https://github.com/anmarjarjees/fssd-javascript/blob/master/lesson11-forms/c.form-interacting.html
- Good Quick Tutorial ➔ https://www.tutorialrepublic.com/javascript-tutorial/javascript-event-listeners.php

Happy Coding 😊