Lab 5: Working with Arrays and Loops [JavaScript ES6]

Learning Outcomes:

- Apply ES6 JavaScript as a front-end scripting language.
- Understand how to create and work on Arrays and Objects.
- Understand various loops in JavaScript.

Instructions:

- For this lab, you will be learning how to work with arrays and various loops in JavaScript.
  1. Create an HTML file (i.e., **index.html**) with a simple form element, and a separate JS file (e.g., **script.js**).
  2. Create an input field with type text and a button saying '**Calculate**'
  3. The input button should take in numbers separated with a comma(','). e.g., `20,31,1,55,6`
  4. The input values should be saved in an array and using all the loops (e.g., for, forEach and map) calculate the total number of prime numbers, composite numbers and neither (e.g., `1`). for e.g.:

     ```
     [20,31,1,55,6]

     Prime Numbers: 1

     Composite Numbers: 3

     Neither: 1
     ```

  5. Display the count of prime numbers, composite numbers and neither (e.g., `1`) on the DOM in a `<span>` or `<p>` tag.

  Note: Display count results for different loops in separate elements (<p> or tags).

- For this lab, you may choose to use any JavaScript built-in function.

  Note: If you do choose to use any of JS' built-in functions, please ensure you document which built-in function you are using and why.

- As in previous labs and as specified in the **Submission Section**, for Lab 5 you will be expected to submit a README file, a Git Repo, and a remotely accessible lab on Timberlea. ***See Brightspace Lab 5 module.***
- Ensure you have set the proper file and folder permissions for your Lab 5 work.

  Note: For your files to be accessible through a browser for testing and grading, you must ensure you are using the correct file permission settings on your files and folders. On a shared server, such as Timberlea, it is recommended to **use '755' (i.e., rwxr-xr-x) on folders**, and **'644' (i.e., rw-r--r--) on individual files**. You can set your file permissions easily through an FTP client by right clicking on the file or folder you want to set specific permission settings. Depending on your FTP client, you will need to click on **'Get Info'** or **'File Permissions'**. Once on the file permissions window, you can simply enter the numeric value described above.

- Visit **https://web.cs.dal.ca/~yourCSID/csci3172/lab5/** on any browser and ensure you can view your work.

  Note: Failure to submit your work through Timberlea will result in a grade of **ZERO (0).** Failure to ensure your work is remotely accessible through a web browser, using the specified URL will result in a grade of **ZERO (0).** If you can see your work through the specified URL, then the TAs and Instructor will also be able to view and mark it.

- Regarding the look-and-feel of your assignment, you have complete creative freedom for this assignment. You are encouraged to work towards an aesthetically pleasing website that applies the design and development principles you have learned thus far in your academic and/or web development career. You may use Creative Commons images and/or logos with proper author attribution (provided through code comments, and/or **README.txt** file).

  Note: Do keep in mind that as part of this assignment, you are expected to work individually, you may discuss ideas with your classmates, but do refrain from sharing any code.

- **Include in your README.txt file,** the URL from which your lab can be accessed. All pages you develop for this assignment will need to be accessible through that link.

  Note: If you decide to use and modify any existing code, e.g., code found on online or printed sources or code used during in-class/tutorial examples, you are expected to provide author attribution in your code comments, and a more detail explanation of your sources in your README file (i.e., providing an explanation of why the piece of code is necessary for your work, where, how and why the code or section of code was modified). Keep in mind that simply stating "code was modified" does not provide sufficient information required in your programming assignments.

Submission:

- For this lab, you will need to **submit your work through Timberlea, Brightspace, and GITLab as follows:**

Submitting your Work through Timberlea

- As part of this lab, you will need to create a **'lab5'** directory inside of your **'csci3172** directory on Timberlea. **See Lab 1 instructions** *on how to log onto Timberlea using an FileZilla, and create directories.*
- Once you have completed your lab, upload your work into your **'lab5'** directory on Timberlea.

Note: You will need to ensure your submission includes all required files needed for your Lab 5 (i.e., image files, stylesheets, folders), and that your new directory and individual files have the correct **folder permissions**(i.e., **755**) and **file permissions** applied (i.e., **644**), respectively.

- Your Lab 5 submission will be expected to follow proper folder structure, i.e., images should be inside an 'image', 'images', or 'img' folder, CSS stylesheets should be inside a 'styles' or 'css' folder, and JS scripts should be inside a 'script' or 'js' folder.
- Ensure you have set the proper file and folder permissions for your Lab 5 work.

Note: In order for your files to be accessible through a browser for testing and grading, you must ensure you are using the correct file permission settings on your files and folders. On a shared server, such as Timberlea, it is recommended to **use '755' (i.e., rwxr-xr-x) on folders**, and **'644' (i.e., rw-r--r--) on individual files**. You can set your file permissions easily through an FTP client by right clicking on the file or folder you want to set specific permission settings. Depending on your FTP client, you will need to click on **'Get Info'** or **'File Permissions'**. Once on the file permissions window, you can simply enter the numeric value described above.

- Visit **https://web.cs.dal.ca/~yourcsusername/csci3172/lab5/** on any browser and ensure you can view your work.

Note: Failure to submit your work through Timberlea will result in a grade of **ZERO (0)**. Failure to ensure your work is remotely accessible through a web browser, using the specified URL will result in a grade of **ZERO (0)**.

- No validation is required for this lab as JS cannot be validated as HTML and CSS can be.
- Test your lab to ensure cross-browser compatibility. In this case, you are looking for your functions to be consistent across browsers.

Submitting your Work through Brightspace

- Download the **README template** available on Brightspace. *See Resources section on left-hand side menu on Brightspace.* There are TWO versions of this template, you may use whichever you feel more comfortable with.
- Edit the README template to include any citations for your code and/or images used for this Lab.

Note: If the work you are submitting as part of your Lab is work done by you without the use of any external sources, then please specify so within your README file.

- Depending on the version of the template you chose, rename your README file as: **L#_LastName_FirstName_README.md OR L#_LastName_FirstName_README.txt**

Note: Ensure your README file includes the URL to your Lab for remote access.

Submitting your Work through GitLab

- Create a **git** repository on the **FCS Gitlab site**, and clone it to your local system using the following command:

```
git clone *your repo https link*
```

- Copy the HTML or JS file to the local copy of your repo and push it to the git repo using the following commands:

```
git add .
git commit -m "your commit message"
git push
```

- Setup your GITlab repo as a private project and add the course **Teaching Assistants (TAs) and Instructor as 'Maintainers'** to your project, using their **CS IDs.** *See Lab 1 Brightspace module.*

Note: The CSIDs for this course will be provided during our lab session. Failure to add the course CS ID as 'Maintainer' for your work on GitLab will result in a maximum possible grade of 50%.

Marking Rubric:

The following grading criteria will be used for marking your lab:

Dimensions	Does Not Meet Expectations	Meets Expectations	Exceeds Expectations
HTML (5%)	Student's index file is empty or does not render properly or at all. **(0 - 1 points)**		Student's index file properly includes the use of a form as specified in the instructions. **(5 points)**
Design (5%)	Student's overall design is not consistent, or aesthetically pleasing. Design includes broken elements, e.g., images, links. **(0 - 1 points)**	Student's overall design is somewhat consistent. Includes some broken elements but it is not an overall distracting or confusing design. **(2 - 3 points)**	Student's overall design is consistent and aesthetically pleasing. The implemented design is overall very easy to use. **(5 points)**
For loop Implementation (30%)	Array is not traversed using a 'for' loop. **(0 - 5 points)**	Array is traversed properly using 'for' loop but logic for finding count of prime numbers and composite numbers is not correct. **(10 - 20 points)**	Array is traversed properly using a 'for' loop and the logic for finding count of prime numbers and composite numbers is correct. **(25 – 30 points)**
ForEach loop implementation (30%)	Array is not traversed using a 'forEach' loop. **(0 - 5 points)**	Array is traversed properly using 'forEach' loop but logic for finding count of prime numbers and composite numbers is not correct. **(10 - 20 points)**	Array is traversed properly using a 'forEach' loop and the logic for finding count of prime numbers and composite numbers is correct. **(25 - 30 points)**
Map function Implementation (30%)	Array is not traversed using a 'map' function. **(0 - 5 points)**	Array is traversed properly using 'map' loop but logic for finding count of prime numbers and composite numbers is not correct. **(10 - 20 points)**	Array is traversed properly using a 'map' loop and the logic for finding count of prime numbers and composite numbers is correct. **(25 – 30 points)**
Git repository	Code is not pushed to repo, and/or TAs and Instructor not added as maintainers. **(-50 points)**		Code is properly pushed to git repo. TAs and Instructor added as maintainers. **(0 points)**
Timberlea	Code and files are not properly uploaded to Timberlea. **(-100 points)**		All files are uploaded to Timberlea with proper access provided to each file and folder. **(0 points)**