Lab 2: Starting with JavaScript ES6 [JavaScript ES6]

Learning Outcomes:

- Become comfortable with JavaScript syntax, creating variables and using operators.
- Understand the difference between var, let and const keywords in JavaScript ES6.
- Learn the scope for each declaration type.

Instructions:

- For this lab, you will be learning the basics of JavaScript. Specifically, you will be creating variables, manipulating data and printing out results in the browser's **JS Console**.
- For this lab, you will not be modifying any data on the webpage but rather working around the script tag and using the browser console to work with data. This is a common technique used by developers when debugging JavaScript code.
- For this lab, you will be expected to create a JS script (e.g., index.html or index.js) to carry out the following operations:
  - **(a)** Create a function to calculate the area of a circle with a radius of 6. Ensure you use '**`const`**' for this calculation.
  - **(b)** Create a function that implements the following arithmetic functions: addition, subtraction, multiplication, and division. Your function must accept two numbers as arguments and return the result as output.
  - **(c)** Create a function to calculate the area of a rectangle with a length of 20, and width of 15.
- Add comments to your script to answer the following questions:
  - **(a)** What is the scope of a **`const`** variable?
  - **(b)** When were the **`let`** and **`const`** keywords added to JavaScript?

Note: There is **no UI** expected from this lab submission. The functions you create for this lab are simply expected to return a value and print it in the browser's JS Console. However, if you want to create a UI and know how to access the document object and fetch data from input form/page you can go ahead and take that extra step (which will be a topic covered in later labs), but no marks will be deducted if you do not create a UI.

- As in previous labs and as specified in the **Submission Section**, for Lab 2 you will be expected to submit a README file, a Git Repo, and a remotely accessible lab on Timberlea. ***See Brightspace Lab 2 module.***
- Ensure you have set the proper file and folder permissions for your Lab 2 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 Bluenose, 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/lab2/** 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, though this lab does not require a UI, 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 **'lab2'** 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 **'lab2'** directory on Timberlea.

Note: You will need to ensure your submission includes all required files needed for your Lab 2 (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 2 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 2 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 Bluenose, 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/lab2/** 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
Area of circle (10%)	Student’s JavaScript file is empty or does not include required information to compute the area of a circle. **(0 - 3 points)**	Student’s JavaScript file contains the required information to compute the area of a circle but does not use **const** keyword. **(5 - 7 points)**	Student’s JavaScript file contains all the required information to compute the area of a circle and const is used to compute the area of the circle. **(10 points)**
Understanding the scope of ‘const’ (10%)	Student’s script (or README) does not provide an answer **(0 points)**	Student’s script (or README) file has somewhat of an answer but no examples to express their understanding **(5 - 7 points)**	Student’s script (or README) file has a clear answer and provides examples to further express their understanding. **(9 - 10 points)**
Arithmetic Calculator (40%)	Student has implemented 1 or 2 arithmetic functions / operations, and/or arithmetic functions / operations do not accept two parameters. **(0 - 10 points)**	Student has implemented 3 or 4 arithmetic operations but some arithmetic functions do not accept two parameters. **(20 - 30 points)**	Student has implemented all 4 arithmetic operations and each function accepts two parameters. **(35 - 40 points)**
Area of Rectangle (10%)	Student’s JavaScript file does not calculate the area of the rectangle. **(0 points)**	Student’s JavaScript file calculates the area of the rectangle, but function is buggy or poorly implemented. **(5 - 7 points)**	Student’s JavaScript file calculates the area of the rectangle. The function accepts length and width as its two input parameters. **(9 - 10 points)**
Introduction of LET and CONST (5%)	Student’s README file provides no answer. **(0 points)**	Student’s script (or README) file has somewhat of an answer but no indication of either year of introduction or ECMAScript version is missing. **(2 - 3 points)**	Student’s script (or README) file has a clear answer and the year of introduction and ECMAScript version is clearly stated. **(5 points)**
Git repository (15%)	Code is not pushed to repo. **(0 points)**		Code is properly pushed to git repo. TAs and Instructor added as maintainers. **(15 points)**
Timberlea (10%)	Code and files are not properly uploaded to Timberlea. **(0 points)**		All files are uploaded to Timberlea with proper access provided to each file and folder. **(10 points)**
Git repository (15%)	Code is not pushed to repo. **(0 points)**		Code is properly pushed to git repo. TAs and Instructor added as maintainers. **(15 points)**