INFO1111: Computing 1A Professionalism

2023 Semester 1

Self-Learning Report

Submission number: 1

Github link: https://github.com/JerryJJJJin/INFO1111-.git

Student name	Jerry Jin
Student ID	530477150
Topic	JavaScript *Note: This must be the same as was in your topic approval*
Levels already achieved	none
Levels in this report	A

Contents

1. Level A: Initial Understanding

1.1. Level A Demonstration

1. Embedding javascript into a webpage to be executed by the web browser
2. javascript functions, build functions and pass data through them
3. Accessing webpage elements using Document Object Model (DOM)

1.2. Learning Approach

For this self-learning topic, I approached learning by following these steps:

1. Identify my goals: my goal for the course is to demonstrate level A objectives first and foremost. This guided my path of learning as everything I came across I tried to relate to my 3 goals as stated in 1.1.

2. Doing general research: As I have never done any software development before I did not know exactly what javascript was used for. So I googled about the functionality of javascript and asked a few of my older friends regarding their experience with JS.

3. Learning the basics: I went onto Codecademy and started a free introduction to javascript course to learn the fundamentals such as syntax, data types, operators and functions.

4. Expanding my knowledge: while learning the basics, I went on youtube to watch some tutorial videos and easy follow-along videos.

5. Putting it all together: By following the youtube videos, I tried to replicate their javascript project and customized it to my liking. This helped me as while I was learning I always wondered how to visualize the javascript code or what the code represented visually.

1.3. Challenges and Difficulties

The following is a list of difficulties I experienced when learning JS:

1. Understanding Syntax: When first doing some initial research about JS, I found the example codes to be really hard to understand due to two reasons. 1, I had no knowledge about JS in general and 2, the syntax used is different to languages such as Python. For example, console.log is essentially print in Python.

2.Live server: When watching the youtube tutorial videos, most assume that you know the basics of web development such as how to visualize your code in VS code for example. It was frustrating at first as I did not know what this function was called so I could not even search on google regarding the issue.

3.Html: As websites are written in Html and JS is used for its functionality, inevitably, during the learning phase I came across some Html which I did not understand.

4.DOM was the most challenging part to learn for this level, as it modifies the webpage it sometimes is difficult to follow from which step it started or which aspect of the webpage is it changing and how is it able to do that.

1.4. Learning Sources

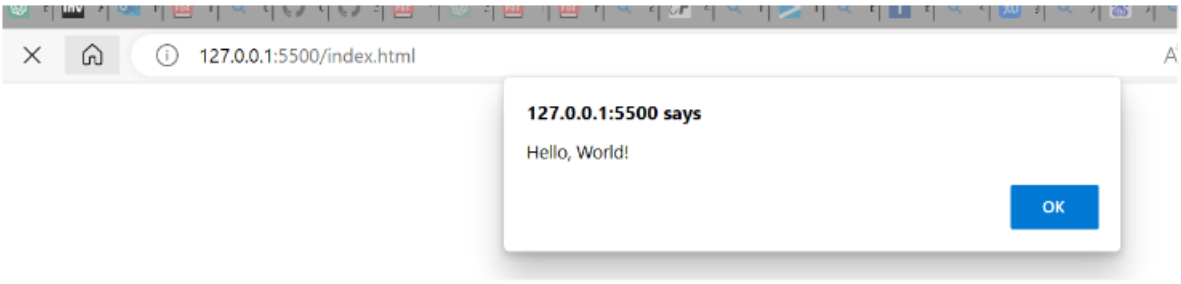
Learning Source - What source did you use? (Note: Include source details such as links to websites, videos etc.).	Contribution to Learning - How did the source contribute to your learning (i.e. what did you use the source for)?
https://www.youtube.com/watch?v=821C5aJ3SLM	Linking JS to Html
https://www.youtube.com/watch?v=W6NZfCO5SIk	basic knowledge of JS, what it is, variable, functions
https://www.youtube.com/watch?v=FOD408a0EzU	JS function
https://www.youtube.com/watch?v=DGaRuK7D92M	DOM exercise
https://www.codecademy.com/courses/introduction-to-javascript/lessons/introduction-to-javascript	introduction to JS, data types, basic syntax and built in functions

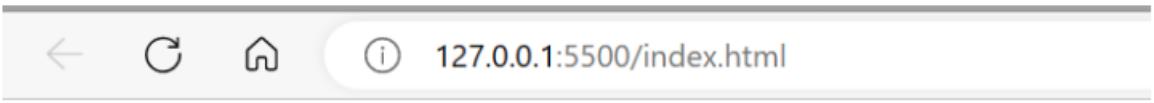
1.5. Application artifacts

Embedding javascript into a webpage to be executed by the web browser

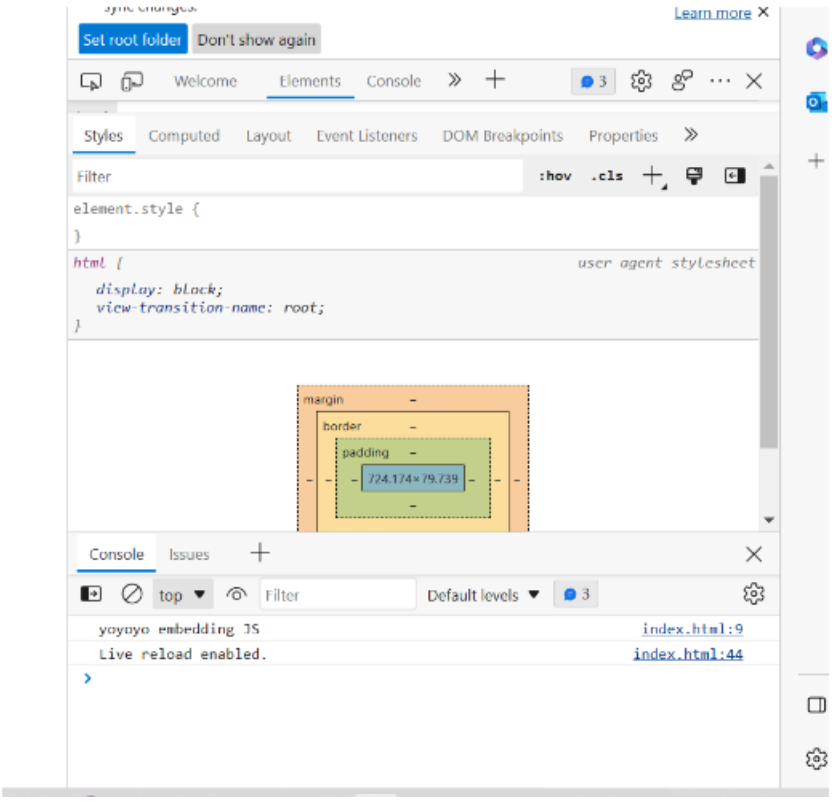
```
<!DOCTYPE html>
<html>
  <head>
    <title>Embedding javascript into a webpage to be executed by the web browser</title>
    <script src = "Hello.js">
        // inside the script tag, we can write JavaScript
    </script>
    <script>
        console.log('yoyoyo embedding JS')
    </script>
  </head>
<body>
    <h1>What's your name user!</h1>
    <h2>Nice to meet you</h2>
    <script>document.body.innerHTML = '<h1>Javascript says hello to you</h1>'</script>
</body>
</html>
```

Results





Javascript says hello to you



What does it do?

First, upon opening the webpage, an alert will pop up saying “Hello World!”
Then in the console, it will print “yoyoyo embedding JS”
Finally the webpage will display “Javascript says hello to you”

How does it work?

Generally, JS code is executed line by line hence the web browser upon loading the webpage will interpret the code from top to bottom. JS code is placed within the <script> tags of an HTML document which embeds it into the webpage for the browser to execute. However, the placement of these <script> tags within the head/body section will affect how and when the JS code is being executed.

As the head section is executed first, it can be used to run JS code before any content is loaded on to the webpage. Eg. initializing libraries or global variables. However, for demonstration purposes, I implemented the built in function alert to display a message box with an ok button to show that it is in fact executed first as the webpage is blank at that point.

<script src = "Hello.js">

// inside the script tag, we can write JavaScript

</script>

So this chunk will be executed first. It sources and run JS code from the Hello.js file which contains alert("Hello, World!"); which causes an alert window to pop up.

<script>

console.log('yoyoyo embedding JS')

</script>

Then it will execute this part which essentially prints 'yoyoyo embedding JS' into the console

<h1>What's your name user!</h1>

<h2>Nice to meet you</h2>

Then it will execute these 2 lines printing "What's your name user!" followed by "Nice to meet you" in the next line. However you will not see these two lines as →

<script>document.body.innerHTML = '<h1>Javascript says hello to

you</h1>'</script>

This code is executed next which replaces the two messages in the body and display "Javascript says hello to you" onto the webpage with h1 (heading 1) font size.

How did I create it?

Using the basic HTML file, Inside the head section, I added a script tag that referred to an external JS file "Hello.js". This script tag is using the src attribute to specify which file it is souring code from

Next, I added another script tag inside the head section that uses the console.log() function to print a message to the browser console when executed

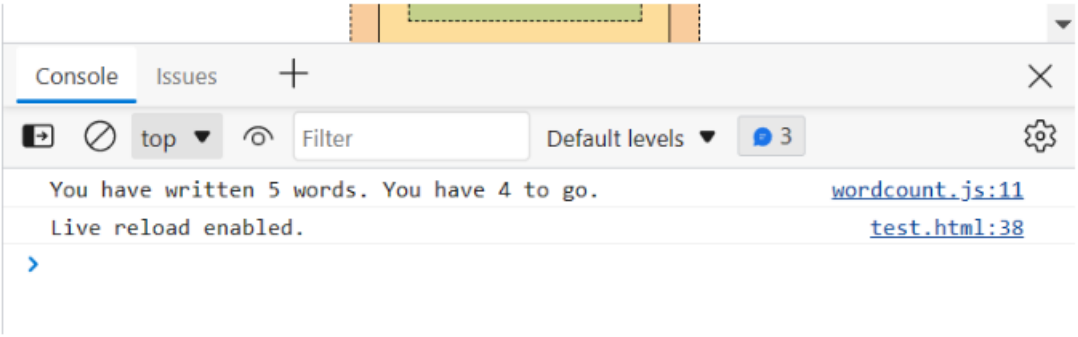
Then, I added another script tag inside the body section that uses the document.body.innerHTML to change the message of the body element to a Header message Finally opened the file with the liver server extension for visualization

Javascript functions, build functions and pass data through them

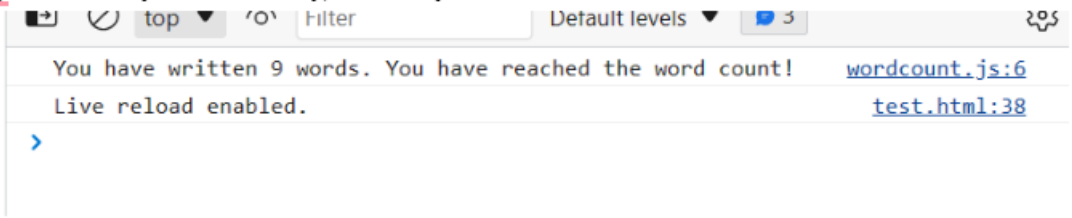
```
function wordcount(text, target) {
    let words = text.split(" ");
    let wordcount = words.length;

    if (target * 0.9 <= wordcount && wordcount <= target * 1.1) {
        console.log("You have written " + wordcount + " words. You have
reached the word count!");
    }

    else if (wordcount < target * 0.9) {
        wordsleft = target * 0.9 - wordcount;
        console.log("You have written " + wordcount + " words. You have "
+ wordsleft + " to go.");
    }

    else if (wordcount > target * 1.1) {
        wordsover = wordcount - target * 1.1;
        console.log("You have written " + wordcount + " words. You are " +
wordsover + " words over.");
    }
}

wordcount(essay, 10);
```

Results

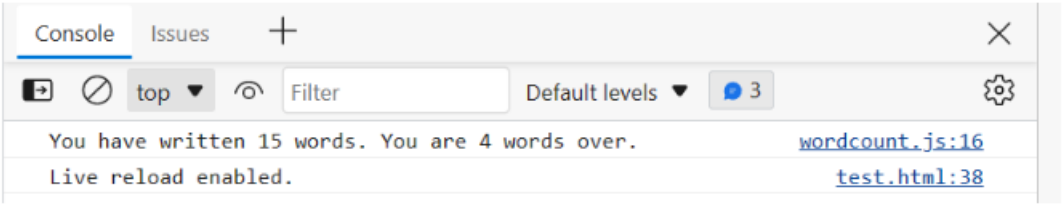
If essay = “Hello my name is Jerry!”



If essay = “Hello my name is Jerry, I am very cool!”



If essay = “Hello my name is Jerry, I love to chat and chat and chat and chat.”



What does it do?

This is a function that aims to count the number of words in an essay and if it reached a 10-word count (just for the ease of demonstration the word count is low). If the essay is within plus minus 10 percent of the word count then it is within the target. However, if the wordcount of the essay is < 9 or > 11 i.e. greater than the 10 percent range then it has not reached the 10 wordcount target.

How does it work?

function wordcount(text, target)

The function wordcount takes in two parameters: text and target

let words = text.split(" ");

let wordcount = words.length;

Then it splits the text using the split() function by removing the spaces forming an array and counts the number of words through the length function.

if (target * 0.9 <= wordcount wordcount <= target * 1.1)

else if (wordcount < target * 0.9)

else if (wordcount > target * 1.1)

Then it checks if the word count is within plus minus 10 percent of target word count using conditional statement and mathematical and logical operators.

If the word count is within the target range, it prints a message to the console indicating that it has been reached.

If the word count is less than 90 percent of the target, it calculates the number of words left to reach the target and prints a message to the console indicating how many words have been written and how many are left to reach the target.

If the word count is more than 110 percent of the target, it calculates the number of words over the target and prints a message to the console indicating how many words have been written and how many are over the target.

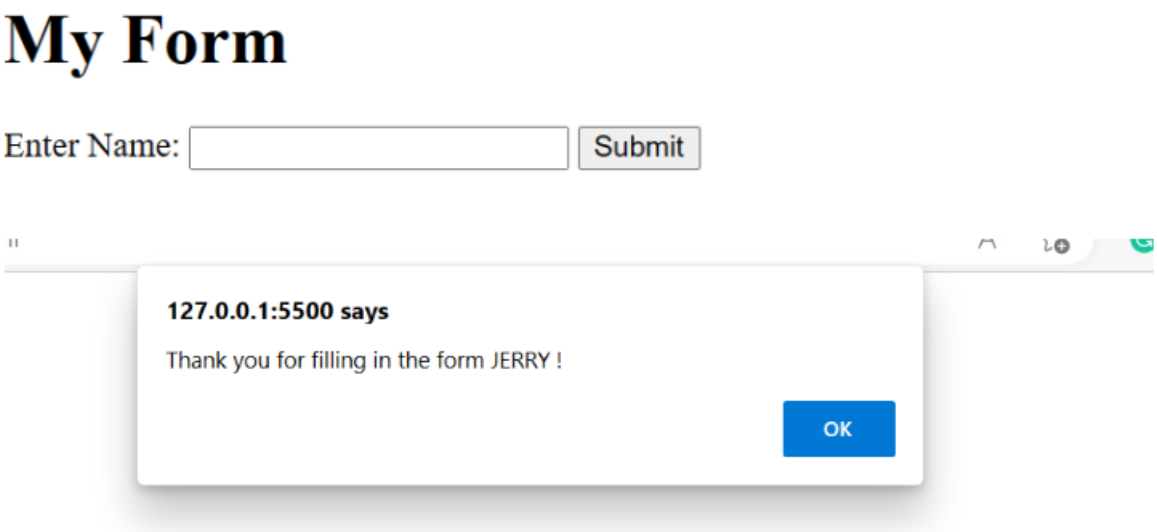
How did I create it?

1. Defining the function with parameters
2. Count the number of words
3. Write Conditional statements for the respective situations
4. Write the respective print statements
5. Call the function with essay and 10 as arguments

Accessing webpage elements using Document Object Model (DOM)

```html
<!DOCTYPE html>
<html>
<head>
    <title>My Form</title>
    <script>
        function displayGreeting() {
            let name = document.getElementById("nameInput").value;
            alert("Thank you for filling in the form " +
name.toUpperCase() + " !")
        }
    </script>
</head>
<body>
    <h1>My Form</h1>
    <label>Enter Name:</label>
    <input type="text" id="nameInput">
    <button onclick="displayGreeting()">Submit</button>
</body>
</html>
```

Results



What does it do?

The webpage has a form for which the user will enter his name. Upon clicking the submit button, the program will alert the user with a Thank you message.

How does it work

<h1>My Form</h1>

<label>Enter Name:</label>

<input type="text" id="nameInput">

This will create a Heading My Form and an empty box to enter name. Whatever the user inputs, it will be stored with an id of nameInput

<button onclick="displayGreeting()">Submit</button>

This attaches an event listener to the submit button, when clicked by the user, the displayGreeting function is called

function displayGreeting()

let name = document.getElementById("nameInput").value;

alert("Thank you for filling in the form " +

name.toUpperCase() + " !")

The displayGreeting function retrieves the name from the input element using document.getElementById("nameInput").value and displays a personalized greeting using alert().

How did I create it?

1. Write the displayGreeting function in the head section as this should be loaded first
2. Write the heading and box for the form
3. Create the variable for the user input
4. Create the submit button as an event listener to interact with the user

2. Level B: Basic Application

Whilst level A is about doing something simple with the topic to just show that you have started to be able to use the tool or technology, level B is about doing something practical that might actually be useful.

2.1. Level B Demonstration

This is a short description of the application that you have developed in order to demonstration your understanding. (50-100 words).

2.2. Application artifacts

Include here a description of what you actually created (what does it do? How does it work? How did you create it?). Include any code or other related artefacts that you created (these should also be included in your github repository).

If you do include screengrabs to show what you have done then these should be annotated to explain what it is showing and what the application does.

3. Level C: Deeper Understanding

Level C focuses on showing that you have actually understood the tool or technology at a relatively advanced level. You will need to compare it to alternatives, identifying key strengths and weaknesses, and the areas where this tool is most effective.

3.1. Strengths

What are the key strengths of the item you have learnt? (50-100 words)

3.2. Weaknesses

What are the key weaknesses of the item you have learnt? (50-100 words)

3.3. Usefulness

Describe one scenario under which you believe the topic you have learnt could be useful. (50-100 words)

3.4. Key Question 1

Note: This question is in the table in the 'Self Learning: List of Topics' page on Canvas. (50-100 words)

3.5. Key Question 2

Note: This question is in the table in the 'Self Learning: List of Topics' page on Canvas. (50-100 words)

4. Level D: Evolution of skills

4.1. Level D Demonstration

This is a short description of the application that you have developed. (50-100 words). **IMPORTANT:** *You might wish to submit this as part of an earlier submission in order to obtain feedback as to whether this is likely to be acceptable for level D.*

4.2. Application artifacts

Include here a description of what you actually created (what does it do? How does it work? How did you create it?). Include any code or other related artefacts that you created (these should also be included in your github repository).

If you do include screengrabs to show what you have done then these should be annotated to explain what it is showing and what the application does.

4.3. Alternative tools/technologies

Identify 2 alternative tools/technologies that can be used instead of the one you studied for your topic. (e.g. if your topic was Python, then you might identify Java and Golang)

4.4. Comparative Analysis

Describe situations in which both your topic and each of the identified alternatives would be preferred over the others (100-200 words).