**Score: \_\_\_\_\_**

**WA1 – Web Abstractions and Front End: HTML**

**Activities**

COMP256 – Computing Abstractions

Dickinson College

Spring 2023

Prof. Grant Braught

**Name:**

**Introduction:**

Today’s class introduced the topic of web abstractions as the set of abstractions that bridge the gap between the user experience in visiting a site using a browser and the mechanics that make that possible. The distinction between the front-end and the back-end and some of the key technologies used in each were introduced. Finally, some of the basics of HTML were presented. Today’s activities will reinforce the ideas behind web abstractions, front-end and back-end technologies and give you a little practice with writing HTML.

They are not required viewing, but if you would like another take on short introductions to the web and HTML you can check out Jeremy in the following Web Demystified videos:

* What is "The Web" and how does it work? (3:23)
	+ <https://www.youtube.com/watch?v=O_GWbkXIqEY>
* What's HTML and how does it work? (7:16)
	+ <https://www.youtube.com/watch?v=PORRrz3Y8Vc>

**Web Abstractions:**

🔑 1. Recall that abstractions hide details that are not relevant to our task while allowing us to focus our attention to others that are. Describe some of the details that are hidden by web abstractions and some that we focus our attention on.

🔑 2. Name the three main front-end web technologies and give a sentence of your own words describing the purpose of each.

🔑 3. Name the three main back-end web technologies and give a sentence of your own words describing the purpose of each.

4. Job/Internship adds for web developers often use job titles. Research the job titles below and describe what they mean. Try to identify how they relate to the ideas of front-end and back-end as described above.

 a. UI Designer

 b. UX Designer

 c. Full-Stack Developer

**HTML Elements & The DOM:**

The following exercises will let you check your mastery of some of the terminology of HTML, practice creating some HTML elements and check your understanding of the relationship between the HTML source code and the Document Object Model.

🔑 5. Consider the HTML element below:

 <h3 title="Hover Text!">Heading with Hover Text Example</h3>

Identify each of the following parts of the above element:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | Opening (start) Tag |  |  |
|  | Closing (end) Tag |  |  |
|  | Content |  |  |
|  | Attribute |  |  |
|  | Attribute Name |  |  |
|  | Attribute Value |  |  |
|  |  |  |  |

🔑 6. Give HTML source code for properly nested elements that would be rendered the text shown below exactly as shown (i.e. the exact words with the exact formatting).

 a. Some *italic* text and some **bold** text.

 b. Some ***bold italic*** text.

 c. A mix of **bold and *bold italic*** text.

7. Give the source code for an HTML Page that would generate the following DOM. Note you may not have seen the ul or li elements before. However, if you assume that each has an opening and closing tag, you should still be able to write the source code from the given DOM.



🏆 8. Draw the tree for the Document Object Model (DOM) for the following HTML Page.

<html>

 <head>

 <title>Hello</title>

 </head>

 <body>

 <h1>Introduction</h1>

 <p>Some <em>paragraph</em>.</p>

 <h2>A Sub-Section</h2>

 <p>More <strong><em>text</em> here</strong>.</p>

 </body>

</html>

**Setting Up the Web Server:**

To help you learn about and experiment with web development technologies you will be using a Docker container similar to the others we have been using. This container has tools and servers that will help you experience some of the basics of both front-end and back-end web development.

Use the following steps to create and run the Docker container for these Web Abstractions activities:

1. Create the new container using the following command in a terminal on your machine (MacOS or WSL in windows):

docker create --name Comp256Web --publish 5901:5901 --publish 6901:6901 braughtg/comp256-web:1.0.0

 2. Using the Docker Desktop application, find the “Containers” tab.

3. On that tab, find the entry for the Comp256Web container.

4. Click the “Play” button () to start the COMP256Web container.

5. Open the TigerVNC application and connect to localhost:5901.

6. A window that looks like the other containers we have used should appear.

* Note: While this container looks just like the others, it contains different software that has been customized for the Web Abstractions unit.

7. Open a Terminal or File Manager and check that your home directory contains the api and www directories that you will be using in this and future activities.

**Creating a First Web Page:**

🔑 9. Use a text editor (e.g. MousePad or VS Codium) to create a new plain text file named FirstPage.html in the www folder in your home directory.

Copy the HTML for the first page example from class (given below) into that file and save it.

<html>

 <head>

 <title>My First Page</title>

 </head>

 <body>

 <h1>Hello World!</h1>

 <p>This is my <em>very first</em>

 <strong>HTML Page</strong>.</p>

 </body>

</html>

Once you have saved the page, open the Firefox browser in the container and visit the URL: <http://localhost:8080/www/FirstPage.html> to see the page that you created.

Now modify the text of the page so that your name will be displayed somewhere on the page and paste a screenshot of the rendered page here.

**More HTML:**

The *Mozilla Developer Network* (MDN) is a definitive source for a lot of resources that can be helpful in learning HTML.

Read the MDN *Getting Started with HTML* document linked below. This document reviews some of what we saw in class (skim that if you like). It will also introduce a few new things that we didn’t talk about but that are required to complete the exercises below:

* <https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics>

🔑 10. Give an example of improperly nested elements that is different than the one given in the MDN document.

🔑 11. What does the alt attribute of the img tag do? Why is it important?

12. What happens to the rendered HTML if you put extra spaces or line breaks between the words in the content of a paragraph (i.e. <p>) element in the HTML source code? Hint: Try modifying your FirstPage.html file from earlier to test it out.

13. Once you know a little HTML it is fairly easy to pick up more from tutorials and references. The Lists section of the MDN *Getting Started with HTML* document linked above describes HTML elements for creating ordered and unordered list.

a. Give the HTML source for an *unordered list element* that contains the names of the courses you are taking this semester as items. Be sure to test your element by modifying your FirstPage.html file from earlier.

b. Give the HTML source for an *ordered list element* that contains your top 3-5 favorite foods as items. Be sure to test your element by modifying your FirstPage.html file from earlier.

14. Use a text editor to create a new plain text file named SecondPage.html in the comp256 folder. In that file, create a complete HTML page about some topic you are interested in.

Your page must include the html, head, title, and body elements.

The body of your page must contain:

* At least two different headings with content
* At least two paragraph elements with content
* At least one list (ordered or unordered)
* At least one image
* At least one link.

a. Paste the HTML source code for your page here.

b. Paste a screenshot of your rendered html page here.

🏆 15. As mentioned earlier, once you know a little HTML it is fairly easy to pick up more from tutorials and references. Read the MDN page on *Advanced Text Formatting* and add at least two of the elements that it describes to your answer from #14. Do not add a separate answer here, instead just update your HTML source code and screen shot for #14 to include your new elements.

* <https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Advanced_text_formatting>

🏆 16. Read the MDN page on *HTML Table Basics* and add a table to your page in question #14. Do not add a separate answer here, instead just update your HTML source code and screen shot for #14 to include your table.

* <https://developer.mozilla.org/en-US/docs/Learn/HTML/Tables/Basics>

Optional: To help me improve and scope these activities for future semesters please consider providing the following feedback.

a. Approximately how much time did you spend on this activity outside of class time?

b. Please comment on any particular challenges you faced in completing this activity.