Web Programming Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Activity 2.3 – Nutrition Label

Points: \_\_\_\_\_\_\_\_\_\_ / 45

Typography is the art of styling your text to be easily readable and suit its purpose. In this activity, you'll use typography to build a nutrition label webpage. You'll learn how to style text, adjust line height, and position your text using CSS.

1. First, read this [article](https://developer.mozilla.org/en-US/docs/Web/CSS/Universal_selectors) about the universal selector. As you read, answer the following questions in your engineering notebook...

a. What elements does the universal selector effect?

b. Why might you use the universal selector?

2. Next, read this [article](https://developer.mozilla.org/en-US/docs/Web/CSS/font-weight) about the font-weight property. As you read, answer the following questions in your engineering notebook...

a. What is the smallest and largest value that font-weight can be set to? How do they change the text?

b. What value is equivalent to the setting normal? What about the setting bold?

c. How do the lighter and bolder settings change the font-weight?

3. Next, read this [article](https://blog.hubspot.com/website/html-span) about the <span> element. As you read, answer the following questions in your engineering notebook...

a. What is a <span> element?

b. When should a <span> be used?

c. What is the difference between a <span> and a <div>?

4. Next, read this [article](https://www.shecodes.io/athena/16675-how-to-add-more-than-one-class-to-an-html-element) about applying multiple classes to an element. As you read, answer the following question in your engineering notebook…

a. How do you apply two classes to an element?

4. And finally, read this [article](https://developer.mozilla.org/en-US/docs/Web/CSS/:not) about the :not() CSS pseudo-class. As you read, answer the following questions…

a. What does the :not() CSS pseudo-class represent?

b. When might you use the :not() CSS pseudo-class?

5. Download and extract the project folder from GitHub using the link provided in Google Classroom. Once extracted, rename the project folder using proper naming practices.

6. Next, open the file named *index.html*. Add a comment at the top of the document with the project name, your name, today’s date, and the title of the course on separate lines. Then open the file named *styles.css* (located in the styles folder) and add a comment at the top of the document with the same information. All content you add to the documents should be under these comments.

7. Login to [freeCodeCamp](https://www.freecodecamp.org/). Using the steps provided under [*Learn Typography by Building a Nutrition Label*](https://www.freecodecamp.org/learn/2022/responsive-web-design/), complete the website. You should complete the work in Visual Studio Code and copy and paste the solution to each step into freeCodeCamp to check your answer. The following steps have different instructions for VS Code...

a. Step 4 – When setting up your <link> elements to the different stylesheets, you will need to add one to "https://fonts.googleapis.com/css?family=Open+Sans:400,700,800". For the other <link> element, use href="styles/styles.css" instead of href="styles.css".

8. When your website is complete, please do the following…

a. Compress the project folder and submit it to Google Classroom.

b. Print a copy of *index.html* and *styles.css*. Add these to your engineering notebook. Highlight and annotate your code.

c. Take a screenshot of the finished webpage (open it in a browser and use the *Snip and Sketch* tool). Print a copy of the screenshot and add it to your engineering notebook.

|  |  |
| --- | --- |
| **Project Criteria** | |
| Criteria | Point(s) |
| <head> element contains all the necessary non-visible and visible elements to provide the necessary meta data for the website. | /5 |
| <body> element contains all the necessary visible elements to construct the nutrition label. | /25 |
| Website has the correct overall style and structure. | /5 |
| **Development Mechanics** | |
| Criteria | Point(s) |
| All elements are structured using best practices. | /2 |
| Project folder/director structure follows industry best practices. | /2 |
| All folders/directories and files use proper naming style (i.e., file-name). | /2 |
| All documents use whitespace and comments to help organize the code. | /2 |
| Project root folder/directory has an appropriate name. | /1 |
| All documents have a comment at the top that includes the program’s title, the student’s name, the date, and the course’s title. | /1 |

/ 45