Project Instructions

Project 4: Web Style Guide

Before you start:

1. Download the project files. Inside you will find an index.html file, an 'images' folder and a 'resources' folder that contains an index.html file and a 'css' folder with styles.css and normalize.css files. The styles.css will serve as a guide for creating your Sass.

2. You should already have Sass installed locally on your computer, and be comfortable running Sass commands from the terminal. For more information on installing Sass check out: Installing and Using Sass or Sass: Install Sass. For a refresher on running Sass commands: Compiling Sass to Css

Instructions:

1.) Structure your folders.

- Start by creating a scss folder inside of your project's root folder. All of your Sass files, including sub-folders to organize them will go in here.
- Create sub-folders inside of the scss folder for your base, component, and utilities partials.
- Create a css folder inside of the project's root folder. This is where your output css will be generated.

2.) Create the file structure.

- Change the **normalize.css** file into a Sass partial and save it to the base folder.
- Create a typography partial and save it to the base folder.
- Create at least 4 different component partials that group together the major sections of the site. For example: navigation, grid, form, and button partials.
- Inside of the utilities folder, create variables and mixins partials.
- In each of your scss sub-folders, create an _index.scss file. These files will be used to import individual partials from each sub-folder.
- Create a **styles.scss**. Use this file to import all of the **_index.scss** files from your sub-folders.

3.) Setup Sass to watch for changes.

- From the terminal, navigate to your project's root folder.
- Run the command sass --watch scss:css so that Sass will continuously update your output CSS as you make changes to your Sass.

4.) **Create your variables.**
- Looking through the resources/css/styles.css file, identify common colors and sizes that can be converted to variables.
    - Create at least 5 color variables.
    - Create a variable that stores the size of the media query breakpoint.

5.) **Add styles to the typography partial.**
- Scan the resources/css/styles.css file for styles related to font family, size, and weight.
- Pay particular attention to those that are targeting tag names like a, h1, body, or the universal selector *.
- Add these styles into the typography partial.

6.) **Move styles into components.**
- Transfer the remaining CSS into their respective components. For example, any class selectors that begin with grid, column, or rows could go in the grid component.
- Be sure to update any styles that use color values with the Sass variables you created in the previous step.

7.) **Check your work**
- Double check all of your work, click on the "How you will be graded" tab above and compare your project to the rubrics listed there. Post your project in the #review-my-project channel so that the rest of the community can see your awesome project and give feedback before submitting.

EXTRA CREDIT

1.) **Create a media query mixin**
- Create a mixin that creates a media query inside of the selector that calls the mixin.
    - The mixin should accept a single parameter for the size of the media query breakpoint.
    - The mixin should create a min-width media query only.

2.) **Use the media query mixin for all media queries.**

3.) **Use a nested selector structure for at least one component.**
- Create a base selector for a group of related styles.
- Inside of that selector, use the Sass parent selector: & to append a child class selector to the base. An example would be in the navigation component use .nav as the base selector, and nest .nav-link inside of it.

4.) Use built in Sass function to add hover effect.

- Created a hover effect for buttons.
- For the effect, lighten the button's background color by 15%.
- Add a CSS transition so that the color change fades in.