KIEI-451: INTRODUCTION TO SOFTWARE DEVELOPMENT

Week 1 Homework Instructions

Let's continue practicing building websites with HTML and pushing the completed code up to GitHub.

Your task is to build a website for marketing a business – real or fictional. For this week's assignment, the look-and-feel of the website doesn't matter, only the structure. You will be adding the design/look of the website in next week's assignment (by adding CSS).

1. The website should be for a business – it can be real or fictional, a startup or an established business.
2. Create a new repository in your GitHub.com account. To make it easier for us to find, the repository should be named ***hw1***. It should be a **public** repository and your code should ultimately live at `https://github.com/<your GitHub username>/hw1` . **Submit this URL as your "website URL" for this assignment in Canvas.**
3. Clone this repository using VSCode. Create at least 3 HTML pages, connected via links; i.e. a consumer of your website should not need to know the file structure, rather, they can simply navigate between the pages using links. The "home page" should be named `index.html` , and you should follow the *HTML Guidelines* as listed below. The other two pages can be for whatever makes sense – e.g. an "about us" page, "products" page, "benefits" page, etc.
4. Create a file called `README.md` in the root of your project's code. In the contents of this file, write a quick description of your project. You'll know you did this right if your GitHub repository shows the contents of this file directly below the files.

Linking the pages

If all goes well, your directory structure should resemble something like this:

```
Your code folder
- hw1
-- index.html
-- page2.html
-- page3.html
```

The "home page" (i.e. what visitors will see when they go to your production site) **must be named** `index.html` , and is the only filename that really matters. You can name the other files whatever you want.

Creating links between the pages is as simple as `<a href="page2.html">My second page</a>` . Don't use "absolute" paths, for instance `<a href="C:\\code\username.github.io\page2.html">My` `second page</a>` – just use the filename by itself. This is known as the "relative" path and works great, as long as the files are all in the same directory.

HTML Guidelines

You should use these elements at least once, and use proper indentation and spacing:

1. Headers (h1, h2, etc)
2. p
3. ul/li or table
4. a
5. img

In addition, your `head` and `body` should contain the correct elements. For instance, the "meat" of your HTML should reside within the `body`, and your `head` should only contain things like the page title, meta tags, and other elements that are not on the actual page.

Rubric (10 points total)

- 5 points: 3 distinct pages, linked by common navigation, and the links work on any computer, not just yours.
- 5 points: HTML Guidelines implemented.

Again, be sure to **submit your Github repository's URL via Canvas - this is the ONLY way we will know what your Github username is and where your finished code lives!**

Happy coding!