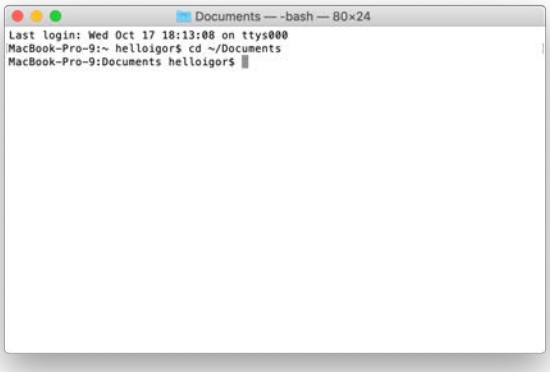
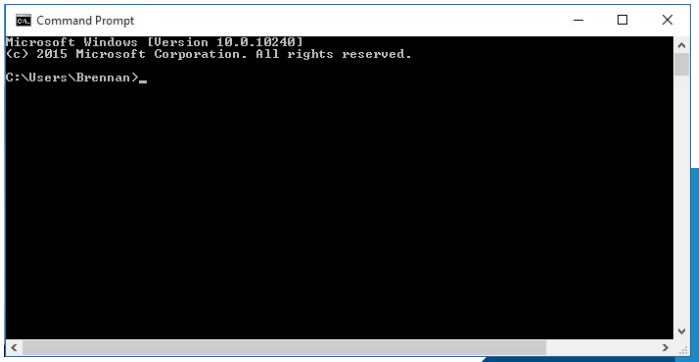
In-Class Exercise/Demo

In-Class Exercise / Demo

Terminal / Command Prompt

- The first thing we want to do is practice using the Command Line Interface or CLI. We do this by using the Terminal on mac or Command Prompt on windows. Terminal or Command Prompt is the interface to the underlying operating system of your computer.
- Launch terminal or command prompt on your computer. You can either do a search on your computer or open through the application folder. On the mac, it is located inside the utilities folder within the application folder. They look something this – terminal (left), command prompt (right):





In-Class Exercise / Demo

Terminal / Command Prompt

- Let's navigate to where we want to keep our project files.
- enter: **cd desktop** *(cd: change directory)*
- You should see something like this:

```
Last login: Tue Aug 31 15:30:15 on ttys001
spritevsion@MacBook-Pro ~ % cd desktop
spritevsion@MacBook-Pro desktop %
```

- Next, enter: **cd WESTCLIFF** - which will take you inside WESTCLIFF folder

In-Class Exercise / Demo

Github

- Let's Get Started with GitHub next!

Who this tutorial is for:

This tutorial assumes no prior knowledge and is suitable for complete beginners as a first project.

What you need:

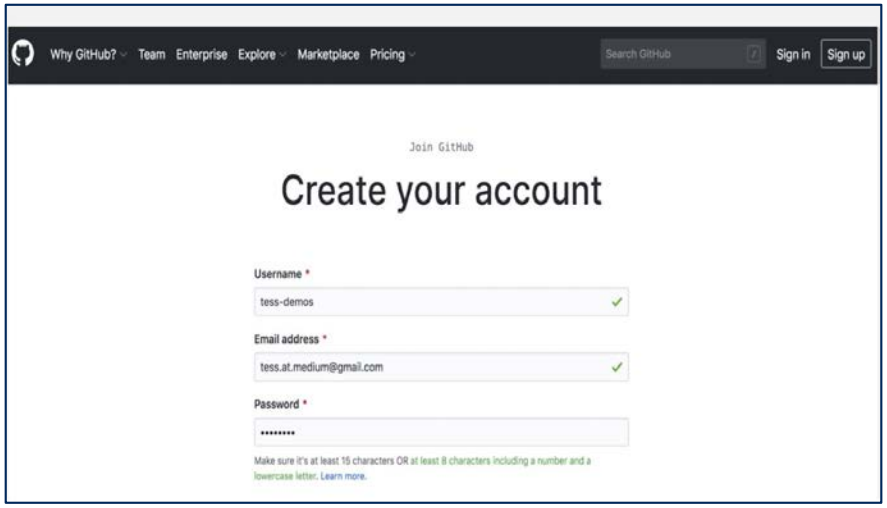
1. a GitHub account (if you already have one set up, skip step 1)
2. a code editor (Visual Studio Code)
3. terminal or command prompt
4. approximately 1 hour

In-Class Exercise / Demo

Github

Sign up for a GitHub account.

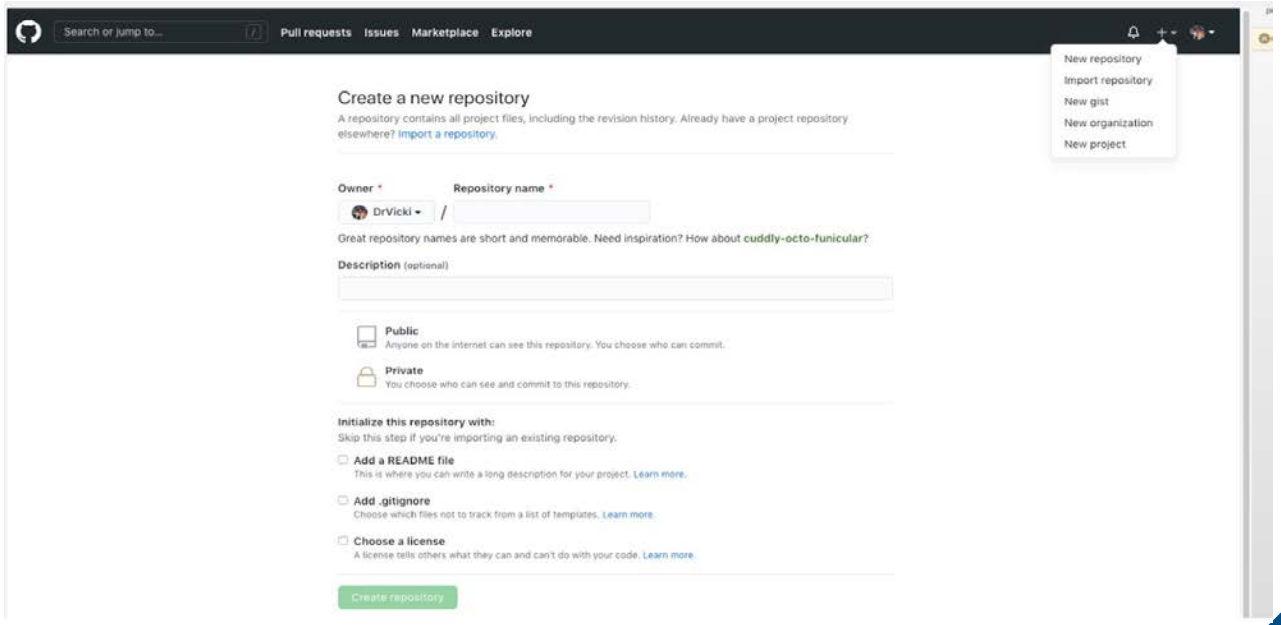
What exactly is GitHub? The Git in GitHub is a version control system, so every time anything changes in our code, the change is tracked. This lets you trace everything you've ever written and changed within a project and revert back to an old version of your code if you need to. Git on its own is a command-line tool. GitHub is where it all comes together. It's where we can store our projects, track all our work and code changes, as well as network with other developers (and check out their projects!).



In-Class Exercise / Demo

Github

Step 1: Create new GitHub Repository. Name it **week1-day1**

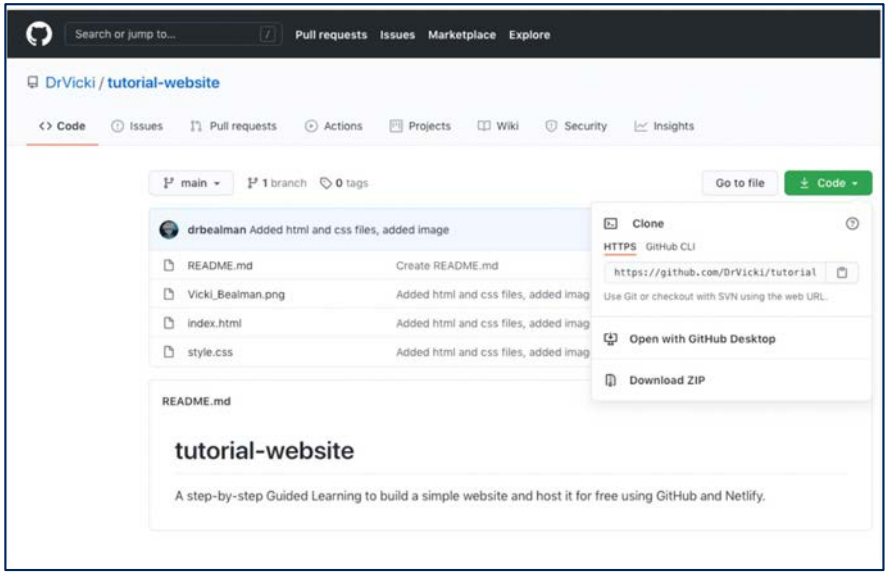


In-Class Exercise / Demo

Github

Step 2: Clone your Repository

Click on “Clone or download” and copy the HTTPS URL



In-Class Exercise / Demo

Terminal / Command Prompt

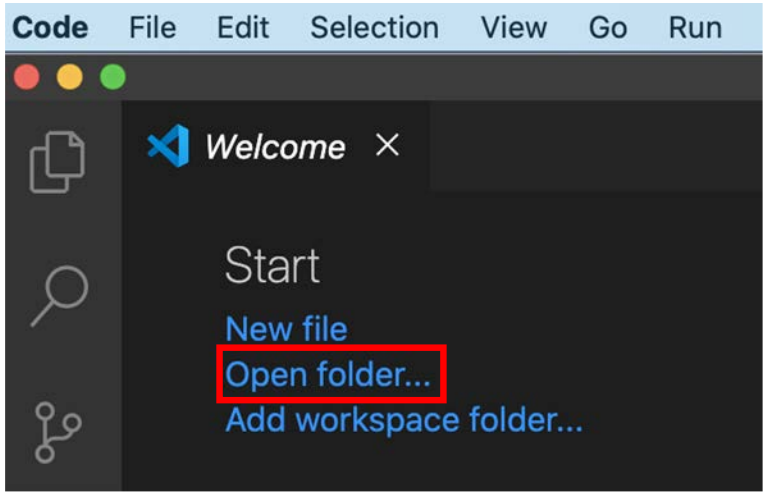
- Return to terminal or command prompt (check to make sure you are still in WESTCLIFF directory).

  [Do this step if you are windows user: install git on your computer (https://git-scm.com/download/win)]
- Enter: **git clone <HTTPS URL from your GitHub repository>**
- Next, enter: **cd week1-day1**
- To see what's inside this folder, enter: **ls** *(ls: list)*
- It should display only 1 file at this time: *README.md*
- Let's create a file. Enter: **touch test.html** (mac) **echo test.html** (windows)
- Enter: **ls** - it should now display 2 files: *README.md* and *test.html*
- We are now done with terminal/command prompt exercise.

In-Class Exercise / Demo

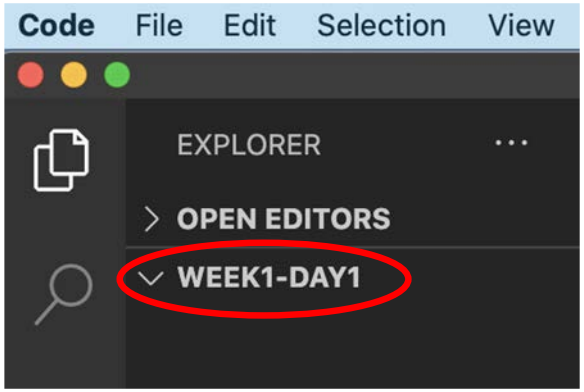
Initial Setup

- Launch Visual Studio Code on your computer.
- Select Open folder from the Start menu:



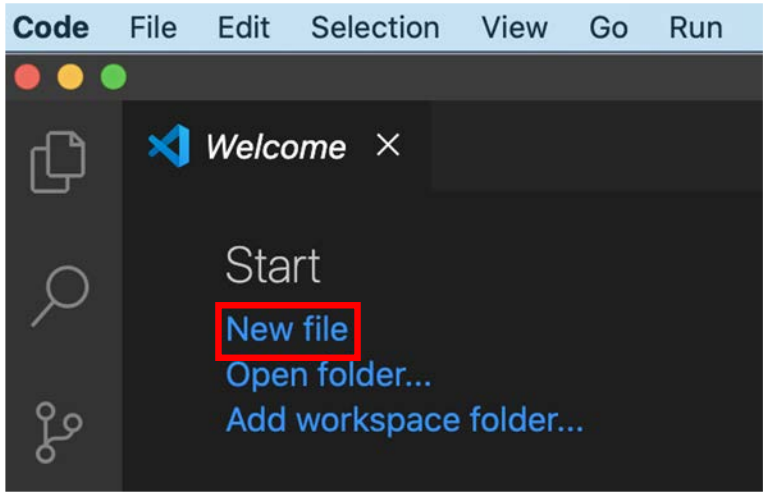
In-Class Exercise / Demo
Initial Setup

- Browse to the WESTCLIFF folder and select the *week1-day1* folder inside it.
- On the explorer (left side bar), it should look something like this:

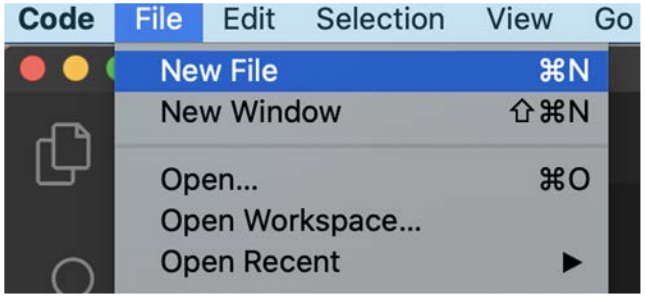


In-Class Exercise / Demo
Initial Setup

- Create a new file:



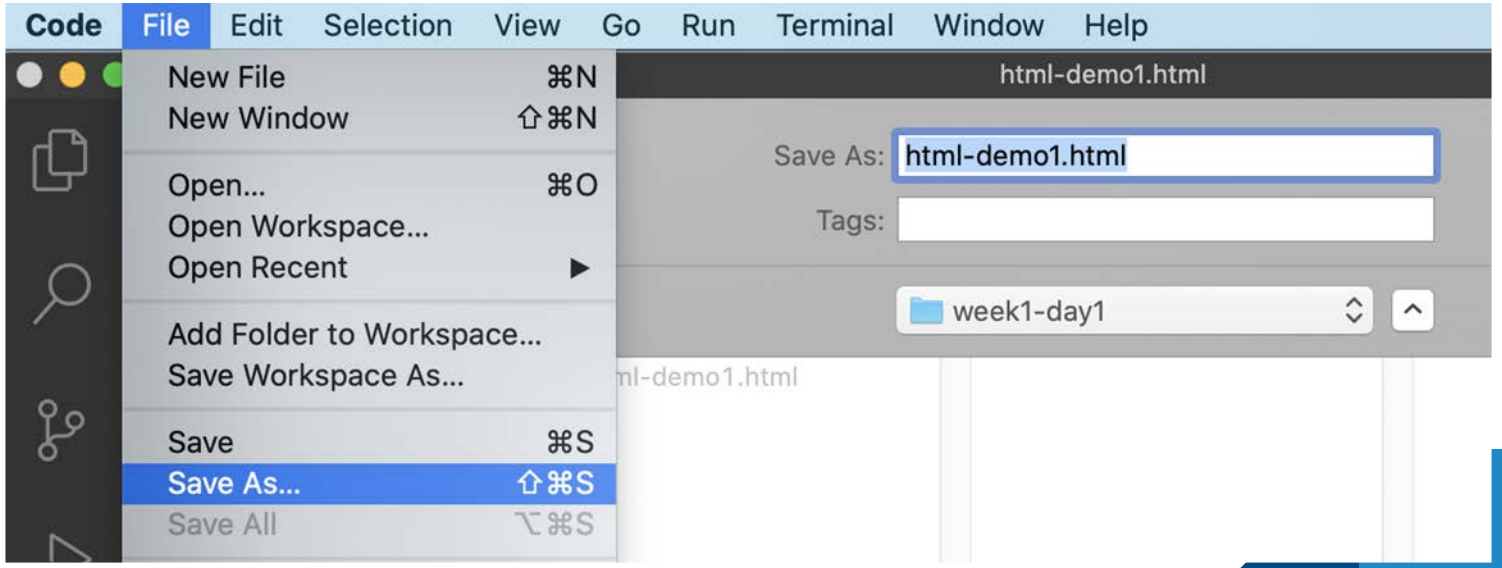
or



In-Class Exercise / Demo

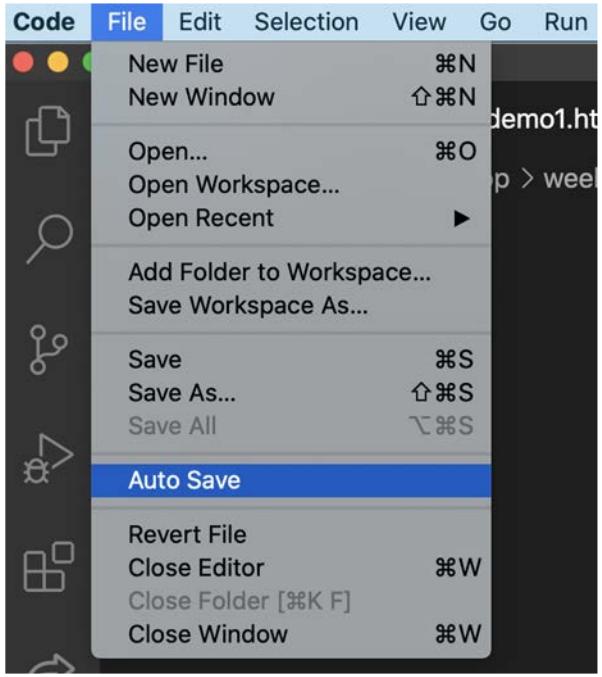
Initial Setup

- Save file as **html-demo1.html** into your *week1-day1* folder.



In-Class Exercise / Demo
Initial Setup

- This step is optional but highly recommended.



In-Class Exercise / Demo

Basic HTML Markup Structure

- On the *html-demo1.html* file, enter the following html markup codes:

```
<!DOCTYPE html>
<html>
<head>
<title>My First Web Page</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

Notes:
Spaces and line spaces are not important between elements. However, you should not have any empty spaces within the element.
Example:
Not okay: < body > or </ body>
Okay: <body>

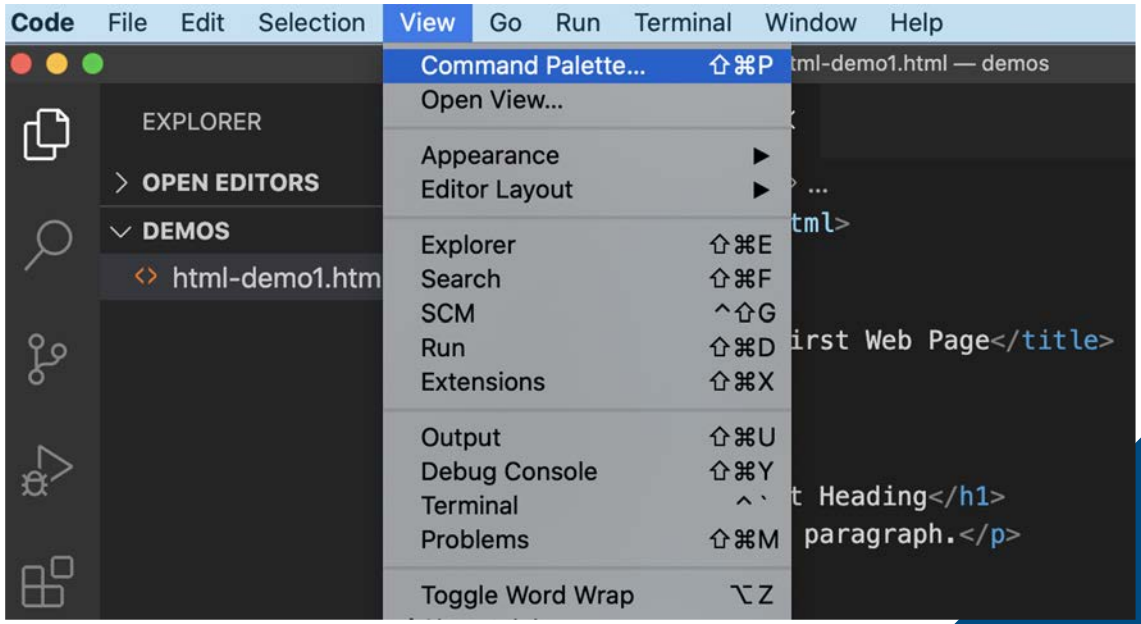
empty line(s)

<h1>…….

In-Class Exercise / Demo

Basic HTML Markup Structure

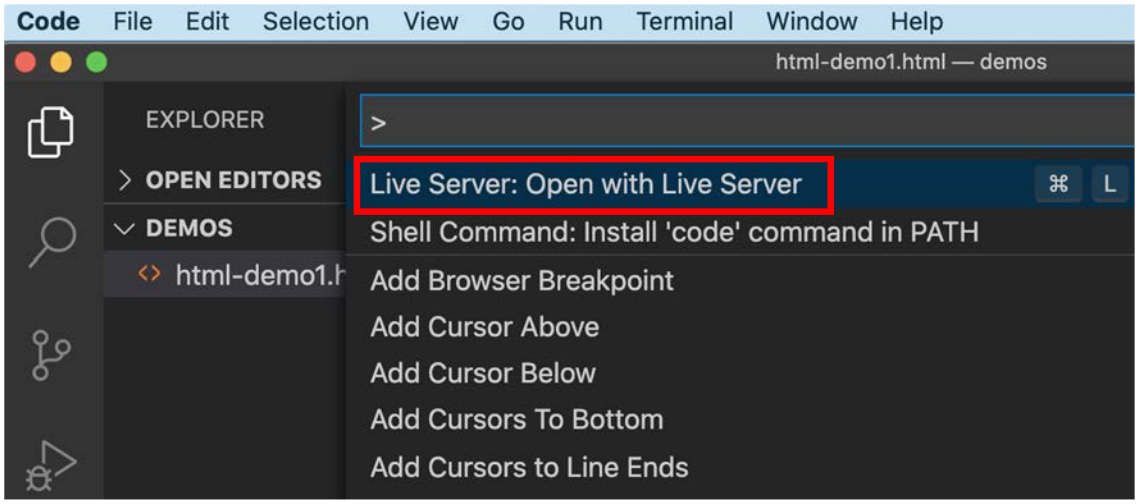
- Visual Studio Code has a feature that allow preview the html file on the browser.
- Here's what you do - first open the **Command Palette**:



In-Class Exercise / Demo

Basic HTML Markup Structure

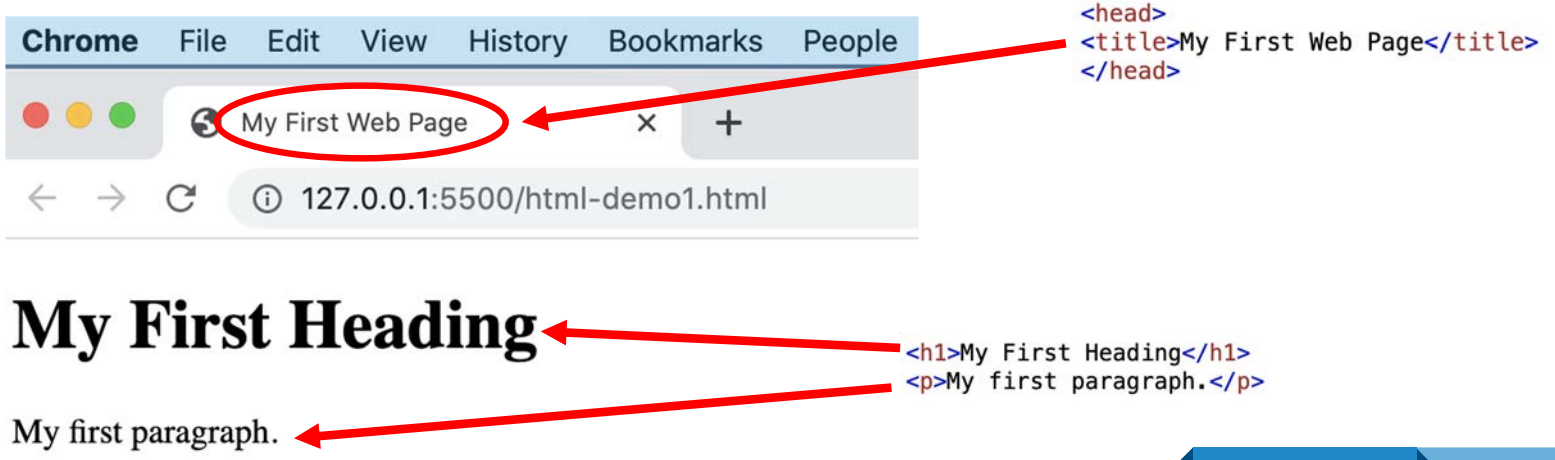
- On the Command Palette, select **Live Server…**



In-Class Exercise / Demo

Basic HTML Markup Structure

- This should open the page on the browser and it should look like this:



In-Class Exercise / Demo

Basic HTML Markup Structure

- Let's add more html markups….we'll start by wrapping all the codes within the <body> element with a container. We'll use a ** ** block element for this and name it *container* using an **id** attribute.

```
<body>

<div id="container">

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</div>

</body>
```

In-Class Exercise / Demo

Basic HTML Markup Structure

- Next let's organize the page into the following areas or sections:
  - **Header** – this is at the very top of the page and usually contains the site brand ex: logo
  - **Navigation** – this will be below the header and will contain the main navigation links
  - **Main** - this will be below the navigation and will contain the main content subject matter
  - **Footer** – this will be at the bottom of the page and usually contains copyright information
- We will use the standard HTML5 descriptive elements to markup these areas. Since they are descriptive, there's no need to provide any special id or class name attributes. However, sometimes we may need to provide an attribute name for certain situations, like styling purposes.

In-Class Exercise / Demo

Basic HTML Markup Structure

- Add the following codes:

```
<div id="container">

<header></header>
<nav></nav>

<main>
  <h1>My First Heading</h1>
  <p>My first paragraph.</p>
</main>

<footer></footer>

</div>
```

In-Class Exercise / Demo

Basic HTML Markup Structure

- Add the following content:

```
<header>My Web Site Brand</header>
<nav>
    <ul>
        <li>Home</li>
        <li>About</li>
        <li>Gallery</li>
        <li>Contact</li>
    </ul>
</nav>

<main>
  <h1>My First Heading</h1>
  <p>My first paragraph.</p>
</main>

<footer>&copy; Copyright 2020</footer>
```

The browser should auto update and this is what it looks like:

In-Class Exercise / Demo

Basic HTML Markup Structure

- The next step is a good convention practice as a web developer.
- Let's add in some HTML comments so that we can explain to ourselves or someone else in the team later on what the code means or changes that were made to the codes:

```
<!-- Container to wrap everything -->
<div id="container">

<!-- Brand of website -->
<header>My Web Site Brand</header>
<!-- Main Navigation -->
<nav>
    <ul>
        <li>Home</li>
        <li>About</li>
        <li>Gallery</li>
        <li>Contact</li>
    </ul>
</nav> <!-- End Main Navigation -->

<!-- Main Content -->
<main>
  <h1>My First Heading</h1>
  <p>My first paragraph.</p>
</main> <!-- End Main Content -->

<!-- Footer Information -->
<footer>&copy; Copyright 2020</footer>

</div> <!-- End Container -->
```

In-Class Exercise / Demo

Commit Code to GitHub

- As you work on your project, it's important to commit your code to your GitHub repository to make sure you never lose any of your work. The most common way to work is to commit your code every time you finish a main feature in your project.

  Beginning August 13, 2021, passwords are no longer accepted for remote github authentication. You need to generate a private token. Visit this link for step-by-step instructions how to generate a token.

- These are the steps you need to do for every exercises (most), assignments and projects:
  - git add .
  - git commit -m "write a brief message here that describes the change you've made to your project files"
  - git remote add origin \ *(shift + return or enter to move to next line)*
    > https://your_username:your_access_token@github.com/your_username/your_git_repo.git (see below **)
  - git push origin main:refs/heads/main

** if you get this message: **error: remote origin already exists** Enter this: **git remote remove origin**
Then enter this to check if it's removed: **git remote** If it didn't print any message, then it's been removed. Now try step 3 above again.

In-Class Exercise / Demo

Commit Code to GitHub

- After pushing, you can now go to your online GitHub repository to check out if the files are copied over there.
- You should be able to see the new files and the commit message you had entered earlier.
- Copy your Github URL (click on the code button).
- Paste this URL in the dropbox in GAP (Week 1 Day 1 Exercise)
- **DUE**: Today 10.30PM PT

Questions?

Congratulations on creating your first web page and, successfully push it to Github repository!

Retain the WESTCLIFF folder for more demo exercises in the next session.

Git Resources:

https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

http://www.compciv.org/recipes/devops/git-and-github-setup/