


## **Code-Along: Using HTML to Structure a Web Page**

Site: [Engage](#)

Course: S010 Introduction to Web Development (April 2023)

Book:  Code-Along: Using HTML to Structure a Web Page

Printed by: Juliana Nocchi Dobal

Date: Sunday, 23 April 2023,

## Table of contents

**Overview**

**Setup**

**Step 1: Create the file folder structure for the site.**

**Step 2: Create the Home Page**

**Step 3: Review the wireframe to see what elements need to be added**

**Step 4: Add the new elements to the index.html file**

**Step 5: Formatting**

**Step 6: Comments**

## Overview

We are going to start working on a simple website that will allow us to explore each of the concepts we will learn in this course and apply them to an ongoing project. This project will be a personal resume and portfolio.

This code along will review the following concepts:

- Markup Language Basics
- Sitemaps & Wireframes
- Elements, Structures, & Attributes
- Formatting Elements in HTML

## Toolkit

### Tools You Have:

- HTML documents are built using elements like `<head>` and `<body>`.
- Some elements are a pair of start (`<head>`) and end (`</head>`) tags.
- Some elements are self-closing and don't need an end tag (like `<hr />`, for example).
- We use elements like `<!DOCTYPE html>`, `<head></head>`, `<body></body>`, and `<html></html>` to define a document structure.

### Tools to Add:

- Use a sitemap to determine which folders are needed for a website.
- Create folders on your computer for your website's files.
- Translate requirements from a wireframe into the skeleton of a page's code.
- Use block elements such as headings (`<h1>` and `<h2>`) and paragraph (`<p>`) tags to create structure in a webpage.

#### Note

You should **type out** all the code presented instead of copying and pasting. Typing out the code improves muscle memory, but it also gives you more experience with troubleshooting typos.

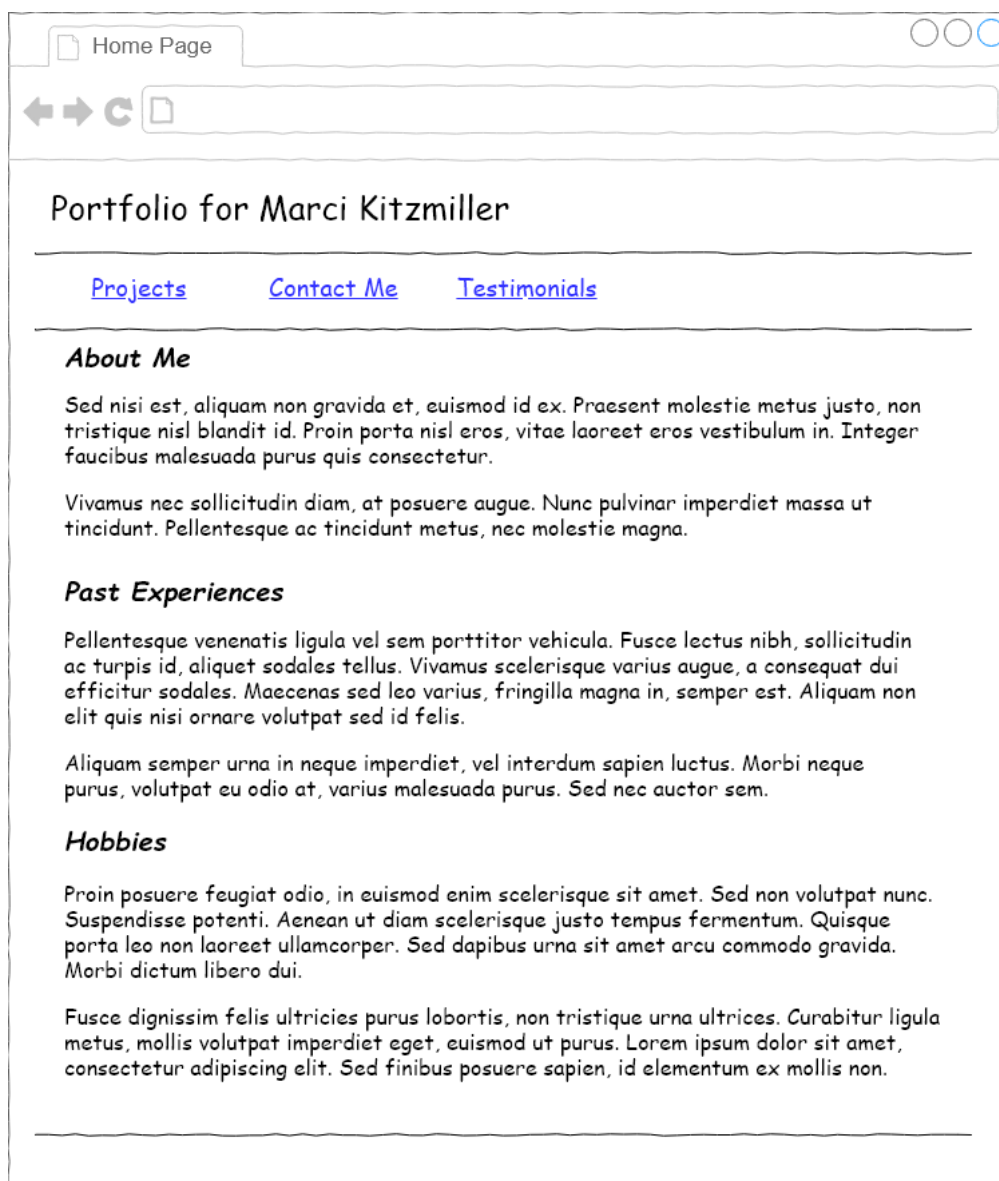
## Setup

The screenshots in this Code-Along use Google Chrome as the Web browser for testing. However, you are encouraged to view the page in at least a couple of other browsers, such as Firefox, Safari, or Edge. Seeing the same content in multiple browsers will give you a wider view of user experiences with your content, as well as help you identify how the browsers are both the same and different at the same time.

Here are the wireframe and sitemap that we'll be working with for this Code-Along. We'll add more wireframes as we go but we need one sitemap for this project. Note, however, that you may go through many revisions of wireframes and sitemaps when you start working with clients.

## Homepage Wireframe

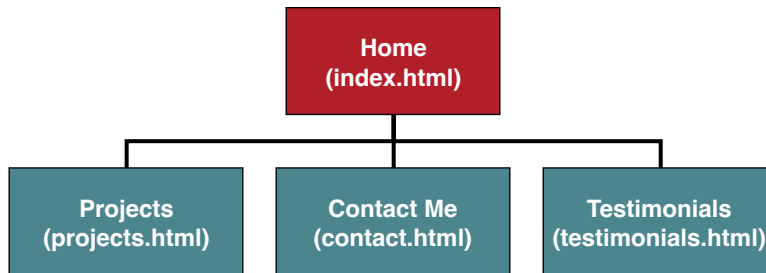
The home page will have some basic information for people who view our portfolio, including an About Me section, a Past Experiences section, and a Hobbies section. Note the title ("Portfolio for Marci Kitzmiller") at the top of the page, with a navigation bar with links for Projects, Contact Me, and Testimonials under the title.



If you wish to create wireframes for yourself, you have a variety of options. The most basic is just old-fashioned paper and pen, but you could use drawing tools in Microsoft Office or Google Drive. There are also specialized tools available for creating wireframes, including [Draw.io](https://draw.io) (a free, Web-based tool) or [Balsamiq](https://balsamiq.com/) (which requires a paid license after the trial period expires).

## Sitemap

The sitemap represents the pages in the website and how they are connected to each other. In this case, we will have a Home (index.html), and it will include links to a Projects page (projects.html), a Contact Me page (contact.html), and a Testimonials (testimonials.html).



### Step 1: Create the file folder structure for the site.

An artist's portfolio is a case or binder that contains the best pieces of the artist's work, all in one spot. That's what we'll be creating. We need a "case" in which to store our pieces. Our case will be a folder we put everything into on our local machine (meaning computer you regularly work at), so find a place on your computer that you can easily remember because we'll be coming back to it often.

- Create a folder named *portfolio* in the folder you have designated for this course on your computer.

## Step 2: Create the Home Page

The sitemap shows that we have four pages planned:

- Home (index.html)
- Projects (projects.html)
- Contact Me (contact.html)
- Testimonials (testimonials.html)

We need the Home page first, so we're going to open our code editor, create the file, and then add in the basic skeleton of tags `<html>`, `<head>`, and `<body>`.

1. Open your code editor of choice (if you haven't already) and create a new file.
2. Save the file as *index.html* **inside** the portfolio folder you created earlier.
3. Add `<!DOCTYPE html>` to the very first line of your file.
4. On the next line, type in `<html>`
5. Add a `<head>` tag on the next line.
6. We need to add the page title inside the `<head>` element so that the browser will display a meaningful name for the tab or title bar. Because tabs are often not very wide, we generally want to keep the title both short and meaning under the opening `<head>` tag, add the title element to include your name and the name of the website.

```
<title>Your Name's Portfolio</title>
```

7. Under the title, close the `</head>` element.
8. For the body section, we will start it with a `<body>` tag.
9. Then hit Enter a couple of times to create empty space to work in, and close the `</body>` element.
10. After closing the body, close the `</html>` element on the last line of text.
11. Save the changes to the file.
12. Open it in your browser. You can open a file in a browser using any of the following methods:
  - Double-click the saved file in the portfolio folder. If the file is saved correctly, it will open in your computer's default browser automatically.
  - Find the file in the portfolio folder and right-click it. Select *Open with* in the context menu and select the browser you want to use.
  - Open the browser you want to use on its own, then use File / Open (or the shortcut key Ctrl + o) and navigate file in your portfolio folder.

### Home Page File Names

Remember that any HTML page must end with the filename extension *.html*, so that the user's computer knows to open the file in a browser. However, there are restrictions to the filenames themselves, too.

A website's homepage can have one of three filenames: *index.html*, *home.html*, or *default.html*. If you name the homepage correctly, you can give visitors just the URL for your website (e.g., <http://www.yournameportfolio.com> or *YourNamePortfolio.com*). When someone visits that site, their browser will look for a page named as a homepage and open it by default. If the homepage is not named correctly, the browser will display a message that the website is not available.

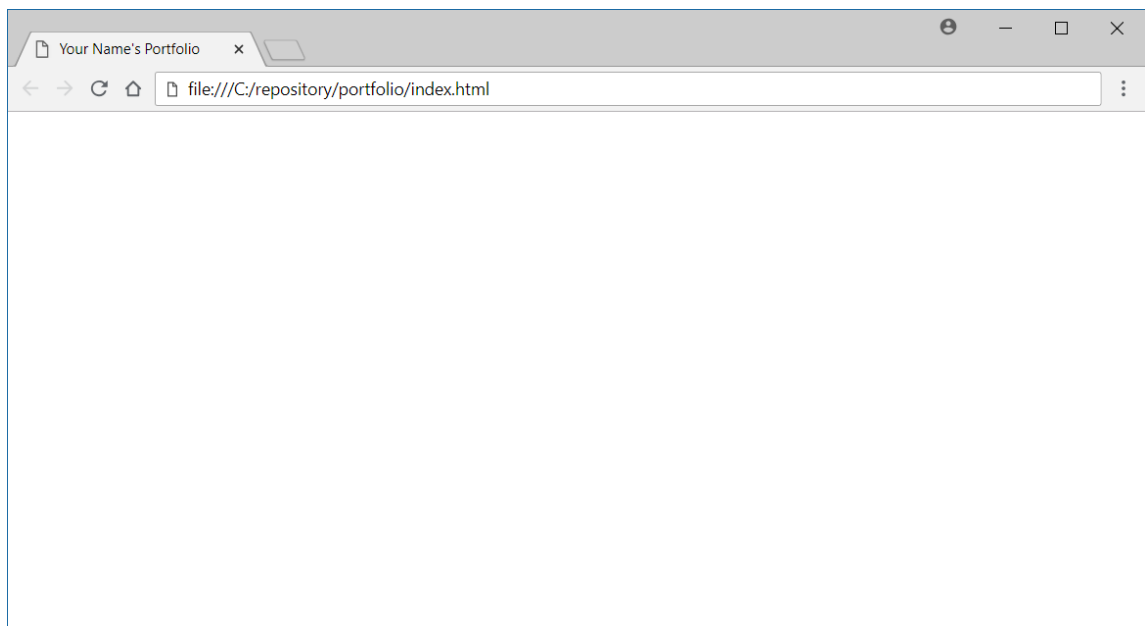
### Self-Check

Does your code editor have the following code?

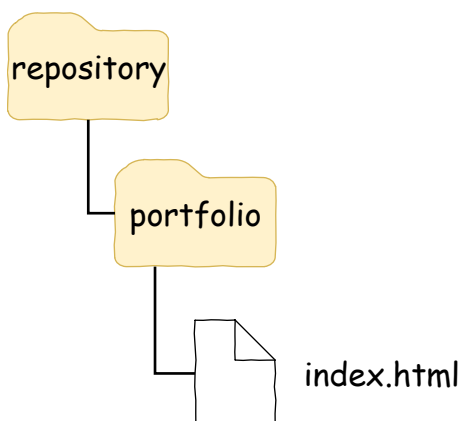
```
<!DOCTYPE html>
<html>
  <head>
    <title>Your Name's Portfolio</title>
  </head>
  <body>

  </body>
</html>
```

When you view the page in your browser, the only thing you should see is your title in the tab. The page is empty because the `<body>` element is empty. However, if you look at the address bar, you will see that it doesn't look like a normal web address (<http://www.someplace.com>). Instead, it displays the path to the `index.html` file on your computer, including all the subfolders to that file.



The exact path will be different from one computer to another, but it should definitely end in `/portfolio/index.html`. If it doesn't, check that the root folder of your repository includes a folder named `portfolio` and that the file named `index.html` is inside that folder,





### Step 3: Review the wireframe to see what elements need to be added

Let's break down our wireframe now. At this point, we just want to make sure we know how to translate the wireframe into HTML. We don't start keying in code quite yet.

- At the very top, we have "Portfolio for Marci Kitzmiller" (but you should use your name instead). This is the biggest line of text on the page and it acts as the main page title, so we will use an `<h1>` element.
- The wireframe shows a horizontal line under the main heading, so we will add a horizontal rule (`<hr>`).
- We will eventually need links to our other pages, but for now, we'll just use paragraphs `<p>` to act as placeholders for the links.
- We will add another horizontal rule, `<hr>`, under the placeholders for the links.
- The main section of the page will include three separate chunks of content: About Me, Past Experiences, and Hobbies.
  - "About Me", "Past Experiences", and "Hobbies" all look like smaller headings than the main heading, so we'll use `<h2>` for each.
  - Each of these sections includes a couple of paragraphs of text, so under each `<h2>`, add a couple of `<p>` elements with some appropriate information about yourself. Keep in mind that you may eventually want to share this with potential employers, so think about what you want others to know about you.
- Finally, add one last `<hr>` at the bottom of the page.

At this point, we have mapped the wireframe to the following elements:

1. `<h1>Portfolio for Marci Kitzmiller</h1>`
2. `<hr>`
3. `<p>Links to our pages will go here</p>`
4. `<hr>`
5. `<h2>About Me</h2>`
6. `<p>Paragraph 1 about you.</p>`
7. `<p>Paragraph 2 with more stuff about you.</p>`
8. `<h2>Past Experiences</h2>`
9. `<p>Paragraph 1 describing something you've accomplished in the past.</p>`
10. `<p>Paragraph 2 with another accomplishment.</p>`
11. `<h2>Hobbies</h2>`
12. `<p>Paragraph 1 describing your favorite hobby.</p>`
13. `<p>Paragraph 2 describing another hobby.</p>`
14. `<hr>`

## Step 4: Add the new elements to the index.html file

Now we are ready to start creating the page. Remember that everything in the wireframe should be visible in the main window browser, so everything we add from this point forward must be nested inside the open and close tags of the body element. Here is the wireframe again:



1. The main heading should be inside an h1 element: `<h1>Portfolio for Your Name</h1>`
2. Add a horizontal line under the main heading.
3. Add a paragraph element that will eventually hold the navigation links.
4. Add a horizontal line under the navigation links.
5. Add the first subheading: `<h2>About Me</h2>`
6. Add two paragraphs of text under the subheading, using a separate paragraph element for each paragraph. Make sure to close each paragraph correctly.
7. Add the second subheading: `<h2>Past Experiences</h2>`
8. Add two more paragraphs under this subheading.
9. Add the third subheading: `<h2>Hobbies</h2>`
10. Add two more paragraphs.
11. Add a third horizontal line under the last paragraph.

Now, save the file and refresh the page in your browser. Compare what the browser displays to the wireframe presented earlier in this lesson. Is everything there? Did you leave anything out?

### Oops?

One of the most common mistakes, even for veteran developers, is to forget to close out a tag so, if something looks wonky when you check out your browser, make certain that all your tags are spelled correctly, have a less than (<) and greater than (>) sign around them, and open and close appropriately.

### Self-Check

Your code should look something like the following, but with text about YOU in it.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Marci Kitzmiller's Portfolio</title>
  </head>

  <body>

    <h1>Portfolio for Marci Kitzmiller</h1>

    <hr>
    <p>Links to our pages will go here.</p>
    <hr>

    <h2>About Me</h2>
    <p>Sed nisi est, aliquam non gravida et, euismod id ex. Praesent molestie metus
    o, non tristique nisl blandit id. Proin porta nisl eros, vitae laoreet eros vestibulum in
    nteger faucibus malesuada purus quis consectetur.</p>
    <p>Vivamus nec sollicitudin diam, at posuere augue. Nunc pulvinar imperdiet massa
    tincidunt. Pellentesque ac tincidunt metus, nec molestie magna.</p>

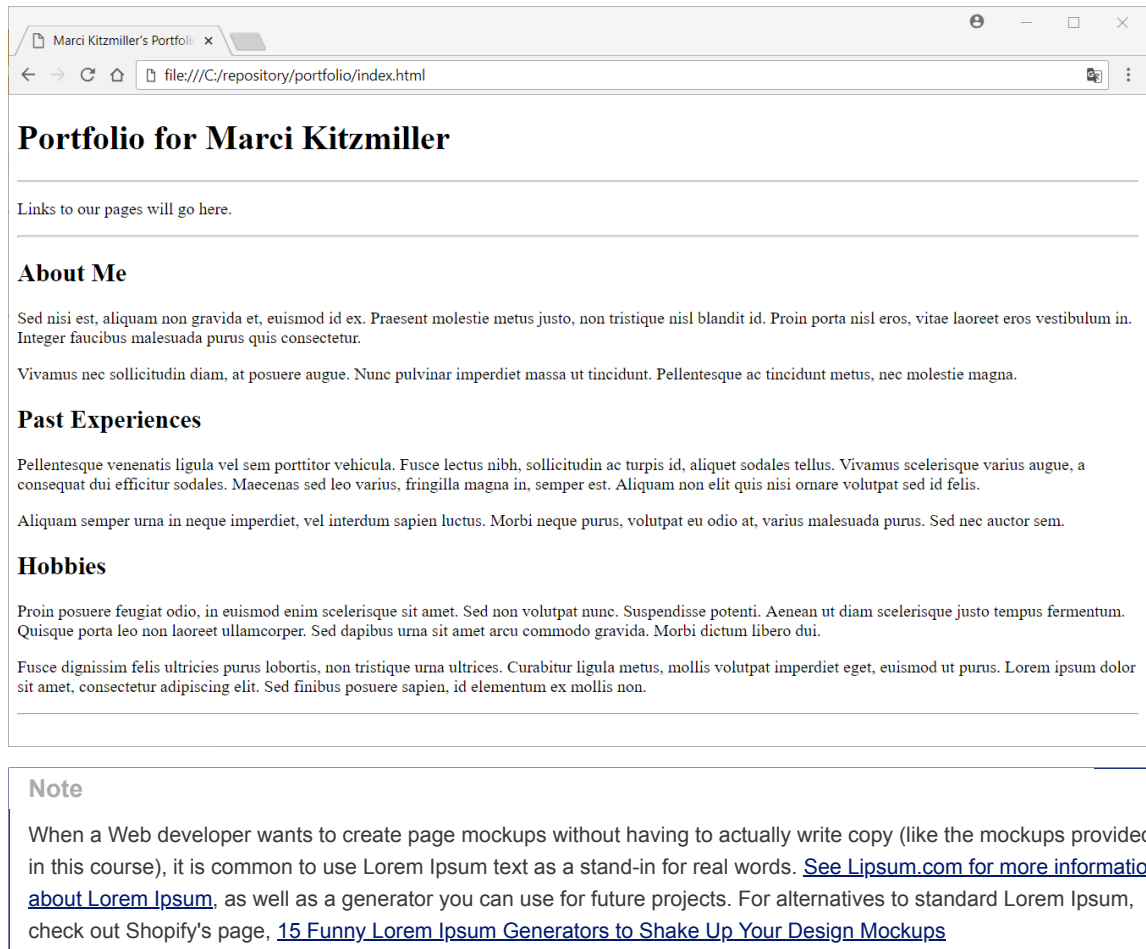
    <h2>Past Experiences</h2>
    <p>Pellentesque venenatis ligula vel sem porttitor vehicula. Fusce lectus nibh,
    icitudin ac turpis id, aliquet sodales tellus. Vivamus scelerisque varius augue, a conse
    dui efficitur sodales. Maecenas sed leo varius, fringilla magna in, semper est. Aliquam
    elit quis nisi ornare volutpat sed id felis.</p>
    <p>Aliquam semper urna in neque imperdiet, vel interdum sapien luctus. Morbi nequ
    urus, volutpat eu odio at, varius malesuada purus. Sed nec auctor sem.</p>

    <h2>Hobbies</h2>
    <p>Proin posuere feugiat odio, in euismod enim scelerisque sit amet. Sed non volu
    t nunc. Suspendisse potenti. Aenean ut diam scelerisque justo tempus fermentum. Quisque
    a leo non laoreet ullamcorper. Sed dapibus urna sit amet arcu commodo gravida. Morbi dic
    libero dui. </p>
    <p>Fusce dignissim felis ultricies purus lobortis, non tristique urna ultrices.
    bitur ligula metus, mollis volutpat imperdiet eget, euismod ut purus. Lorem ipsum dolor
    amet, consectetur adipiscing elit. Sed finibus posuere sapien, id elementum ex mollis no
    p>

    <hr>

  </body>
</html>
```

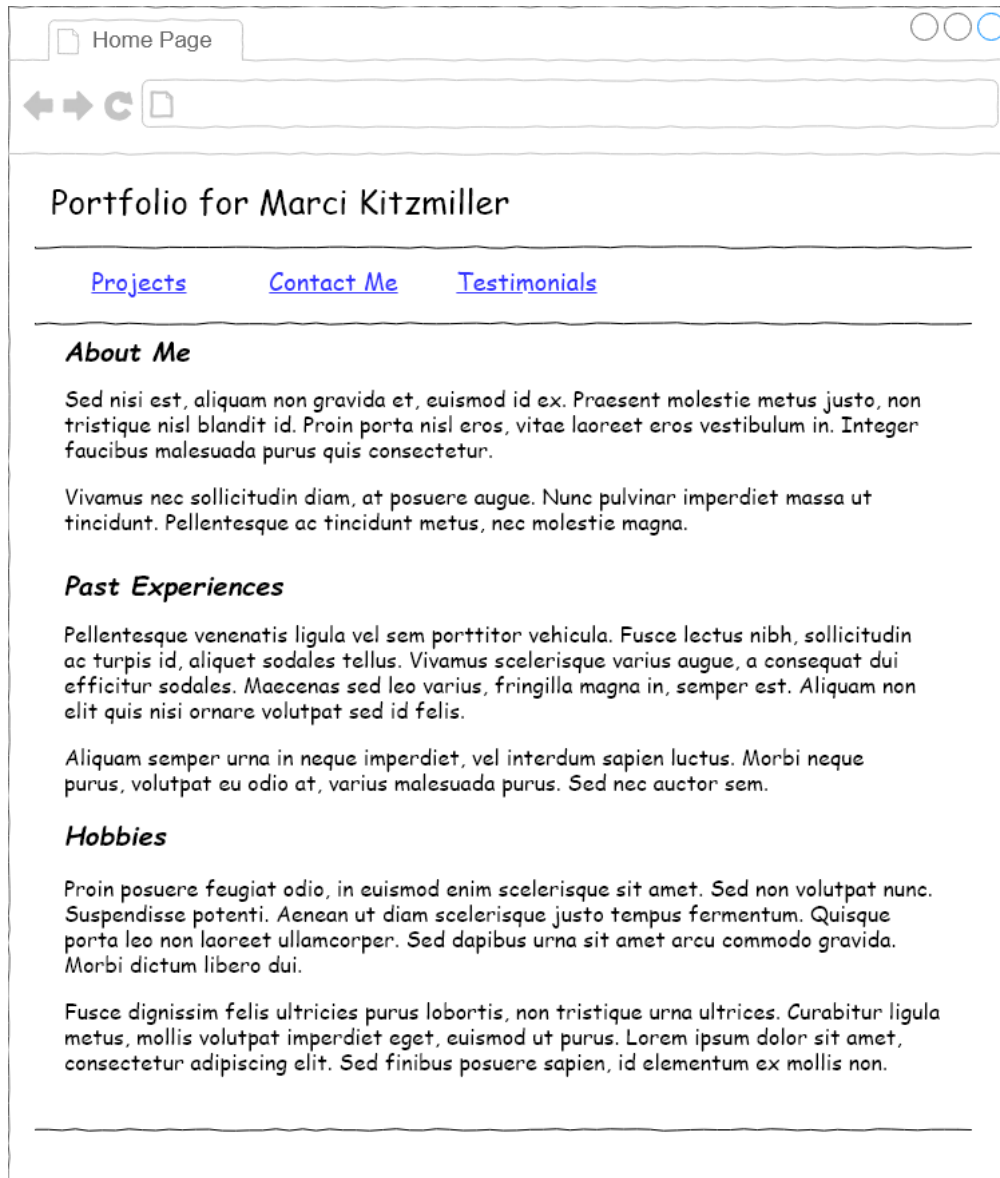
And the page should look something like this in your browser:



## Step 5: Formatting

For the time being, we really don't have any tools for elaborate formatting, so this version of our homepage will look somewhat – not unlike the [original Web pages](#) created by Sir Tim Berners-Lee when he invented the Web in the 1990s. We'll make it look better using CSS in another part of this course. Right now, though, we can add italics.

Let's look at the wireframe again.



If we look at our last Self-Check from Step 4, it looks like we need to italicize the three h2s – About Me, Past Experiences, and Hobbies.

1. Go back to your code editor, find your About Me `<h2>` and then, right after the h2 tag, type `<em>` in between the `<h2>` "About" so you'll have something that looks like:

```
<h2><em>About Me</h2>
```

2. We need to close out the em tag so type `</em>` right after "Me" but before the last closing `</h2>` so the whole line looks like:

```
<h2><em>About Me</em></h2>
```

3. Next, let's add `<em>` around Past Experiences:

```
<h2><em>Past Experiences</em></h2>
```

4. And finally, around Hobbies

```
<h2><em>Hobbies</em></h2>
```

5. Save the index.html file.

6. Test the file in your browser.

### Oops?

Check that the headings have both the h2 format and that they are italicized, like this:

***About Me***

If they are big (like h2) but not italicized, or italicized but not big, check that you have both tags **together** in your code:

```
<h2><em>About Me</em></h2>
```

### Self-Check

Your code right now should be similar to:

```

<!DOCTYPE html>
<html>
  <head>
    <title>Marci Kitzmiller's Portfolio</title>
  </head>

  <body>

    <h1>Portfolio for Marci Kitzmiller</h1>

    <hr>
    <p>Links to our pages will go here.</p>
    <hr>

    <h2><em>About Me</em></h2>
    <p>Sed nisi est, aliquam non gravida et, euismod id ex. Praesent molestie metus ;
o, non tristique nisl blandit id. Proin porta nisl eros, vitae laoreet eros vestibulum i
neger faucibus malesuada purus quis consectetur.</p>
    <p>Vivamus nec sollicitudin diam, at posuere augue. Nunc pulvinar imperdiet massa
tincidunt. Pellentesque ac tincidunt metus, nec molestie magna.</p>

    <h2><em>Past Experiences</em></h2>
    <p>Pellentesque venenatis ligula vel sem porttitor vehicula. Fusce lectus nibh, ;
icitudin ac turpis id, aliquet sodales tellus. Vivamus scelerisque varius augue, a conse
dui efficitur sodales. Maecenas sed leo varius, fringilla magna in, semper est. Aliquam i
elit quis nisi ornare volutpat sed id felis.</p>
    <p>Aliquam semper urna in neque imperdiet, vel interdum sapien luctus. Morbi nequ
urus, volutpat eu odio at, varius malesuada purus. Sed nec auctor sem.</p>

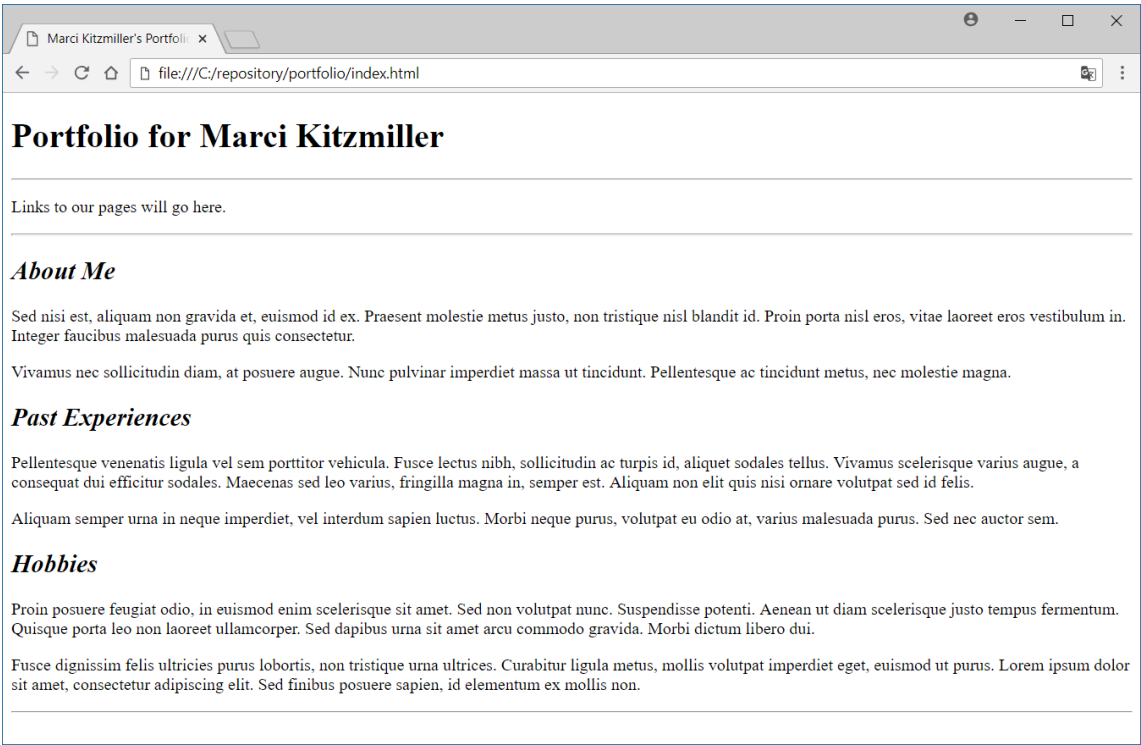
    <h2><em>Hobbies</em></h2>
    <p>Proin posuere feugiat odio, in euismod enim scelerisque sit amet. Sed non volu
t nunc. Suspendisse potenti. Aenean ut diam scelerisque justo tempus fermentum. Quisque p
a leo non laoreet ullamcorper. Sed dapibus urna sit amet arcu commodo gravida. Morbi dic
libero dui. </p>
    <p>Fusce dignissim felis ultricies purus lobortis, non tristique urna ultrices. (
bitur ligula metus, mollis volutpat imperdiet eget, euismod ut purus. Lorem ipsum dolor ;
amet, consectetur adipiscing elit. Sed finibus posuere sapien, id elementum ex mollis noi
p>

    <hr>

  </body>
</html>

```

And the page should look like this in your browser:





## Step 6: Comments

Let's add some comments to the file – information that we want to keep track of and potentially share with other developers, but users won't see in the browser window.

First, it's good practice to document who created the document, when it was created, and when the most recent changes were made. Near the top of the file, between the DOCTYPE declaration and the `<html>` tag, add the following comments, including your name and today's date. We will also include a line for the date of the last revision, but we'll leave it empty for now because this is the original version.

```
<!--  
Name: Your Name  
Date Created: Today's date  
Most recent revision:  
-->
```

Other information like the purpose of the document or your email address or other contact information can be included here as well.

We also want to leave ourselves a reminder of something we plan to do later, namely add the links at the top of the page. Right now we have the following code at the top of the body element:

```
<h1>Portfolio for Marci Kitzmiller</h1>  
  
<hr>  
<p>Links to our pages will go here.</p>  
<hr>
```

Create a new empty line underneath the first `<hr>` just above the "`<p>Links to our pages . . .`" and type in:

```
<!-- We need to add links to projects.html, contact.html, and testimonials.html -->
```

After adding all of the comments (and any others you might want to include now, SAVE the file in your code editor and refresh the browser window.

### Self-Check

Your code should look like:

```

<!DOCTYPE html>
<!--
    Name: Your Name
    Date Created: Today's date
    Most recent revision:
-->
<html>
  <head>
    <title>Marci Kitzmiller's Portfolio</title>
  </head>

  <body>

    <h1>Portfolio for Marci Kitzmiller</h1>

    <hr>
    <!-- We need to add links to projects.html, contact.html, and testimonials.html -->
    <p>Links to our pages will go here.</p>
    <hr>

    <h2><em>About Me</em></h2>
    <p>Sed nisi est, aliquam non gravida et, euismod id ex. Praesent molestie metus ;
o, non tristique nisl blandit id. Proin porta nisl eros, vitae laoreet eros vestibulum i
ninteger faucibus malesuada purus quis consectetur.</p>
    <p>Vivamus nec sollicitudin diam, at posuere augue. Nunc pulvinar imperdiet massa
tincidunt. Pellentesque ac tincidunt metus, nec molestie magna.</p>

    <h2><em>Past Experiences</em></h2>
    <p>Pellentesque venenatis ligula vel sem porttitor vehicula. Fusce lectus nibh, ;
icitudin ac turpis id, aliquet sodales tellus. Vivamus scelerisque varius augue, a conse
dui efficitur sodales. Maecenas sed leo varius, fringilla magna in, semper est. Aliquam i
elit quis nisi ornare volutpat sed id felis.</p>
    <p>Aliquam semper urna in neque imperdiet, vel interdum sapien luctus. Morbi neq
urus, volutpat eu odio at, varius malesuada purus. Sed nec auctor sem.</p>

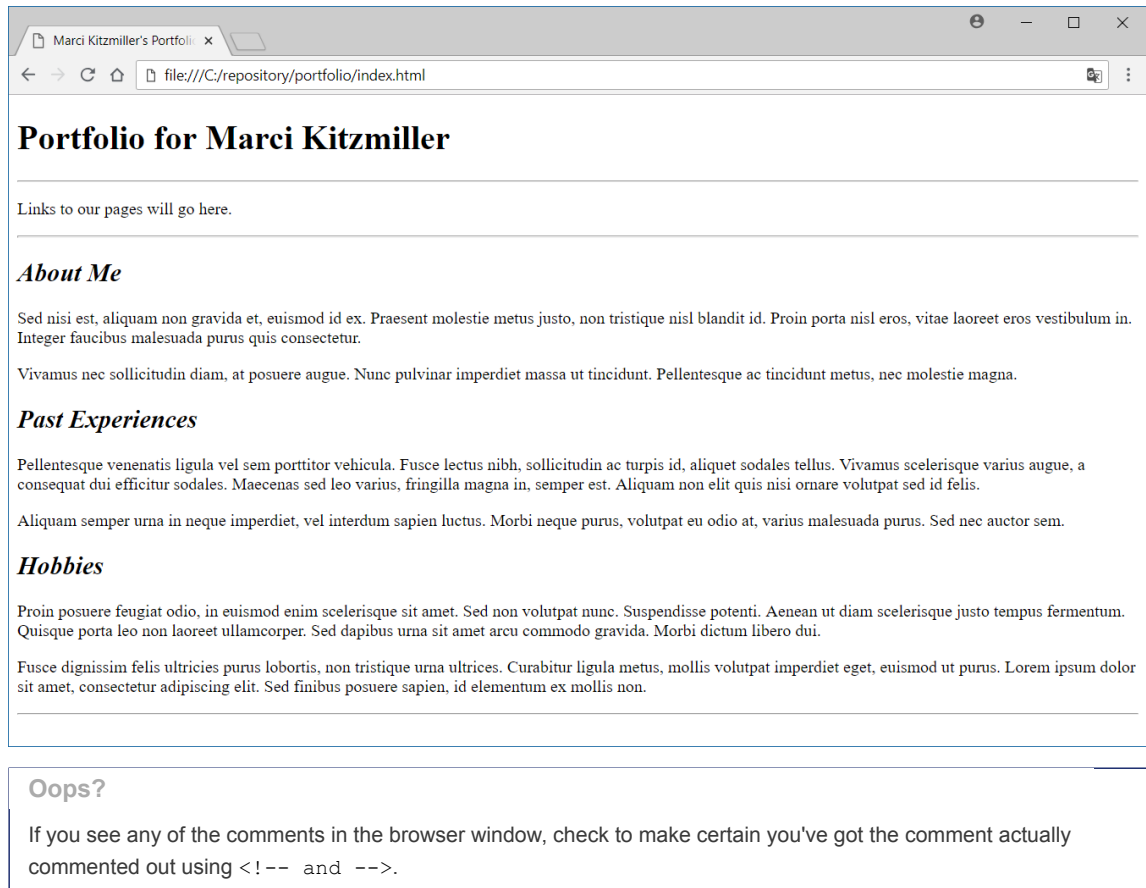
    <h2><em>Hobbies</em></h2>
    <p>Proin posuere feugiat odio, in euismod enim scelerisque sit amet. Sed non volu
t nunc. Suspendisse potenti. Aenean ut diam scelerisque justo tempus fermentum. Quisque p
a leo non laoreet ullamcorper. Sed dapibus urna sit amet arcu commodo gravida. Morbi dicitur
libero dui. </p>
    <p>Fusce dignissim felis ultricies purus lobortis, non tristique urna ultrices. (
bitur ligula metus, mollis volutpat imperdiet eget, euismod ut purus. Lorem ipsum dolor ;
amet, consectetur adipiscing elit. Sed finibus posuere sapien, id elementum ex mollis non
p>

    <hr>

  </body>
</html>

```

And the page should look exactly the same in your browser. It should still show ONLY your title, horizontal rules, the paragraph hold a place for our links later, and our three sections with their placeholder paragraphs.



## Wrapping It Up

You have now completed your first, personalized Web page! Here are the steps you completed:

1. Created the folder for your portfolio site on your local machine.
2. Created the Home Page file, *index.html*.
3. Checked out the wireframe to see what you needed to put into the starting skeleton for your index.html page like `<h1>` and `<body>`
4. Added in the elements that were specific to the index.html page like the heading tags and paragraph tags with placeholder text.
5. Added a little HTML formatting.
6. Added comments about the file and what we need to do next!