Boilerplate

In this section we are going to create a project starter or boiler plate code file that we can reuse in all of the following HTML, CSS, and JavaScript projects. Instead of having to recreate files. It is somewhat standard practice to duplicate and rename template code for production.

First create a file, and name it what ever you would like, for the purposes of this demo I will name my file code.

Next open your file in visual studio code, It should look like the following:

A screenshot of a computer program

Description automatically generated

In the empty folder create the following three files: index.html, style.css, and script.js. You should have the following:

A screenshot of a computer program

Description automatically generated

Now in the HTML, I’m going to go ahead and use Emet and do exclamation enter and that is going to give use the following code:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

</body>

</html>

This is a form of a boilerplate of HTML document. It gives us an HTML page with the HTML 5 document type tag, with a couple head tags and a body tag.

In the title tag were going to change the title tag to the following

<title>My Project</title>

This can further be customized for each project instance.

Next in the body tag we are going to add the following H1 tag:

 <h1>Project Starter.</h1>

Now to open this page, we first save our program, and we are now are faced with two options:

1. We could directly type the file directory into the browser like so:

A computer screen with white text

Description automatically generated

1. The second option is to use live server to dynamically render our page to the viewport(fancy word for the browser window), meaning that when ever we make a change our page will update automatically. To use live server right-click the page and click on Open with Live Server:

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

When this is done we can see that are page is open on my local host or my loopback address, which is <http://127.0.0.1/> in Port 5500, and the current file is index.html.

Next we want to make sure that we include our stylesheet, and the JavaScript in our index.html file.

Back in the index.html file make the following additions above the title tag:

First we type link and hit enter, then we fill in the href with our style.css file like so:

<link rel="stylesheet" href="style.css">

Afterword’s to link to our JavaScript we will make the following changes below the h1 tag and above the ending body tag we type script:src then press enter, then add the source JavaScript file to get the following:

<script src=""></script>

Great now our index.html boilerplate file has a reference to the style.css and the script.js file. We are pretty much done with our index.html file with one exception.

Next one thing I would like to mention is in some of the projects we will be tackling we will be using font awesome. So we have to include the font awesome style sheet and we have a couple ways of doing that

1. Is to go the following url <https://fontawesome.com/start> type in your e-mail and hit “send kit code” and it will give you a link to the font awesome css. But as astute web developers we know that the less people that have your email address/personal information the better.
2. The better alternative is to go to the following url <https://cdnjs.com/> and search for font-awesome and it will provide us with the following link tag:

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css" integrity="sha512-z3gLpd7yknf1YoNbCzqRKc4qyor8gaKU1qmn+CShxbuBusANI9QpRohGBreCFkKxLhei6S9CQXFEbbKuqLg0DA==" crossorigin="anonymous" referrerpolicy="no-referrer" />

Next place this link tag above the style sheet link tag and we can now use font-awesome!

So some projects might start off with this link tag already here, but for now lets gohead and comment this out for now. Then on project where we use it, you can uncommented it out, simple as.

Moving to our stylesheet file (style.css) we do want to start with a default font. The default font for this course is the Roboto font. Go the url <https://fonts.google.com/specimen/Roboto> and search for the Regular 400 font and the Bold 700 and select both; making sure that the @import radio button is selected next copy the following and paste it directly into the style sheet :

@import url('https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&display=swap');

We can now use the roboto font. Next we are going to apply our font to the body with with following inclusion:

body {

    font-family: 'Roboto',sans-serif;

}

Now if we move to browser, we can see the following change:A screenshot of a computer

Description automatically generated

Now, we also want to set on the universal selector meaning anything within the brackets applies to everything. Next we add a box-sizing style, this makes any border box we add padding, or if we add ant border on to an element, it dose not effect the width. it should look like the following:

\*{

    box-sizing: border-box;

}

Back in the body style we are going to get rid of the margin and we are also going to display flex. It should look like the following:

body {

    font-family: 'Roboto',sans-serif;

    display: flex;

    margin: 0;

}

The reason for this is because I want to center everything. Most of the projects are directly centered in the viewport.

Another topic I should mention a bit about is the Flex Box. To demonstrate this, we can move back to the index.html file and the following h tags:

<body>

    <h1>Project Starter.</h1>

    <h2>Hello</h2>

    <h3>Hello</h3>

    <script src="script.js"></script>

</body>

Now by setting our body style to display flex it is going to place all of our header in a row like so:

A screenshot of a computer

Description automatically generated

Now we don’t want this, if we want it to go back vertical, but still be a flex box, back in the style.css file we can change the flex direction with the following style:

body {

    font-family: 'Roboto',sans-serif;

    display: flex;

    flex-direction: column;

    margin: 0;

}

If we look at the browser we can see the following:

A screenshot of a computer

Description automatically generated

Now, since this is a flex box, I can use

align-items: center;

To center our heading like so:

A screen shot of a computer

Description automatically generated

Since our flex box is a column then align-items is going to pertain to aligning the elements horizontally, and if it a row it will center it vertically. Now that we have it center horizontally, to completely center the body we can add the following style:

justify-content: center;

This style alone doesn’t do much because our flex boxes do not have a height set like so:

height:100vh;

this sets the height to 100 vh, which is a viewport height and that means the body takes up the entire height of the browser. The last thing we will add is a overflow style:

overflow:hidden;

this is because we don’t want scroll bars to be shown by default