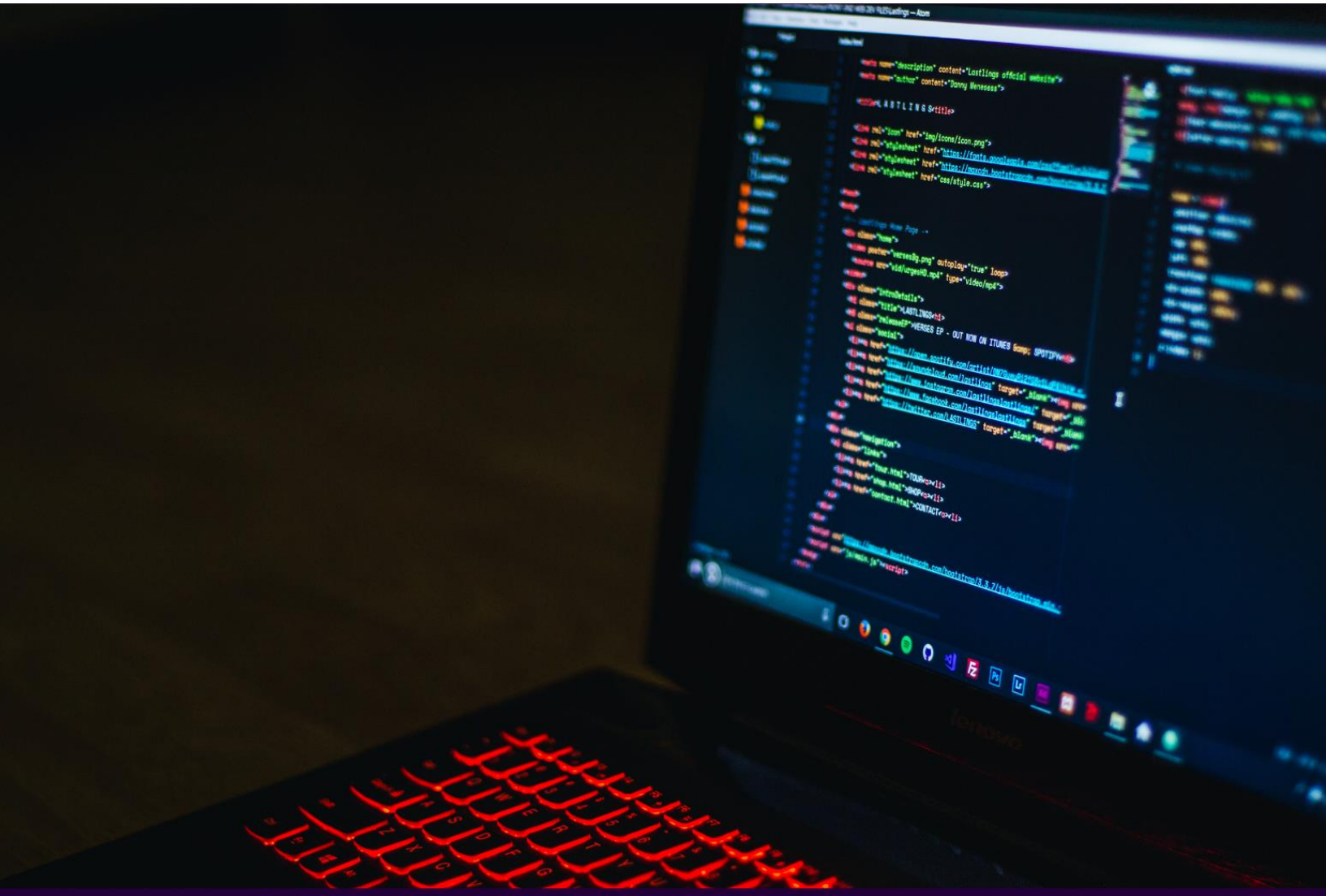
Computer Coding

Learn to Program



HTML & CSS: Web Design

Astro Clare Technology

Astro Clare Technology © Flint, MI EST 2022

“ HELLO CODER “

We at Astro Clare Technology are proud to have you receive this book! There will be many fun activities for you to enjoy. If you are a parent purchasing this book for your child to venture into they great world of becoming an software engineer, we thank you for trusting us ! We have many great activities for you and your child to enjoy together.

Our main objective is to get individuals to begin their journey in coding, programming and computing technologies. We see the lack of resources decline over the years and understand that purchasing these courses at Universities are even more challenging financially and timely. So we will due more than just “teach” you how to become the greatest engineer!

You are going to learn many skills such as: critical thinking, logical outlooks, problem solving skills, discipline and career ready structure and even more !

Each successful assessment from you will grant an Certificate from Astro Clare Technology along with an PDF of your course completion to show your next employer.

Once again, we thank you for trusting in Astro Clare Technology and hope you put your best code forward.

Clarence Scott
CEO & Founder

OUR WEB DESIGN GOALS

- **Plan and Structure:** Learn how to plan your web design and structure your flow chart. Understanding your goal and purpose.

- **Write HTML Markup**: Learning our text-editor, creating files, implementing code (known as markup in HTML) to build our webpage.

- **Apply styling with CSS:** Using inline cascading styling, learn about external files and connecting them to HTML files.

- **Test and Debug**: Preview the web page to ensure it displays as intended. Check for any errors or inconsistencies in layout and functionality, these are known as “bugs in code”. We’ll fix these.

- **Publishing and Maintenance:** Upload your HTML your html file to the internet or intranet and any files in the web pages folder, to a web server. For this book you will be creating a portfolio to upload.

Overview

HTML5 is a markup language used for structuring and presenting content on the World Wide Web (WWW). This is the fifth version for the **H**yper**T**est **M**arkup **L**anguage. In other words HTML is what you see today in your browser or “internet” when you are using your computing devices.

HTML is the center of every web page. Behind every thing you search on the web is mainly html or some form of html code. This is used so your browser can display, text, images and tables. In this book you’ll work with html code to control the structure and elements of your web page.

Browsers look for html documents to display the webpage. The webpage can include may types of elements.

Now that you have a little more information on what we will be doing, let’s begin building your very own web page. If you do not have a test editor we will walk through steps on downloading a text editor. For this book we will use Visual Studio Code, developed by Microsoft.

Pre-Requisites

In order to begin coding with us you only need 3 things: Computer (with monitor), Text Editor and the drive to want to code.

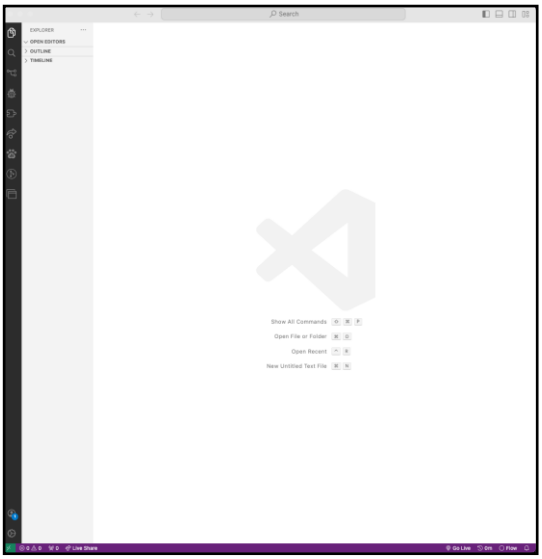
First, let's make sure you have the visual studio code software downloaded. Head over to Microsoft's website and download Visual Studio Code, or click the link here Visual Studio Download. In the Extensions tab after you have it downloaded install the extension Live Server by Ritwick Day and you will be ready to continue. If you need help refer to our Youtube Video **" Setting Up Visual Studio for Development".**

Secondly, ensure you are in a quiet setting and can focus on coding and studying the information in this book. Focus is key, determination is power. Code is the goal.

Lastly, grab a snack and something to drink (always have a water near you) so that you can stay mentally strong while studying and coding. The brain will become very tired and you will become mentally drained if you do not.

Creating Our Docs

"Hello World"

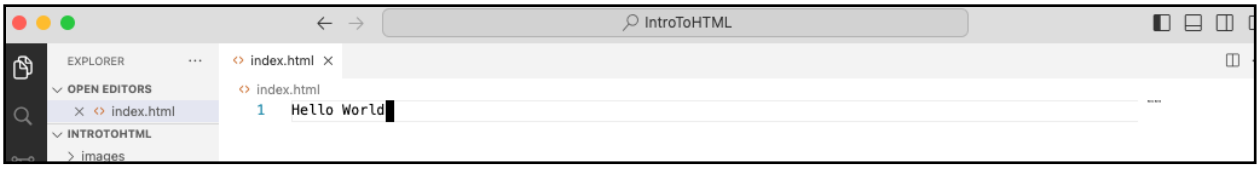


Now that you are ready to code, let's take a look at our setup in visual studio.
If everything was done correctly your screen should look like this:

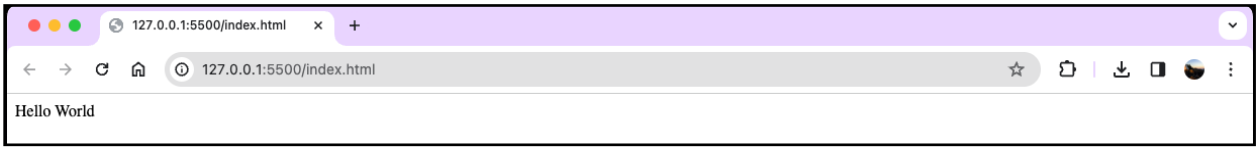
If so, you have successfully downloaded Visual Studio Code. If you hover over the "Open Editors" section you can add a new file. Do that.

In that file we are going to add the text: "Hello World" and save the file. Once the popup shows to name your file, create a folder where you are at and name the folder "Portfolio" and then name your text file "index.html" and save it.

Reopen Visual Studio with this new folder by either closing the app and opening in the folder path by right-clicking or by clicking the "file" tab in the Visual Studio window and proceeding from there.



Once done your new layout should look like this.
If you have a similar layout the bravo you are ready! Assuming you have downloaded the Live Server extension down in the right hand corner you should see "Go Live". If so click that and watch what happens next.



You should have your webpage showing! It will say "Hello World" just as you put in the code.
Now that we have our HTML doc created let's dig deeper. Replace Hello world with the text

“html:5” When the drop down shows click enter and your new Visual Studio page should look like this:

Headings & Comments

HTML includes six levels of headings which are ranked according to relativity. **(1-6) <h1>, <h2>, <h3> ….** In order to properly use this tags we need make sure we close out the element's tag out.

So if you use the H1 tag you would do this <h3> Hello World </h3> and that would be the proper way to use the tag.

For you, inside the body tag, place in an *H1* and *H3* element inside the first put your name. In the second place in your new career title "Web Designer".

Ex.)
<body>
  <h1>Astro Clare Technology</h1>
  <h3>Web Designer</h3>
</body>

Once you have done this and see it reflected in your browser lets add a comment. The cool thing about comments is that it does not affect your code when done properly. SO, you can add as many comments as you want in your code for any coding language.

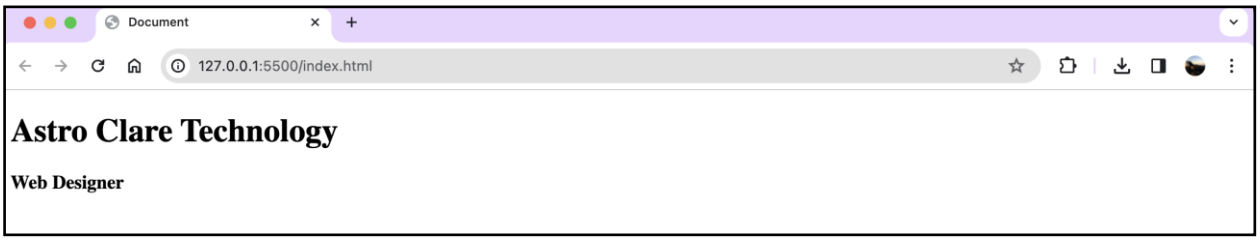
Comments are powerful in coding because it helps you and other developers understand your code and where your thinking was at with execution. Above your H1 element, lets add a comment to let someone know this is your header for you webpage. A comment is initiated with and angled bracket exclamation point and then 2 hyphens, it is then closed with 2 hyphens and one angled bracket. Similar to elements.

Ex.) <! - - this is a comment - - >

After you add your code and refresh your browser you should notice that your comment is not showing in your code, congrats you have successfully added a comment in your code for the next developer to know where your heading section is at. Your code should now look something like this.

```
8   <body>
9       <!-- This is the header section for my webpage.-->
10      <h1>Astro Clare Technology</h1>
11      <h3>Web Designer</h3>
12  </body>
13  </html>
```

Your webpage should look something similar to this now:



Elements

All HTML documents are made up of HTML elements. These start with and **start tag <element>** and end with the **closing tag </element>.** Some elements do not require an closing tag, these are known as single elements. An example of an single element would be the break tag, represented as such **</br >**. To gather a more clear logical understanding, HTML documents are just scripting, or branching with elements to create a web page.

Paragraphs

To create a n paragraph in HTML, you simply need to use the element, ** **, and close it with ** **. Type a paragraph about your interest in web design. Be sure to close your paragraph out correctly. Save your code and refresh your browser to see your changes.

If done correctly you should see your full paragraph in uniform with the other heading elements.

When typing paragraphs think of writing an essay. When you create a new paragraph in your essay, you are creating a new element. So with each paragraph in your code make sure to close out the old one and then create a new element. Let's place a comment above our paragraph that says " This is start of the main information on the webpage."

```
Ex.)    <!-- This is the start of the main information on the webpage.-->
    <p>I enjoy teaching others how to code and create software applications!</p>
```

Attributes

Attributes are additional information about an element or tag, including modifiers. The value in an attribute is what modifies it. We will learn more about attributes in the CSS section of this book. We can use attributes to change things in text or containers in our webpage. Such as positioning, color, size and many more things.

For example, we will align our h1 and h3 elements to the center of the screen. How? We do this by taking the opening tag and inserting the code " **align="center"** ". This will align our code to the center of the browser screen. Try it!

```
Ex.)    <!-- This is the header section for my webpage.-->
    <h1 align="center">Astro Clare Technology</h1>
    <h3 align="center">Web Designer</h3>
```

<u>This is a table of other attributes you ca add to text</u>

<b> - bold text	<small> - smaller text
- important text	<del> - deleted text
<i> - italic text	<ins> - inserted text
- emphasized text	<sub> - subscript text
<mark> - marked text	<sup> - superscript text

Images

Adding images to your webpage will be essential as you progress as a developer. This is really fun to do!

For the example in this book we will use an avatar generator to add an image to our portfolio. Head to the website Avatar Maker (Link To Avatar Creator), and create an avatar once you are finished with that, follow the steps below to increase your skill level by adding this image to your webpage.

When working with external documents and files in HTML it is good practice to have relative directories, or folders, for each type. For now, let's just go to our Portfolio folder and add a new folder with the name "images". Then download or place your avatar image in that folder.

The correct way to add an image to your HTML file is by adding the image element. This is done with the following code:

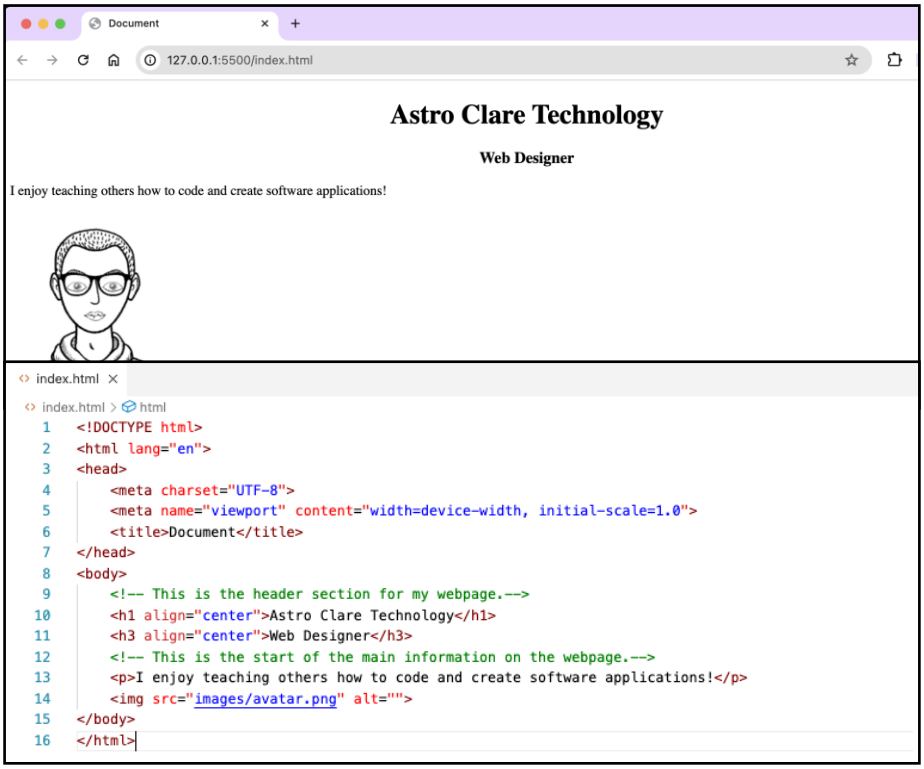
Ex.) <img src=" " alt=" " >

There are some key things to note when using the image element. **Img** tells the HTML document what to look for, in this case an image. The **src** attribute is telling the HTML file where he source of the image is coming from. In this case ours will be "images/avatar.png". **Alt** tells the HTML file the alternate name for the image in the event the image doesn't load properly. This helps users who may have difficulty with sight, or using features such as, text-to-talk, understand what is being displayed.

In our case we will use the tag as such:

<img src="images/avatar.png" alt="Image of the web designer as an avatar.">

Your webpage and should now look similar to something like this:



List & Tables

Lists

In HTML, lists serve a variety of use. You have order lists that are initiated with the tag **<ol>** as well as unordered lists that are initiated with the tag **<ul>**. Both unordered and ordered list need to have list items in them. To use these we simply add **<li>**, for each item. For your portfolio let's add an unordered list, with 4 list items. This will later become our navigation menu when styling in CSS.

Be sure to add a comment above the list element to let other developers know, this is a navigation menu container.

For the 4 item names let's add the following: **home, experience, services, contact**. Your code and web page should now look like this:

Ex.)

```
 <!-- This is the start of the
navigation menu container.-->
   <ul>
      <li>home</li>
      <li>experience</li>
      <li>services</li>
      <li>contact</li>
   </ul>
```

Tables

Tables are used for many cases, sometimes you can use a table instead of a list and vice versa. For your portfolio we will not be adding a table but it is great to know about them.

In HTML you create a table with **<table>**, then we dived the table into rows **<tr>**, then we divide the rows into columns, in HTML these are called table data columns or table data for short, **<td>.** You can add anything you want inside of the table data section. Images, text, links etc.

A quick example of this is:

```
<table>
      <tr>
            <th> Header 1 </th>
            <th> Header 2 </th>
      </tr>
      <tr>
            <td> Column 1 </td>
            <td> Column 2 </td>
      </tr>
</table>
```

****** This demonstrates a table with 2 rows and 2 columns. TRY IT! ******

Links

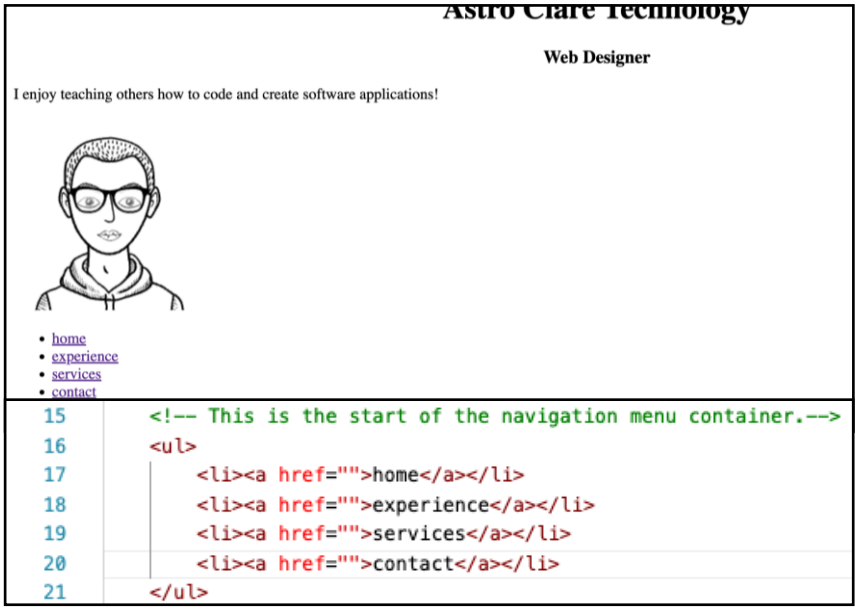
Links are one of the most essential parts of your web page. You can add links to images or text that will give your user the ability to click on them to direct to another file, web page or hierarchy. We add links using the ** Name for the link** attribute.

You can add an attribute to the tag element called blank. Adding this element is best used for links that are outside of your web page's dictionary of directories. Adding this attribute is simple yet affective, simply place _**blank** after the last quotation of your links url.
Ex.) ** YouTube

Above, we created an unordered list. Let's turn each list item into an link, these are going to be links that lead to no where for now until we get the rest of the portfolio built.

For the text in each list, wrap the link tag around the text and not the list tag.
Your code and portfolio should now look like this:

Ex.)



Block & Inline Elements

Block Elements

HTML block elements are elements that start a new “line” or section of information. Examples of block elements are : <h1>, <p>, <table>, , <form> etc.

For your portfolio we are going to transform some of the code we have built already. We are going to create some block elements, and build **div containers.** The best thing you have done to make this possible is create comments in your code.

We will be using the ** ** element quite frequently in web development and some forms of software engineering. This is a block element that is used for containers for other HTML elements. It is greatly combined with **CSS** for styling as well as **JavaScript** for bringing life to our page. When you are creating a div element you have to remember some key components of making one, let’s build a div container for our header and then break down the key aspects of it.

*** *Copy this Code if you need help* ***

```
   <!-- This is the header section for my webpage.-->
<div class=“header’ id=“header”>
   <h1 align="center">Astro Clare Technology</h1>
   <h3 align="center">Web Designer</h3>
</div>

<div class=“header’ id=“header”>
```

You have the opening tag **div.** Then the **class** name this is what we use mainly for CSS styling, which is identified in CSS as **.header,** note the period symbol. Lastly, we have the **id** name, this is used for JavaScript, which is identified in CSS as **#header**, note the hashtag symbol. We will dive deeper into CSS and JavaScript in later lessons but for now this is what you need to know to create styling sheets for your elements.

If you have been following along, you should have 3 comment sections. Let’s create 4 div containers: header, main, navigation, footer. Move your navigation area above your main section.

```
<body>
    <!-- This is the header section for my webpage.-->
    <div class="header" id="header">
        <h1 align="center">Astro Clare Technology</h1>
        <h3 align="center">Web Designer</h3>
    </div>
    <!-- This is the start of the navigation menu container.-->
    <div class="navigation" id="navigation">
        <ul>
            <li><a href="">home</a></li>
            <li><a href="">experience</a></li>
            <li><a href="">services</a></li>
            <li><a href="">contact</a></li>
        </ul>
    </div>
    <!-- This is the start of the main information on the webpage.-->
    <div class="main-section" id="main-section">
        <p>I enjoy teaching others how to code and create software applications!</p>
        <img src="images/avatar.png" alt="" >
    </div>
    <div class="footer" id="footer"> <p>Footer Section!</p> </div>
</body>
```

What we have done here is created 4 div containers, turning our multiple elements into children elements of their respective parent containers.

For example, instead of us having our name and career, in the header section, as 2 block elements that we style separately, when we place them in a header container we can style them at the same time with the same code without interfering with other elements that are the same in other containers or areas. Once again in the CSS book we will dive deeper into why that is so important.

Making our code manageable and easily readable is most important. Creating the div elements has increased our readability and organizational aspect of our webpage.

Here are a few more examples of block elements. In the JavaScript book, we will learn how to use the “script” block element inline.

APPLET – embedded java applet	Iframe- inline frame
INS – inserted text	MAP – image map
OBJECT – embedded object	SCRIPT – script within HTML doc

Inline Elements

Inline elements are displayed without breaks in lines. Examples of this are <b>, <a>, , <img>, <input>. Some of your inline elements will require some CSS styling, others such as <b>, <i> “italic”, are freestanding when it comes to their usage.

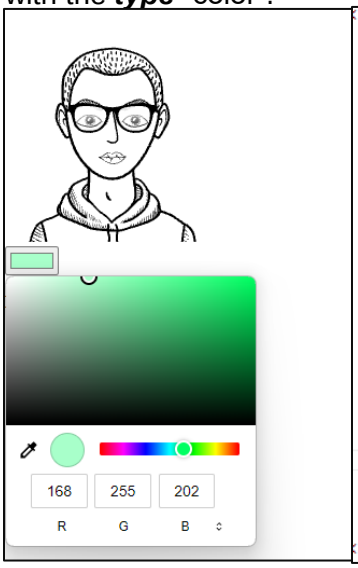
Forms

Collecting user information is very important in HTML. Capturing information with input has been increasingly easy over the past few years with many contact form websites being used. It's always good to learn how to code and style your own forms for collecting information especially if you plan on being a full stack developer.

We have 2 method to use when collecting user information, **GET** and **POST.** When using the GET method the form data will be visible in the page address. Using POST is best for updating data or if sensitive information is being captured. The security is better due to the submitted data not bein g visible in the page address. To have the data collected and sent to your server from the form you will need to have a way to take user input. We do this with the **<input>** element. With this element we can take text, password, urls, uploads and many more!

Another key attribute to use in your form is the *action* attribute. We use this to redirect users after they submit a form. For your project. In our code we are going to add a very basic form into our code to create a color wheel!

Let's begin with creating the form block element and then we will add an input element with the ***type*** "color".




```
<body>
    <!-- This is the header section for my webpage.-->
    <div class="header" id="header">
        <h1 align="center">Astro Clare Technology</h1>
        <h3 align="center">Web Designer</h3>
    </div>
    <!-- This is the start of the navigation menu container.-->
    <div class="navigation" id="navigation">
        <ul>
            <li><a href="">home</a></li>
            <li><a href="">experience</a></li>
            <li><a href="">services</a></li>
            <li><a href="">contact</a></li>
        </ul>
    </div>
    <!-- This is the start of the main information on the webpage.-->
    <div class="main-section" id="main-section">
        <p>I enjoy teaching others how to code and create software applications!</p>
        <img src="images/avatar.png" alt="" >
        <form action="">
            <input type="color" name="" id="">
        </form>
    </div>
    <div class="footer" id="footer"> <p>Footer Section!</p> </div>
</body>
```

Try out your color wheel !! Other inputs are available such as checkboxes, radio buttons, submit buttons and more! HTML5 updated a new attribute for forms called placeholder. For input and textarea elements, placeholder offers a *hint* for the user, to let them know what information to enter into that field. Best practice to use for your input fields and information gathering fields is the required attribute. Simply add required at the end of your input element. This will force the user to have input information in order to continue.

Here is a Form cheat sheet for the different input element types you can use.

INPUT TYPES	ATTRIBUTE TYPES
- color - date - datetime - datetime-local - email - month - number - range - search - tel - time - url - week	- autofocus - form - formaction - formenctype - formmethod - formnovalidate - formtarget - height and width - list - min and max - multiple - pattern (regexp) - placeholder - required

Websites like https://formsubmit.co/ and many others help create forms.

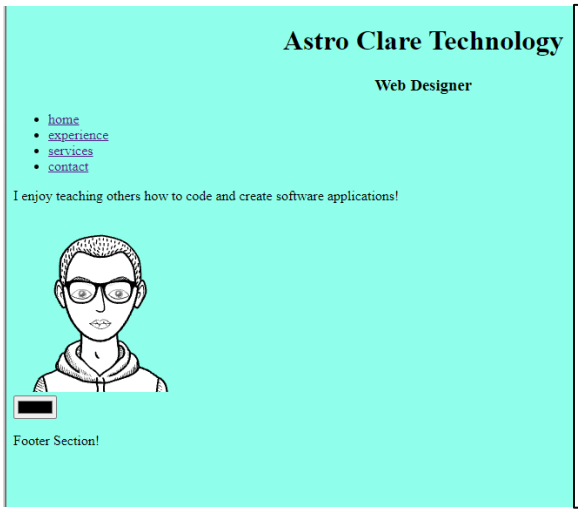
➢ **Google Forms**: Google Forms is a popular option for creating surveys, quizzes, and forms. It's user-friendly and integrates well with other Google services.
➢ **Typeform**: Typeform offers beautifully designed forms with a focus on a conversational interface, making the form-filling experience more engaging.
➢ **JotForm**: JotForm allows users to create forms with a drag-and-drop interface. It offers a variety of templates and customization options.
➢ **Formstack**: This platform offers form-building tools and also allows for easy integration with other applications.
➢ **Wufoo**: Wufoo is known for its user-friendly interface and customization options for creating online forms.
➢ **Cognito Forms**: This platform provides flexible and powerful forms for a variety of uses, with customization and integration options.

Colors

HTML colors are used with hexadecimal characters:
0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

We use the wheel of red, green and blue light **(RGB)**, when determining our colors. Hex values are written using the hashtag symbol **(#),** followed by 3 or 6 hex characters. This is supported by all browsers. There are over 16 million color combinations using the hex values. I want you to go your body tag and add the attribute “bgcolor” we are going to CHANGE THE COLOR OF OUR SITE!

Go to your color wheel and pick a color that will show the black text on your site well. Your code and web page should look something similar to this.




```
<body bgcolor="#8FFFEC">
    <!-- This is the header section for my webpage.-->
    <div class="header" id="header">
        <h1 align="center">Astro Clare Technology</h1>
        <h3 align="center">Web Designer</h3>
    </div>
    <!-- This is the start of the navigation menu container.-->
    <div class="navigation" id="navigation">
        <ul>
            <li><a href="">home</a></li>
            <li><a href="">experience</a></li>
            <li><a href="">services</a></li>
            <li><a href="">contact</a></li>
        </ul>
    </div>
    <!-- This is the start of the main information on the webpage.-->
    <div class="main-section" id="main-section">
        <p>I enjoy teaching others how to code and create software applications!</p
        <img src="images/avatar.png" alt="" >
        <form action="">
            <input type="color" name="" id="">
        </form>
    </div>
    <div class="footer" id="footer"> <p>Footer Section!</p> </div>
</body>
```

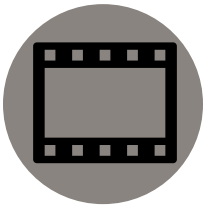
Audio & Video

In HTML5 the **<audio>** and **<video>** element was introduced, this enables us as developers to embed audio in our web pages. The <audio and <video> element creates an audio player inside the browser, the text in the element will display if there is an failure in your audio files. There are two ways to specify the source url for the audio file. We can do this by a similar method to finding images and that is by the source attribute.

Attributes of the <audio> (or <video>) element are ***controls***, ***loop*** and ***autoplay***. Controls specifies that audio controls should be displayed such as play/pause button etc. **Autoplay** start playing the file as soon as it is ready without the user permission. **Loop** will cause the audio to replay every time it is finished There are 3 supported file formats for <audio> elements: MP3, WAV and OGG.

<audio src="audio.mp3" controls>
Audio Element Unable to Initiate.
</audio>

<video controls>
<source src="video.mov">
</video>



Outro

Now that you have the fundamentals of using HTML, we highly recommend using the CSS book to deepen your knowledge and skills within web development. Below are some of the things we covered in this book that will help aid your career in software engineering.

Headings & Comments

HTML includes six levels of headings ranked from 1 to 6. Ensure you properly use tags by opening and closing them. For practice, create an H1 and H3 element inside the body tag, displaying your name and new career title. Integrate comments to enhance code understanding.

Elements

All HTML documents consist of HTML elements, starting with an opening tag and ending with a closing tag. Learn about block and inline elements, paragraphs, attributes, images, lists, tables, links, and more. Enhance your coding skills and build a foundation for web development.

Forms

Collecting user information is crucial. Learn about form elements, input types, and attributes. Explore methods like GET and POST, and discover the importance of the action attribute. Incorporate forms into your webpage, ensuring a user-friendly and interactive experience.

Colors, Audio & Video

Explore HTML colors using hexadecimal characters. Add vibrancy to your webpage by changing the background color. Delve into embedding audio and video elements, understanding attributes like controls, loop, and autoplay. Enhance user engagement with multimedia elements.

As you conclude this book, you've taken significant strides toward becoming a proficient coder. Your journey doesn't end here- it's just the beginning. Continue to explore, experiment, and most importantly enjoy the process of bringing your ideas to life through code**.**

HAPPY CODING !