INSTALLATION OF CODE EDITOR & HTML FUNDAMENTALS

Download file and follow along!

Lecture 1, Week 1

TODAY

- ❏ INSTALLATION OF CODE EDITOR
- ❏ HTML FUNDAMENTALS
- ❏ HTML IMAGES, TABLES & FORMS

INSTALLATION OF CODE EDITOR

You probably already have a basic text editor on your computer. By default Windows includes Notepad and macOS comes with TextEdit..

For web development, you can probably do better than Notepad or TextEdit.

Visual Studio Code, a free editor, which offers live previews and code hints is highly recommended.

Open your browser and type in the URL below to download and install Visual Studio Code on your computer: https://code.visualstudio.com

HTML FUNDAMENTALS

HTML stands for **Hypertext Markup Language**. It allows the user to create and structure sections, paragraphs, headings, links, and blockquotes for web pages and applications.

HTML is not a programming language, meaning it doesn’t have the ability to create dynamic functionality. Instead, it makes it possible to organize and format documents, similarly to Microsoft Word.

When working with HTML, we use simple code structures (tags and attributes) to mark up a website page.

For example, we can create a paragraph by placing the enclosed text within a starting ***<p>*** and closing ***</p>*** tag.

HISTORY OF HTML

HTML was invented by **Tim Berners-Lee**, a physicist at the CERN research institute in Switzerland. He came up with the idea of an Internet-based hypertext system.

Hypertext means a text that contains references (links) to other texts that viewers can access immediately. He published the first version of HTML in 1991, consisting of 18 HTML tags.

Since then, each new version of the HTML language came with new tags and attributes (tag modifiers) to the markup.

According to Mozilla Developer Network's **HTML Element Reference**, currently, there are 140 HTML tags, although some of them are already obsolete (not supported by modern browsers).

Due to a quick rise in popularity, HTML is now considered an official web standard. The HTML specifications are maintained and developed by the World Wide Web Consortium (W3C). You can check out the latest state of the language anytime on **W3C's website**.

HTML documents are files that end with a ***.html*** or ***.htm*** extension. You can view them using any web browser (such as Google Chrome, Safari, or Mozilla Firefox). The browser reads the HTML file and renders its content so that internet users can view it.

Usually, the average **website includes several different HTML pages**. For instance: home pages, about pages, contact pages would all have separate HTML documents.

Each HTML page consists of a set of **tags** (also called **elements**), which you can refer to as the building blocks of web pages. They create a hierarchy that structures the content into sections, paragraphs, headings, and other content blocks.

Most HTML elements have an opening and a closing that use the ***<tag></tag>*** syntax.

Below, you can see a code example of how HTML elements can be structured:

```
<div>

 <h1>The Main Heading</h1>

 <h2>A catchy subheading</h2>

 <p>Paragraph one</p>

 <img src="/" alt="Image">

 <p>Paragraph two with a <a
href="https://example.com">hyperlink</a></p>

</div>
```

- The outmost element is a simple division (***<div></div>***) you can use to mark up bigger content sections.
- It contains a heading (***<h1></h1>***), a subheading (***<h2></h2>***), two paragraphs (***<p></p>***), and an image (***<img>***).
- The second paragraph includes a link (***<a></a>***) with a href attribute that contains the destination URL.

- The image tag also has two attributes: ***src*** for the image path and ***alt*** for the image description.

Overviewing The Most Used HTML Tags

HTML tags have two main types: **block-level** and **inline tags**.

- Block-level elements take up the full available space and always start a new line in the document. Headings and paragraphs are a great example of block tags.
- Inline elements only take up as much space as they need and don’t start a new line on the page. They usually serve to format the inner contents of block-level elements. Links and emphasized strings are good examples of inline tags.

Block-Level Tags

The three block level tags every HTML document needs to contain are **<html>**, **<head>**, and **<body>**.

- The **<html></html>** tag is the highest level element that encloses every HTML page.
- The **<head></head>** tag holds meta information such as the page’s title and charset.
- Finally, the **<body></body>** tag encloses all the content that appears on the page.

```
<html>

 <head>

   <!-- META INFORMATION -->

 </head>

 <body>

   <!-- PAGE CONTENT -->

 </body>

</html>
```

- Headings have 6 levels in HTML. They range from ***<h1></h1>*** to ***<h6></h6>,*** where h1 is the highest level heading and h6 is the lowest one. Paragraphs are enclosed by ***<p></p>***, while blockquotes use the ***<blockquote></blockquote>*** tag.
- Divisions are bigger content sections that typically contain several paragraphs, images, sometimes blockquotes, and other smaller elements. We can mark them up using the ***<div></div>*** tag. A div element can contain another div tag inside it as well.
- You may also use ***<ol></ol>*** tags for ordered lists and ***<ul></ul>*** for unordered ones. Individual list items must be

enclosed by the ***<li></li>*** tag. For example, this is how a basic unordered list looks like in HTML:

```
<ul>

  <li>List item 1</li>

  <li>List item 2</li>

  <li>List item 3</li>

</ul>
```

Inline Tags

Many inline tags are used to format text. For example, a

<strong></strong> tag would render an element in **bold**,

whereas *<em></em>* tags would show it in *italics*.

Hyperlinks are also inline elements that require ***<a></a>*** tags and **href** attributes to indicate the link's destination:

```
<a href="https://example.com/">Click me!</a>
```

Images are inline elements too. You can add one using ***<img>*** without any closing tag. But you will also need to use the ***src*** attribute to specify the image path, for example:

```
<img src="/images/example.jpg" alt="Example image">
```

If you want to learn more HTML tags, consider checking w3schools.com/html

So…What is HTML?

HTML is the main markup language of the web. It runs natively in every browser and is maintained by the World Wide Web Consortium.

You can use it to create the content structure of websites and web applications. It's the lowest level of frontend technologies, that serves as the basis for styling you can add with CSS and functionality you can implement using JavaScript.

A Simple HTML Document Example

```
<!DOCTYPE html>
    <html lang="en">
        <head>

            <meta charset="utf-8">
            <title>Page Title</title>

        </head>
        <body>

             <h1>This is a Heading</h1>
             <p>This is a paragraph.</p>
             <p>This is another paragraph.</p>

        </body>

    </html>
```

Example Explained

❑ All HMTL documents begin with the **<!DOCTYPE html>** declaration. This tells the browser that this document is an HTML5 document.

❑ After the <!DOCTYPE html> declaration, the HTML document itself begins with the opening tag of the root element and ends with the closing tag of the root element. In the example above, can you identify the root element of the HMTL document?

❑ The **<html>** element is the root element of an HTML page.

❑ The **<head>** element in an HTML document is the part that is not displayed in the web browser when the page is loaded. It contains information such as the page <title>, links to CSS, links to custom favicons, and other metadata (data about the HMTL, such as the author, and important keywords that describe the document).

❑ The **<title>** element is one of the elements that are enclosed in the <head> element. It specifies a title for the document.

❑ Another element that can be added to the <head> element is the <meta> element. The **<meta>** element specifies metadata i.e the data that describes the web page in a meaningful way than can be understood by web crawlers and browsers.

❑ The <meta> element also specifies the document’s character encoding:

```
<meta charset="utf-8">
```

Utf-8 is a universal character set that includes any character from any human language. Adding the line of code above to your HTML document means that your web page will be able to handle displaying any language.

❑ Just about all websites you’ll see will employ CSS to make them look visually appealing. You can add CSS to an HTML by inserting the **<link>** element in the <head> tag.

❑ The <body> element defines the document’s body, and is a container for all the visible contents you see in web pages.

❑ The <h1> element defines a large heading. HTML headings are defined with the <h1> to <h6> tags. <h1> defines the largest and <h6> defined the smallest heading.

HTML IMAGES, TABLES & FORMS

Images can improve the design and the appearance of a web page.

HTML Images Syntax

The HTML <img> tag is used to embed an image in a web page.

Images are not technically inserted into a web page; images are linked to web pages. The <img> tag creates a holding space for the referenced image.

The <img> tag is empty, it contains attributes only, and does not have a closing tag.

The <img> tag has two required attributes:

- src - Specifies the path to the image
- alt - Specifies an alternate text for the image

```
<img src="url" alt="alternatetext">
```

The src Attribute

The required src attribute specifies the path (URL) to the image.

Note: When a web page loads; it is the browser, at that moment, that gets the image from a web server and inserts it into the page. Therefore, make sure that the image actually stays in the same spot in relation to the web page, otherwise your visitors will get a broken link icon. The broken link icon and the alt text are shown if the browser cannot find the image.

The alt Attribute

The required alt attribute provides an alternate text for an image, if the user for some reason cannot view it (because of slow connection, an error in the src attribute, or if the user uses a screen reader).

The value of the alt attribute should describe the image:

Example

```
<img src="img_chania.jpg" alt="Flowers in Chania">
```

If a browser cannot find an image, it will display the value of the alt attribute:

Example

```
<img src="wrongname.gif" alt="Flowers in Chania">
```

Image Size - Width and Height

You can use the style attribute to specify the width and height of an image.

Example

```
<img src="img_girl.jpg" alt="Girl in a jacket"
style="width:500px;height:600px;">
```

Alternatively, you can use the width and height attributes:

Example

```
<img src="img_girl.jpg" alt="Girl in a jacket" width="500" height="600">
```

The width and height attributes always define the width and height of the image in pixels.

Note: Always specify the width and height of an image. If width and height are not specified, the web page might flicker while the image loads.

Example

```
<!DOCTYPE html>

<html>

        <head>

                <style>

                img {

                  width: 100%;
        }

        </style>

    </head>

<body>

    <img src="html5.gif" alt="HTML5 Icon" width="128" height="128">

    <img src="html5.gif" alt="HTML5 Icon"
    style="width:128px;height:128px;">

</body>

</html>
```

Images in Another Folder

If you have your images in a sub-folder, you must include the folder name in the src attribute:

Example

```
<img src="/images/html5.gif" alt="HTML5 Icon"
style="width:128px;height:128px;">
```

Images on Another Server/Website

Some web sites point to an image on another server.

To point to an image on another server, you must specify an absolute (full) URL in the src attribute:

Example

```
<img src="https://www.w3schools.com/images/w3schools_green.jpg"
alt="W3Schools.com">
```

Common Image Formats

Here are the most common image file types, which are supported in all browsers (Chrome, Edge, Firefox, Safari, Opera):

Abbreviation	File Format	File Extension
APNG	Animated Portable Network Graphics	.apng
GIF	Graphics Interchange Format	.gif
ICO	Microsoft Icon	.ico, .cur
JPEG	Joint Photographic Expert Group image	.jpg, .jpeg, .jfif, .pjpeg, .pjp
PNG	Portable Network Graphics	.png
SVG	Scalable Vector Graphics	.svg

HTML TABLES

HTML tables allow web developers to arrange data into rows and columns.

Example

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy

Define an HTML Table

The `<table>` tag defines an HTML table.

Each table row is defined with a `<tr>` tag. Each table header is defined with a `<th>` tag. Each table data/cell is defined with a `<td>` tag.

By default, the text in `<th>` elements are bold and centered.

By default, the text in `<td>` elements are regular and left-aligned.

Example

A simple HTML table:

```
<table style="width:100%">
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

HTML FORMS

An HTML form is used to collect user input. The user input is most often sent to a server for processing.

The <form> Element

The HTML <form> element is used to create an HTML form for user input:

```
<form>

.

form elements

.

</form>
```

The <input> Element

The HTML <input> element is the most used form element.

An <input> element can be displayed in many ways, depending on the type attribute.

Here are some examples:

Type	Description
<input type="text">	Displays a single-line text input field

<input type="radio">	Displays a radio button (for selecting one of many choices)
<input type="checkbox">	Displays a checkbox (for selecting zero or more of many choices)
<input type="submit">	Displays a submit button (for submitting the form)
<input type="button">	Displays a clickable button

The <label> Element

Notice the use of the `<label>` element in the example above.

The `<label>` tag defines a label for many form elements.

The `<label>` element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focus on the input element.

The `<label>` element also help users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the `<label>` element, it toggles the radio button/checkbox.

The `for` attribute of the `<label>` tag should be equal to the `id` attribute of the `<input>` element to bind them together.

Radio Buttons

The `<input type="radio">` defines a radio button.

Radio buttons let a user select ONE of a limited number of choices.

Example

A form with radio buttons:

```
<form>
  <input type="radio" id="male" name="gender" value="male">
  <label for="male">Male</label><br>
  <input type="radio" id="female" name="gender" value="female">
  <label for="female">Female</label><br>
  <input type="radio" id="other" name="gender" value="other">
  <label for="other">Other</label>
</form>
```

Checkboxes

The `<input type="checkbox">` defines a checkbox.

Checkboxes let a user select ZERO or MORE options of a limited number of choices.

Example

A form with checkboxes:

```
<form>
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> I have a car</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label for="vehicle3"> I have a boat</label>
</form>
```

The Submit Button

The `<input type="submit">` defines a button for submitting the form data to a form-handler.

The form-handler is typically a file on the server with a script for processing input data.

The form-handler is specified in the form's `action` attribute.

Example

A form with a submit button:

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>

  <input type="text" id="lname" name="lname" value="Doe"><br><br>

  <input type="submit" value="Submit">

</form>
```

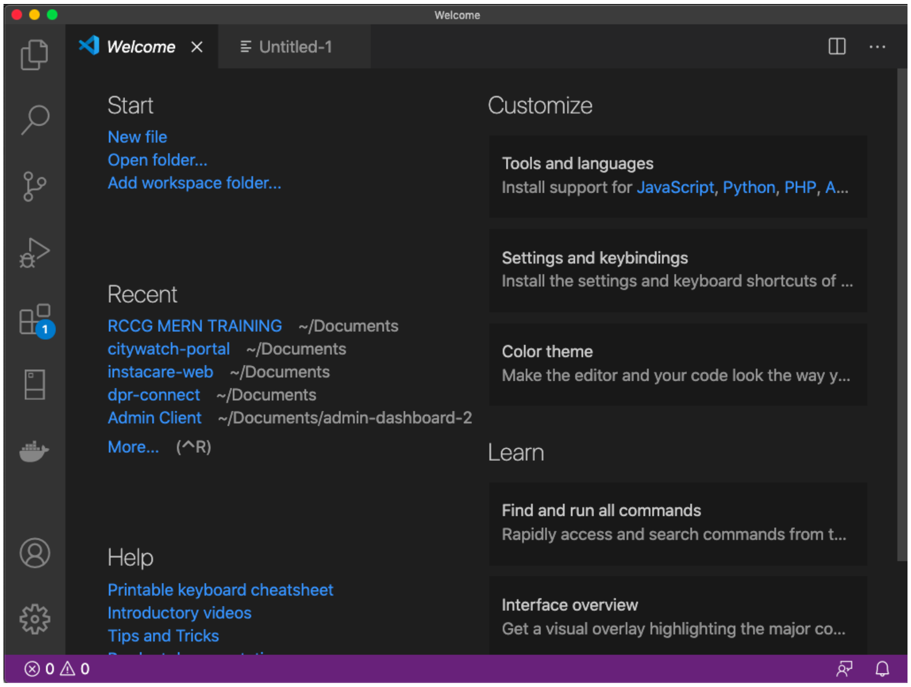
EXERCISE

You are to create an HTML Document that includes the following elements.

1. DIV
2. HEADING
3. SUB HEADING
4. PARAGRAPH
5. AN IMAGE

STEPS

I. Open VS Code Editor on your Laptop



Start

New file

Open folder...

Add workspace folder...

Recent

RCCG MERN TRAINING ~/Documents

citywatch-portal ~/Documents

instacare-web ~/Documents

dpr-connect ~/Documents

Admin Client ~/Documents/admin-dashboard-2

More... (^R)

Help

Printable keyboard cheatsheet

Introductory videos

Tips and Tricks

Customize

Tools and languages
Install support for JavaScript, Python, PHP, A...

Settings and keybindings
Install the settings and keyboard shortcuts of ...

Color theme
Make the editor and your code look the way y...

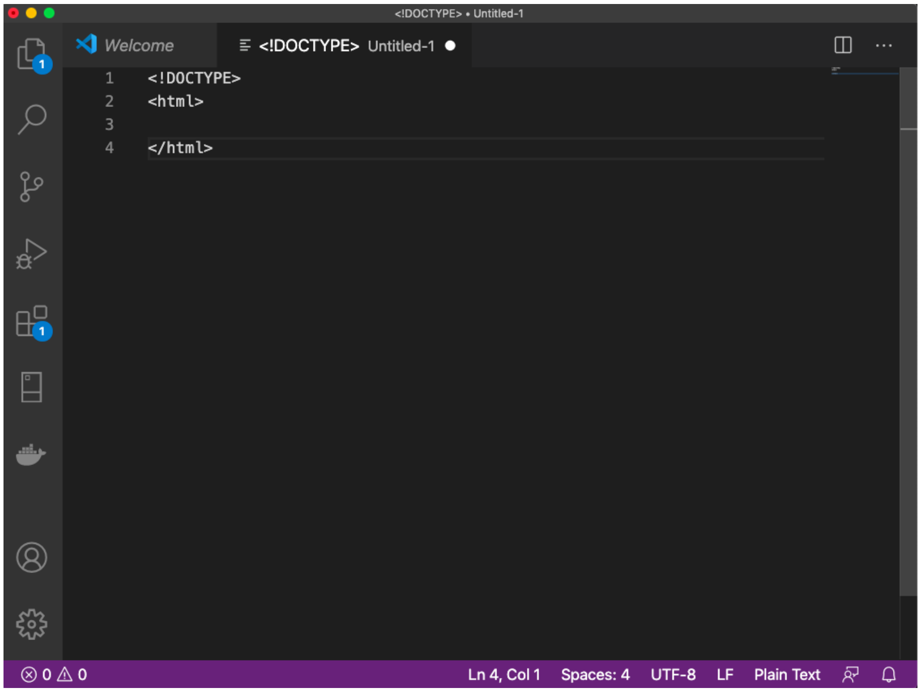
Learn

Find and run all commands
Rapidly access and search commands from t...

Interface overview
Get a visual overlay highlighting the major co...

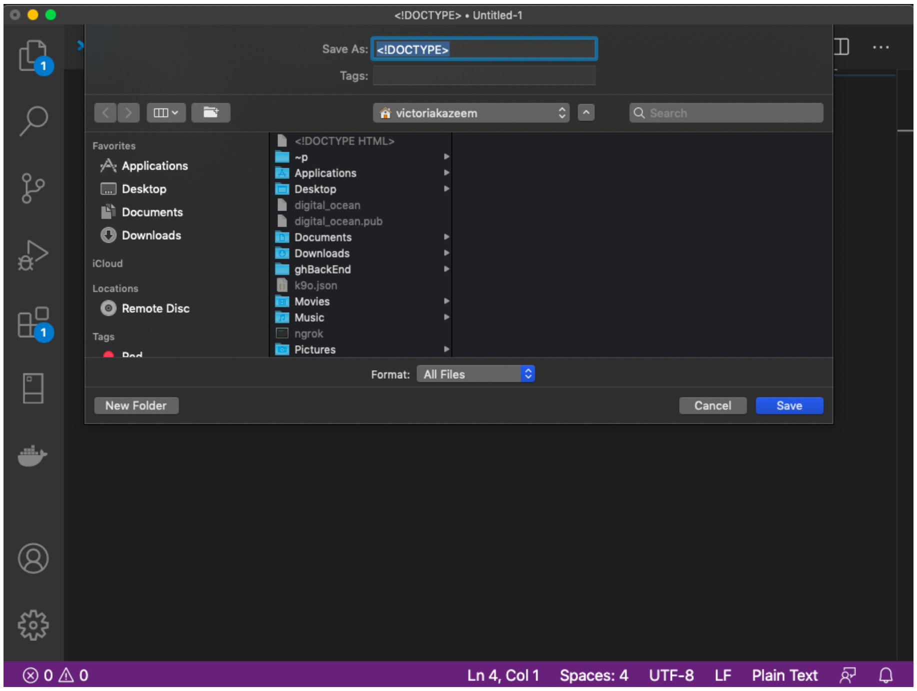
II. On the “New file” link to create a new file. Or Go to the File Menu at the top of the page and select “New File”.

III. In the new file you created, begin typing by making the <!DOCTYPE html> declaration and adding the root element of every web page.



```
<!DOCTYPE>
<html>

</html>
```

IV. Go to your file menu and click on the “Save As” sub-menu.



Save As: <!DOCTYPE>

Tags:

Format: All Files

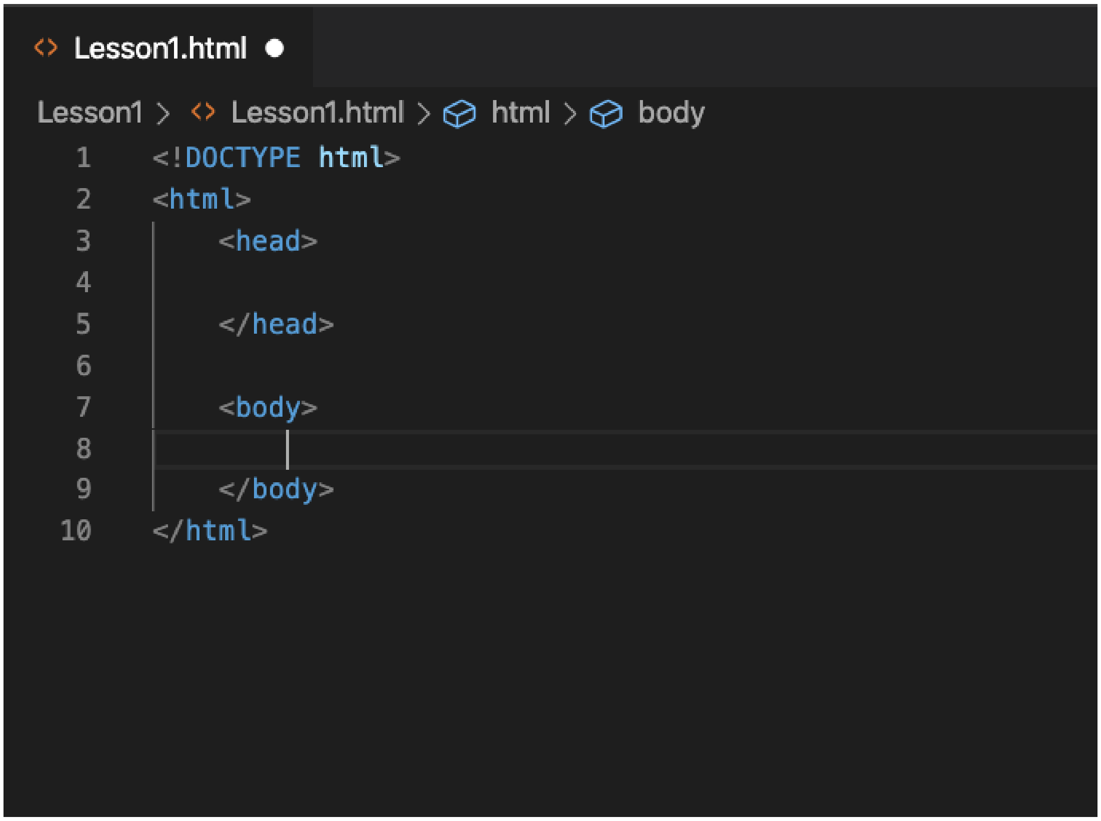
V. Click on your “Documents” folder in the popup that is displayed to you, then on the same popup, you’ll a see a button “New Folder”.

VI. Click on the “New Folder” button and name it “TWS Web Dev Training”.

VII. Clink on the “Enter” key on your keyboard, then double click on the folder you just created.

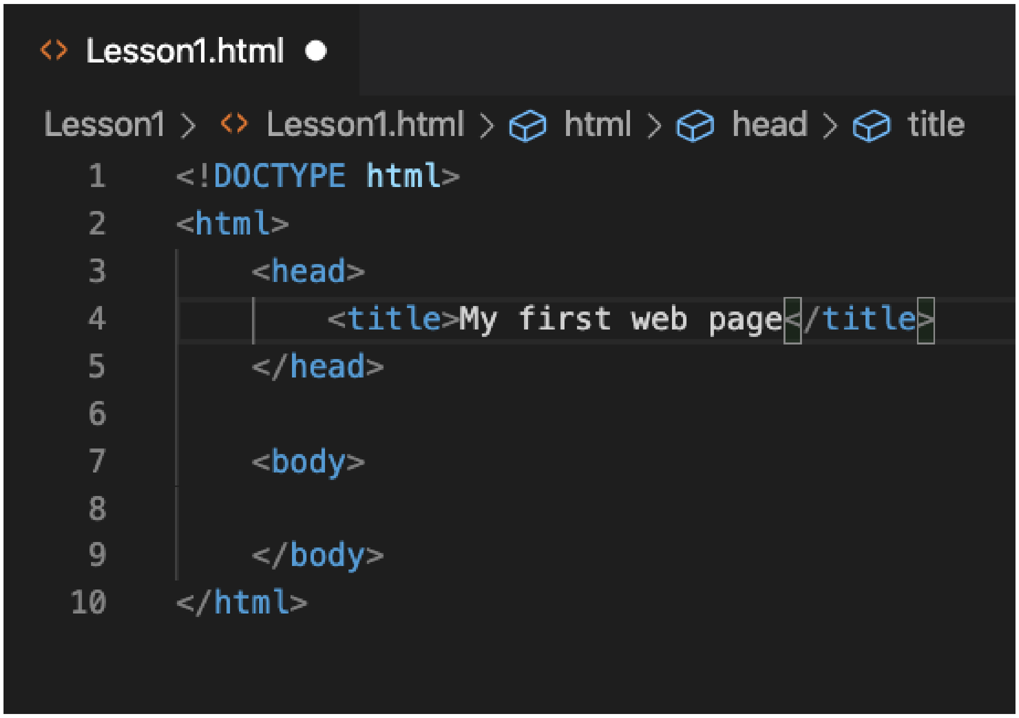
VIII. The popup will still be open because we are yet to save the file in the new folder we just created. Type “Lesson1” in the “File Name” field and in the “Save as type” options, select HTML.

IX. In the lesson1.html file, inside the root element, i.e in between the opening and closing html tags, add the head tags and the body tags.



```
<!DOCTYPE html>
<html>
    <head>

    </head>

    <body>

    </body>
</html>
```

X. Add your title tag inside the head element.



```
<!DOCTYPE html>
<html>
    <head>
        <title>My first web page</title>
    </head>

    <body>

    </body>
</html>
```

XI. In your body element, add a div element.

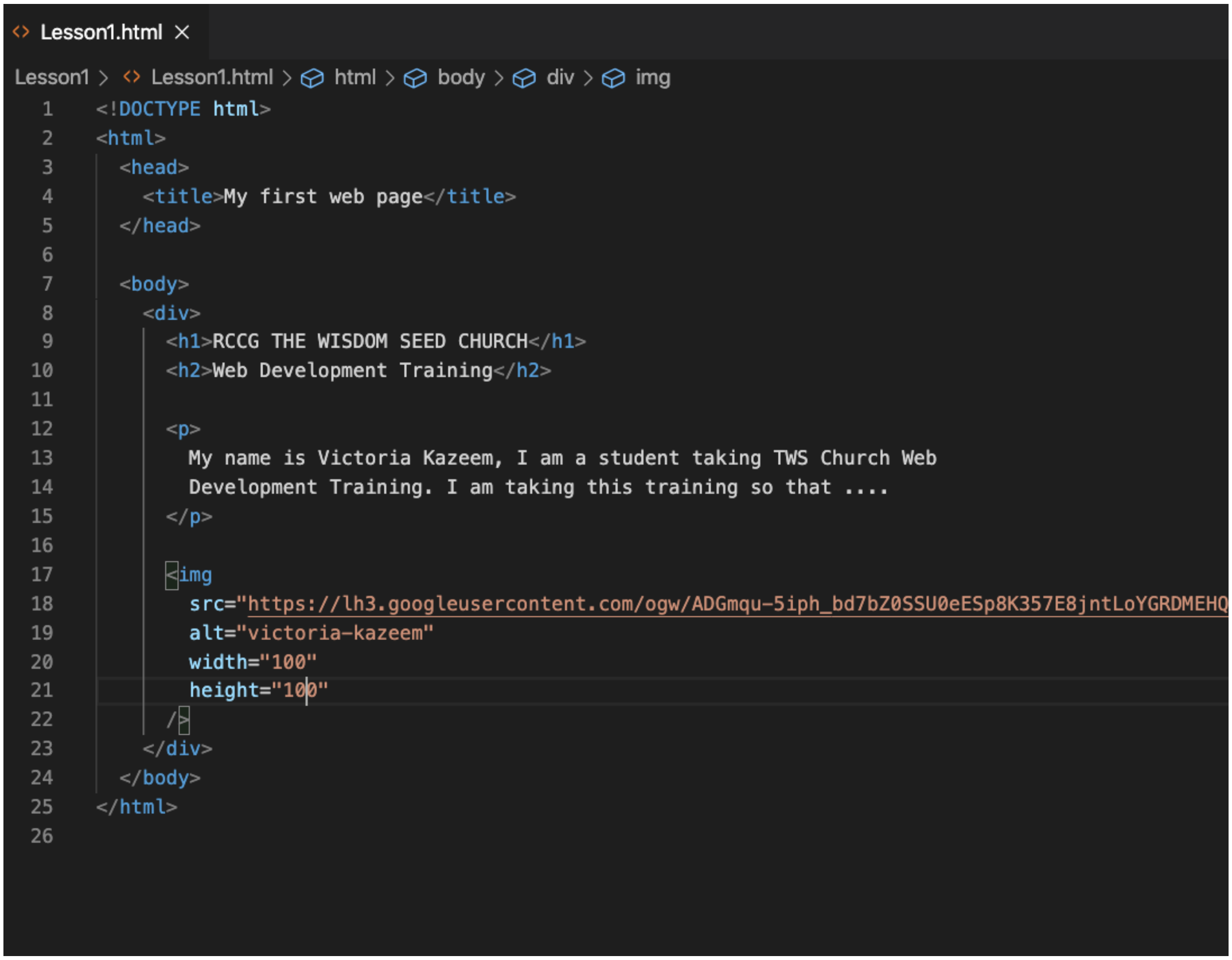
XII. Inside the div, add 1 heading and 1 subheading.

XIII. The text in the heading should be “RCCG THE WISDOM SEED CHURCH”

XIV. The text in the subheading should be “Web Development Training”

XV. Add a paragraph element. The content of this paragraph should be a brief introduction of yourself and why you are taking this training.

XVI. Add an image element, give it a width of “100” and a height of “100”.

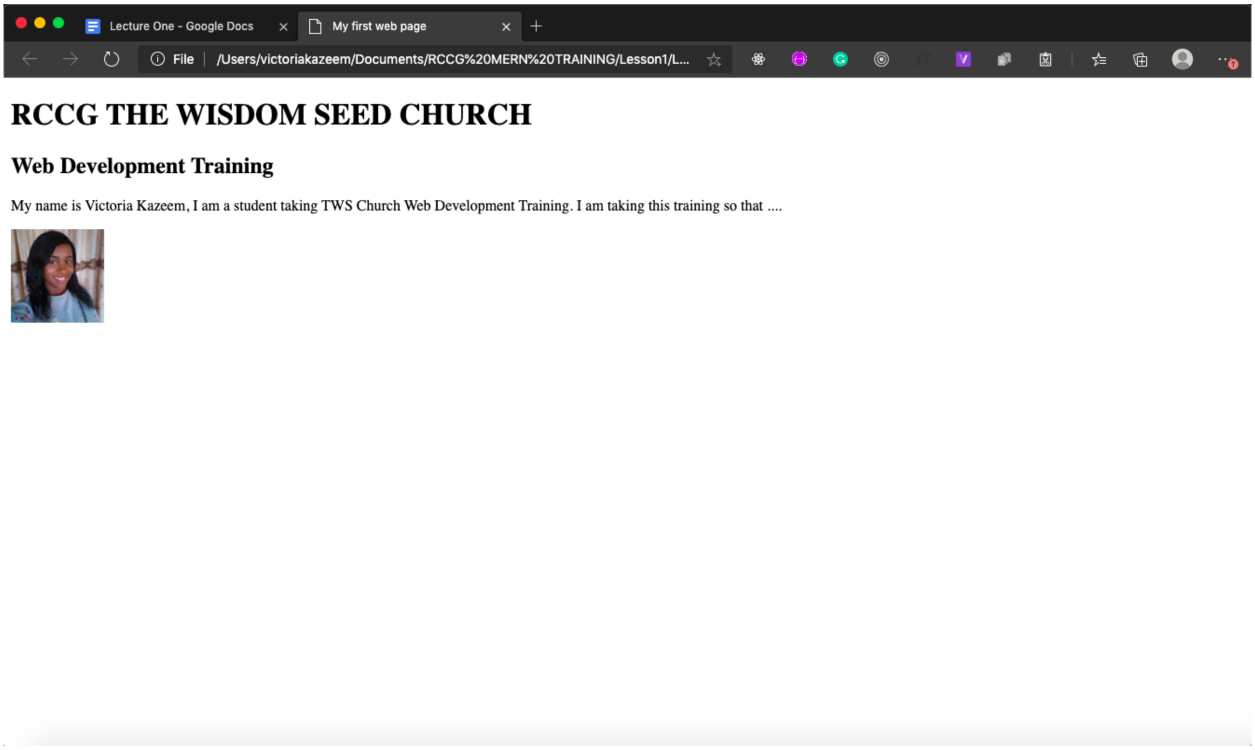


```html
<!DOCTYPE html>
<html>
  <head>
    <title>My first web page</title>
  </head>

  <body>
    <div>
      <h1>RCCG THE WISDOM SEED CHURCH</h1>
      <h2>Web Development Training</h2>

      <p>
        My name is Victoria Kazeem, I am a student taking TWS Church Web
        Development Training. I am taking this training so that ....
      </p>

      <img
        src="https://lh3.googleusercontent.com/ogw/ADGmqu-5iph_bd7bZ0SSU0eESp8K357E8jntLoYGRDMEHQ
        alt="victoria-kazeem"
        width="100"
        height="100"
      />
    </div>
  </body>
</html>
```

This is what your file should look like when you are done.

Go to your file explorer > Document > TWS Web Dev Training and double-click on your Lesson1.html file to open it in your browser.

This is what mine looks like in my browser.


Lecture One - Google Docs
My first web page
File
/Users/victoriakazeem/Documents/RCCG%20MERN%20TRAINING/Lesson1/L...

RCCG THE WISDOM SEED CHURCH

Web Development Training

My name is Victoria Kazeem, I am a student taking TWS Church Web Development Training. I am taking this training so that ....

ASSIGNMENT.

1. Inside your Lesson1.html file, add the following elements:
   (i) A TABLE (ii) A FORM

2. What does HTML stand for?

3. What does the <!DOCTYPE html> signify?

4. What is the root element of every web page?

5. Go to w3schools.com/html and read more about HTML. Write some short notes on what you learn.