

# In Class Workshop: HTML

COM EM757

Dr. Ayse D. Lokmanoglu

October 16th 2024



# Recap what is HTML

- HTML stands for **HyperText Markup Language**
- It's the **language of the web**, used to structure content on web pages
- In this lecture, we'll cover:
  - Basic structure of an HTML document
  - Common HTML tags
  - How to use HTML for layout and content

**Now everyone open their VS Code from their folder**

# HTML lingo

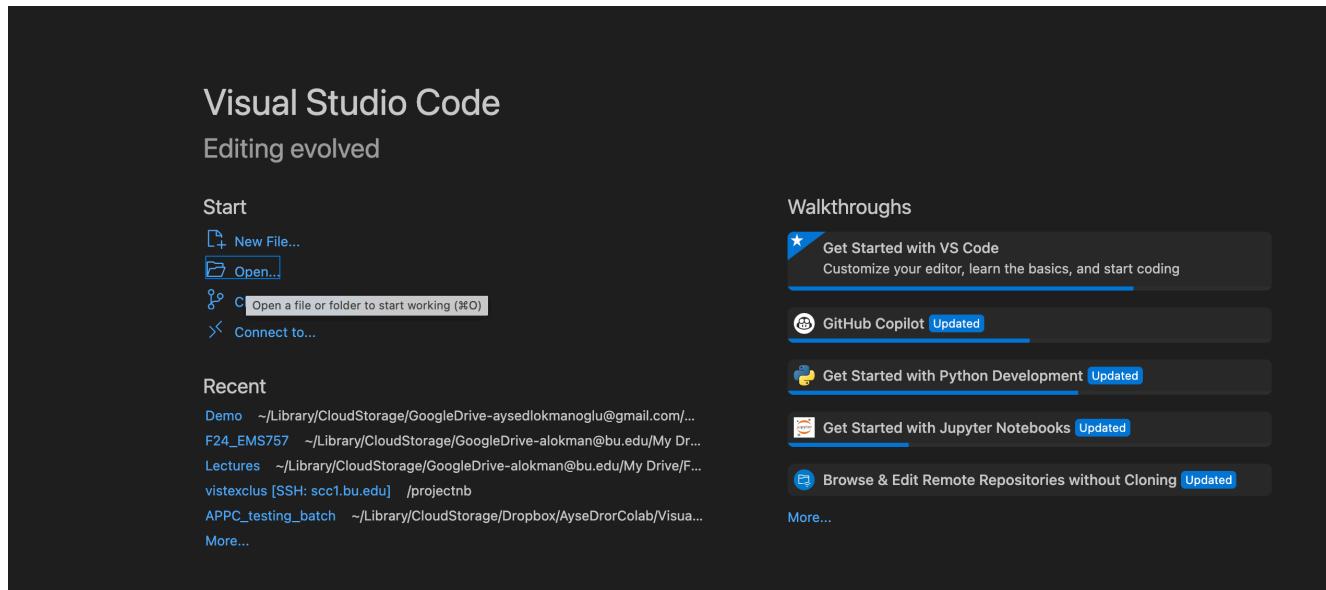
The content and structure of information on a web page

- Wraps text content with opening `<tag>` and closing `</tag>` pairs (think like paranthesis with text).
- The name within the tags is called an **element**.  
The syntax: `<element> content </element>`  
E.g.: `<p>This is a paragraph</p>`

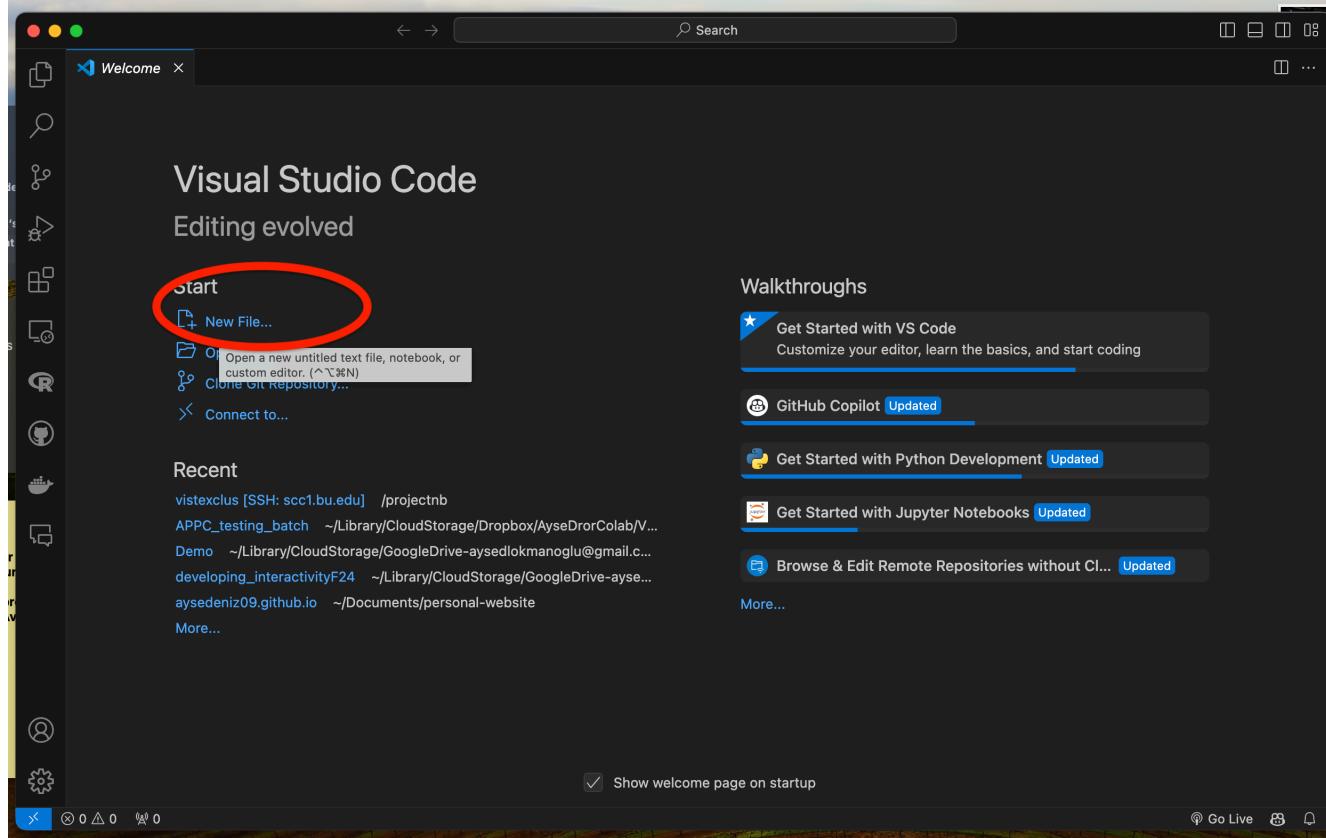
# Recap from Lecture 3

## First let's open our working space

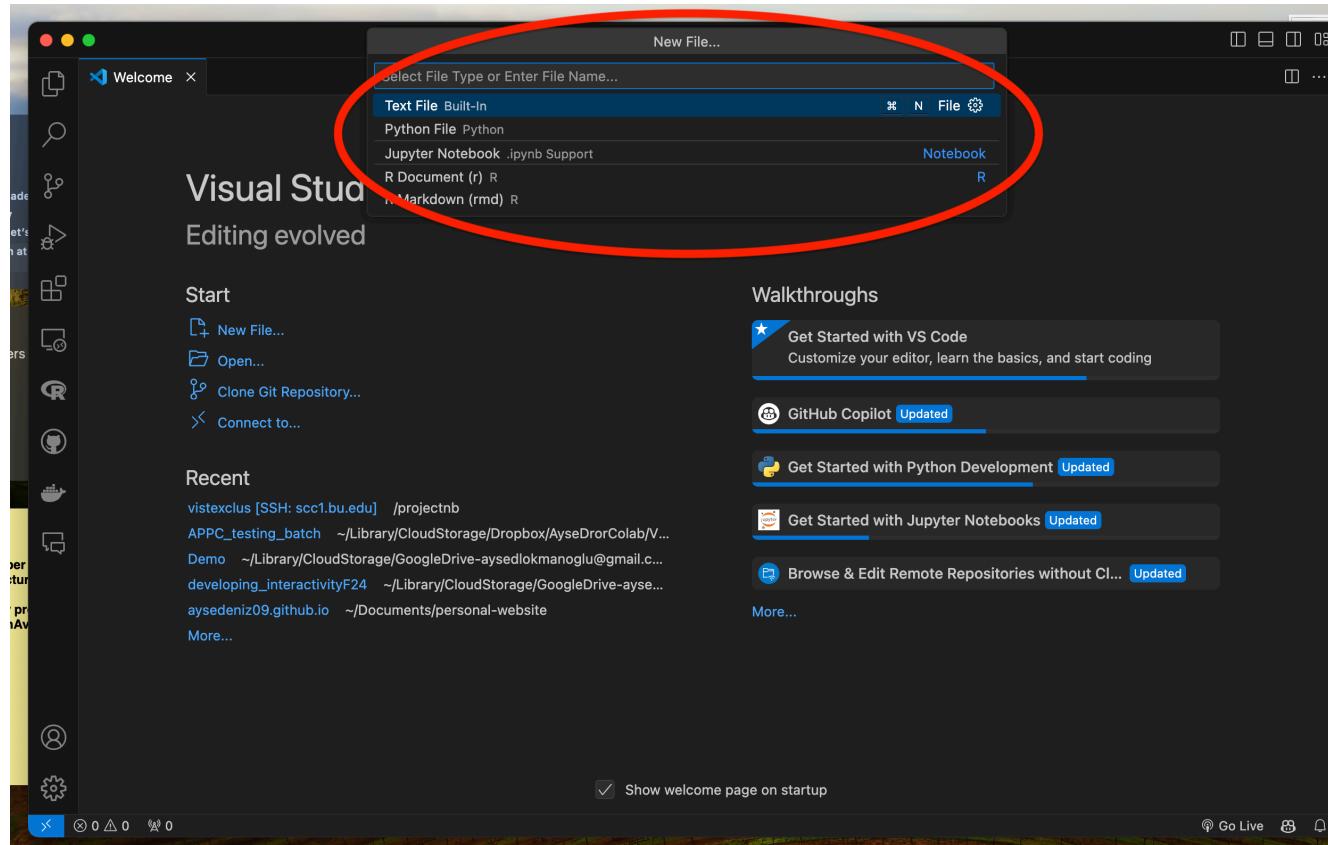
Open Folder and create a subfolder called images



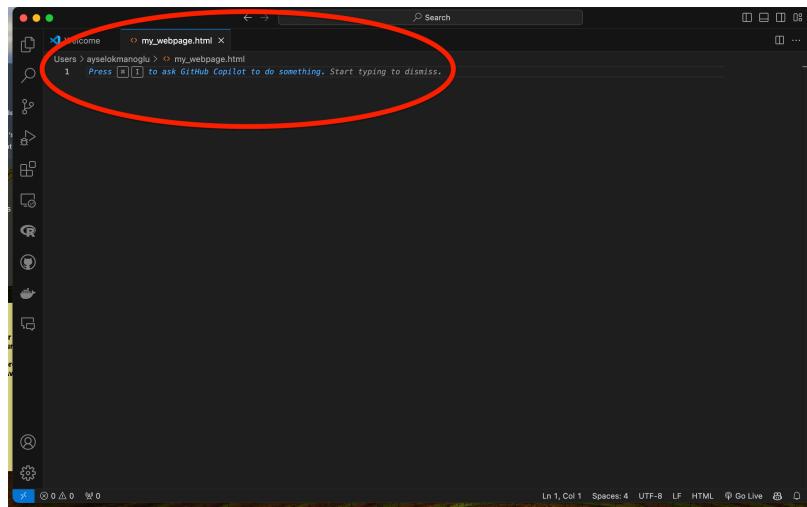
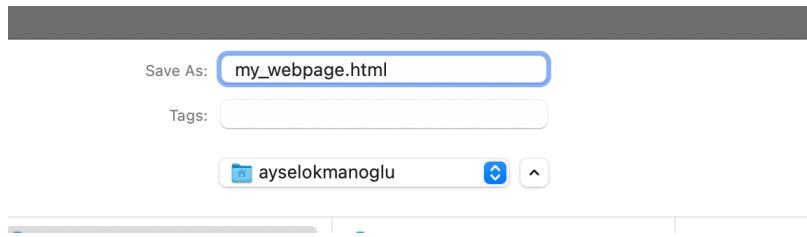
# Let's open a text file



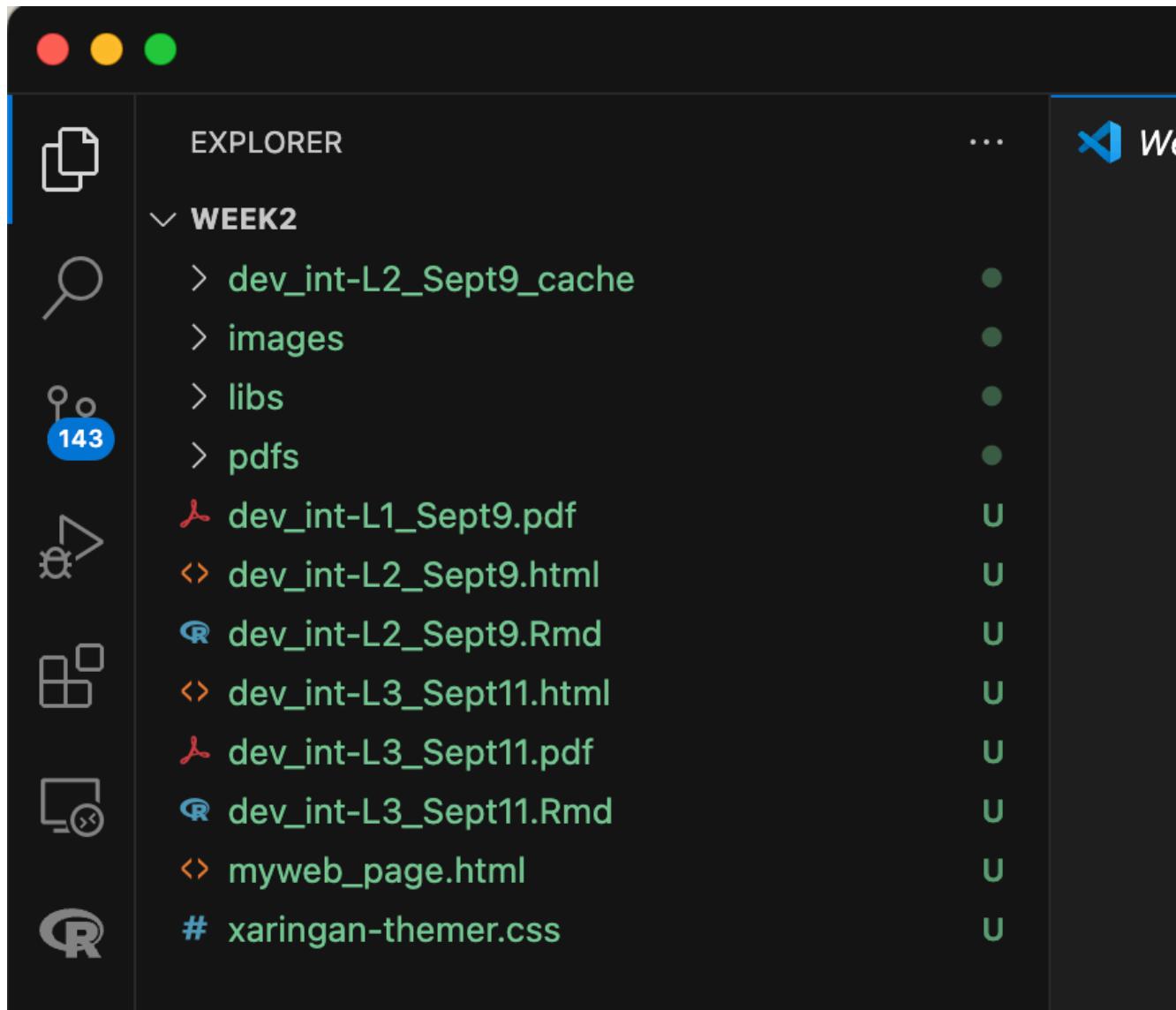
## Select text file



# Save as html



# Now you see your folder on the left hand side



# Open a file notes

1. Click on File -> New File in VS Code.
2. Save the file with an appropriate extension (e.g., my\_webpage.html).
3. Saving the file with .html triggers VS Code to assist with syntax highlighting and basic structure generation for HTML.

# **Task 1: Create Your First HTML Element**

- Open a new .html file.

# The 2 Elements of HTML

1. Block elements

2. Inline elements

# 1. Block Elements

- Represent larger sections of content, like paragraphs, lists, headings, sections, or articles.
- Take up the full width of their container and start on a **new line**.
- Browsers typically add space or margin around them by default.
- Layout and appearance can be customized with CSS.

# The Structure of an HTML Document

Every HTML document follows a basic structure:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>My First Heading</h1>
    <p>My first paragraph.</p>
  </body>
</html>
```

My First  
Heading

My first paragraph.

# Let's break it down

<!DOCTYPE html>

- Tells the browser to expect HTML5

`<html>` Tag

- The root element of an HTML document
- Contains the entire content of the page

`<head>` Tag

- Contains metadata and information about the document

E.g. `<title>`, `<meta>`, `<link>`

# <body> Tag

The <body> tag contains all the visible content:

- Headings
- Paragraphs
- Images
- Links

```
<body>
  <h1>Welcome to My Webpage</h1>
  <p>Learning HTML is fun!</p>
</body>
```

Welcome to  
My Webpage

Learning HTML is fun!

# Task 2: Create Your First HTML Element

- Write a `<p>` tag with some text content inside.
- Save and open the file in a browser to see how it displays.

# Headings in HTML

Use `<h1>` to `<h6>` for headings, with `<h1>` being the most important

```
<h1>Main Heading</h1>
<h2>Subheading</h2>
<h3>Sub-subheading</h3>
<h4>Sub-sub-subheading</h4>
<h5>Sub-sub-sub-subheading</h5>
<h6>Sub-sub-sub-sub-subheading</h6>
```

# Main Heading

## Subheading

### Sub-subheading

Sub-sub-subheading

Sub-sub-sub-subheading

Sub-sub-sub-sub-subheading

## Task 3: Build the Basic Structure

- Change the `<title>` to your own custom page title.
- Add another `<h2>` heading under the `<h1>`.

## Task 4: Create a Heading Hierarchy

- Add at least three headings of different levels (`<h1>`, `<h2>`, `<h3>`) to your page.
- Use the headings to structure different sections of your content.

## 2. Inline Elements

- Represent smaller chunks of content, such as bold text, code, images, or links.
- Stay on the **same line** with other inline elements (do not trigger line breaks).
- Must be nested within a **block element**.
- Occupy only as much space as their content requires.
- CSS can be used to modify their default styles.

```
<p>
  Observe how this <mark>highlighted text</mark>
  seamlessly blends into the paragraph.
  You can also include interactive elements like
  <input type="radio" name="preference" /> and
  <input type="checkbox" name="agree" />, which are inline and align
  effortlessly within the text flow.
</p>
```

Observe how this highlighted text seamlessly blends into the paragraph. You can also include interactive elements like ○ and □, which are inline and align effortlessly within the text flow.

# Paragraphs and Text

- Use the `<p>` tag to create paragraphs:

```
<p>This is a paragraph of text.</p>
```

You can add formatting with elements like:

- `<strong>` for bold text
- `<em>` for italicized text

```
<p>This is a <strong>paragraph</strong>
```

This is a **paragraph** of text.

# Other common inline elements

<p>

The term <abbr title="Cascading Style Sheets">CSS</abbr> is widely used in web design. You can use a <span style="color: red;">span</span> tag to apply custom styles to part of your text.

I like to use the <kbd>Ctrl</kbd> + <kbd>C</kbd> keys to copy text.

This is <mark>highlighted text</mark> in HTML.

Computer output is often displayed in a <code>code</code> tag.

</p>

The term CSS is widely used in web design. You can use a span tag to apply custom styles to part of your text. I like to use the Ctrl + C keys to copy text. This is highlighted text in HTML. Computer output is often displayed in a code tag.

# Task 5: Add Content to the Body

- Create an additional `<h2>` heading and a `<p>` paragraph under the `<body>`.
- Add two paragraphs with content about yourself or any topic.
- Make part of the text bold using `<strong>`, and another part italicized using `<em>`.

# Comments or notes to self

- HTML comments are written as `<!-- comment content -->`.
- They are used to document your HTML file or to "comment out" sections of code temporarily.

```
<!-- This section will be hidden -->
<!-- <h2>Under Construction</h2>
<p>This part of the site is coming soon!</p> -->

<p>Welcome to my webpage!</p>
```

## Output:

Welcome to my webpage!

## Why Use Comments?

- Comments can help you explain code functionality to yourself or others.
- They can be used to disable code while testing or debugging without deleting it.

## Note:

- Remember that comments are not visible to users when viewing the page in a browser.
- Avoid adding comments with sensitive information, as they are still visible in the HTML source code.
- Do not use --- inside your comments, only for opening and closing the comment.

# Task 6: Add Comments to Your HTML

- Add a comment at the top of your HTML file to describe the purpose of the page.
- Try commenting out a section of your HTML code and observe the changes in the browser.

# Lists in HTML

## Unordered Lists

- Use `<ul>` for bulleted lists

```
<ul>
  <li>First item</li>
  <li>Second item</li>
</ul>
```

## Ordered Lists

- Use `<ol>` for numbered lists

```
<ol>
  <li>Step one</li>
  <li>Step two</li>
</ol>
```

# Nested Lists

- A list that contains another list

```
<ul>
  <li>Cats are domesticated animals</li>
  <li>Cats have a wide range of breeds
    <ul>
      <li>Siamese</li>
      <li>Persian</li>
      <li>Maine Coon</li>
    </ul>
  </li>
</ul>
```

- Cats are domesticated animals
- Cats have a wide range of breeds
  - Siamese
  - Persian
  - Maine Coon

# Nested tags

- We can also nest tags inside tags
- **H/e be careful top open and close them correctly**

Incorrect version

```
<body>
  <p>
    Learning to code is <em>extremely <strong>valuable</em></strong>
    in today's world.
    Here's a quick to-do list:
  </p>
  <ol>
    <li>Finish HTML tutorial</li>
  </ol>
</body>
```

Output

Learning to code is *extremely valuable* in today's world. Here's a quick to-do list:

1. Finish HTML tutorial

## Correct version

```
<body>
  <p>
    Learning to code is <em>extremely <strong>valuable</strong></em>
    in today's world.
    Here's a quick to-do list:
  </p>
  <ol>
    <li>Finish HTML tutorial</li>
  </ol>
</body>
```

## Output

Learning to code is *extremely **valuable*** in today's world. Here's a quick to-do list:

1. Finish HTML tutorial

# Task 7: Create a List

- Add an unordered list (`<ul>`) of your favorite foods.
- Add an ordered list (`<ol>`) of steps to complete a task (like cooking a recipe).

# HTML Tag Attributes

- Tags can have additional information in the form of **attributes**.

```
<element attribute="value" attribute="value"> content </element>
```

# Images

- Either do it from your computer (I recommend an images folder)
- Attributes:
  - src: Specifies the image URL.
  - alt: Provides alternative text for accessibility and for cases when the image cannot be displayed.
  - title: Gives a tooltip when the mouse hovers over the image.

```

```

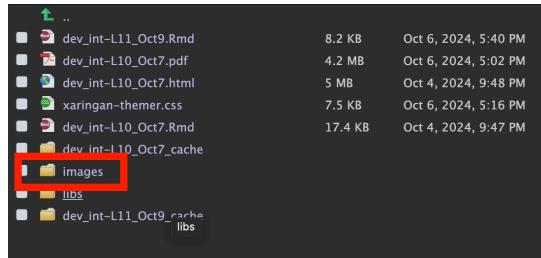


# Images continued

- Or from the internet
- Don't forget to cite your images! Check [HTML Charactr Entity References](#)

```

```



# Task 8: Add an Image

- Download an image to your project folder.
- Add the image to your HTML using the `<img>` tag.
- Or, use an image from an online URL.

# Links (or anchors)

- Use the `<a>` tag to create hyperlinks
- Attributes: href (Hypertext REference): Specifies the destination URL.
- Absolute URLs: Links to an external site (e.g., "<http://www.google.com>").
- Relative URLs: Links to another page within the same site (e.g., "my-other-page.html").

```
<a href="https://2024.philemerge.com/section_a/assignment/Demo/index.html">Visit Our Class Website</a>
```

[Visit Our Class Website](https://2024.philemerge.com/section_a/assignment/Demo/index.html)

# Task 9: Create Links

- Add a link to your favorite website using the `<a>` tag.
- Add an email link using `mailto:` as the href.

# Forms

- Forms allow user input and data submission
- Common form elements include:
  - **<input>**: text boxes, checkboxes, radio buttons
  - **<label>**: labels for form elements
  - **<button>**: buttons for form actions

small-code[

```
<form action="/submit" method="post">
  <label for="name">Name:</label>
  <input type="text" id="name" name="n
  <button type="submit">Submit</button>
</form>
```

Name:

]

# Task 10: Create a Simple Form

- Add a form to your HTML file with an `<input>` for the user's name.
- Add a submit button to the form.

# Self-Closing Tags (Void Elements)

- Some tags do not contain content and are self-closing, like break `<br>`.

```
<br>
<hr>
```

- *Note: You may see `<br />` or `<hr />` in some HTML versions. In HTML5, the `/` is optional and ignored.*

Now let's see your own website!

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Simple HTML Page</title>
  </head>
  <body>
    <h1>Hello World!</h1>
    <p>This is my first HTML page.</p>

    <h2>About Me</h2>
    <ul>
      <li>Web Developer</li>
      <li>HTML Enthusiast</li>
    </ul>

    <h2>Contact</h2>
    <p>Email me at <a href="mailto:example@example.com">example@example.com</a></p>
  </body>
</html>
```

# Creating Multiple Pages in HTML

- To create a multi-page website, create separate `.html` files (e.g., `index.html`, `about.html`, `contact.html`).
- Each page has its own HTML structure but shares a consistent navigation.

# **Task 11: Create simple multiple pages**

- Create one for index.html, about.html and contact.html
- Use all the information from previous slides

# Adding a Navigation Bar

- A navigation bar helps users move between different pages on your site.
- Elements:
  - **<nav>**: Container for navigation links.
  - **<ul>**: Unordered list for navigation items.
  - **<a href="...">**: Links to different pages.

```
<nav>
  <ul>
    <li><a href="index.html">Home</a></li>
    <li><a href="about.html">About</a></li>
    <li><a href="contact.html">Contact</a></li>
  </ul>
</nav>
```

- Home
- About
- Contact

# Task 12: Add navigation bar to all your pages

- Add the same `<nav>` element to top of each HTML file (`index.html`, `about.html`, `contact.html`).
- This will allow users to easily navigate between pages.

```
<html>
<head>
  <title>Home</title>
</head>
<body>
  <nav>
    <ul>
      <li><a href="index.html">Home</a></li>
      <li><a href="about.html">About</a></li>
      <li><a href="contact.html">Contact</a></li>
    </ul>
  </nav>

  <h1>Welcome to My Website</h1>
  <p>This is the home page.</p>
</body>
</html>
```

# HTML Elements Cheat Sheet

- HTML elements are the building blocks of HTML pages
- Elements are represented by tags
- Tags are enclosed in angle brackets
- Tags usually come in pairs: opening tag and closing tag

For full list: [MDN Web Docs](#)

# The <a> Element

- Represents a hyperlink, linking to another resource
- Can be used for navigation within a page or to other pages

```
<a href="https://example.com">Visit Example</a>
```

# The <div> Element

- A block-level container for other elements
- Used for layout and grouping content without semantic meaning

```
<div class="container">
  <p>This is inside a div.</p>
</div>
```

# The <p> Element

- Represents a paragraph of text
- A block-level element that contains regular text content

```
<p>This is a paragraph.</p>
```

# The <img> Element

- Embeds an image in the document
- Uses the src attribute to specify the image path and alt for alternative text

```

```

# The <ul> and <ol> Elements

- <ul> : Creates an unordered (bulleted) list
  - <ol> : Creates an ordered (numbered) list

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
</ul>

<ol>
  <li>First item</li>
  <li>Second item</li>
</ol>
```

# The <table> Element

- Creates a table for displaying tabular data
- Used for rows, for headers, and for data cells

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

# The <form> Element

- Creates a form for user input and submission
- Contains form elements like  ,  ,

```
<form action="/submit" method="post">
  <label for="name">Name:</label>
  <input type="text" id="name" name="name">
  <button type="submit">Submit</button>
</form>
```

# The <section> Element

- Represents a thematic grouping of content, such as chapters, headers, or articles
- Can be nested within other sections for better content organization

```
<section>
  <h2>Section Title</h2>
  <p>Section content goes here.</p>
</section>
```

# The <header> Element

- Represents introductory content for a section or page
- Typically contains headings, navigation links, or other relevant intro content

```
<header>
  <h1>Welcome to My Website</h1>
  <nav>
    <a href="/home">Home</a>
    <a href="/about">About</a>
  </nav>
</header>
```

# The <footer> Element

- Represents a footer for a section or the entire page
- Typically contains information like author, copyright, and contact details

```
<footer>
  <p>&copy; 2024 My Website</p>
</footer>
```

# The <span> Element

- A generic inline container for phrasing content
- Used to style or manipulate parts of text without affecting its structure

```
<p>This is <span style="color: red;">highlighted</span> text.</p>
```

# The <h1> to <h6> Elements

- Represent headings in descending order of importance (<h1> being the highest)
- Used to structure the hierarchy of content on the page

```
<h1>Main Heading</h1>
<h2>Subheading</h2>
<h3>Sub-subheading</h3>
```

# The `<em>` and `<strong>` Elements

- `<em>`: Emphasizes text, typically rendered in italics
- `<strong>`: Strongly emphasizes text, typically rendered in bold

```
<p>This is <em>important</em> and this is <strong>very important</strong>.</p>
```

# The

## | and “Elements”

**<blockquote>**: For block-level quotations, displayed as a separate block **<q>**: For inline quotations within text

```
<blockquote>
  <p>“This is a blockquote.”</p>
</blockquote>

<p>The raven said <q>“Nevermore.”</q></p>
```

# The <code> and <pre> Elements

- <code>: For inline code snippets, usually in a monospace font
- <pre>: For preformatted text with preserved whitespace and line breaks

```
<p>Use <code>&lt;div&gt;</code> to create a division.</p>
```

```
<pre>
function helloWorld() {
  console.log("Hello, world!");
}
</pre>
```

