Objectives:

- Introduction to Html
- Structure of Html document
- Head and Body Elements
- Working with Tables

Introduction to Html:

HTML stands for **HyperText Markup Language**. It was designed to present and share information over the internet, serving as the foundation for creating web pages. The primary role of HTML is to define the basic structure and content of a webpage. Using HTML, developers can display various elements such as buttons, paragraphs, images, links, lists, and more.
Elements in HTML are represented by **tags**, which define the type and purpose of the content. There are two main types of elements:

1. **Elements with content**: These require both an opening and a closing tag. Examples include paragraphs (`<p>...</p>`), headings (`<h1>...</h1>`), and lists (`<ul>...</ul>`).
2. **Self-closing elements**: These do not contain any content or data inside them and are represented by a single tag. Examples include images (`<img>`), line breaks (`<br>`), and input fields (`<input>`).
   Additionally, **attributes** can be added to HTML elements to provide further control over their behavior, appearance, or functionality.
   HTML files are saved with the **.html** extension. To view an HTML file, all we need is a web browser, which interprets the code and renders it into a visible and interactive webpage.

Structure of Html Document:

The first line of an HTML document is the **document type declaration**, which specifies the version of HTML we are using. For HTML5, we write `<!DOCTYPE html>`. This tells the browser that the document is an HTML5 file.
After the document type declaration, we put `<html>` tag ,this tag enclose the entire webpage. The content inside the `<html>` tag is divided into two main sections: the **head** and the **body**.

1. `<head>` **Section**:
   The `<head>` section contains metadata about the webpage, such as the title, description, keywords, author information, and links to external resources like stylesheets or scripts.

Elements inside the `<head>` tag aren't displayed on the webpage but provide important information for browsers and search engines.

2. `<body>` **Section**:
   The `<body>` section contains the actual content of the webpage that is displayed to users. we put inside it text, images, links, buttons, and other elements that user will see and interact with when they load the page.

Example:

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Webpage</title>
    <meta name="description" content="This is a sample webpage.">
  </head>
  <body>
    <h1>Welcome to My Webpage</h1>
    <p>This is a paragraph of text.</p>
  </body>
</html>
```

Head Elements:

Head:

The `<head>` section of an HTML document contains metadata and other information that is not displayed directly on the webpage but is essential for browsers, search engines, and other tools. Below are the most common elements found inside the `<head>` tag:

title:

Defines the title of the webpage, which appears in the browser's title bar or tab.

```
<title>My Title</title>
```

meta:

Provides **metadata** about the HTML document, such as character encoding, author, description, and viewport settings.

Character encoding:

Specifies the character encoding for the document. UTF-8 is the most commonly used encoding.

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

Viewport settings:

Ensures the webpage is responsive and scales properly on different devices.

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

Page Description:

Provides a brief description of the webpage, often used by search engines in search results.

```
<meta name="description" content="This is a sample webpage about HTML meta tags.">
```

Keywords:

Specifies a list of keywords relevant to the webpage (less commonly used by modern search engines)

```
<meta name="keywords" content="HTML, meta tags, web development, SEO">
```

Author Information:

Specifies the author of the webpage.

```
<meta name="author" content="Ali tigui">
```

Open Graph Metadata:

Used for social media sharing (e.g., Facebook, Twitter) to control how the page appears when shared.

```
<meta property="og:title" content="Sample Webpage">
<meta property="og:description" content="This is a sample webpage about meta
tags.">
<meta property="og:image" content="https://www.example.com/image.jpg">
```

Twitter Card Metadata:

Similar to Open Graph but specific to Twitter.

```
<meta name="twitter:card" content="summary_large_image">
<meta name="twitter:title" content="Sample Webpage">
<meta name="twitter:description" content="This is a sample webpage about meta
tags.">
<meta name="twitter:image" content="https://www.example.com/image.jpg">
```

Link:

Used to establish relationships between the current document and external resources. It is commonly used to link stylesheets, icons, fonts, and other assets that are essential for the webpage's functionality and appearance.

Linking Stylesheets (CSS):

The most common use of the `<link>` tag is to connect an external CSS file to style the webpage.

```
<link rel="stylesheet" href="styles.css">
```

Linking Favicons:

A favicon is a small icon displayed in the browser tab or bookmark bar.

```
<link rel="icon" href="favicon.ico" type="image/x-icon">
```

Linking Alternate Versions of the Page:

Provide alternate versions of the page, such as a PDF or a different language.

```
<link rel="alternate" href="document.pdf" type="application/pdf">
```

Linking Fonts:

Import custom fonts from external sources like Google Fonts.

```
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto">
```

Linking RSS Feeds:

RSS (Really Simple Syndication) feeds are a way for websites to distribute their content in a standardized format. By linking an RSS feed in our HTML document, we allow users and applications to easily subscribe to updates from ourwebsite.

```
<link rel="alternate" type="application/rss+xml" href="feed.xml">
```

Linking Preload Resources:

Preload critical resources (e.g., fonts, images) to improve page load performance.

```
<link rel="preload" href="font.woff2" as="font" type="font/woff2" crossorigin>
```

Link tag attributes:

- **`rel`** : Specifies the relationship between the current document and the linked resource. Common values include:
  - `stylesheet` for CSS files
  - `icon` for favicons
  - `preload` for preloading resources
  - `alternate` for alternate versions or RSS feeds
- **`href`** : Specifies the URL of the external resource.
- **`type`** : Defines the MIME type of the linked resource (e.g., `text/css` , `image/x-icon`).
- **`media`** : Specifies the media type or device the resource is intended for (e.g., `screen` , `print`).
- **`as`** : Used with `rel="preload"` to specify the type of resource being preloaded (e.g., `font` , `script` , `image`).
- **`crossorigin`** : Indicates whether the resource should be fetched with a CORS request (used for fonts or other cross-origin resources).

style:

Used to embed **internal CSS styles** directly within an HTML document. When placed inside the `<head>` section, it allows us to define styles that apply to the entire webpage.

```
    <style>
      /* CSS rules go here */
      body {
```

```
        font-family: Arial, sans-serif;
        background-color: #f0f0f0;
      }
      h1 {
        color: #333;
      }
      p {
        font-size: 16px;
        line-height: 1.6;
      }
    </style>
```

script:

Used to include **JavaScript code** that executes when the page is loaded.

Inline JavaScript:

We can write JavaScript code directly inside the `<script>` tag

```
    <script>
      // JavaScript code goes here
      alert("Hello, World!");
    </script>
```

External JavaScript:

We can link to an external JavaScript file using the `src` attribute:

```
<script src="script.js"></script>
```

base:

It specifies a **base URL** or **default target** for all relative URLs in the document. This can simplify the management of links, scripts, and other resources by providing a common starting point for relative paths.
The `<base>` tag has two primary attributes:

1. **`href`** : Specifies the base URL for all relative URLs in the document.
2. **`target`** : Defines the default target for all links and forms in the document.

```
<base href="https://www.example.com/" target="_blank">
```

noscrip:

used to provide **fallback content** for users who have disabled JavaScript in their browsers or are using browsers that do not support JavaScript. This ensures that our website remains functional and accessible even when JavaScript is unavailable.

Providing Alternate Content:

If a feature relies on JavaScript, we can provide an alternate version using the `<noscript>` tag:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Noscript Example</title>
    <script>
      document.write("<p>This content is loaded with JavaScript.</p>");
    </script>
  </head>
  <body>
    <noscript>
      <p>This content is displayed because JavaScript is disabled.</p>
    </noscript>
  </body>
</html>
```

Redirecting Users:

We can use the `<noscript>` tag to redirect users to a non-JavaScript version of our website:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Noscript Example</title>
    <script>
      // JavaScript code
    </script>
    <noscript>
      <meta http-equiv="refresh" content="0; url=https://www.example.com/no-js-
```

```
version">
    </noscript>
  </head>
  <body>
    <p>Welcome to the JavaScript-enabled version of the website.</p>
  </body>
</html>
```

Body Elements:

The `<body>` tag in HTML is used to define the main content of a web page. It contains all the visible elements that users see and interact with, such as text, images, links, forms, and other media.

Paragraphs:

The `<p>` tag in HTML is used to **define and insert paragraphs** into a web page. It is a block-level element that separates text into distinct sections, making content more readable and organized.

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

Heading:

HTML provides **six levels of headings**, ranging from `<h1>` to `<h6>`. These tags are used to define headings and subheadings in a webpage, helping to structure content hierarchically.

1. **`<h1>`**: The highest level of heading, typically used for the main title of the page. There should usually be only one `<h1>` per page for SEO best practices.
2. **`<h2>`**: Used for major section headings.
3. **`<h3>`**: Used for subheadings under `<h2>` sections.
4. **`<h4>`**: Used for further subsections.
5. **`<h5>`**: Used for minor headings.
6. **`<h6>`**: The lowest level of heading, used for small or less important headings.

```
<h1>Main Title</h1>
<h2>Section Heading</h2>
<h3>Subheading</h3>
<h4>Minor Subheading</h4>
```

```
<h5>Small Heading</h5>
<h6>Least Important Heading</h6>
```

Breaking Line in HTML

To create a **line break** in HTML, use the `<br>` tag.This tag is a self-closing element that forces text or content to move to the next line. It is useful when we want to add space between lines without starting a new paragraph.

```
<p>This is the first line.<br>This is the second line.</p>
```

Horizontal Line in HTML

To create a **horizontal line** use the `<hr>` tag. This tag is used to separate content sections visually with a horizontal line.Like `<br>`, it is a self-closing element.

```
<p>This is the first section of content.</p>
<hr>
<p>This is the second section of content.</p>
```

Styling Text in HTML:

HTML provides several tags to style and format text for emphasis, readability, and specific use cases. Below are the commonly used tags:

`<em>` Tag:

Used to emphasize text. Browsers typically render it in *italics*, it Indicates that the text has stress emphasis.

```
<p>This is <em>important</em> text.</p>
```

Output: This is *important* text.

`<i>` Tag:

Used to style text in *italics* without implying emphasis, it used for technical terms, foreign phrases, or thoughts.

```
<p>The term <i>et cetera</i> is often abbreviated as <i>etc.</i></p>
```

Output: The term *et cetera* is often abbreviated as *etc.*

`<u>` Tag

Used to underline text,it highlighting misspelled words or proper nouns.

```
<p>This is <u>underlined</u> text.</p>
```

Output: This is underlined text.

`<b>` Tag

Used to make text **bold** without implying importance.It Commonly used for Keywords, product names, or headings.

```
<p>This is <b>bold</b> text.</p>
```

Output: This is **bold** text.

`<strong>` Tag:

Used to indicate text with **strong importance**. Browsers typically render it in bold,it Indicates that the text is of high importance.

```
<p>This is <strong>very important</strong> text.</p>
```

Output: This is **very important** text.

`<sup>` Tag

Used for **superscript** text, which appears slightly above the normal line of text, it used for Mathematical exponents, footnotes, or ordinal indicators.

```
<p>E = mc<sup>2</sup></p>
```

Output: E = mc².

`<sub>` Tag

Used for **subscript** text, which appears slightly below the normal line of text., it used for Chemical formulas, mathematical indices, or footnotes.

```
<p>H<sub>2</sub>O is the chemical formula for water.</p>
```

Output: H_2O is the chemical formula for water.

Special Characters:

Special characters (e.g., symbols, accents, or reserved characters) can be added to HTML using **character entities**. These are codes that represent characters not available on a standard keyboard or have special meanings in HTML.
Example :

```
<p>Copyright &copy; 2023 My Company. All rights reserved.</p>
<p>Price: &euro;50.00</p>
<p>5 &lt; 10 is true.</p>
```

Output:
Copyright © 2023 My Company. All rights reserved.
Price: €50.00
5 < 10 is true.

Lists in HTML:

HTML provides three types of lists to organize and structure content: **unordered lists**, **ordered lists**, and **definition lists**.

Unordered Lists (`<ul>`):

Used to create a list of items where the order does not matter.Each item in the list is wrapped in a `<li>` (list item) tag.

```
<ul>
  <li>Apple</li>
  <li>Banana</li>
  <li>Orange</li>
</ul>
```

Output:

- Apple
- Banana
- Orange

Ordered Lists (`<ol>`)

Used to create a list of items where the order matters (e.g., steps in a recipe). Each item is wrapped in a `<li>` tag.

```
<ol>
  <li>Preheat the oven.</li>
  <li>Mix the ingredients.</li>
  <li>Bake for 30 minutes.</li>
</ol>
```

Output:

1. Preheat the oven.
2. Mix the ingredients.
3. Bake for 30 minutes.

Definition Lists (`<dl>`)

Used to create a list of terms and their definitions (like a dictionary).
- **Syntax**:
- `<dl>` : Wraps the entire list.
- `<dt>` : Defines the term.
- `<dd>` : Defines the description or definition of the term.

```
<dl>
  <dt>HTML</dt>
  <dd>HyperText Markup Language, used for creating web pages.</dd>
  <dt>CSS</dt>
  <dd>Cascading Style Sheets, used for styling web pages.</dd>
</dl>
```

Output:
HTML
HyperText Markup Language, used for creating web pages.
CSS
Cascading Style Sheets, used for styling web pages.

Nested Lists

We can nest lists inside other lists to create sub-items or hierarchical structures.

```
<ul>
  <li>Fruits
    <ul>
      <li>Apple</li>
      <li>Banana</li>
    </ul>
  </li>
  <li>Vegetables
    <ul>
      <li>Carrot</li>
      <li>Broccoli</li>
    </ul>
  </li>
</ul>
```

Output:

- Fruits
  - Apple
  - Banana
- Vegetables
  - Carrot
  - Broccoli

Class and ID Attributes in HTML

The `class` and `id` attributes are used to identify and style HTML elements. They are essential for applying CSS styles and JavaScript functionality to specific elements or groups of elements.

`class` Attribute

Used to define a **group of elements** that share the same styling or behavior.Multiple elements can have the same class name.

```
<p class="highlight">This is a highlighted paragraph.</p>
<p class="highlight">This is another highlighted paragraph.</p>
```

`id` Attribute:

Used to uniquely identify a **single element** on a page. Each `id` must be unique within a page.

```
<p id="intro">This is the introduction paragraph.</p>
<p>This is a regular paragraph.</p>
```

Links in HTML:

Links are one of the most fundamental features of HTML, allowing users to navigate between web pages, sections of a page, or external resources. The `<a>` (anchor) tag is used to create hyperlinks.

Basic Link Syntax:

- **Tag**: `<a>`
- **Attribute**: `href` (hypertext reference) specifies the destination of the link.
- **Content**: The text or element inside the `<a>` tag becomes clickable.

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

Output:
Visit Example.com

Types of Links

Absolute Links
Point to a full URL (including the protocol, e.g., `https://`). Used for linking to external websites.

```
<a href="https://www.google.com">Go to Google</a>
```

Relative Links
Point to a file or resource within the same website.

```
<a href="about.html">About Us</a>
```

This will take us to the about .html page in our website.
Internal Links (Anchor Links)
Point to a specific section within the same page using the `id` attribute.

```

<a href="#section2">Jump to Section 2</a>
```

```
<h2 id="section2">Section 2</h2>
<p>This is the content of Section 2.</p>
```

It is used to navigate through our webpage.

Email Links

Open the user's default email client with a pre-filled email address.

```
<a href="mailto:example@example.com">Send Email</a>
```

Phone Links

Open the user's default phone app with a pre-filled phone number.

```
<a href="tel:+1234567890">Call Us</a>
```

Link Attributes

`target` Attribute

Specifies where to open the linked resource.

- `_blank` : Opens the link in a new tab or window.
- `_self` : Opens the link in the same tab (default behavior).

```
<a href="https://www.example.com" target="_blank">Open in New Tab</a>
```

`title` Attribute

Provides additional information about the link (displayed as a tooltip).

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

`<div>` and `<span>` tags:

`<div>` Tag

The `<div>` tag is a **special tag** used as a **container** to divide and group elements together, making them fall under the same section, it can be helpfull when we want style elements

```
<div>
  <h2>This is a heading inside a div</h2>
```

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

`<span>` Tag

Unlike the `<div>` tag, the `<span>` tag is another container element used to group or wrap elements together. `<span>` primarily used to style or manipulate small portions of text or inline content.

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

Inline vs Block Elements in HTML

Block Elements

Block elements in HTML are the elements we use to structure our webpage and divide it into block sections. Each time we create a block element, it adds a new line before and after it. Another feature of block elements is that they take up the full width of their container. Some examples of block elements are `<div>`, `<h1>` to `<h6>`, `<p>`, `<ul>`, and `<li>`.

Inline Elements

Inline elements in HTML are the elements we use to style or modify small portions of content within a block. Unlike block elements, inline elements do not start on a new line; instead, they flow within the text or content of a block element. They only take the width to of their content
Some examples of inline elements are `<span>`, `<a>`, `<strong>`, `<em>`, `<img>`, and `<br>`.

Tables:

Creating table

In HTML, we create tables using the `<table>` tag. Inside the `<table>` tag, we add table rows using the `<tr>` (table row) tag. Inside each row, we add table data using the `<td>` (table data) tag or table headers using the `<th>` (table header) tag.
When we create a table, borders are not visible. To add borders, we use the `border` attribute in the `<table>` tag.

```
<table border>
<caption>Example Table</caption>
    <tr>
        <th>Header 1</th>
```

```
        <th>Header 2</th>
        <th>Header 3</th>
    </tr>
    <tr>
        <td>Row 1, Cell 1</td>
        <td>Row 1, Cell 2</td>
        <td>Row 1, Cell 3</td>
    </tr>
    <tr>
        <td>Row 2, Cell 1</td>
        <td>Row 2, Cell 2</td>
        <td>Row 2, Cell 3</td>
    </tr>
</table>
```

Output:

Header 1	Header 2	Header 3
Row 1, Cell 1	Row 1, Cell 2	Row 1, Cell 3
Row 2, Cell 1	Row 2, Cell 2	Row 2, Cell 3

Employee Details

Name	Department	Salary
John Doe	Marketing	$60,000
Jane Smith	Engineering	$80,000

Employee Details

Name	Department	Salary
John Doe	Marketing	$60,000
Jane Smith	Engineering	$80,000

Name		Age
John	Doe	25
Jane	Smith	30

Name		Age
John	Doe	25
Jane	Smith	30

ID	Name	Age
	John Doe	25
2	Jane Smith	30

ID	Name	Age
	John Doe	25
2	Jane Smith	30

ID	Name		Age
	First Name	Last Name	
1	John	Doe	25
2	Jane	Smith	30

ID	Name		Age
	First Name	Last Name	
1	John	Doe	25
2	Jane	Smith	30

Objective

Create a **personal portfolio webpage** using HTML to showcase your skills, projects, and contact information.

Requirements:

1. Basic Structure (`<head>` & `<body>`)
   - Use `<!DOCTYPE html>` and proper HTML5 structure.
   - Add a meaningful `<title>` (e.g., "John Doe - Portfolio").
   - Include meta tags for **description**, **viewport**, and **UTF-8 encoding**.
2. Navigation Bar (Menu)

- Create a navigation bar (`<nav>`) with links to:
  - **Home** (links to the top of the page).
  - **About Me** (links to a section).
  - **Projects** (links to a section).
  - **Contact** (links to a section).

3. Sections (Using `<div>` or `<section>`)
   - **Hero Section (Home)**:
     - A heading (`<h1>`) with your name.
     - A short introduction (`<p>`).
     - An image (use `<img>` with `alt` text).
   - **About Me Section**:
     - A heading (`<h2>`).
     - A paragraph (`<p>`) describing yourself.
     - An **unordered list** (`<ul>`) of skills.
   - **Projects Section**:
     - A heading (`<h2>`).
     - A **table** (`<table>`) listing:
       - **Project Name** (header `<th>`).
       - **Description** (header `<th>`).
       - **Year** (header `<th>`).
     - At least 3 projects (rows).
   - **Contact Section**:
     - A heading (`<h2>`).
     - A **link** (`<a>`) to your email (`mailto:`).
     - A **phone link** (`tel:`).
     - A **social media link** (e.g., LinkedIn, GitHub).
4. Footer
   - A `<footer>` with:
     - Copyright symbol (`&copy;`) and current year.
     - A **horizontal line** (`<hr>`) above it.
5. Bonus (Optional)
   - Add a **favicon** (use `<link rel="icon">`).
   - Use `<strong>` , `<em>` , or `<span>` for text styling.

Solution

You can find our solution here:
https://alitigui.github.io/Front_end_solutions/Lecture2/solution1/portfolio.html

Task 2:

Objective:

Design a **weekly school timetable** using advanced HTML table features (`colspan`, `rowspan`, `<caption>`).

Requirements:

1. Basic Table Structure
   - Use `<table border="1">` for visible borders
   - Add a `<caption>` (e.g., "Class 10A Weekly Timetable").
2. Table Headers (`<thead>`)
   - First row: Days of the week (`<th>` for Monday to Friday).
   - Second row: Time slots (`<th>` for "8:00 AM", "9:00 AM", etc.).
3. Table Body (`<tbody>`)
   - Use `rowspan` for subjects that span multiple hours (e.g., Math from 8 AM to 10 AM).
   - Use `colspan` for merged cells (e.g., "Lunch Break" across all days).
   - Include at least **5 subjects** (Math, Science, History, etc.).
4. Special Rows
   - **Lunch Break**: A row with `colspan="5"`.
   - **Free Period**: A cell with `rowspan="2"`.
5. Footer (`<tfoot>`)
   - Add a footer with a note (e.g., "Timetable subject to change").
6. Bonus (Optional)
   - Add a **"Notes" column** with `<ul>` for reminders.
   - Use `<abbr>` for abbreviations (e.g., `<abbr title="Physical Education">PE</abbr>`).

Solution

You can find our solution here:
https://alitigui.github.io/Front_end_solutions/Lecture2/solution2/timetable.html