

Session 1: Advanced HTML Topics

Front-End Web Development Bootcamp

Embedding Multimedia

What is Multimedia in HTML?

- HTML allows embedding **audio** and **video** files, which are essential for creating interactive and rich media websites.

- Embedding Audio (<audio>):

- The <audio> tag is used to embed sound files like MP3, Ogg, or WAV.

```
<audio controls>
```

```
<source src="audio-file.mp3" type="audio/mp3">
```

```
Your browser does not support the audio element.
```

```
</audio>
```

- Attributes:
 - controls: Adds play, pause, and volume control buttons.
 - autoplay: Starts playing automatically.
 - loop: Loops the audio once it finishes.

Embedding Multimedia

Embedding Video (<video>):

- The <video> tag is used to embed video files like MP4, WebM, or Ogg.

```
<video controls>
```

```
  <source src="video-file.mp4" type="video/mp4">
```

```
  Your browser does not support the video element.
```

```
</video>
```

- Attributes:
 - controls: Adds play, pause, and volume control buttons.
 - autoplay: Automatically starts playing the video.
 - muted: Starts the video with no sound.

Activity 1: Embed an Audio and video File

Objective:

Practice using the `<audio>` and `<video>` tag to embed a file.

- Instructions:
- use an MP3 audio file and a MP4 video file.
- Embed the audio using the `<audio>` tag.
- Embed the video using the `<video>` tag.
- Add the controls attribute to allow users to play, pause, and control the volume. Optionally, add the autoplay attribute and set it to true to play the audio automatically.

HTML Tables

- Tables (<table>):
- Used to display tabular data (rows and columns).
- Table Elements:
 - <th>: Table header (bold and centered).
 - <td>: Table data (standard table cell).
 - <tr>: Table row.
 - <caption>: Table title or description.

```
<table>
  <tr>
    <th>Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>John</td>
    <td>25</td>
  </tr>
  <tr>
    <td>Jane</td>
    <td>30</td>
  </tr>
</table>
```

HTML Forms

Form Elements:

- Forms are used to collect user input.
- Forms contain different input fields like text, password, checkboxes, etc.

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

Form Method - GET vs POST

What are the **GET** and **POST** Methods?

Forms can use either the **GET** or **POST** method to send data to a server. The choice of method determines how data is sent and how secure the submission is.

Form Method - GET vs POST

GET Method:

- **Purpose:** The GET method appends form data to the URL, making it visible in the browser's address bar.
- **Usage:** It is typically used for **search forms** or actions where data does not need to be kept secret.
- **Characteristics:**
 - Data is visible in the URL.
 - Limited amount of data can be sent (about 2000 characters).
 - Can be bookmarked or cached.

```
<form action="/search" method="get">  
  <label for="query">Search:</label>  
  <input type="text" id="query" name="query">  
  <input type="submit" value="Search">  
</form>
```


Form Method - GET vs POST

POST Method:

- **Purpose:** The POST method sends data in the body of the request, making it **invisible** in the URL.
- **Usage:** Typically used for **sending sensitive data**, such as login credentials, personal information, or form submissions.
- **Characteristics:**
 - Data is hidden from the URL.
 - More data can be sent (no size limit).
 - **More secure** because data is not visible in the URL.

```
<form action="/submit" method="post">  
  <label for="username">Username:</label>  
  <input type="text" id="username" name="username">  
  <input type="password" id="password" name="password">  
  <input type="submit" value="Submit">  
</form>
```

Basic <input> Types

```
<form action="/submit" method="post">
  <label for="username">Username:</label>
  <input type="text" id="username" name="username">
  <label for="password">Password:</label>
  <input type="password" id="password" name="password">
  <label for="reset">Reset:</label>
  <input type="checkbox" id="reset" name="reset" value="reset">
  <input type="submit" value="Submit">
</form>
```

Text Input (<input type="text">)

- Used to allow users to enter single-line text.

Password Input (<input type="password">)

- Used for entering sensitive information, such as passwords. The text entered is masked.

Submit Button (<input type="submit">)

- Used to submit the form data to the server.

Checkbox Input (<input type="checkbox">)

- Used to allow users to select one or more options from a set of choices.

Textarea (<textarea>)

- Used for multi-line text input (e.g., comments, descriptions).

HTML Formatting Elements

HTML Formatting Elements:

These are elements that change the presentation of content on the page without altering its structure.

****: Bold text.

<i>: Italic text.

<u>: Underlined text.

<mark>: Highlighted text.

<small>: Smaller text.

****Bold Text****

<i>Italic Text**</i>**

<u>Underlined Text**</u>**

<mark>Highlighted Text**</mark>**

<small>Smaller Text**</small>**

HTML Styles

What Are HTML Styles?

- Inline Styles allow you to apply CSS directly within an HTML element.
- Styles are added using the style attribute inside HTML tags.
- While inline styles are convenient for quick changes, it's best practice to use external or internal stylesheets for better maintainability.

```
<p style="color: ■blue; font-size: 20px;">This is a styled paragraph.</p>
```

HTML Text Formatting

HTML Text Formatting: Formatting elements affect how text appears on the page, making it bold, italic, or underlined.

****: Strong emphasis, typically bolded.

****: Emphasized text, typically italicized.

****: Deleted text (strikethrough).

<ins>: Inserted text (underlined).

```
<strong>This is strong text.</strong>
```

```
<em>This is emphasized text.</em>
```

```
<del>This is deleted text.</del>
```

```
<ins>This is inserted text.</ins>
```

Block vs. Inline Elements

Block Elements:

Block elements take up the full width available and start on a new line.

Common block elements: `<div>`, `<p>`, `<h1>`, ``, ``.

```
<div>This is a block-level element.</div>
```

```
<p>This is a paragraph, which is a block-level element.</p>
```

Block vs. Inline Elements

Inline Elements:

Inline elements only take up as much width as their content requires and do not start on a new line.

Common inline elements: ``, `<a>`, ``, ``.

Key Difference:

- Block elements take up full width and push subsequent content to a new line.
- Inline elements fit inside block elements without disrupting the layout.

```
<span>This is an inline element.</span>
```

```
<a href="https://www.example.com">Click here</a>
```

Activity 2: Use Inline Styles and Text Formatting

Objective: Apply inline styles and text formatting to a paragraph.

- **Instructions:**

- Create a paragraph of text and apply different inline styles (e.g., change color, font size).
- Apply HTML text formatting to certain parts of the paragraph (e.g., make one word bold, another italic, etc.).

Adding a Favicon

What is a Favicon?

- A **favicon** is a small icon that appears in the browser tab when visiting a website, often used to brand the site.
- To add a favicon, link it in the `<head>` section using the `<link>` tag:

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

Font Awesome

Using Font Awesome Icons:

Font Awesome is a popular library of vector icons that can be used on webpages.

To use Font Awesome:

Include the Font Awesome CDN link in the <head> of your HTML document.

Use the <i> tag with the corresponding class to display an icon.

```
<head>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/
font-awesome/6.0.0-beta3/css/all.min.css">
</head>
<body>
  <i class="fas fa-heart"></i>
</body>
```

Activity 3: Add a Favicon and Font Awesome Icon

Practice adding a favicon and using Font Awesome icons.

Instructions:

- Add a favicon to your webpage by linking to an icon (use any .ico image or Font Awesome icon).
- Add a Font Awesome icon (e.g., a heart or star) somewhere in your HTML content.