Handling Media

Topic will covered

- iFrame
- Playing and controlling audio
- Playing and controlling video
- Embedding a video

<iframe>: The Inline Frame element

An <iframe> (short for inline frame) is an HTML element used to embed another HTML document within the current document. It allows you to display content from another source, such as a webpage within your own webpage

Ex. <iframe src = "/html/menu.htm" width = "555" height = "200"></iframe>

Attributes:

1. src Attribute:

Specifies the URL of the content to be embedded   within the iframe.

html

```
<iframe src="https://www.example.com"></iframe>
```

2. width and height Attributes:

Specifies the width and height of the iframe.

```
Html: <iframe src="https://www.example.com" width="600" height="400"></iframe>
```

3. frameborder Attribute:

Specifies whether or not to display a border around the iframe.

```
html: <iframe src="https://www.example.com" width="600" height="400"frameborder="0"></iframe>
```

•

4. scrolling Attribute:

Specifies whether or not to display scrollbars within the iframe.

Html: <iframe src="https://www.example.com" width="600" height="400" scrolling="auto"></iframe>

5. sandbox Attribute:

Specifies a set of extra restrictions on the content within the iframe.

Html: <iframe src="https://www.example.com" width="600" height="400" sandbox="allow-scripts"></iframe>

6. allow Attribute:

Specifies a set of permissions for the content within the iframe.

Html: <iframe src="https://www.example.com" width="600" height="400" allow="fullscreen"></iframe>

Embedding an Audio File

```
<audio controls>
  <source src="audio_file.mp3" type="audio/mpeg">
</audio>
```

In this example:

src: Specifies the URL of the audio file.

type: Specifies the MIME type of the audio file. In this case, audio/mpeg for MP3 files.

controls: Adds audio controls (play, pause, volume) to the audio player.

Replace "audio_file.mp3" with the URL or path to your audio file.

What is MINE ?

"MIME" stands for Multipurpose Internet Mail Extensions. It's an internet standard that extends the format of email to support text in character sets other than ASCII, as well as attachments of audio, video, images, and application programs. In the context of web development, MIME types are also used to specify the type of content being served by a web server.

For example, when a browser requests a resource from a server, the server responds with the appropriate MIME type in the HTTP header. This MIME type tells the browser how to handle the received content. For example, for HTML files, the MIME type is typically text/html, for JPEG images it's image/jpeg, for JavaScript files it's application/javascript, and so on.

Changing the Audio File:

If you want to change the audio file dynamically, you can use JavaScript to modify the src attribute of the <audio> element. Here's a simple example using JavaScript

Ex.

```
<audio id="myAudio" controls>
  <source src="audio_file1.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
</audio>

<button onclick="changeAudio()">Change Audio</button>

<script>
function changeAudio() {
  var audio = document.getElementById("myAudio");
  audio.src = "audio_file2.mp3"; // Change the source to the new audio file
  audio.play(); // Start playing the new audio
}
</script>
```

- We have an audio element with an id attribute set to "myAudio".

- We have a button with an onclick attribute calling the changeAudio() function.

- Inside the changeAudio() function, we get the audio element by its ID, then set its src attribute to the new audio file URL ("audio_file2.mp3" in this case) and call play() to start playing the new audio.

- Replace "audio_file1.mp3" and "audio_file2.mp3" with the URLs or paths to your audio files.

Embedding a Video:

```
<video width="640" height="360" controls>
  <source src="video_file.mp4" type="video/mp4">
 </video>
```

<video>: This is the video element.
width and height: These attributes specify the dimensions of the video player.
controls: This attribute adds video controls such as play, pause, and volume.
<source>: This is a child element of <video> used to specify multiple sources for the video. The browser will choose the first source it supports. You can have multiple <source> elements with different formats to ensure compatibility across different browsers.
src: This attribute specifies the URL of the video file.
Type: This attribute specifies the MIME type of the video file. In this case, video/mp4 for MP4 files.

Changing the Video File:

If you want to change the video file dynamically, you can use JavaScript to modify the src attribute of the <video> element. Here's a simple example using JavaScript

Ex.

```
function changeVideo() {

  var video = document.getElementById("myVideo");

  video.src = "video_file2.mp4"; // Change the source to the new video file

  video.play(); // Start playing the new video
}
```

- We have a video element with an id attribute set to "myVideo".

- We have a button with an onclick attribute calling the changeVideo() function.

- Inside the changeVideo() function, we get the video element by its ID, then set its src attribute to the new video file URL ("video_file2.mp4" in this case) and call play() to start playing the new video.

- Replace "video_file1.mp4" and "video_file2.mp4" with the URLs or paths to your video files. When the button is clicked, it will change the source of the video element to the new video file and start playing it.

Your browser does not support the video tag?

This message "Your browser does not support the video tag." appears when the browser does not support the <video> element or the video format specified in the <source> element.

- Incorrect File Path or URL
- Unsupported Video Format
- Browser Compatibility

Handling media :

- **Embedding Audio and Video:**
- **Controlling Playback:**
- **Adding Subtitles and Captions**
- **Providing Fallbacks:**