**Unit 5 HTML Working with Graphics and Images**

When adding images to a web page, we use the <img> element. This element allows us to embed a simple image directly within our HTML content. Here are the basics:

1. **The**<img>**Element**: The <img> element is a **void element**, meaning it cannot have child content or an end tag. It requires two essential attributes:

* src: This attribute contains a URL pointing to the image you want to embed. It can be either a relative URL (if the image is in the same directory as your HTML page) or an absolute URL.
* alt: The alt attribute provides alternative text for the image. It’s essential for accessibility and SEO. If the image fails to load, the alt text will be displayed instead.

1. **Relative URLs**: When embedding an image from the same directory as your HTML page, use a relative URL. For example:

**HTML**

<img src="dinosaur.jpg" alt="Dinosaur" />

If the image is in a subdirectory (e.g., an “images” folder within the same directory), adjust the path accordingly:

**HTML**

<img src="images/dinosaur.jpg" alt="Dinosaur" />

1. **Descriptive Filenames**: Search engines consider image filenames for SEO. Therefore, choose descriptive filenames (e.g., “dinosaur.jpg” rather than “img835.png”).
2. **Absolute URLs (Not Recommended)**: While you can use absolute URLs to link to images hosted elsewhere, it’s better to host your images on the same server as your HTML. Relative URLs are more efficient and easier to maintain.

**Unit 6: Working with Media**

**Audio Element**

The HTML audio Element is used to embed sound content in documents. It may contain one or more audio sources represented using the src attribute or the source element.  The <audio> element allows you to load, play, pause, and set audio properties using the HTML DOM

* **Basic Usage**: To include an audio player on your webpage, use the <audio> tag along with the <source> tag. Here’s an example:

-The controls attribute adds audio controls like play, pause, and volume.

-The source element specifies alternative audio files (in different formats) that the browser can choose from. The browser will use the first recognized format.

-The text between the <audio> and </audio> tags will be displayed only in browsers that do not support the audio element.

* The <audio> element is supported by various browsers. Here’s a list of browser compatibility:

-Internet Explorer

-Chrome

-Firefox

-Safari

-Opera

**Video Element**

The **HTML video element** is used to embed a media player that supports video playback directly into a web page. It’s a powerful tool for displaying videos and can also be used for audio content. Let’s explore its features:

1.Basic Usage:

* To include a video on your webpage, use the <video> element.
* The controls attribute adds video controls (play, pause, volume, etc.).
* The source element specifies alternative video files in different formats (MP4 and Ogg). The browser will use the first recognized format.
* The text between the <video> and </video> tags is displayed in browsers that don’t support the video element.

2.Autoplay

* To start a video automatically, use the autoplay attribute.
* However, note that Chromium browsers often don’t allow autoplay. You can use muted after autoplay to allow muted autoplay.

3. Supported video formats:

* MP4

**Working With Captions and Subtitles**

Certainly! Adding captions and subtitles to HTML videos is essential for enhancing accessibility and making content more inclusive. Let’s explore how you can achieve this:

1.Captions vs. Subtitles:

* **Captions** are intended for viewers who are deaf or hard of hearing. They provide a textual representation of spoken dialogue, sound effects, and other relevant audio information.
* **Subtitles**, on the other hand, are primarily for viewers who can hear but may not understand the language spoken in the video. They display translated dialogue or text.

2.Using the track Element:

* HTML allows us to specify subtitles for a video using the <track> element.
* The track element has various attributes:

-src: Specifies the file path to the subtitle file (usually in WebVTT format).

-label: Describes the language or purpose of the subtitle (e.g., “English”).

-kind: Indicates the type of track (use “subtitles” for subtitles).

-srclang: Specifies the language code (e.g., “en” for English).

-default: Marks the default subtitle track.

**Embedding Media via iframes**

HTML iframes are a powerful way to embed external content within a web page.

What is an iframe?

* An iframe (short for “inline frame”) is an HTML element used to display another web page or document within the current web page.
* It allows you to embed content from external sources seamlessly.
* The iframe element has attributes like height and width that can be adjusted.
* The src attribute is used to specify the source of the video file.

**Unit 7 HTML Content Identification**

**HTML Language Support**

 When working with HTML, it’s essential to declare the language of your web page. This helps search engines and browsers understand the content better. Here are some key points regarding language support in HTML:

* Always include the lang attribute inside the <html> tag to declare the language of your web page.
* The lang attribute indicates the primary language of the content within a specific element (e.g., a paragraph or a heading).
* It’s essential for accessibility.
* while HTML provides support for the lang attribute, it doesn’t automatically translate content. It merely helps browsers and search engines understand the language context.

**HTML Generic Elements, Div and Span**

Div Element**:**

* The div element is a block-level container. It is used for structuring and organizinglarger sections of a web page.
* Typically, we use <div> to wrap entire sections, such as headers, footers, sidebars, or content blocks.

Span Element:

* The span element is an inline-level container. It is used for styling smaller portions of text, images, or other inline content.
* Unlike div, which creates a new block, <span> doesn’t introduce a line break. It wraps content within the same line.
* You’ll often use span to apply specific styles (such as color, font, or background) to individual words, letters, or small phrases.

**Unit 8 HTML Integration**

Certainly! **HTML integration** is a fascinating topic that allows you to seamlessly incorporate various elements into your web pages.

**HTML Pages**

* Structure of the whole HTML file:

Once the HTML file is built, there are a few crucial parts that every web page needs.

Firstly, the file should begin with a doctype statement, which indicates the era of this HTML file. In the past, there were different doctype declarations for older HTML versions. By including this one, we are saying, "Hey, this is a modern web page, so follow modern best practices and treat it accordingly."

Next, we enclose everything else on the page within an HTML element, which means an element named HTML. It tells us that all the content within it is HTML. Place the opening HTML tag at the top and the closing HTML tag at the bottom.

At the beginning, specify a few things about the web page. Declare the language being used and the content flow direction. This code is for a website using US-based English that reads from left to right.

Inside the HTML element, there are two main parts where everything goes: the head and the body. Create them using the head and body elements. The head contains all the metadata that the browser needs to know but will not display on the page. The body, on the other hand, is for all the content and is composed of various elements already discussed in this course. The body is where most of the action happens.

The doctype declaration, HTML head, and body elements are the essential building blocks of every web page.

**Document Head**

The **HTML**<head**> element** plays a crucial role in structuring an HTML document. It serves as a container for various metadata and other essential elements. Let’s delve into the specifics:

1.The script element

The script tag is a commonly used element in an HTML document's head. It instructs the browser to load a JavaScript file. Although it is typically placed at the end of the document, some also include it in the head.

* Example:

<script src=”my-javascript-file.js”></script>

2.Style Element (<style>):

* The <style> element defines style information for a single HTML page.
* You can use it to set CSS rules for elements within the document.
* Example:

<style>

body {

background-color: powderblue;

}

h1 {

color: red;

}

p {

color: blue;

}

</style>

3.Link Element (<link>):

* The <link> element establishes a relationship between the current document and an external resource.
* It is commonly used to link to external style sheets.
* Example:

<link rel="stylesheet" href="mystyle.css">

4.Meta Element (<meta>):

* The <meta> element provides additional information about the web page.
* It specifies the character set, page description, keywords, author, and viewport settings.
* Metadata is not displayed on the page but is used by browsers, search engines, and other web services.
* Examples:
  + Define character set:

<meta charset="UTF-8">

**Content Structuring**

 There are six important elements to understand:

* **Header**: Usually, a prominent strip across the top containing a big heading, logo, and perhaps a tagline. The header remains consistent across different webpages.
* **Section:** The section element is used to mark sections of content. For example, in a long essay with subheadings, each segment can be wrapped in a section element. It is also useful for dividing different topic zones on a website. Each section typically starts with its own headline.
* **Main Content**: The central area containing unique content for a specific webpage. This could be videos, articles, maps, or other relevant information.
* **Sidebar**: Peripheral information, links, ads, etc. Contextual to the main content.
* **Footer**: A strip at the bottom with fine print, copyright notices, or contact info. Often used for SEO purposes.
* **Article**: An article often starts with a title, subtitle, author's name, and publication date, which can also be considered a header. Many web pages end with a footer at the bottom, containing links, copyright information, and additional details about the company. However, footers can also appear elsewhere. Some articles begin with metadata like hashtags or share buttons, which are suitable for a footer element. The article element wraps around any type of content unit, whether it is a long-written article, a short snippet, a teaser card, a tweet, or even an app element. It represents a standalone unit of content.

**When it all Comes Together**

Let's start putting the pieces together. This will give you a sense of how we are going to assemble web pages out of hundreds of nested elements, each conveying meaning and each playing off of the other, adding up to the whole.

When you are unsure what the best way to mark up a page is, travel around the web, find similar sites, and use developer tools to see how others have decided to use HTML elements in their content. The exact way to combine these HTML elements differs with every webpage created. It is hard to be sure it is done correctly because the right way to do it always depends on what the content is, and what the page is for.

In fact, usually, there are several good ways to do it, and none of them are necessarily right or better. There is a bit of an art to structuring HTML. There is a lot of creative freedom here. We are, after all, trying to represent human communication in code, and while code may want to be correct and perfect, human connection is not always.

**Unit 9: Working with Forms and Interactive Elements**

An **HTML form** serves as means to collect user input, which is often sent to a server for processing. Here are the key components of an HTML form:

1.The <form> Element:

* The <form> element is used to create an HTML form for user input.
* It acts as a container for various input elements like text fields, checkboxes, radio buttons, and submit buttons.
* Example:

<form>

<label for="fname">First name:</label><br>

<input type="text" id="fname" name="fname"><br>

<label for="lname">Last name:</label><br>

<input type="text" id="lname" name="lname">

</form>

2.The <input> Element:

* The <input> element is the most used form element.
* It can be displayed in various ways based on the type of attribute.
* Examples of input types:

<input type="text">: Displays a single-line text input field.

<input type="radio">: Represents a radio button (for selecting one choice out of many).

<input type="checkbox">: Represents a checkbox (for selecting zero or more choices).

<input type="submit">: Displays a submit button for form submission.

<input type="button">: Creates a clickable button.

* Text fields are often defined using <input type="text">.

**3.Text Fields**:

* The <input type="text"> defines a single-line input field for text input.
* Example:

<form>

<label for="fname">First name:</label><br>

<input type="text" id="fname" name="fname"><br>

<label for="lname">Last name:</label><br>

<input type="text" id="lname" name="lname">

</form>

* By default, the width of an input field is 20 characters.

4.Radio Buttons:

* The <input type="radio"> defines a radio button.
* Radio buttons allow users to select one choice from a limited set.
* Example:

<p>Choose your favorite Web language:</p>

<form>

<input type="radio" id="html" name="fav\_language" value="HTML">

<label for="html">HTML</label><br>

<input type="radio" id="css" name="fav\_language" value="CSS">

<label for="css">CSS</label><br>

<input type="radio" id="javascript" name="fav\_language" value="JavaScript">

<label for="javascript">JavaScript</label>

</form>

5.Checkboxes:

* The <input type="checkbox"> defines a checkbox.
* Checkboxes allow users to select zero or more choices.
* Explore more about HTML form elements and their usage to create interactive and user-friendly forms!