# TERM -1  MODULE - 1

# HTML [Assignment]

**1** Are the HTML tags and elements the same thing?

- No html tags and elements are related concepts but they are not exectly the same things.

.html tags - tags are the basic building blocks of html. they are used to decline elements on web page tags consists of opening tag content and closing tag for example in HELLO WORLD  <P> is a opening tag.hello world is the contant and <P> is the closing tag

.HTML elements - An html elements consists of an opening tag content and a closing tag it is a complete set of tags that defined a specific piece of content on a web page an html elements and <P> and  are the opening and closing tag respectively.

**2**. What are tags and attributes in HTML?

- in html tags and attributes are fundamental componets  used to structure and define the content of a web page.

html tag -

- Tags are keyword enclosed in angle brackets < > that defkined structure and appreance of html elemets on a web page.

- tag usally come in pairs an opening tag and closing tag the openiong tag denots the beginnings of a elements and the closing tag signifies the end of that elemets.

html attributes -

-Attributes provide  addoitional information about html elements they are always included in the opening tag and are wrriten name value pairs

-Attributes modify the behavior or appence of the elemets and are used to pass etra infromation about the elements to the browser.

-Example: In <a href="https://www.example.com">Visit Example</a>, href is an attribute of the anchor (<a>) element. It specifies the hyperlink reference.

[<a href="https://www.example.com" target="_blank">Visit Example</a>]

3.What are void elements in HTML?

- Void elements, also known as self-closing or empty elements, are HTML elements that do not have content and do not require a closing tag. Instead, they may include attributes, and their entire definition is contained within a single opening tag. Void elements are typically used for embedding media, line breaks, images, and other elements where additional content is not applicable.

1 <img> [image elements]

example . <img src="example.jpg" alt="Example Image">

2 <br> [line break]

example . This is a paragraph.<br>This is a new line.

3 <hr> [horizontal rule]

example . This is a paragraph.

<hr>

This is another paragraph.

4 <input> [input elements]

example . <label for="username">Username:</label>

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

**4.** What are HTML Entities?

- HTML entities are special codes used to represent reserved characters, symbols, or entities in html. They are a way to include characters that have special meanings in HTML (such as <, >, &, etc.) without causing issues with the html structure. HTML entities begin with an ampersand (&) and end with a semicolon (;).

html entities

< represents < (less than)

> represents > (greater than)

& represents & (ampersand)

" represents " (double quotation mark)

' represents ' (apostrophe/single quotation mark)

example.[ 5 < 10 ]

**5**. What are different types of lists in HTML?

- In HTML, there are three main types of lists: ordered lists (<ol>), unordered lists (<ul>), and definition lists (<dl>). Each type of list is used to structure and present information in a specific way.

- ordered lists [<ol>]

An ordered list is used to represent a list of items in a specific order, typically with numbers.

Each list item is defined using the <li> (list item) tag.

- Example

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

- unordered list [ <ul> ]

An unordered list is used to represent a list of items without a specific order, typically with bullet points.

Each list item is defined using the <li> (list item) tag.

- Example

```
<ul>
  <li>Item one</li>
  <li>Item two</li>
  <li>Item three</li>
</ul>
```

Defination‌al list [ <dl> ]

A definition list is used to represent a list of terms and their corresponding definitions.

It consists of term-definition pairs, where each term is defined using the <dt> (definition term) tag and each definition is defined using the <dd>  tag.

- Example

```
<dl>
  <dt>Term 1</dt>
  <dd>Definition 1</dd>
  <dt>Term 2</dt>
  <dd>Definition 2</dd>
</dl>
```

**6**. What is the ‘class’ attribute in HTML?

- In Html, the class attribute is used to assign one or more class names to an html element. A class is a way to apply styles or behaviors to multiple elements on a web page. By defining styles in a separate CSS (Cascading Style Sheets) file or within a <style> block in the HTML document, you can apply those styles to elements with a specific class.

- tagname is the HTML element to which you want to apply the class, and classname1, classname2, etc., are the names of the classes you want to assign. Multiple classes are separated by spaces.

- Example

[ This is a highlighted paragraph. ]

**7**.What is the difference between the ‘id’ attribute and the ‘class’ attribute of HTML elements ?

- The id attribute and the class attribute in HTML serve different purposes and have distinct use cases:

- id Attribute:

The id attribute is used to uniquely identify a single HTML element on a page.

An id must be unique within the HTML document, meaning no two elements should have the same id.

- Example

[ This is a unique element. ]

- class Attribute

The class attribute is used to group together and apply styles or behaviors to multiple HTML elements.

Unlike the id, the same class can be applied to multiple elements on a page, allowing you to style or manipulate several elements in a consistent way.

- Example:

Copy code

[ This is a highlighted paragraph.

Another highlighted paragraph. ]

**8**.What are the various formatting tags in HTML?

- In HTML, formatting tags are used to structure and format the content on a web page. Here are some of the commonly used formatting tags:

- Headings (<h1> to <h6>):

Used to define headings of different levels. <h1> is the highest level, and <h6> is the lowest.

- Ex

[ <h1>This is a Heading 1</h1>

<h2>This is a Heading 2</h2>

<!-- ... -->

<h6>This is a Heading 6</h6> ]

<h6>This is a Heading 6</h6>

- Paragraph ( ):

Defines a paragraph of text.

[ This is a paragraph of text. ]

Ex -

- Bold () and Strong ():

 is used for bold text, and  is used to indicate strong importance.

[ Bold Text

Important Text ]

- Italic () and Emphasis ():

 is used for italic text, and  is used to emphasize text.

Ex -

[Italic Text

Emphasized Text ]

- Underline (<u>):

Used to underline text.

Ex -

[ <u>Underlined Text</u> ]

**9**. How is Cell Padding different from Cell Spacing?

- In HTML, specifically in the context of tables, cell padding and cell spacing are attributes that affect the layout and spacing of cells within a table.

- Cell Padding:

Cell padding refers to the space between the content of a table cell and the cell's borders.

It is controlled by the cellpadding attribute in the <table> tag.

The value of cellpadding specifies the number of pixels of space to be added inside each cell.

Ex -

```
<table cellpadding="10">
 [  <tr>
   <td>Cell 1</td>
   <td>Cell 2</td>
 </tr>
</table> ]
```

- Cell Spacing:

Cell spacing refers to the space between adjacent cells in a table.

It is controlled by the cellspacing attribute in the <table> tag.

The value of cellspacing specifies the number of pixels of space to be added between adjacent cells.

Ex -

```
<table cellspacing="5">
 <tr>
   <td>Cell 1</td>
   <td>Cell 2</td>
 </tr>
</table>
```

**10** How can we club two or more rows or columns into a single row or column in an HTML table?Certainly! Here's the theoretical explanation:

1. Merging Columns with colspan:

- The colspan attribute is used to merge columns in an HTML table.

- It is applied to a <td> (table cell) element and specifies the number of columns that the cell should span.

- This attribute allows you to create a cell that occupies multiple adjacent columns in a single row.

2. Merging Rows with rowspan :

- The rowspan attribute is used to merge rows in an HTML table.

- It is applied to a <td> (table cell) element and specifies the number of rows that the cell should span.

- This attribute allows you to create a cell that occupies multiple adjacent rows in a single column.

In both cases, the values assigned to colspan and rowspan determine the extent of the merging. You can adjust these values based on the number of columns or rows you want the cell to span.

These attributes provide flexibility in designing HTML tables, allowing for more complex layouts where cells can span across multiple rows or columns.

**11** What is the difference between a block-level element and an inline element?

The main difference between block-level elements and inline elements in HTML lies in how they are displayed and how they interact with other elements.

1. Block-level Elements:

- Display: Block-level elements typically start on a new line and take up the full width available, extending the entire width of their container.

- Structure: They create "blocks" or structural divisions in the HTML document, and they can contain other block-level and inline elements.

- Examples: , <p>, <h1> to <h6>,<ul>,<li>,<table>,<form>, etc.

2. Inline Elements :- Display: Inline elements do not start on a new line. They only take up as much width as necessary and do not force a new line after the element.

- Structure: They flow within the content and do not create new blocks. They are often used for small, inline-level styling or functionality.

Examples: , <a>, , , <img>, <br>, <input>, etc.

Usage:

- Block-level elements are commonly used for larger structural elements like paragraphs, headings, and div containers.

- Inline elements are used for smaller elements or for elements that are part of the content flow, like links and spans within a paragraph.

Nesting:

- Block-level elements can contain both block-level and inline elements.

- Inline elements, however, should only contain other inline elements or text.

-Understanding the difference between block-level and inline elements is crucial for proper HTML structure and CSS styling, as they behave differently in terms of layout and formatting within a webpage.

## 12 How to create a Hyperlink in HTML?

Certainly! Here's the theoretical explanation without an example:

To create a hyperlink in HTML

1. Anchor Element <a>

- The `<a>` (anchor) element is used to create hyperlinks in HTML.

- It is an inline-level element that defines a hyperlink by wrapping around the content intended to be clickable.

2. Href Attribute:

- The href attribute is a required attribute within the <a> element.

- It specifies the destination URL (Uniform Resource Locator) to which the hyperlink points.

- The URL can be an absolute URL (e.g., "https://www.example.com") or a relative URL (e.g., "page.html").

3. Link Text:

- The content between the opening <a> tag and the closing </a> tag represents the visible text or content of the hyperlink.

- Users click on this text to navigate to the specified URL.By using the `<a>` element with the href attribute, you can establish links between different web pages or resources within your HTML documents. This mechanism is fundamental for creating navigation and connecting content on the web.

## 13 What is the use of an iframe tag?

Certainly! Here's the theoretical explanation without an example:

The <iframe> (inline frame) tag in HTML is used for the following purposes:

1. Embedding External Content:

- <iframe> allows you to embed content from external sources, such as videos, maps, or other webpages, within the current HTML document.

2. Displaying PDFs or Documents:

- It is used to display PDF files or other document types within a webpage, providing users with the ability to view the content without leaving the page.

3. Creating Inline Frames:

- <iframe> is used to create inline frames within a document, dividing a section of the page to display content from another source while maintaining the main content's integrity.

4. Implementing Responsive Design:

- By specifying the width and height attributes or using CSS, <iframe> can be used to control the size and appearance of the embedded content, making it suitable for responsive web design.

5. Loading Content Dynamically:

- JavaScript can be employed to dynamically change the content of an <iframe> by manipulating its src attribute. This enables the creation of dynamic and interactive web pages.

It's important to use <iframe> responsibly, especially regarding security considerations. Care should be taken to ensure that the content being embedded is trustworthy, and security best practices should be followed to prevent issues like clickjacking.

## 14 What is the use of a span tag? Explain with example?

Certainly! Here's the theoretical explanation without an example:

The <iframe> (inline frame) tag in HTML is used for embedding external content within a document. Its primary purposes include:

1. Content Embedding:

- The <iframe> tag allows the inclusion of content from external sources or other webpages directly within the current HTML document.

2. External Source Integration:

- It is commonly used to embed content such as videos, maps, or documents from external sources like YouTube, Google Maps, or PDF files.

3. Inline Frames:- <iframe> creates an inline frame, enabling the display of content within a specific rectangular area on the page while keeping the rest of the content intact.

4. Responsive Design:

- The width and height attributes of the <iframe> tag can be adjusted to control the size of the embedded content, facilitating the implementation of responsive design.

5. Dynamic Content Loading:

- JavaScript can be used to dynamically change the content of an <iframe> by manipulating its src attribute. This allows for the loading of content dynamically based on user interactions.

6. Isolation of Content:

- <iframe> provides a level of isolation, allowing the embedded content to function independently of the main document. This can be useful for encapsulating and securing external content.

When using <iframe>, it's essential to consider security implications, especially when embedding content from external sources. Proper validation and trust in the embedded content source help prevent security vulnerabilities such as clickjacking.

## 15 How to insert a picture into a background image of a web page?

Certainly! Here's the theoretical explanation without an example:

To insert a picture into the background image of a web page, you typically use CSS (Cascading Style Sheets). The process involves:

1. HTML Structure:

- Ensure you have a basic HTML structure in place, with appropriate tags for your content.

2. CSS Styling:

- Use the CSS `background-image` property to set the background image for a specific element (e.g., the body or a container element).

- Adjust other background properties such as background-size, background-position, and background-repeat` based on your design requirements.

3. Overlay Image:

- If you want to overlay another image on top of the background image, use additional HTML and CSS to position and style an element containing the overlay image.

- The overlay image can be positioned using CSS properties like position: absolute; and adjusted with top, left, right, or bottom properties.

4. Responsive Design:

- Consider making your design responsive by using percentage-based widths, heights, or media queries to adapt to different screen sizes.

5. Image Paths:

- Ensure that the paths to your image files are correct, and the images are accessible from your HTML file.

- Specify the correct file extensions (e.g., .jpg,.png) for your image files.

By applying these principles, you can create a web page with a background image and overlay additional images on top for a visually appealing design. Adjust the CSS properties based on your specific design goals and layout requirements.

## 16 How are active links different from normal links?

Active links and normal links refer to different states of a hyperlink on a web page. The distinction between them is based on the user's interaction with the link. Here's an explanation of the differences:

1. Normal Links:

- Normal links, also known as default or unvisited links, represent the initial state of a hyperlink.

- These links have not been clicked or interacted with by the user.

- By default, browsers usually display unvisited links with a different color (commonly blue) to differentiate them from visited links.

2. Active Links:

- Active links represent the state of a hyperlink while it is being clicked or activated by the user.

- This state is transitional and occurs in the split second when a user is clicking on the link but hasn't released the mouse button yet.

- Stylistically, active links often have a different appearance, such as a change in color or background, to provide visual feedback to the user during the clicking process.

- The active state is followed by the visited state if the user releases the mouse button and the link has been successfully activated.

In summary, the primary difference lies in the user's interaction:

- Normal links represent the default, unvisited state.

- Active links represent the state of a link while it is being clicked or interacted with.Both normal and active link states are part of the overall link styling, and you can customize their appearance using CSS to enhance the user experience on your website.

## 17 What are the different tags to separate sections of text?

Certainly! Here's the theoretical explanation without examples:

1. Heading Tags <h1> to <h6>: - Heading tags represent different levels of section headings in HTML, ranging from <h1> (highest importance) to <h6> (lowest importance).

2. Paragraph Tag <p>:

- The <p> tag is used to define paragraphs of text, separating blocks of content into individual paragraphs.

3. Division Tag :

- The  tag is a generic container used to group and separate sections of content. It is a block-level element without inherent semantic meaning.

4. Section Tag <section>:

- The <section> tag defines thematic groupings of content within a document. It helps separate distinct sections or chapters of content.

5. Article Tag <article>:

- The `<article>` tag represents an independent piece of content that can be distributed and reused. It is often used to separate articles or blog posts.

6. Header and Footer Tags (<header>, <footer>):

- The <header>and <footer> tags define header and footer sections of a document or a specific section, providing structural organization.

These tags play a crucial role in structuring HTML documents, providing semantic meaning to different sections of content. Proper use of these tags enhances document readability, accessibility, and search engine optimization (SEO).

## **18** What is SVG?

Scalable Vector Graphics (SVG) is an XML-based vector image format designed for describing two-dimensional vector graphics. Unlike raster images (e.g., JPEG, PNG, GIF), which are composed of a grid of pixels and may lose quality when resized, SVG graphics are resolution-independent and can be scaled to any size without losing clarity.

Key features of SVG include:

1. Vector Format:

- SVG images are defined using mathematical expressions to describe shapes, lines, and colors.

- They are composed of paths, shapes, and text elements, making them scalable without loss of quality.

2. XML-Based:

- SVG files are written in XML (eXtensible Markup Language), making them human-readable and easily editable with a text editor.

3. Interactivity and Animation:

- SVG supports interactivity and animation through JavaScript. Elements can be scripted to respond to user actions or events.

4. Accessibility:

- SVG images can be embedded directly into HTML documents, allowing for easy integration with web content.

- Text content within SVG is selectable and searchable, enhancing accessibility.

5. Wide Browser Support:

- Most modern web browsers support SVG, making it a widely adopted standard for web graphics.

6. Inline and External Usage:

- SVG images can be embedded directly within HTML using the <svg> element, or they can be included as external files.

7. Graphics Editing Software Compatibility:

- SVG is compatible with various graphics editing software, and many design tools can export images in SVG format.

8. Scalability:

- SVG graphics can be scaled, rotated, and transformed without loss of quality, making them suitable for responsive design and a variety of display sizes.

SVG is commonly used for web graphics, logos, icons, charts, maps, and other types of visual content on the web. Its ability to provide high-quality graphics while maintaining a small file size and its support for interactivity make SVG a versatile and popular choice for web developers and designers.

## **19** What is difference between HTML and XHTML ?

HTML (Hypertext Markup Language) and XHTML (Extensible Hypertext Markup Language) are both markup languages used for creating and structuring content on the web. However, there are some key differences between the two:

1. Syntax:

- HTML: HTML has a more lenient syntax. It allows for some flexibility in writing tags and attributes. Tags can be written in uppercase or lowercase, and attribute values don't necessarily need to be enclosed in double or single quotes.

- XHTML : XHTML has a stricter syntax. It requires tags to be written in lowercase, attribute values must be enclosed in quotes, and all tags must be properly nested.

2. Document Structure:

- HTML: HTML documents have a looser structure. Elements can be opened and closed without strict adherence to XML rules. XHTML: XHTML follows the rules of XML and requires a well-formed document structure. Every element must be properly nested, and tags must be closed in the correct order.

3. Case Sensitivity:

- HTML: HTML is case-insensitive. Tags and attribute names can be written in uppercase or lowercase.

- XHTML: XHTML is case-sensitive. All tags and attribute names must be written in lowercase.

4. Attribute Minimization:

- HTML: HTML allows attribute minimization, where certain boolean attributes like checked, disabled, or readonly can be written without a value (e.g., <input checked>).

- XHTML: XHTML does not allow attribute minimization. All attributes must have a value (e.g.,<input checked="checked">).

5. Error Handling:

- HTML: HTML browsers are forgiving and often render documents even if they contain errors. Browsers attempt to interpret and display the content, even if the markup is not entirely correct.

- XHTML: XHTML has stricter error handling. If an XHTML document contains errors, browsers may not render it properly or at all. Errors can result in a "parsing error" and may prevent the document from being displayed.

6. Media Types:

- HTML: HTML is primarily used for web browsers. It does not explicitly define a content type.

- XHTML: XHTML is designed to work with different types of XML processors and is more versatile in terms of content type. It can be used with various XML applications beyond web browsers.

While both HTML and XHTML are used to create web content, the choice between them depends on factors such as the desired syntax, error handling, and the specific requirements of the project or application. HTML5, the latest version of HTML, incorporates some features of XHTML, and modern web development often follows HTML5 standards.

## **20** What are logical and physical tags in HTML?

In HTML, the terms "logical tags" and "physical tags" are often used to describe different types of tags based on their purpose or function within the document. These terms were more prominent in older versions of HTML but are less relevant in modern web development, especially with the adoption of HTML5. However, the distinction is still occasionally used to explain certain concepts.

1. Logical Tags:

- Logical tags, sometimes referred to as "semantic tags," are tags that define the meaning or purpose of the content they enclose.

- These tags convey the structure and meaning of the content, making it more understandable to both developers and browsers.

- Examples of logical tags include <header>, <footer>, <nav>, <article>, <section>, <aside>, <figure>, and <figcaption>. These tags were introduced in HTML5 to provide a more meaningful structure to web documents.

2. Physical Tags:

- Physical tags, on the other hand, are tags that define the appearance or presentation of the content.

- They are focused on the visual styling and layout of the document.

- Examples of physical tags include older HTML tags like <b> (bold), <i> (italic), <font>, <u> (underline), and <strike> (strikethrough). These tags were more commonly used in HTML 4 and earlier versions to apply styling directly to the content.

It's important to note that with the evolution of HTML and the emphasis on semantic markup, the use of physical tags has become less common. Instead, modern web development encourages the use of logical tags that provide a clearer structure and meaning to the content. Cascading Style Sheets (CSS) is now the preferred method for styling and presentation, allowing for a cleaner separation of content and design.

In summary, while the terms "logical tags" and "physical tags" were more relevant in older HTML versions, the shift towards semantic HTML and the separation of content and presentation with CSS has made the distinction less pronounced in modern web development.