Web Designing Assignment Term-1

Module (HTML) -1

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

⇨ No, HTML tags and elements are not exactly the same thing, although they are closely related.

⇨ **HTML Tags:** Tags are the fundamental building blocks of HTML. They are used to define elements within an HTML document. Tags are composed of angle brackets < > and usually come in pairs: an opening tag and a closing tag. For example, is an opening tag for a paragraph element, and is its corresponding closing tag.

⇨ **HTML Elements:** Elements are made up of tags, content, and attributes. An element consists of a start tag, content, and an end tag. The start tag marks the beginning of the element, the end tag marks the end of the element, and the content is what's between the start and end tags. For example, in Hello, World! , and together form the element, with "Hello, World!" being the content.

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

⇨ **Tags:** HTML tags are the building blocks of HTML documents. They define the structure and content of the elements within the document. Tags are enclosed in angle brackets < > and typically come in pairs: an opening tag and a closing tag. The opening tag denotes the beginning of an element, and the closing tag marks its end. For example, is an opening tag for a paragraph element, and is its corresponding closing tag.

⇨ **Attributes:** HTML attributes provide additional information about an element. They are added to the opening tag of an element and consist of a name-value pair separated by an equals sign. Attributes modify the behavior or appearance of an element and are used to specify things like the element's

ID, class, source URL, and more. For example, the href attribute in an <a> tag specifies the URL the link should point to, and the src attribute in an <img> tag specifies the image source. Example: Click Here.

3. What are void elements in HTML?

⇨ In HTML, void elements, also known as empty elements or self-closing elements, are elements that do not have any content between an opening tag and a closing tag. These elements are self-contained and do not require a separate closing tag. Void elements are used to insert multimedia, line breaks, images, and other types of content into a webpage.

⇨ Examples of void elements include:
<img>: Inserts an image into the document.

: Inserts a line break.
<hr>: Inserts a horizontal line.
<input>: Defines an input control.
<meta>: Provides metadata about the HTML document.

4. What are HTML Entities?

⇨ Html entities are special codes used to represent reserved characters, symbols, and other entities in HTML documents. They are primarily used to ensure that these characters render correctly in web browsers, especially when the characters have special meanings in HTML or might be misinterpreted by the browser.

⇨ There are two main types of HTML entities:

⇨ **Character Entities:** Character entities are used to represent special characters that have reserved meanings in HTML, such as <, >, ", ', and &. These entities ensure that these characters are displayed as intended without being interpreted as part of HTML syntax. For example:

< represents the less-than symbol <.

> represents the greater-than symbol >.

" represents the double quotation mark ".

' represents the apostrophe (single quotation mark) '.

& represents the ampersand &.

⇨ **Numeric Character References:** Numeric character references allow you to represent characters using their Unicode code points. Each character has a unique Unicode code point, which can be represented as either decimal (&#xxx;) or hexadecimal (&#xhhhh;) values, where xxx or hhhh are the code point numbers. For example:

< represents the less-than symbol <.

> represents the greater-than symbol >.

" represents the double quotation mark ".

' represents the apostrophe (single quotation mark) '.

& represents the ampersand &.

5. What are different types of lists in HTML?

⇨ In HTML, there are three main types of lists:

⇨ **Ordered List(<ol>):** An ordered list is a list of items where each item is marked with a number or another sequence indicator. It is typically used when the order of items matters. Each item in an ordered list is wrapped within `<li>` (list item) tags. For

⇨ **Unordered List (<ul>):** An unordered list is a list of items where each item is marked with a bullet point or another similar symbol. It is typically used when the order of items does not matter. Like ordered lists, each item in an unordered list is wrapped within `<li>` tags.

Definition List (<dl>): A definition list consists of a series of terms and their definitions. It is created using `<dl>` (definition list), `<dt>` (definition term), and `<dd>` (definition description) tags. Each `<dt>` tag represents a term, and each `<dd>` tag represents its corresponding definition.

6. **What is the 'class' attribute in HTML?**

⇨ In HTML, the class attribute is used to specify one or more class names for an element. Classes are used to apply styles or behavior to multiple elements on a web page. The class attribute allows you to group elements together based on shared characteristics or purposes, which can then be targeted and styled using CSS (Cascading Style Sheets) or manipulated with JavaScript.

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

⇨ **Purpose:**

id Attribute: The id attribute is used to uniquely identify a single element on a web page. Each id value must be unique within the entire HTML document. It is typically used when there is only one instance of an element with a specific identifier or when an element requires unique identification for styling or scripting purposes.

class Attribute: The class attribute is used to group multiple elements together based on shared characteristics or purposes. Multiple elements can share the same class, and a single element can have multiple classes. It is used for styling or scripting purposes when applying styles or behaviors to multiple elements simultaneously.

⇨ **Uniqueness:**

id Attribute: Each id value must be unique within the HTML document. Attempting to assign the same id value to multiple elements will result in invalid HTML and may cause unexpected behavior.

class Attribute: Multiple elements can share the same class, and a single element can have multiple classes. Classes can be reused across multiple elements within the same document.

⇨ **Styling and Scripting:**

id Attribute: id attributes are often used for unique styling or scripting purposes. They provide a way to target specific elements directly in CSS or JavaScript.
class Attribute: class attributes are commonly used for applying styles or behaviors to multiple elements at once. They allow for efficient styling and scripting by grouping related elements together.

8. What are the various formatting tags in HTML?

⇨ **I**n HTML, formatting tags are used to specify the appearance or style of text or content within a webpage. Here are some common formatting tags in HTML:

⇨ **<b>:** This tag is used to make text bold.

Example: <b>This text is bold</b

⇨ **:** Similar to <b> this tag is used to indicate strong importance, often displayed as bold by browsers.

Example: This text is strongly emphasized

⇨ **:** This tag is used to make text italicized.

Example: This text is italicized

⇨ **:** Similar to , this tag is used to indicate emphasis, often displayed as italicized by browsers

Example: This text is emphasized

⇨ **<u>:** This tag is used to underline text.

Example: <u>This text is underlined</u>

⇨ **<s> or <strike>:** These tags are used to create strikethrough text.

Example: <s>This text has a strikethrough</s>

⇨ **<sub>:** This tag is used to display subscript text.

Example: H₂O

⇨ **<sup>:** This tag is used to display superscript text.

Example: x²

⇨ **<mark>:** This tag is used to highlight text with a marker.

Example: <mark>This text is highlighted</mark>

⇨ **<small>:** This tag is used to display text in a smaller font size.

Example: <small>This text is smaller</small>

⇨ **<big>:** This tag (deprecated in HTML5) was used to display text in a larger font size.

Example: <big>This text is bigger</big>

⇨ **:** This tag (deprecated in HTML5) was used to define the font characteristics of text.

Example: `This text is red

9. How is Cell Padding different from Cell Spacing?

⇨ In HTML tables, both cell padding and cell spacing are attributes used to control the space around and between cells, respectively. However, they serve different purposes and affect different aspects of the table layout:

⇨ **Cell Padding:**

Purpose: Cell padding controls the space between the content of a cell and the cell's borders.
Attribute: Cell padding is set using the cellpadding attribute within the <table> tag or via CSS.
Effect: When cell padding is applied, it adds space inside each cell, between the cell's content and its borders. This space is uniform around all sides of the cell's content.
Example: <table cellpadding="10">

⇨ Cell Spacing:

Purpose: Cell spacing controls the space between adjacent cells in the table.
Attribute: Cell spacing is set using the cellspacing attribute within the <table> tag or via CSS.
Effect: When cell spacing is applied, it adds space between the borders of adjacent cells. This space is visible as a gap or margin between cells.
Example: <table cellspacing="5">

10. How can we club two or more rows or columns into a single row or column in an HTML table?

Merging Rows (Vertical Merge):

⇨ To merge multiple rows into a single row, use the rowspan attribute on a cell within the first row you want to merge.

⇨ The value of rowspan specifies the number of rows the cell should span.
Apply rowspan only to the first cell in the group of rows you want to merge.

⇨ The subsequent cells in the same column of the following rows will be automatically merged into the single row.

Merging Columns (Horizontal Merge):

⇨ To merge multiple columns into a single column, use the colspan attribute on a cell within the first column you want to merge.

⇨ The value of colspan specifies the number of columns the cell should span.

⇨ Apply colspan only to the first cell in the group of columns you want to merge.

⇨ The subsequent cells in the same row of the following columns will be automatically merged into the single column.

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

Layout Behavior:

⇨ Block-level elements start on a new line and take up the full width available, creating a block-level box in the layout.

⇨ Inline elements flow within the text content and only take up as much width as necessary, allowing them to appear alongside other inline elements.

Default Display:

⇨ Block-level elements have a display property set to block by default.

⇨ Inline elements have a display property set to inline by default.

Allowed Content:

- ⇨ Block-level elements can contain both block-level elements and inline elements.
- ⇨ Inline elements cannot contain block-level elements but can contain other inline elements.

12. How to create a Hyperlink in HTML?

- ⇨ In HTML, you can create hyperlinks using the <a> (anchor) element. Hyperlinks are used to navigate to other web pages or resources by clicking on them. Here's how to create a hyperlink in HTML:

```
<a href="URL">Link Text</a>
```

- ⇨ <a>: This is the anchor element used to create the hyperlink.
- ⇨ href="URL": This attribute specifies the destination URL that the hyperlink points to.
- ⇨ Replace "URL" with the actual URL of the webpage or resource you want to link to.

  For example, https://www.example.com

- ⇨ Link Text: This is the text that will be displayed as the clickable link. Replace `"Link Text"` with the text you want to display as the link.

13. What is the use of an iframe tag?

⇨ The <iframe> (inline frame) tag in HTML is used to embed another HTML document within the current document. It allows you to display content from another webpage or resource within a designated area of the current webpage. The <iframe> element creates a window (or frame) in the current webpage where the content of the specified URL is loaded.

⇨ Here's the basic syntax of the <iframe> tag:

```
<iframe src="URL" width="width" height="height"
frameborder="0"></iframe>
```

⇨ **src:** This attribute specifies the URL of the webpage or resource you want to embed within the iframe.

⇨ **Width and height:** These attributes specify the dimensions of the iframe window.

⇨ **frameborder:** This attribute specifies whether or not to display a border around the iframe. Setting it to `"0"` removes the border.

Example usage of the `<iframe>` tag:

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

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

⇨ The tag in HTML is a generic inline container that is used to apply styles to a specific section of text or to group inline elements together for styling purposes. Unlike block-level elements such as , which create a new block-level container, the element does not create a new line or break in the content flow. Instead, it wraps around content and applies styles or functionality to that content.

⇨ Here's an example of how the tag can be used:

```
<p>This is a <span style="color: blue">blue</span> text.</p>
```

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

⇨ If you want to insert a picture into a background image of a web page,you can achieve this by using HTML and positioning the images appropriately. One way to do this is by using the <img> tag to insert the picture onto the page and positioning it absolutely or relatively within a container that has the background image set.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Background Image with Picture Example</title>
</head>
<body>
    <div style="position: relative; width: 100%; height: 400px; background-image: url('background-image.jpg'); background-size: cover; background-position: center;">
        <img src="picture.jpg" alt="Picture" style="position: absolute; top: 50%; left: 50%; transform: translate(-50%, -50%); max-width: 100%; max-height: 100%;">
    </div>
    <div>
        <h1>Welcome to my website</h1>
        <p>This is some text on the webpage.</p>
    </div>
</body>
</html>
```

16.How are active links different from normal links?

⇨ Active links and normal links differ in their appearance and behavior based on the user's interaction.

1. **Normal Links:**

⇨ Normal links, also known as unvisited links, are displayed in their default state before any interaction from the user.

⇨ They typically appear as underlined text in a browser's default link color.

⇨ Once visited, they may change color to indicate their visited state, depending on the browser's default settings.

2. **Active Links:**

⇨ Active links represent links that are currently being interacted with by the user.

⇨ They are activated when the user clicks on them but hasn't released the mouse button yet.

⇨ Active links may have a different appearance compared to normal links to provide visual feedback to the user that the link is being interacted with.

⇨ Commonly, active links are styled to indicate that they are being clicked, such as changing their color or background color, or applying an underline or other visual effect.

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

⇨ In HTML, there are various tags you can use to separate sections of text based on their semantic meaning or intended purpose. Here are some common ones:

Paragraph :

⇨ Used to represent paragraphs of text.

⇨ Creates a block-level element with space above and below the text.
Example: ` This is a paragraph of text.

Heading<h1> to <h6>:

⇨ Used to represent headings of different levels.

⇨ <h1> is the highest level (most important), and `<h6>` is the lowest level (least important).

⇨ Creates block-level elements with varying font sizes and default margins.

⇨ Example: `<h1>This is a heading</h1>`

Division :

⇨ Used to divide content into sections or groups for styling or scripting purposes.

⇨ Provides a generic container for content.

⇨ Does not imply any specific meaning or styling by itself.
Example: `<div>This is a division.</div>`

Span:

- ⇨ Used to apply styles or scripting to specific portions of text within a paragraph or other inline elements.

- ⇨ Creates an inline element that does not introduce line breaks.
  Example: This is a highlighted text.
  Preformatted<pre>:

- ⇨ Used to display preformatted text, such as code or ASCII art, preserving white spaces and line breaks.

- ⇨ Maintains the original formatting of the text.
  Example: <pre>This is preformatted text with multiple spaces and line breaks. </pre>

  Blockquote <blockquote>:

- ⇨ Used to represent a section of quoted text from another source.

- ⇨ Typically indents the text and provides visual distinction.
  Example: `<blockquote>This is a quoted text.</blockquote>`

  Horizontal Rule<hr>:

- ⇨ Used to create a thematic break or horizontal rule between sections of content.

- ⇨ Renders as a horizontal line across the page.
  Example: <hr>

18. What is SVG?

⇨ SVG stands for Scalable Vector Graphics. It is an XML-based vector image format for two-dimensional graphics with support for interactivity and animation. SVG images are scalable, meaning they can be resized without loss of quality, making them suitable for a wide range of applications, from simple icons to complex illustrations.

Here are some key features and characteristics of SVG:

⇨ **Vector-Based:** SVG images are defined using geometric shapes, such as lines, curves, and polygons, rather than pixels. This allows them to be scaled to any size without losing sharpness or clarity.

⇨ **Text-Based Format:** SVG files are written in XML, which is a plain text format. This makes SVG files human-readable and easily editable with any text editor or vector graphics software.

⇨ **Scalability:** SVG images can be scaled up or down to any size without loss of quality. This makes them ideal for responsive web design and high-resolution displays.

⇨ **Interactivity:** SVG supports interactivity through event handlers and scripting using JavaScript. This allows for dynamic and interactive graphics, such as hover effects, clickable regions, and animations.

⇨ **Accessibility:** SVG images can be easily styled and annotated using CSS and HTML attributes, making them accessible to screen readers and other assistive technologies.

⇨ **Animation:** SVG supports animation through SMIL (Synchronized Multimedia Integration Language) or JavaScript, allowing for the creation of animated graphics and effects.

⇨ **Browser Support:** Most modern web browsers support SVG, including Chrome, Firefox, Safari, Edge, and Opera. Internet Explorer (IE) has limited support for SVG, particularly in older versions.

19. What is difference between HTML and XHTML?

⇨ HTML (Hypertext Markup Language) and XHTML (Extensible Hypertext Markup Language) are both markup languages used to create web pages, but they have some key differences in syntax and rules. Here are the main differences between HTML and XHTML:

⇨ **Syntax:**

HTML: HTML has more lenient syntax rules compared to XHTML. For example, in HTML, elements don't necessarily need to be closed, and attribute values don't need to be quoted.

⇨ **XHTML:** XHTML follows stricter syntax rules based on XML. All elements must be properly nested and closed, and attribute values must be quoted. XHTML documents must also be well-formed XML documents.

Document Structure:

⇨ **HTML:** In HTML, documents can have optional elements and attributes, and elements like `<html>`, `<head>`, and `<body>` are often implied rather than required.

⇨ **XHTML:** XHTML documents must follow a stricter structure, with a mandatory DOCTYPE declaration, root element <html>, head section <head>, and body section <body>. All elements and attributes must be explicitly declared.

Case Sensitivity:

⇨ **HTML:** HTML is case-insensitive. Tags, attribute names, and attribute values can be written in any case (lowercase, uppercase, or mixed case).

⇨ **XHTML:** XHTML is case-sensitive. Tags, attribute names, and attribute values must be written in lowercase.

Error Handling:

- **HTML:** In HTML, browsers are forgiving of errors, and they try to render the content even if the markup is not well-formed or contains errors.

- **XHTML:** XHTML requires strict adherence to XML rules, and even minor errors can cause the document to fail to render properly. Browsers may be less forgiving of syntax errors in XHTML documents.

MIME Type:

- **HTML:** The MIME type for HTML documents is usually text/html.

- **XHTML:** The MIME type for XHTML documents is usually `application/xhtml+xml`.

Compatibility:

- **HTML:** HTML is widely supported by web browsers and is the standard markup language for most web pages.

- **XHTML:** XHTML adoption has been slower due to its stricter syntax requirements. While modern web browsers support XHTML, older browsers may have limited or inconsistent support.

20. What are logical and physical tags in HTML?

⇨ In HTML, the terms "logical tags" and "physical tags" refer to different approaches to structuring and styling content within a web page.

Logical Tags:

⇨ Logical tags are elements that describe the meaning or purpose of the content, rather than dictating its appearance.

⇨ They are focused on semantics, or the meaning of the content, and are intended to provide structure and organization to the document.

⇨ Examples of logical tags include <header>, <footer>, <nav>, <article>, <section>, <aside>, <main>, <figure>, <figcaption>, etc.

⇨ Logical tags are typically used to create a well-structured, semantically meaningful document that is accessible to both humans and machines.

Physical Tags:

⇨ Physical tags are elements that are primarily used for formatting and styling purposes, rather than conveying the meaning or structure of the content.

⇨ They are focused on the presentation, appearance, and layout of the content on the page.

⇨ Examples of physical tags include <b> (bold), <i> (italic), <u> (underline), <font>, <strike>, , <s>, etc.

⇨ Physical tags are often associated with older HTML practices and are generally discouraged in modern web development due to their lack of semantic meaning and separation of concerns.