# HTML ASSIGNMENT

**Module 1) – Foundation**

**Do search for web-site, http, URL etc given topics in lecture.**

**Answer:**

in a lecture about web development or internet-related topics, discussing concepts like "website," "HTTP," and "URL" could cover the following points:

1. *Website*: A collection of web pages grouped under a single domain and accessible over the internet. It's designed to provide information, services, or resources to users. Websites can range from simple static pages to complex web applications.

2. *HTTP (Hypertext Transfer Protocol)*: It's the protocol used for transferring data over the web. It governs how web servers and browsers communicate, enabling the retrieval of HTML documents, images, videos, etc. HTTP follows a request-response model where a client (browser) sends a request to a server, and the server responds with the requested data.

3. *URL (Uniform Resource Locator)*: It's the address used to identify resources on the internet. A URL consists of several components:

- Protocol (e.g., http://, https://)
- Domain name (e.g., www.example.com)

- Path (optional, specifying the location of a specific resource on the server)

- Query parameters (optional, used for passing data to a web server)

Explaining these concepts in a lecture might involve discussing their functionalities, how they work together to deliver content on the web, and their significance in understanding and building web-based applications or services.

**Module 2) Fundamentals of IT**

**Do search for domain, hosting, SEO etc.**

**Answer:**

Certainly! In a module discussing web development or online presence, terms like "domain," "hosting," and "SEO" play crucial roles:

1. *Domain*: It's the unique name that identifies a website. Domains are used in URLs to access websites on the internet. For instance, in the URL "www.example.com," "example.com" is the domain name. Domains are registered and managed through domain registrars.

2. *Hosting*: Web hosting refers to the service that allows individuals or organizations to make their websites accessible via the World Wide Web. It involves storing website files, databases, and other resources on servers that are connected to the

internet. Web hosting providers offer various hosting options such as shared hosting, VPS (Virtual Private Server) hosting, dedicated hosting, etc.

3. *SEO (Search Engine Optimization)*: It's the process of optimizing a website to increase its visibility and rank higher in search engine results pages (SERPs). SEO involves various strategies and practices, including optimizing content with relevant keywords, improving website structure and user experience, obtaining quality backlinks, and adhering to search engine guidelines to enhance organic traffic.

Discussing these concepts in a module might involve exploring their importance in establishing an online presence, understanding how domain names play a role in branding and accessibility, how hosting services support website availability, and the significance of SEO in improving a website's visibility and traffic from search engines.

**Module 3) HTML**

**Q.1 Are the HTML tags and elements the same things ?**

**Answer:**

HTML tags and elements are related but not exactly the same. Tags are the markup characters used to define the start and end of an element, while elements are the building blocks of an HTML document defined by tags and typically include content and attributes enclosed within those tags. Essentially, elements are made up of opening and closing tags, along with their content, whereas tags are the markup used to

create elements.

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

**Answer:**

In HTML, tags are markup elements used to define the structure and appearance of content. They enclose content and typically consist of an opening tag, content, and a closing tag. Attributes, on the other hand, provide additional information about an element and are included within the opening tag. Attributes modify the behavior or appearance of an element and are written as name-value pairs within the tag. They enhance the functionality or presentation of the HTML elements.

**Q.3 What are void elements in HTML? With Example.**

**Answer:**

Void elements in HTML are those that don't have any content nested within them and don't require a closing tag. They self-close in the opening tag itself. Examples of void elements include `<br>`, `<img>`, `<input>`, `<meta>`, `<link>`, and `<hr>`. For instance:

html

```
<img src="example.jpg" alt="Example Image">
<br>
<input type="text" name="username">
```

In these examples, `<img>`, `<br>`, and `<input>` are void elements that don't need a closing tag because they don't contain any content.

**Q.4 What are HTML Entities? With Example.**

**Answer:**

HTML entities are special codes used to represent characters that have special significance in HTML or characters that are not easily typed or displayed with a keyboard. These entities start with an ampersand `&`, followed by a code or name, and end with a semicolon `;`. For instance:

- `&amp;` represents the ampersand character `&`.
- `&lt;` represents the less-than sign `<`.
- `&gt;` represents the greater-than sign `>`.
- `&quot;` represents the double quotation mark `"`.
- `&copy;` represents the copyright symbol `©`.

For example:

html

This is an example of an & entity.

Copyright © 2023

In this code, `&amp;` represents the ampersand in the first paragraph, and `&copy;` represents the copyright symbol in the second paragraph.

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

**Answer:**

In HTML, there are three main types of lists:

1. *Ordered Lists (`<ol>`)*: These are numbered lists where each list item is prefixed with a number. Example:

```
html
<ol>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
</ol>
```

2. *Unordered Lists (`<ul>`)*: These lists are bulleted, with each list item typically prefixed with a bullet point. Example:

```
html
<ul>
    <li>Item A</li>
    <li>Item B</li>
    <li>Item C</li>
</ul>
```

3. *Definition Lists (`<dl>`)*: These lists consist of terms and their corresponding definitions. Each term is defined using `<dt>`, and its definition is enclosed in `<dd>`. Example:

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

These different list types allow you to structure and present information in various formats within an HTML document.

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

**Answer:**

The 'class' attribute in HTML is used to assign one or more class names to an element, allowing the element to be styled or targeted by CSS or JavaScript. It doesn't affect how the element behaves or functions; rather, it serves as a hook for styling or scripting purposes.

Here's an example of how the 'class' attribute is used:

html

This is a paragraph with a specific class applied.

In this example, the 'class' attribute is applied to a `<p>` (paragraph) element, assigning it the class name 'highlighted'. With CSS, you can then target this class to apply specific styling, like changing the text color, font, background, or any other stylistic properties associated with the 'highlighted' class.

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

**Answer:**

The 'id' attribute and the 'class' attribute in HTML serve different purposes:

- *'id' attribute*: It uniquely identifies an element on a page. Each 'id' must be unique within the HTML document. It's typically used when you want to target a specific element for styling or scripting purposes.

- *'class' attribute*: It's used to group elements together based on similar characteristics. Multiple elements can share the same 'class'. It's primarily used to apply common styling or scripting to several elements.

Here's an example demonstrating their use:

html

<h2 id="main-heading">Main Heading</h2>

This paragraph has a specific class applied.

Another paragraph with the same class.

In this example, the 'id' attribute is used to uniquely identify the main heading (`<h2>`) element as 'main-heading'. The 'class' attribute is applied to two `<p>` elements, both using the 'highlighted' class to style them similarly. The 'id' is unique, while

the 'class' can be applied to multiple elements sharing common characteristics.

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

**Answer:**

In HTML, there are several formatting tags that allow you to control the appearance or style of the content:

1. `<strong>`: Indicates strong importance or emphasis (usually displayed as bold).

2. `<em>`: Denotes emphasized text (typically displayed in italics).

3. `<u>`: Underlines the enclosed text.

4. `<s>`: Renders text with a strikethrough.

5. `<mark>`: Highlights text in a specific color to indicate relevance.

6. `<sub>`: Formats text as subscript (displayed below the normal text line).

7. `<sup>`: Formats text as superscript (displayed above the normal text line).

8. `<code>`: Represents a snippet of code.

9. `<pre>`: Defines preformatted text, maintaining spaces and line breaks.

These tags are used to convey meaning or apply specific styling to the content within an HTML document. However, it's often

recommended to use CSS for styling purposes instead of these tags for better separation of content and presentation.

**Q.9 How is Cell Padding different from Cell Spacing? With Example.**

**Answer:**

Cell padding and cell spacing are attributes used in HTML tables to control the space around the content within a table cell and the space between the cells, respectively.

- *Cell Padding (`cellpadding`)*: It defines the space between the content of a table cell and the cell's border. This attribute determines the internal spacing within each cell.

- *Cell Spacing (`cellspacing`)*: It defines the space between individual cells in a table. It controls the gap or distance between adjacent cells.

Here's an example demonstrating their use:

html

```
<table border="1" cellpadding="10" cellspacing="5">
  <tr>
    <td>Cell 1</td>
    <td>Cell 2</td>
```

```
    </tr>
    <tr>
        <td>Cell 3</td>
        <td>Cell 4</td>
    </tr>
</table>
```

In this example, `cellpadding="10"` specifies a padding of 10 pixels within each cell, adding space between the content and the cell border. `cellspacing="5"` creates a 5-pixel gap between adjacent cells in the table.

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

**Answer:**

In HTML tables, you can merge cells to create a single larger cell across rows or columns using the `rowspan` and `colspan` attributes.

- *`rowspan`*: Merges cells vertically, spanning multiple rows.

- *`colspan`*: Merges cells horizontally, spanning multiple columns.

Here's an example demonstrating how to merge cells:

```
html
<table border="1">
  <tr>
    <td rowspan="2">Row 1, Col 1</td>
    <td colspan="2">Row 1, Col 2 and Col 3</td>
  </tr>
  <tr>
    <td>Row 2, Col 2</td>
    <td>Row 2, Col 3</td>
  </tr>
</table>
```

In this example:

- The first cell spans two rows (`rowspan="2"`), merging Row 1 and Row 2 in the first column.

- The second cell spans two columns (`colspan="2"`), merging Col 2 and Col 3 in the first row.

This technique allows you to create more complex table layouts by merging cells across rows and columns.

**Q.11 What is the difference between a block-level element and**

**an inline element?**

**Answer:**

Block-level elements and inline elements in HTML differ in their behavior and how they interact with other elements on a webpage.

- *Block-level elements*:

  - They typically start on a new line and stretch out to the full width available.

  - Examples include `<div>`, `<p>`, `<h1>` - `<h6>`, `<ul>`, `<li>`, `<table>`, etc.

  - They can contain other block-level and inline elements.

- *Inline elements*:

  - They don't start on a new line and only occupy the space that their content needs.

  - Examples include `<span>`, `<a>`, `<strong>`, `<em>`, `<img>`, `<input>`, `<br>`, etc.

  - They can't contain block-level elements and are often used within block-level elements to style portions of text or elements within a block.

In summary, block-level elements typically create structure and layout, while inline elements are more focused on styling and manipulating text or smaller portions within a block-level element.

**Q.12 How to create a Hyperlink in HTML? With Example.**

**Answer:**

To create a hyperlink in HTML, you use the `<a>` tag (anchor tag) with the `href` attribute to specify the destination URL. Here's an example:

html

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

In this example:

- `<a>` is the anchor tag used to create a hyperlink.

- `href="https://www.example.com"` is the attribute specifying the URL where the link will take the user when clicked.

- The text "Click here to visit Example" between the opening and closing `<a>` tags is the clickable text displayed to the user.

**Q.13 What is the use of an iframe tag? With Example.**

**Answer:**

The `<iframe>` tag in HTML is used to embed another HTML document within the current document. It allows you to display content from another source or webpage within a specified area on your webpage.

Here's an example:

html

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

Explanation of the attributes used in the example:

- `src="https://www.example.com"`: Specifies the URL of the page or content to be displayed within the `<iframe>`.

- `width="600"` and `height="400"`: Define the width and height of the `<iframe>` element, specifying its dimensions.

- `title="Example"`: Provides a title for the `<iframe>` for accessibility and SEO purposes.

This code will embed the content of the "https://www.example.com" webpage within the `<iframe>` element on your webpage.

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

**Answer:**

The `<span>` tag in HTML is an inline container used to apply styles or manipulate a specific section of text within a larger block of content. It doesn't add any line breaks by default and is

often used with CSS to style or manipulate the enclosed content.

Here's an example:

html

This is a blue and bold text.

In this example:

- `<span>` is used to enclose the words "blue and bold" within the paragraph.

- `style="color: blue; font-weight: bold;"` is an inline CSS style applied to the `<span>` element, changing the color to blue and setting the text to bold.

The `<span>` tag is handy when you want to apply specific styles or perform JavaScript manipulations on a specific part of the text within a larger block of content.

**Q.15 How to insert a picture into a background image of a web page? With Example.**

**Answer:**

To set an image as the background of a web page, you can use CSS. Here's an example:

HTML:

```
html
<!DOCTYPE html>
<html>
<head>
  <title>Background Image Example</title>
  <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
  <div class="content">
    <!-- Your page content goes here -->
  </div>
</body>
</html>
```

CSS (styles.css):

```
css
body {
  background-image: url('path/to/your/image.jpg');
```

```
  background-size: cover; /* Adjust the size of the background */
  background-position: center; /* Position the background */
  /* Additional background properties can be added as needed */
}

.content {
  /* Styles for your content */
  /* This div will hold your page content */
}
```

Replace `'path/to/your/image.jpg'` with the actual path to your image file.

In this example, the image specified in the CSS for the `body` element will be set as the background for the entire webpage. Adjust the `background-size`, `background-position`, and other properties as needed to fit your design preferences. Your actual content will be placed within the `<div class="content">` element.

**Q.16 How are active links different from normal links?**

**Answer:**

Active links and normal links differ based on their states and appearance:

- *Normal Links*: These are the default links that are styled as defined in CSS or browser defaults. They are displayed according to their default appearance until they are interacted with.

- *Active Links*: An active link represents a link that is currently being interacted with or selected. For example, when a user clicks on a link, it becomes active until the action is completed or until the user navigates away from the page. This state might be styled differently to provide visual feedback to the user about the action they are performing.

Styling for active links often includes changing colors, underlines, or backgrounds to give users a visual indication that they have selected or clicked on that particular link. Once the action associated with the link is completed (like navigating to a new page), it reverts to its default or visited state depending on the CSS styling applied.

In CSS, you can use the `:active` pseudo-class to style active links and provide visual feedback to users when they click on them.

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

**Answer:**

In HTML, there are several tags used to separate or structure sections of text:

1. `<p>`: Defines a paragraph, separating blocks of text.

2. `<h1>` to `<h6>`: Headings of different levels, organizing text into hierarchical sections, with `<h1>` being the most important and `<h6>` the least important.

3. `<div>`: A generic block-level container used to separate or group sections of content.

4. `<span>`: An inline container used to apply styles or manipulate a specific portion of text within a larger block of content.

5. `<blockquote>`: Denotes a block of quoted text from another source.

6. `<pre>`: Defines preformatted text, maintaining spaces and line breaks as they appear in the HTML code.

Each of these tags serves different purposes in organizing or presenting sections of text within an HTML document, providing structure and aiding in styling or content manipulation.

**Q.18 What is SVG?**

**Answer:**

SVG stands for Scalable Vector Graphics. It's an XML-based vector image format used for describing two-dimensional vector graphics. SVG images are resolution-independent, meaning they can be scaled to any size without losing quality, as they are composed of mathematical descriptions of shapes rather than pixels.

SVG is widely used for various purposes, including icons, logos, illustrations, charts, and interactive graphics on the web. It supports a range of features like shapes, paths, text, gradients, transparency, and animation, making it a versatile format for creating graphical content that can adapt to different screen sizes and resolutions.

**Q.19 What is difference between HTML and XHTML?**

**Amswer:**

HTML (Hypertext Markup Language) and XHTML (Extensible Hypertext Markup Language) are both markup languages used to structure and present content on the web, but they have some differences:

1. *Syntax*:

- HTML: It has a more forgiving syntax and allows certain errors or omissions without causing issues.

- XHTML: It follows a stricter syntax similar to XML, where elements must be properly nested and closed.

2. *Parsing*:

- HTML: Browsers are more lenient with parsing HTML, allowing for some errors to be corrected or overlooked.

- XHTML: Errors in the document structure can cause parsing to fail or render the document improperly.

3. *Document Structure*:

- HTML: Case sensitivity for tags and attributes is not enforced.

- XHTML: Tags and attributes must be written in lowercase, and attributes must have values enclosed in quotes.

4. *Compatibility*:

- HTML: Older browsers may handle HTML better due to its more forgiving nature.

- XHTML: It may require a stricter adherence to standards and might face compatibility issues with older browsers.

Overall, XHTML was developed as a reformulation of HTML using XML standards for a cleaner, more structured approach. However, HTML5, the latest version of HTML, has incorporated many features from XHTML, making the distinctions between the two less significant in modern web development.

**Q.20 What are logical and physical tags in HTML?**

**Answer:**

In HTML, logical tags and physical tags refer to the traditional distinction between the structure of a document and the presentation of that structure.

- *Physical Tags*: These are HTML tags that define the visual appearance or formatting of the content. Examples include `<b>` for bold, `<i>` for italic, `<font>` for changing text color or font

size. These tags directly impact how the content looks on the page. They were commonly used in older versions of HTML but are deprecated in favor of CSS for styling purposes.

- *Logical Tags*: These are tags that define the structure or meaning of the content without specifying its visual appearance. Examples include `<div>` for dividing content into sections, `<p>` for paragraphs, `<header>`, `<footer>`, `<nav>`, etc. These tags provide structure to the content, making it more accessible and meaningful for search engines and assistive technologies. They separate the structure or semantics of the content from its presentation, promoting better web standards and accessibility.

With the evolution of HTML and the introduction of CSS for styling, the focus has shifted towards using logical tags to define content structure and relying on CSS for visual presentation, moving away from using physical tags for styling purposes.