Question: Define HTML. What is the purpose of HTML in web development?

Answer:

HTML (Hyper Text Markup Language**)** is the standard markup language used to structure and design web pages. It defines how text, images, and multimedia content are displayed in a web browser.

The role of HTML in web development is to provide the backbone and structure of a web page. It defines the layout and organization of content on a web page by using a ranked structure of elements. HTML tags allow developers to specify the type of content being displayed and how it should be presented.

web developers can define the headings, paragraphs, lists, images, links, and other elements that make up the content of a web page. They can also create forms for user input, embed multimedia content like videos and audio, and add semantic elements for accessibility and search engine optimization.

Question: Explain the basic structure of an HTML document. Identify the mandatory tags and their purposes.

Answer:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <title>Structure of HTML Document</title>
    </head>
    <body>
        <h1>Hello World</h1>
        <p>how are you ?</p>
    </body>
</html>
```

1. <!DOCTYPE HTML>: The <!DOCTYPE html> declaration is placed at the beginning of the document. It tells the browser that the document follows HTML5 standards, ensuring consistent rendering across browsers.
2. <html> Tag: The <html> tag wraps the entire document, serving as the root element of an HTML page. It typically includes the lang attribute to specify the language of the content
3. <head> Section: The <head> section contains metadata, scripts, styles, and other information not displayed directly on the page but essential for functionality and SEO.
4. <body> Section: The <body> section contains all the visible content of the web page, including text, images, videos, links, and more. This is where you'll add the main elements to display on the page.

Question: What is the difference between block-level elements and inline elements in HTML? Provide examples of each.

Answer:

A block-level element always starts on a new line, and the browsers automatically add some space before and after the element.

An inline element does not start on a new line. It takes up as much width as necessary.

Block level elements:

<address>

<dt>

<hr>

<section>

<header>

<form>

Inline elements:

<textarea>

<a>

<b>

<i>

<q>

Question: Discuss the role of semantic HTML. Why is it important for accessibility and SEO? Provide examples of semantic elements.

Answer:

Its providing meaningful structure to a webpage by using tags that clearly describe the content within them, enhancing accessibility for users.

Search engines use semantic HTML to better understand the content of a webpage. By using appropriate tags, developers can help search engines categorize and index content more effectively. For example, using <h1> for the main title and <h2> for subheadings provides a clear hierarchy that search engines can interpret.

Example of semantic HTML:

- <article>
- <aside>
- <details>
- <figure>
- <footer>
- <header>
- <main>
- <mark>
- <nav>
- <section>
- <summary>
- <time>

Question: What are HTML forms used for? Describe the purpose of the input, textarea,

select, and button elements.

Answer:

HTML forms are used to collect information from users on websites and apps.

<input>:

The <input> element is used to create various types of interactive controls in a web form. It can accept user input in different formats

<textarea>:

The <textarea> element is used to create a multi-line text input field. It allows users to enter larger amounts of text, such as comments, messages, or descriptions.

<select>:

The <select> element creates a dropdown list from which users can select one or more options.

<button>:

The <button> element is used to create a clickable button that can trigger actions when clicked.

Question: Explain the difference between the GET and POST methods in form submission. When should each be used?

Answer:

GET is used for retrieving data like searching, filtering, or paging, whereas POST is used for submitting forms, modifying data, or creating new resources.

- Use GET when:
    - You are retrieving data without causing side effects.
    - The data is not sensitive and can be included in the URL.
    - You want the request to be bookmarkable or shareable.
- Use POST when:
    - You are submitting data that modifies server state or requires processing.
    - The data is sensitive and should not be exposed in the URL.
    - You need to send large amounts of data or complex data structures.

Question: What is the purpose of the label element in a form, and how does it improve accessibility?

Answer:

A <label> is used to create a caption for a form control. The <label> can be associated with a form control either implicitly by placing the control element inside the label element, or explicitly by using the for attribute.

Question: Explain the structure of an HTML table and the purpose of each of the following elements: <table>, <tr>, <th>, <td>, and <thead>.

Answer:

```
<table>
    <tr>
      <th>Firstname</th>
      <th>Lastname</th>
      <th>Age</th>
    </tr>
    <tr>
      <td>Priyank</td>
      <td>Patel</td>
      <td>24</td>
    </tr>
    <tr>
      <td>Arjun</td>
      <td>shah</td>
      <td>32</td>
    </tr>
    <tr>
      <td>Samar</td>
      <td>khan</td>
      <td>41</td>
    </tr>
  </table>
```

<table>:

Defines the structure for organizing data in rows and columns within a web page.

<tr>:

Represents a row within an HTML table, containing individual cells.

<th>:

Shows a table header cell that typically holds titles or headings.

<td>:

Represents a standard data cell, holding content or data.

<thead>:

Defines the header section of a table, often containing column labels.

Question: What is the difference between colspan and rowspan in tables? Provide examples.

Answer:

Colspan allows you to merge or combine adjacent table cells horizontally, creating a single, wider cell that spans across multiple columns.

```
<table>
    <thead>
      <tr>
        <th colspan="2">Name</th>
        <th>Class</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Mahesh</td>
        <td>patel</td>
        <td>1</td>
      </tr>
      <tr>
        <td>Sahil</td>
        <td>khan</td>
        <td>3</td>
      </tr>
      <tr>
        <td>Shivam</td>
        <td>jain</td>
        <td>5</td>
      </tr>
    </tbody>
  </table>
```

rowspan determines how many rows a specific cell in a table should cover. When a cell spans multiple rows, it occupies the space of those rows within the table.

```
<table>
    <tr>
      <th>Name</th>
      <th>Class</th>
      <th rowspan="3">PVM School</th>
    </tr>
    <tr>
      <td>Rajan</td>
      <td>10</td>
    </tr>
    <tr>
      <td>Anvar</td>
      <td>11</td>
    </tr>
  </table>
```

Question: Why should tables be used sparingly for layout purposes? What is a better alternative?

Answer:

Tables are designed to present tabular data, not for layout. Using tables for layout can confuse the semantic meaning of the HTML, making it harder for search engines and assistive technologies (like screen readers) to interpret the content correctly.

CSS Grid is a two-dimensional layout system that allows for more complex layouts. It enables designers to create grid-based layouts with rows and columns, making it easier to design responsive interfaces.

Flexbox is a layout model that allows for responsive and flexible layouts. It provides a more efficient way to arrange items in a one dimensional space.Flexbox is particularly useful for aligning items and distributing space within a container.