

HTML Basics

Define HTML. What is the purpose of HTML in Web development?

HTML stands for Hyper Text Markup Language.

HTML provide the structure and foundation for creating web pages and applications. HTML is the backbone of website.

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

The basic structure of an HTML document follows a specific pattern, containing essential elements that define the structure of the page and its content.

`<!DOCTYPE html>`: Declares the document type (HTML5).

`<html>`: The root element that encapsulates the entire document.

`<head>`: Contains metadata and other information about the page.

`<title>`: Provides a title for the document (appears in the browser tab).

`<body>`: Contains the actual content that is visible on the webpage.

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

Feature	Block Level	In Line
Layout	Takes Full Width, Start with new line	Takes only as much width as content
Content Flow	Create a new block of content	Stay with the flow of document
Contain	Other blockline element and inline element	Only inline element
Example	<code><div></code> , <code><p></code> , <code><h1></code> , <code><section></code>	<code></code> , <code><a></code> , <code></code>

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

Semantic HTML role is to practice of using HTML elements that clearly describe their meaning in a way that both humans and machines (like web browsers and search engines) can understand.

Accessibility refers to making websites usable for people with disabilities, including visual, auditory, motor, and cognitive impairments.

SEO refers to the practice of optimizing a website to improve its visibility and ranking on search engines like Google, Bing, etc.

EX:

The <header> element defines introductory content or navigational links. It typically contains a logo, site title, or a navigation menu.

The <footer> element represents the footer section of a page, typically containing copyright information, contact details, or links to privacy policies or terms of service.

HTML Forms

What are HTML forms used for? Describe the purpose of the input, text area, select, and button elements.

HTML forms are used to collect and submit user input on websites.

Input

It allows users to input data, such as text, numbers, dates, passwords, and more.

It can be used for single-line text input, checkboxes, radio buttons, or buttons.

Text Area

To allow users to input larger amounts of text, such as messages, comments, or reviews.

It's commonly used for things like contact forms, feedback forms, and text areas for writing posts or emails.

Select

To provide a list of options for users to choose from. It's commonly used in forms for selecting categories, countries, or other predefined options.

You can use the **<option>** element inside a **<select>** to define the individual choices.

Button

To provide a button for submitting the form, resetting the form, or performing custom actions using JavaScript.

It can also be used in conjunction with JavaScript to trigger events on the page, such as opening a modal or performing AJAX requests.

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

Features	GET	POST
Data Location	Appended to the URL as query parameters.	Sent in the body of the HTTP request.
Data Visibility	Visible in the browser's address bar.	Hidden from the user (not in the URL).
Data Size	Limited by URL length (usually ~2,000 chars).	No practical size limitation.
Caching	Can be cached by the browser.	Cannot be cached by default.
Use Cases	Retrieving data (e.g., search forms).	sending sensitive data, creating/updating resources.
Security	Not suitable for sensitive data (e.g., passwords).	Suitable for sensitive data (e.g., login credentials).
Idempotency	Idempotent (repeated GET requests have the same result).	Non-idempotent (repeated POST requests can cause changes).

When Use GET

You are sending non-sensitive data.

The form is used to retrieve data (e.g., search queries, filtering, navigation).

You want the ability to bookmark or share the result.

The data is small and easily encoded in the URL.

When Use POST

The form contains sensitive data (e.g., passwords, credit card details).

You are submitting large amounts of data or files (e.g., form submissions with images or lengthy text).

You are creating or updating resources on the server (e.g., submitting a new user registration, processing payments).

You want to avoid URL length limitations.

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

The <label> element is a simple yet powerful tool for improving the accessibility, usability, and functionality of forms. By clearly associating labels with form inputs:

- **Screen reader users** get valuable context about what each form field represents.
- **Users with motor impairments** can more easily interact with form fields, since clicking the label focuses the corresponding input.
- **Keyboard and focus navigation** is improved, allowing better form navigation for users who rely on keyboards or other assistive technologies.
- The overall **clarity** of the form is enhanced, making it easier for everyone to understand and fill out the form accurately.

In short, using <label> is a crucial practice for making web forms **more accessible** and **user-friendly**, benefiting all users, especially those with disabilities.

HTML Tables

Explain the structure of an HTML table and the purpose of each of the following elements:

<table>, **<tr>**, **<th>**, **<td>** and **<thead>**.

HTML table is used to display data in a grid format, with rows and columns.

Each row contains multiple cells, and each cell can contain text, images, or other types of content.

<table>: The container element for the table.

<tr>: Represents a table row.

<th>: Represents a table header cell, typically used for column or row headers.

<td>: Represents a table data cell, used to hold the actual content in the table.

<thead>: Represents the header section of the table, typically used to group header rows.

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

- **colspan** is used to make a cell span horizontally across multiple columns in a table.
- **rowspan** is used to make a cell span vertically across multiple rows.

Both attributes help create more complex and organized table structures, especially when you need to merge data or headers across rows and columns.

```
<table border="1">
```

```
<tr>
```

```
<th rowspan="2">Row 1, Column 1 (spans 2 rows)</th>
```

```
<th colspan="2">Header spanning two columns</th>
```

```
</tr>
```

```
<tr>
```

```
<td>Row 2, Column 2</td>
```

```
<td>Row 2, Column 3</td>
```

```
</tr>
```

```
<tr>
```

```
<td>Row 3, Column 1</td>
```

```
<td>Row 3, Column 2</td>
```

```
<td>Row 3, Column 3</td>
```

```
</tr>
```

</table>

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

Tables should be used sparingly for layout purposes because they are meant for presenting tabular data, not general page structure. Misusing tables can cause accessibility issues, as screen readers may not interpret content correctly, and it can hinder SEO. Additionally, tables are not responsive by default, making them difficult to adapt for different screen sizes. This can lead to complex and hard-to-maintain code.

A better alternative is to use CSS Grid or Flexbox for layout. These CSS techniques are more flexible, allowing for responsive, adaptive designs. They also provide cleaner, more semantic HTML, making the code easier to maintain and ensuring better accessibility and performance.