Contact: +91-9702328507
Email: mumbaicodingclub@gmail.com
www.mumbaicodingclub.com

HTML Block and Inline Elements

Introduction to HTML Elements

- **HTML Elements**: Define the structure and content of a webpage.
- **Types of Elements:**
  - **Block Elements**: Start on a new line and take up the full width available.
  - **Inline Elements**: Stay in line with the surrounding text and take up only as much width as necessary.
  - **Inline-Block Elements**: Behave like inline elements but can have width and height applied.

Block Elements in HTML

- **Definition:** Block elements start on a new line and occupy the full width of their container.
- **Common Block Elements:**
  - ** **
  - **<p>**
  - **<h1> - <h6>**
  - **<ul>, <ol>, **
  - **<section>, <header>, <footer>**

Inline Elements in HTML

- **Definition**: Inline elements do not start on a new line and only take up as much space as necessary.
- Common Inline Elements:
  -
  - <a>
  - ,
  - <img>
  - <input>

Inline-Block Elements in HTML

- **Definition:** Inline-block elements are displayed like inline elements but allow for width and height to be set, unlike regular inline elements.
- **Use Cases**: Ideal for creating elements that need to flow inline but also require size control, like buttons or navigation items.
- **Common Example:**
  - Custom-styled or with display: inline-block.

CSS Styling of Block Elements

- CSS Properties
  - Block elements can be styled with properties like width, height, margin, padding.
  - Example:
    - div { width: 100%; margin: 10px; padding: 20px; }
  - Container Role: Block elements can contain other block and inline elements.

CSS Styling of Inline Elements

- CSS Properties
  - Inline elements can be styled with properties like font-style, color, text-decoration, background.
  - Example
    - a { color: blue; text-decoration: none; }
  - Limitations: Width, height, margin, and padding (top and bottom) generally cannot be applied directly unless converted to block or inline-block.

CSS Styling of Inline-Block Elements

- CSS Properties
  - Inline-block elements can be styled with properties like width, height, margin, and padding.
  - Example:
    - .button { display: inline-block; width: 150px; height: 50px; margin: 10px; background-color: #007BFF; color: white; text-align: center; line-height: 50px; }
  - Advantages: Combines inline layout with block-level control.

Block vs. Inline vs. Inline-Block - Key Differences

- Display
  - Block: Full width, starts on a new line.
  - Inline: Takes up as much width as needed, remains in line with text.
  - Inline-Block: Behaves like an inline element but allows width, height, and other block-level styling.

What Are Semantic Elements

- Definition: HTML semantic elements are tags that clearly describe their meaning in a human- and machine-readable way.
- Purpose: Enhance readability and accessibility, improve SEO, and provide better structure for documents.

Importance of Semantic HTML

- Readability: Easier for developers to understand the structure and purpose of content.
- Accessibility: Screen readers and other assistive technologies can better interpret the content.
- SEO Benefits: Search engines can better index and rank content.

Common Semantic Elements

- Header (<header>): Defines a header for a document or section.
- Footer (<footer>): Defines a footer for a document or section.
- Article (<article>): Represents a self-contained piece of content that could be distributed independently.
- Section (<section>): Represents a section in a document, often with a heading.
- Nav (<nav>): Defines navigation links.
- Aside (<aside>): Represents content tangentially related to the content around it (e.g., a sidebar).
- Main (<main>): Represents the dominant content of the <body>.

Benefits of Using Semantic Elements

- SEO: Improves search engine ranking by making the content easier to understand.
- Maintenance: Easier to maintain and update code.
- Accessibility: Enhances the browsing experience for users with disabilities.

Browser Support and Compatibility

- Overview: All modern browsers support semantic elements, but older browsers may not fully recognize them. Polyfills and fallbacks can be used if needed.

Introduction to HTML Forms

- **Definition**: HTML forms are used to collect user inputs and submit them to a server.
- **Importance**: Key for user interaction, data collection, and communication with web applications.

Form Structure Overview

- Basic Tags:
  - <form>: Wrapper for form elements.
  - <input>: Various types for different user inputs.
  - <label>: Associates text with an input element.
  - <button>: Defines buttons for submission or other actions.

Basic Form Elements

- Text Input (<input type="text">): Single-line text fields.
- Password Input (<input type="password">): Conceals user input.
- Email Input (<input type="email">): Validates email format.
- Number Input (<input type="number">): Restricts input to numbers.
- Radio Buttons (<input type="radio">): Single choice from multiple options.
- Checkboxes (<input type="checkbox">): Multiple selections possible.
- Submit Button (<input type="submit"> or <button type="submit">): Submits the form.

Form Controls and Grouping

- Select and Option (<select>, <option>): Dropdown menus.
- Textarea (<textarea>): Multi-line text input.

Form Validation

- required: Ensures input is provided.
- minlength / maxlength: Specifies length limits.
- pattern: Enforces input patterns (e.g., regex).
- type: Specific types like email, number, etc.