Semantics

Block Vs Inline Elements

In HTML, elements are categorized as either block-level or inline elements. The distinction between block and inline elements has implications for their default behaviour in terms of layout and how they interact with other elements

Block Elements	Inline Elements
They use the entire webpage width, effectively blocking any other elements from being placed on the left or the right side.	They only take up as much width as necessary to show the element's contents and after that, other elements can be made to match the inline element.
Block elements always start on a new line.	Inline elements do not start from a new line.
Examples: , <p>, <h1>, <h6>, <nav>, etc.	Examples:<b>, , <img> etc

Div Tag

Definition: The (division) element is a fundamental container in HTML used to group and structure other HTML elements. It does not inherently provide any styling or semantic meaning but is a flexible tool for layout and organisation.

Example:

```
<div id="main-content">
  <p>This is the main content area.</p>
</div>
```

Useful Attributes:

- **id:** Assigns a unique identifier to the for styling or scripting purposes.
  Example:

```
<div id="main-content">...</div>
```

Use Case: Employed to create webpage sections, group related elements, or apply styling to a specific content block.

Span Tag

Definition: The element is an inline container that applies styles or scripting to a specific text portion within a block-level element. Unlike , it doesn't introduce a new line and is typically used for small, inline styling.
Example:

```
<p>This is <span style="color: red;">highlighted</span> text.</p>
```

Useful Attributes:

- **id:** Assigns a unique identifier to the for styling or scripting purposes.
  Example:

```
<p>This is <span id="highlighted-text">highlighted</span> text.</p>
```

Use Case: Useful for applying styles to a specific word or phrase within a larger text block.

Semantic vs Non-Semantic Elements

- Semantic elements clearly define the content they carry. Tags used to enclose them are called semantic tags. For example, a paragraph element, CONTENT tells that its content is a paragraph.

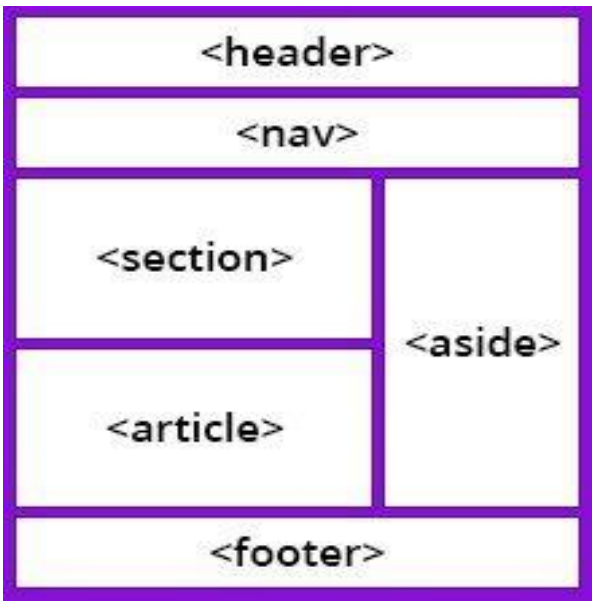
- Some of the semantic elements are:
  article, aside, details, figcaption, figure, footer, header, main, mark, nav, section, etc.

- Non-semantic elements don't tell any information about the content that they carry.

- Tags used to enclose these are called non-semantic tags. For example, a div element, CONTENT can carry any content (headings, paragraphs, links, etc) but the content as a whole has no defined semantics. Hence div is a non-semantic element.

  Note: Semantically correct HTML helps search engines, screen readers, and other user devices determine the significance and context of web content.

Semantic Tags: Building Blocks of HTML Structure



Header Tag

Definition: The <header> element represents a header section in HTML. It can contain headings (<h1> to <h6>), logos, navigation links, and other introductory content. It typically appears at the top of a page or a section.
Example:

```
<header>
 <h1>Your Website</h1>
 <nav>
  <ul>
   <li><a href="#">Home</a></li>
   <li><a href="#">About</a></li>
   <!-- More navigation items -->
  </ul>
 </nav>
</header>
```

Useful Attributes:

- **id:** Assigns a unique identifier to the <header> for linking or styling.
- **role:** Specifies the role of the <header> for accessibility.

Example:

```
<header id="site-header" role="banner">
 <h1>Your Website</h1>
</header>
```

Use Case: Defines a webpage's introductory or navigational part.

Footer Tag

Definition: The <footer> element represents a footer section in HTML. It contains information about the document, copyright information, author details, or links to related resources.
Example:

```
<footer>
  <p>&copy; 2024 Your Name. All rights reserved.</p>
</footer>
```

Useful Attributes:

- **id:** Assigns a unique identifier to the <footer> for linking or styling.
- **role:** Specifies the role of the <footer> for accessibility.

Example:

```
<footer id="site-footer" role="contentinfo">
  &copy; 2024 Your Name
</footer>
```

Use Case: Provides concluding information or additional links at the bottom of a webpage.

• Main Tag

Definition: The <main> element represents the main content of the document, excluding headers, footers, or sidebars. It helps in improving the document's accessibility and structure.

Example:

```
<main>
  <h2>Main Content</h2>
  <p>This is the main content of the webpage.</p>
</main>
```

Useful Attributes:

- **id:** Assigns a unique identifier to the <main> for linking or styling.

Example:

```
<main id="main-content">
  <h2>Main Content</h2>
  <p>This is the main content of the webpage.</p>
</main>
```

Use Case: Centralizes the primary content of a webpage, enhancing semantic HTML structure.

- ## Nav Tag

Definition: The <nav> element defines a set of navigation links. It's commonly employed to create menus, lists of links, or other navigation structures.
Example:

```
<nav>
 <ul>
  <li><a href="#">Home</a></li>
  <li><a href="#">Products</a></li>
  <!-- Additional navigation items -->
 </ul>
</nav>
```

Useful Attributes:

- **id:** Assigns a unique identifier to the <nav> for linking or styling.
  Example:

```
<nav id="main-navigation">
 <ul>
  <li><a href="#">Home</a></li>
  <!-- Additional navigation items -->
 </ul>
</nav>
```

Use Case: Group navigation links aid site navigation and accessibility.

- ## Section Tag

Definition: The <section> element represents a generic section of content within a document. It helps in organising content and is often used for thematic grouping.
Example:

```
<section>
 <h2>Section Title</h2>
 <p>This is the content of the section.</p>
</section>
```

Useful Attributes:

- **id:** Assigns a unique identifier to the <section> for linking or styling.
  Example:

```
<section id="intro-section">
 <h2>Introduction</h2>
 <p>Welcome to our website!</p>
</section>
```

Use Case: Encloses related content to enhance document structure and readability.

- ## Article Tag

Definition: The <article> element represents self-contained content that can be distributed and reused independently. It's suitable for articles, blog posts, comments, or content that can stand alone.
Example:

```
<article>
 <h3>Article Title</h3>
 <p>This is the content of the article.</p>
</article>
```

Useful Attributes:

- **id:** Assigns a unique identifier to the <article> for linking or styling.
  Example:

```
<article id="blog-post">
  <h3>Blog Post Title</h3>
  <p>Content of the blog post goes here.</p>
</article>
```

Use Case: Encapsulates content meant to be distributable and reusable, improving document structure.

Aside Tag

Definition: The <aside> element marks content tangentially related to the content around it. It's often used for sidebars, pull quotes, or information boxes.
Example:

```
<aside>
  <p>Related information goes here.</p>
</aside>
```

Useful Attributes:

- **id:** Assigns a unique identifier to the <aside> for linking or styling.
  Example:

```
<aside id="sidebar">
  <p>Related information goes here.</p>
</aside>
```

Use Case: Contains content that is not the main focus but complements the primary content.

- ## Details Tag

Definition: The <details> element is used to create a disclosure widget from which the user can obtain additional information. It's often used in conjunction with the <summary> element.
Example:

```
<details>
 <summary>Click to reveal more information</summary>
 <p>Additional details go here.</p>
</details>
```

Useful Attributes:

- **id:** Assigns a unique identifier to the <details> for linking or styling.
- **open:** Specifies that the <details> element should be open (visible) by default.
  Example:

```
<details id="additional-info" open>
 <summary>Click to reveal more information</summary>
 <p>Additional details go here.</p>
</details>
```

Use Case: Creates an expandable/collapsible section for additional details.

- ## Summary Tag

Definition: The <summary> element is used as the visible heading for a <details> element. It provides a label or title for the content hidden inside a disclosure widget.
Example:

```
<details>
 <summary>Click me</summary>
```

```
  <p>Content revealed when clicked.</p>
</details>
```

Useful Attributes:

- id: Assigns a unique identifier to the <summary> for linking or styling.
  Example:

```
<details>
  <summary id="reveal-info">Click me</summary>
  <p>Content revealed when clicked.</p>
</details>
```

Use Case: Provides a concise label for the content within a disclosure widget.

Additional Elements

Favicon

Definition: A favicon is a small icon displayed in the browser tab or next to the URL. It is linked using the <link> tag in the <head> section.
Example:

```
<link rel="icon" href="favicon.ico" type="image/x-icon">
```

Useful Attributes:

- **rel:** Specifies the relationship between the current document and the linked favicon.
- **href:** Specifies the location (URL) of the favicon file.
- **type:** Specifies the MIME type of the linked file.
  Example:

```
<link rel="icon" href="favicon.png" type="image/png">
```

Use Case: Enhances the website's branding and provides a recognisable icon in browser tabs.

- ## Meta Tag

Definition: The <meta> tag is used to provide metadata about the HTML document, such as character set, description, keywords, and viewport settings.
Example:

```
<meta charset="UTF-8">
<meta name="description" content="A brief description of the webpage.">
```

Useful Attributes:

- **charset:** Specifies the character encoding for the HTML document.
- **name:** Defines the name of a metadata property.
- **content:** Provides the value for the metadata property.

Example:

```
<meta charset="UTF-8">
<meta name="description" content="A brief description of the webpage.">
```

Use Case: Offers additional information about the document for browsers and search engines.

- ## Code Tag

Definition: The <code> tag is used to define a piece of code within the text. It's commonly used to represent code snippets, variable names, or inline code.
Example:

```
<p>This is a <code>console.log('Hello, World!');</code> within a paragraph.</p>
```

Use Case: Highlights and stylizes code within the text, improving readability.

- ## Figure Tag

Definition: The <figure> element is used to encapsulate media content, such as images or videos, along with their captions. It provides a semantic container for multimedia content and is particularly useful when you want to associate a caption with an image or a video.
Example:

```
<figure>
  <img src="image.jpg" alt="Description">
  <figcaption>Caption goes here</figcaption>
</figure>
```

Useful Attributes:

- **id:** Assigns a unique identifier to the <figure> for linking or styling.

  Example:

```
<figure id="image-container">
  <img src="image.jpg" alt="Description">
  <figcaption>Caption goes here</figcaption>
</figure>
```

Use Case: Group and semantically associate an image or video with its corresponding caption.

- ## Figcaption Tag

Definition: The <figcaption> element is used to provide a caption or description for the content inside a <figure> element. It allows you to add context or additional information related to the media content within the <figure>.

Example:

```
<figure>
 <img src="image.jpg" alt="Description">
 <figcaption>This is an informative caption.</figcaption>
</figure>
```

Useful Attributes:

- **id:** Assigns a unique identifier to the <figcaption> for linking or styling.

  Example:

```
<figure>
        <figcaption id="image-caption">This is an informative caption.</figcaption>
</figure>
```

Use Case: Enhances accessibility and provides context by adding captions to images or videos within a <figure> element.

Conclusion:

The fundamental HTML elements and tags are crucial for building well-structured and semantically meaningful web pages. This comprehensive guide overviews various HTML elements commonly used in web development, their attributes and practical examples.

Starting with the building blocks of structure, we explored tags like and , which offer flexibility in organising and styling content. The <header> and <footer> tags define the header and footer sections, contributing to a page's structure and accessibility. The <main> tag centralises the primary content, enhancing document structure.

Navigation is streamlined with the <nav> tag, while thematic grouping is achieved through the <section> tag. The <article> tag encapsulates independent and distributable content, while <aside> complements the main content with related information. The <details> and <summary> tags provide a convenient way to create expandable and collapsible sections.

Additionally, we explored elements like <figure> and <figcaption>, designed for media content and associated captions, contributing to a more accessible and visually appealing presentation.

Including attributes such as id, role, open, and specific examples allows for customisation, styling, scripting, and improved accessibility. The <code> tag, illustrated with a JavaScript snippet, showcases how to highlight and stylise code within text for better readability.

Whether you are a beginner or looking to reinforce your HTML knowledge, this guide is a valuable resource for mastering essential HTML elements and tags, facilitating the creation of well-structured and semantically rich web content. As you delve deeper into web development, the principles outlined here will form the foundation for building dynamic and visually appealing websites.

References

1. https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_flow_layout/Block_and_inline_layout_in_normal_flow
2. https://developer.mozilla.org/en-US/docs/Web/HTML/Element/div
3. https://developer.mozilla.org/en-US/docs/Web/HTML/Element/span
4. https://developer.mozilla.org/en-US/docs/Glossary/Semantics#semantics_in_html
5. https://developer.mozilla.org/en-US/docs/Web/HTML/Element/meta
6. https://developer.mozilla.org/en-US/docs/Web/HTML/Element/code