

Semantic HTML

Web Development Essentials - Session 4

Session Overview

Learning Goals for Today:

- Understand what semantic HTML is
- Learn common semantic elements and their purpose
- Explore the benefits of using semantic HTML for accessibility, SEO, and maintainability
- Build a webpage using semantic HTML

What is Semantic HTML?

Definition:

- Semantic HTML uses meaningful tags that describe the content they enclose.
- It helps both developers and browsers understand the structure and meaning of a webpage.

Example:

- Non-semantic: `<div>`, ``
- Semantic: `<header>`, `<article>`, `<footer>`

Why Use Semantic HTML?

Key Benefits:

- **Accessibility:** Screen readers and other assistive technologies can better understand the content.
- **SEO (Search Engine Optimization):** Search engines prioritize content that's well-structured using semantic tags.
- **Maintainability:** Code is more readable and easier to maintain for developers.
- **Future-Proofing:** Browsers prioritize semantic elements for rendering.

Common Semantic HTML Elements

Structural Elements:

- `<header>`: Defines a header for a document or section.
- `<nav>`: Represents a navigation section.
- `<main>`: Represents the main content of the document.
- `<footer>`: Defines a footer for a document or section.
- `<section>`: Groups related content.
- `<article>`: Represents an independent, self-contained piece of content.
- `<aside>`: Contains content related to the main content but not central to it (e.g., sidebars).

Non-Semantic vs. Semantic HTML

Non-Semantic Example:

```
<div class="header">Header content</div>
<div class="main">Main content</div>
<div class="footer">Footer content</div>
```

Semantic Example:

```
<header>Header content</header>
<main>Main content</main>
<footer>Footer content</footer>
```

- The second example is clearer and more meaningful for both developers and browsers.

Using `<header>` and `<footer>`

Header:

- Typically contains navigation, logos, or introductory content.

Footer:

- Used for contact information, copyright, or external links.

```
<header>
  <h1>Website Title</h1>
  <nav>
    <ul>
      <li><a href="#home">Home</a></li>
      <li><a href="#about">About</a></li>
    </ul>
  </nav>
</header>
```

```
<footer>
  <p>© 2024 My Website</p>
  <a href="mailto:info@example.com">Contact Us</a>
</footer>
```

<main> and <section>

Main:

- Holds the central content of the page.
- Only one `<main>` element per page is allowed.

```
<main>
  <h2>Main Article Heading</h2>
  <p>This is the main content of the page.</p>
</main>
```

Section:

- Divides content into thematic sections.

```
<section>
  <h3>About Us</h3>
  <p>We are a web development company.</p>
</section>
```


<article> and <aside>

Article:

- For self-contained content like blog posts, news articles, etc.

```
<article>
  <h3>Blog Post Title</h3>
  <p>This is a blog post on web development.</p>
</article>
```

Aside:

- For side content, such as ads, related links, or extra information.

```
<aside>
  <h4>Related Links</h4>
  <ul>
    <li><a href="#">Related Link 1</a></li>
    <li><a href="#">Related Link 2</a></li>
  </ul>
</aside>
```

Practical Example: Page Layout with Semantic HTML

```
<header>
  <h1>My Website</h1>
  <nav>
    <ul>
      <li><a href="#home">Home</a></li>
      <li><a href="#about">About</a></li>
    </ul>
  </nav>
</header>

<main>
  <article>
    <h2>Main Article</h2>
    <p>This is the main content of the page.</p>
  </article>
```

```
  <aside>
    <h3>Related Links</h3>
    <ul>
      <li><a href="#">Link 1</a></li>
      <li><a href="#">Link 2</a></li>
    </ul>
  </aside>
</main>

<footer>
  <p>© 2024 My Website</p>
  <a href="mailto:info@example.com">Contact Us</a>
</footer>
```

Hands-On Activity

Goal: Build a simple webpage using semantic HTML.

- Header with navigation links
- Main content section with an article
- Sidebar with related links using `<aside>`
- Footer with contact information

Instructions:

- Use your code editor (VS Code)
- Structure the webpage with semantic elements
- Test your page in the browser

Best Practices for Semantic HTML

Use Semantic Elements When Appropriate:

- Avoid overusing `<div>` and `` when there are better semantic alternatives.

Accessibility:

- Semantic HTML enhances the experience for screen readers and other assistive devices.

SEO:

- Search engines prefer structured, well-annotated content.

Maintainability:

- Well-organized semantic HTML makes collaboration easier for teams.

Summary

- **What We Learned Today:**
 - The importance of semantic HTML
 - Key semantic elements (`<header>`, `<footer>`, `<main>`, `<article>`, etc.)
 - How to build a webpage with semantic structure

Questions?

Q&A Session

- Any questions before we wrap up?

Thank You & See You in the Next Class!