

INTERNATIONAL ADVANCED LEVEL

INFORMATION TECHNOLOGY

UNIT 2

Unit 2, Topic 11 – The Semantic Web



Academic Year: 2024-2025

✍ Prepared by: Irikefe Hope Oniovosa

1. INTRODUCTION TO SEMANTIC HTML

Overview

- **Semantic HTML** uses elements whose names describe their meaning and role, rather than how they look.
- **Why It Matters:**
 - **Readability & Maintainability:** Code is easier for developers to understand and update.
 - **SEO:** Search engines use meaningful tags to index your content more accurately.
 - **Accessibility:** Assistive technologies (screen readers, voice browsers) rely on semantic structure to navigate pages.

Demonstration

<!-- Non-semantic -->

<div class="header">

 <div class="nav">...</div>

 <div class="content">...</div>

 <div class="footer">...</div>

</div>

<!-- Semantic -->

<header>...</header>

<nav>...</nav>

<main>...</main>

<footer>...</footer>

- Compare: the second snippet immediately communicates each section's purpose.

2. STRUCTURAL SEMANTIC ELEMENTS (20 MINUTES)

Key Elements & Roles

- `<header>`: Introductory content or navigational aids for its nearest ancestor (page or section).
- `<nav>`: A section of links for primary site navigation.
- `<main>`: The page's main content, unique per page.
- `<article>`: A self-contained composition (e.g., blog post, news story).
- `<section>`: Thematic grouping of content, usually with a heading.
- `<aside>`: Tangential content (e.g., pull-quotes, sidebars).
- `<footer>`: Information about its nearest ancestor (page or section) such as authorship or related links.

Usage Patterns

`<article>`

`<header><h1>Article Title</h1></header>`

`<section>...</section>`

`<aside>Related resources</aside>`

`<footer>Published Jan 1, 2025</footer>`

`</article>`

Hands-On Exercise

- Provide a simple wireframe (header, hero, two content blocks, sidebar, footer).
- Students map `<header>`, `<nav>`, `<main>`, two `<section>`, `<aside>`, and `<footer>` around the given content.

3. NON-SEMANTIC GROUPING

Elements

- `<div>`: Generic block-level container.
- `<span>`: Generic inline container.

Discussion

- Use these only when no semantic tag applies (e.g., grouping for styling).
- Overusing `<div>/<span>` leads to “div soup” that obscures meaning.

4. SEO AND ASSISTIVE TECHNOLOGIES

SEO Benefits

- Search engines parse `<article>`, `<h1>–<h6>`, `<nav>`, etc., to prioritize content and understand page structure.

Accessibility Benefits

- Screen readers announce landmarks (`<nav>`, `<main>`) and allow quick navigation via keyboard shortcuts (D for main, N for nav).
- Semantic headings provide a table-of-contents style navigation.

5. SEMANTIC INLINE ELEMENTS

Elements & Uses

- `<strong>`: Importance (renders as bold).
- `<em>`: Emphasis (renders as italics).
- `<blockquote>`: Block-level quotations.
- `<abbr>`: Abbreviations, with a title attribute for expansion.

- `<cite>`: Titles of works or references.
- `<address>`: Contact information.
- `<mark>`: Highlighted text.

Demonstration

`<p>This is a <strong>critical</strong> update.</p>`

`<p><em>Note:</em> Changes take effect immediately.</p>`

`<blockquote cite="https://example.com">Quote text...</blockquote>`

`<p>The CEO's initials are <abbr title="Chief Executive Officer">CEO</abbr>.</p>`

`<address>Acme Corp.<br>123 Main St.</address>`

Interactive Practice

- Provide a plain paragraph. Students wrap parts in `<strong>`, `<em>`, `<mark>`, `<abbr>` as appropriate.

6. SELF-CONTAINED CONTENT

Elements

- `<figure>`: Container for media and its caption.
- `<figcaption>`: Caption for the media.

Exercise

`<figure>`

``

`<figcaption>Figure 1: High-level system architecture.</figcaption>`

`</figure>`

- Students embed an image or code snippet and wrap it in `<figure>/<figcaption>`.

7. WAI-ARIA FOR ENHANCED SEMANTICS

Overview of ARIA

- **Roles:** Describe widget types (e.g., role="button", role="dialog").
- **States/Properties:** Communicate dynamic information (aria-expanded="false", aria-live="polite").

Demonstration

Convert a <div> dropdown into an accessible widget:

```
<div class="dropdown" role="button" tabindex="0"
  aria-haspopup="true" aria-expanded="false">
  Menu ▼
  <ul role="menu" hidden>
    <li role="menuitem">Option 1</li>
    <li role="menuitem">Option 2</li>
  </ul>
</div>
<script>
  const dd = document.querySelector('.dropdown');
  const menu = dd.querySelector('ul');
  dd.addEventListener('click', ()=>{
    const open = dd.getAttribute('aria-expanded') === 'true';
    dd.setAttribute('aria-expanded', !open);
    menu.hidden = open;
  });
</script>
```

8. SUMMARY AND REVIEW

Recap

- Semantic HTML gives meaning to your markup, boosting **readability**, **SEO**, and **accessibility**.
- Structural elements (<nav>, <article>, etc.) outline the page; inline semantics (,) clarify content.
- When semantics fall short, use ARIA roles/properties to fill in.

Quick Q&A

- **“When should you use <section> vs. <div>?”**
Use <section> when content is thematically grouped and usually has a heading; otherwise, use <div>.
- **“How does <nav> help assistive technologies?”**
It marks primary navigation, allowing screen-reader users to jump directly to the menu.

ASSESSMENT

Formative

- **Short Quiz:**
A quick multiple-choice and true/false quiz where students identify the correct semantic element for a given piece of content (e.g., <nav> vs. <div>, <section> vs. <article>).
- **Peer Review:**
In pairs, students exchange a simple non-semantic HTML page. Each reviews the other's code and proposes how to replace generic <div>/ tags with appropriate semantic elements (and where to add ARIA roles).

Summative

- **Project:**
Provide a multi-section web-page mock-up (wireframe or HTML prototype). Students must mark it up using:

- Structural semantic elements
(<header>, <nav>, <main>, <article>, <aside>, <footer>, etc.)
- Inline semantics (, , <abbr>, <mark>, etc.)
- WAI-ARIA enhancements for any custom widgets (e.g., role attributes, aria-* states)
- **Written Reflection:**
Students to submit a 300–500 word reflection explaining:
 - How their use of semantic HTML and ARIA roles improves **accessibility** (for screen readers, keyboard navigation)
 - How semantics can boost **SEO** by providing clearer content structure to search engines

ADDITIONAL RESOURCES

Videos

- **Semantic HTML Tutorial**
 [Learn Semantic HTML YouTube](#)
- **Introduction to WAI-ARIA**
 [Introduction to WAI-ARIA \(Accessible Rich Internet Applications\)](#)

Websites

- **MDN Web Docs – Semantic HTML**
 <https://developer.mozilla.org/en-US/docs/Glossary/Semantics>
- **MDN Web Docs – WAI-ARIA**
 <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA>
- **WebAIM – Accessibility Guidelines**
 <https://webaim.org>

Tools

- **Lighthouse (Chrome DevTools):** Built-in audit for performance, accessibility, and best practices.
- **aXe by Deque:** Free browser extension that highlights accessibility issues.
- **SEO Analysis Tools:** Browser extensions like MozBar or online tools (e.g., Google Search Console) to review semantic markup impact on search rankings.