

BONNIE COMPUTER HUB

Empowering Through Technology

FRONTEND WEB DEVELOPMENT – BEGINNER TRACK

WEEK 1/8: HTML & WEB FOUNDATIONS

SESSIONS

SESSION 1: HTML STRUCTURE, TAGS, FORMS & INPUTS

SESSION 2: MEDIA, SEMANTICS & ACCESSIBILITY

SESSION 3: DEVELOPER TOOLS, PAGE STRUCTURE

PREPARED BY: *Bonnie Computer Hub Team*

www.bonniecomputerhub.com

info@bonniecomputerhub.com

+254 729 820 689

SESSION 1: HTML STRUCTURE, TAGS, FORMS & INPUTS

Introduction

Objective: Learn the fundamentals of HTML, including document structure, common tags, and how to create forms for user input.

Relevance: HTML (HyperText Markup Language) is the backbone of every webpage. Understanding HTML structure and tags is essential as it provides the foundation upon which all web content is built. As frontend developers, this knowledge enables you to create the structure that CSS will style and JavaScript will make interactive.

Lesson Overview

Topics Covered:

- What is HTML and how does it work?
- Basic HTML document structure
- Common HTML tags and their purposes
- Creating forms and form elements
- Input types and their attributes

Core Content

What is HTML?

HTML stands for HyperText Markup Language. It's not a programming language but a markup language that tells web browsers how to structure the content on a webpage.

- **HyperText** refers to text that contains links to other texts
- **Markup** means using tags to define elements within the document
- HTML documents are plain text files with .html extension

Basic HTML Document Structure

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>My First Webpage</title>
</head>
<body>
<h1>Hello, World!</h1>
<p>This is my first webpage.</p>
</body>
</html>
```

Key components explained:

- `<!DOCTYPE html>`: Declares the document type and version of HTML
- `<html>`: The root element of an HTML page
- `<head>`: Contains meta-information about the document
- `<meta>`: Provides metadata like character encoding and viewport settings
- `<title>`: Sets the title displayed in the browser tab
- `<body>`: Contains all the content visible on the webpage

Common HTML Tags

Text-Related Tags

Headings: `<h1>` to `<h6>` (from most to least important)

- `<h1>Main Heading</h1>`
- `<h2>Subheading</h2>`
- `<h3>Section Heading</h3>`

Paragraphs: <p> for regular text content

- <p>This is a paragraph of text.</p>

Text Formatting:

- Bold text
- Italic text
- <u>Underlined text</u>
- <mark>Highlighted text</mark>

List Tags

Ordered Lists: Numbered lists using and

```
<ol>  
<li>First item</li>  
<li>Second item</li>  
<li>Third item</li>  
</ol>
```

Unordered Lists: Bullet point lists using and

```
<ul>  
<li>Apple</li>  
<li>Banana</li>  
<li>Orange</li>  
</ul>
```

Links and Images

Links: <a> tags with the href attribute

```
<a href="https://www.bonniecomputerhub.com">Visit Bonnie Computer  
Hub</a>
```

Images: tag with src and alt attributes

```

```

Structural Tags

Divs: <div> acts as a container for other elements

```
<div>
```

```
<h2>Section Title</h2>
```

```
<p>Content inside the div container.</p>
```

```
</div>
```

Span: for inline elements

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

HTML Forms and Inputs

Forms are used to collect user input. They're created using the <form> element and contain various input elements.

Basic Form Structure

```
<form action="/submit-form" method="post">
```

```
<!-- Form elements go here -->
```

```
<input type="submit" value="Submit">
```

```
</form>
```

- action: Specifies where to send the form data when submitted
- method: Defines the HTTP method (GET or POST)

Common Input Types

1. Text Input

```
<label for="username">Username:</label>
```

```
<input type="text" id="username" name="username" placeholder="Enter your username">
```

2. Password Input

```
<label for="password">Password:</label>
```

```
<input type="password" id="password" name="password">
```

3. Email Input

```
<label for="email">Email:</label>  
<input type="email" id="email" name="email"  
placeholder="example@email.com">
```

4. Number Input

```
<label for="age">Age:</label>  
<input type="number" id="age" name="age" min="1" max="120">
```

Checkbox

```
<input type="checkbox" id="subscribe" name="subscribe">  
<label for="subscribe">Subscribe to newsletter</label>
```

6. Radio Buttons

```
<p>Select your gender:</p>  
<input type="radio" id="male" name="gender" value="male">  
<label for="male">Male</label><br>  
<input type="radio" id="female" name="gender" value="female">  
<label for="female">Female</label><br>  
<input type="radio" id="other" name="gender" value="other">  
<label for="other">Other</label>
```

7. Dropdown Select Menu

```
<label for="country">Country:</label>  
<select id="country" name="country">  
<option value="">Select a country</option>  
<option value="usa">United States</option>  
<option value="canada">Canada</option>  
<option value="uk">United Kingdom</option>  
</select>
```

8. Text Area

```
<label for="message">Message:</label>
```

```
<textarea id="message" name="message" rows="4" cols="50"></textarea>
```

9. File Upload

```
<label for="file">Select a file:</label>
```

```
<input type="file" id="file" name="file">
```

10. Submit Button

```
<input type="submit" value="Submit Form">
```

Form Validation Attributes

- required: Field must be filled before submitting
- minlength/maxlength: Minimum/maximum number of characters
- min/max: Minimum/maximum values for number inputs
- pattern: Regular expression to validate input

```
<input type="text" id="username" name="username" required minlength="4"
maxlength="20">
```

```
<input type="email" id="email" name="email" required>
```

```
<input type="tel" id="phone" name="phone" pattern="[0-9]{3}-[0-9]{3}-[0-9]{4}">
```

Assignments

Practical Task 1: Create Your First Webpage

1. Create a new HTML file named index.html
2. Set up the proper HTML document structure
3. Add a main heading with your name
4. Include a paragraph about yourself
5. Add an unordered list of your three favorite hobbies
6. Include a link to a website you frequently visit

Practical Task 2: Build a Simple Registration Form

Create an HTML form that includes:

1. Text inputs for first name, last name, and username
2. Email input
3. Password input with a confirmation field
4. Radio buttons for selecting gender
5. Checkbox for agreeing to terms and conditions
6. Dropdown menu for selecting country
7. Submit button

Knowledge Check

Reflection Questions

1. What is the purpose of the `<!DOCTYPE html>` declaration?
2. How do `<div>` and `` elements differ, and when would you use each?
3. Why is it important to include the `alt` attribute with images?
4. What's the difference between the GET and POST methods in forms?

Quick Quiz

1. Which tag is used to create a hyperlink?
 - a) `<link>`
 - b) `<a>`
 - c) `<href>`
 - d) `<hyperlink>`
2. Which of these is a semantic HTML element?
 - a) `<div>`
 - b) ``
 - c) `<article>`
 - d) ``

3. What attribute is required in an tag?
 - a) width
 - b) height
 - c) src
 - d) border
4. What does HTML stand for?
 - a) HyperText Markup Language
 - b) High-Tech Modern Language
 - c) Home Tool Markup Language
 - d) Hyper Technical Meta Language

Additional Resources

Further Reading

- [MDN Web Docs: HTML Basics](#)
- [W3Schools HTML Tutorial](#)
- [HTML5 Doctor: HTML5 Elements](#)

Tools

- [HTML Validator](#)
 - [CodePen](#) - Practice HTML online
 - [Visual Studio Code](#) - Recommended code editor
-

SESSION 2: MEDIA, SEMANTICS & ACCESSIBILITY

Introduction

Objective: Learn how to incorporate various media elements into web pages, understand semantic HTML, and implement web accessibility principles.

Relevance: Modern websites include more than just text. As frontend developers, you need to know how to properly integrate images, audio, video, and other media. Additionally, semantic HTML and accessibility principles ensure your websites are usable by all people, including those with disabilities, which is not only a legal requirement in many cases but also leads to better user experiences and SEO outcomes.

Lesson Overview

Topics Covered:

- Embedding different types of media in HTML
- Understanding and implementing semantic HTML
- Web accessibility principles and practices
- ARIA roles and attributes

Core Content

Media Elements in HTML

Images

The most basic way to add images is using the `` tag:

```

```

Important attributes:

- `src`: Path to the image file
- `alt`: Alternative text (for accessibility)
- `width` and `height`: Dimensions in pixels

Responsive Images:

```
<picture>  
<source media="(min-width: 768px)" srcset="large-image.jpg">  
<source media="(min-width: 480px)" srcset="medium-image.jpg">  
  
</picture>
```

Audio

HTML5 introduced the <audio> element:

```
<audio controls>  
<source src="audio.mp3" type="audio/mpeg">  
<source src="audio.ogg" type="audio/ogg">
```

Your browser does not support the audio element.

```
</audio>
```

Attributes:

- controls: Shows the audio controls (play/pause, volume)
- autoplay: Starts playing automatically
- loop: Plays the audio repeatedly
- muted: Audio is muted by default

Video

Videos can be embedded using the <video> element:

```
<video width="320" height="240" controls>  
<source src="movie.mp4" type="video/mp4">  
<source src="movie.webm" type="video/webm">
```

Your browser does not support the video tag.

```
</video>
```

Attributes:

- controls: Shows video controls
- poster: Specifies an image to show before the video is played

- autoplay, loop, muted: Similar to audio attributes

iframes

Embedding external content like maps or videos from other sites:

```
<!-- YouTube video -->
```

```
<iframe width="560" height="315"  
src="https://www.youtube.com/embed/VIDEO_ID" frameborder="0"  
allowfullscreen></iframe>
```

```
<!-- Google Maps -->
```

```
<iframe src="https://www.google.com/maps/embed?pb=MAP_DATA"  
width="600" height="450" style="border:0;" allowfullscreen=""  
loading="lazy"></iframe>
```

Semantic HTML

Semantic HTML uses tags that convey meaning about the content they contain, not just how it should look.

Why Use Semantic HTML?

- Improves accessibility for screen readers
- Enhances SEO
- Makes code more maintainable
- Provides better structure

Common Semantic Elements

<header>

<h1>Website Title</h1>

<nav>

Home

About

Contact

</nav>

</header>

<main>

<section id="about">

<h2>About Us</h2>

<p>Information about the company.</p>

</section>

<article>

<h2>Blog Post Title</h2>

<p>Published on <time datetime="2023-09-15">September 15, 2023</time></p>

<p>Article content goes here...</p>

<figure>

<figcaption>Caption for the image</figcaption>

</figure>

</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>&copy; 2023 Bonnie Computer Hub. All rights reserved.</p>
</footer>
```

Key semantic elements explained:

- <header>: Introductory content or navigation
- <nav>: Navigation links
- <main>: Main content of the page
- <section>: Standalone section
- <article>: Independent, self-contained content
- <aside>: Content related to surrounding content
- <figure> and <figcaption>: Image with caption
- <footer>: Footer of the page or section
- <time>: Time or date

Web Accessibility

Web accessibility means designing websites so that people with disabilities can use them. This includes people with visual, auditory, physical, speech, cognitive, and neurological disabilities.

WCAG (Web Content Accessibility Guidelines)

These guidelines are organized around four principles, often referred to as POUR:

- **Perceivable:** Information must be presentable to users in ways they can perceive
- **Operable:** User interface components must be operable
- **Understandable:** Information and operation must be understandable
- **Robust:** Content must be robust enough to be interpreted by a variety of user agents

Accessibility Best Practices

1. **Use semantic HTML** (as covered above)
2. **Provide alternative text for images**

```

```

3. **Create accessible forms**

```
<form>
```

```
<div>
```

```
<label for="name">Name:</label>
```

```
<input type="text" id="name" name="name" required>
```

```
</div>
```

```
<div>
```

```
<label for="email">Email:</label>
```

```
<input type="email" id="email" name="email" required>
```

```
<span id="email-error" class="error" role="alert"></span>
```

```
</div>
```

```
<button type="submit">Submit</button>
```

```
</form>
```

4. **Use proper heading structure** (h1-h6 in hierarchical order)
5. **Ensure keyboard navigation works**
 - All interactive elements should be keyboard accessible

- Visual focus indicators should be clear

6. Add skip navigation links

```
<a href="#main-content" class="skip-link">Skip to main content</a>
<!-- Other header content -->
<main id="main-content">
<!-- Main content -->
</main>
```

7. Make tables accessible

```
<table>
<caption>Monthly Savings</caption>
<thead>
<tr>
<th scope="col">Month</th>
<th scope="col">Amount</th>
</tr>
</thead>
<tbody>
<tr>
<th scope="row">January</th>
<td>$100</td>
</tr>
<tr>
<th scope="row">February</th>
<td>$150</td>
</tr>
</tbody>
</table>
```


ARIA (Accessible Rich Internet Applications)

ARIA is a set of attributes that define ways to make web content more accessible to people with disabilities.

Important ARIA Concepts

- **Roles:** Define what an element is or does
- **Properties:** Define characteristics of elements
- **States:** Define current conditions of elements

Common ARIA Attributes

<!-- Alert -->

<div role="alert">Your form has been submitted successfully!</div>

<!-- Button that's not a button element -->

<div role="button" tabindex="0">Click me</div>

<!-- Custom checkbox -->

<div role="checkbox" aria-checked="false" tabindex="0">I agree to terms</div>

<!-- Dialog -->

<div role="dialog" aria-labelledby="dialog-title">

<h2 id="dialog-title">Confirm Action</h2>

<p>Are you sure you want to proceed?</p>

<button>Yes</button>

<button>No</button>

</div>

<!-- Hidden content -->

<div aria-hidden="true">This will be hidden from screen readers</div>

<!-- Required fields -->

<label for="username">Username:</label>

<input id="username" aria-required="true">

Important note: The first rule of ARIA is not to use it if native HTML can provide the same functionality. Always prefer semantic HTML when possible.

Assignments

Practical Task 1: Multimedia Webpage

Create a webpage about your favorite hobby that includes:

1. A header with title and navigation
2. At least one image with appropriate alt text
3. An embedded video (YouTube or a video file)
4. An audio player with music or sound related to the hobby
5. Proper semantic structure (header, main, sections, footer)

Practical Task 2: Accessibility Evaluation

1. Visit a popular website
2. Evaluate its accessibility using browser developer tools
3. Identify three accessibility issues and propose solutions
4. Write a short report on your findings

Knowledge Check

Reflection Questions

1. Why is semantic HTML important for both SEO and accessibility?
2. How would you make a complex form accessible to screen reader users?
3. What are the key differences between <article> and <section> elements?
4. In what situations would you use ARIA attributes instead of native HTML elements?

Quick Quiz

1. Which HTML5 element represents the main content of a document?
 - a) <content>
 - b) <main>
 - c) <body>
 - d) <section>
2. What does the alt attribute provide for images?
 - a) Alternative style options
 - b) Alternative text for users who cannot see the image
 - c) Attribution for the image source
 - d) Alternative loading mechanisms
3. Which of these is NOT a semantic HTML element?
 - a) <nav>
 - b) <div>
 - c) <article>
 - d) <footer>
4. What does ARIA stand for?
 - a) Advanced Rich Internet Applications
 - b) Accessible Rich Internet Applications
 - c) Advanced Responsive Internet Architecture
 - d) Accessible Resource Integration Application

Additional Resources

Further Reading

- [MDN Web Docs: HTML multimedia](#)
- [Web Accessibility Initiative \(WAI\)](#)
- [The A11Y Project](#)
- [HTML5 Semantic Elements Guide](#)

Tools

- [WAVE Web Accessibility Evaluation Tool](#)
 - [Axe: Accessibility Testing Tools](#)
 - [Color Contrast Checker](#)
 - [Screen Reader - NVDA \(Free\)](#)
-

SESSION 3: DEVELOPER TOOLS, PAGE STRUCTURE

Introduction

Objective: Learn to use browser developer tools effectively, understand proper HTML page structure, and create well-organized web pages.

Relevance: Browser developer tools are essential for frontend development, allowing you to inspect, debug, and optimize your code. Understanding proper page structure ensures that your websites are well-organized, maintainable, and provide a good user experience.

Lesson Overview

Topics Covered:

- Browser developer tools overview and usage
- HTML document structure best practices
- Creating well-organized layouts
- Common page components and design patterns
- Meta tags and their importance

Core Content

Browser Developer Tools

Developer tools (DevTools) are built into all modern browsers and provide essential functionality for web developers.

Accessing DevTools

- Chrome/Edge: F12 or Ctrl+Shift+I (Cmd+Option+I on Mac)
- Firefox: F12 or Ctrl+Shift+I
- Safari: Enable developer menu in preferences, then Cmd+Option+I

Key DevTools Panels

1. Elements Panel (HTML & CSS)

- Inspect and modify the DOM and CSS in real-time
- See the box model for any element
- Check accessibility properties

2. Console Panel

- View JavaScript errors and logs
- Execute JavaScript code directly
- Monitor network operations

3. Network Panel

- Monitor network requests
- Check loading times
- Inspect headers and responses

4. Sources Panel

- View and debug JavaScript code
- Set breakpoints
- Monitor variables during execution

5. Application Panel

- Inspect cookies, local storage, and session storage
- Manage cache
- Work with web app manifests

Common DevTools Tasks

Inspecting Elements:

- Right-click on any element and select "Inspect"
- Use the Element selector tool (**Q**) to click directly on elements
- Navigate the DOM tree to find specific elements

Modifying Styles:

- Edit CSS properties in the Styles pane
- Toggle CSS properties on/off
- Add new CSS rules for testing

Debugging JavaScript:

- Set breakpoints by clicking on line numbers
- Step through code execution
- Monitor variable values in the Scope pane

Testing Responsiveness:

- Use the device toolbar (mobile icon) to simulate different devices
- Test different screen sizes by dragging handles
- Set custom dimensions for specific device testing

Proper HTML Document Structure

Basic Structure with Best Practices

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<!-- Character encoding - should be the first meta tag -->
```

```
<meta charset="UTF-8">
```

```
<!-- Viewport settings for responsive design -->
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<!-- SEO meta tags -->
```

```
<meta name="description" content="Description of your page for search engines">
```

```
<meta name="keywords" content="relevant, keywords, for, your, page">
```

```
<meta name="author" content="Your Name or Company">
```

```
<!-- Social media meta tags -->
<meta property="og:title" content="Page Title">
<meta property="og:description" content="Description for social media sharing">
<meta property="og:image" content="url-to-image.jpg">
<meta property="og:url" content="https://yourwebsite.com/page">

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

<!-- CSS stylesheets -->
<link rel="stylesheet" href="normalize.css">
<link rel="stylesheet" href="styles.css">

<!-- Page title -->
<title>Page Title | Bonnie Computer Hub</title>
</head>
<body>
<!-- Skip navigation for accessibility -->
<a href="#main-content" class="skip-link">Skip to main content</a>

<!-- Header -->
<header>
<div class="logo">

</div>
<nav>
<ul>
<li><a href="index.html">Home</a></li>
```



```
<li><a href="about.html">About</a></li>
<li><a href="services.html">Services</a></li>
<li><a href="contact.html">Contact</a></li>
</ul>
</nav>
</header>
```

```
<!-- Main content -->
<main id="main-content">
<section class="hero">
<h1>Welcome to Bonnie Computer Hub</h1>
<p>Empowering through technology</p>
<a href="#services" class="btn">Our Services</a>
</section>
```

```
<section id="services" class="services">
<h2>Our Services</h2>
<div class="services-grid">
<article class="service-card">
<h3>Web Development</h3>
<p>Custom websites built to meet your needs.</p>
</article>
<article class="service-card">
<h3>App Development</h3>
<p>Mobile applications for iOS and Android.</p>
</article>
<article class="service-card">
<h3>IT Training</h3>
```

<p>Professional development courses for all levels.</p>

</article>

</div>

</section>

<section class="testimonials">

<h2>What Our Clients Say</h2>

<div class="testimonial-slider">

<figure class="testimonial">

<blockquote>

<p>Bonnie Computer Hub transformed our business with their innovative solutions.</p>

</blockquote>

<figcaption>— Onduso Bonface, CEO of Company</figcaption>

</figure>

<!-- More testimonials... -->

</div>

</section>

</main>

<aside class="cta-sidebar">

<h3>Ready to Get Started?</h3>

<p>Contact us today for a free consultation.</p>

Contact Us

</aside>

<!-- Footer -->

```
<footer>
<div class="footer-columns">
<div class="footer-column">
<h4>Bonnie Computer Hub</h4>
<p>Empowering through technology</p>
<p>123 Main Street<br>City, State 12345</p>
</div>
```

```
<div class="footer-column">
<h4>Quick Links</h4>
<ul>
<li><a href="index.html">Home</a></li>
<li><a href="about.html">About</a></li>
<li><a href="services.html">Services</a></li>
<li><a href="contact.html">Contact</a></li>
</ul>
</div>
```

```
<div class="footer-column">
<h4>Connect With Us</h4>
<div class="social-links">
<a href="#" aria-label="Facebook">FB</a>
<a href="#" aria-label="Twitter">TW</a>
<a href="#" aria-label="Instagram">IG</a>
<a href="#" aria-label="LinkedIn">LI</a>
</div>
</div>
</div>
```

```
<div class="copyright">
<p>&copy; 2025 Bonnie Computer Hub. All rights reserved.</p>
</div>
</footer>
```

```
<!-- JavaScript (placed at the end for better performance) -->
<script src="script.js"></script>
</body>
</html>
```

Common Page Components and Design Patterns

Navigation Patterns

1. Horizontal Navigation Bar

```
<nav class="main-nav">
<ul>
<li><a href="#">Home</a></li>
<li><a href="#">About</a></li>
<li><a href="#">Services</a></li>
<li><a href="#">Contact</a></li>
</ul>
</nav>
```

2. Dropdown Navigation

```
<nav class="main-nav">
<ul>
<li><a href="#">Home</a></li>
<li>
<a href="#">Services</a>
<ul class="dropdown">
```

```

<li><a href="#">Web Design</a></li>
<li><a href="#">Development</a></li>
<li><a href="#">Consulting</a></li>
</ul>
</li>
<li><a href="#">Contact</a></li>
</ul>
</nav>

```

3. Hamburger Menu (for mobile)

```

<button class="menu-toggle" aria-expanded="false" aria-controls="main-menu">
<span class="sr-only">Menu</span>
<span class="hamburger"></span>
</button>
<nav id="main-menu" class="main-nav">
<!-- Navigation items -->
</nav>

```

Hero Sections

```

<section class="hero">
<div class="hero-content">
<h1>Welcome to Bonnie Computer Hub</h1>
<p class="tagline">Empowering through technology</p>
<div class="cta-buttons">
<a href="#services" class="btn primary">Our Services</a>
<a href="#contact" class="btn secondary">Contact Us</a>
</div>
</div>
<div class="hero-image">


```

</div>

</section>

Card Layouts

<section class="card-grid">

<article class="card">

<div class="card-content">

<h3>Web Development</h3>

<p>Create responsive websites with modern technologies.</p>

Learn more

</div>

</article>

<article class="card">

<!-- Similar structure for other cards -->

</article>

</section>

Call to Action (CTA) Sections

<section class="cta-section">

<h2>Ready to start your journey?</h2>

<p>Join thousands of students who have transformed their careers with us.</p>

Sign Up Today

</section>

Contact Forms

<section class="contact-section">

<h2>Get in Touch</h2>

<form class="contact-form" action="/submit-form" method="post">

<div class="form-group">

```

<label for="name">Name</label>
<input type="text" id="name" name="name" required>
</div>
<div class="form-group">
<label for="email">Email</label>
<input type="email" id="email" name="email" required>
</div>
<div class="form-group">
<label for="message">Message</label>
<textarea id="message" name="message" rows="5" required></textarea>
</div>
<button type="submit" class="btn submit-btn">Send Message</button>
</form>
</section>

```

Testimonial Sections

```

```html
<section class="testimonials">
<h2>What Our Students Say</h2>
<div class="testimonial-container">
<blockquote class="testimonial">
<p>"The courses at Bonnie Computer Hub gave me the skills and confidence to
start my career in web development."</p>
<cite>— Maria Johnson, Frontend Developer</cite>
</blockquote>

<blockquote class="testimonial">

```

```
<p>"Instructors are knowledgeable and the hands-on approach made learning programming enjoyable and effective."</p>
```

```
<cite>— John Smith, Software Engineer</cite>
```

```
</blockquote>
```

```
</div>
```

```
</section>
```

## Footer Structure

```
<footer class="site-footer">
```

```
<div class="footer-top">
```

```
<div class="footer-column">
```

```
<h3>About Us</h3>
```

```
<p>Bonnie Computer Hub is dedicated to empowering students through technology education since 2010.</p>
```

```
</div>
```

```
<div class="footer-column">
```

```
<h3>Quick Links</h3>
```

```
<ul class="footer-links">
```

```
Home
```

```
Courses
```

```
Resources
```

```
Contact
```

```

```

```
</div>
```

```
<div class="footer-column">
```

```
<h3>Contact Info</h3>
```

```
<address>
```



```
123 Education St.

Tech City, TC 12345

123-456-7890

info@bonniecomputerhub.com</
a>
</address>
</div>
</div>

<div class="footer-bottom">
<p>© 2023 Bonnie Computer Hub - Empowering through technology</p>
</div>
</footer>
```

## Meta Tags and Their Importance

Meta tags provide information about your HTML document that isn't displayed on the page but is used by browsers and search engines.

### Essential Meta Tags

#### 1. Character Encoding

```
<meta charset="UTF-8">
```

This specifies the character encoding for the document and should be the first meta tag in your document.

#### 2. Viewport

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

This helps with responsive design by controlling how the page is displayed on mobile devices.

#### 3. Description

```
<meta name="description" content="Learn web development with Bonnie Computer Hub's comprehensive courses for beginners and advanced developers.">
```

Provides a summary of the page content, which may be displayed in search engine results.

#### 4. **Keywords**

```
<meta name="keywords" content="web development, HTML, CSS, JavaScript, programming courses">
```

Although less important than in the past, this can still help with SEO.

#### 5. **Author**

```
<meta name="author" content="Bonnie Computer Hub">
```

Specifies the page author.

## Social Media Meta Tags

### 1. Open Graph Protocol (Facebook)

```
<meta property="og:title" content="Frontend Web Development Course |
Bonnie Computer Hub">
<meta property="og:description" content="Learn modern frontend
development skills from industry experts.">
<meta property="og:image"
content="https://www.bonniecomputerhub.com/images/course-banner.jpg">
<meta property="og:url"
content="https://www.bonniecomputerhub.com/courses/frontend">
<meta property="og:type" content="website">
```

### 2. Twitter Card

```
<meta name="twitter:card" content="summary_large_image">
<meta name="twitter:site" content="@BonnieComputerHub">
<meta name="twitter:title" content="Frontend Web Development Course">
<meta name="twitter:description" content="Learn modern frontend
development skills from industry experts.">
<meta name="twitter:image"
content="https://www.bonniecomputerhub.com/images/course-banner.jpg">
```

## SEO Meta Tags

### 1. Robots

```
<meta name="robots" content="index, follow">
```

Tells search engines whether to index the page and follow links.

### 2. Canonical URL

```
<link rel="canonical"
href="https://www.bonniecomputerhub.com/courses/frontend">
```

Specifies the preferred URL for the page to avoid duplicate content issues.

## Assignments

### Practical Task 1: Developer Tools Practice

1. Visit three different websites and use browser developer tools to:
  - Inspect their HTML structure
  - Examine their CSS properties
  - Check for JavaScript errors in the console
  - Analyze network requests and page loading times
2. Create a report documenting your findings, including screenshots of interesting elements or techniques you discovered.

### Practical Task 2: Create a Complete Webpage Structure

Build a complete HTML structure for a personal portfolio website that includes:

1. Proper document setup with all essential meta tags
2. Header with navigation
3. Hero section
4. Portfolio/projects section (using cards)
5. About me section
6. Skills section
7. Contact form
8. Footer with social media links
9. Proper semantic structure throughout

## Knowledge Check

### Reflection Questions

1. How can browser developer tools help you identify and fix issues in your web pages?
2. Why is it important to include meta tags in your HTML documents?
3. How does proper page structure contribute to SEO and accessibility?
4. What are the advantages of using semantic HTML elements for page structure instead of generic divs?

## Quick Quiz

1. Which developer tool panel would you use to inspect HTTP requests?
  - a) Elements
  - b) Console
  - c) Network
  - d) Sources
2. Which meta tag is essential for responsive web design?
  - a) `<meta name="description">`
  - b) `<meta name="keywords">`
  - c) `<meta name="viewport">`
  - d) `<meta name="author">`
3. What is the purpose of Open Graph meta tags?
  - a) To improve website loading speed
  - b) To control how content appears when shared on social media
  - c) To provide instructions to web crawlers
  - d) To set up webpage translations
4. Which HTML element should contain the primary navigation links?
  - a) `<header>`
  - b) `<nav>`
  - c) `<menu>`
  - d) `<links>`

## Additional Resources

### Further Reading

- [Chrome DevTools Documentation](#)
- [MDN: HTML — Structuring the Web](#)
- [The Open Graph Protocol](#)
- [HTML5 Doctor: Structuring a Page with HTML5](#)

## Tools

- [HTML5 Outliner](#) - Check document structure
- [Meta Tags](#) - Generate and test meta tags
- [Lighthouse](#) - Audit page quality
- [Can I Use](#) - Check browser compatibility