HTML Module Plan

Prerequisites

- Basic computer skills
- VS Code installed

Session 1: HTML Introduction & Basic Structure

Review & Introduction

- Course overview
- What is HTML and why it matters

Part 1: HTML Document Structure

Demo:

- Create HTML file
- DOCTYPE declaration
- HTML, head, body tags
- Title tag
- View in browser

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My First HTML Page</title>
</head>
<body>
    <!-- Content goes here -->
</body>
</html>
```

This creates the basic structure of an HTML document with the required DOCTYPE declaration, html, head, and body tags. The title appears in the browser tab.

Apply:

- Create your first HTML file
- Add proper structure
- Set a title

Part 2: Headings and Paragraphs

Demo:

- Headings (h1-h6)
- Paragraphs (p)
- Comments

```html
<h1>Main Heading</h1>
<h2>Subheading</h2>
<h3>Section heading</h3>
<!-- More heading levels: h4, h5, h6 -->

<p>This is a paragraph of text. HTML will automatically wrap the text to fit the
screen.</p>

<!-- This is a comment that won't be displayed in the browser -->
```

Headings provide structure to content with h1 being the most important. Paragraphs contain blocks of text. Comments help document code but aren't visible to users.

Apply:

- Add main heading with your name
- Add secondary heading
- Write paragraphs about yourself
- Add comments

Part 3: Text Formatting

Demo:

- Basic formatting: b, i, u
- Semantic formatting: strong, em, mark
- Line breaks (br)
- Horizontal rules (hr)

```html
<!-- Basic formatting -->
<p>This is <b>bold</b> and this is <i>italic</i> and this is <u>underlined</u>.
</p>

<!-- Semantic formatting -->
<p>This is <strong>important</strong> and this is <em>emphasized</em> and this is
<mark>highlighted</mark>.</p>

<!-- Line breaks -->
<p>This is line one.<br>This is line two.</p>

<!-- Horizontal rule -->
<hr>
```

Text formatting allows emphasis of content. Semantic elements (strong, em) have meaning beyond just visual styling. Line breaks force new lines within paragraphs, and horizontal rules create visual dividers.

Apply:

- Format key words
- Add section dividers
- Use line breaks

Homework

- Enhance your page with more headings and paragraphs
- Try different text formatting
- Add section about your education

Session 2: Links & Images

Review

- Quick review of session 1
- Check homework

Part 1: Links

Demo:

- Basic links (a tag)
- Internal vs external links
- Email links (mailto:)
- Target attribute
- ID links

```html
<!-- External link -->
<a href="https://www.example.com">Visit Example Website</a>

<!-- Internal link (to another page in same site) -->
<a href="about.html">About Us</a>

<!-- Email link -->
<a href="mailto:contact@example.com">Contact Us</a>

<!-- Open in new tab -->
<a href="https://www.example.com" target="_blank">Visit Example (New Tab)</a>

<!-- Link to page section (using ID) -->
<a href="#section1">Jump to Section 1</a>

<!-- Section with ID (target for link above) -->
<h2 id="section1">Section 1</h2>
```

Links connect pages and resources. The href attribute specifies the destination. Different types serve different purposes (external sites, internal pages, emails). The target attribute controls how links open.

Apply:

- Add links to websites
- Create email link
- Make links open in new tab
- Create page section links

Part 2: Images

Demo:

- Adding images (img tag)
- src, alt, width, height attributes
- Image file types
- File organization

```html
<!-- Basic image -->
<img src="images/photo.jpg" alt="Description of the image">

<!-- Image with size attributes -->
<img src="images/logo.png" alt="Company Logo" width="200" height="100">

<!-- Responsive image -->
<img src="images/responsive-image.jpg" alt="Responsive image" style="max-width:100%;">
```

The img element displays images. The src attribute points to the image file, while alt provides a text description for accessibility. Width and height attributes control size.

Apply:

- Add 2-3 images with alt text
- Adjust image sizes
- Create images folder

Part 3: Multi-Page Setup

Demo:

- Block vs inline elements
- Creating second page
- Navigation between pages
- File structure

```html
<!-- Block elements (create new lines) -->
<div>This is a block element</div>
```

```
<p>This is another block element</p>

<!-- Inline elements (flow within text) -->
This text has <span>an inline element</span> within it.

<!-- Basic navigation between pages -->
<nav>
    <a href="index.html">Home</a>
    <a href="about.html">About</a>
    <a href="contact.html">Contact</a>
</nav>
```

Block elements create new lines, while inline elements flow within text. Multi-page sites need consistent navigation to connect pages.

Apply:

- Create second HTML page
- Add navigation links
- Test navigation

Homework

- Create "My Hobbies" page with:
  - 3+ hobbies with descriptions
  - Images for each hobby
  - Links to related sites
  - Navigation back to home

Session 3: Lists

Review

- Review of links and images
- Check homework

Part 1: Unordered Lists

Demo:

- Creating unordered lists (ul, li)
- Changing bullet style
- Nested lists

```
<!-- Basic unordered list -->
<ul>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
</ul>
```

```
<!-- With style attribute -->
<ul style="list-style-type: square;">
    <li>Square bullet item</li>
    <li>Another square bullet</li>
</ul>

<!-- Nested list -->
<ul>
    <li>Main item 1
        <ul>
            <li>Sub-item 1.1</li>
            <li>Sub-item 1.2</li>
        </ul>
    </li>
    <li>Main item 2</li>
</ul>
```

Unordered lists display items with bullets. The list-style-type can change bullet appearance. Lists can be nested to show hierarchy.

Apply:

- Create list of favorite foods
- Add nested categories
- Try different content

Part 2: Ordered Lists

Demo:

- Creating ordered lists (ol, li)
- Changing number type
- Starting number
- Nested ordered lists

```
<!-- Basic ordered list -->
<ol>
    <li>First step</li>
    <li>Second step</li>
    <li>Third step</li>
</ol>

<!-- With type attribute -->
<ol type="A">
    <li>Item A</li>
    <li>Item B</li>
    <li>Item C</li>
</ol>

<!-- With start attribute -->
```

```html
<ol start="5">
    <li>This is item #5</li>
    <li>This is item #6</li>
</ol>

<!-- Nested ordered list -->
<ol>
    <li>Main step
        <ol type="a">
            <li>Sub-step a</li>
            <li>Sub-step b</li>
        </ol>
    </li>
    <li>Next main step</li>
</ol>
```

Ordered lists use numbers or letters for sequence. The type attribute changes numbering style (1, A, a, I, i), and start sets the initial number.

Apply:

- Create top 5 list
- Make step-by-step instructions
- Combine list types

Part 3: Audio Elements

Demo:

- Audio tag
- Controls and autoplay
- Multiple sources
- File formats

```html
<!-- Basic audio player -->
<audio controls>
    <source src="audio/sample.mp3" type="audio/mpeg">
    <source src="audio/sample.ogg" type="audio/ogg">
    Your browser does not support audio playback.
</audio>

<!-- With autoplay and loop -->
<audio controls autoplay loop>
    <source src="audio/background.mp3" type="audio/mpeg">
</audio>
```

The audio element adds sound to webpages. Controls attribute shows playback buttons, and multiple source elements provide fallback formats for browser compatibility.

Apply:

- Add audio to your page
- Set up controls
- Test playback

Homework

- Create recipe page with:
    - Ingredients list (unordered)
    - Steps list (ordered)
    - Description list for terms
    - Optional audio

Session 4: Tables

Review

- Review of lists and audio
- Check homework

Part 1: Basic Table Structure

Demo:

- Table structure (table, tr, td)
- Headers (th)
- Table sections (thead, tbody, tfoot)
- Basic formatting

```html
<!-- Basic table -->
<table border="1">
    <thead>
        <tr>
            <th>Name</th>
            <th>Age</th>
            <th>Country</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>John</td>
            <td>25</td>
            <td>USA</td>
        </tr>
        <tr>
            <td>Maria</td>
            <td>30</td>
            <td>Spain</td>
        </tr>
    </tbody>
    <tfoot>
        <tr>
```

```
            <td colspan="3">Total: 2 persons</td>
        </tr>
    </tfoot>
</table>
```

Tables organize data in rows and columns. The thead, tbody, and tfoot elements group table sections. Table header (th) cells are distinguished from regular data (td) cells.

Apply:

- Create simple 3×4 table
- Add headers
- Organize with sections

Part 2: Spanning Cells

Demo:

- Column spans (colspan)
- Row spans (rowspan)
- Combined spans

```
<!-- Table with column and row spans -->
<table border="1">
    <tr>
        <th>Time</th>
        <th>Monday</th>
        <th>Tuesday</th>
        <th>Wednesday</th>
    </tr>
    <tr>
        <td>9:00-10:00</td>
        <td>Math</td>
        <td>English</td>
        <td>Science</td>
    </tr>
    <tr>
        <td>10:00-12:00</td>
        <td colspan="2">Workshop</td>
        <td>Art</td>
    </tr>
    <tr>
        <td>12:00-1:00</td>
        <td colspan="3">Lunch</td>
    </tr>
    <tr>
        <td>1:00-3:00</td>
        <td>History</td>
        <td rowspan="2">Project</td>
        <td>Music</td>
    </tr>
    <tr>
        <td>3:00-4:00</td>
        <td>PE</td>
        <td>Computer</td>
    </tr>
</table>
```

Colspan and rowspan attributes allow cells to span multiple columns or rows, useful for schedules and complex data.

Apply:

- Create weekly schedule
- Use spans for longer activities
- Add day/time headers

Part 3: Table Accessibility

Demo:

- Adding captions
- Scope attribute
- Basic styling
- Borders and spacing

```
<!-- Accessible table -->
<table border="1">
    <caption>Monthly Expenses</caption>
    <thead>
        <tr>
            <th scope="col">Category</th>
            <th scope="col">January</th>
            <th scope="col">February</th>
            <th scope="col">March</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <th scope="row">Rent</th>
            <td>$1000</td>
            <td>$1000</td>
            <td>$1000</td>
        </tr>
        <tr>
            <th scope="row">Utilities</th>
            <td>$150</td>
            <td>$140</td>
            <td>$130</td>
        </tr>
        <tr>
            <th scope="row">Groceries</th>
```

```
                <td>$350</td>
                <td>$375</td>
                <td>$340</td>
            </tr>
        </tbody>
    </table>
```

Caption provides a title for the table. The scope attribute helps screen readers identify row/column headers, improving accessibility for users with disabilities.

Apply:

- Add captions and accessibility
- Make tables more readable
- Add zebra stripes if time allows

Homework

- Create product comparison table:
  - 4+ products, 5+ features
  - Use merged cells
  - Add headers and caption
  - Apply basic styling

Session 5: Forms

Review

- Review of tables
- Check homework

Part 1: Form Basics

Demo:

- Creating forms (form tag)
- Text inputs and labels
- Input types: text, email, password
- Placeholder and required attributes

```
<!-- Basic form -->
<form action="submit.php" method="post">
    <div>
        <label for="name">Name:</label>
        <input type="text" id="name" name="name" placeholder="Enter your name"
required>
    </div>
    <div>
        <label for="email">Email:</label>
        <input type="email" id="email" name="email" placeholder="Enter your email"
```

```
required>
    </div>
    <div>
        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required>
    </div>
</form>
```

Forms collect user input. Labels improve accessibility by connecting to input fields. The required attribute enforces mandatory fields, while placeholders provide hints.

Apply:

- Create basic info form:
  - Name field
  - Email field
  - Password field
  - Add labels
  - Set required fields

Part 2: More Input Types

Demo:

- Checkboxes and radio buttons
- Number, date, and color inputs
- Select dropdowns
- Textarea

```
<!-- Various input types -->
<form>
    <!-- Radio buttons -->
    <div>
        <p>Gender:</p>
        <input type="radio" id="male" name="gender" value="male">
        <label for="male">Male</label>
        <input type="radio" id="female" name="gender" value="female">
        <label for="female">Female</label>
        <input type="radio" id="other" name="gender" value="other">
        <label for="other">Other</label>
    </div>

    <!-- Checkboxes -->
    <div>
        <p>Interests:</p>
        <input type="checkbox" id="coding" name="interests" value="coding">
        <label for="coding">Coding</label>
        <input type="checkbox" id="music" name="interests" value="music">
        <label for="music">Music</label>
        <input type="checkbox" id="sports" name="interests" value="sports">
```

```
            <label for="sports">Sports</label>
        </div>

        <!-- Other input types -->
        <div>
            <label for="birthday">Birthday:</label>
            <input type="date" id="birthday" name="birthday">
        </div>
        <div>
            <label for="favorite-color">Favorite Color:</label>
            <input type="color" id="favorite-color" name="color">
        </div>
        <div>
            <label for="comments">Comments:</label>
            <textarea id="comments" name="comments" rows="4" cols="50"></textarea>
        </div>
        <div>
            <label for="country">Country:</label>
            <select id="country" name="country">
                <option value="usa">USA</option>
                <option value="canada">Canada</option>
                <option value="uk">United Kingdom</option>
                <option value="australia">Australia</option>
            </select>
        </div>
    </form>
```

HTML5 offers specialized input types for different data. Radio buttons allow single selection from options, while checkboxes allow multiple selections. Textarea provides multi-line text input, and select creates dropdown menus.

Apply:

- Add to your form:
  - Gender selection (radio)
  - Interests (checkboxes)
  - Birthday (date)
  - Favorite color (color picker)
  - Comments (textarea)

Part 3: Form Organization

Demo:

- Fieldset and legend
- Buttons (submit, reset)
- HTML5 validation
- Form action and method

```html
<!-- Organized form with fieldsets -->
<form action="process.php" method="post">
    <fieldset>
        <legend>Personal Information</legend>
        <div>
            <label for="fullname">Full Name:</label>
            <input type="text" id="fullname" name="fullname" required>
        </div>
        <div>
            <label for="email">Email:</label>
            <input type="email" id="email" name="email" required>
        </div>
    </fieldset>

    <fieldset>
        <legend>Preferences</legend>
        <div>
            <label for="favorite-website">Favorite Website:</label>
            <input type="url" id="favorite-website" name="website">
        </div>
        <div>
            <label for="satisfaction">Satisfaction (1-10):</label>
            <input type="range" id="satisfaction" name="satisfaction" min="1" max="10">
        </div>
    </fieldset>

    <div>
        <input type="submit" value="Submit Form">
        <input type="reset" value="Clear Form">
    </div>
</form>
```

Fieldset groups related form elements with a legend as title. Submit buttons send data, while reset buttons clear the form. Form validation helps ensure correct data input.

Apply:

- Organize form with fieldsets
- Add submit and reset buttons
- Add validation
- Improve styling

Homework

- Create registration form with:
  - Personal details section
  - Account preferences
  - 8+ input types
  - Proper validation

    - Fieldsets
    - Good styling

Advanced Session: Semantic HTML

Review

- Review of all HTML concepts
- Check homework

Part 1: Semantic Elements

Demo:

- Header, footer, nav, main
- Article, section, aside
- Figure, figcaption, time
- Accessibility and SEO benefits

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Semantic HTML Example</title>
</head>
<body>
    <header>
        <h1>Website Title</h1>
        <nav>
            <ul>
                <li><a href="/">Home</a></li>
                <li><a href="/about">About</a></li>
                <li><a href="/contact">Contact</a></li>
            </ul>
        </nav>
    </header>

    <main>
        <section>
            <h2>Main Content Section</h2>
            <article>
                <h3>Article Title</h3>
                <p>Article content goes here.</p>
                <time datetime="2023-05-15">May 15, 2023</time>
            </article>
        </section>

        <aside>
            <h3>Related Information</h3>
            <p>Sidebar content goes here.</p>
        </aside>
    </main>
    <footer>
        <p>&copy; 2023 Example Company</p>
    </footer>
</body>
</html>
```

Semantic elements describe their meaning to browsers and developers. Header, footer, nav, and main define page regions, while article, section, and aside organize content. This improves accessibility and search engine optimization.

Apply:

- Convert div layout to semantic
- Add ARIA roles
- Test with screen reader

Part 2: Embedding Content

Demo:

- Embedding videos (iframe)
- Embedding maps
- HTML entities
- Video tag

```
<!-- YouTube video embed -->
<iframe width="560" height="315" src="https://www.youtube.com/embed/dQw4w9WgXcQ"
        title="YouTube video player" allowfullscreen></iframe>

<!-- Google Maps embed -->
<iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3022.1013991811613!2d-74.00594523469134!3d40.7117845179329!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x89c25a197c06b7cb%3A0x40a06c78f79e5de6!2sStatue%20of%20Liberty%20National%20Monument!5e0!3m2!1sen!2sus!4v1658575262693!5m2!1sen!2sus"
        width="600" height="450" style="border:0;" allowfullscreen="" loading="lazy"></iframe>

<!-- HTML entities -->
<p>Special characters: &copy; copyright, &reg; registered trademark, &trade; trademark, &lt; less than, &gt; greater than, &amp; ampersand</p>

<!-- Video tag -->
<video width="320" height="240" controls>
    <source src="movie.mp4" type="video/mp4">
    Your browser does not support the video tag.
</video>
```

Iframe elements embed external content like videos and maps. HTML entities display special characters. The video element provides native video playback with browser controls.

Apply:

- Add YouTube video
- Add Google Map
- Use special characters

Part 3: Final Project Planning

Demo:

- Planning multi-page site
- Wireframes
- File organization
- Browser testing

Apply:

- Plan your final project
- Create site map and wireframes
- Set up folder structure
- Start homepage

Final Assessment

Create a multi-page website using HTML only (no CSS yet) with:

- Homepage with semantic structure
- About page with lists and tables
- Gallery page with images
- Contact page with form
- Navigation between all pages
- At least one embedded element
- Proper semantic HTML

Resources

- MDN Web Docs - HTML
- W3Schools HTML Tutorial
- HTML Validator
- HTML Cheat Sheet