1. Introduction to HTML

Time: 1–2 hours

- 1.1 What is HTML?
- 1.2 Importance of HTML in Web Development
- 1.3 How Browsers Interpret HTML
- 1.4 Project: Set up a basic "Hello World" HTML page
    - Save as index.html, open in a browser with <h1>Hello, World!</h1>.

2. Setting Up Your Development Environment

Time: 1 hour

- 2.1 Choosing a Code Editor
    - Download VS Code from code.visualstudio.com. Install it, then open it—no prior experience needed!
- 2.2 Installing Useful Extensions for HTML
    - In VS Code, click the Extensions icon (sidebar), search "Live Server," and click Install to view your HTML live.
- 2.3 Basic Shortcuts & Productivity Tips
- 2.4 Task: Install VS Code and set up an HTML boilerplate
    - Create index.html (right-click Explorer > New File) with: html

    -
      ```
      <!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <title>My First Page</title> </head> <body> <h1>Hello, World!</h1> </body> </html>
      ```

3. Basic Structure of an HTML Document

Time: 2–3 hours

- 3.1 HTML Document Anatomy (<!DOCTYPE html>, <html>, <head>, <body>)
- 3.2 Understanding Meta Tags and Their Uses
- 3.3 Including External Files (CSS, JavaScript, Fonts)
- 3.4 Project: Create a simple personal bio page
    - Include your name, a short bio, and a favorite quote.

4. Essential HTML Elements

Time: 3–4 hours

- 4.1 Headings (<h1> to <h6>)

- 4.2 Paragraphs (<p>) and Line Breaks (<br>)
- 4.3 Lists (<ul>, <ol>, <li>)
- 4.4 Links (<a> and Its Attributes)
- 4.5 Images (<img> and Its Attributes)
- 4.6 Project: Build a simple blog page
  - Use headings, paragraphs, a list, a link, and an image.

5. Forms and User Input

Time: 3–4 hours

- 5.1 Creating Forms (<form>, <label>, <input>, <button>)
- 5.2 Input Types (text, email, password, checkbox, radio, file, etc.)
- 5.3 Form Validation (required, pattern, min/max, etc.)
- 5.4 HTML5 Form Enhancements (datalist, placeholder, autocomplete)
- 5.5 Project: Create a signup form with validation
  - Include name, email, and password fields with validation.

6. Tables for Data Representation

Time: 2–3 hours

- 6.1 Creating Tables (<table>, <tr>, <td>, <th>)
- 6.2 Table Headers and Footers (<thead>, <tbody>, <tfoot>)
- 6.3 Merging Cells (colspan, rowspan)
- 6.4 Styling and Formatting Tables
- 6.5 Project: Create a pricing table for a product
  - Design 3 plans with features and prices.

7. HTML5 Semantic Elements

Time: 2–3 hours

- 7.1 Importance of Semantic HTML for SEO & Accessibility
- 7.2 Common Semantic Elements (<header>, <nav>, <section>, <article>, <aside>, <footer>)
- 7.3 When to Use Non-Semantic Elements (<div>, <span>)
- 7.4 Project: Build a semantic blog layout
  - Include a header with nav, an article, and a footer.
- 7.5 CSS Bridge Task: Add basic styling
  - Create style.css, link it in <head> with <link rel="stylesheet" href="style.css">, and add:
    css

  -
    ```
    header { background-color: #f0f0f0; padding: 10px; } h1 { color: blue; } p { font-size: 18px; }
    ```
  - Apply to your blog layout (1–2 hours extra).

Mastery Milestone: Personal Homepage Project

Time: 3–5 hours

- 7.6 Project: Build a personal homepage
  - Use a semantic <header> (name/title), <nav> (menu links), <section> (bio), <img> (photo), and an embedded <video> or <audio>.
  - Optional: Style it with the CSS from 7.5.
- **Why**: Celebrate your HTML mastery before moving to CSS!

---

8. HTML5 Multimedia (Audio, Video, and Graphics)

Time: 2–3 hours

- 8.1 Embedding Audio (<audio>) and Video (<video>)
- 8.2 Using the <source> Tag for Multiple Formats
- 8.3 Introduction to Scalable Vector Graphics (SVG)
  - Try: <svg width="100" height="100"><circle cx="50" cy="50" r="40" fill="red" /></svg>. Change the color!
- 8.4 Using the <canvas> API for Drawing
  - Add: <canvas id="myCanvas" width="200" height="100"></canvas>. Optional JS: Draw a line (revisit with JavaScript later).
- 8.5 Project: Embed a YouTube video and an audio player on a webpage

---

9. Advanced HTML Features (*)

Time: 3–4 hours

Note: "Optional—revisit after CSS/JavaScript basics!"

- 9.1 Creating Progress Indicators (<progress>) and Meters (<meter>)
- 9.2 Using <details> and <summary> for Collapsible Content
- 9.3 Working with <template>, <slot>, and <shadow-root> for Web Components
- 9.4 Project: Build a collapsible FAQ section

10. Browser Compatibility & Best Practices (*)

Time: 2–3 hours

Note: "Optional—revisit after CSS/JavaScript basics!"

- 10.1 Understanding Cross-Browser Compatibility Issues
- 10.2 Testing HTML Features with Can I Use
- 10.3 Using Feature Detection (<noscript>, Modernizr)
- 10.4 Writing Fallbacks for Unsupported Features
- 10.5 Task: Test an HTML5 feature on different browsers

11. HTML APIs and Integrations (*)

Time: 4–5 hours

Note: "Optional—revisit after CSS/JavaScript basics!"

- 11.1 Geolocation API (Getting User Location)
- 11.2 Drag and Drop API (Making Elements Draggable)
- 11.3 Web Storage (localStorage, sessionStorage)
- 11.4 Fetch API for Fetching Data from APIs
- 11.5 Project: Build a weather app using the Geolocation API

---

12. Accessibility & SEO Best Practices (*)

Time: 2–3 hours

- 12.1 Understanding ARIA Attributes
- 12.2 Using alt Attributes and aria-label
- 12.3 Structuring HTML for SEO (Headings, Meta Tags, Open Graph)
- 12.4 Mobile-Friendly HTML (Viewport Meta Tag, Responsive Elements)
- 12.5 Project: Create a fully accessible and SEO-friendly webpage

13. Performance Optimization in HTML (*)

Time: 2 hours

- 13.1 Using Lazy Loading for Images and Videos (loading="lazy")
- 13.2 Minifying HTML for Faster Load Times
- 13.3 Reducing Unused HTML Elements
- 13.4 Prefetching and Preloading Resources (rel="preload")
- 13.5 Task: Optimize a slow-loading webpage

14. Version Control & Collaboration (Git/GitHub Basics) (*)

Time: 2–3 hours

- 14.1 Introduction to Version Control
- 14.2 Using Git for Tracking Changes
  - Install Git from git-scm.com, sign up at github.com.
- 14.3 Collaborating on GitHub (Pull Requests, Branches)
- 14.4 Hosting HTML Pages on GitHub Pages
- 14.5 Project: Push an HTML project to GitHub Pages