### ****Beginner Topics**** (Foundation)

**HTML Basics**

* 1. Introduction to HTML and its purpose.
  2. Structure of an HTML document (<!DOCTYPE>, <html>, <head>, <body>).
  3. Basic tags (<h1>-<h6>, <p>, <a>, <img>, <ul>, <ol>, <li>).

**Attributes and Properties**

* 1. Global attributes (id, class, style, title).
  2. Specific attributes for tags (src, alt for <img>, href for <a>).

**Forms and Inputs**

* 1. Form elements (<form>, <input>, <textarea>, <button>, <select>, <option>).
  2. Input types (text, password, email, number, file, etc.).
  3. Form attributes (action, method, name).

**Media Elements**

* 1. Embedding images (<img>).
  2. Videos and audio (<video>, <audio>, controls, autoplay, loop).

### ****Intermediate Topics**** (Building Real-World Layouts)

**Semantic HTML**

* 1. Using semantic elements (<header>, <footer>, <article>, <section>, <nav>, <main>).
  2. Accessibility and SEO benefits.

**Tables**

* 1. Creating tables with <table>, <tr>, <td>, <th>.
  2. Table attributes (colspan, rowspan, border, etc.).

**HTML Forms with Validation**

* 1. Required fields and basic HTML validation (required, pattern, maxlength).
  2. Input constraints (min, max, step).

**Links and Navigation**

* 1. Relative vs absolute URLs.
  2. Linking within the same page (<a href="#section">).
  3. Creating navigation menus.

### ****Advanced Topics**** (For Professional Web Development)

**Advanced Attributes**

* 1. Custom data attributes (data-\*).
  2. ARIA attributes for accessibility.

**HTML5 Features**

* 1. <canvas> for drawing graphics.
  2. <svg> for vector graphics.
  3. Geolocation API and Web Storage (localStorage, sessionStorage).

**Meta Tags**

* 1. <meta> for character encoding, viewport settings, and SEO.
  2. Open Graph and Twitter Cards for social media optimization.

**Web Components**

* 1. Shadow DOM, templates (<template>), and custom elements.

**Performance Optimization**

* 1. Lazy loading for images and iframes.
  2. Minifying HTML code.

**Version Control in HTML**

* 1. Working with Git for HTML projects.
  2. Best practices for code collaboration.

### ****HTML in MERN Context****

**HTML for React**

* 1. JSX (HTML-like syntax in React).
  2. Embedding dynamic content and components.

**HTML and Node.js**

* 1. Serving HTML files using Express.js.
  2. Building static vs dynamic HTML with Node.

**HTML for MongoDB Integration**

* 1. Using forms to send data to a backend (GET/POST requests).
  2. Validating and processing user input for databases.

### Learning Resources:

* **Documentation**: [MDN Web Docs](https://developer.mozilla.org/en-US/docs/Web/HTML" \t "_new).
* **Practice Platforms**: CodePen, freeCodeCamp, LeetCode (for front-end challenges).
* **Projects**: Build portfolio websites, landing pages, and integration with React/Node.