

HTML (Sessions 1–2)

Session 1: Introduction to HTML

Overview of HTML structure (DOCTYPE, tags, elements)

Basic tags: <html>, <head>, <title>, <body>, headings, paragraphs, links (<a>), lists (,)

Images: , attributes like alt, src

Practice: Create a simple webpage with headings, paragraphs, and links.

Session 2: Working with Forms and Tables

Form elements: <form>, <input>, <label>, <select>, <button>, <textarea>

Input types: text, email, password, radio, checkbox, submit

Tables: <table>, <tr>, <td>, <th>, colspan, rowspan

Practice: Build a form with validation and a table to display data.

Revised CSS Sessions (Sessions 3–6)***

Session 3: Introduction to CSS

- Basic syntax, selectors, properties
- Types of selectors: element, class, ID, and their specificity
- Working with text: font-family, font-size, font-weight, color, text-align
- Box model: margin, padding, border, width, height
- Specificity hierarchy: inline styles, IDs, classes, elements
- Practice: Style the webpage created in previous sessions using different selectors and understand the impact of specificity.

Session 4: CSS Layout and Positioning (Including Inline, Internal, External Styles)

- Display properties: `block`, `inline-block`, `inline`, `none`
- Internal CSS: using `

JavaScript (Sessions 7–14)

Session 7: Introduction to JavaScript (Variables and Data Types)

- JavaScript basics: variables (let, const, var)
- Data types:
 - Primitive types: string, number, boolean, undefined, null, symbol, bigint
 - Non-primitive types: Objects (including arrays and functions)
- Differences between primitive and non-primitive types
- Basic operations and type checking (typeof)
- Practice: Create a simple interactive page with basic variables, use different data types, and perform operations on them.

Session 8: Functions and Events

- JavaScript functions: declarations, expressions, arrow functions
- Event handling: onclick, addEventListener, event types
- Practice: Create a form that validates input using JavaScript functions.

Session 9: Hoisting and Scope

- Hoisting: How JavaScript moves variable and function declarations to the top of the scope
- Hoisting behavior for variables (var, let, const) and functions
- Scope: global scope vs local scope
- Practice: Write code that demonstrates hoisting and its effects, and explore scoping with functions and variables.

Session 10: Working with APIs (AJAX and Fetch)

- Introduction to APIs: what they are, how they work, and types (REST, etc.)
- Using Fetch API to make HTTP requests (GET, POST)
- Handling JSON data and understanding API responses
- Practice: Fetch data from a public API and display it on a webpage.

Session 11: Callbacks in JavaScript

- Introduction to callbacks: functions passed as arguments
- Callback patterns and how they work
- Common use cases: handling asynchronous behavior with callbacks (e.g., making multiple API requests in sequence)

- **Practice:** Use callbacks to perform API actions in sequence, handling user input and API responses.

Session 12: Promises and Asynchronous JavaScript

- **Introduction to Promises:** understanding how promises work (states: pending, fulfilled, rejected)
- **Promise chaining:** `.then()`, `.catch()`
- **Practice:** Refactor callback examples to use promises for handling asynchronous operations, improving code readability.

Session 13: Async/Await and Advanced Event Handling

- **Async/Await:** simplifying asynchronous code when working with promises
- **Advanced event handling:** event delegation, preventing default behavior
- **Practice:** Use `async/await` to refactor promise-based code, and handle advanced events in web applications.

Session 14: JavaScript DOM Projects

- **Build a small project** integrating HTML, CSS, and JavaScript
- **Create an interactive app** fetching data from an API (e.g., weather app, product search)