



No.14/20,4th A cross, 20th main, 1st stage, BTM LAYOUT Bangalore-560029

Website: www.techxonicinnovation.com, Email: info@techxonicinnovation.com Mobile: 9606399915, 9731808050.

Syllabus

ADVANCE TOPIC

HTML Essentials

Review of HTML Structure

HTML consists of elements and attributes that define the structure and content of a web page. Common elements include headings, paragraphs, lists, images, and links. Attributes such as id, class, and href help in styling and linking content.

Advanced HTML5 Features

Forms allow user input through various elements such as text fields, checkboxes, radio buttons, and dropdowns. HTML5 introduced new input types like email, date, and range, along with built-in validation. Multimedia elements like audio and video enable embedding media without external plugins. HTML5 APIs like Geolocation and Local Storage provide enhanced functionality.

Accessibility in HTML

Semantic HTML improves accessibility by using meaningful elements like header, nav, section, and footer. ARIA roles and attributes help screen readers interpret web content correctly. Best practices include using descriptive alt text for images and ensuring keyboard navigability.

Advanced CSS Fundamentals

CSS Selectors

CSS allows styling elements using various selectors. Basic selectors target elements by tag name, class, or ID. Advanced selectors include attribute selectors, combinators for selecting related elements, and pseudo-classes for targeting states like hover or focus.

CSS Variables and Layouts

CSS variables store reusable values for colors, fonts, and spacing, improving maintainability. Layout techniques like Flexbox provide efficient alignment and distribution of elements, allowing easy responsiveness and dynamic positioning.

CSS Grid Layout and Sass

Deep Dive into CSS Grid

CSS Grid enables complex layouts with rows and columns, offering precise control over element placement. It allows defining grid templates, naming areas, and automatically placing items in an optimized manner.

CSS Preprocessors (Sass/SCSS)

Sass extends CSS with features like variables, nesting, mixins, and functions. It improves code organization and reusability by enabling modular stylesheets. Projects using Sass benefit from better maintainability and efficiency.

JavaScript Fundamentals

JavaScript Syntax and Data Types

JavaScript is a versatile language with different data types, including numbers, strings, booleans, and objects. Understanding how data is stored and manipulated is essential for dynamic web development.

Functions, Scopes, and Closures

Functions allow code reuse and modularity. Scope determines variable accessibility, with closures enabling functions to retain access to variables even after execution. These concepts are crucial for writing efficient code.

Advanced JavaScript Concepts

DOM Manipulation and Event Handling

The Document Object Model (DOM) represents web pages in a structured manner. JavaScript enables dynamic changes to content and styles. Event handling captures user interactions like clicks and keypresses, allowing interactivity.

Asynchronous JavaScript

JavaScript can execute tasks asynchronously using callbacks, promises, and async/await. These concepts help handle API requests, loading external resources, and improving user experience by preventing blocking operations.

Introduction to ES6+ Features

Modern JavaScript includes features like arrow functions for cleaner syntax, destructuring for easier data extraction, and modules for organizing code into separate files, improving readability and maintainability.

ES6+ Syntax

ES6 introduced new syntax features like the spread operator for combining arrays, rest parameters for handling multiple function arguments, and template literals for dynamic string construction. These features simplify code.

Error Handling in JavaScript

JavaScript includes error handling mechanisms to catch and handle unexpected issues. Proper error management ensures robust applications and prevents crashes due to unforeseen conditions.

Deep Dive into the Fetch API

The Fetch API enables making network requests to retrieve and send data from servers. Understanding how to handle API responses and errors is essential for building dynamic, data-driven applications.

This structured content provides a detailed foundation for HTML, CSS, and JavaScript, covering both fundamental and advanced concepts essential for web development.

OLD SYLLABUS TOPIC

Basics of HTML 5:

Structure of an HTML document (<!DOCTYPE>, <html>, <head>, <body>)

Common HTML tags (<h1>-<h6>, <p>, <a>, , , , , <div>,)

Semantic HTML

Importance of semantic elements

Usage of tags like <header>, <footer>, <article>, <section>, <aside>, <main>

Forms and Input Elements

Form structure (<form>, <input>, <textarea>, <select>, <button>)

Input types (text, password, email, checkbox, radio, etc.)

Form attributes (action, method, required, etc.)

Media and Embedding

Using , <audio>, <video>, and <iframe>

Accessibility considerations with media

Basics of CSS :

CSS syntax (selectors, properties, values)

Inline, internal, and external CSS

Selectors and Specificity:

Types of selectors (class, ID, attribute, pseudo-classes, pseudo-elements)

Understanding specificity and inheritance

Box Model:

Understanding padding, margin, border, and content

Modifying box model with box-sizing

Flexbox:

Introduction to Flexbox

Containers and items

Common layout patterns (row, column, alignment, justification)

Grid Layout

Introduction to CSS Grid

Grid properties (grid-template-rows, grid-template-columns)

Creating complex layouts

Responsive Design Introduction:

Media queries and breakpoints

CSS Animations and Transitions:

Preprocessors

Introduction to SASS/LESS

Variables, nesting

JavaScript Basics:

Data types and variables (let, const, var)

Operators (arithmetic, comparison, logical)

Control structures (if, switch, loops)

Functions and Scope:

Function declarations, expressions, arrow functions

Understanding scope (global vs local)

DOM Manipulation:

Selecting elements (getElementById, querySelector, etc.)

Modifying elements (textContent, innerHTML, styles)

Adding/removing classes, creating elements, event listeners

Events:

Understanding event bubbling.

Common events (click, change, keypress)

ES6 Features:

Template literals

Destructuring

Default parameters

Asynchronous JavaScript

Understanding callbacks

Promises and async/await

Working with JSON

Parsing and stringifying JSON data

Debugging using DOM manipulation

Building a Simple Project

- Topics:

- Combine HTML, CSS, and JavaScript
- Project planning and structure
- Best practices for coding

- Practice: Build a small project (e.g., a to-do list, a personal portfolio) using everything we've learned.

briefing on current industry technical languages and frameworks. Exploring on the variety of designation and opportunities in the industry