7-Week JavaScript Course Outline

Table of Contents

7-Week JavaScript Course Outline

Objective: Provide students with a comprehensive understanding of JavaScript, from the fundamentals to advanced concepts, enabling them to build dynamic, interactive web applications.

Week 1: JavaScript Basics

Day 1: Introduction to JavaScript

- What is JavaScript?
- JavaScript's role in web development
- Setting up the environment (IDE, browser console)
- Writing and running your first script

Day 2: Variables and Data Types

- Declaring variables: var, let, const
- Primitive data types (string, number, boolean, etc.)
- Type conversion and coercion

Day 3: Operators and Expressions

- Arithmetic, comparison, and logical operators
- String concatenation and template literals
- Practice: Simple calculations and comparisons

Day 4: Control Structures

- Conditional statements: if, else if, else, switch
- Loops: for, while, do...while
- Practice: Number guessing game

Day 5: Functions

- Defining and invoking functions
- Parameters and return values
- Arrow functions and function expressions
- Practice: Create a function to find the maximum of three numbers

Week 2: Core JavaScript Concepts

Day 6: Arrays

- Declaring and working with arrays
- Common array methods (push, pop, map, filter, etc.)
- Practice: Manipulate a list of student grades

Day 7: Objects

- Creating and using objects
- Accessing and modifying properties
- Nested objects and arrays
- Practice: Model a library system

Day 8: DOM Manipulation Basics

- What is the DOM?
- Selecting elements: getElementById, querySelector
- Modifying elements: innerHTML, textContent, style
- Practice: Highlight specific text on a webpage

Day 9: Events

- Understanding events and event listeners
- Common event types (click, mouseover, keydown)
- Event delegation
- Practice: Create an interactive dropdown menu

Day 10: Error Handling and Debugging

- try...catch and throwing errors
- Debugging tools in the browser
- Writing clean, readable code
- Practice: Fix buggy code exercises

Week 3: Intermediate JavaScript

Day 11: Advanced Array and Object Methods

- Higher-order functions (map, reduce, filter)
- Object methods and destructuring
- Practice: Analyze an array of sales data

Day 12: Working with Dates and Times

- Date object basics
- Formatting and manipulating dates
- Practice: Countdown timer for an event

Day 13: JavaScript Classes

- Introduction to ES6 classes
- Constructor functions
- Methods and inheritance
- Practice: Create a class for a bank account system

Day 14: Scope and Closures

- Global vs local scope
- Closures and their applications
- Practice: Create a closure-based counter

Day 15: Finalizing Week Concepts

- Review and build a small project using week concepts

Week 4: Asynchronous JavaScript

Day 16: Introduction to Asynchronous Concepts

- Synchronous vs asynchronous behavior
- Callbacks
- Practice: Simulate a callback-based task

Day 17: Promises

- Creating and handling promises
- then and catch methods
- Practice: Simulate a delayed API response

Day 18: async/await

- Converting promises to async/await
- Error handling with try...catch
- Practice: Fetch mock data with async/await

Day 19: Fetch API

- Making GET and POST requests
- Handling JSON data
- Practice: Fetch and display user data from a public API

Day 20: Project Work

- Build a weather app or user directory using asynchronous JavaScript

---

Week 5: Advanced JavaScript Concepts

Day 21: JavaScript Modules

- Importing and exporting modules
- Advantages of modular code
- Practice: Split a project into multiple files

Day 22: ES6+ Features

- Destructuring, spread/rest operators
- Default parameters and template literals
- Practice: Refactor code with ES6+ features

Day 23: Regular Expressions

- Basics of regex and patterns
- Practical use cases in validation
- Practice: Validate email and phone numbers

Day 24: JavaScript Patterns

- Introduction to design patterns
- Singleton and module patterns
- Practice: Implement a reusable pattern in a project

Day 25: Review and Application

- Quick review of advanced concepts
- Work on a guided mini-project

Week 6: Integration with Web Technologies

Day 26: Introduction to APIs

- What are APIs?
- RESTful principles
- Practice: Build a small app interacting with a public API

Day 27: Working with LocalStorage and SessionStorage

- Storing and retrieving data locally
- Differences between LocalStorage and SessionStorage
- Practice: Save form inputs to LocalStorage

Day 28: Introduction to Frameworks/Libraries

- Brief overview of React and Vue.js
- How JavaScript integrates with frameworks
- Practice: Build a simple React/Vue app scaffold

Day 29: JavaScript Build Tools

- Introduction to tools like Webpack and Babel
- Why and when to use them
- Practice: Set up a simple project with Webpack

Day 30: Wrap-Up and Review

- Comprehensive review of all topics
- Prepare for the final project

Week 7: Capstone Project

Day 31-35: Build a Capstone Project

- Suggested projects:

- **Interactive Quiz App**: Fetch questions from an API, track scores, and display results.
- **Expense Tracker**: Manage and visualize monthly expenses with local storage support.
- **Simple Chat Application**: Real-time chat using a public API or WebSocket.

Day 36-37: Presentations and Feedback

- Present final projects to peers or instructors
- Peer review and constructive feedback

Day 38: Final Assessment

- Quiz covering theoretical and practical knowledge
- Evaluation of final projects