JAVASCRIPT COURSE SYLLABUS

Month 1: JavaScript Basics (Fundamentals & Core Concepts)

Weeks 1-2: JavaScript Syntax & Control Flow

- Day 1: Introduction to JavaScript & Setup
  - Setting up the development environment
  - Basic syntax: Variables (`var`, `let`, `const`) and Data Types (strings, numbers, booleans)
- Day 2: Operators and Expressions
  - Arithmetic, comparison, logical, and assignment operators
  - Understanding expressions and statements
- Day 3: Conditional Statements (Part 1)
  - `if`, `else if`, `else` control flow
- Day 4: Conditional Statements (Part 2)
  - `switch` case statements
- Day 5: Practice Exercises
  - Simple decision-making problems using conditionals

Weeks 3-4: Functions & Looping

- Day 1: Introduction to Functions
  - Function declaration, invocation, and function expressions
  - Return values and parameters
- Day 2: Arrow Functions & Scope
  - Understanding arrow functions
  - Scope and Hoisting in JavaScript (Block vs. Function Scope)
- Day 3: Loops in JavaScript (Part 1)
  - `for`, `while` loops: Iterating over arrays and counters
- Day 4: Loops in JavaScript (Part 2)
  - `for...of`, `for...in` loops: Working with objects and arrays
- Day 5: Looping Practice
  - Problems involving array manipulation and iterating over data structures

Month 2: Intermediate JavaScript + Basic Data Structures

Weeks 1-2: Arrays, Strings, and Objects

- Day 1: Arrays in JavaScript
  - Array declaration, accessing elements, and basic methods like push , pop , shift , unshift
- Day 2: Array Manipulation (Part 1)
  - More advanced array methods: map , filter , reduce
- Day 3: Strings in JavaScript
  - String methods: split , slice , substring , replace
- Day 4: Objects and Arrays of Objects
  - Creating objects, accessing and modifying object properties
  - Arrays of objects and nested structures
- Day 5: Object-Oriented Programming (Introduction)
  - Creating constructors, classes, and using this keyword

Weeks 3-4: Introduction to Data Structures

- Day 1: Recursion (Basics)
  - What is recursion, and when to use it
  - Writing basic recursive functions (e.g., factorial, Fibonacci)
- Day 2: Arrays as Data Structures
  - Introduction to common array-based algorithms (finding max/min, reversing arrays)
- Day 3: Stacks and Queues (Theory)
  - Introduction to stack and queue data structures
  - Basic operations (push, pop, enqueue, dequeue)
- Day 4: Implementation of Stacks and Queues in JavaScript
  - Building custom stack and queue structures using arrays
- Day 5: Problem Solving Using Stacks and Queues
  - Simple problems like balancing parentheses, reversing words, etc.

Month 3: Advanced JavaScript + Intermediate Data Structures and Algorithms

---

Weeks 1-2: More Advanced JavaScript Concepts

- Day 1: Closures in JavaScript
  - Understanding closures, lexical scope, and practical use cases
- Day 2: Asynchronous JavaScript (Part 1)
  - Introduction to asynchronous programming: Callbacks, Promises
- Day 3: Asynchronous JavaScript (Part 2)
  - `async` and `await` , handling asynchronous data flows
- Day 4: JavaScript Event Loop and Concurrency
  - How JavaScript handles asynchronous code execution
- Day 5: Error Handling
  - `try` , `catch` , `finally` , and custom error handling

Weeks 3-4: More Complex Data Structures

- Day 1: Linked Lists (Theory)
  - Introduction to singly and doubly linked lists
  - Operations: Insertion, deletion, traversal
- Day 2: Linked Lists Implementation
  - Writing linked lists in JavaScript
- Day 3: Trees (Introduction)
  - Binary trees, binary search trees: Introduction and operations (insertion, searching)
- Day 4: Graphs (Theory)
  - Introduction to graphs, basic traversal algorithms (DFS, BFS)
- Day 5: Practice Problems Using Linked Lists and Trees
  - Medium-level problems involving tree and linked list traversal