# Mastering Javascript

### 1. **Introduction to JavaScript**

* What is JavaScript?
* History and evolution of JavaScript
* How JavaScript works (Execution Context and Call Stack)
* Embedding JavaScript in HTML (Script Tag)
* Basic Development Environment Setup

### 2. **JavaScript Basics**

* Variables and Data Types
  + var, let, const
  + Primitive Data Types: String, Number, Boolean, Undefined, Null, Symbol
  + Non-primitive Types: Objects, Arrays, Functions
* Operators
  + Arithmetic, Assignment, Comparison, Logical, Bitwise, and Ternary Operators
* Control Structures
  + If-Else Statements
  + Switch Statements
  + Loops (For, While, Do-While)
  + Break, Continue

### 3. **Functions**

* Defining and Invoking Functions
* Parameters and Arguments
* Function Expressions vs Declarations
* Arrow Functions
* Higher-Order Functions
* Callbacks
* Immediately Invoked Function Expressions (IIFE)

### 4. **Objects and Arrays**

* Creating and Accessing Objects
* Object Methods and Properties
* "this" Keyword and its context
* Prototypes and Inheritance
* Arrays: Creating, Accessing, and Manipulating
* Array Methods: map, filter, reduce, forEach, find, etc.
* Spread Operator and Rest Parameters

### 5. **DOM Manipulation**

* Document Object Model (DOM) Basics
* Selecting Elements (getElementById, querySelector, etc.)
* Modifying Elements (InnerHTML, TextContent, Styles)
* Adding and Removing Elements
* Event Handling (addEventListener, onClick, etc.)
* Form Handling and Validation

### 6. **Error Handling and Debugging**

* Try, Catch, Finally
* Throwing Errors
* Debugging with Console and Breakpoints
* Common JavaScript Errors

### 7. **Asynchronous JavaScript**

* Callbacks and Callback Hell
* Promises
  + Creating Promises
  + Resolving and Rejecting
  + .then, .catch, .finally
* async/await Syntax
* Error Handling in Asynchronous Code

### 8. **JavaScript in the Browser**

* Window and Document Objects
* Timers (setTimeout, setInterval)
* Browser Events (Click, Load, Mouse Events, etc.)
* Local Storage, Session Storage, Cookies

### 9. **ES6+ Features**

* Block Scoping with let and const
* Template Literals
* Destructuring (Objects and Arrays)
* Default Parameters
* Classes and Constructor Functions
* Modules (import/export)
* Symbols and Iterators
* Generators (yield, next)

### 10. **Advanced JavaScript**

* Closures
* Scope and Hoisting
* Event Loop and the Call Stack
* JavaScript Engines (V8, SpiderMonkey)
* Memory Management and Garbage Collection
* Recursion
* Context and "this"
* Currying and Partial Application

### 11. **Object-Oriented Programming in JavaScript**

* Creating and Using Classes
* Class Inheritance
* Getters and Setters
* Static Methods and Properties
* Private Properties and Methods (# Syntax)

### 12. **Functional Programming in JavaScript**

* Pure Functions
* Immutability
* Function Composition
* Map, Reduce, Filter in Functional Programming
* Recursion vs Iteration
* First-Class Functions

### 13. **Asynchronous Patterns**

* Understanding the Event Loop
* Handling Promises Sequentially and Concurrently
* Chaining Promises
* Async/Await Patterns
* Using Fetch API
* Working with APIs (AJAX, HTTP Methods)

### 14. **JavaScript Testing**

* Introduction to Testing
* Writing Unit Tests
* Test-Driven Development (TDD)
* JavaScript Testing Frameworks (Jest, Mocha, Chai)
* Mocking and Stubbing Functions
* Debugging and Handling Errors

### 15. **JavaScript Modules and Build Tools**

* Modular JavaScript (ES Modules vs CommonJS)
* Importing and Exporting Modules
* JavaScript Build Tools: Webpack, Babel, Parcel
* Using NPM/Yarn for Package Management

### 16. **JavaScript Design Patterns**

* Singleton Pattern
* Module Pattern
* Observer Pattern
* Factory Pattern
* MVC Pattern (Model-View-Controller)
* Event Delegation

### 17. **JavaScript and the Web**

* AJAX and XMLHttpRequest
* Fetch API vs AJAX
* Working with Web APIs
* Using APIs with JSON
* CORS and Security Concerns

### 18. **Modern JavaScript Frameworks**

* Introduction to Frontend Frameworks (React, Angular, Vue)
* React Basics (Components, Props, State, Hooks)
* Angular Basics (Components, Services, Directives)
* Vue.js Basics (Directives, Components, Vue Instance)
* State Management (Redux, Vuex)
* Single-Page Applications (SPA) and Routing

### 19. **JavaScript and Backend Development**

* Introduction to Node.js
* Setting Up Node.js Environment
* Working with Files and Directories
* Creating a Web Server with Express.js
* RESTful APIs and CRUD Operations
* Connecting to Databases (MongoDB, MySQL)

### 20. **Performance Optimization and Best Practices**

* Memory Leaks and How to Avoid Them
* Efficient DOM Manipulation
* Debouncing and Throttling
* Lazy Loading
* Web Workers for Parallel Processing
* Caching and Local Storage

### 21. **Security in JavaScript**

* Cross-Site Scripting (XSS) and Prevention
* Cross-Site Request Forgery (CSRF)
* Secure Data Storage
* Input Validation and Sanitization
* Secure JavaScript Coding Practices

### 22. **TypeScript Basics (Optional)**

* Introduction to TypeScript
* Setting Up TypeScript
* Type Annotations and Type Inference
* Interfaces and Classes in TypeScript
* Advanced Types (Union, Intersection, Generics)

### 23. **JavaScript Projects**

* Beginner Projects (To-Do App, Calculator)
* Intermediate Projects (Weather App, Chat Application)
* Advanced Projects (Real-Time Collaboration Tools, E-Commerce Platform)

————————

This Table of Contents provides a progressive roadmap for mastering JavaScript from the fundamentals to expert-level concepts and practices. You can dive deeper into each topic as you advance through your learning journey.

#software/languages/javascript