

---

## Beginner JavaScript Topics

### Basics & Syntax

- Identifiers
- Literals
- Number Literals
- Escape Sequences
- Primitive vs Non-Primitive Data Types
- User-Defined Data Types
- Variables: let, var, const
- Hoisting
- Temporal Dead Zone (TDZ)
- Value of let and const in TDZ
- Comments in JS
- Strict Mode ("use strict")
- Conditional Operator / Ternary Operator
- Type Casting vs Type Coercion
- Boolean Coercion
- Truthy & Falsy Values
- Operator Precedence
- Bitwise Operators
- Nullish Coalescing Operator (??)
- Optional Chaining (?.)
- Spread vs Rest Operators
- Destructuring
- Difference between == and ===

---

### Control Flow & Loops

- if-else vs switch
- for, while, do-while Loops
- continue and break

- Labels in Loops
- 

## Data Structures & Methods

- String Methods
  - Array Methods
  - Array.prototype Chain
  - map, filter, reduce, forEach
  - Jagged Arrays
  - Sets & Maps (Set, Map)
  - Weak Collections (WeakSet, WeakMap)
  - structuredClone()
  - slice, shallow copy, deep copy
- 

## Data Handling

- typeof, isArray
  - NaN
  - undefined vs null
  - JSON (Definition, Methods, JSON.parse, JSON.stringify)
- 

## Intermediate JavaScript Topics

### Functions

- Function Declaration & Expression
- Arrow Function vs Regular Function
- IIFE (Immediately Invoked Function Expression)
- Alternative for IIFE
- Pure vs Impure Functions
- First-Class Functions
- Higher-Order Functions
- Callback Functions
- Callback Hell
- Function Currying

- Function Composition
  - Partial Application
  - Function Borrowing (call, apply, bind)
  - Named Parameters
  - Generator Functions
  - Factory Functions
  - Constructor Functions
  - Copy Constructor
  - Types of Constructors
  - Purpose of Constructor
  - Function Returning a Promise Syntax
- 

## Closures & Scoping

- Closures (Applications, Drawbacks)
  - Variable Shadowing / Illegal Shadowing
  - Global Execution Context
  - Execution Context & Call Stack
  - Scoping (Block, Function, Global)
- 

## Objects & Prototypes

- Object Declaration & Literals
- Object Prototype
- Prototype Chain / Prototypal Inheritance
- Classical Inheritance vs Prototypal
- prototype vs \_\_proto\_\_
- Prototype Pollution
- Object.defineProperty()
- Object.freeze() / Deep Freeze()
- Proxy Objects & Traps
- Weak References
- Symbols

- Proxy Use Cases
- 

## **Classes & OOP**

- Classes & Objects
  - Class Constructor, Methods, Subclass
  - extends Keyword
  - Private Properties in Classes
  - Encapsulation
  - Abstraction
  - Inheritance
  - Polymorphism
  - Decorators
- 

## **Asynchronous JavaScript**

- Synchronous vs Asynchronous
- Single Thread vs Multi Thread
- Thread Pool & Process
- Web APIs
- Event Loop
- Microtask Queue
- Event Flow in JS
- Promises
- Promise States
- Promise Methods (then, catch, finally)
- Promise Combinators (all, allSettled, race, any)
- Async/Await
- Convert Callback Hell to Promise
- Function Returning Promise
- Generator Function Practical Use (Yield Example)
- Debouncing
- Memoization

---

## **Events & DOM**

- What is DOM?
- DOM Manipulation Methods
- append, createChild, querySelector
- Add Event Listeners
- Event Propagation
- Event Capturing
- Event Bubbling
- Event Delegation
- Stop Event Bubbling
- Event Flow
- Event Emitter

---

## **Advanced JavaScript Topics**

### **Engine & Runtime**

- JS Engine vs Runtime
- ECMAScript (ES Versions, ES6 Features)
- JIT (Just-In-Time Compilation)
- Garbage Collection (Mark and Sweep)
- Memory Leaks
- Boxing / Unboxing
- REPL (Read Eval Print Loop)

---

### **Advanced Language Features**

- Design Patterns
- Polyfills
- Pipeline Operator (|>)
- eval()
- Getters & Setters
- Object Property Immutability

- Making Objects Immutable
  - Deep Freeze
  - Proxy Traps (Detailed)
  - WeakRef
  - Error Handling & Throwing Errors
  - Debugger Statement
  - Date Formatting & Manipulation
  - Construct Query Params from Object
  - Network Request (Fetch / AJAX)
  - Construct JSON from URL
- 

### **Advanced Functional Programming**

- Currying (Examples)
  - Function Composition
  - Function Chaining
  - Higher-Order Functions
  - Function Currying Practical
- 

### **Advanced Topics in Prototypes & OOP**

- Prototype Pollution
  - Proxy & Reflection API
  - WeakMap Use Cases
  - Inheritance Patterns
  - Symbols in Class Design
  - Object.freeze, seal, defineProperty Advanced
  - Private Fields in Classes
- 

### **Expert / Modern JavaScript Concepts**

#### **Advanced Asynchronous & Performance Concepts**

- Promise.all vs Promise.allSettled
- Promise.any vs Promise.race

- Async/Await Patterns
  - Event Loop Deep Dive
  - Microtask Queue vs Callback Queue
  - Task Scheduling
  - Request Animation Frame
  - Web Workers
  - Rate Limiting (Axios / API)
  - Memoization & Caching
  - JIT Compilation
  - Shadow Realms (*optional advanced topic*)
- 

### **Advanced Design & Patterns**

- Module Pattern
  - Factory Pattern
  - Observer Pattern
  - Singleton Pattern
  - Decorator Pattern
  - Proxy Pattern
- 

### **Misc. Advanced JS Concepts**

- JIT Compiler Internals
  - Typed JavaScript vs TypeScript
  - Dynamically Typed vs Statically Typed
  - Function Pipeline Operator
  - Symbol & Meta-Programming
  - Truthy/Falsy Deep Rules
  - Boxing and Unboxing
  - JSON Handling & Transformation
- 

### **Miscellaneous / Non-JS Topics**

### **Networking & APIs**

- REST APIs
  - PUT vs POST (Idempotency)
  - What is Idempotency
  - Cluster vs Fork
  - TCP vs UDP
  - Rate Limiting
  - API Versioning
  - Axios Interceptors
  - Axios CancelToken
  - Morgan (Request Logger)
- 

## **System & Runtime Internals**

- Thread Pool vs Process
  - Cluster Module (Node.js)
  - Event Emitter (Node Core)
  - Engine vs Runtime (Node vs Browser)
- 

## **Miscellaneous**

- Session vs Cookies
  - Memory Leak
  - Type Assertion
  - Generics (generic keyword, Generic Functions)
  - Time Travel (Debugging Context)
  - Network Request to Fetch JSON
  - Follow JS Coding Conventions
-