**To crack JavaScript interviews (especially for freshers or entry-level roles like MERN stack developer or frontend/backend JS developer), you need to focus on two key areas:**

**🧠 PART 1: JavaScript Logical Problem Solving Topics**

**These are essential to solve coding questions in interviews:**

**1. Data Types & Variables**

* **Primitive vs Reference types**
* **undefined, null, NaN, typeof**

**2. Control Flow**

* **if, else, switch, ternary (? :)**

**3. Loops and Iterations**

* **for, while, do...while**
* **for...in vs for...of**
* **map(), filter(), reduce(), forEach()**

**4. Functions**

* **Function declaration vs expression**
* **Arrow functions**
* **Callback functions**
* **Recursion**

**5. Arrays**

* **Common methods: push, pop, shift, unshift, slice, splice, includes, indexOf, find, sort, reverse**
* **Solving problems like:**
  + **Reverse an array**
  + **Find largest/smallest number**
  + **Remove duplicates**
  + **Merge arrays**
  + **Find frequency of elements**

**6. Strings**

* **String methods: split, join, includes, substring, replace, trim**
* **Problems like:**
  + **Palindrome check**
  + **Anagram check**
  + **Reverse a string**
  + **Count vowels/consonants**
  + **Remove duplicates**

**7. Objects**

* **Key-value pairs**
* **Looping through objects**
* **Comparing objects**
* **Object destructuring**
* **Cloning objects (shallow vs deep copy)**

**8. Math and Logic**

* **Prime numbers**
* **Fibonacci**
* **Factorial**
* **Sorting and searching**
* **Number conversion (binary, decimal)**

**📘 PART 2: Theoretical JavaScript Interview Topics**

**These help in technical discussions and MCQs:**

**1. ES6+ Features**

* **let, const**
* **Arrow functions**
* **Template literals**
* **Spread/rest operator**
* **Destructuring**
* **Default parameters**
* **Modules (import, export)**
* **Optional chaining (?.)**

**2. DOM Manipulation**

* **getElementById, querySelector, etc.**
* **Event listeners (onclick, addEventListener)**
* **Form handling**

**3. Event Loop & Async**

* **Call stack, Web APIs, Callback Queue**
* **setTimeout, setInterval**
* **Promises**
* **async/await**
* **Microtask vs Macrotask**

**4. Hoisting, Scope, Closures**

* **Function and variable hoisting**
* **Block scope vs function scope**
* **Closures use cases**

**5. This Keyword**

* **this in global, object, arrow function, event handler**

**6. Prototype and Inheritance**

* **Prototype chain**
* **\_\_proto\_\_, Object.create**

**7. Error Handling**

* **try, catch, finally**
* **throw**

**8. Memory Management**

* **Garbage collection**
* **Avoiding memory leaks**

**9. Browser APIs & Storage**

* **LocalStorage, SessionStorage, Cookies**
* **Fetch API**

**10. Miscellaneous**

* **Debouncing/Throttling**
* **Polyfills**
* **Type coercion & comparison (== vs ===)**
* **Event Bubbling vs Capturing**
* **Shallow vs Deep Copy**

**✅ Bonus for Interviews**

* **Basic Data Structures: Stack, Queue, HashMap**
* **Coding platforms practice: LeetCode (easy), Codewars, JS Challenges**
* **Mock Interviews: Practice explaining your approach out loud**

**Want a roadmap or weekly plan to cover these step-by-step with practice problems?**