**📅 JavaScript Daily Revision Plan (15 Days)**

**🔹 Day 1 – Basics**

* Variables (var, let, const)
* Data Types
* Operators
* Type Conversion  
  ✅ Practice: 10 chhoti problems (e.g., add numbers, compare strings, type check).

**🔹 Day 2 – Control Flow**

* if / else
* switch
* Loops (for, while, do while)
* for...in vs for...of  
  ✅ Practice: Write program to print even numbers 1–100, multiplication table.

**🔹 Day 3 – Functions**

* Function declaration vs expression
* Arrow functions
* Default/rest parameters
* Scope  
  ✅ Practice: Factorial function, sum of array function.

**🔹 Day 4 – Closures & Higher-Order Functions**

* Closures (examples)
* Higher-Order Functions  
  ✅ Practice: Make a function that returns another function for multiplication.

**🔹 Day 5 – Arrays (Part 1)**

* map, filter, reduce
* find, some, every  
  ✅ Practice: Filter out even numbers, sum all numbers using reduce.

**🔹 Day 6 – Arrays (Part 2) + Objects**

* sort, forEach
* Spread & Rest operators
* Object methods (keys, values, entries)
* Object destructuring  
  ✅ Practice: Sort names alphabetically, extract values from object.

**🔹 Day 7 – DOM Manipulation (Part 1)**

* Selecting elements
* Changing content & styles
* Event listeners (click, input)  
  ✅ Practice: Make a button that changes page background color on click.

**🔹 Day 8 – DOM Manipulation (Part 2)**

* Creating/removing elements
* Event bubbling
* Form handling  
  ✅ Practice: Todo list app (add/remove tasks dynamically).

**🔹 Day 9 – ES6+ Features**

* Template literals
* Destructuring (recap)
* Spread/Rest (recap)
* Modules (import/export)
* Optional chaining, Nullish coalescing  
  ✅ Practice: Rewrite older functions using ES6 features.

**🔹 Day 10 – Async JS (Part 1)**

* Callbacks
* Promises (.then, .catch)  
  ✅ Practice: Simulate async task (setTimeout → resolve after 2s).

**🔹 Day 11 – Async JS (Part 2)**

* async/await
* fetch API (GET request)
* try...catch  
  ✅ Practice: Fetch posts from JSONPlaceholder API.

**🔹 Day 12 – OOP (Part 1)**

* this keyword
* Constructor functions
* Classes  
  ✅ Practice: Create a Car class with properties & method drive().

**🔹 Day 13 – OOP (Part 2)**

* Inheritance (extends, super)
* Encapsulation, Abstraction, Polymorphism (basic level)  
  ✅ Practice: Vehicle base class → Bike & Car subclasses.

**🔹 Day 14 – Advanced JS**

* Execution context & Call stack
* Hoisting
* Closures (revision with real-world use case)
* Event loop, microtasks vs macrotasks  
  ✅ Practice: Predict output of tricky hoisting & async examples.

**🔹 Day 15 – Final Project Day**

Pick 1 small project:

* **Option A**: Calculator
* **Option B**: Weather app (using API + async/await)
* **Option C**: Quiz app