**🔑 Core JavaScript Topics You MUST Know before diving to react**

**1. Basics & Syntax**

* **Variables**: let, const, var (and why you almost never use var)
* **Data types**: string, number, boolean, null, undefined, object, array
* **Operators:** + - \* / %, == vs ===, logical operators (&&, ||, !)
* **Template literals:** `Hello ${name}`

**2. Control Flow**

* if, else if, else
* switch
* Loops: for, while, for...of, for...in

**3. Functions**

* Function declarations vs expressions
* Arrow functions (() => {})
* Parameters, default values, return values
* Scope (global vs local vs block scope)
* Closures (important for React concepts like hooks)

**4. Objects & Arrays**

* Creating and using objects { key: value }
* Array methods: .map(), .filter(), .reduce(), .forEach(), .find()
* Spread/rest operator: ...
* Destructuring: const {name} = obj;, const [a, b] = arr;

**5. DOM Manipulation**

* document.querySelector, getElementById
* Changing text, HTML, styles (.innerText, .innerHTML, .style)
* Event listeners: addEventListener('click', fn)
* Creating/removing elements dynamically

*(Even though React will handle the DOM for you, knowing this helps you understand what React is actually doing behind the scenes.)*

**6. ES6+ Features**

* let & const
* Arrow functions
* Template literals
* Modules (import / export)
* async/await
* Default + named exports

**7. Asynchronous JS**

* setTimeout, setInterval
* Callbacks (know what they are, even if you don’t use them much)
* Promises (.then().catch())
* async/await (super important for fetching APIs in React/Next.js)
* fetch() API (getting data from an external API)

**8. Error Handling**

* try...catch
* Throwing custom errors

**9. Classes & OOP (basic understanding only)**

* Creating classes, constructors
* Methods inside classes
* this keyword

**10. JSON & APIs**

* Converting objects to JSON: JSON.stringify()
* Parsing JSON: JSON.parse()
* Fetching external data and rendering it

✅ If you’re confident in all of these, React will feel **much smoother**.  
✅ TypeScript will also make more sense because you’ll already know how JS works with objects, arrays, and functions.

JavaScript Learning Projects Roadmap

# Beginner Projects (Basics, Control Flow, Functions)

* Calculator (basic): Input two numbers, choose an operation (+ - \* /), and display the result. Practice: variables, operators, functions.
* Guess the Number Game: Computer picks a random number; user tries to guess it. Give hints like 'too high' or 'too low.' Practice: conditionals, loops, functions, Math.random().
* To-Do List (basic): Add and remove tasks in a list. Practice: DOM manipulation, event listeners, arrays.

# Intermediate Projects (Arrays, Objects, DOM, ES6)

* Quiz App: Show multiple-choice questions, track score. Practice: arrays, objects, DOM updates.
* Digital Clock: Shows current time, updates every second. Practice: setInterval, Date object.
* Weather App: Fetch data from a weather API and display current weather. Practice: fetch(), async/await, JSON, DOM.
* Expense Tracker: Add incomes/expenses, calculate totals. Practice: arrays, objects, DOM, .map(), .filter(), .reduce().

# Advanced Projects (Async, APIs, Error Handling)

* Movie Search App: Use an API (like OMDb) to search for movies and display posters/details. Practice: async/await, error handling, DOM updates.
* Notes App with Local Storage: Add, edit, delete notes, and save them even after refresh. Practice: objects, arrays, localStorage.
* Weather + Geolocation App: Detect user’s location with JS, then fetch weather for that location. Practice: async/await, geolocation API, error handling.