

JavaScript Learning Path for React

1. Master ES6+ Syntax

- let, const
- Arrow Functions `()=>{}`
- Template Literals ``Hello ${name}``
- Object and Array Destructuring
- Spread and Rest Operators `{...obj}`, `(...args)`

Why? React code is full of these! It's the new standard.

2. Deepen your Understanding of Functions and Scope

- Function Declarations vs Expressions
- Closures
- Lexical Scope
- Understanding 'this'

Why? React uses a lot of callbacks and function-based logic.

3. Learn Array Methods Like a Pro

- `map()`, `filter()`, `reduce()`
- `forEach()`, `find()`, `some()`, `every()`

Why? In React, you will use `map()` A LOT to render lists.

4. Grasp Promises and Async/Await

- Writing clean asynchronous code
- `fetch()` API calls
- `try...catch` error handling

Why? React apps fetch data from APIs all the time.

5. Understand Event Handling

- How browser events work
- `onclick`, `onchange`, `onsubmit`
- Event object basics

Why? React uses synthetic events based on these.

6. Learn JavaScript Modules

- `import` and `export`

Why? Every React component is a module.

7. Understand Conditional Logic

- Ternary operators condition ? doThis : doThat
- Short-circuit evaluation condition && doThis

Why? JSX in React often uses these inside templates.

8. Practice Object Manipulation

- Adding, updating, deleting object properties

Why? React states are often objects, and updating them immutably is important.

9. Get Comfortable with Basic OOP (Optional but Useful)

- Classes and Constructors
- Methods inside classes

Why? For understanding older React Class Components.

10. Explore Optional Topics (Helpful but not urgent)

- Immutable updates (without mutating arrays/objects)
- Debouncing and Throttling
- Event Loop (to better understand re-renders)
- Browser APIs like localStorage

Bonus Tip:

Once you feel okay with the above, start building small things like a Counter App, Todo App using Vanilla JS, then React will feel super natural!