React.js - How It Works (Beginner Friendly + Interview Notes)

React is a JavaScript library for building user interfaces. To understand React, you need a mental model of how it works under the hood. This helps not only in coding but also in interviews.

1. Component = Function

In React, a component is just a JavaScript function that returns JSX (UI description). Whenever state or props change, React calls the function again to calculate new UI.

2. State with useState

`useState` allows you to declare state inside a component. It returns a state variable and a function to update it. State changes cause re-renders.

3. What happens when you call setState / setCount

React does not update the value immediately. It marks the state as changed and schedules a re-render. During re-render, the component function runs again with the updated value.

4. Batching of state updates

React batches multiple state updates inside the same event for performance. Example: calling setCount(count + 1) three times may only result in count increasing by 1, because all updates saw the same old value. Solution: use functional updates like setCount(prev => prev + 1).

5. React’s lifecycle (mental model)

React has two main phases: Render Phase (calculate new UI) and Commit Phase (apply updates to the DOM).

6. Virtual DOM

React maintains a lightweight copy of the DOM (Virtual DOM). After re-rendering, React compares the new Virtual DOM with the old one and only updates the parts that changed, making UI updates fast and efficient.

7. Example flow

Click button → call setCount → React schedules update → re-renders component with new state → compares Virtual DOM → updates browser DOM.

■ Common React Definitions for Interview

- Component: A reusable piece of UI written as a JavaScript function or class.
- State: Data that changes over time within a component.

- Props: Inputs passed to a component from its parent.
- Virtual DOM: A lightweight JavaScript representation of the real DOM for efficient updates.
- Reconciliation: The process React uses to compare old and new Virtual DOM and update the real DOM.
- Hooks: Special functions (like useState, useEffect) that let you use React features in function components.
- JSX: A syntax extension that lets you write HTML-like code inside JavaScript.