

## 1. Basics

What is React, and why use it?

What is the Virtual DOM?

What is JSX?

Difference between functional and class components?

What is a component in React?

How does one-way data binding work?

## 2. Props & State

What are props?

What is state?

Difference between props and state?

What is state lifting?

What is prop drilling?

How do you avoid prop drilling?

### 3. Hooks (VERY IMPORTANT)

What is a Hook?

Rules of Hooks?

useState vs useReducer?

useEffect use cases?

Why does useEffect run twice in StrictMode?

useMemo vs useCallback?

What is useRef used for?

What is a custom hook?

### 4. Lifecycle (Class & Functional)

What are the phases of component lifecycle?

componentDidMount vs componentDidUpdate vs componentWillUnmount?

useEffect equivalent of lifecycle methods?

## 5. Rendering & Reconciliation

What is reconciliation in React?

Why are keys important?

What causes a re-render?

How to prevent unnecessary re-renders?

## 6. Context API

What is Context?

When to use Context?

Limitations of Context?

Context vs Redux?

## 7. Performance Optimization

What is React.memo?

useMemo vs memo?

What is code splitting?

What is lazy loading in React?

## 8. Routing

What is React Router?

Difference between BrowserRouter and HashRouter?

What is useNavigate?

How do you protect routes?

## 9. Forms

Controlled vs Uncontrolled components?

Why use React Hook Form?

How does debouncing help in forms?

## 10. API & Data Fetching

How to call APIs in React?

Why use Axios over fetch?

What is React Query?

What is optimistic update?

## 11. Redux (if included)

What is Redux?

What are actions, reducers, store?

What is Redux Toolkit?

redux-thunk vs redux-saga?

## 12. Advanced Concepts

What are Portals?

What are Error Boundaries?

What are Higher Order Components?

What are Render Props?

What is Suspense?

What is Concurrent Rendering?

### 13. Webpack / Build Tools

What is tree-shaking?

Difference between dev vs production build?

What does Babel do?