

◇ Core Basics

- What is React? (Library vs Framework)
- React Project Setup (CRA, Vite, Next.js)
- JSX (JavaScript XML)
- Components (Functional & Class components)
- Props (Passing data between components)
- State (useState)
- Rendering lists (`.map()`)
- Conditional rendering (`&&`, ternary, `if`)

◇ Events & Forms

- Handling events (`onClick`, `onChange`, etc.)
- Controlled vs Uncontrolled components
- Forms & Input handling
- Two-way binding

◇ Styling

- Inline styles
- CSS Modules
- Styled Components
- Tailwind / Sass basics

◇ **React Hooks (very important)**

- useState
- useEffect (dependencies, cleanup)
- useContext
- useRef
- useReducer
- useMemo
- useCallback
- Custom Hooks

◇ **Component Communication**

- Parent to Child (props)
- Child to Parent (callback props)
- Sibling communication (lifting state up)
- Context API (global state sharing)

◇ **Routing**

- React Router (v6+)
 - BrowserRouter, Routes, Route
 - Link, NavLink
 - useNavigate, useParams, useLocation
 - Nested routes
 - Protected routes

◇ **State Management**

- Local state with hooks

- Context API
- Redux Toolkit
- Zustand, Jotai, Recoil (alternatives)

◇ Data Fetching & APIs

- Fetch API / Axios
- useEffect for fetching
- Async/await in React
- React Query / TanStack Query (caching, pagination)

◇ Performance Optimization

- React.memo
- useMemo, useCallback
- Code splitting (lazy & Suspense)
- Virtualization (react-window, react-virtualized)

◇ Advanced Concepts

- Higher Order Components (HOC)
- Render Props
- Error Boundaries
- Portals
- Refs & Forward Refs
- Controlled vs uncontrolled inputs
- Context vs Redux
- Immutable state updates

◇ Testing

- Unit testing with Jest
- React Testing Library
- Mocking API calls
- Snapshot testing

◇ Deployment

- Build & deploy (Vercel, Netlify, GitHub Pages)
- Environment variables
- Production optimization

◇ Modern React (must-know)

- Server Components (React 18+ / Next.js 13+)
- Suspense & Concurrent rendering
- Hooks-only approach (no class components in new projects)
- Next.js basics (if going full-stack React)

✅ If you're preparing for **interviews**, focus more on:

- Props vs State
- useState, useEffect, useContext
- Lifecycle (old: componentDidMount, new: useEffect)
- React Router basics
- Redux Toolkit
- Performance optimization (memo, useCallback, useMemo)
- Event handling & forms