

⬅️ Swipe For More !



# 100 React JS Interview Questions





save for later



1. What is React JS?
2. Who developed React JS and why?
3. What is a React Component?
4. Differentiate between Class-based and Functional components in React.
5. What is JSX?
6. What are the lifecycle methods in React?
7. Explain the significance of the 'key' prop in React.
8. What are Props in React?
9. How is state different from props?
10. What is the Virtual DOM?
11. How does React handle events?
12. Explain how 'setState' works.
13. What is the significance of this in React components?
14. How do you create a form in React?
15. What are controlled components?
16. How do you conditionally render components?
17. What is a higher-order component?
18. Explain the purpose of 'render' in a class component.
19. What is a PureComponent?
20. What are stateless components?
21. What is Redux?
22. How is Redux different from Context API?
23. What are the three principles of Redux?



24. What is the significance of `store` in Redux?
25. Explain the role of reducers in Redux.
26. What are actions in Redux?
27. What is a thunk in the context of Redux?
28. How do you handle asynchronous actions in Redux?
29. What is React Router?
30. How do you implement navigation in React applications?
31. What are the different types of components that you can render using React Router?
32. What is a Fragment in React?
33. How do you optimize a large React application?
34. What are Hooks in React?
35. Explain the `useEffect` Hook.
36. What is the Context API?
37. How can you manage state locally with the `useState` Hook?
38. How do you handle side-effects in functional components?
39. What are the rules of Hooks?
40. How can you optimize components with `useMemo` and `useCallback`? 
41. Explain the concept of React Fiber.
42. What is the significance of `shouldComponentUpdate` lifecycle method?
43. Explain how React's reconciliation algorithm works.
44. How to handle errors in React?

45. How to implement server-side rendering (SSR) in React?
46. Explain the concept of Suspense in React.
47. What is React.memo, and when should you use it?
48. How do you optimize re-renders for a list of items in React?
49. Explain how React Portal works.
50. What are the best practices for managing state in large-scale React applications?
51. How do you implement internationalization (i18n) in a React application?
52. Explain how React Lazy Loading works.
53. How do you handle state in a component tree?
54. What is Prop Drilling, and how do you avoid it?
55. Explain how the `useReducer` Hook works.
56. How can you test React components?
57. Explain the importance of code splitting in React.
58. What are forward refs in React?
59. What is the significance of `componentDidCatch` lifecycle method? 
60. What are the best practices to optimize component renders in React?
61. How to implement a virtualized list in React?
62. What are the design patterns in React?
63. How to design high-order components (HOCs)?
64. How to integrate GraphQL with React?



65. How to implement offline capabilities in a React application?
66. Explain in-depth how React's Context API works.
67. How to optimize a React application for SEO?
68. How do you handle complex state logic in React?
69. How to ensure accessibility (a11y) in a React application?
70. How to create customizable and reusable components?
71. What are the different state management libraries available for React, and how do they differ?
72. Explain the differences and similarities between React and React Native.
73. How do you implement animations in React?
74. How to handle CSS in JS with React?
75. How can you make a React application more secure?
76. How to optimize the initial load time of a React application?
77. How to implement lazy loading for images in React?
78. Explain the role of render props in React.
79. How to do prop types validation in React?
80. How to implement infinite scrolling in a React application? 📱
81. How to design a scalable and maintainable React project structure?
82. How to manage side-effects in a large-scale React application?
83. How to implement a custom Hook in React?
84. Explain how React interacts with Web Workers.

85. How to ensure type safety in a React application using TypeScript?
86. How to implement and manage themes in a React application?
87. How to handle real-time data with React?
88. How to handle complex user interactions and state transitions in React?
89. Explain how React batches updates.
90. What are the advanced optimization techniques in React?
91. How to implement a fully customizable design system in React?
92. How to integrate WebAssembly with React?
93. How to implement a rich text editor in React?
94. How to manage global state without Redux or Context API in React?
95. How to implement responsive design in React without using libraries?
96. How to architect a React application for multiple platforms?
97. How to improve the performance of a React application using Web Workers? 
98. How to optimize the performance of large forms in React?
99. How to implement multi-language support in a React application with optimal bundle size?
100. How to design a decoupled, testable, and maintainable React application?

**i post about Tech,  
Coding and Career**

Follow for more content like this

@alighouri