

100 React Interview Questions

1. React Basics & Conceptual Questions (1–20)

1. What is React?
 2. What are the main features of React?
 3. Why use React over frameworks like Angular or Vue?
 4. What is JSX and why is it used?
 5. Difference between JSX and HTML.
 6. What is the Virtual DOM and how does it work?
 7. Difference between real DOM and virtual DOM.
 8. Difference between a React component and a React element.
 9. What are functional components?
 10. What are class components?
 11. Advantages of functional components over class components.
 12. Difference between controlled and uncontrolled components.
 13. Difference between props and state.
 14. Difference between stateful and stateless components.
 15. Difference between local state and global state.
 16. Why is React called a “declarative” library?
 17. Difference between one-way binding and two-way binding in React.
 18. What are keys in React and why are they important?
 19. Difference between React and React Native.
 20. Difference between single-page applications (SPA) and multi-page applications (MPA).
-

2. Props (21–35)

21. What are props in React?
22. Can props be modified inside a component?

23. How do you pass props to child components?
 24. Difference between default props and required props.
 25. How can props be validated in React?
 26. Difference between props and children.
 27. How can functions be passed as props?
 28. Difference between passing props as objects vs individual props.
 29. Difference between props in class components and functional components.
 30. Difference between read-only props and mutable state.
 31. Difference between state and props in terms of re-rendering.
 32. Difference between shallow and deep prop comparison.
 33. Difference between props drilling and context API.
 34. Difference between props and context.
 35. Can props contain functions that modify the parent state?
-

3. State (36–50)

36. What is state in React?
37. Difference between useState in functional components and this.state in class components.
38. Difference between state and props in terms of scope.
39. Difference between mutable and immutable state updates.
40. Difference between updating state synchronously vs asynchronously.
41. Difference between local state and derived state.
42. Difference between simple state and complex nested state.
43. Difference between state in functional components vs class components.
44. Difference between multiple state variables vs a single object state.
45. Difference between state and ref.
46. Difference between shallow and deep updates in state.
47. Difference between state lifting vs context API.

- 48. Difference between controlled state and uncontrolled DOM state.
 - 49. Difference between updating state directly vs using setter functions.
 - 50. Difference between useReducer and useState for complex state.
-

4. Hooks (51–70)

- 51. What are hooks in React?
 - 52. Why were hooks introduced?
 - 53. Difference between class components and functional components with hooks.
 - 54. What is useState and when to use it?
 - 55. Difference between multiple useState calls vs a single state object.
 - 56. What is useEffect and when is it used?
 - 57. Difference between useEffect with empty dependency, no dependency, and specific dependency.
 - 58. Difference between useEffect and lifecycle methods like componentDidMount/componentDidUpdate.
 - 59. How does useEffect cleanup work?
 - 60. Difference between useEffect and useLayoutEffect.
 - 61. What is useRef and when is it used?
 - 62. Difference between useRef and useState.
 - 63. Difference between useRef for DOM access vs storing mutable values.
 - 64. What is useContext and how is it used?
 - 65. Difference between useContext and prop drilling.
 - 66. How to create a custom hook and why would you use one?
 - 67. Difference between custom hooks and built-in hooks.
 - 68. Difference between using a custom hook vs repeating logic in components.
 - 69. What is useNavigate and when is it used?
 - 70. Difference between programmatic navigation using useNavigate vs Link component.
-

5. Event Handling (71–80)

- 71. How do you handle events in React?
 - 72. Difference between React events and HTML events.
 - 73. How does React handle event binding in class components?
 - 74. How do arrow functions affect event handlers?
 - 75. Difference between inline event handlers and bound functions.
 - 76. Difference between passing arguments directly vs via arrow functions in event handlers.
 - 77. Difference between synthetic events and native events.
 - 78. How to prevent default behavior in React events.
 - 79. How to stop event propagation in React.
 - 80. Difference between bubbling and capturing in React events.
-

6. Conditional Rendering & Lists (81–90)

- 81. Difference between if-else, ternary operator, and logical && in rendering.
 - 82. Difference between rendering null and undefined.
 - 83. Difference between rendering false vs null.
 - 84. Difference between keys in static vs dynamic lists.
 - 85. Difference between controlled and uncontrolled list rendering.
 - 86. Difference between map() and forEach() in rendering lists.
 - 87. Difference between conditional rendering in class vs functional components.
 - 88. Difference between React.Fragment and div wrapper.
 - 89. Difference between short-circuit rendering and ternary rendering.
 - 90. Difference between rendering components directly vs storing in a variable first.
-

7. Routing & Navigation (91–100)

- 91. What is React Router?
- 92. Difference between BrowserRouter, HashRouter, and MemoryRouter.

- 93. Difference between `<Link>` and `<Tag>` in navigation.
 - 94. Difference between nested routes and dynamic routes.
 - 95. Difference between `redirect` and `navigate`.
 - 96. Difference between `useParams` and query strings.
 - 97. Difference between programmatic navigation and declarative navigation.
 - 98. Difference between client-side routing and server-side routing.
 - 99. Difference between React Router v5 and v6 (basic understanding).
 - 100. Difference between route-level and component-level rendering.
-