108 React interview questions to hire top developers

Questions

1. What is React, in simple terms?

2. Imagine you're building with LEGOs. How is React like using LEGOs to build a big structure?

3. What are components in React? Think of them as building blocks.

4. What is JSX? Why do we use it in React?

5. Can you explain what 'props' are in React, like giving toys to a component?

6. What is 'state' in React? How is it different from 'props'?

7. What does it mean for a component to be 're-rendered'?

8. Why is it important to keep React components small and focused?

9. What is the virtual DOM? How does React use it to update the real DOM?

10. How do you handle events in React, like when someone clicks a button?

11. What is the purpose of using 'key' props when rendering lists of items in React?

12. Describe a situation where you might need to use 'state' in a component.

13. What is the difference between a class component and a functional component in React?

14. What are React Hooks? Can you name a few common ones?

15. How can you conditionally render content in React? (Show or hide things)

16. Explain the concept of 'unidirectional data flow' in React.

17. What are some advantages of using React over plain JavaScript for building UIs?

18. How do you pass data from a parent component to a child component?

19. What is the purpose of the `useEffect` Hook? Can you give a simple example?

20. How do you update the state of a component? Why can't you directly modify it?

21. What is component composition in React? Why is it useful?

22. How can you prevent a component from re-rendering unnecessarily?

23. What is the role of the ReactDOM library?

24. Describe the process of building a simple form in React. What components and techniques would you use?

25. What are some common tools or libraries used with React for development?

26. How would you handle making an API call from a React component?

27. What is the significance of the `render()` method in a class component?

28. Explain how you would approach debugging a React application if something isn't working as expected.

29. What is JSX?

30. What are React components?

31. What is the difference between state and props?

32. Explain the concept of virtual DOM.

33. What is the purpose of the `render()` method in React?

34. How do you handle events in React?

35. What are controlled and uncontrolled components?

36. What is the significance of keys in React lists?

37. How can you conditionally render content in React?

38. What are React hooks? Name a few common ones.

39. Explain the purpose of `useState` hook.

40. What does the `useEffect` hook do?

41. How can you pass data from a parent component to a child component?

42. How can a child component communicate with its parent?

43. What are some ways to style React components?

44. What is component composition in React?

45. What are fragments and why are they useful?

46. Explain the purpose of prop drilling and how to avoid it.

47. What are some common React lifecycle methods (for class components)?

48. What is the difference between a class component and a functional component?

49. What are higher-order components (HOCs)?

50. What is the context API in React?

51. How do you handle forms in React?

52. What is the purpose of using `refs` in React?

53. What are some advantages of using React?

54. Describe the flow of data in React.

55. Explain the importance of immutability in React state.

56. How can you optimize React component rendering?

57. What are some tools you use for debugging React applications?

58. What is `create-react-app`?

59. Explain the concept of render props in React and provide a use case.

60. Describe the difference between controlled and uncontrolled components.

61. How can you optimize performance in a React application dealing with frequent updates?

62. What are React Fragments and why are they useful?

63. Explain the purpose of React context and how it can be used for state management.

64. Describe the process of creating and using custom hooks in React.

65. How do you handle errors in React components, including error boundaries?

66. What is the significance of keys in React lists and what happens if they are not unique?

67. Explain the concept of code splitting in React and how it improves performance.

68. How does React's `memo` function work, and when should you use it?

69. Describe the differences between `useMemo` and `useCallback` hooks.

70. How would you implement a debouncing or throttling function in a React component?

71. Explain how to use React Portals for rendering content outside the DOM hierarchy.

72. What is the purpose of the `forwardRef` API in React?

73. Discuss different strategies for handling forms in React, including validation.

74. How can you implement server-side rendering (SSR) with React?

75. Explain the benefits of using TypeScript with React.

76. How do you test React components, including unit and integration tests?

77. Describe the process of setting up and using a linter like ESLint with React.

78. What are the common anti-patterns to avoid when working with React?

79. Explain how to use the `useReducer` hook for more complex state management.

80. How do you implement animation in React, and what are some libraries that can help?

81. Describe the process of migrating a legacy JavaScript codebase to React.

82. How can you create a responsive design in React that adapts to different screen sizes?

83. Explain how to handle authentication and authorization in a React application.

84. How does React's reconciliation process differ from traditional DOM manipulation, and what are its performance implications?

85. Describe your experience with different state management solutions in React, such as Redux, Zustand, or Context API, and explain their trade-offs.

86. Explain the concept of Higher-Order Components (HOCs) in React and provide examples of when you would use them.

87. How do you optimize React applications for performance, considering factors like code splitting, memoization, and virtualized lists?

88. What are React Hooks, and how do they simplify component logic compared to class components?

89. Describe your approach to testing React components, including unit, integration, and end-to-end testing strategies.

90. Explain the difference between controlled and uncontrolled components in React, and discuss their respective use cases.

91. How does Server-Side Rendering (SSR) with React work, and what are its benefits and drawbacks?

92. Describe your experience with accessibility (a11y) in React applications and the techniques you use to ensure compliance.

93. How do you handle asynchronous operations and data fetching in React components, considering different approaches like async/await and Promises?

94. Explain the concept of React Context and how it can be used to share data between components without prop drilling.

95. How do you approach debugging React applications, and what tools or techniques do you find most helpful?

96. Describe your experience with different React component libraries like Material UI, Ant Design, or Chakra UI, and explain your preference.

97. How do you manage and optimize React application's bundle size, considering techniques like tree shaking and dynamic imports?

98. Explain the role of Webpack or other module bundlers in React development and how they contribute to the build process.

99. How do you handle forms in React, including validation, submission, and error handling?

100. Describe your experience with React Native for building mobile applications and the challenges you encountered.

101. Explain the concept of code splitting in React and its benefits for improving application performance.

102. How do you ensure that React components are reusable and maintainable in a large codebase?

103. Describe your experience with different testing frameworks like Jest, Mocha, or Cypress, and explain your preference.

104. How would you approach styling React components, considering options like CSS Modules, Styled Components, or Emotion?

105. Explain the concept of memoization in React and how it can be used to optimize component rendering.

106. How do you handle different environments (e.g., development, staging, production) in React applications, and how do you configure environment variables?

107. Describe your experience with CI/CD pipelines for React applications and the tools you use to automate deployment.