



REACTJS QUESTIONS

TOP-100

BASIC TO ADVANCE ONE-LINER QUESTIONS



Certainly! Here are 50 important questions and answers for React:

1. **What is React?**

React is a JavaScript library for building user interfaces.

2. **What is JSX?**

JSX is a syntax extension for JavaScript used in React to write HTML-like code within JavaScript.

3. **Explain Virtual DOM.**

Virtual DOM is a lightweight copy of the actual DOM, used by React to improve performance by minimizing actual DOM updates.

4. **What's the purpose of `render()` in React?**

The `render()` method returns a React element that represents the UI for a component.

5. **How can you create a component in React?**

You can create a component using either classes or functions.

6. **What is a state in React?**

State is an object that holds data which may change over time and influences the rendering of a component.

7. **What is a prop?**

Props (short for properties) are inputs to a React component that are passed down from its parent.

8. **Explain the concept of hooks in React.**

Hooks are functions that allow you to "hook into" React state and lifecycle features in function components.

9. **What is the purpose of `useState()` hook?**

`useState()` allows you to add state to functional components.

10. **What is the significance of `useEffect()` hook?**

`useEffect()` is used to manage side effects in functional components, like data fetching and DOM manipulation.

11. **What are controlled components in React?**

Controlled components are form elements whose values are controlled by React using state.

12. **What is the difference between `props` and `state`?**

`props` are read-only and passed from parent to child, while `state` is mutable and managed within a component.

13. **Explain the purpose of `componentDidMount()` lifecycle method.**

`componentDidMount()` is called after a component is rendered and added to the DOM, suitable for fetching data.

14. **What is the purpose of `componentWillUnmount()` lifecycle method?**

`componentWillUnmount()` is called before a component is removed from the DOM, used for cleanup.

15. **What is Redux?**

Redux is a state management library for managing the state of large-scale applications.

React Router hooks like `useHistory` and `useLocation` provide access to router-related functionalities in functional components.

16. **What are reducers in Redux?**

Reducers are functions that specify how the state changes in response to dispatched actions.

17. **What is the Redux store?**

The Redux store holds the state tree of your application.

18. **What is the role of the `connect()` function in React-Redux?**

`connect()` connects a React component to the Redux store, enabling it to access state and dispatch actions.

19. **What is the purpose of the `dispatch()` method in Redux?**

`dispatch()` is used to send actions to the Redux store to trigger state changes.

20. **What is the context API in React?**

The context API provides a way to share data between components without explicitly passing props through intermediate components.

21. **What is React Router?**

React Router is a library for adding routing to React applications.

22. **What is lazy loading in React?**

Lazy loading involves loading only the necessary parts of your application when they are actually needed.

23. **What is server-side rendering (SSR) in React?**

SSR involves rendering React components on the server and sending the HTML to the client, improving initial load time and SEO.

24. **Explain PropTypes in React.**

PropTypes are used to specify the type of data a component's props should receive.

25. **What is the purpose of key prop in React?**

The `key` prop is used to give React a hint which elements in a list should be updated, added, or removed.

26. **What is React Developer Tools?**

React Developer Tools is a browser extension that helps you inspect React component hierarchies and state.

27. **What is the role of Fragments in React?**

Fragments allow you to group multiple children without introducing extra nodes to the DOM.

28. **What is a higher-order component (HOC) in React?**

HOCs are functions that take a component and return a new component with enhanced features.

29. **What is the use of `defaultProps` in React?**

`defaultProps` is used to set default values for props when they are not provided.

30. **What are hooks in React Router?**

React Router hooks like `useHistory` and `useLocation` provide access to router-related functionalities in functional components.

31. ****Explain memoization in React.****

Memoization is an optimization technique used to cache expensive function calls, improving performance.

32. ****What are controlled and uncontrolled components in forms?****

Controlled components have their values controlled by React, while uncontrolled components manage their own values.

33.

****What is Redux Thunk?****

Redux Thunk is a middleware that allows you to write action creators that return functions instead of plain actions.

34. ****What is the use of `mapStateToProps` in React-Redux?****

`mapStateToProps` is a function used to map parts of the Redux state to the props of a React component.

35. ****What is the use of `mapDispatchToProps` in React-Redux?****

`mapDispatchToProps` is a function used to map Redux actions to the props of a React component.

36. ****What is React Portals?****

Portals allow you to render components outside the parent DOM hierarchy.

37. ****What are hooks in React Native?****

React Native hooks like `useEffect` and `useState` offer the same functionality as in React.

38. ****What is the purpose of the `shouldComponentUpdate()` method?****

`shouldComponentUpdate()` allows you to control if a component should re-render based on changes in its props or state.

39. ****What is the use of `memo` in React?****

The `memo` function is used to memorize the output of a component and avoid unnecessary renders.

40. ****What are the lifecycle methods in a class component?****

Lifecycle methods are methods that are automatically called at different stages of a component's life cycle.

41. ****What is Redux Saga?****

Redux Saga is a middleware library for handling side effects in Redux applications.

42. ****What is Redux Selector?****

A Redux Selector is a function that extracts a specific piece of state from the Redux store.

43. ****What are the differences between `localStorage`, `sessionStorage`, and `cookies` in React?****

`localStorage` and `sessionStorage` store data on the client-side, while `cookies` are also sent to the server with every request.

44. ****What is the React strict mode?****

React strict mode is a development mode that helps identify potential problems in your application.

45. ****What is the purpose of React Fiber?****

React Fiber is a reimplementation of the core algorithm that powers React's rendering process, improving performance and enabling new features.

46. ****What is the purpose of the ``getDerivedStateFromProps()`` method?****

``getDerivedStateFromProps()`` is used to update state based on changes in props before rendering.

47. ****Explain the concept of unidirectional data flow in React.****

Unidirectional data flow means that data in a React application flows in a single direction, from parent to child.

48. ****What is Babel in React?****

Babel is a JavaScript compiler that helps you use the latest ECMAScript features in your code.

49. ****What is the React ``Suspense`` component?****

The ``Suspense`` component lets you specify the fallback content to display while waiting for components to load asynchronously.

50. ****How do you optimize performance in React applications?****

Performance optimization involves using techniques like code splitting, lazy loading, memoization, and minimizing renders to enhance app speed and responsiveness.

51. ****What is React Router DOM?****

React Router DOM is a collection of navigational components that provide routing functionality for web applications.

52. ****What is the purpose of the ``exact`` prop in React Router?**

The ``exact`` prop ensures that a route matches exactly, preventing partial matches.

53. ****What is the React ``forwardRef`` function?**

``forwardRef`` allows you to pass a ref to a child component, facilitating direct access to the child's DOM node.

54. ****What is the significance of the ``key`` prop in lists?**

The ``key`` prop helps React identify each element in a list, enabling efficient updates and reordering.

55. ****Explain the use of the ``useCallback()`` hook in React.**

``useCallback()`` is used to memoize functions, preventing unnecessary re-creation of functions on each render.

56. ****What is the React DevTools Profiler?**

The Profiler is a tool that helps identify performance bottlenecks by measuring component render times.

57. ****What are higher-order components (HOCs) and their benefits?**

HOCs are functions that enhance components by providing extra props or behavior, promoting code reusability.

58. ****What is the React ``ContextType`` property?**

``ContextType`` is a property used in class components to consume context values without using the ``Consumer`` component.

59. ****What is the purpose of the ``React.memo()`` ?**

``React.memo()`` is a higher-order component that memoizes a functional component to prevent unnecessary renders.

60. ****Explain the use of the ``useMemo()`` hook in React.**

``useMemo()`` memoizes the result of a computation, optimizing performance by avoiding repeated computations.

61. ****What is the `useReducer()` hook in React?****

`useReducer()` is an alternative to `useState()` that manages state changes through a reducer function.

62. ****What is the React `ErrorBoundary` component?****

An `ErrorBoundary` catches errors in components during rendering and allows graceful error handling.

63. ****What is the significance of `shouldComponentUpdate()` in functional components?**

`React.memo()` replaces `shouldComponentUpdate()` for functional components by automatically handling props comparison.

64. ****What are CSS-in-JS libraries, and why might you use them in React?**

CSS-in-JS libraries allow you to write CSS styles directly in your JavaScript code, providing scoped styles and better performance.

65. ****Explain the `useLayoutEffect()` hook in React.**

Similar to `useEffect()`, `useLayoutEffect()` runs after render but before the browser repaints, making it suitable for DOM measurements.

66. ****What is the React `StrictMode`?**

React `StrictMode` is a tool that helps identify and fix problematic code by highlighting potential issues during development.

67. ****What are React Hooks Rules of Use?**

Hooks should only be called at the top level, not inside loops, conditions, or nested functions.

68. ****What is the `default` export in React modules?**

The `default` export allows you to import a module without using curly braces.

69. ****What is the React `Suspense` component used for?**

The `Suspense` component lets you delay rendering of components until certain conditions are met, such as data fetching completion.

70. ****What is the significance of the `useDebugValue()` hook?**

`useDebugValue()` is used to display custom labels for custom hooks in React DevTools.

71. ****Explain the concept of hooks rules dependency array.**

The dependency array in hooks ensures that the effect runs when the values in the array change.

72. ****What is the purpose of `React.lazy()`?**

`React.lazy()` allows you to dynamically import components, enabling code splitting and better performance.

73. ****What is the React `Profiler` API used for?**

The `Profiler` API is used to measure the render time of a component and its descendants.

74. ****What are the performance optimizations in React apps?**

Optimizations include memoization, code splitting, lazy loading, and reducing renders to enhance app speed.

75. ****What is the role of `shouldComponentUpdate()` in class components?**

`shouldComponentUpdate()` allows you to prevent unnecessary renders by comparing current and next props or state.

76. **What is the React `Router` component?**

The `Router` component provides a wrapper for routing elements in React applications.

77. **What is the purpose of the `useContext()` hook?**

`useContext()` lets you consume a context in functional components without using the `Consumer`.

78. **What is the React `PureComponent` class?**

`PureComponent` is a class component that automatically implements a `shouldComponentUpdate()` check for shallow props and state comparisons.

79. **Explain the concept of code splitting in React.**

Code splitting involves splitting your

bundle into smaller chunks to load only the necessary parts when needed.

80. **What is the significance of the `displayName` property in React components?**

The `displayName` property allows you to assign a custom name to a component that will be shown in React DevTools.

81. **What are React Hooks Rules of Use?**

Hooks should only be called at the top level, not inside loops, conditions, or nested functions.

82. **What is the `default` export in React modules?**

The `default` export allows you to import a module without using curly braces.

83. **What is the React `Suspense` component used for?**

The `Suspense` component lets you delay rendering of components until certain conditions are met, such as data fetching completion.

84. **What is the significance of the `useDebugValue()` hook?**

`useDebugValue()` is used to display custom labels for custom hooks in React DevTools.

85. **Explain the concept of hooks rules dependency array.**

The dependency array in hooks ensures that the effect runs when the values in the array change.

86. **What is the purpose of `React.lazy()`?**

`React.lazy()` allows you to dynamically import components, enabling code splitting and better performance.

87. **What is the React `Profiler` API used for?**

The `Profiler` API is used to measure the render time of a component and its descendants.

88. **What are the performance optimizations in React apps?**

Optimizations include memoization, code splitting, lazy loading, and reducing renders to enhance app speed.

89. **What is the role of `shouldComponentUpdate()` in class components?**

`shouldComponentUpdate()` allows you to prevent unnecessary renders by comparing current and next props or state.

90. **What is the React `Router` component?**

The `Router` component provides a wrapper for routing elements in React applications.

91. ****What is the purpose of the `useContext()` hook?****

`useContext()` lets you consume a context in functional components without using the `Consumer`.

92. ****What is the React `PureComponent` class?****

`PureComponent` is a class component that automatically implements a `shouldComponentUpdate()` check for shallow props and state comparisons.

93. ****Explain the concept of code splitting in React.****

Code splitting involves splitting your bundle into smaller chunks to load only the necessary parts when needed.

94. ****What is the significance of the `displayName` property in React components?****

The `displayName` property allows you to assign a custom name to a component that will be shown in React DevTools.

95. ****What are React Hooks Rules of Use?****

Hooks should only be called at the top level, not inside loops, conditions, or nested functions.

96. ****What is the `default` export in React modules?****

The `default` export allows you to import a module without using curly braces.

97. ****What is the React `Suspense` component used for?****

The `Suspense` component lets you delay rendering of components until certain conditions are met, such as data fetching completion.

98. ****What is the significance of the `useDebugValue()` hook?****

`useDebugValue()` is used to display custom labels for custom hooks in React DevTools.

99. ****Explain the concept of hooks rules dependency array.****

The dependency array in hooks ensures that the effect runs when the values in the array change.

100. ****What is the purpose of `React.lazy()` ?**

`React.lazy()` allows you to dynamically import components, enabling code splitting and better performance.

