HTML

CSS

JS

TECH BY WEBCODER

React.Js

Question

Question

1. What is React.js?

➡React.js is a JavaScript library used for building user interfaces. It allows developers to create reusable UI components and efficiently update the UI when the data changes.

2. What are the key features of React.js?

➡React.js offers features like virtual DOM, component-based architecture, one-way data flow, JSX syntax, and a rich ecosystem of libraries and tools.

3. What is JSX?

➡ JSX (JavaScript XML) is an extension to JavaScript that allows you to write HTML-like syntax within JavaScript code. It is used to describe the structure and appearance of React components.

4. What is the significance of the virtual DOM in React.js?

➡The virtual DOM is a lightweight copy of the actual DOM. React uses the virtual DOM to optimize and speed up the process of updating the real DOM by comparing the current virtual DOM with the previous one.

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

➡Functional components are stateless and are typically written as plain JavaScript functions. They are simpler and easier to test. Class components, on the other hand, have a state, can use lifecycle methods, and are written as ES6 classes.

6. What is the React.StrictMode?

➡ React.StrictMode is a component that helps highlight potential problems in an application. It enables additional checks and warnings in the development mode to help identify and address potential bugs and deprecated features.

7.What is the purpose of the React.Fragment component?

➡React.Fragment is a built-in component in React that allows you to group multiple elements without adding an extra node to the DOM. It is useful when you need to return multiple elements from a component's render method without introducing unnecessary wrapping elements.

8. What is the significance of the "key" attribute when rendering an array of components?

➡The "key" attribute is used to give a unique identity to each element in an array of components. It helps React efficiently update and re-render the components by identifying which items have changed, been added, or removed.

9. What is the useReducer hook in React?

➡The useReducer hook is a built-in hook in React that allows you to manage state using a reducer function. It is an alternative to useState and is useful for managing more complex state logic or state transitions.

10. What is the purpose of the useContext hook in React?

➡The useContext hook is used to consume values from a React context. It allows functional components to access context values without nesting multiple layers of components or using render props.

11.What is the purpose of the useCallback() hook?

➡The useCallback() hook is used to memoize functions in functional components. It returns a memoized version of the callback function that only changes if one of the dependencies has changed. It is useful for optimizing performance in scenarios where functions are passed as props.

12. What is the purpose of the useMemo() hook?

➡ The useMemo() hook is used to memoize values in functional components. It allows you to cache the result of an expensive computation and only recalculate it when the dependencies have changed. It is useful for optimizing performance in scenarios where calculations are computationally expensive.

13. What is the purpose of the useReducer() hook?

➡ The useReducer() hook is used to manage state in functional components using the reducer pattern. It is an alternative to useState() and is suitable for managing more complex state or state transitions. It returns the current state and a dispatch function to update the state.

14. What is the purpose of the useRef() hook?

➡The useRef() hook is used to create a mutable reference that persists across component renders. It returns a mutable object with a current property that can be used to store values or reference DOM nodes or other React elements.

15.What is the purpose of the useLayoutEffect() hook?

➡The useLayoutEffect() hook is similar to useEffect(), but it runs synchronously after all DOM mutations. It is useful when you need to perform operations that require access to the DOM immediately after React has performed all updates.

16.What is the purpose of the useEffect() hook in React?

➡ The useEffect() hook is used to perform side effects in functional components. It allows you to run code after the component has rendered and handle actions such as data fetching, subscriptions, or manually updating the DOM. The effect runs after every render unless dependencies are specified.

17. What is the purpose of the useContext() hook in React

➡The useContext() hook is used to consume values from a React context. It allows functional components to access context values without nesting multiple layers of components or using render props.

18.What is the purpose of the useReducer() hook in React

➡The useReducer() hook is used to manage state in functional components using the reducer pattern. It is an alternative to useState() and is suitable for managing more complex state or state transitions. It returns the current state and a dispatch function to update the state.

19. What is the purpose of the useRef() hook in React?

➡ The useRef() hook is used to create a mutable reference that persists across component renders. It returns a mutable object with a current property that can be used to store values or reference DOM nodes or other React elements

20. What is the purpose of the useMemo() hook in React?

➡The useMemo() hook is used to memoize values in functional components. It allows you to cache the result of an expensive computation and only recalculate it when the dependencies have changed. It is useful for optimizing performance in scenarios where calculations are computationally expensive.

21. What is the purpose of the forwardRef() function?

➡ The forwardRef() function is used to forward a ref from a higher-order component (HOC) to a wrapped component. It allows the wrapped component to receive a ref directly and access the underlying DOM node or React component.

22. What is the purpose of React.lazy() and Suspense in React?

➡ React.lazy() is a function that allows you to lazily load a component, which means the component is loaded only when it is actually needed. Suspense is a component that enables displaying fallback content while the lazy-loaded component is loading.

23. What is the purpose of the useImperativeHandle() hook?

➡ The useImperativeHandle() hook allows a functional component toexpose specific functions or values to the parent component through a ref. It is useful when you want to provide a more imperative API for a component that is primarily written as a functional component

24. What is the purpose of the useLayoutEffect() hook?

➡ The useLayoutEffect() hook is similar to useEffect(), but it runs synchronously after all DOM mutations. It is useful when you need to perform operations that require access to the DOM immediately after React has performed all updates.

25. What is the purpose of the useDebugValue() hook?

➡The useDebugValue() hook is used to display a custom label for custom hooks in React DevTools. It helps to provide more meaningful names for custom hooks when debugging and inspecting the component hierarchy.

26. What is the purpose of the constructor in a React component?

➡ The constructor is used to initialize the state and bind event handlers in a class component. It is called before the component is mounted.

27. What is state in React.js?

➡State is an object that holds data and determines how a component renders and behaves. It is private and fully controlled by

the component itself. **28. What is the difference between state and props in React.js?** ➡A controlled component is a component where the form data is
handled by React components. The React component that renders the form also controls what happens in that form on subsequent user input .

29. What is a controlled component?

➡The "import" keyword in Java is used to import classes, interfaces, or packages into a source file. It allows using classes from other packages without specifying their fully qualified names.

30. What are React lifecycle methods?

➡React lifecycle methods are special methods that are called at specific points in a component's lifecycle. These methods include

componentDidMount, componentDidUpdate, componentWillUnmount, and many others.

31. What is the purpose of the "Character" class in Java?

➡The "Character" class in Java provides methods for working with individual characters, such as checking for character types (letters, digits, whitespace), converting case, and performing character-based operations.

32. What is the purpose of the "Integer" class in Java?

➡ The "Integer" class in Java is a wrapper class that provides methods for working with integer values, such as converting strings to integers, performing arithmetic operations, and converting integers to different representations (binary, hexadecimal).

33. What is the purpose of the "Double" class in Java?

➡The "Double" class in Java is a wrapper class that provides methods for working with double-precision floating-point values. It offers functionality for parsing strings, performing arithmetic operations, and converting doubles to different representations (binary, hexadecimal).

34.What is the purpose of the "System" class in Java?

➡The "System" class in Java provides access to system resources and allows interaction with the system environment. It contains methods for standard input/output, error output, current time, copying arrays, and more.

35. What is the purpose of the "File" class in Java?

➡The "File" class in Java is used to represent and manipulate file and directory paths. It provides methods for creating, deleting, renaming, and querying file properties such as size, last modified date, and permissions.

36. What is the purpose of the useContext() hook?

➡The useContext() hook is used to consume values from a React context. It allows functional components to access context values without nesting multiple layers of components or using render props.

37. What is the purpose of the useImperativeHandle() hook?

➡The useImperativeHandle() hook allows a functional component to expose specific functions or values to the parent component through a ref. It is useful when you want to provide a more imperative API for a component that is primarily written as a functional component.

38. What is the purpose of the useDebugValue() hook?

➡ The useDebugValue() hook is used to display a custom label for custom hooks in React DevTools. It helps to provide more meaningful names for custom hooks when debugging and inspecting the component hierarchy.

39. What is the purpose of the useTransition() hook?

➡ The useTransition() hook is used in React to coordinate the rendering of concurrent transitions or animations. It allows for smoother user experiences by delaying the rendering of new updates until the transitions/animations have completed.

40. What is the purpose of the useQuery() hook in React Query?

➡The useQuery() hook is a part of the React Query library and is used to fetch and manage data in React components. It provides a declarative way to define and execute queries, handle caching, and manage the loading and error states of the data.

41. What is the difference between state and props in React Native?

➡ The difference between state and props in React Native is the same as in React.js. State is managed within a component and can be changed, while propsare passed to a component from its parent and cannot be modified directly by the component receiving them.

42. What is the purpose of AsyncStorage in React Native?

➡ AsyncStorage is a simple, asynchronous, persistent key-value storage system provided by React Native. It allows you to store data on the device's disk and retrieve it later, making it useful for caching data or storing user preferences.

43. What is the purpose of the Expo framework in React Native?

➡ Expo is a set of tools, libraries, and services built on top of React Native. It provides a simplified development workflow, pre-configured native modules, and access to device features, allowing developers to build and deploy React Native apps faster.

44.What is the purpose of the shouldComponentUpdate() method?

➡ The shouldComponentUpdate() method is a lifecycle method in React that determines whether a component should re-render or not. By implementing this method and returning false under certain conditions, you can optimize the performance of your application.

45. What is the React DevTools?

➡ React DevTools is a browser extension that allows you to inspect and debug React component hierarchies. It provides a set of tools for inspecting components, examining props and state, and profiling performance

46. What is the purpose of the componentDidCatch() method in React class components?

➡The componentDidCatch() method is a lifecycle method in React class components that is called when an error is thrown in a child component. It allows the parent component to handle the error and display fallback UI instead of the crashed component.

47. What is the purpose of the getDerivedStateFromProps() method in React class components?

➡The getDerivedStateFromProps() method is a static lifecycle method in React class components that is called when the props of a component change. It allows the component to update its state based on the new props, but it is used less frequently due to its complexity.

48. What is the purpose of the getSnapshotBeforeUpdate() method in React class components?

➡The getSnapshotBeforeUpdate() method is a lifecycle method in React class components that is called right before the changes from a component update are flushed to the DOM. It allows the component to capture information from the DOM before it potentially changes.

49. What is the purpose of the ReactDOMServer package in React?

➡ The ReactDOMServer package provides server-side rendering APIs for React. It allows you to render React components on the server and send the resulting HTML to the client, enabling faster initial page loads and better SEO.

50. What is the purpose of the ReactDOM.hydrate() method?

➡The ReactDOM.hydrate() method is similar to ReactDOM.render(), but it is used for rehydrating server-rendered HTML. It attaches event listeners and preserves the existing server-renderedmarkup and behavior while allowing React to take over the management of the component tree.

Thank you !

Follow For More