

Basic Questions

1. **What is React?**
 - React is a JavaScript library for building user interfaces, primarily for single-page applications.
 2. **What are components in React?**
 - Components are reusable, independent pieces of UI in React. They can be **functional** or **class-based**.
 3. **What is JSX?**
 - JSX (JavaScript XML) allows writing HTML inside JavaScript and makes the syntax easier to use.
 4. **What is the difference between state and props?**
 - **State** is a component's internal data, while **props** are external data passed from parent to child components.
 5. **What is the virtual DOM?**
 - A lightweight copy of the real DOM that React updates efficiently to improve performance.
-

Intermediate Questions

6. **What is the difference between functional and class components?**
 - Functional components are simpler and use hooks, while class components use lifecycle methods.
 7. **What are hooks in React?**
 - Hooks allow functional components to manage state and lifecycle features without using class components.
 8. **What is useState()?**
 - A React hook that allows functional components to manage state.
 9. **What is useEffect()?**
 - A hook that handles side effects like data fetching and subscriptions.
 10. **What is the useRef() hook?**
 - A hook that provides a reference to a DOM element or a mutable value that persists across renders.
-

Advanced Questions

11. **What is React Context API?**
 - A way to manage global state without prop drilling.
12. **What is Redux?**
 - A state management library for React that centralizes state in a global store.
13. **What is reconciliation in React?**
 - The process of updating the Virtual DOM and comparing it with the real DOM for efficient rendering.

14. What are React Portals?

- A feature that allows rendering components outside the main DOM hierarchy.

15. What is lazy loading in React?

- A technique to load components only when needed to improve performance.

16. What is memoization in React?

- Memoization is an optimization technique used to improve performance by caching the results of expensive function calls.
- **React.memo** is used to prevent unnecessary re-renders of functional components.

17. What is the difference between `useCallback()` and `useMemo()`?

- **useCallback** returns a memoized function, preventing unnecessary re-creations.
- **useMemo** returns a memoized **computed value** and re-computes only when dependencies change.

18. What is the difference between controlled and uncontrolled components?

- **Controlled components** store form data in React state.
- **Uncontrolled components** use the DOM for state management (via `useRef`).

19. How does React handle reconciliation?

- React compares the Virtual DOM with the previous version and updates only the changed parts using **diffing and fiber reconciliation**.

20. What are higher-order components (HOC)?

- A function that takes a component and returns a new component with additional behavior.

21. What is the React Fiber architecture?

- Fiber is a new reconciliation algorithm in React 16+ that improves rendering performance and enables features like Suspense and Concurrent Mode.

22. What is Concurrent Mode in React?

- A set of features that allows React to work on multiple tasks simultaneously, improving responsiveness and rendering performance.

23. What is Suspense in React?

- A feature that lets components **wait** for data before rendering, improving user experience.

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

- Rendering React components on the server instead of the client to improve SEO and performance. **Next.js** is commonly used for SSR.

25. What is hydration in React?

- The process of attaching event listeners to pre-rendered HTML from SSR to make it interactive.

26. What is React's new useOptimistic() hook?

- A React 18+ hook used to handle optimistic UI updates for better user experience.

27. What are React Suspense boundaries?

- They define fallback UI while waiting for content (e.g., lazy-loaded components or data fetching).

28. What is React's new useTransition() hook?

- A hook for marking state updates as **non-urgent**, improving UI responsiveness.

29. How does React Router work?

- React Router manages client-side navigation using components like `<Routes>`, `<Route>`, and `<Link>`.

30. What is progressive hydration?

- A technique where React loads and hydrates only critical components first and then progressively hydrates the rest.