

React Interview Question And Answer

Basic React Questions

1. **What is React?**

- React is a JavaScript library for building user interfaces, primarily for single-page applications where data changes dynamically.

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 is a syntax extension for JavaScript that allows writing HTML-like code within JavaScript files. JSX is transpiled to React.createElement calls.

4. **What are props in React?**

- Props (short for properties) are read-only inputs passed from parent to child components, allowing data flow between components.

5. **What is state in React?**

- State is a mutable data object that determines how a component renders and behaves. Changes in state trigger re-rendering of components.

6. **Difference between functional and class components?**

- Class components allow the use of lifecycle methods and are stateful, while functional components were stateless until the introduction of Hooks in React 16.8.

7. **What are React Hooks?**

- Hooks allow functional components to use state and other React features like lifecycle methods. Common hooks are useState, useEffect, etc.

8. **What is the Virtual DOM?**

- The Virtual DOM is an in-memory representation of the real DOM elements, allowing React to efficiently update only parts of the UI when data changes.

9. **What is the difference between controlled and uncontrolled components?**

- Controlled components rely on state to manage form inputs, while uncontrolled components maintain their own internal state without relying on React state.

10. **How do you create a functional component in React?**

Code-

```
function Greeting() {  
  
  return <h1>Hello, World!</h1>;  
}
```

}

Intermediate React Questions

11. What is the useState Hook?

- useState is a React hook that allows functional components to have local state.

Code-

```
const [count, setCount] = useState(0);
```

12. What is the useEffect Hook?

- useEffect allows you to perform side effects in functional components, like fetching data or manipulating the DOM.

13. How do you handle events in React?

- Events in React are handled similarly to HTML, but the syntax uses camelCase.

Code-

```
<button onClick={handleClick}>Click Me</button>
```

14. What are React Fragments?

- React Fragments allow grouping of child elements without adding extra nodes to the DOM.

Code-

```
<React.Fragment>
  <Child1 />
  <Child2 />
</React.Fragment>
```

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

- Keys help React identify which elements have changed, are added, or are removed. They should be unique for each list item.

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

- HOCs are functions that take a component and return a new component with additional props or behavior.

17. What is the context API in React?

- The Context API allows sharing state across components without prop drilling. It's useful for managing global state.

18. What is PropTypes?

- PropTypes is a way to enforce type checking on the props passed to components.

Code-

```
MyComponent.propTypes = {  
  name: PropTypes.string.isRequired,  
};
```

19. How can you optimize performance in React applications?

- Use techniques like memoization (React.memo), code-splitting, lazy loading, and avoiding unnecessary re-renders by controlling state updates carefully.

20. What is the useRef hook?

- useRef creates a mutable reference to a DOM element or a value that persists across renders.

Advanced React Questions

21. What are React Portals?

- Portals allow rendering children components into a DOM node outside the parent component hierarchy.

Code-

```
ReactDOM.createPortal(child, container);
```

22. What is code-splitting in React?

- Code-splitting allows you to load only the parts of your application needed at the moment, improving performance. It can be done using dynamic import() and React.lazy.

23. What is Redux?

- Redux is a state management library that centralizes the application's state in a global store, making it predictable and easier to manage.

24. What is the useReducer hook?

- useReducer is a hook used for managing complex state logic. It works like Redux reducers within a component.

Code-

```
const [state, dispatch] = useReducer(reducer, initialState);
```

25. How do you memoize components in React?

- You can use React.memo to memoize functional components and prevent re-rendering unless props change.

Code-

```
const MemoizedComponent = React.memo(Component);
```

26. What is the difference between useEffect and useLayoutEffect?

- `useEffect` runs after the render phase, while `useLayoutEffect` runs synchronously after DOM mutations, allowing you to measure the DOM.

27. What is reconciliation in React?

- Reconciliation is the process React uses to compare the Virtual DOM with the real DOM to determine what needs to be updated.

28. What is the purpose of StrictMode?

- StrictMode is a tool for highlighting potential problems in an application, such as deprecated features, side effects, and unsafe lifecycle methods.

29. How does React handle forms?

- Forms in React can be controlled or uncontrolled. Controlled forms rely on React state, whereas uncontrolled forms use `useRef` for direct access to form elements.

30. How does React's `useCallback` work?

- `useCallback` memoizes a function so that it is not redefined on every render, which can help with performance optimization.

Miscellaneous React Questions

31. What are the common lifecycle methods in class components?

- `componentDidMount`, `componentDidUpdate`, and `componentWillUnmount` are common lifecycle methods in class components.

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

- SSR is rendering React components on the server and sending the fully rendered HTML to the client. This improves performance and SEO.

33. What is the difference between client-side routing and server-side routing?

- Client-side routing is handled by JavaScript, allowing for faster navigation without reloading the page, while server-side routing involves making requests to the server for every route change.

34. How does React Router work?

- React Router is a library for handling routing in React applications. It matches URL paths to components, enabling navigation without refreshing the page.

35. What is lazy loading in React?

- Lazy loading refers to dynamically loading components only when they are needed, reducing the initial load time.

36. What is the `useImperativeHandle` hook?

- `useImperativeHandle` customizes the instance value that is exposed when using `ref` in a parent component, allowing you to control the child component's behavior.

37. What are React fibers?

- Fibers are the new reconciliation algorithm in React, allowing it to break rendering work into chunks and pause and resume work, leading to smoother rendering.

38. How do you handle errors in React?

- Error boundaries, which are React components, catch JavaScript errors anywhere in the component tree and display a fallback UI.

39. What is suspense in React?

- Suspense allows you to delay rendering part of the UI until some asynchronous task (e.g., data fetching) is complete.

40. What is Concurrent Mode in React?

- Concurrent Mode is a set of new features that help React apps stay responsive by allowing React to interrupt rendering when needed and work on high-priority updates first.

Coding-Specific React Questions

41. How do you lift state up in React?

- Lifting state up means moving the state to the closest common ancestor of the components that need to share the state.

42. How do you pass data between sibling components in React?

- You pass data between sibling components by lifting state up to their parent and passing it down as props.

43. What is the useMemo hook?

- useMemo memoizes a computed value, preventing expensive calculations from running on every render.

44. What is the useContext hook?

- useContext provides a way to consume context values in functional components, allowing components to subscribe to context changes.

45. What are compound components in React?

- Compound components are components that work together to share internal state, allowing for flexible and reusable component composition.

46. How do you implement pagination in React?

- Pagination can be implemented using state to store the current page and slicing the data array to display items for that specific page.

47. What is hydration in React?

- Hydration is the process of rendering a React application on the client, where the server-rendered HTML is reused and React takes over managing the UI.

48. How do you debounce input in React?

- Debouncing can be implemented by delaying a function call using `setTimeout` and clearing it when new input occurs.

49. How can you share logic between components in React?

- You can share logic between components using HOCs, render props, or custom hooks.

50. What is a render prop in React?

- A render prop is a technique for sharing code between components by passing a function as a prop to control what is rendered.