# React Interview Questions

Download PDF Print

Use the filter to quickly find topics like hooks, context, state management, reconciliation, performance, and testing.

Filter questions (e.g., hooks, state, props, reconciliation

Showing 60 of 60

1. ## What is React.js?

   React.js is a JavaScript library for building user interfaces, developed and maintained by Facebook.

2. ## What are the advantages of using React.js?

   Some of the advantages of using React.js include increased performance through virtual DOM, reusable components, and a strong community and support.

3. ## What are state and props in React.js?

   State and props in React.js are used for managing and passing data between components. Props are data passed from a parent component to a child component, while state is used to manage and store data within a component.

4. ## What is the purpose of JSX in React.js?

   JSX is a syntax extension for JavaScript used in React.js that allows for writing HTML-like code within JavaScript. It allows developers to write components in a more readable and intuitive manner.

5. ## What is the difference between state and props?

   State and props are both used for data management in React.js, but they serve different purposes. Props are data passed from a parent component to a child component, and cannot be changed within the child component. State, on the other hand, is used to manage data that can change within a component.

6. ## What is a React component?

   A React component is a piece of code that represents a part of a user interface. Components can be reused throughout an application and accept inputs (props) and manage their own state.

7. ## What is the lifecycle method of a React component?

   Lifecycle methods are methods that are automatically called at specific points during the lifecycle of a React component. Some of the most commonly used lifecycle methods include `componentDidMount`, `shouldComponentUpdate`, and `componentWillUnmount`.

8. ## How does React handle updates and rendering?

   React uses a virtual DOM to efficiently update the real DOM when the state or props of a component change. This allows for fast and efficient updates without having to re-render the entire component tree.

9. ## What is a Higher Order Component (HOC) in React?

   A Higher Order Component (HOC) is a pattern in React where a component wraps another component to reuse logic. HOCs are used to abstract complex logic and make it reusable across multiple components.

10. ## What is the purpose of the key property in React?

    The key property in React is used to identify each item in an array of items that are being rendered. It helps React to track which items have changed, added, or removed, so that it can efficiently update the UI.

11. ## What is the difference between stateless and stateful components in React?

    Stateless components, also known as functional components, are components that receive props and render them to the UI, but do not manage any state. Stateful components, also known as class components, manage their own state and render it to the UI.

12. ## What is the use of Redux in React?

    Redux is a state management library that can be used with React. It provides a centralized store for the state of an application, making it easier to manage and debug the state, especially in large applications.

13. ## What is the use of React Router in React?

    React Router is a library that provides routing for React applications. It allows for navigation between different parts of an application and ensures that the correct components are displayed based on the URL.

14. ## What is the difference between React.createElement and JSX?

    `React.createElement` is a low-level API for creating React elements, whereas JSX is a syntax extension for JavaScript that allows for writing HTML-like code within JavaScript. JSX compiles to calls to `React.createElement` at build time.

15. ## What is the purpose of the React context API?

The React context API is used to pass data down to components without having to pass it down through props at every level of the component tree. It allows for easier data management, especially for data that needs to be shared across many components.

16. ## What is the use of Refs in React?

    Refs are used to access the properties of DOM elements in React. They allow for direct access to the properties of a DOM element, making it easier to work with the values and styles of elements without having to modify the component's state or props.

17. ## What is the purpose of the React Hooks API?

    The React Hooks API allows for using state and other React features without having to write a class component. Hooks make it easier to reuse logic across components and share state between components.

18. ## What is the difference between props and state in React?

    Props are values passed down from a parent component to a child component, while state is an object that represents the internal state of a component. Props are read-only and cannot be changed within a component, while state can be updated within a component and triggers a re-render.

19. ## What is the use of memoization in React?

    Memoization is a technique used to optimize the performance of React components. It involves caching the result of a component's render so that it does not need to be re-computed if the inputs (props and state) do not change. This can lead to significant performance gains, especially in large applications with many components.

20. ## What is the purpose of the React Lazy API?

    The React Lazy API allows for lazy-loading components, meaning that the components are loaded only when they are needed. This can lead to faster initial load times and better performance, especially for large applications with many components.

21. ## What is the use of the useEffect hook in React?

    The useEffect hook allows for the execution of side effects in a functional component. Side effects can include API calls, subscriptions, and updates to the DOM. The useEffect hook takes a callback function as an argument and executes it after the component has finished rendering.

22. ## What is the use of the useState hook in React?

    The useState hook is used to add state to functional components in React. It returns an array with two values: the current state and a function for updating the state. The useState hook takes an initial value as an argument and allows for functional components to manage their own state.

23. ## What is the difference between server-side rendering and client-side rendering in React?

    Server-side rendering is when the initial render of a React application is done on the server and sent to the client as a fully-rendered page. Client-side rendering is when the initial render is done on the client and the page is updated as the user interacts with the application. Server-side rendering can lead to faster initial load times, but requires more processing on the server.

24. ## What is the use of the useContext hook in React?

    The useContext hook is used to access data from the React context in a functional component. The useContext hook takes a context object as an argument and returns the current value of the context. It allows for functional components to access data from the context without having to pass it down through props.

25. ## What is the purpose of the React Portals API?

    The React Portals API allows for rendering a component outside of its parent component. This can be useful for creating modals, tooltips, or other elements that need to be rendered outside of the normal component hierarchy.

26. ## What is the use of the React Router library?

    The React Router library is used to manage routing in React applications. It allows for defining multiple routes in an application, with each route corresponding to a different component. The React Router library also provides features such as URL parsing, navigation, and dynamic route matching.

27. ## What is the use of the useReducer hook in React?

    The useReducer hook is used to manage complex state updates in functional components. It takes a reducer function and an initial state as arguments and returns the current state and a dispatch function for updating the state. The useReducer hook can be a more efficient alternative to useState in cases where state updates depend on previous state values.

28. ## What is the use of the React Suspense API?

    The React Suspense API allows for declaratively waiting for a component to load before rendering it. This can be useful for loading asynchronous data or lazy-loading components, without having to manually handle loading states or error handling. The React Suspense API can also be used to show a loading indicator while data is being fetched or a component is being loaded.

29. ## What is the purpose of the React Concurrent Mode?

    The React Concurrent Mode is a experimental mode for React that allows for multiple components to render and update in parallel. This can lead to better performance and improved user experience, especially on slow or low-end devices. The React Concurrent Mode is still in development and should not be used in production applications at this time.

30. ## What is the use of the React Profiler API?

    The React Profiler API allows for measuring the performance of React components and identifying areas for optimization. The React Profiler API provides information about the render performance of components, including the time spent rendering, the number of render updates, and the component's state and props.

31. ## What is the use of the React Hooks API?

    The React Hooks API allows for using state and other React features in functional components. React Hooks provide a way to add state and other functionality to functional components, without having to convert them into class components. React Hooks include built-in hooks such as useState, useEffect, and useContext, as well as the ability to create custom hooks.

32. ## What is the use of the React Lazy API?

    The React Lazy API allows for lazily loading components or code. This can improve the performance of an application by reducing the amount of code that needs to be downloaded and parsed by the browser. React Lazy takes a function as an argument that returns a component, and returns a new component that can be rendered when needed.

33. ## What is the use of the React memo API?

    The React memo API allows for memoizing a component so that it only re-renders if its props have changed. This can improve the performance of an application by avoiding unnecessary render updates. React memo takes a component as an argument and returns a new component that only re-renders if its props have changed.

34. ## What is the purpose of the React Fragments API?

    The React Fragments API allows for returning multiple components from a single component, without adding extra nodes to the DOM. React Fragments are used to group multiple components together, without adding an additional wrapper element. React Fragments are specified using the <React.Fragment> syntax.

35. ## What is the use of the React Error Boundaries API?

    The React Error Boundaries API allows for handling errors in React components. Error boundaries are components that catch JavaScript errors anywhere in their child component tree, log those errors, and display a fallback UI instead of the component tree that crashed. Error boundaries are created using class components that implement the componentDidCatch lifecycle method.

36. ## What is the use of the React portals API?

    The React portals API allows for rendering components outside of the current component tree. Portals provide a way to render children into a DOM node that exists outside of the component hierarchy. This can be useful for rendering dialogs, tooltips, or other components that should be positioned relative to the root DOM node, rather than the parent component.

37. ## What is the use of the React Refs API?

    The React Refs API allows for accessing DOM nodes or React elements created in the render method. Refs provide a way to access the underlying DOM node or React component instance, for purposes such as focus management, measurement, or animation. Refs are created using the useRef hook or the createRef method, and attached to a DOM node or component using the ref attribute.

38. ## What is the use of the React Context API?

    The React Context API allows for sharing data between components without having to pass props down manually through every level of the component tree. Context provides a way to pass data through the component tree without having to pass props down manually at every level. The React Context API includes the useContext hook for accessing context data in functional components and the React.createContext method for creating context objects.

39. ## What is the use of the React useEffect hook?

    The useEffect hook is used to run side effects in functional components. Side effects can include things like data fetching, manual DOM updates, or interaction with other APIs. The useEffect hook takes a callback function as an argument and runs it after every render, by default. The useEffect hook can also be configured to only run when certain dependencies change or to only run on mount and unmount.

40. ## What is the use of the React useState hook?

    The useState hook is used to add state to functional components. State is a way to manage internal component data that can change over time, such as user input or data fetched from an API. The useState hook takes an initial state value as an argument and returns an array with the current state value and a function for updating the state value. The state value and the update function should be destructured when used in the component.

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

State and props are both used for holding data and passing data between components in React, but they have different purposes. Props (short for properties) are passed from a parent component to its children components, and are used for defining the initial values for a component's attributes. State, on the other hand, is used to hold data that can change within a component and affect its rendering. State is managed internally within a component, and can only be updated using setState methods.

### 42. What is a Higher Order Component (HOC) in React?

A Higher Order Component (HOC) is a function that takes a component and returns a new component with additional props. HOCs are used for adding logic or additional behavior to existing components, such as adding data fetching, providing a theme, or handling authentication. HOCs are created by defining a function that takes a component as an argument, and returns a new component that wraps the original component with additional props.

### 43. What is a React Router and why is it used?

React Router is a library for routing in React applications. It provides the ability to declare multiple routes in a React application and programmatically navigate between those routes. React Router is used for building single-page applications (SPAs) where the URL is updated without a full page reload, and the appropriate content is displayed based on the current URL. React Router makes it possible to build complex navigation patterns, such as nested routes, dynamic routes, and protected routes.

### 44. What is the difference between server-side rendering and client-side rendering in React?

Server-side rendering and client-side rendering are two different approaches for rendering React applications. Server-side rendering involves rendering the React application on the server, and sending the fully rendered HTML to the client. This provides a fast initial load time, as the client does not have to wait for JavaScript to execute and fetch data before the content is displayed. Client-side rendering, on the other hand, involves rendering the React application in the browser, after the initial HTML and JavaScript has been loaded. This approach provides a more dynamic and interactive user experience, as the client can fetch and update data in real-time, without having to reload the page.

### 45. What is the Virtual DOM in React and why is it used?

The Virtual DOM is a lightweight in-memory representation of the actual DOM in React. The Virtual DOM is used to improve the performance of React applications by allowing for efficient updates to the actual DOM. When a component's state changes, React updates the Virtual DOM, and then calculates the minimum number of updates necessary to update the actual DOM. This results in faster and more efficient updates, as only the necessary changes are made to the actual DOM, rather than re-rendering the entire component tree.

### 46. What is the use of refs in React?

Refs in React are used to access the properties of DOM elements. They are created using the `React.createRef()` method, and attached to a React component using the `ref` attribute. Refs are used for accessing the value of an input field, triggering a focus or blur event, measuring the dimensions of a DOM element, and for integration with non-React libraries that manipulate the DOM directly.

### 47. What is the purpose of the React `key` prop?

The `key` prop in React is used to provide a unique identifier for a list of components. When rendering a list of components, React uses the `key` prop to keep track of each component and determine which components have been added, removed, or changed. Using a `key` prop ensures that React can update the list efficiently, without having to re-render the entire list. It is important to use a unique and stable value for the `key` prop, such as a database ID, rather than using the index of the component in the list.

### 48. What is the purpose of the `constructor` method in React components?

The `constructor` method in a React component is used for initializing state and binding `this` to methods. The `constructor` is called before the component is mounted, and is the only place where `this.state` can be directly assigned. In addition, if a component has methods that need to access `this`, the methods must be bound to `this` in the constructor, or using arrow functions. The `constructor` method is optional in a React component, and is not required if the component does not have any initial state or methods that need to access `this`.

### 49. What is the difference between uncontrolled and controlled components in React?

In React, uncontrolled components are components that manage their own state, while controlled components are components that have their state managed by a parent component. An uncontrolled component uses a traditional HTML form element, such as an input, with a `defaultValue` or `defaultChecked` attribute to set its initial value. A controlled component, on the other hand, uses the `value` and `onChange` props to control its value. Controlled components provide a more explicit and deterministic way to manage the value of a form element, as the value is always set through props, and can only be updated through the `onChange` handler.

### 50. What is the purpose of the `defaultProps` property in React?

The `defaultProps` property in React is a static property on a component class that is used to define default values for props that are not provided. If a prop is not provided to a component, the value of the corresponding `defaultProps` property will be used instead. The `defaultProps` property is a good way to provide default values for optional props, and to provide documentation for the expected types and values of props.

### 51. What is the Virtual DOM in React?

The Virtual DOM in React is a lightweight in-memory representation of the actual DOM. When a React component updates, instead of updating the real DOM directly, React updates the Virtual DOM. Then, React calculates the difference between the previous Virtual DOM and the new Virtual DOM, and applies only the minimum number of changes required to the real DOM. This approach allows React to update the UI efficiently, without having to re-render the entire component tree on every change.

52. ## What is the purpose of the `shouldComponentUpdate` lifecycle method in React?

    The `shouldComponentUpdate` lifecycle method in React is used to control when a component should re-render. By default, a React component re-renders every time its state or props change. However, in some cases, it may be desirable to prevent a component from re-rendering if certain conditions are not met. The `shouldComponentUpdate` method is called before rendering, and returns a boolean value indicating whether the component should re-render. If `shouldComponentUpdate` returns `false`, the component will not re-render, and its children will also not re-render. This method can be used to improve the performance of a React application, by preventing unnecessary re-renders.

53. ## What is the purpose of the `setState` method in React?

    The `setState` method in React is used to update the state of a component. When a component's state changes, the component re-renders to reflect the changes. The `setState` method is used to schedule an update to the component's state, and triggers a re-render of the component. The `setState` method takes an object as its argument, with properties for the new state values, and merges the new values with the existing state. The `setState` method is asynchronous, which means that multiple `setState` calls may be batched together for performance reasons, and may not immediately update the component's state.

54. ## What is the difference between props and state in React?

    In React, `props` and `state` are both used to store data that affects the rendering of a component, but they have different purposes and characteristics. `Props` are properties that are passed down from a parent component to its children, and are used to communicate data and information from a parent to its children. `Props` are immutable and cannot be changed by the child component. On the other hand, `state` is an internal piece of data for a component, and is used to store values that can change within the component. `State` can be updated using the `setState` method, and triggers a re-render of the component when it changes.

55. ## What is the purpose of the `render` method in React?

    The `render` method in React is used to define the structure and content of a component. The `render` method returns a tree of React elements, which are the building blocks of a React component. The `render` method should be pure and have no side effects, such as network requests or DOM.

56. ## What is the difference between stateless and stateful components in React?

    In React, a component can be either stateless or stateful. Stateless components, also known as functional components, are components that receive props and render UI, but do not manage state. Stateless components are simple, fast, and easy to test. On the other hand, stateful components, also known as class components, are components that manage state and have access to lifecycle methods. Stateful components are used when a component needs to keep track of information that can change, such as form input values or whether a modal is open or closed. Stateful components are more complex and slower than stateless components, but are necessary when building interactive applications with React.

57. ## What is the purpose of the `key` prop in React?

    The `key` prop in React is used to distinguish between elements in a list of elements. When rendering a list of elements, React uses the `key` prop to keep track of which elements are added, removed, or reordered. By providing a unique `key` for each element in a list, React can efficiently update the list when changes occur, without having to re-render the entire list. The `key` prop should be a unique value for each element in the list, and should not change for the lifetime of the element. Commonly, the `key` prop is set to the ID of the item in the list, but any unique value can be used as a key.

58. ## What is the difference between a controlled and uncontrolled component in React?

    In React, a component can be either controlled or uncontrolled. A controlled component is a component that is controlled by React, and its state is updated through the use of props and callbacks. In a controlled component, the value of an input element, such as a text field or checkbox, is set by the state of the parent component, and changes to the input are made through event handlers. An uncontrolled component is a component that is not controlled by React, and its state is managed directly by the DOM. In an uncontrolled component, the value of an input element is managed by the DOM, and changes to the input are not automatically reflected in the state of the parent component.

59. ## What is the purpose of the `ref` prop in React?

    The `ref` prop in React is used to access the DOM elements or React elements created by a component. A `ref` is a reference to a DOM node or a React element, and can be used to read or modify the properties of that node or element. The `ref` prop can be used in class components or functional components with the useRef hook. Refs are useful when you need direct access to the DOM or a React element, for example, to focus an input field, measure the size of an element, or trigger an animation.

60. ## What is the purpose of the `constructor` method in React?

The `constructor` method in React is a special method that is called when a class component is created. The `constructor` method is used to initialize the state of a component.

---

Tip: When answering React questions, mention render cycles, state updates, and how hooks behave across renders.

SEARCH

Search topics... Go

INTERVIEW TOPICS

| | |
|---|---|
| Java | Q&A |
| Java 8 | Q&A |
| Spring | Q&A |
| Hibernate | Q&A |
| React JS | Q&A |
| SQL | Q&A |
| MongoDB | Q&A |
| JavaScript | Q&A |
| Kafka | Q&A |
| Design Patterns | Q&A |
| Microservices | Q&A |
| REST API | Q&A |
| Behavioral | Q&A |

QUICK ACTIONS

Start System Design | Practice Coding | Earn a Certificate

RECOMMENDED READING

- Learn Java 17 Programming
- Learn JavaScript Quickly
- SQL QuickStart Guide
- System Design Interview

See all ➜

NEED HELP?

Have a topic request or correction? We'd love to hear from you.

Contact Us

## THANK YOU!

Your comments and complements are always the inspiration that keeps us motivated.

## ABOUT

PROJECTS

ABOUT US

BLOG

AWARDS

## ADDRESS

Sterling, Virginia, 20166, US

info@techventurellc.com

641 451 3463

641 451 3463

## FOLLOW US