React JS Interview Questions and Answers for Freshers

As the demand for React.js developers continues to skyrocket, it's crucial to equip yourself with the knowledge and confidence to ace any interview. Here are the top 30 reactjs interview questions and answers for freshers in 2023.

1. What is React?

React is a popular JavaScript library used for building user interfaces. It allows developers to create reusable UI components and efficiently update the user interface when the underlying data changes. React follows a component-based architecture and uses a virtual DOM for efficient rendering.

2. What is JSX?

JSX is a syntax extension for JavaScript that allows developers to write HTML-like code within JavaScript. It enables the blending of HTML and JavaScript, making it easier to describe the structure and appearance of React components. JSX code is transpiled into regular JavaScript by tools like Babel.

3. What are the features of React?

- Component-based architecture
- Virtual DOM for efficient rendering
- JSX syntax for easy component creation
- Unidirectional data flow
- Reusable and composable components
- Support for server-side rendering
- Strong developer community and ecosystem

4. What are the advantages of using React?

- Enhanced performance with the virtual DOM
- Reusable components for efficient development
- Unidirectional data flow simplifies debugging and maintenance
- Support for server-side rendering for better initial loading speed and SEO
- Active community and vast ecosystem of libraries and tools
- Smooth integration with other libraries or existing projects

5. How is React different from Angular?

React and Angular are both popular JavaScript frameworks for building web applications, but they have some key differences:

- React is a library while Angular is a complete framework.
- React uses a virtual DOM, while Angular uses a real DOM.
- React follows a component-based architecture, while Angular uses a hierarchical structure.
- React uses JSX for templating, while Angular uses HTML templates.
- React is more flexible and allows developers to choose additional libraries, while Angular provides a more opinionated approach.

Also, If you are new to React, you might find it helpful to start with a complete ReactJS tutorial in 2023 to gain a comprehensive understanding of the library's fundamentals and best practices.

6. What is the Virtual DOM?

The Virtual DOM is a lightweight JavaScript object representation of the actual browser DOM. React uses the Virtual DOM to efficiently update and render components. When there are changes to the data, React calculates the difference between the previous and current Virtual DOM and applies only the necessary updates to the real DOM, resulting in improved performance.

7. What are the lifecycle methods of a React component?

This is one of the frequent react js interview questions in 2023. React components have several lifecycle methods that allow developers to hook into different phases of a component's life. The key lifecycle methods are:

- **componentDidMount()`:** Invoked after the component is mounted (inserted into the DOM).
- **`componentDidUpdate()`:** Invoked after the component is updated (re-rendered).
- **`componentWillUnmount()`:** Invoked before the component is unmounted and destroyed.
- **`render()`:** Responsible for rendering the component's JSX representation.
- **`shouldComponentUpdate()`:** Determines if the component should re-render or not.

8. What are the different ways to create a React component?

There are two main ways to create a React component:

- **Functional Components:** These are stateless components written as JavaScript functions. They receive props as arguments and return JSX elements.
- **Class Components:** These are stateful components defined as JavaScript classes. They extend the `React.Component` class and have additional features such as lifecycle methods and state management.

9. What is state and props in React?

- - State: State represents the internal data of a component. It is mutable and can be updated using `setState()`. State changes trigger re-rendering of the component and its child components.
- - Props: Props (short for properties) are read-only values passed from a parent component to its child components. They are used to provide data or configuration to components.

10. What is the difference between state and props?

- State is managed within a component and can be changed, while props are passed from parent components and cannot be modified by the child components.

- State is used for internal component data and can trigger re-rendering when updated, whereas props provide data and configuration to components, allowing parent components to communicate with their children.

If you're looking to enhance your programming skills, consider enrolling in a comprehensive Java certification training program to unlock your full potential.

11. How to pass props to a React component?

Props can be passed to a React component by adding attributes to the component when it is used in the parent component's JSX code. The values of these attributes become the props in the child component and can be accessed within the child component using `props`.

12. How to use state in a React component?

To use state in a React component, you need to define a state object within the component's class constructor using `this.state = { ... }`. You can then access and update the state using **`this.state`** and `this.setState()` respectively. When state changes, React automatically re-renders the component to reflect the updated state.

13. How to handle events in React?

In React, event handling is done by attaching event listeners to elements within components. Event handlers are defined as methods in the component class and are then assigned to the appropriate event using JSX. When the event occurs, the associated event handler is invoked, allowing you to perform the desired actions.

14. How to create a functional component in React?

A functional component in React is created by defining a JavaScript function that returns JSX elements. The function takes props as its parameter and uses them to render the component's output. Functional components are stateless and can be created using arrow functions or regular functions.

15. How to create a class component in React?

A class component in React is created by defining a class that extends the `React.Component` class. The component class must implement a `render()` method, which returns JSX elements representing the component's output. Class components can have state, lifecycle methods, and more advanced functionality compared to functional components.

16. What is the difference between functional and class components?

The main difference between functional and class components is the syntax and additional features they support. Functional components are simpler, written as JavaScript functions, and do not have state or lifecycle methods. Class components are defined as classes, and can have state, lifecycle methods, and other features like component-level methods.

Looking for inspiration? Explore a captivating array of Java project ideas that will fuel your creativity and enable you to showcase your programming expertise in exciting and innovative ways.

17. What is React Router?

React Router is a popular routing library for React applications. It allows developers to handle navigation and routing within a single-page application (SPA). React Router provides declarative routing through components, enabling the rendering of different components based on the current URL or route.

18. What is Redux?

Redux is a predictable state management library for JavaScript applications, including React. It provides a central store that holds the application state and allows components to access and update the state through actions and reducers. Redux helps manage complex application states and enables predictable data flow.

19. What is React Native?

React Native is a framework developed by Facebook that allows developers to build native mobile applications using React and JavaScript. It uses native components for rendering, resulting in highly performant and truly native mobile apps for both iOS and Android platforms.

20. What are the best practices for writing React code?

- Organize your code into reusable and modular components.
- Follow the single responsibility principle and keep components focused.
- Use meaningful and descriptive naming conventions.
- Utilize functional components whenever possible.
- Keep state management simple and localized.
- Write unit tests for your components and application logic.
- Optimize performance by using memoization and avoiding unnecessary re-renders.
- Follow established coding conventions and style guidelines.
- Leverage popular libraries and tools to streamline development processes.

21. What are the different types of React components?

There are mainly two types of React components:

- **Functional Components:** These are stateless components written as JavaScript functions. They receive props as arguments and return JSX elements. Functional components are easier to read, test, and maintain.
- **Class Components:** These are stateful components defined as JavaScript classes. They extend the `React.Component` class and have additional features such as lifecycle methods and state management. Class components are useful when you need to manage an internal state or use lifecycle methods.
- Embark on your coding journey with an engaging and comprehensive Java tutorial for beginners, designed to equip you with the foundational knowledge and practical skills needed to kickstart your programming career.

22. What is the difference between stateless and stateful components?

- Stateless components, also known as functional components, do not have an internal state. They receive data via props and render the output based on the given props. They are primarily responsible for presenting data and receiving user input.
- Stateful components, also known as class components, have an internal state managed using `this.state`. They can modify their state using `setState()` and trigger re-renders based on the updated state. Stateful components are used when a component needs to maintain and manage its own data.

To understand the concept of Java Exponent and how it works, refer to our in-depth article on What is Java Exponent.

23. What is the difference between class and functional components?

The key differences between class and functional components are as follows:

- **Syntax:** Class components are defined as JavaScript classes, while functional components are defined as JavaScript functions.
- **State and Lifecycle:** Class components can have state, manage their own lifecycle methods, and have access to lifecycle methods like `componentDidMount()` and `componentDidUpdate()`. Functional components do not have lifecycle methods or state management.
- **Complexity:** Functional components are simpler, easier to read, and generally preferred for their simplicity. Class components are useful when you need to manage state or use lifecycle methods.

24. What is the use of React.PureComponent?

React.PureComponent is a base class provided by React that is similar to a regular class component. However, it includes a built-in implementation of `shouldComponentUpdate()` that performs a shallow comparison of the component's props and state. If there are no changes, it prevents unnecessary re-renders, optimizing performance.

25. What is the use of React.memo?

React.memo is a higher-order component (HOC) that memoizes the result of a functional component. It memorizes the rendered output and reuses it if the component's props remain the same, avoiding unnecessary re-renders. It is used to optimize functional components that may re-render frequently but don't rely on changes in props.

Accelerate your career trajectory with Java certification jobs and training and pave the way for professional success in the dynamic world of software development.

26. What is the use of React.lazy?

React.lazy is a feature introduced in React 16.6 that enables code-splitting and lazy loading of components. It allows you to load a component lazily when it is needed, rather than including it in the initial bundle. This helps in reducing the bundle size and improving the initial loading speed of the application.

27. What is the use of React.Suspense?

React.Suspense is another feature introduced in React 16.6 that works together with React.lazy for code-splitting. It allows you to define fallback content to be displayed while the lazy-loaded component is being loaded. This fallback content could be a loading spinner or any other placeholder, providing a better user experience during component loading.

28. What is the use of React.Fragment?

React.Fragment is a component introduced in React 16 that allows you to return multiple elements without adding an extra wrapping element. It is useful when you want to render multiple elements in a component without introducing unnecessary wrapper divs or other elements.

29. What is the use of React.createElement? (continued)

such as when dynamically creating elements or when using React without JSX. It takes three arguments: the type of the element (e.g., HTML tag or custom component), optional properties or attributes for the element, and optional children elements.

30. What is the use of React.cloneElement?

React.cloneElement is a method provided by React that allows you to clone and modify a React element. It is typically used when you want to add or override props of a React element while maintaining its original type and key. This is useful when working with components that accept and modify their children's elements.

Here are the top Reactjs interview questions and answers for freshers, that will equip you with the necessary knowledge to ace your interviews. If you are looking for React js interview questions for experienced, below are the following.

React JS Interview Questions and Answers for Experienced

As React.js continues to dominate the tech landscape, employers are seeking top talent who can demonstrate a deep understanding of its intricacies. Here are our meticulously curated set of react js interview questions, to help you crack your next interview.

1. What are the performance benefits of using React?

React uses a virtual DOM, a lightweight representation of the actual DOM. This allows React to only update the actual DOM when necessary, which can lead to significant performance improvements. Additionally, React uses a unidirectional data flow, which helps to prevent race conditions and other performance issues.

2. How to optimize React components for performance?

There are a number of things you can do to optimize React components for performance. Some of the most important things include:

- Using functional components instead of class components
- Using pure components whenever possible
- Avoiding unnecessary re-renders
- Using memoization to cache expensive calculations
- Using lazy loading for large components

3. How to test React components?

There are a number of ways to test React components. Some of the most popular methods include:

- Unit testing
- Integration testing
- End-to-end testing

Unit testing is the most basic form of testing. It involves testing individual components in isolation. Integration testing tests how components interact with each other. End-to-end testing tests the entire application from start to finish.

4. How to debug React components?

Among the frequently asked React.js interview questions for experienced candidates, this one stands out. There are a number of ways to debug React components. Some of the most popular methods include:

- Using the React DevTools
- Using console.log()
- Using breakpoints

The React DevTools are a powerful tool that can be used to inspect React components and their state. They can also be used to debug React components.

console.log() is a JavaScript function that can be used to print values to the console. This can be helpful for debugging React components.

Breakpoints can be used to stop the execution of a program at a specific point. This can be helpful for debugging React components.

To learn more about the roles and responsibilities of Java developers, check out our detailed guide on Java Developers Roles.

5. How to scale React applications?

These kinds of React js interview questions are popular in 2023 and is often inquired about during interviews with experienced individuals. There are a number of things you can do to scale React applications. Some of the most important things include:

- Using a component library
- Using a state management library
- Using a routing library
- Using a testing library

A component library can help you to create reusable components. This can help to improve the maintainability and scalability of your application.

A state management library can help you to manage the state of your application. This can help to improve the performance and scalability of your application.

A routing library can help you to manage the navigation of your application. This can help to improve the user experience and scalability of your application.

A testing library can help you to test your application. This can help to improve the quality and scalability of your application.

6. How to work with React in a team environment?

There are a number of things you can do to work with React in a team environment. Some of the most important things include:

- Using a version control system
- Using a code review process
- Using a communication tool

A version control system can help you to track changes to your code. This can help to prevent conflicts and improve collaboration.

A code review process can help you to find and fix errors in your code. This can help to improve the quality of your application.

A communication tool can help you to communicate with your team members. This can help to improve collaboration and productivity.

7. How to use React with other libraries and frameworks?

This is one of the most asked react js interview questions for experienced. React can be used with a variety of other libraries and frameworks. Some of the most popular libraries and frameworks that can be used with React include:

- Redux
- React Router
- React Native

Redux is a state management library that can be used with React. It can help you to manage the state of your application in a centralized way.

React Router is a routing library that can be used with React. It can help you to manage the navigation of your application.

React Native is a framework that can be used to build native mobile applications using React.

8. What are the latest trends in React?

Some of the latest trends in React include:

- Hooks
- Suspense
- Context
- Incremental Static Regeneration (ISR)

Hooks are a new way to write React components. They allow you to use state and other React features without having to use class components.

Suspense is a new feature that allows you to lazy load components. This can help to improve performance.

Context is a new way to share data between components. It can help to improve code reuse and reduce boilerplate code.

Incremental Static Regeneration (ISR) is a new feature that allows you to pre-render parts of your application. This can help to improve performance.

If you're interested in expanding your programming skills beyond Java, consider exploring our comprehensive guide on JavaScript for an introduction to this versatile language.

9. What are the challenges of using React?

Some of the challenges of using React include:

- Learning curve
- State management
- Performance
- Testing

Learning curve

React can be a bit challenging to learn, especially for developers who are not familiar with JavaScript frameworks. There are a lot of concepts to learn, such as components, state, and props.

State management

Managing state in React can be a challenge. There are a number of different state management libraries available, each with its own pros and cons. It can be difficult to choose the right state management library for your application.

Performance

React can be a bit slow, especially for large applications. There are a number of things you can do to improve the performance of your React application, such as using functional components, us e

Managing state in React can be a challenge. There are a number of different state management libraries available, each with its own pros and cons. It can be difficult to choose the right state management library for your application.

Performance

React can be a bit slow, especially for large applications. There are a number of things you can do to improve the performance of your React application, such as using functional components, using pure components, and using memoization.

Testing

Testing React applications can be a challenge. React components are often deeply nested, which can make it difficult to test them effectively. There are a number of different testing libraries available, each with its own pros and cons. It can be difficult to choose the right testing library for your application.

11. What are the pros and cons of using React Router?

Pros:

- React Router is a powerful routing library that can be used to manage the navigation of your React application.
- It is easy to use and learn.
- It is well-documented and supported.
- It is used by a large number of developers, which means that there is a large community of support available.

Cons:

- React Router can be a bit complex to use, especially for large applications.
- It can be difficult to debug React Router applications.
- There are a number of other routing libraries available, each with its own pros and cons. It can be difficult to choose the right routing library for your application.

12. What are the pros and cons of using Redux?

Pros:

- Redux is a powerful state management library that can be used to manage the state of your React application.
- It is easy to use and learn.
- It is well-documented and supported.
- It is used by a large number of developers, which means that there is a large community of support available.

Cons:

- Redux can be a bit complex to use, especially for large applications.
- It can be difficult to debug Redux applications.
- There are a number of other state management libraries available, each with its own pros and cons. It can be difficult to choose the right state management library for your application.

13. What are the pros and cons of using React Native?

Pros:

- React Native is a powerful framework that can be used to build native mobile applications using React.
- It is easy to learn and use.
- It is well-documented and supported.
- It is used by a large number of developers, which means that there is a large community of support available.

Cons:

- React Native can be a bit complex to use, especially for large applications.
- It can be difficult to debug React Native applications.

For a detailed walkthrough and practical examples, delve into our in-depth article on How to Call a Method in Java, where you'll gain a deeper understanding of the process and improve your Java programming skills.

14. What are the best practices for writing React code that is both performant and maintainable?

Here are some best practices for writing React code that is both performant and maintainable:

- Use functional components instead of class components.
- Use pure components whenever possible.
- Avoid unnecessary re-renders.
- Use memoization to cache expensive calculations.
- Use lazy loading for large components.
- Use a component library.
- Use a state management library.
- Use a routing library.
- Use a testing library.
- Write unit tests.
- Write integration tests.
- Write end-to-end tests.
- Use a version control system.
- Use a code review process.

15. What is your favorite React library or framework?

(Sample Answer) My favorite React library is React Router. I think it is a very powerful and easy-to-use library for managing the navigation of React applications. Further, in these kind of react js interview questions the answers need to be concise clear and to the point which creates a quality impression on recruiters.

18. What are the pros and cons of using React hooks?

React hooks are a new feature that was introduced in React 16.8. They allow you to use state and other React features without having to use class components.

Pros of using React hooks:

- **Concise and easier to read:** Hooks are more concise and easier to read than class components. This is because they are written as functions, which are more familiar to JavaScript developers.
- **More flexible:** Hooks are more flexible than class components. This is because they can be used to implement a wider range of functionality.
- **Easier to test:** Hooks are easier to test than class components. This is because they are written as functions, which can be unit tested more easily.
- **Cons of using React hooks:**
- **Not as well-documented:** Hooks are not as well-documented as class components. This can make it difficult to learn how to use them.
- **Can be more difficult to debug:** Hooks can be more difficult to debug than class components. This is because they are more complex and there are more things that can go wrong.