Top 30 React Interview Questions and Answers

Are you preparing to appear for the interview of your dream job as a React developer? Or are you planning to switch your career to become a React developer? Wouldn't it be great if you knew what questions your potential employer would be going to ask you in your React interview? In this post, we will be sharing the top 30 React Interview Questions and Answers, covering ReactJS, React Native, and Redux.

This post will provide answers to the top React questions so that you can learn and memorize all important concepts or even craft your own insightful responses to clear a React interview.

ReactJS Interview Questions and Answers

Q1: What is React?

Ans. React is an open-source JavaScript library that is used to develop complex and interactive web and mobile UI. It is used for building web browser apps, creating dynamic libraries, and for building UI's that are perfectly equipped to render large datasets. React follows the component-based approach to create reusable and complex UIs from small and isolated pieces of code called components.

Q2. What are the features of React?

Ans. The features of React are:

- ReactJS follows a unidirectional data flow.
- It uses virtual DOM instead of the real DOM.
- React has a fully component-based architecture.
- It uses server-side rendering.

Q3. What are the advantages of React?

Ans. The advantages of React are:

- It ensures faster rendering.

- It is SEO-friendly.
- React combines the speed of JavaScript and uses a new approach to render web pages to make them dynamic and responsive.
- It can be used for the development of both web and mobile apps.
- JSX makes components/blocks code readable.
- It is easy to integrate with other frameworks, such as Angularand Meteor.

Q4. What are the disadvantages of React?

Ans. The disadvantages of React are:

- While using React, you need more code for development compared to its counterparts.
- It is just a library and not a full-fledged framework.
- Not using an isomorphic approach to exploit applications can lead to SEO indexing problems.
- React's large size library takes time to understand.
- ReactJS uses JSX, a syntax extension that allows HTML with JavaScript mixed together. For new developers, it may result in complexity in the learning curve.

Q5. What is JSX?

Ans. JSX is a syntax extension to JavaScript that comes with the full power of JavaScript. JSX stands for JavaScript XML. It enables you to write HTML in React.

JSX produces React elements and enables developers to put any JavaScript expressions within braces inside JSX. Each React element is a JavaScript object that you can store in a variable or pass around in your program. Here is an example:

```
render(){

  return(

<div>

<h1> Hello World </h1>

      </div>

  );
}
```

Q6. What is the difference between Real DOM and Virtual DOM?

Ans. The differences between the Real DOM and the Virtual DOM are:

Real DOM	**Virtual DOM**
1. Real DOM takes all the HTML elements and wraps them in a tree-like structured object.	1. It has the same features as the Real DOM object but it can't write and display things on the screen like Real DOM.
2. Updates slow.	2. Updates faster.
3. Manipulation is slow.	3. Manipulation is faster because nothing gets drawn onscreen.
4. Memory wastage is more.	4. No memory wastage.
5. It represents the document as nodes and objects.	5. A virtual DOM object is like a lightweight copy of Real DOM.

Q7. Explain Virtual DOM.

Ans. DOM stands for "Document Object Model". The virtual DOM (VDOM) is a virtual representation or a copy of the real DOM. It is a programming concept where an ideal, or virtual, representation of a UI is kept in memory and synced with the real DOM. Thus, whenever the state of the application changes, the virtual DOM gets updated instead of the real DOM.

Q8. Explain the different phases of the ReactJS component lifecycle?

Ans. There are four different phases of the ReactJS components lifecycle: Initialization, Mounting, Updating, and Unmounting.

1. Initialization: This is the first stage in the ReactJS lifecycle. In this stage, the component is constructed with the provided properties and a default state.
2. Mounting: It is the phase that involves putting elements into the DOM. It includes componentWillMount, and componentDidMount lifecycle methods.
3. Updating: This stage updates the stage of a component and repaints the application. This phase includes, shouldComponentUpdate, componentWillUpdate lifecycle methods.
4. Unmounting: It is the final stage, in which the component is removed from the page.

Q9. What is React.createClass?

Ans. Since JavaScript didn't have classes, React has its own class system. React.createClass allows you to generate component classes. Thus, the component class uses a class system implemented by React.

With ES6, React enables you to implement component classes that use ES6 JavaScript classes. While the way of doing is different, the end result would be the same.

Q10. Explain the lifecycle methods of ReactJS?

Ans. The lifestyle methods of ReactJS are:

1. componentWillMount: This method is used for App-level configuration in the root component. It is executed before rendering
2. componentDidMount: This method is executed after first rendering only on the client-side. In this, all AJAX requests, DOM or state updates, and more take place.
3. componentWillReceiveProps is invoked when the props are updated and before the next render is called.
4. shouldComponentUpdate: It should return a true or false value and will determine if the component will be updated or not. This is set to true by default. If you are sure that a component doesn't need to render after state or props are updated, then you can return a false value.
5. componentWillUpdate: It is invoked just before re-rendering the component.
6. componentDidUpdate: It is invoked just after rendering and is used to update the DOM in response to prop or state changes.
7. componentWillUnmount: It is called after the component is unmounted from the DOM.

Q11. What is the purpose of render() in React?

Ans. The render() function in React displays the specified HTML code inside the specified HTML element. It is mandatory for each React component to have a render(). This function takes 2 arguments, namely HTML code and an HTML element, and returns a single element which is the representation of the native DOM component.

If more than one HTML element is to be rendered, then they must be grouped inside one enclosing tag, such as <form>, <group>, and <div>.

Q12. What is the significance of Keys in React?

Ans. Keys are used in React to identify unique VDOM Elements with their corresponding data driving the UI and help React identify which items have changed, are added, or are removed. Keys should be a unique number or string.

Q13. What is Props?

Ans. Short for properties, Props are arguments passed into React components. They are like function arguments in JavaScript. Props accept arbitrary inputs and return React elements detailing what should be displayed on the screen. Props maintain a unidirectional data flow and are only passed down from the parent component to the child components throughout the application. A child component can never send a prop back to the parent component. They are immutable.

Q14. What are States in React?

Ans. States are the source of data. They are objects which determine components rendering and behavior. While props are immutable, States are mutable and can change over time. States create dynamic and interactive components and are accessed by using this.state(). While using the State component, you will set a component’s default state, access the current state, and update the state.

Q15. What is an Event in React?

Ans. React are actions based on user events. An event could be triggered by the user action or a system-generated event. React has the same events as HTML, such as click, change, and mouseover.

React events are written in camelCase syntax and are passed as functions instead of strings. Event arguments have a specific set of properties and behavior.

Q16. What is the difference between stateful and stateless components?

Ans. The difference between Stateful and Stateless Components are:

Stateful (Class) Component	**Stateless (Functional) Component**
1. Stateful Component stores information about the component's state change in memory.	1. It calculates the internal state of the components.
2. It has complex UI Logic.	2. It accepts properties(props) in function and return HTML (JSX).
3. It contains the data related to past, current, and possible future changes in state.	3. It does not contain the data related to the past, current, and possible future state changes.
4. It must have the render() method returning HTML.	4. There is no render method used in the Stateless component.
5. Have the authority to change the state.	5. Do not have the authority to change the state.

Q17. What are refs in React?

Ans. Refs are a function in React that acts as a way to reference a DOM element or a class component from within a parent component. It enables you to read and modify the value of a child component, without making use of props. Refs can be helpful when using third party libraries as well in animations.

Q18. What do you know about the controlled and uncontrolled components?

Ans. The differences between controlled and uncontrolled components are:

Controlled Components	**Uncontrolled Components**
They do not maintain their internal state.	They maintain their internal state.
Data is controlled by the parent component.	Data is controlled by the DOM itself.
They accept current value as a prop.	These components use ref to get their current values.
They allow validation control.	Uncontrolled components do not allow validation control.
Controlled components have good control over the form elements and data.	They have limited control over the form elements and data.

Q19. What are Higher Order Components (HOC)?

Ans. Higher Order Components are an advanced way of reusing the component logic. They are a pattern derived from React's compositional nature and are not considered a part of the React API. HOC are functions that take a component and return a new component. HOCs are pure components as they can accept any dynamically provided child component, however, they will not copy or modify any behavior from their input components.

Q20. What are Pure Components in React?

Ans. Pure components are those components that render the same output for the same state and props. They are the simplest and fastest components that can be written. They can replace any component which only has a render(). Thus, a function is said to be pure if:

- Its return value is determined only by its input values
- Its return value is always the same for the same input values

Q21. What is the use of Reducer?

Ans. Reducers are pure functions that describe how the state of the application changes when an action is triggered. Reducers take the previous state and action and then return a new state.

In simple terms, if you repeat the same test with the same input as arguments, you can always expect the same output.

Q22. What are React Hooks?

Ans. Hooks enable you to use state and other React features without writing a class. These in-built functions developers to use state and lifecycle methods inside functional components. They do not work inside classes. By using Hooks, you can completely avoid the use of lifecycle methods, like componentDidMount, componentDidUpdate, and componentWillUnmount.

Redux Interview Questions and Answers

Q23. What is Redux?

Ans. Redux is a predictable state container for JavaScript applications. It is a state management tool that enables developers to:

- write applications that behave in a consistent manner
- run in a variety of environments, and
- can be tested easily

Although it is widely used with React, it can be used with other JavaScript frameworks or libraries as well.

Q24. What are the three fundamental principles of Redux?

Ans. The three fundamental principles of Redux are:

1. Single source of truth. It means that the state of your application is stored in an object tree within a single store. The single state tree will make it easier to make changes and inspect the application.
2. State is read-only. It means that the only way to change the state is to trigger an action.
3. Changes are made with pure functions. You need pure functions to specify how the state tree is transformed by actions.

Q25. What are the components of Redux?

Ans. Redux has 4 components, namely:

1. Action: It is an object that tells what happened. Actions send data from the application to the store and are the only source of information for the store.
2. Reducer: Reducers specify how the state of the application will change changes in response to actions sent to the store. While Actions specify what happened, Reducers determine how the state will change in response to an action.
3. Store: It is where the State or Object tree of the entire application is saved. The Store brings Action and Reducer together
4. View: It shows the data provided by the Store.

React Native Interview Questions and Answers

Q26. What is the difference between Hot Reloading and Live Reloading in React Native?

Ans. Live reloading reloads or refreshes the entire app when a file changes.

Hot reloading only refreshes the files that were changed without losing the application's state. It just displays the code changes without restarting the app.

Q27. What is the difference between ReactJS and React Native?

Ans. The differences between React JS and React Native are:

ReactJS	**React Native**
1. It focuses on Web Development.	1. It is an extension of React and focuses on Mobile User Interfaces (Android and iOS).
2. ReactJS is a JavaScript library that enables developers to create a high performing UI Layer.	2. It is an entire framework for building cross-platform apps.

3. ReactJS uses Virtual DOM to render browser code.	3.React Native uses Native APIs to render components in mobile apps.
4. It uses HTML to render apps.	4. It doesn’t use HTML to render the app.

Q28: Explain the use of Flexbox in React Native?

Ans. Flexbox is a layout implementation in React Native. It offers an efficient way to create UIs and control positioning in React Native. It provides 3 properties, namely flexDirection, justifyContent, and alignItems to achieve the desired layout.

Q29. Explain the use of the ScrollView component.

Ans. The ScrollView component is a generic scrolling container. It can contain multiple components and views. However, the scrollable items need not be homogeneous. It enables you to scroll both vertically and horizontally by setting the horizontal property.

Q30. What do you mean by Component Driven Development (CDD)?

Ans. Component Driven Development (CDD) is a methodology that holds the developing process which is revolving around the components. It is a process that creates UIs from the bottom up by starting at the components level and ending at the pages or screen level. It is a concept that applies to pattern libraries, design systems, and modern JavaScript frameworks.

Visit Naukri Learning website for more