ReactJS Interview Questions You Need to Know

With answers and examples.

Explain the Virtual DOM and how it works in ReactJS.

The Virtual DOM in ReactJS is a lightweight copy of the actual DOM that is used to improve performance by minimizing the number of updates to the actual DOM.

For example, when a user interacts with a React component, the Virtual DOM updates first, and then React compares the updated Virtual DOM with the previous version to determine which parts of the actual DOM need to be updated.

What is JSX and how is it different from HTML?

JSX is a syntax extension for JavaScript that allows developers to write HTML-like code within their JavaScript files, and it is different from HTML because it is not valid JavaScript code.

For example, in React, you can use JSX to define a component's UI like this:

```
const element = <h1>Hello, world!</h1>;
```

How does ReactJS handle state management?

ReactJS handles state management by allowing developers to define and update state within a component using the useState hook, which triggers a re-render of the component when the state changes.

For example, in a simple counter component, you can define and update the state like this:

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

setCount(count + 1);
```

What is the significance of props in ReactJS?

In ReactJS, props are used to pass data from one component to another component down the component tree, and they are read-only

Example: To pass data from a parent component to a child component, we can use props. We can pass the data as props when we are calling the child component, and then access the props in the child component

Explain the concept of React Hooks and how they are used

React Hooks are a pattern that allows developers to use state and other React features in functional components, and they are JavaScript functions that manage state's behavior and side effects by isolating them from a component

Example: The `useState()` hook allows us to create, track, and update a state in functional components

What is the difference between controlled and uncontrolled components in ReactJS?

In ReactJS, controlled components are components that have their state and behavior controlled by the parent component, while uncontrolled components manage their own state internally

Example: A controlled component is bound to a value, and its changes will be handled in code by using event-based callbacks, while an uncontrolled component is handled by the DOM itsel

How does ReactJS handle server-side rendering?

In ReactJS, server-side rendering renders the React components on the server and sends the output as HTML content to the client, which can be combined with client-side rendering to create an isomorphic app.

Example: To implement server-side rendering in a React app, we can use a library like Next.js, which provides a framework for server-side rendering and other features like automatic code splitting and static exporting

What is the significance of the key prop in ReactJS?

In ReactJS, the key prop is used to identify elements across renders, and it is most commonly used when rendering a list of items to help React identify which items have changed, are added, or are removed.

Example: When rendering a list of items in React, we should assign a unique key to each item using the key prop to help React identify which items have changed, are added, or are removed

Explain the concept of higher-order components in ReactJS.

In ReactJS, a higher-order component is a function that takes a component and returns a new component, which is used to compose components for code reuse and logic abstraction.

Example: We can use a higher-order component to add some functionality to a component, such as logging or authentication, by wrapping the component with the higher-order component

How does ReactJS handle performance optimization?

In ReactJS, performance optimization can be achieved through code-splitting, which helps to "lazy-load" just the things that are currently needed by the user, and React lazy loading, which allows for the dynamic rendering of components.

Example: We can use React.lazy() function to render a dynamic import as a normal component, which makes it simple to construct components that are loaded only when needed

Hi, I'm Rahul

I'll help you get better at React, Follow along.