React JS Intervie Questions

Easy

Medium

Hard

@Maheshpal Singh Rathore
@mpsrathore2020
@Kiran
@kiran

What is React.js and how does it other JavaScript libraries?

- *React.js is a JavaScript library for bu interfaces.*
- *It allows developers to create reusa components and manage the state o those components.*
- *It differs from other JavaScript libra it focuses specifically on the view lay application, making it a great choice complex, large-scale user interfaces.*

@Maheshpal Singh Rathore
@mpsrathore2020

What are the advantages of usin React.js?

- *Reusable components.*
- *Virtual DOM for efficient updates a rendering.*
- *Good performance.*
- *Strong developer community and s*
- *Easy integration with other libraries frameworks.*
- *Can be used on the client and serve*


@Maheshpal Singh Rathore


@mpsrathore2020

❓How does the virtual DOM in Rea

- *React uses a virtual DOM (Document O*
  to optimize updates and rendering.

- *The virtual DOM is a lightweight in-mem*
  representation of the actual DOM.

- *When the state of a component change*
  compares the virtual DOM with the act
  only makes changes to the actual DOM
  necessary, which is much more efficien
  rendering the entire page.

@Maheshpal Singh Rathore
@mpsrathore2020

❓How does React.js handle updat rendering?

- *When a component's state change re-render that component and its c components to reflect the new stat*
- *React uses a virtual DOM to optimi by only re-rendering the specific pa actual DOM that have changed.*
- *This helps to improve the performa application.*

@Maheshpal Singh Rathore
@mpsrathore2020

What are the components in Rea and how are they used?

- *Components in React.js are the bui blocks of a React application.*
- *They are used to create reusable U elements.*
- *Components can be either function class-based and can be nested to c more complex UI elements.*
- *Components accept inputs called p manage their own state.*

@Maheshpal Singh Rathore
@mpsrathore2020

How does React.js handle state a

- *State in React.js refers to the data that determine a component's b render its content.*
- *State can be changed within a comp will trigger a re-render.*
- *Props (short for properties) are inp a component from its parent.*
- *They are read-only and cannot be ch the component.*


@Maheshpal Singh Rathore


@mpsrathore2020

What is JSX and how is it used in React.js?

- *JSX is a syntax extension for Java allows developers to write HTML-li in their JavaScript code.*

- *It is used in React to describe the st content of a component.*

- *JSX is transpiled to plain JavaSc being executed, so it is compatible browsers.*

@Maheshpal Singh Rathore
@mpsrathore2020

What is the component lifecycl React.js?

- *The component lifecycle in React.js the different stages a compon through, from its creation to its dest*

- *The main lifecycle methods include.*

1. componentDidMount: executed afte render .

2. componentDidUpdate: executed a update.

3. componentWillUnmount: executed b component is removed from the DOM.

@Maheshpal Singh Rathore

@mpsrathore2020

How do you use event handling React.js?

- *Event handling in React.js is done onEventName syntax, where Even the name of the event you want t such as onClick or onSubmit.*

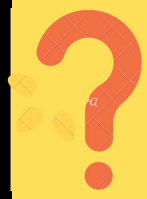
- *Event handlers are passed as pro component and are typically defi arrow functions or bound metho*

@Maheshpal Singh Rathore
@mpsrathore2020

What is the significance of prop React.js?

- *Props are used to pass data from component to a child component.*
- *Props provide a way to make comp reusable and configurable.*
- *Props components are read-only components.*

@Maheshpal Singh Rathore
@mpsrathore2020



How do you use forms and form validation in React.js?

- *Forms and form validation in React.j typically implemented using controll components, where the form input v are stored in the state and updated user interacts with the form.*

- *Form validation is then performed b checking the values in the state agai of rules.*

@Maheshpal Singh Rathore
@mpsrathore2020

How do you handle routing in a application?

- *Routing in a React.js application is handled using a library such as Re Router.*

- *This library provides components for defining routes and navigating them.*

@Maheshpal Singh Rathore
@mpsrathore2020

How do you use React.js with a s management library such as Red

- *React.js can be used with a state management library such as Redux integrating the Redux store with the components.*

- *This allows for better management shared state between components.*

@Maheshpal Singh Rathore
@mpsrathore2020

What is the significance of Highe Components (HOC) in React.js?

- *Higher Order Components (HOC) in are components that wrap other components to add additional func*

- *They are significant because they a code reuse and abstract common functionality into a single, reusable component.*

@Maheshpal Singh Rathore
@mpsrathore2020

How do you use Hooks in React

- *Hooks were introduced in React 16. allow for using state and other Rea features without writing a class component.*

- *Hooks make it easier to reuse logic between components and provide flexible and concise code.*

- *They are significant because they c more flexible and concise code*


@Kiran Kanwar Rathore


@kiranrathore123

How do you use Context API in

- *The Context API in React.js is a fea*
  that allows for sharing data betwe
  components without passing prop
  through multiple levels of compon

- *This is useful for data that is neede*
  many components throughout an
  application.


@Kiran Kanwar Rathore


@kiranrathore123

How can you optimize the perfo of a React.js application?

- *Performance of React.js applicati be optimized through techniques using the shouldComponentUpda lifecycle method and lazy loading*

- *Memoization can also be used to the performance of React.js appli*

@Kiran Kanwar Rathore
@kiranrathore123

How do you test React.js compon

- *React.js components can be tested*
  various testing libraries, such as Je
  and Enzyme.

- *These libraries provide APIs for wr*
  and running unit tests for React
  components.

@Kiran Kanwar Rathore
@kiranrathore123

How does Server-side renderin in React.js?

- *Server-side rendering in React.js i rendering the initial HTML on the rather than in the browser.*

- *This can help improve performan especially for slower devices or lo bandwidth connections.*

@Kiran Kanwar Rathore
@kiranrathore123

How does React.js handle differ types of errors?

- *React.js handles different types of through various means, such as the catch statement, the use of error boundaries, and global error handl*

- *Error boundaries are React compo that catch JavaScript errors anywh their child component tree, log the and display a fallback UI.*

@Kiran Kanwar Rathore
@kiranrathore123

What is the significance of Reac lifecycle methods?

- *React.js lifecycle methods are use manage the various stages of a component's lifecycle, such as mo updating, and unmounting.*

- *Lifecycle methods can be used to actions such as fetching data, set subscriptions, or updating the con state.*

@Kiran Kanwar Rathore
@kiranrathore123

Can you explain how React's reco algorithm works and why it's impo

- *React's reconciliation algorithm is the which React updates the DOM in respo changes in the components' state or p*

Here are three key points about how it wo

1. *Virtual DOM: React uses a virtual repr the DOM, called the Virtual DOM, to ke changes and update the actual DOM e*

2. Tree comparison: When a change occur compares the updated Virtual DOM tree w previous Virtual DOM tree to determine th number of updates required to bring the c into sync with the updated Virtual DOM.

@Kiran Kanwar Rathore
@kiranrathore123

3. Update optimization: React uses heuri
optimizations to minimize the number of
required and make the update process a
possible. The use of the Virtual DOM and
reconciliation algorithm make React app
fast, even for large and complex user int

- *React's reconciliation algorithm is im*
  because it allows React to update th
  interface efficiently and with minima
  making it well-suited for complex and
  applications. Additionally, the use of
  DOM provides a clear separation bet
  user interface and the actual DOM, m
  easier to reason about the behavior o
  application.

@Kiran Kanwar Rathore
@kiranrathore123

Can you explain the concept of "l up" in React and why it's importa

- *"Lifting state up" is a concept in React to the process of sharing state betwee components by moving it from lower-l components to higher-level componen*

Here are three key points about why this

1. *Centralized management: By lifting sta can centralize the management of sta few higher-level components, making i understand and maintain the applicat*
2. *Reusability: When state is lifted up, low components that need access to that receive it as props. This makes it easie those components in different parts o application, as they are not tightly cou state they depend on.*

@Kiran Kanwar Rathore
@kiranrathore123

3. Improved performance: Moving state
help improve performance, as React's re
algorithm can take advantage of the fac
a few components are changing instead
update many components individually.

- *Lifting state up is a critical concept i*
  can help improve the structure and
  maintainability of your applications.
  centralizing state management and r
  components more reusable, you can
  cleaner and more efficient code.

@Kiran Kanwar Rathore
@kiranrathore123

Can you explain the use of Redux w and how it differs from using React state management?

Use of Redux With React :

- *Centralized store: Redux is a state mana library that provides a centralized store application. The store contains the stat whole application and can be updated u and reducers.*
- *Improved scalability: Redux makes it ea manage the state of a large or complex as all the state is contained in a single s updates are made using well-defined ac reducers.*
- *Better separation of concerns: By using can separate the state management fro presentation of the components, making understand and maintain the applicatio*

@Kiran Kanwar Rathore
@kiranrathore123

Difference between React's built-in st
management and Redux:

- *Local vs global: React's built-in state m*
  is local to individual components, whil
  provides a global store for the whole a

- *Scalability: React's built-in state manc*
  become cumbersome in large or comp
  applications, while Redux provides a m
  solution.

- *Separation of concerns: React's built-i*
  management is closely tied to the pres
  the components, while Redux provides
  modular and scalable solution by sepa
  state management from the presentat

@Kiran Kanwar Rathore
@kiranrathore123

Can you explain the difference be stateless and stateful component

Difference between stateless and sta components in React:

- *State management: Stateful compon maintain their own state, while statel components receive all the data they props from higher-level components.*

- *Reusability: Stateless components ar more reusable, as they do not mainto and rely solely on the props they rece*

- *Performance: Stateless components faster and use less memory, as they d to manage their own state.*

@Kiran Kanwar Rathore
@kiranrathore123

Can you explain the concept of "c components" in React and why th important?

Concept of controlled components in Re

- *Controlled by React: Controlled compo React are components that have their behavior controlled by React, rather th user or the DOM.*
- *Better control: By controlling the value behavior of a component, you can mor manage the behavior of the componen that it behaves as expected.*
- *Improved reliability: Controlled compo help improve the reliability of your app you have more control over the behavi component and can ensure that it beh expected.*

@Kiran Kanwar Rathore
@kiranrathore123

Can you explain the concept of "re updates" in React and how it diffe traditional data binding?

Concept of "reactive updates" in React:

- *Reactive nature: Reactive updates in R the way that React updates the user in response to changes in the data. React UI reactively, meaning that it updates t response to changes in the data.*
- *Improved performance: Reactive upda improve performance by only updating the UI that have changed, rather than r the entire UI.*
- *Dynamic updates: Reactive updates al dynamic updates to the UI, as the UI is automatically updated in response to c the data.*

@Kiran Kanwar Rathore

@kiranrathore123

Differences from traditional data bind

- *Two-way vs one-way: Traditional dat*
  often involves two-way binding, whe
  changes in the UI can also update th
  React, updates are one-way, with ch
  the data causing updates to the UI.

- *Declarative nature: React uses a dec*
  approach to updating the UI, whereas
  traditional data binding often uses a
  imperative approach.

- *Efficient updates: React's reactive up*
  are more efficient than traditional da
  binding, as React only updates the pa
  the UI that have changed.

@Kiran Kanwar Rathore
@kiranrathore123

Can you explain how React hand performance optimization, such loading and memoization?

React's performance optimization tec

- *Lazy loading: Lazy loading in React loading components only when the needed, rather than loading all co upfront. This can improve perform reducing the amount of data that loaded and processed.*

- *Memoization: Memoization in Reac caching the results of expensive co so that they can be reused in the fu than recomputing the results each can improve performance by reduc amount of redundant computation*

@Kiran Kanwar Rathore
@kiranrathore123

- *Virtual DOM: React uses a virtual DOM is a lightweight in-memory representa the actual DOM, to update the UI effic This can improve performance by min the number of actual DOM updates th required.*

@Kiran Kanwar Rathore
@kiranrathore123

Did you find this pos helpful?

@Maheshpal Singh Rathore

@mpsrathore2020

@Kiran Kanwar

@kiranrathore1

Credit - Internet

Like , save, and share this with friends!!!