ReactJS Interview Questions

SWIPE NEXT

Khushi_Kumari

What is React?

React (aka React.js or ReactJS) is an open-source front-end JavaScript library that is used for building composable user interfaces, especially for single-page applications. It is used for handling view layer for web and mobile apps based on components in a declarative approach.

React was created by Jordan Walke, a software engineer working for Facebook. React was first deployed on Facebook's News Feed in 2011 and on Instagram in 2012.

What are the major features of React?

The major features of React are:

- Uses JSX syntax, a syntax extension of JS that allows developers to write HTML in their JS code.
- It uses Virtual DOM instead of Real DOM considering that Real DOM manipulations are expensive.
- Supports server-side rendering which is useful for Search Engine Optimization(SEO).
- Follows Unidirectional or one-way data flow or data binding.
- Uses reusable/composable UI components to develop the view.

What is Virtual DOM?

The Virtual DOM (VDOM) is an in-memory representation of Real DOM. The representation of a UI is kept in memory and synced with the "real" DOM. It's a step that happens between the render function being called and the displaying of elements on the screen. This entire process is called reconciliation.

What is the difference between Shadow DOM and Virtual DOM?

The Shadow DOM is a browser technology designed primarily for scoping variables and CSS in web components. The Virtual DOM is a concept implemented by libraries in JavaScript on top of browser APIs.

What is React Fiber?

Fiber is the new reconciliation engine or reimplementation of core algorithm in React v16. The goal of React Fiber is to increase its suitability for areas like animation, layout, gestures, ability to pause, abort, or reuse work and assign priority to different types of updates; and new concurrency primitives.

How to create components in React?

Components are the building blocks of creating User Interfaces(UI) in React. There are two possible ways to create a component.

Functional Components: This is the simplest way to create a component. Those are pure JavaScript functions that accept props object as the one and only one parameter and return React elements to render the output:

```
function Greeting({ message }) {
    return <h1>{`Hello, ${message}`}</h1>;
}
```

Class Components: You can also use ES6 class to define a component. The above functional component can be written as a class component:

```
class Greeting extends React.Component {
    render() {
        return <h1>{`Hello, ${this.props.message}`}</h1>;
    }
}
```

Khushi_Kumari

What is the difference between state and props?

In React, both state and props are plain JavaScript objects and used to manage the data of a component, but they are used in different ways and have different characteristics.

The state entity is managed by the component itself and can be updated using the setter(setState() for class components) function. Unlike props, state can be modified by the component and is used to manage the internal state of the component. Moreover, changes in the state trigger a re-render of the component and its children. The components cannot become reusable with the usage of state alone.

On the otherhand, props (short for "properties") are passed to a component by its parent component and are read-only, meaning that they cannot be modified by the own component itself. Also, props can be used to configure the behavior of a component and to pass data between components. The components become reusable with the usage of props.

What are stateless components?

If the behaviour of a component is independent of its state then it can be a stateless component. You can use either a function or a class for creating stateless components. But unless you need to use a lifecycle hook in your components, you should go for functional components. There are a lot of benefits if you decide to use functional components here; they are easy to write, understand, and test, a little faster, and you can avoid the this keyword altogether.

What is the use of react-dom package?

The react-dom package provides DOM-specific methods that can be used at the top level of your app. Most of the components are not required to use this module. Some of the methods of this package are:

1. render()
2. hydrate()
3. unmountComponentAtNode()
4. findDOMNode()
5. createPortal()

What is the difference between Element and Component?

React Element:

An Element is a plain object describing what you want to appear on the screen in terms of the DOM nodes or other components. Elements can contain other Elements in their props. Once an element is created, it cannot be mutated.
The JavaScript representation(Without JSX) of React Element would be as follows:

const element=React.createElement("h2",{id:"d1"}, "React Counter Example");

React Component:

It can be a function or class with a **render()** method. In either case, it takes props as an input, and returns a JSX tree as the output. It has to keep references to its DOM nodes and to the instances of the child components.

```
function Counter() {
  return <h2>React Counter Example!</h2>;
}
ReactDOM.render(<Counter/>
,document.getElementById('root'));
```

What is React Router?

React Router is a powerful routing library built on top of React that helps you add new screens and flows to your application incredibly quickly, all while keeping the URL in sync with what's being displayed on the page.

How to use React label element?

If you try to render a <label> element bound to a text input using the standard for attribute, then it produces HTML missing that attribute and prints a warning to the console.

```
<label for={'user'}>{'User'}</label>
<input type={'text'} id={'user'} />
```

Since for is a reserved keyword in JavaScript, use htmlFor instead.

```
<label htmlFor={'user'}>{'User'}</label>
<input type={'text'} id={'user'} />
```

How can we find the version of React at runtime in the browser?

You can use React.version to get the version.

```
Const REACT_VERSION = React.version;
ReactDOM.render(
  <div>{`React version: ${REACT_VERSION}`}</div>,
  documen[illegible]tById("app")
);
```

Khushi_Kumari

How events are different in React?

Handling events in React elements has some syntactic differences:

1. React event handlers are named using camelCase, rather than lowercase.
2. With JSX you pass a function as the event handler, rather than a string.

How to enable production mode in React?

You should use Webpack's DefinePlugin method to set NODE_ENV to production, by which it strip out things like propType validation and extra warnings. Apart from this, if you minify the code, for example, Uglify's dead-code elimination to strip out development only code and comments, it will drastically reduce the size of your bundle.

Why you can't update props in React?

The React philosophy is that props should be immutable(read only) and top-down. This means that a parent can send any prop values to a child, but the child can't modify received props.

Why ReactDOM is separated from React?

The React team worked on extracting all DOM-related features into a separate library called ReactDOM. React v0.14 is the first release in which the libraries are split. By looking at some of the packages, react-native, react-art, react-canvas, and react-three, it has become clear that the beauty and essence of React has nothing to do with browsers or the DOM.

To build more environments that React can render to, React team planned to split the main React package into two: react and react-dom. This paves the way to writing components that can be shared between the web version of React and React Native.

What is the difference between React and ReactDOM?

The react package contains React.createElement(), React.Component, React.Children, and other helpers related to elements and component classes. You can think of these as the isomorphic or universal helpers that you need to build components. The react-dom package contains ReactDOM.render(), and in react-dom/server we have server-side rendering support with ReactDOMServer.renderToString() and ReactDOMServer.renderToStaticMarkup().

What is reconciliation?

Reconciliation is the process through which React updates the Browser DOM and makes React work faster. React use a diffing algorithm so that component updates are predictable and faster. React would first calculate the difference between the real DOM and the copy of DOM (Virtual DOM) when there's an update of components. React stores a copy of Browser DOM which is called Virtual DOM. When we make changes or add data, React creates a new Virtual DOM and compares it with the previous one. This comparison is done by Diffing Algorithm. Now React compares the Virtual DOM with Real DOM. It finds out the changed nodes and updates only the changed nodes in Real DOM leaving the rest nodes as it is. This process is called Reconciliation.

What are the recommended ways for static type checking?

Normally we use PropTypes library (React.PropTypes moved to a prop-types package since React v15.5) for type checking in the React applications. For large code bases, it is recommended to use static type checkers such as Flow or TypeScript, that perform type checking at compile time and provide auto-completion features.

What is Lifting State Up in React?

When several components need to share the same changing data then it is recommended to lift the shared state up to their closest common ancestor. That means if two child components share the same data from its parent, then move the state to parent instead of maintaining local state in both of the child components.

What are Higher-Order Components?

A higher-order component (HOC) is a function that takes a component and returns a new component. Basically, it's a pattern that is derived from React's compositional nature.

We call them pure components because they can accept any dynamically provided child component but they won't modify or copy any behavior from their input components.

```
const EnhancedComponent =
higherOrderComponent(WrappedComponent);
```

HOC can be used for many use cases:

1. Code reuse, logic and bootstrap abstraction.
2. Render hijacking.
3. State abstraction and manipulation.
4. Props manipulation.

Why fragments are better than container divs?

Below are the list of reasons to prefer fragments over container DOM elements,

1. Fragments are a bit faster and use less memory by not creating an extra DOM node. This only has a real benefit on very large and deep trees.
2. Some CSS mechanisms like Flexbox and CSS Grid have a special parent-child relationships, and adding divs in the middle makes it hard to keep the desired layout.
3. The DOM Inspector is less cluttered.

What are portals in React?

Portal is a recommended way to render children into a DOM node that exists outside the DOM hierarchy of the parent component. When using CSS transform in a component, its descendant elements should not use fixed positioning, otherwise the layout will blow up.

ReactDOM.createPortal(child, container);

The first argument is any render-able React child, such as an element, string, or fragment. The second argument is a DOM element.

Why React uses className over class attribute?

The attribute names written in JSX turned into keys of JavaScript objects and the JavaScript names cannot contain dashes or reversed words, it is recommended to use camelCase whereever applicable in JSX code. The attribute class is a keyword in JavaScript, and JSX is an extension of JavaScript. That's the principle reason why React uses className instead of class. Pass a string as the className prop.

```
render() {
    return <span className={
'menu navigation-menu'}>{'Menu'}</span>
}
```

Do Hooks replace render props and higher order components?

Both render props and higher-order components render only a single child but in most of the cases Hooks are a simpler way to serve this by reducing nesting in your tree.

Khushi_Kumari

How to use styles in React?

The style attribute accepts a JavaScript object with camelCased properties rather than a CSS string. This is consistent with the DOM style JavaScript property, is more efficient, and prevents XSS security holes.

```
const divStyle = {
    color: "blue",
    backgroundImage: "url(" + imgUrl + ")",
  };
```

```
function HelloWorldComponent() {
    return <div style={divStyle}>Hello World!</
div>;
  }
```

Style keys are camelCased in order to be consistent with accessing the properties on DOM nodes in JavaScript (e.g. node.style.backgroundImage).

What are the <Router> components of React Router v6?

React Router v6 provides below 4 <Router> components:

1. **<BrowserRouter>**:Uses the HTML5 history API for standard web apps.
2. **<HashRouter>**:Uses hash-based routing for static servers.
3. **<MemoryRouter>**:Uses in-memory routing for testing and non-browser environments.
4. **<StaticRouter>**:Provides static routing for server-side rendering (SSR).

The above components will create browser, hash, memory and static history instances. React Router v6 makes the properties and methods of the history instance associated with your router available through the context in the router object.

What is the purpose of push() and replace() methods of history?

A history instance has two methods for navigation purpose.

1. push()
2. replace()

If you think of the history as an array of visited locations, push() will add a new location to the array and replace() will replace the current location in the array with the new one.

How to conditionally apply class attributes?

You shouldn't use curly braces inside quotes because it is going to be evaluated as a string.

```
<div className="btn-panel {this.props.visible ? 'show' : 'hidden'}">
```

Instead you need to move curly braces outside (don't forget to include spaces between class names):

```
<div className={'btn-panel ' + (this.props.visible ? 'show' : 'hidden')}>
```

Template strings will also work:

```
<div className={`btn-panel ${this.props.visible ? 'show' : 'hidden'}`}>
```

How React Router is different from history library?

React Router is a wrapper around the history library which handles interaction with the browser's window.history with its browser and hash histories. It also provides memory history which is useful for environments that don't have global history, such as mobile app development (React Native) and unit testing with Node.

What are the advantages of React?

Below are the list of main advantages of React,

1. Increases the application's performance with Virtual DOM.
2. JSX makes code easy to read and write.
3. It renders both on client and server side (SSR).
4. Easy to integrate with frameworks (Angular, Backbone) since it is only a view library.
5. Easy to write unit and integration tests with tools such as Jest.

What are the limitations of React?

Apart from the advantages, there are few limitations of React too,

1. React is just a view library, not a full framework.
2. There is a learning curve for beginners who are new to web development.
3. Integrating React into a traditional MVC framework requires some additional configuration.
4. The code complexity increases with inline templating and JSX.
5. Too many smaller components leading to over engineering or boilerplate.

Hey! I'm Khushi,
Here to share programming related tips and tricks to help you enhance your coding skills.

Follow Me To
Enhance Your Coding Skills.

Khushi._.exe Khushi._Kumari Khushi Kumari

Which question do you think is essential for every React interview?