FullStack.Cafe - Never Fail Tech Interview (https://www.fullstack.cafe)

39 Advanced React Interview Questions You Must Clarify (2020 Update)

Originally published on 39 Advanced React Interview Questions You Must Clarify (2020 Update) | FullStack.Cafe (https://www.fullstack.cafe\blog\react-js-interview-questions)

- Q1 : What is virtual DOM?
- Q2 : What are the differences between a class component and functional component?
- Q3 : What are refs used for in React?
- Q4 : Describe how events are handled in React.
- Q5 : What is the difference between state and props?
- Q6 : How to create refs?
- Q7 : What are Higher-Order components?
- Q8 : What is the purpose of using super constructor with props argument?
- Q9 : What are controlled components?
- Q10 : What is equivalent of the following using React.createElement?
- Q11 : What can you tell me about JSX?
- Q12 : Given the code defined above, can you identify two problems?
- Q13 : Why should not we update the state directly?
- Q14 : What are the different phases of ReactJS component lifecycle?
- Q15 : What are the lifecycle methods of ReactJS?
- Q16 : What do these three dots (...) in React do?

What does the ... do in this React (using JSX) code and what is it called?

```
<Modal {...this.props} title='Modal heading' animation={fal
```

- Q17 : What are advantages of using React Hooks?
- Q18 : What are React Hooks?
- Q19 : What is useState() in React?

Explain what is the use of useState(0) there:

```
...
const [count, setCounter] = useState(0);
const [moreStuff, setMoreStuff] = useState(...);
...

const setCount = () => {
   setCounter(count + 1);
   setMoreStuff(...);
   ...
};
```

Q20 : What is StrictMode in React?

Q21 : Why do class methods need to be bound to a class instance?

Q22 : What is prop drilling and how can you avoid it?

Q23 : Describe Flux vs MVC?

Q24 : What is the difference between a controlled component and an uncontrolled component?

Q25 : What is wrong with this code?

Q26 : What is the React context?

Q27 : What is React Fiber?

Q28 : How to apply validation on Props in ReactJS?

Q29 : What is the difference between ReactJS and Angular?

Q30 : What is the difference between using constructor vs getInitialState in React?

Q31 : When is it important to pass props to super(), and why?

Q32 : How to conditionally add attributes to React components?

Is there a way to only add attributes to a React component if a certain condition is met?

Q33 : Do Hooks replace render props and higher-order components?

Q34 : How would you go about investigating slow React application rendering?

Q35 : When would you use StrictMode component in React?

Q36 : What is a pure function?

Q37 : How does React renderer work exactly when we call setState?

Q38 : What is the key architectural difference between a JavaScript library such as React and a JavaScript framework such as Angular?

Q39 : How to avoid the need for binding in React?

Answers

Q1: What is virtual DOM? ★

Topic: React

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.

Q2: What are the differences between a class component and functional component? ★★

Topic: React

> Class components allows you to use additional features such as local state and lifecycle hooks. Also, to enable your component to have direct access to your store and thus holds state.
>
> When your component just receives props and renders them to the page, this is a stateless component, for which a pure function can be used. These are also called dumb components or presentational components.

Q3: What are refs used for in React? ★★

Topic: React

Refs are an escape hatch which allow you to get direct access to a DOM element or an instance of a component. In order to use them you add a ref attribute to your component whose value is a callback function which will receive the underlying DOM element or the mounted instance of the component as its first argument.

```
class UnControlledForm extends Component {
  handleSubmit = () => {
    console.log("Input Value: ", this.input.value)
  }
  render () {
    return (
      <form onSubmit={this.handleSubmit}>
        <input
          type='text'
          ref={(input) => this.input = input} />
        <button type='submit'>Submit</button>
      </form>
    )
  }
}
```

Above notice that our input field has a ref attribute whose value is a function. That function receives the actual DOM element of input which we then put on the instance in order to have access to it inside of the handleSubmit function.

It’s often misconstrued that you need to use a class component in order to use refs, but refs can also be used with functional components by leveraging closures in JavaScript.

```
function CustomForm ({handleSubmit}) {
  let inputElement
  return (
    <form onSubmit={() => handleSubmit(inputElement.value)}>
      <input
        type='text'
        ref={(input) => inputElement = input} />
      <button type='submit'>Submit</button>
    </form>
  )
}
```

Q4: Describe how events are handled in React. ★★

Topic: React

In order to solve cross browser compatibility issues, your event handlers in React will be passed instances of SyntheticEvent, which is React's cross-browser wrapper around the browser's native event. These synthetic events have the same interface as native events you're used to, except they work identically across all browsers.

What's mildly interesting is that React doesn't actually attach events to the child nodes themselves. React will listen to all events at the top level using a single event listener. This is good for performance and it also means that React doesn't need to worry about keeping track of event listeners when updating the DOM.

Q5: What is the difference between state and props? ★★

Topic: React

Both props and state are plain JavaScript objects. While both of them hold information that influences the output of render, they are different in their functionality with respect to component. i.e,

Props get passed to the component similar to function parameters
state is managed within the component similar to variables declared within a function.

Q6: How to create refs? ★★

Topic: React

Refs are created using React.createRef() method and attached to React elements via the ref attribute. In order to use refs throughout the component, just assign the ref to the instance property with in constructor.

```
class MyComponent extends React.Component {
  constructor(props) {
    super(props);
    this.myRef = React.createRef();
  }
  render() {
    return <div ref={this.myRef} />;
  }
}
```

And:

of 14

of 14

of 14

of 14

No ratings yet

Document • 252 pages

React Questions

Cricket Australia

No ratings yet

Document • 157 pages

300+ React Js Interview Questions With Answers

Mukesh Mangal

100% (1)

Document • 24 pages

React JS Notes

amitfegade121

100% (5)

Document • 303 pages

React Book - Freebie

Codin Moldovanu

100% (8)

Document • 191 pages

Merged PDF React Questions

Yorch Canul

No ratings yet

From Everand

50 Recipes for Programming Node.js

Jamie Munro

★★★☆☆ 3/5 (4)

From Everand

React.js Essentials

Artemij Fedosejev

★★★★☆ 4/5 (9)

of 14