40 ReactJS Advanced Interview Questions: Get Hired in 2024:

Table of Contents

ReactJS Interview Questions on Components

ReactJS Redux Interview Questions

ReactJS Router Questions

ReactJS Styling Questions

Choose The Right Software

Development Program

Conclusion

Most Asked ReactJS Interview Questions

1. What is ReactJS?
2. Why ReactJS is Used?
3. How Does ReactJS work?

4. What are the features of ReactJS?
5. What is JSX?
6. How to create components in ReactJS?
7. What are the advantages of ReactJS?

8. Differentiate between real DOM and virtual DOM?
9. What are forms in ReactJS?
10. How is React different from React Native?

Basic Level - ReactJS Interview Questions

Here are some React Interview Questions on basic concepts.

1. What are the features of React?

JSX: JSX is a
extension to Ja
used with Rea
what the user
look like. By us

	can write HTM the same file t contains JavaS
	Components: the building bl React applicat single app usu multiple compo the user interfa independent, r

	that can be pro separately.
	Virtual DOM: F lightweight rep the real DOM i and that is kno virtual DOM. W of an object ch DOM changes

	in the real DO updating all th
	One-way data binding: React binding keeps modular and fa unidirectional that when des app, you often

	components w components.
	High performa updates only t components th changed, rathe all the compon This results in web applicatio

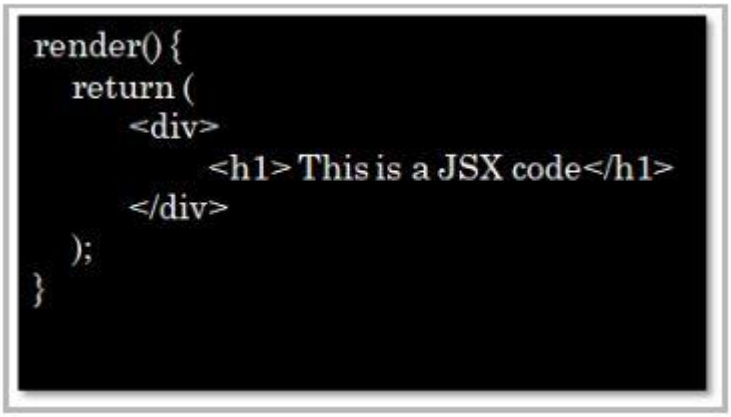
Want a Top Software Development Job? Start Here!

Full Stack Developer - MERN StackExplore Program

2. What is JSX?

JSX is a syntax extension of

JavaScript. It is used with React to describe what the user interface should look like. By using JSX, we can write HTML structures in the same file that contains JavaScript code.



3. Can web browsers read JSX directly?

. Web browsers cannot read JSX directly. This is because they are built to only read regular JS objects

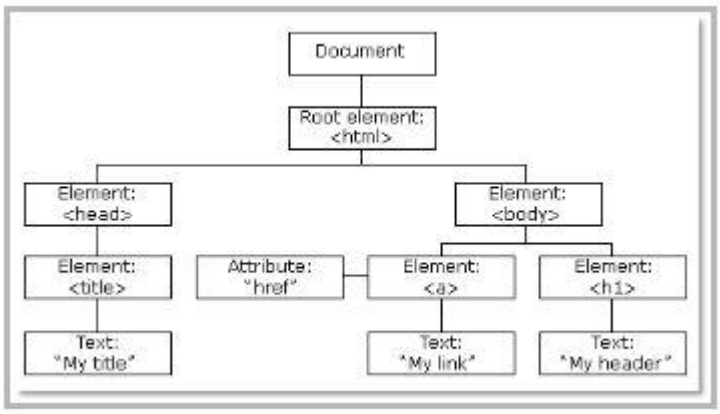
and JSX is not a regular JavaScript object

- For a web browser to read a JSX file, the file needs to be transformed into a regular JavaScript object. For this, we use Babel

4. What is the virtual DOM?

DOM stands for Document Object Model. The DOM represents an HTML document with a logical tree structure. Each branch of the

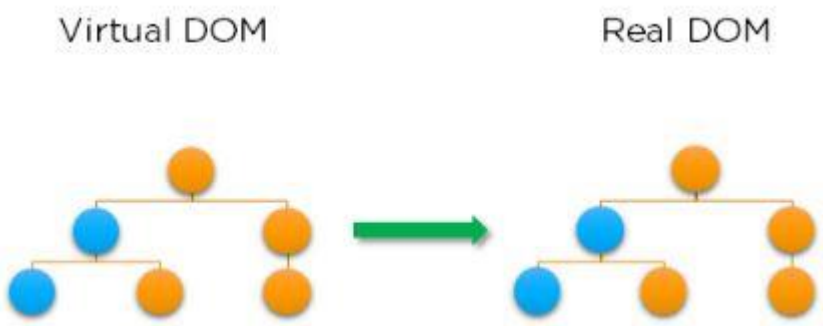
tree ends in a node, and each node contains objects.



React keeps a lightweight representation of the real DOM in the memory, and that is

known as the virtual DOM. When the state of an object changes, the virtual DOM changes only that object in the real DOM, rather than updating all the objects. The following are some of the most

frequently asked react interview questions.



Want a Top Software Development Job? Start Here!

Full Stack Developer - MERN

StackExplore Program

5. Why use React instead of other frameworks, like Angular?

Easy creation of dy
applications: React
it easier to create dy

	web applications be it provides less codi and provides more functionality, whereа with JavaScript applications, code te to get complex very quickly.
	Improved performance: React

virtual DOM, which
web applications pe
faster. Virtual DOM
compares its previo
state and updates o
those components i
real DOM, whose st
have changed, rathe
than updating all the
components — like

	conventional web applications.
	Reusable components: Compo are the building bloc any React applicatio and a single app us consists of multiple components. These components have th

	own logic and contro and they can be reu through the applicat which, in turn, dramatically reduces development time o application.
	Unidirectional data flow: React follows a unidirectional data f

This means that wh
designing a React a
we often nest child
components within p
components. And si
the data flows in a s
direction, it become
easier to debug erro
and know where the
problem occurs in a

	application at the moment.
	Dedicated tools for d debugging: Faceboo released a chrome extension that we ca to debug React applications. This m the process of debu

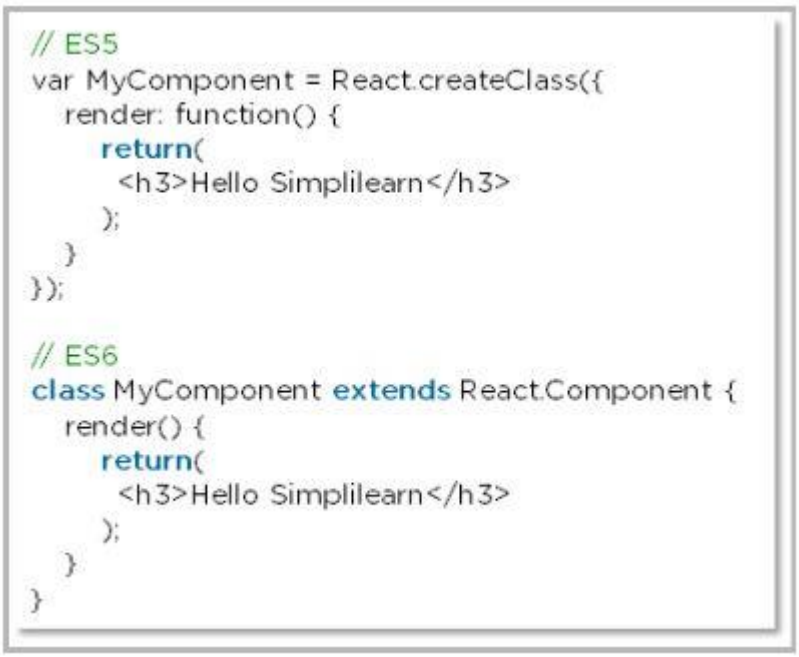
	React to web applic faster and easier.

6. What is the difference between the ES6 and ES5 standards?

This is one of the most frequently asked react interview questions.

These are the few instances where ES6 syntax has changed from ES5 syntax:

. Components and Function



```
// ES5
var MyComponent = React.createClass({
   render: function() {
      return(
        <h3>Hello Simplilearn</h3>
      );
   }
});

// ES6
class MyComponent extends React.Component {
   render() {
      return(
        <h3>Hello Simplilearn</h3>
      );
   }
}
```

. exports vs export

```
sqoop export --connect
jdbc:mysql://localhost/retail_db -username
root --password cloudera --table dept --
export-dir /user/cloudera/departments
```

. require vs import

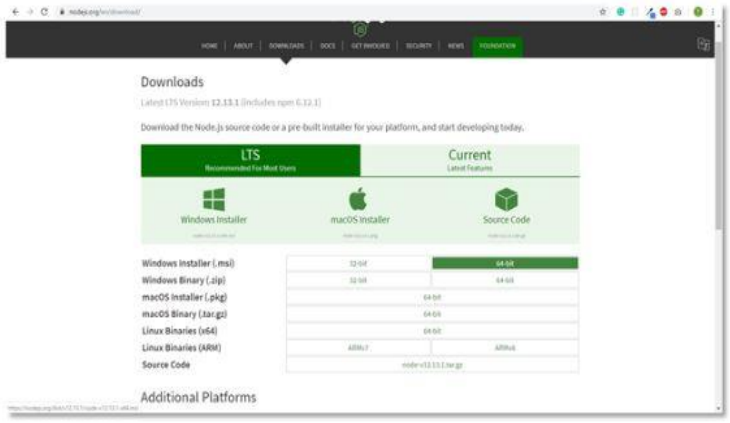
```
// ES5
var React = require('react');

// ES6
import React from 'react';
```

7. How do you create a React app?

These are the steps for creating a React app:

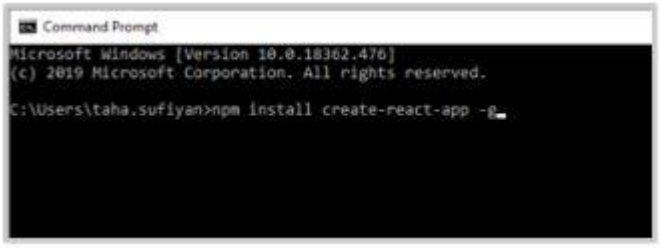
. Install NodeJS on the computer because we need npm to install the React library. Npm is the node package manager that contains many

JavaScript libraries, including React.

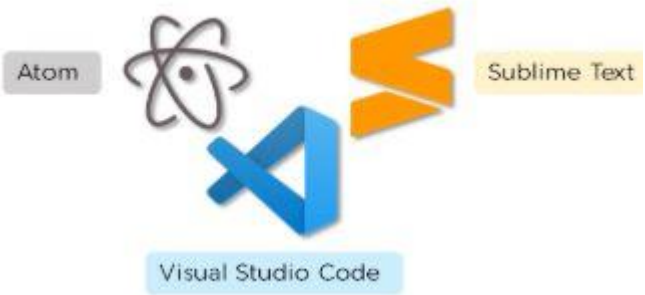


. Install the create-react-app package using the

command prompt or terminal.



. Install a text editor of your choice, like VS Code or Sublime Text.



We have put together a set of Node.js interview questions in case you would like to explore them. Please note, This is one of the most frequently asked react interview questions.

8. What is an event in React?

An event is an action that a user or system may trigger, such as pressing a key, a mouse click, etc.

- React events are named using camelCase, rather

than lowercase in HTML.

- With JSX, you pass a function as the event handler, rather than a string in HTML.

```
<Button
onPress={lightItUp} />
```

Want a Top Software Development Job? Start Here!

Full Stack Developer - MERN StackExplore Program

9. How do you create an event in React?

A React event can be created by doing the following:

```
class Simple extends React.Component {
  work() {
    alert("Good Work!");
  }
  render() {
    return (
      <button onClick={this.work}>Do some work!</button>
    );
  }
}
```

10. What are synthetic events in React?

. Synthetic events combine the

response of different browser's native events into one API, ensuring that the events are consistent across different browsers.

. The application is consistent regardless of the browser it is

running in.

Here, preventDefault is a synthetic event.

```
function ActionLink() {
  function handleClick(e) {
    e.preventDefault();
    console.log('You just clicked a Link.');
  }
  return (
    <a href="#" onClick={handleClick}>
      Click_Me
    </a>
  );
}
```

11. Explain how lists work in React

. We create lists in React as we do in

regular JavaScript. Lists display data in an ordered format

. The traversal of lists is done using the map() function

```
const names = ['Kohli', 'Saif', 'Arun', 'Aamir', 'Arif'];

const listOfNames = () => {
   const listItems = names.map((name) =>
        <li key={name}>
        {name}
        </li>
   );
   return (
        <ul>{listItems}</ul>
   );
}
```

12. Why is there a need for using keys in Lists?

Keys are very important in lists for the following reasons:

. A key is a unique identifier and it is used to identify which items have

changed, been updated or deleted from the lists

. It also helps to determine which components need to be re-rendered instead of re-rendering all the components every time. Therefore, it

increases performance, as only the updated components are re-rendered

13. What are forms in React?

React employs forms to enable users to interact with web applications.

. Using forms, users can interact with the application and enter the required information whenever needed. Form contain certain elements, such as text fields, buttons,

checkboxes, radio buttons, etc

- Forms are used for many different tasks such as user authentication, searching, filtering, indexing, etc

14. How do you create forms in React?

We create forms in React by doing the following:

```
class NameForm extends React.Component {
  this.state = {value: ''};

  handleChange(event) {
    this.setState({value:  event.target.value});
  }

  handleSubmit(event) {
    alert('A name was entered: ' + this.state.value);
    event.preventDefault();
  }

  render() {
    return (
      <form onSubmit={this.handleSubmit.bind(this)}>
        <label>
          Name:
          <input type="text"  value={this.state.value}
onChange={this.handleChange.bind(this)} />
        </label>
        <input type="submit"  value="Submit" />
      </form>
    );
  }
}
```

The above code will yield an input field

with the
label Name and a
submit button. It will
also alert the user
when the submit
button is pressed.

15. How do you write
comments in React?

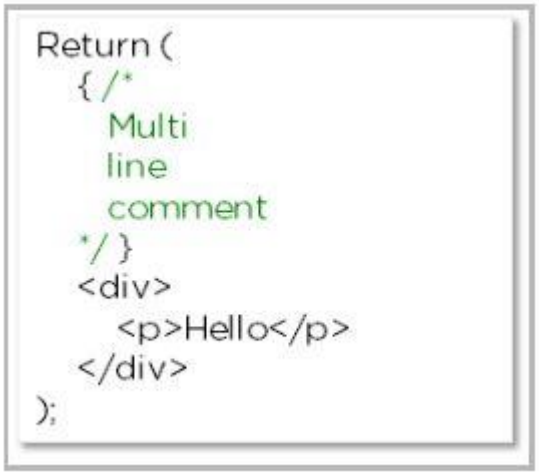
There are basically
two ways in which

we can write comments:

. Single-line comments

```
In [8]: #Returns sum of two values
        def sum(a, b):
            return a + b

        x = sum(4, 7)
        print(x)

        11
```

. Multi-line comments



16. What is an arrow function and how is it used in React?

. An arrow function is a short way of writing a function to React.

. It is unnecessary to bind ‘this’ inside the constructor when using an arrow function. This prevents bugs caused by the use of ‘this’ in React callbacks.

Without Arrow function

```
render() {
  return(
     <MyInput onChange={this.handleChange.bind(this) } />
  );
}
```

With Arrow function

```
render() {
  return(
     <MyInput onChange={ (e) => this.handleOnChange(e) } />
  );
}
```

17. How is React different from React Native?

	React

Release	2013
Platform	Web
HTML	Yes
CSS	Yes

Prerequisites	JavaScript, HTML, CSS

18. How is React different from Angular?

	Angular
Author	Google

Architecture	Complete MVC
DOM	Real DOM
Data-Binding	Bi-direction
Rendering	Client-Side

Performance	Comparati slow

In case you have any doubts about these Basic React interview questions and answers, please leave your questions

in the comment section below.

Want a Top Software Development Job? Start Here!

Full Stack Developer - MERN StackExplore Program

ReactJS Interview Questions on Components

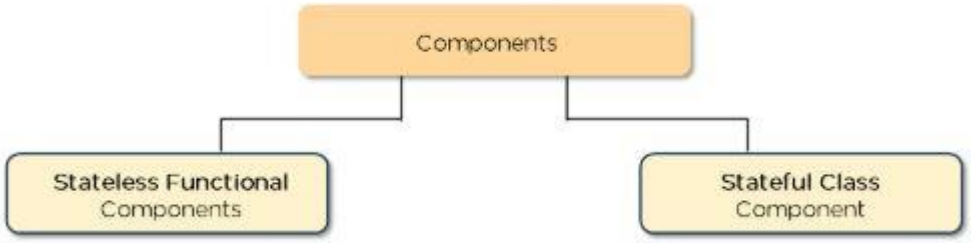
Here are some React Interview Questions on components.

19. What are the components in React?

Components are the building blocks of any React application, and a single app usually consists of multiple components. A component is essentially a piece of the user interface. It splits the user

interface into independent, reusable parts that can be processed separately.

There are two types of components in React:



. Functional Components: Thes e types of components have no state of their own and only contain render methods, and therefore are also called stateless components. They

may derive data from other components as props (properties).

```
function Greeting(props) {
  return <h1>Welcome to {props.name}</h1>;
}
```

. Class Components: Thes e types of components can hold and manage their own state and have a separate render method to return JSX on the screen. They are also called Stateful

components as they can have a state.

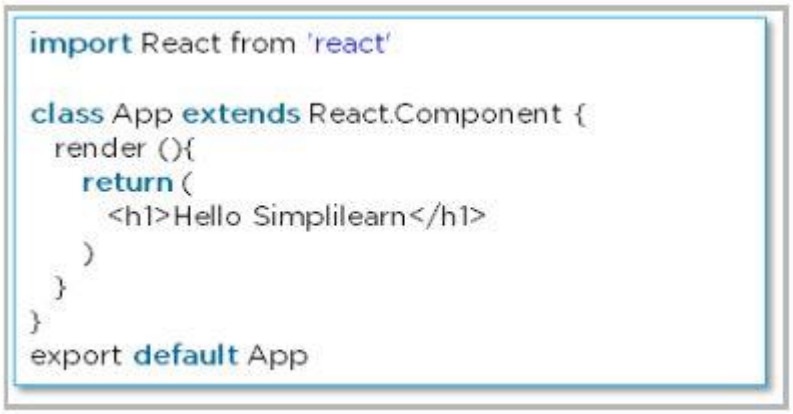
```
class Greeting extends React.Component {
  render() {
    return <h1>Welcome to {this.props.name}</h1>;
  }
```

```
}
```

20. What is the use of render() in React?

. It is required for each component to have a render() function. This function returns the HTML, which is to be displayed in the component.

. If you need to render more than one element, all of the elements must be inside one parent tag like , <form>.



```
import React from 'react'

class App extends React.Component {
  render (){
    return (
      <h1>Hello Simplilearn</h1>
    )
  }
}
export default App
```

21. What is a state in React?

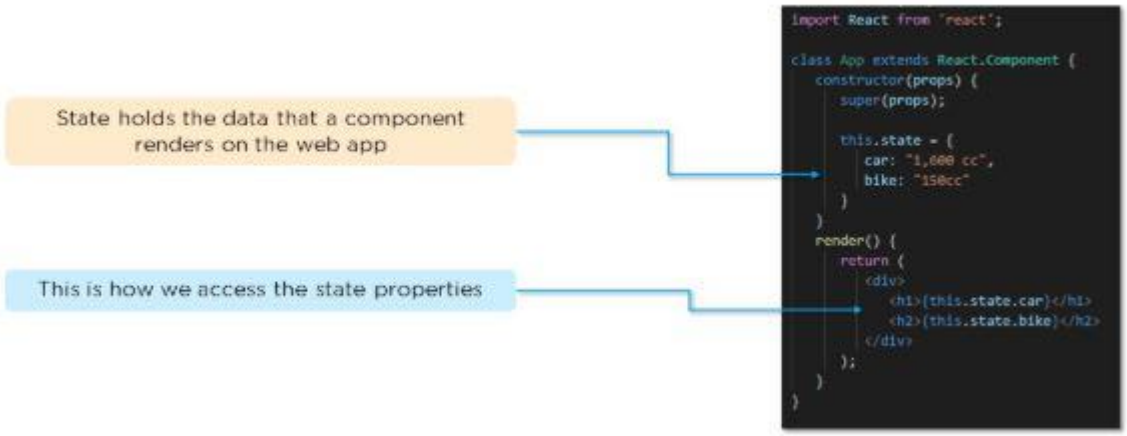
. The state is a built-in React object that is used to contain data or information about the component. The state in a component can change over time,

and whenever it changes, the component re-renders.

- The change in state can happen as a response to user action or system-generated events. It determines the

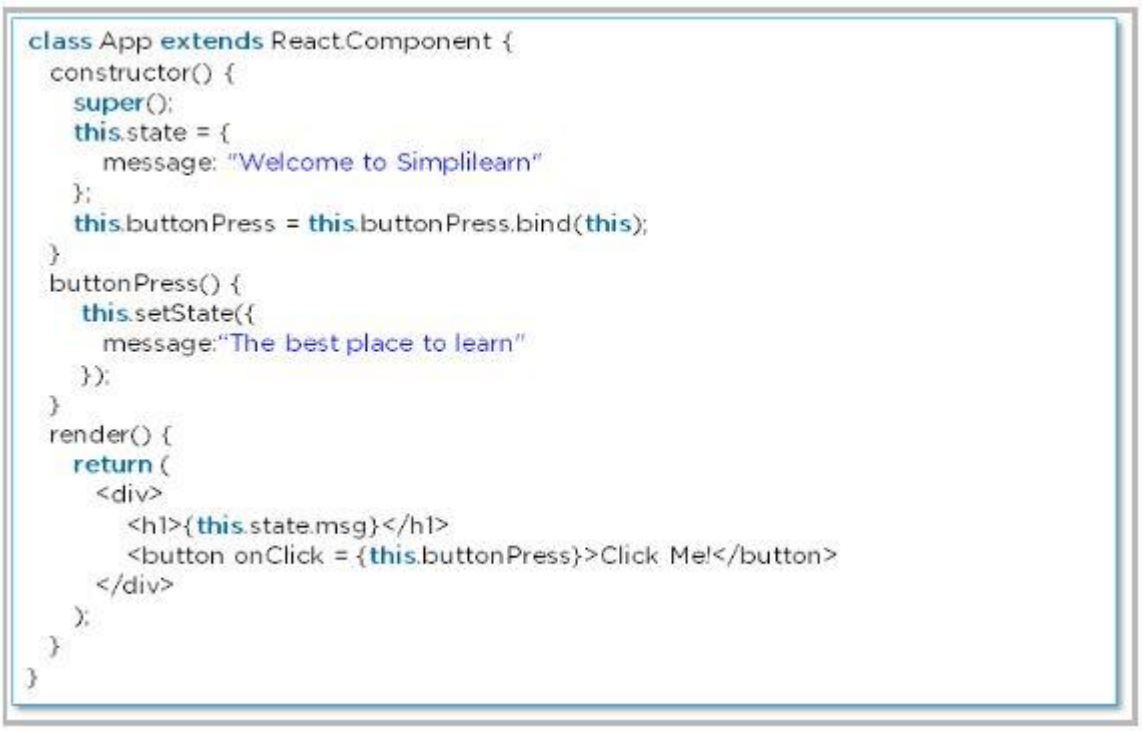
behavior of the component and how it will render.

22. How do you implement state in React?



23. How do you update the state of a component?

We can update the state of a component by using the built-in ‘setState()’ method:



```
class App extends React.Component {
  constructor() {
    super();
    this.state = {
      message: "Welcome to Simplilearn"
    };
    this.buttonPress = this.buttonPress.bind(this);
  }
  buttonPress() {
    this.setState({
      message:"The best place to learn"
    });
  }
  render() {
    return (
      <div>
        <h1>{this.state.msg}</h1>
        <button onClick ={this.buttonPress}>Click Me!</button>
      </div>
    );
  }
}
```

24. What are props in React?

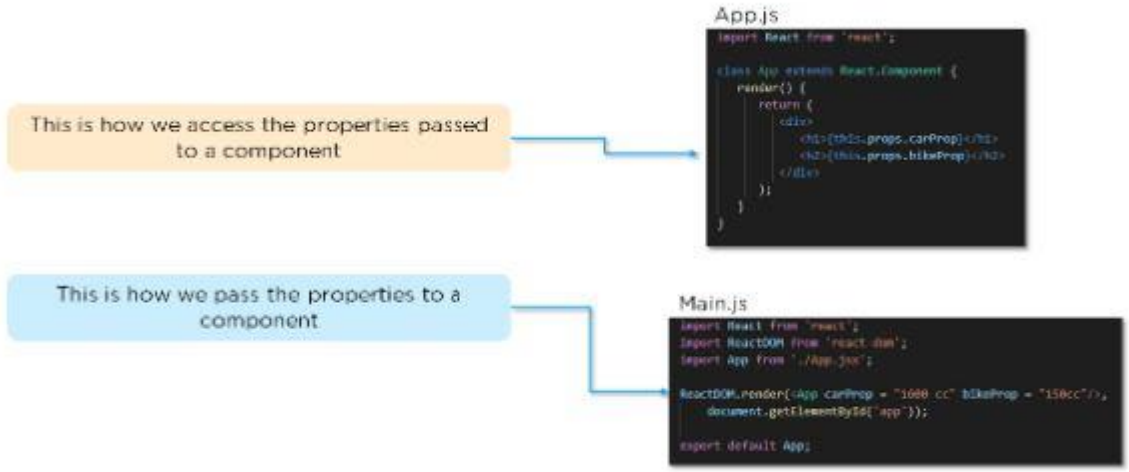
. Props are short for Properties. It is a React built-in object that stores

the value of attributes of a tag and works similarly to HTML attributes.

- Props provide a way to pass data from one component to another component. Props are passed to the

component in the same way as arguments are passed in a function.

25. How do you pass props between components?



Want a Top Software Development Job? Start Here!

Full Stack Developer - MERN StackExplore Program

26. What are the differences between state and props?

	State
Use	Holds information about the component

Mutability	Is mutable
Read-Only	Can be changed
Child components	Child component cannot access

Stateless components	Cannot have state

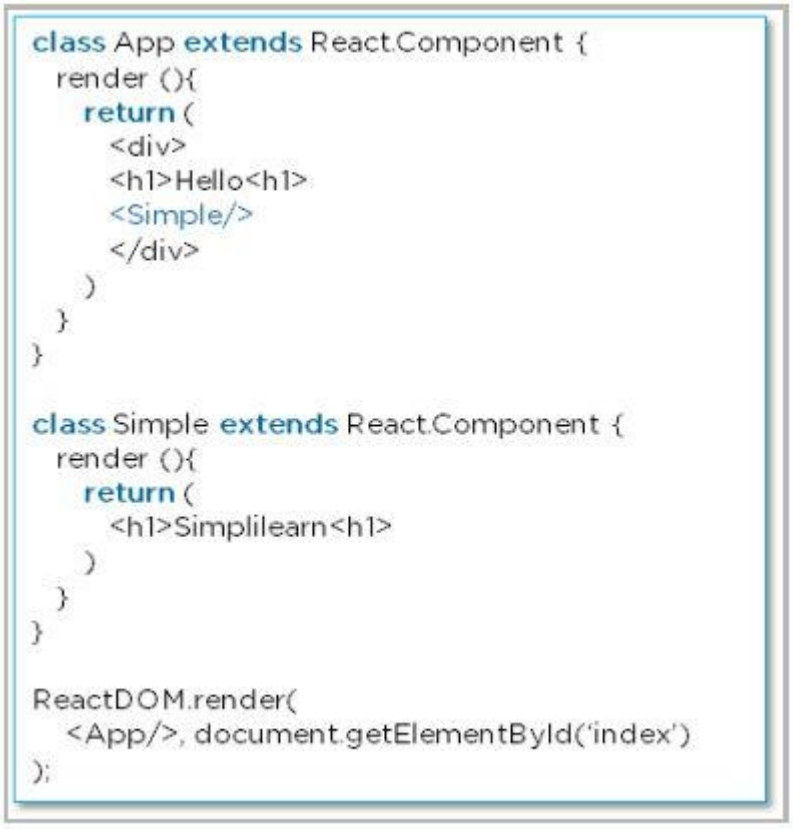
27. What is a higher-order component in React?

A higher-order component acts as a container for other components. This helps to keep components simple

and enables re-usability. They are generally used when multiple components have to use a common logic.

28. How can you embed two or more components into one?

We can embed two or more components into one using this method:



```
class App extends React.Component {
  render (){
    return (
      <div>
      <h1>Hello<h1>
      <Simple/>
      </div>
    )
  }
}

class Simple extends React.Component {
  render (){
    return (
      <h1>Simplilearn<h1>
    )
  }
}

ReactDOM.render(
  <App/>, document.getElementById('index')
);
```

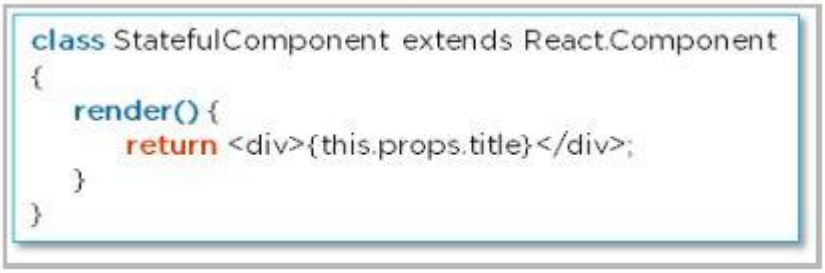
29. What are the differences between

class and functional components?

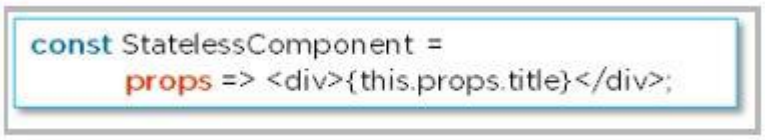
	Class Components
State	Can hold or manage state

Simplicity	Complex as compared to the stateless component
Lifecycle methods	Can work with all lifecycle methods
Reusability	Can be reused

. Class components example:



. Functional components example:



30. Explain the lifecycle methods of components.

- getInitialState(): This is executed before the creation of the component.
- componentDidMount(): Is executed when the component gets rendered and placed on the DOM.

. shouldComponent Update(): Is invoked when a component determines changes to the DOM and returns a “true” or “false” value based on certain conditions.

. componentDidUpdate(): Is invoked immediately after rendering takes place.
. componentWillUnmount(): Is invoked immediately before a component is destroyed and

unmounted permanently.

So far, if you have any doubts about the above React interview questions and answers, please ask your questions in the section below.

ReactJS Redux Interview Questions

Here are some ReactJS Interview Questions on the ReactJS Redux concept.

31. What is Redux?

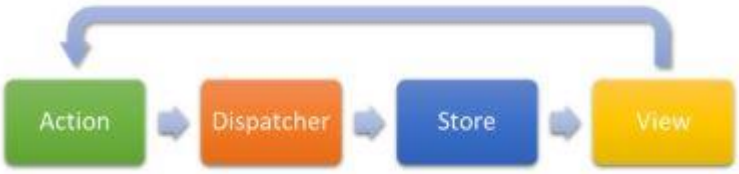
Redux is an open-source, JavaScript library used to manage the application state.

React uses Redux to build the user interface. It is a predictable state container for JavaScript applications and is used for the entire application’s state management.

32. What are the components of Redux?

. Store: Holds the state of the application.
. Action: The source information for the store.
. Reducer: Specifies how the

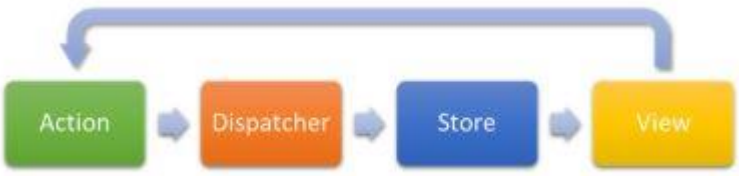
application's state changes in response to actions sent to the store.



33. What is the Flux?

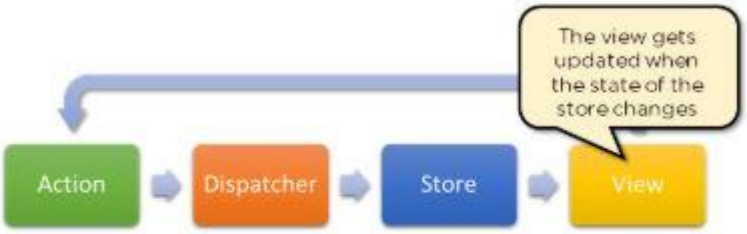
. Flux is the application architecture that

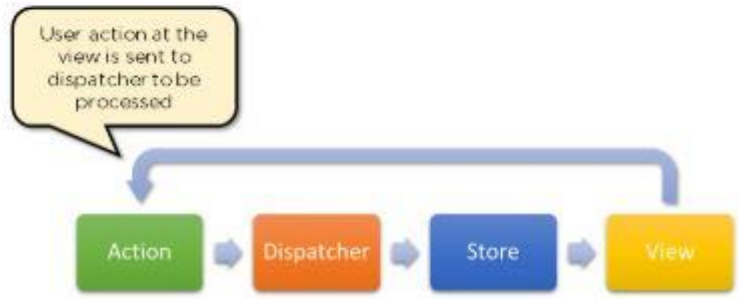
Facebook uses for building web applications. It is a method of handling complex data inside a client-side application and manages how data flows in a React application.



. There is a single source of data (the store) and triggering certain actions is the only way way to update them.The actions call the dispatcher, and then the store is triggered and updated with their

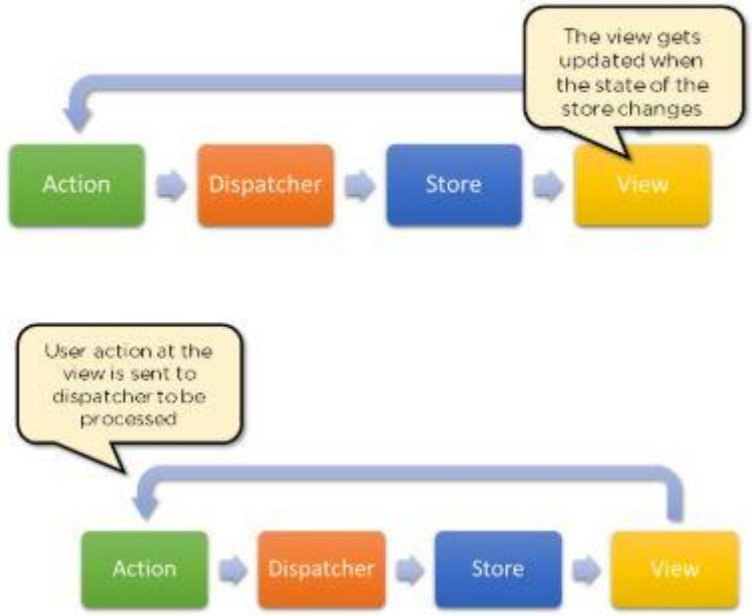
own data accordingly.





. When a dispatch has been triggered, and the store updates, it will emit a change event

that the views can rerender accordingly.



34. How is Redux different from Flux?

SN	Redux	Flux

1.	Redux is an open-source JavaScript library used to manage application State	Flux is a architect and not a framewo or library

2.	Store’s state is immutable	Store’s state is mutable
3.	Can only have a single-store	Can have multiple stores
4.	Uses the concept of reducer	Uses the concept

		the dispatch

So far, if you have any doubts about these React interview questions and answers, please leave your questions in the section below.

ReactJS Router Questions

Here are some ReactJS Interview Questions on React Router concepts.

35. What is React Router?

React Router is a routing library built on top of React, which is used to create routes in a

React application. This is one of the most frequently asked react interview questions.

36. Why do we need to React Router?

. It maintains consistent structure and behavior and is

used to develop single-page web applications.

. Enables multiple views in a single application by defining multiple routes in the React application.

37. How is React routing different from

conventional routing?

SN	React Routing	Conve routing
1.	Single HTML page	Each v a new file
2.	The user navigates multiple	The us naviga multipl

	views in the same file	for eac view
3.	The page does not refresh since it is a single file	The pa refresh every t user naviga
4.	Improved performance	Slower perforr

38. How do you implement React routing?

We can implement routing in our React application using this method:

Considering we have the components App, Ab

out, and Contact in our application:

```
const routing = (
  <Router>
    <div>
      <h1>React Router Example</h1>
      <Route path="/" component={App} />
      <Route path="/about" component={About} />
      <Route path="/contact" component={Contact} />
    </div>
  </Router>
)
```

Hope you have no doubts about this ReactJS interview questions article, in case of any difficulty, please leave your

problems in the section below.

ReactJS Styling Questions

Here are some ReactJS Interview Questions on Styling concept ReactJS.

39. How do you style React components?

There are several ways in which we can style React components:

. Inline Styling

```
class Simple extends React.Component {
 render() {
  return (
   <div>
   <h1 style={{color: "blue"}}>Hello Simple!</h1>
   </div>
  );
 }
}
```

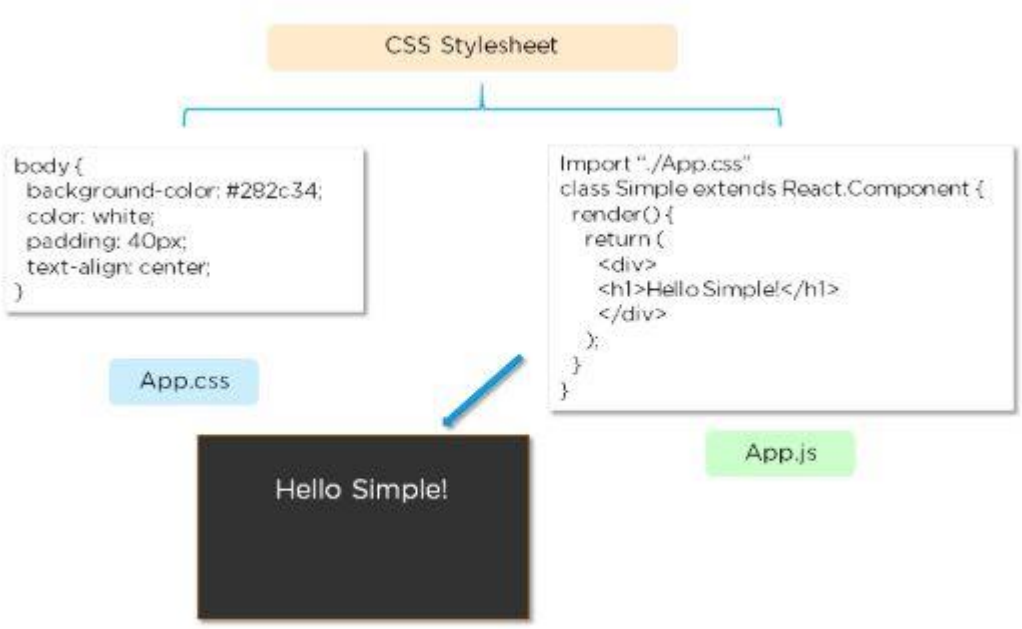
Hello Simple!

. JavaScript Object

```
class Simple extends React.Component {
  render() {
    const simpleStyle = {
      color: "white",
      backgroundColor: "Green",
      margin: "8px",
      fontFamily: "Open Sans"
    };
    return (
      <div>
      <h1 style={simpleStyle}>Hello Simple!</h1>
      </div>
    );
  }
}
```

Hello Simple!

. CSS Stylesheet



40. Explain the use of CSS modules in React.

- The CSS module file is created with the .module.css ext ension
- The CSS inside a module file is available only for the component that

imported it, so there are no naming conflicts while styling the components.

These are all the basic to advanced

ReactJS interview questions that are frequently asked in interviews. We hope these ReactJS interview questions will be helpful in clearing your interview round. All the best for your upcoming job

interview! Suppose you want to learn more about ReactJS components, I suggest you click here!

Choose The Right Software Development Program

This table compares various courses offered by Simplilearn, based on several key features and details. The table provides an overview of the courses' duration, skills you will learn, additional benefits,

among other important factors, to help learners make an informed decision about which course best suits their needs.

gram ne	Full Stack Java Developer	Automa Testing Masters Prograr

o	IN	All
versity	Simplilearn	Simplile
rse ation	6 Months	11 Mon
ing erience uired	Basic Knowledge	Basic Knowle

s You Learn	15+ Skills Including Core Java, SQL, AWS, ReactJS, etc.	Java, AWS, A Testing TDD, et
itional efits	Interview Preparation Exclusive Job Portal	Structur Guidan Learn From

	200+ Hiring Partners	Experts Hands- Training
t	$$	$$
	Explore Program	Explore Progra

Conclusion

Hope this article was able to give you a

better understanding about the different ReactJS interview questions that can be asked in an interview and help you prepare for it better. If you are looking to enhance your software development skills

further, we would highly recommend you to check Simplilearn's Full Stack Java Developer Course. This program can help you hone the right skills and make you job-ready in no time.

Preparing for a ReactJS interview involves not only understanding the intricacies of this popular JavaScript library but also ensuring that your user interface (UI) design skills are up to par. ReactJS is

widely used for building dynamic and interactive user interfaces, making a strong grasp of UI design principles crucial for creating engaging web applications.

To excel in interviews that probe

your ReactJS acumen, a solid foundation in both theoretical concepts and practical application is essential. Engaging with a Full Stack Developer - MERN Stack that encompasses

ReactJS within its curriculum not only prepares you with the requisite knowledge but also equips you with hands-on experience through project-based learning.