



Great Learning > Blog > Interview Questions > Top 50 React JS Interview Questions and Answers for 2024

Top 50 React JS Interview Questions and Answers for 2024

By [Great Learning Team](#) / Updated on Nov 9, 2023 / 13955



React is an open-source JavaScript framework that allows developers to create simple, quick, and scalable online apps. It is efficient and adaptable. React was built by Jordan Walke, a software engineer at Facebook. It was originally used on Facebook's news feed in 2011 and then on Instagram in 2012. React allows developers with a Javascript experience to easily create online applications. React is widely regarded as the most rapidly rising Javascript framework. This blog covers the top React JS Interview Questions that will help you ace your upcoming interview.

Interview Questions ▼

Articles

Interview Questions

Free Courses

Quick Links

Interview Questions

- Company Interview Questions
- Technical Interview Questions

Table of contents

- [What is JSX?](#)
- [What are the advantages of using React?](#)
- [How to create components in React?](#)
- [What is the difference between element and component?](#)
- [What is the difference between functional and class component?](#)
- [Why can't browsers read JSX?](#)
- [What are pure components?](#)

1] What is React?

React is a flexible and efficient open source front-end JavaScript library which was developed by Jordan Walke, a software engineer from Facebook in 2011. It is used for building user interfaces especially for single page applications. It is used for developing view layer of web and mobile apps. React was first deployed on Facebook's News Feed in 2011 and later used in Whatsapp and Instagram.

2] How to install React?

For Installing of React follow below steps:

STEP 1: Installing NPM

Begin the installation of ReactJS by installing NPM (Node Package Manager).

To install NPM on Ubuntu Linux:

```
$ sudo apt install npm
```

- The installation is complete, you can verify the version of NPM installed using the command:

```
$ npm -version
```

The latest version at the time of writing this is v7.6.3 as captured in the output.

Installation of NPM also install Node.JS and check this version.

```
$ node -version
```

- The latest version at the time of writing this is v12.22.1 as captured in the output.

STEP 2: Installing create-react-app utility

- Install the react tool

```
$ sudo npm -g create-react-app
```

where: -g (Globally install)

- Once install the tool then check his version

```
$ create-react-app -version
```

STEP 3: Create first React App

- \$ npx create-react-app app_name
- Created a React app

```
$ cd app_name
```

- Start the application

\$ npm start

3] What are the features of React?

The main features of React are:

- JSX – JSX stands for Javascript XML. It is a syntax used by ReactJS similar to XML or HTML.
- Components – ReactJS application is made up of multiple reusable components, and each component has its own logic and controls.
- Virtual DOM – A VirtualDOM is a depiction of original DOM. React uses VirtualDOM instead of RealDOM.
- One-way Data Binding – React follows unidirectional data flow or one-way data binding.

4] What is JSX?

JSX stands for Javascript XML. It is a React extension which allows us to write Javascript codes similar to XML or HTML. The JSX file makes React application robust which boosts its performance. Like XML or HTML, JSX tags have a tag name, attributes, and children.

Example:

```
class App extends React.Component {  
  render() {  
    return(  
      <div>  
        <h1> { 'HELLO WORLD' } </h1>  
      </div>  
    )  
  }  
}
```

```
}  
}
```

5] What are the advantages of using React?

- Uses VirtualDOM – As name indicates, VirtualDOM is a virtual representation of RealDOM. React uses VirtualDOM to provide the view. Using VirtualDOM increases the efficiency of React applications.
- SEO friendly – React allows developers to develop user engaging interfaces which will be easily navigated in various search engines. It also allows server-side rendering, thus boosts the SEO of an app.
- React is focused and easy-to-learn.
- It boosts productivity and facilitates further maintenance.
- Creating dynamic web applications becomes easier.
- Reusable components.

6] How to create components in React?

The two possible ways to create components are as follows:

i. Function Components – This is the easiest way to create a component. These are pure JavaScript functions that accept props object as parameter and returns React elements.

Example:

```
Function Welcome({msg}) {  
    return <h1> {'Hello, ${msg}'} </h1>
```

```
}
```

ii. Class Components – A class component is a more identified way of defining a React component.

Example:

```
Class Welcome extends React.Component {  
  render() {  
    return <h1> {'Hello, ${this.props.msg}'} </h1>  
  }  
}
```

7] What is the difference between element and component?

Element – An element is an object describing what you would like to project on the screen in terms of the DOM nodes or other components. Elements can contain other elements in their props. Once a component is created , it's never mutated.

Example:

```
<button class="green"> </button>
```

Component – A component is a function or class that accepts an input and returns a React element. It has to keep references to its DOM nodes and to the instances of the child components.

Example:

```
const LogIn = () => (  
  <div>  
    <p> Login </p>  
    <button> Continue </button>  
    <button color = "blue"> Cancel </button>  
  </div>  
) ;
```

8] What is the difference between functional and class component?

Functional Components	Class Components
These are pure JavaScript functions that accept props object as parameter and returns React elements.	A class component is a more identified way of defining a React component.
render() method is not used.	render() method is used.
Also known as stateless components as they simply accept data and display them in the same form.	Also known as stateful components as they implement state and logic.
React lifecycle methods cannot be used.	React lifecycle methods can be used.

9] Why can't browsers read JSX?

Browsers can read JavaScript objects but JSX is not a JavaScript object. Hence, to allow a browser to read JSX, first, we need to modify JSX file into a JavaScript object using JSX transformers and then pass it to the browser.

10] What are pure components?

Pure components are the components whose output is determined only by its input that is props. Pure component is similar to component except that it handles the `shouldComponentUpdate` method.

11] What is state in React?

State of a component is an object that holds some information which will change over the lifetime of the component. State is analogous to props, but is private and fully controlled by the component. i.e, it's not accessible to any component besides the one that owns and sets it.

Example:

```
import React from 'react' ;
class StateFun extends React.Component {
  constructor(props) {
    super(props)
    this.state = {
      message: 'Welcome to React world'
    }
  }
  render() {
    return (
      <div>
        <h1>{this.state.message}</h1>
      </div>
    )
  }
}
```

```
export default StateFun;
```

12] What is props in React?

Props are passed as input to components. They are either single values or objects holding a group of values that are passed to components. The data is passed down from a parent component to a child component.

13] What is the difference between state and props?

Props	State
Props are read-only.	State changes can be asynchronous.
Props cannot be modified.	State can be modified using this.setState .
Props allow you to pass data from one component to other component as an argument.	State holds information about the components.
Props make component reusable.	State cannot make components reusable.
Props are external and are controlled by renders.	State is internal and are controlled by React Component.

14] Why should we not update state directly?

If we update state directly, then it will not re-render the component.

```
//incorrect way  
this.state.msg= 'Hello World'
```

Use `setState()` method instead, as it schedules an update to a component's state object.

```
//correct way  
this.setState({ msg : 'Hello World' })
```

15] What is the Virtual DOM? How Virtual DOM works?

A virtual DOM is a basic JavaScript object which is simply the copy of the real DOM. It is a tree that consists of the elements, attributes and content as objects and their properties. React render function then creates a node tree out of the React components. It then updates this tree in response to the mutations within the data model.

The Virtual DOM works in 3 steps as mentioned below –

1. Whenever any data changes, the whole UI is re-rendered in Virtual DOM representation.
2. Then the difference between the earlier DOM representation and the new one is determined.
3. Once it is determined, the real DOM will be updated with only the things that have changed.

16] What is the purpose of `render()` in React?

Each React component should have a `render()`. It returns a single React element which is the representation of the native DOM component. If more than one HTML element

needs to be rendered, then they must be grouped together inside one enclosing tag such as <form>, <group>, <div> etc. This function must be kept pure that is, it must return an equivalent result each time it is invoked.

17] What are the different lifecycles methods in React?

Each component in React undergoes 3 phases : Mounting, Updating and Unmounting.

- **Mounting –**

There are 4 built-in lifecycles that are called when a component is mounted.

- i. `constructor()` – Constructor method is used to set the initial state of the component.
- ii. `getDerivedStateFromProps()` – This method is called before rendering the elements into the DOM.
- iii. `render()` – This method returns HTML elements.
- iv. `componentDidMount()` – This method is called right after the component is rendered.

- **Updating –** Updates in React occur due to changes in props and state. The following methods are called when a component is updated or re-rendered.

- i. `getDerivedStateFromProps()` – This method is called again when a component is being updated.
- ii. `shouldComponentUpdate()` – This method is called before rendering the component.

iii. `render()` – This method re-renders the HTML inside the DOM.

iv. `getSnapshotBeforeUpdate()` – This method stores previous state of the component.

v. `componentDidUpdate()` – This method is called after the component gets re-rendered.

- **Unmounting** – This is the last phase of React lifecycle.

i. `componentWillUnmount()` – This method is called after the component gets re-rendered.

18] Explain strict mode in React.

Strict mode is a tool for pointing out potential problems in an application. Like Fragment, Strict mode does not render any visible UI. It activates additional checks and warnings for its descendants.

19] What are react hooks?

Hooks are the functions which “*hooks into*” React state. It allows you to use React features without writing a class.

20] What is reconciliation?

Reconciliation is the process through which React updates the DOM. When a component's state changes, React has to decide whether it is necessary to update the DOM. It does this by creating a virtual DOM and comparing it with the current DOM. In this context, the virtual DOM will contain the new state of the component.

21] What are keys in React?

Key is used to identify which items have changed, are added, or are removed. Keys should be given to the elements inside the array .

Example:

```
const numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9];
const listItems = numbers.map((number) =>

  <li key={number.toString()}>
    {number}
  </li>
);
```

22] What is rendering in React?

Rendering is the most important procedure used in frontend development. In React, the `render()` method is the only required method in a class component, and is responsible for describing the view to be rendered to the browser window.

The term “render prop” refers to a technique for sharing code between React components using a prop whose value is a function.

A component with a render prop takes a function that returns a React element and calls it instead of implementing its own render logic.

Example:

```
render() {  
  return (  
    <div style={{ height: '100vh' }} onMouseMove=  
    {this.handleMouseMove}>  
      <h1>Move the mouse around!</h1>  
      <p>The mouse position is ({this.state.x}</p>  
    </div>  
  );  
}
```

23] How to write comments in React?

Two types of comment –

1] Single line comment:

In a single line comment we use double backslash “//”.

Example:

```
export default function App() {  
  return (  
    <div className="App">  
      <h1>Hello </h1>  
      // <h2>Start editing to see some magic happen!</h2>  
    </div>  
  );  
}
```

2] Multiple line comment:

In Multiple line comment we use “ /* --- */ ”.

Example:

```
export default function App() {  
  return (  
    <div className="App">  
      <h1>Hello </h1>  
      <h2>Start editing to see some magic happen!</h2>  
      /*<h2>Start editing to see some magic happen!</h2>  
      <h2>Start editing to see some magic happen!</h2>*/  
    </div>  
  );  
}
```

24] What is context?

Context provides a way to pass data via component tree without passing it to props manually at every level.

For example, authenticated user, locale preference, UI theme need to be accessed in the application by many components.

```
Const {Provider, Consumer} = React.createContext (defaultValue)
```

25] What is the purpose of using super constructor with props argument?

A child class constructor cannot make use of this reference until the `super()` method is invoked. The main reason of using `super()` constructor is to grant access to child constructors to use `this.props` reference.

Example:

```
class Demo extends React.Component {  
  constructor(props) {  
    super(props)  
  
    console.log(this.props) // prints { name: 'John', age: 42 }  
  }  
}
```

26] How to run a React project?

STEP 1: Installing NPM

Begin the installation of ReactJS by installing NPM (Node Package Manager) .

To install NPM on Ubuntu Linux

```
$ sudo apt install npm
```

The installation is complete, you can verify the version of NPM installed using the command:

```
$ npm -version
```

The latest version at the time of writing this is v7.6.3 as captured in the output.

Installation of NPM also install Node.JS and check his version

```
$ node -version
```

The latest version at the time of writing this is v12.22.1 as captured in the output.

STEP 2: Installing create-react-app utility

Install the react tool

```
$ sudo npm -g create-react-app
```

where: -g (Globally install)

Once install the tool then check his version

```
$ create-react-app -version
```

STEP 3: Create first React App

```
$ npx create-react-app app_name
```

Created a React app

```
$ cd app_name
```

Start the application

\$ npm start

27] What is a router in React JS?

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.

28] How to setup React JS?

When you create react app using create-react-app App_name that time generate some folders like as: build/, node_modules/, public/, src/, .gitignore/, README.md, package-lock.json, Package.json.

In these folders, some folders or files are not useable. Such as, .gitignore, package-lock.json and in a public folder, this file is not usable: favicon.ico, logo192.png, manifest.json, logo512.png, robots.txt. Replace your files with this file and use.

29] What is Flux and Redux?

- Flux – Flux is a Javascript pattern for UI which runs on a unidirectional data flow. It is useful when your project has dynamic data and you need to keep the data updated in an effective manner.
- Redux – Redux is a predictable state container for JavaScript apps based on the Flux design pattern. Redux can be used together with React, or with any other view library. It is tiny (about 2kB) and has no dependencies. React Redux is the official React UI bindings layer for Redux. It lets your React components read data from a Redux store, and dispatch actions to the store to update state.

30] How to connect MongoDB with React JS?

React is a front end framework you will not have access to things on your backend such as methods like `db.collection.insert()`. You will in turn have a separation of front end code and back end code. They will talk to each other through HTTP requests (GET, POST, PUT, DELETE).

Example:

```
var mongoose = require('mongoose');
var Message = require('../models/message');

app.post('/message', (request, response) => {
  var newMessage = new Message(request.body);
  newMessage.save((error, doc) => {
    if (error) {
      response.send(error);
    } else {
      response.send(doc);
    }
  });
});
```

31] What is babel in React JS?

- Babel is a toolchain that is mainly used to convert ECMAScript code into a backwards compatible version of JavaScript.
- Babel has support for the latest version of JavaScript through syntax transformers.
- Babel can convert JSX syntax.

32] What is rendered in React JS?

Rendering is a simple technique for sharing a code between one component to another component. The below component uses render prop which returns a React element.

Example:

```
import React from 'react';
class Render extends React.Component {
  constructor(props) {
    super(props);
    this.handleMouseMove = this.handleMouseMove.bind(this);
    this.state = { x: 0 };
  }
  handleMouseMove(event) {
    this.setState({
      x: event.clientX,
    });
  }
  render() {
    return (
      <div style={{ height: '200vh' }} onMouseMove=
{this.handleMouseMove}>
        <h1>Move the mouse around!</h1>
        <p>The current mouse position is ({this.state.x})</p>
      </div>
    );
  }
}

export default Render;
```

33] How to redirect to another page in React JS?

To redirect from one page to another [page in reactJS using `react-router`. The react-router package provides a <Redirect> component in React Router. Rendering a

<Redirect> will navigate to a new location. Like server-side redirects, the new location will override the current location.

Example:

```
import React from 'react';
import { withRouter } from "react-router-dom";

class Route extends Component {
  handleClick = () => {
    this.props.history.push("path/to/push");
  }
  render() {
    return (
      <div>
        <button onClick={this.handleClick} type="button"> Next
Page</button>
      </div>
    );
  }
}

export default Route;
```

34] How to pass data from one component to another in ReactJS?

Using props we can pass data from one component to another component, You can use props to pass data from parent to child component.

Example:

-App.js

--ParentComponent.js

```
import React from 'react'
import ChildComponent from './ChildComponent'; //import childComponent

class ParentComponent extends React.Component {
  render() {
    return(
      <div>
        <ChildComponent message="Data from first component"/>
        //call childComponent
      </div>
    );
  }
}
export default ParentComponent;
```

---Child Component.js

```
import React from 'react';
const ChildComponent = (props) => {
  return(
    <h2> {props.message} </h2>
  );
}
export default ChildComponent; //export childComponent
```

35] How to get the enter key in React js?

In this example you want to use `handleSearch()` using press Enter key.

Example:

```
onKeyPress={
  (event) =>
    event.key === 'Enter' && handleSearch() //check the equality
}
```

36] How to create a table in ReactJS?

Example:

```
import React from 'react';
function App() {
  return (
    <div className="container">
      <h1>Simple Inventory Table</h1>
      <table>
        <thead>
          <tr>
            <th>Product Id    </th>
            <th>Product Name  </th>
            <th>Product Category </th>
            <th>Product Price  </th>
          </tr>
        </thead>
      </table>
    </div>
  );
}
export default App;
```

37] What is npm in React JS?

- NPM (Node Package Manager) is the default package manager for Node.js and is written in Javascript.

- It is the default package manager that comes with NodeJS when you install it.
- npm is an abbreviation used for the node package manager. It is a package manager for JavaScript.

38] How to create a website in React JS?

STEP 1: Create a react app

```
$ npx create-react-app App_name
```

STEP 2: Install the dependencies

STEP 3: Create components

STEP 4: Deploy the application

39] How to install bootstrap in React JS?

The best way to install react- bootstrap via NPM (Node Package Manager).

The command for installing react-bootstrap in react js is below:

```
$ npm install react-bootstrap bootstrap
```

After successfully installation import the bootstrap library

Example –

```
import Button from 'react-bootstrap/Button';
```

```
import { Button } from 'react-bootstrap'; //import bootstrap button
```

40] How to connect react js with a database?

Install MySQL and then import in the react component

Example:

```
import React, { Component } from 'react';
import mysql from 'mysql'; //import mysql

class Users extends Component {
  constructor(props) {
    super(props);
    const connection = mysql.createConnection({
      host: 'localhost',
      user: 'root',
      password: '', //null password
      database: 'react_prac'
    });
    connection.connect();
    console.log(connection);
  }
  render() {
    return (
      <div> Data Base </div>
    );
  }
}
export default Users;
```

41] How to import js files in react?

First you export the js file that means every js file in reactjs is a component, export the file **App.js**.

Example:

```
import React from 'react';
class ExportFile ( () = >{
  render(
    return(
      <h1> Export file</h1>
    )
  )}
Export default ExportFile
```

42] How to set background images in React.js?

- **Using External URL:**

```
function App() {
  return (
    <div style={{
      backgroundImage: `url("https://via.placeholder.com/500")`
    }}>
      //using third party website set image
      Hello World
    </div>
  );
}
```

- **Using Local Machine:**

```
import React from "react";
import background from "../img/image_name.png";
// import image from local storage
```

```
function App() {
  return (
    <div style={{ backgroundImage: `url(${background})` }}>
      //using local storage
      Hello World
    </div>
  );
}
export default App;
```

43] How to check the React JS version?

- When you create a react app that time generates a package.json file .In package.json check dependencies: {} section.
- The second way is:

```
$create-react-app -version
```

44] How to declare a variable in React JS?

Examples:

- `const element = <h1>Hello, world!</h1>; //using JSX`
- `const name = 'Josh Perez';`
- `const element = <h1>Hello, {name}</h1>;`

```
ReactDOM.render(
  element,
  document.getElementById('root')
);
```

45] How to get the enter key in React js?

In this example you want to use `handleSearch()` using press Enter key.

Example:

```
onKeyPress={
  (event) =>
    event.key === 'Enter' && handleSearch()    //check the equality
}
```

46] How to do crud operations in React js?

Reactjs CRUD Application is designed with 2 main layers:

- React.js components let you split the UI into independent, reusable pieces, and think about each piece in isolation.
- Ajax is used by Reactjs component to fetch (post/put/get/delete) data from remote rest api by http request

47] How to debug React JS?

One of the cyclic ways to debug React code on the web using `console.log()`, `console.warn`, `console.error`, and similar statements. you will do `console.log()` ,You will then see the result in the browser inspector.

48] How to get multiple checkbox values in React JS?

Example:

```
import React from 'react';
class const CheckBox = props => {
  return (
    <li>
    <input key={props.id}
      onClick={props.handleClick}
      type="checkbox"
      checked={props.isChecked} value={props.value} />
      {props.value}
    </li>
  )
}
export default CheckBox;
```

49] When to use a Class Component over a Function Component?

If the component needs state or lifecycle methods then use class components. Otherwise use functional components. From React 16.8 with the addition of Hooks, you could use state, lifecycle methods available in class components right in your function component.

Example –

```
1 import React from 'react'
2 class Class_component extends React.Component {
3   render() {
4     return <h1>{`Hello, ${this.props.message}`}</h1>
5   }
6 }
```

50] What is the use of refs?

refs is used to return the reference value to the element.

Sharing is caring:



Great Learning Team

Great Learning's Blog covers the latest developments and innovations in technology that can be leveraged to build rewarding careers. You'll find career guides, tech tutorials and industry news to keep yourself updated with the fast-changing world of tech and business.

Recommended for you

**Top 30+ Embedded C
Interview Question
and Answers 2024**

**Top 60+ Cloud
Computing Interview
Questions (2024)**

**Top 90+ Tableau
Interview Question
and Answers in 2024**

**Top 30 Infosys
Interview Questions |
2024**

**Top 50+ Selenium
Interview Questions
and Answers in 2024**

**Mobile Testing
Interview Questions
and Answers 2024**

Leave a Comment

Your email address will not be published. Required fields are marked *

Type here..



Name*

Email*

Website

☐ Save my name, email, and website in this browser for the next time I comment.

☐ I'm not a robot

reCAPTCHA

[Privacy](#) - [Terms](#)

Post Comment »

Free Courses

Python for Machine Learning

Data Science Foundations

Deep Learning with Python

Introduction to Cyber Security

Introduction to Digital Marketing

Java Programming

[View More →](#)

Blog Categories

Data Science

Artificial Intelligence

Popular Courses

PGP In Data Science and Business Analytics

PGP In Artificial Intelligence And Machine Learning

PGP In Management

PGP In Cloud Computing

Software Engineering Course

PGP In Digital Marketing

[View More →](#)

Salary Blogs

[Salary Calculator](#)

Interview Questions

Java Interview Questions

Python Interview Questions

SQL Interview Questions

Selenium Interview Questions

Machine Learning Interview Questions

NLP Interview Questions

[View More →](#)

Career

Cybersecurity

Full Stack Development

Study Abroad

Study In USA

Data Architect Salary

Cloud Engineer Salary

Software Engineer Salary

Product Manager Salary



[About Us](#) [Contact Us](#) [Privacy Policy](#)

[Terms of Use](#) [Great Learning Careers](#)

© 2013 – 2023 Great Learning Education Services Private Limited

(Formerly known as Great Lakes E-Learning Services Private Limited).All

rights reserved

Get our android app

Get our ios app