

By OnlineInterviewQuestions.com

React Js Interview Questions

What is React.js?

Reactjs is an open-source javascript UI(User Interface) library, collaborate and maintained by Facebook and Instagram and also a huge community of developers. That's designed to take your application state and allow you to write code that makes that status input and return virtual DOM, It's a certainly better way to manipulate the DOM and handle events which are really just a set of the plain.

Every brand wants to build an interactive UI for its mobile users as it engages them for a longer period and results in more conversions. Thus, brands pick the best talent from the developer community and offer them attractive packages. In case you are a developer who is willing to join the industry, here are a few React.js interview questions that answer about all the major elements of React Js. In this post, we have added some Basic and **Advanced React Js Interview Questions**.

Thus, these questions are compiled with both basic and advanced level questions on React Js. Preparing for your interview with these questions, will give you an edge over the others and will help you crack the exam.

Q1. What is ReactJS ?

React js is javascript based UI Library developed at Facebook, to create an interactive, stateful & reusable UI components. It is one of the most popular [javascript frameworks](#) that is created for handling the presentation layer for the web and mobile apps.

Q2. List some advantages of ReactJS ?

Advantages of React Js

- **React.js is extremely efficient:** React.js creates its own virtual DOM where your components actually live. This approach gives you enormous flexibility and amazing gain in performance. React.js also calculates what are the changes needed to be made in DOM. This process of React.js avoids expensive DOM operations and make updates in a very client manner.
- **It makes writing Javascript easier:** React.js uses a special syntax called JSX, which allows you to mix HTML with Javascript. The user can drop a bit of HTML in the render function without having to concatenate strings, this is another fantastic thing. React.js turns those bits of HTML into functions with a special JSXTransformer.
- **It gives you out-of-the-box developer tools:** When you start your journey with React.js, do not forget to install official React.js chrome extension. It makes debugging your application much easier. After you

install the extension, you will have a direct look into the virtual DOM as if you were browsing a regular DOM tree in the elements panel. Isn't it pretty amazing!

- **It's awesome for SEO:** One of the biggest problems with other Javascript frameworks is that they do not search engine friendly. Though there have been some improvements in this area, search engines generally have trouble reading Javascript heavy applications. React.js stands out from the crowd because you can run React.js on the server, and the virtual DOM will be rendered to the browser as a regular web page.
- **UI Test Cases:** It is extremely easy to write UI test cases because the virtual DOM system implemented entirely in JS.

Source: <http://www.pro-tekconsulting.com/blog/advantages-disadvantages-of-react-js/>

Q3. What are Components in ReactJS ?

React Components let you split the UI into independent, reusable pieces, and think about each piece in isolation. Conceptually, components are like JavaScript functions. They accept arbitrary inputs (called “props”) and return React elements describing what should appear on the screen.

Below is ample component written in ES6 class to display a welcome message on the screen.

```
class Welcome extends React.Component {
  render() {
    return <h1>Hello, {this.props.name} </h1>;
  }
}
const element = <Welcome name="Sara" />;
ReactDOM.render(
  element,
  document.getElementById('root')
);
```

source: <https://facebook.github.io/react/docs/components-and-props.html>

Q4. What is JSX in React Js?

JSX is an XML/HTML-like syntax used by React that extends ECMAScript so that XML/HTML-like text can co-exist with JavaScript/React code. The syntax is intended to be used by preprocessors (i.e., transpilers like Babel) to transform HTML-like text found in JavaScript files into standard JavaScript objects that a JavaScript engine will parse.

Basically, by using JSX you can write concise HTML/XML-like structures (e.g., DOM like tree structures) in the same file as you write JavaScript code, then Babel will transform these expressions into actual JavaScript code. Unlike the past, instead of putting JavaScript into HTML, JSX allows us to put HTML into JavaScript.

By using JSX one can write the following JSX/JavaScript code:

```
var nav = (
  <ul id="nav">
```

`Home`

`About`

`Clients`

`Contact Us`

`);`

And Babel will transform it into this:

```
var nav = React.createElement(
  "ul",
  { id: "nav" },
  React.createElement(
    "li",
    null,
    React.createElement(
      "a",
      { href: "#" },
      "Home"
    )
  ),
  React.createElement(
    "li",
    null,
    React.createElement(
      "a",
      { href: "#" },
      "About"
    )
  ),
  React.createElement(
    "li",
    null,
    React.createElement(
      "a",
      { href: "#" },
      "Clients"
    )
  ),
  React.createElement(
    "li",
    null,
    React.createElement(
      "a",
      { href: "#" },
      "Contact Us"
    )
  )
);
```

source:<https://www.reactenlightenment.com/react-jsx/5.1.html>

Q5. Explain life Cycle of React JS Component ?

React JS Component Lifecycle

Each component has several “lifecycle methods” that you can override to run code at particular times in the process. Methods prefixed with will are called right before something happens, and methods prefixed with did are called right after something happens.

Mounting

These methods are called when an instance of a component is being created and inserted into the DOM:

- constructor()
- componentWillMount()
- render()
- componentDidMount()

Updating

An update can be caused by changes to props or state. These methods are called when a component is being re-

rendered:

- componentWillReceiveProps()
- shouldComponentUpdate()
- componentWillUpdate()
- render()
- componentDidUpdate()

Unmounting

This method is called when a component is being removed from the DOM:

- componentWillUnmount()

Other APIs

Each component also provides some other APIs:

- setState()
- forceUpdate()

Class Properties

- defaultProps

- displayName

Instance Properties

- props
- state

Source: <https://facebook.github.io/react/docs/react-component.html>

Q6. List some features of ReactJS ?

Undoubtedly today React is among of one the best JavaScript UI frameworks. It comes with a lot of features that helps programmers to create beautiful application easily, we have listed some of them below.

- It's Adaptability
- Free and Open Source
- Decorators from ES7
- Server-side Communication
- Asynchronous Functions & Generators
- Flux Library
- Destructuring Assignments
- Usefulness of JSX

Also, Read [React Native Interview questions](#)

Q7. How to use Events in ReactJS ?

Using Events in React js is very similar to handling event on DOM elements. The difference is only in syntax like.

1. The name of event in React js is always in camelCase.
2. With JSX you pass a function as the event handler, rather than a string.

Lets understand by an example:

```
// In HTML
<button onclick="activateAccount()">
  Activate Account
</button>
```

```
//In React
<button onClick={activateAccount}>
  Activate Account
</button>
```

Another difference is that in React js you cannot return false to prevent default behavior in React. You must call preventDefault explicitly. Read more from <https://facebook.github.io/react/docs/handling-events.html>

Q8. What is flux in JavaScript?

Flux is an application architecture for creating data layers in JavaScript applications. It was designed at Facebook along with the React view library. Flux is not a framework or a library. It is simply a new kind of architecture that complements React and the concept of Unidirectional Data Flow.

further reading <https://scotch.io/tutorials/getting-to-know-flux-the-react-js-architecture>

Q9. What are refs in ReactJs ?

In React ref is used to store the reference of element or component returned by the component render() configuration function. Refs should be avoided in most cases, however, they can be useful when we need DOM measurements or to add methods to the components.

Refs can be used in the following cases

- Managing focus, text selection, or media playback.
- Triggering imperative animations.
- Integrating with third-party DOM libraries.

Q10. What are stateless components in ReactJS?

Stateless components are components that don't have any state. When something is stateless, it calculates its internal state but it never directly mutates it. For creating a stateless components No class and this keyword is needed. You can create a stateless components using plain functions or Es6 arrow function. Below is an example of stateless component in react.

```
//In Es6 const Pane = (props) =>
```

```
{props.children}
```

```
;
```

```
//In Es5 const Username = ({ username }) =>
```

```
The logged in user is: {username}
```

Q11. What is the difference between State and props in React Js ?

Props are shorthand for properties. they are very similar to an argument is passed to a pure javascript function. Props of the component are passed from parent component which invokes component. During a component's life cycle props should not change consider them as immutable. In React all props can be accessible with this.props.

```
import React from 'react';
class Welcome extends React.Component {
  render() {
    return <h1>Hello {this.props.name}</h1>;
  }
}
```

```

    }
  }
  const element = ;

```

State are used to create dynamic and interactive components in React. State is heart of react component that makes it alive and determines how a component renders & behaves.

```

// simple state example
import React from 'react';
class Button extends React.Component {
  constructor() {
    super();
    this.state = {
      count: 0,
    };
  }
  updateCount() {
    this.setState((prevState, props) => {
      return { count: prevState.count + 1 }
    });
  }
  render() {
    return (<button
      onClick={() => this.updateCount()}
      >
      Clicked {this.state.count} times
    </button>);
  }
}
export default Button;

```

Q12. What are Synthetic events in React Js?

SyntheticEvent is a cross-browser wrapper around browser's native event. In React all of your event handlers will be passed instances of SyntheticEvent. The synthetic event works the same way as the event system of browsers, the only difference is that the same code will work across all browsers.

Below is a simple example of how to listen for a click event in react

```

import React, { Component } from 'react';

class ShowAlert extends Component {

  showAlert() {
    alert("Im an alert");
  }

  render() {
    return (
      <button onClick={this.showAlert}>show alert </button>
    );
  }
}

```

```
export default ShowAlert;
```

Q13. What is the difference between Dom and virtual Dom in ReactJS ?

DOM is the acronym for Document Object Model. Dom is also called HTML DOM as it is an abstraction of structured code called HTML for web developers. Dom and HTML code are interrelated as the elements of HTML are known as nodes of DOM. It defines a structure where users can create, alter, modify documents and the content present in it. So while HTML is a text, DOM is an in-memory representation of this text.

Virtual DOM is an abstraction of abstraction as it has been derived from HTML DOM. It is a representation of DOM objects like a lightweight copy. The virtual DOM was not invented by React, it is only used and provided for free.

Q14. Enlist the advantages and disadvantages of ReactJS ?

React.js is used by web developers for creating large web pages without reloading the entire page. It uses the data and can be changed over time.

The following are the advantages of using React.js-

- 1- React makes Search engine optimization (SEO) easy.
- 2- It is very efficient as it ensures readability and easy maintenance.
- 3- It gives extraordinary developer tools to web developers and makes Java coding easier for them.
- 4- UI test cases.

The following are the disadvantages of React-

- 1- Some major configurations are required for integrating React js with traditional MVC framework such as substituting erb with React js.
- 2- It is a steep learning process for people who are new to web development world.

Q15. What are the controlled components and uncontrolled components in ReactJS ?

Controlled component is more advisable to use as it is easier to implement forms in it. In this, form data are handled by React components. A controlled input accepts values as props and callbacks to change that value.

The **uncontrolled component** is a substitute for controlled components. Here form data is handled by DOM itself. In uncomfortable components, the ref can be used to get the form values from DOM.

Q16. Explain the difference between functional and class components.

Functional components are those components that returns React elements as a result. These are just simple old JavaScript functions. React 0.14 has given a new shortcut for creating simple stateless components that are known as functional components. These components make use of easy JavaScript functions.

Class components – most of the tech-savvy people are more familiar with class components as they have been around the corner for a longer time. These components make use of plain old javascript objects for creating pages, mixins, etc in an identical way. Using React's create a class factory method, a literal is passed in defining the methods of a new component.

Q17. What do you understand by mixin or higher order components in ReactJs ?

Higher order components (HOC) is a function that takes component as well as returns a component. It is a modern technique in React that reuses the component logic. However, Higher order components are not a part of React API, per se. These are patterns that emerge from React's compositional nature. In other words, HOC's are functions that loop over and applies a function to every element in an array.

Q18. How is flux different from redux?

Difference Between FLUX and Redux

FLUX

Flux is a container for application state and logic that are registered to callbacks.

It is an observer pattern that is modified to fit React.

It is a fancy name given to observer pattern and Facebook has developed tools to aid the implementation of these patterns.

Flux supports actions that are simple JavaScript objects.

REDUX

Redux is a container for JavaScript apps.

It offers interesting features such as writing applications, testing in different environments such as a server, client, etc.

In redux, actions can be functions or promises.

Redux is the first choice for web developers because it offers live code editing.

Q19. How is ReactJS different from angular and VUE js?

Angular Js – developed by Google, angular is a typescript based JavaScript application framework. It is also known as Super-heroic JavaScript MVW Framework. It was developed with the motive to encounter the challenges of creating single page applications. There are several versions of angular such as Angular 2+, Angular 2 or ng2. Angular is the rewritten, mostly incompatible successor to AngularJS which means AngularJS is the oldest framework.

React– React was developed by Facebook in March 2013. It is a JavaScript library that is used for building user

interfaces. React creates large web applications and also provides speed, scalability, and simplicity.

Vue Js- Launched in February 2014, Vue is the most famous and rapidly growing framework in JS field. Vue is an intuitive, fast and composable MVVM for building interactive interfaces. It is extremely adaptable and several JavaScript libraries make use of this. Vue is also a web application framework that helps in making advanced single page applications.

Q20. What is the use of arrow function in ReactJS?

Arrow functions are extremely important for React operations. It prevents this bugs. Arrow functions make it easier to predict the behavior of this bugs when passed as callbacks. They don't redefine the value of this within their function body. Hence, prevents bugs caused by the use of this within callbacks.

Q21. What are refs in ReactJS?

Refs are used for managing focus, selecting text and triggering animations. It also integrates with third-party libraries. Refs help in getting the reference to a DOM (Document Object Model) node. They return the node that we are referencing. Whenever we need DOM measurements, we can use refs.

Q22. What is the purpose of render() function in ReactJS?

render() function is used to update the UI. For this, you have to create a new element and send it to ReactDOM.render(). React elements are immutable and once you create an element, you cannot change its attributes. Thus, elements are like a single frame and it depicts the UI at some point. ReactDOM.render() controls the content of the container node you pass and if there is any DOM element already present in it then it would be replaced when first called.

Q23. What is React Router?

React Router is a routing library developed on top of the react which is employed to create the routing in react apps. In single page websites, there is only a single HTML page, we are reusing the same HTML page to render the various components based on the navigation. But in multi-page websites, you will get a totally new page from the server when you operate.

Q24. What is event pooling in react?

In React, event pooling is a performance optimization technique that reduces the number of event objects that are created.

In JavaScript, when an event such as a click or a key press occurs, an event object is created to represent the event. This event object is then passed to any event listeners that are registered to handle the event.

Normally, a new event object is created for every event that occurs. However, in some cases, creating a new event object for every event can be inefficient, especially if the event is occurring frequently. To improve performance in these cases, React uses **event pooling**, which means that it reuses the same event object for multiple events.

Q25. [What are Styled-Components?](#)

Styled-components are a way to style React components using a mixture of JavaScript and CSS. They allow you to write actual CSS code to style your components, but keep that code in the same file as your JavaScript code, rather than in a separate CSS file.

Styled-components are a popular choice for styling React components because they allow you to keep all of your component's styles in the same place as your component's code, which can make it easier to understand and maintain your application. They also offer various features, such as the ability to define and use variables, to make styling your components more flexible and powerful.

```
import styled from 'styled-components';
const Button = styled.button`
  color: palevioletred;
  font-size: 1em;
  margin: 1em;
  padding: 0.25em 1em;
  border: 2px solid palevioletred;
  border-radius: 3px;
`;
function MyApp() {
  return < Button>Click me! </Button>;
}
```

In above example, the **Button component** is created using the **styled** function, which takes a HTML element as an argument and returns a new styled component. The **styled component** is then created by wrapping a template literal (a string surrounded by backticks) containing CSS styles in the styled function.

The Button component can then be used like any other React component, and the styles defined in the template literal will be applied to the rendered HTML element.

Q26. [What is use of useEvent\(\) React hook?](#)

useEvent is a custom React hook that is used to listen for and handle events. It is not a built-in hook in React, so it is likely that it was created by the developer of the codebase you are working on.

The useEvent hook takes two arguments: the event type and a callback function to be executed when the event is triggered. It returns a function that can be used to remove the event listener.

Here is an example of how useEvent might be used:

```
import { useEvent } from '../customHooks';
function MyComponent() {
  const handleResize = () => {
    console.log('Window was resized!');
  };
  useEvent('resize', handleResize);
  return
```

My component

```
;
}
```

In this example, the useEvent hook is being used to listen for the resize event on the window object, and the handleResize function will be called whenever the window is resized.

Please Visit [OnlineInterviewquestions.com](https://www.onlineinterviewquestions.com) to download more pdfs