**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 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. |
| rendor() method is not used. | rendor() 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 in 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 it’s 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. get SnapshotBeforeUpdate() – 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 entirely 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 bootsrap 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 this 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.handleCheckChieldElement}

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  2  3  4  5  6 | import React from ‘react’’  class Class\_component extends React.Component {    render() {      return <h1>{`Hello, ${this.props.message}`}</h1>    }  } |

**50] What is the use of refs?**

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

### ****1. What is the difference between Virtual DOM and Real DOM?****

|  |  |
| --- | --- |
| **Virtual DOM** | **Real DOM** |
| Changes can be made easily | Changes can be expensive |
| Minimal memory wastage | High demand for memory and more wastage |
| JSX element is updated if the element exists | Creates a new DOM every time an element gets updated |
| Cannot update HTML directly | Able to directly manipulate HTML |
| Faster updates | Slow updates |

### ****2. What is React?****

**React**is a widely used JavaScript library that was launched in 2011. It was created by developers at Facebook, and it is primarily used for frontend development. React uses the component-based approach, which ensures to help you build components that possess high reusability.

React is well known for developing and designing complex mobile user interfaces and web applications.

### ****3. What is the meaning of Virtual DOM?****

A virtual DOM is a simple JavaScript object that is the exact copy of the corresponding real DOM. It can be considered as a node tree that consists of elements, their attributes, and other properties. Using the render function in React, it creates a node tree and updates it based on the changes that occur in the data model. These changes are usually triggered by users or the actions caused by the system.

Next up among these React interview questions, you need to take a look at some of the important features that React offers.

### ****4. What are some of the important features of React?****

React has multiple features that are used for unique purposes. The important ones are as mentioned below:

* React makes use of a single-direction data flow model.
* It deals with complete server-side data processing and handling.
* React uses Virtual DOM that has many advantages of its own.

### ****5. What is the meaning of JSX?****

JSX is the abbreviation of JavaScript XML. It is a file that is used in React to bring out the essence of JavaScript to React and use it for its advantages.

It even includes bringing out HTML and the easy syntax of JavaScript. This ensures that the resulting HTML file will have high readability, thereby relatively increasing the performance of the application.

Consider the following example of a JSX:

render(){

return(

<div>

<h1> Hello Intellipaat learners!</h1>

</div>

);

}

### ****6. Can browsers read a JSX file?****

No, browsers cannot read JSX files directly. It can only read the objects provided by JavaScript. Now, to make a browser read a JSX file, it has to be transformed to a JavaScript object using JSX transformers, and only then it can be fed into the browser for further use in the pipeline.

**Get 100% Hike!**

Master Most in Demand Skills Now !

Top of Form





Bottom of Form

### ****7. Why is React widely used today?****

**React** provides users with an ample number of advantages when building an application. Some of them are as follows:

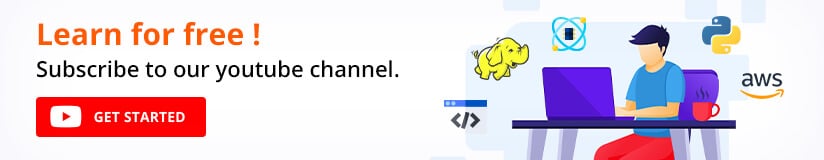
* With React, UI testing becomes very easy.
* React can integrate with Angular and other frameworks easily.
* The high readability index ensures easy understanding.
* React can be used for both client-side and server-side requirements.
* It boosts application performance and overall efficiency.

### ****8. Are there any disadvantages to using React?****

There are some limitations when using React as mentioned below:

* Writing code is complicated as it uses JSX and inline template formatting.
* Beginners might find it tough to cope with its syntaxes and methods.
* The library contains a huge repository of information, which might be overwhelming.
* React is a simple library and not a complete framework hence calls for dependencies.

### ****Check out our Full Stack Web Development Course video:****

[](https://www.youtube.com/user/intellipaaat?sub_confirmation=1)

### ****9. Differentiate between Angular and React.****

The difference between [**Angular and React**](https://intellipaat.com/blog/angular-vs-react/) are as follows:

|  |  |  |
| --- | --- | --- |
| **Comparison Factor** | **Angular** | **React** |
| Created by | Google | Facebook |
| DOM | Real DOM | Virtual DOM |
| Render Support | Client-side | Server-side |
| Architecture | Full MVC support | Only the view aspect of MVC |
| Data Binding | Unidirectional binding | Two-way binding |

***To learn more about React JS, Enroll now in our***[***React certification course***](https://intellipaat.com/react-js-certification-training-course/)***to gain in-depth knowledge.***

### ****10. What is the meaning of the component-based architecture of React?****

In **React**, components are foundations used to build user interfaces for applications. With the component-based system in place, all of the individual entities become completely reusable and independent of each other. This means that rendering the application is easy and not dependent on the other components of the UI.

### ****11. How does rendering work in React?****

Rendering is an important aspect of React as every single component must be rendered. This is done using the render() function. Once the function is called, it returns an element that represents a DOM component.

It is also possible to render more than one HTML element at a time by enclosing the HTML tags and passing them through the render function.

### ****12. What are states in React?****

States form to be one of the vital aspects of React. It is considered as a source of data or objects that control aspects such as component behavior and rendering. In React, states are used to easily create dynamic and interactive components.

***Check out our blog on***[***ReactJS Tutorial***](https://intellipaat.com/blog/reactjs-tutorial/)***to learn more about ReactJS.***

### ****13. What are props in React?****

Props are the shorthand name given to properties in React. Props are read-only components that are immutable in nature. In an application, props follow a hierarchy that is passed down from parents to child components. However, the reverse is not supported. This is done to ensure that there is only a single directional flow in data at all times.

### ****14. What is the use of an arrow function in React?****

An arrow function is used to write an expression in React. It allows users to manually bind components easily. The functionality of arrow functions can be very useful when you are working with higher-order functions particularly.

Consider the following example:

//The usual way

render() {

return(

<MyInput onChange={this.handleChange.bind(this) } />

);

}

//Making use of the arrow function

render() {

return(

<MyInput onChange={ (e) => this.handleOnChange(e) } />

);

}

**Career Transition**

[[](javascript:void(0);)](javascript:void(0);)

[[](javascript:void(0);)](javascript:void(0);)

[[](javascript:void(0);)](javascript:void(0);)

### ****15. What is a higher-order component in React?****

Higher-order components (HOCs) are a widely used technique in React for applying concepts that involve the component reusability logic. They are not a [React Native Firebase](https://intellipaat.com/blog/react-native-firebase/) part of the React API and allow users to easily reuse the code and bootstrap abstraction.

HOCs are also used to allow simple sharing of behaviors across all of the components in React, adding more advances to the efficiency and functioning of the application.

### ****16. What is the meaning of create-react-app in React?****

The create-react app in **React**is a simple command-line interface (CLI) that is used in the creation of React applications, which have no build configuration.

All tools are pre-configured when using the CLI, and this allows users to focus on the code more than on dependencies to develop the application.

The following syntax is used to start a simple project in React:

Create-react-app my-app

***Enroll in Intellipaat’s***[***Full Stack Developer Certification***](https://intellipaat.com/full-stack-web-developer-mean-stack-certification-training/)***to master front-end and back-end technologies.***

### ****17. What are some of the advantages of using create-react-app in React?****

Making use of create-react-app is advantageous in the following way:

* Support for JSX, ES6, and flow statements
* Already built and ready auto-prefixed CSS
* Fast interactive testing components
* Live development servers that help in debugging
* Scripts to handle JSS, CSS, and other files

Next up on these React Redux interview questions, you need to understand the meaning of Redux.

### ****18. What is the meaning of Redux?****

Redux is used to store the state of the application in a single entity. This simple entity is usually a JavaScript object. Changing states can be done by pushing out actions from the application and writing corresponding objects for them that are used to modify the states.

For example:

{

first\_name: ‘John’,

last\_name : ‘Kelly’,

age: 25

}

All of the data is retained by Redux (also called a store).

### ****19. What is the difference between props and states?****

|  |  |  |
| --- | --- | --- |
| **Condition** | **Props** | **States** |
| Changes in child components | Yes | No |
| Parent component changing values | Yes | No |
| Changes inside components | No | Yes |

Next up on this top React interview questions and answers blog, take a look at the questions categorized as intermediate!

## Intermediate React Interview Questions

### ****20. What are the three phases of a component life cycle in React?****

The following are the three phases of a component life cycle:

* **Initial rendering**: This is the phase that involves the beginning of the journey of the component to the DOM.
* **Update**: Here, the component can be updated and rendered again if required after it gets added to the DOM.
* **Unmounting**: The final phase involves the destruction of the component and its eventual removal from the DOM.

### ****21. What are events in React?****

Whenever there are actions performed in React, such as hovering the mouse or pressing a key on the keyboard, these actions trigger events. Events then perform set activities as a response to these triggers. Handling an event in React is very similar to that in the DOM architecture.

**Courses you may like**

[](https://intellipaat.com/advanced-certification-data-science-artificial-intelligence-iit-madras/)

[](https://intellipaat.com/post-graduate-certification-full-stack-web-development/)

[](https://intellipaat.com/pg-certification-business-analysis-project-management-mnit/)

### ****22. How are events created in React?****

Events can be created very easily in React as shown here:

class Display extends React.Component({

show(evt) {

// Code inside

},

render() {

// Render the div with an onClick prop (value is a function)

return (

<div onClick={this.show}>Click Here</div>

);

}

});

### ****23. How is routing in React different from conventional routing?****

Differences between the conventional routing and the routing in React can be shown using the following aspects:

* **Pages**: Each view is considered as a new file in conventional routing while it is considered as a single HTML entity in React.
* **Navigation**: In conventional routing, users have to move across web pages for viewing. In React, the views are not refreshed as objects are re-issued to create new views.

### ****24. Differentiate between Flux and Redux in React.****

|  |  |  |
| --- | --- | --- |
| **Comparison Factor** | **Flux** | **Redux** |
| Components | Components connected to Flux in React | Container components directly connect |
| Dispatcher | Has a dispatcher | No dispatcher |
| Number of Stores | Single store | Multiple stores |
| State | Mutable state | Immutable state |
| Storage | Contains state and logic | State and logic are separate |

### ****25. Can AJAX be used with React?****

Yes, any AJAX library, such as Axios and jQuery AJAX, can be used with React easily. One important thing is to maintain the states of the components, and here too, the props are passed from the parents to the child components.

Child components still cannot send back props to parents, and this factor greatly increases rendering efficiency when dynamic data is considered.

**If you are looking forward to becoming proficient in Angular.js, then make sure to check out Intellipaat’s latest offerings for the**[***Angular JS  Course***](https://intellipaat.com/angular-training/)**.**

### ****26. What is the meaning of synthetic events in React?****

Synthetic events in React are objects that act as cross-browser wrappers, allowing for the use of native events. This is done to ensure that a variety of browsers can run the API and that the event contains all properties.

### ****27. What are stateful components in React?****

Stateful components are entities that store the changes that happen and place them into the memory. Here, the state can be changed, alongside storing information such as past, current, and future changes.

[](https://intellipaat.com/post-graduate-certification-full-stack-web-development/)

### ****28. What are refs in React?****

‘Refs’ is short for references in React. Refs are used to store a reference to a single React element or a React component. This is later returned using the render function.

They are mainly used in the following scenarios:

* To initiate imperative animations
* To join third-party DOM libraries
* To manage focus and apply media playback

### ****29. What are controlled components in React?****

Controlled components in React refer to the components that have the ability to maintain their state. The data is completely controlled by the parent component, and the current value is fetched by making use of props. This is done to notify about any change that occurs when using callbacks.

### ****30. Why is a router required in React?****

A router is very much necessary in React as it is used to manage multiple routes whenever a user types in a URL. If the route is present in the router for that corresponding URL, then the user is taken to the particular route.

To do this, the router library needs to be added in React. It can be done using the following syntax:

<switch>

<route exact path=’/’ component={Home}/>

<route path=’/posts/:id’ component={Newpost}/>

<route path=’/posts’   component={Post}/>

</switch>

### ****31. What are the components of Redux in React?****

Redux consists of four main components as shown below:

* **Action**: An object that describes the call
* **Reducer**: The state change storage unit
* **Store**: the state and object tree storage
* **View**: Displays data provided by the store

### ****32. What are the advantages of using Redux?****

There are many advantages of Redux, and some of them are as given below:

|  |  |
| --- | --- |
| Organized Approach | Redux requires code to be organized, thereby making it consistent and easy to work with |
| Testing Ability | Redux functions are small and isolated, making the code more independent and testable |
| Tools | Developers can track actions and all of the tools in React using Redux easily |
| Community | Redux has a larger community, helping users with efficient and easy-to-use libraries |

### ****33. What are the disadvantages of using MVC in React?****

Among a plethora of advantages of using MVC in React, there are minor problems as stated below:

* A lot of memory wastage occurs.
* DOM manipulation costs a lot.
* The application becomes slow.
* Lots of dependencies are created.
* The complexity of models increases.

Next up among these ReactJS interview questions, you have to understand about pure components.

### ****34. What are pure components in React?****

Pure components are singular entities that are written in React. They are fast and simple to write and have the ability to replace a component that has only the render() function. This is done to ensure that the performance of the application is good and that the code is kept simple at the same time.

Next up on this top React interview questions blog, take a look at the questions categorized as advanced!

[](https://intellipaat.com/full-stack-web-developer-mean-stack-certification-training/)

## Advanced React Interview Questions

### ****35. What are higher-order components (HOCs) used for?****

HOCs are used for a variety of tasks such as:

* Manipulation of props
* State manipulation and abstraction
* Render highjacking
* Code reusing
* Bootstrap abstraction

### ****36. What are keys in React?****

Keys are used in React to check all items and to track changes actively. They are used to directly check if an item has been added or removed from a list.

Consider the following syntax:

function List ({ todos }) {

return (

<ul>

{todos.map(({ task, id} ) => <li key={id}>{task}</li>}

</ul>

)

}

### ****37. Differentiate between a controlled component and an uncontrolled component in React.****

A controlled component, as the name suggests, is a component over which React has complete control. It is the singular point of data for the forms.

An uncontrolled component is one where the form data gets handled by DOM and not the React component. This is usually done using refs in React.

### ****38. How can you tell React to build in the production mode?****

**React** can be coded to directly build into production by setting the process.env.NODE\_ENV variable to production.

**Note**: When React is in production, warnings and other development features are not shown.

### ****39. What is the difference between cloneElement and createElement in React?****

In React, cloneElement is primarily used to clone an element and pass it to new props directly. Whereas, createElement is the entity that JSX gets compiled into. This is also used to create elements in React.

Next up on this top React interview questions and answers blog, take a look at the use of the second argument.

### ****40. What is the use of the second argument that is passed to setState? Is it optional?****

When setState is finished, a callback function is invoked, and the components get re-rendered in React.

Yes, it is an optional argument. Since setState is asynchronous, it takes in another callback function. However, in programming practice, it is always good to use another life cycle method instead of this.

Next up on this top React interview questions and answers blog, you need to take a look at binding.

[](https://intellipaat.com/post-graduate-certification-full-stack-web-development/)

### ****41. Is there a way to avoid the requirement of binding when using React?****

Yes, there are two main ways you can use to avoid the need for binding. They are as follows:

* **Defining the Event Handler as an Inline Arrow function:**

class SubmitButton extends React.Component {

constructor(props) {

super(props);

this.state = {

isFormSubmitted: false

};

}

render() {

return (

<button onClick={() => {

this.setState({ isFormSubmitted: true });

}}>Submit</button>

)

}

}

* **Using a function component along with Hooks:**

const SubmitButton = () => {

const [isFormSubmitted, setIsFormSubmitted] = useState(false);

return (

<button onClick={() => {

setIsFormSubmitted(true);

}}>Submit</button>

)

};

Also, the Event Handler can be defined as an Arrow function, which is eventually assigned to a Class Field to obtain similar results.

### ****42. What is the StrictMode component used in React?****

The StrictMode component when used would benefit users immensely while creating new codebases to understand the components being used.

However, it can fit well in debugging as well because it will help solve the problem faster when it is wrapped with other components, which could be causing the problem.

Next up on these interview questions on React JS, you have to understand how to speed up rendering.

### ****43. What would you do if your React application is rendering slowly?****

The cause of slow rendering in React is mostly because of the number of re-render operations, which are sometimes unnecessary. There are two main tools provided by React to help users here:

* **memo():** This is used to prevent all of the unnecessary re-rendering carried out by the function components.
* **PureComponent:** This is used to ensure that the unnecessary re-rendering of class components is avoided.

### ****44. Can you conditionally add attributes to components in React?****

Yes, there is a way in which you can add attributes to a React component when certain conditions are met.

React has the ability to omit an attribute if the value passed to it is not true.

Consider the following example:

var condition = true;

var component = (

<div

value="foo"

{ ...( condition && { disabled: true } ) } />

);

### ****45. Why is props passed to the super() function in React?****

Props gets passed onto the super() function if a user wishes to access this.props in the constructor.

Consider the following example:

class MyComponent extends React.Component {

constructor(props) {

super(props)

console.log(this.props)

// -> { icon: 'home', … }

}

}

### ****46. What is the difference between using getInitialState and constructors in React?****

When using ES6, users must initialize the state in the constructor and the getInitialState method is defined. This is done using React.createClass as shown in the below example:

class MyComponent extends React.Component {

constructor(props) {

super(props);

this.state = { /\* initial state \*/ };

}

}

This above piece of code is equivalent to the following:

var MyComponent = React.createClass({

getInitialState() {

return { /\* initial state \*/ };

},

});

Next up among these interview questions on React JS, you have to know what predefined props are.

[](https://intellipaat.com/full-stack-web-developer-mean-stack-certification-training/)

### ****47. What are the predefined prop types present in React?****

There are five main predefined prop types in React. They are as follows:

1. PropTypes.bool
2. PropTypes.func
3. PropTypes.node
4. PropTypes.number
5. PropTypes.string

The propTypes can be defined for the user component as shown below:

import PropTypes from 'prop-types';

class User extends React.Component {

render() {

return (

<h1>Welcome, {this.props.name}</h1>

<h2>Age, {this.props.age}

);

}

}

User.propTypes = {

name: PropTypes.string.isRequired,

age: PropTypes.number.isRequired

};

### ****48. What is React Fiber?****

React Fiber is a new engine in React. It is the reimplementation core algorithm in React 16.

The main goal of React Fiber is to ensure that there are incremental rendering facilities for the virtual DOM. This increases efficiency when rendering animations, gestures, etc. and also helps in assigning priority to updates based on the requirement, thereby increasing overall efficiency.

### ****49. What are Hooks in React?****

Hooks are used to make use of the state and other features without having to explicitly write a class. Hooks were added to the React version, v16.8. The stateful logic can be extracted from a component easily, alongside testing and reusing it. All of this is done without making any changes to the component hierarchy.

### ****50. Do you have any certification to boost your candidature for this React.js role?****

With this question, the interviewer is trying to assess if you have any technical experience through learning and implementation. It is always advantageous to have a certification in the technology that you’re applying for.

This creates an impression that you have put your time and effort into learning and implementing the technology. Alongside adding a lot of value to your resume and your knowledge on the topic, it can be used to obtain a well-coveted job in the market!

If you are looking forward to becoming proficient in Web Development, then make sure to check out Intellipaat’s latest offerings for the [Web Development Online Courses](https://intellipaat.com/course-cat/website-development/). With these programs, you can become an expert in Web Development and earn a course certificate as well.

## React JS Interview Questions and Answers - For Beginners

### 1. What is React?

React is a front-end JavaScript library that mainly follows the component-based approach for building user interface (UI) components for a single-page application. It is also used for handling the view layer in both mobile and web apps. Moreover, react plays a crucial role in developing interactive mobile and web UIs. It was created and developed by Jordan Walke; it was deployed first on the Facebook newsfeed in 2011.

### 2. Why React is used?

The following reasons make one use React for building User Interfaces (UI), and they are:

* Easy to learn nature
* Simplicity
* High scalability
* Increase performance

|  |
| --- |
| **Related Article:** [**React Tutorial**](https://mindmajix.com/react-js-tutorial) |

### 3. How does React work?

Below is the sequence of steps which gives an idea about how does react work

* Firstly the react runs the diffing algorithm to identify the changes that are made in the virtual DOM.
* The next step is reconciliation, this is used to update the DOM as per the new features.
* Now, the virtual DOM is lightweight in nature and is detached from the specific implementation of the browser.
* Followingly the ReactElements which are present in virtual DOM are used to build basic nodes.
* Finally, if the ReactComponent changes the state; the diffing algorithm runs faster and identifies the changes. After identification, it automatically updates the DOM with the change difference.

### 4. What are the features of ReactJS?

The features of React JS are as follows:

#### 1. React improves SEO performance

React boost the performance of the SEO to higher levels as a search engine faces the problem while reading JavaScript of high-loaded applications.

#### 2. React acts as a standard for mobile app development

It provides a transition process as an ideal solution for both mobile and web applications for building rich user interfaces.

#### 3. React makes the process of writing components easier

Using React along with JSX will make you write components and code efficiently and clearly.

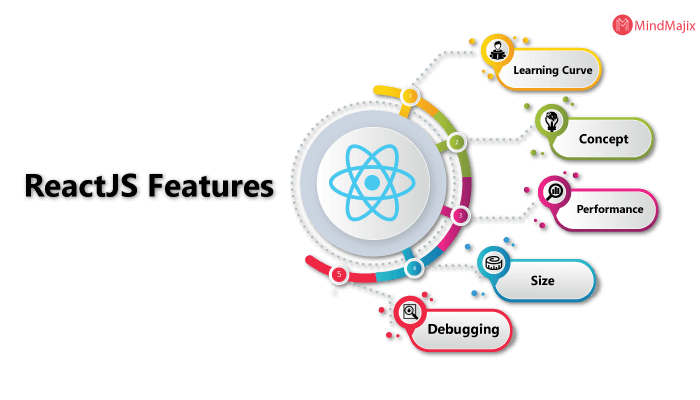
#### 4. React increases efficiency

React boost the efficiency of components by reusing them. This is the reason why it is considered an ideal feature of React. It is considered the most reusable system component.

#### 5. React ensures stable code

It ensures the stability of the code of an application by making use of a downward data flow.

|  |
| --- |
| **Related Article:**[**ReactJS Features**](https://mindmajix.com/introduction-to-react-js) |



### 5. What are the Advantages of React JS?

The advantages of React are as follows

* Usage of JSX makes it easier to read and write code
* Improves the performance of applications with the use of virtual DOM
* Provides an easier way to integrate with frameworks
* It can be shared and rendered both on server and client-side
* Writing integration and unit tests can be made smother by using tools

### 6. How is React different from AngularJS?

The following table shows the major difference between AngularJS and React

|  |  |  |
| --- | --- | --- |
| **Factor** | **React JS** | **AngularJS** |
| **Usage of DOM** | Uses virtual DOM | Uses real DOM |
| **Language** | Uses JavaScript with the extended XML syntax | Uses TypeScript which is the superset of JavaScript |
| **App Structure** | It is represented only using the view of MVC | Made of Complete MVC |
| **Data Binding** | One-way binding | Two-way binding |

|  |
| --- |
| **Related Article:**[**Angular vs React**](https://mindmajix.com/react-js-vs-angular-js) |

### 7. How React JS framework is different as compared to others?

Basically, ReactJS is a limited library that builds UI parts, it is essentially not quite the same as a considerable measure of other JavaScript structures. One common example is AngularJS approaches building an app simply by expanding HTML markup and infusing different developments such as controllers at runtime. Therefore, AngularJS is exceptionally obstinate about the more noteworthy engineering of your application.

### 8. Does React JS use HTML?

No, It uses JSX which is similar to HTML.

[](https://bit.ly/3if9dmk)

### 9. What is JSX?

It is basically a novel dialect of the popular JavaScript that simply integrates the HTML templates into the code of JavaScript. The browser is not capable to read the code simply and thus there is a need for this integration. Generally, WebPack or Babel tools are considered for this task. It has become a very popular approach in the present scenario among developers.

### 10. What are the life Cycles of ReactJS?

1. Initialization
2. State/Property Updates
3. Destruction

### 11. When was ReactJS released?

March 2013

### 12. How is React JS different from Angular JS?

The first difference between both of them is their code dependency. ReactJS depends less on the code whereas AngularJS needs a lot of coding to be done. The packaging on React is quite strong as compared to AngularJS. Another difference is React is equipped with Virtual Dom while Angular has a Regular DOM. ReactJS is all about the components whereas AngularJS focus mainly on the Models, View as well as on Controllers. AngularJS was developed by Google while ReactJS is the outcome of Facebook. These are some of the common differences between the two.

|  |
| --- |
| **Related Article:**[**Angular Tutorial for Beginners**](https://mindmajix.com/angularjs-tutorial) |

### 13. What is Redux?

It is one of the most in-demand libraries for front-end development in today’s growing world. It is defined as the predictable state container mainly designed for JavaScript apps and also it is used for managing the entire state of an application. Redux is very small in size and has no dependencies. It builds applications that are easy to deploy in different environments and easy to test. Redux is very small in size and has no dependencies.

### 14. What is the Use of Redux thunk?

Redux thunk acts as middleware which allows an individual to write action creators that return functions instead of actions. This is also used as a delay function in order to delay the dispatch of action if a certain condition is met. The two store methods getState() and dispatch() are provided as parameters to the inner function.

In order to activate Redux thunk, we must first use applyMiddleware() method as shown below:

import{ createStore, applyMiddleware } from 'redux';

import thunk from 'redux-thunk';

import rootReducer from './reducers/index';

//Note: this API requires redux@>=3.1.0

const store= createStore(

rootReducer,

applyMiddleware(thunk)

);

### 15. What do you know about Flux?

Basically, Flux is a basic illustration that is helpful in maintaining the unidirectional data stream.  It is meant to control construed data's unique fragments to make them interface with that data without creating issues. Flux configuration is insipid; it's not specific to React applications, nor is it required to collect a React application.  Flux is basically a straightforward idea, however, you have to exhibit a profound comprehension of its usage.

### 16. What is the current stable version of ReactJS?

\*\*\*\*\*\*Version: 16.12.0

\*\*\*\*\*\*Release on: Nov 14, 2019

### 17. What is the Repository URL of ReactJS?

<https://github.com/facebook/react>

### 18. What do you know about the component lifecycle in ReactJS?

The component lifecycle is an essential part of this platform. Basically, they have lifecycle events that fall in the three prime categories which are property updates, Initialization, and third is Destruction. They are generally considered as a method of simply managing the state and properties of every reach component.

### 19. Do you think React JS has any limitations? If so, tell a few?

Yes, there are a few drawbacks which are associated with this platform. The leading drawback of ReactJS is the size of its library. It is very complex and creates a lot of confusion among the developers. Also, there are lots of developers all over the world who really don’t like the JSX and inline templating. In addition to this, there is another major limitation of ReactJS and i.e. only cover one layer of the app and i.e.View. Thus to manage the development, developers have to depend on several other technologies which consume time.

**Related Article:**[**React Hooks Interview Questions**](https://mindmajix.com/react-hooks-interview-questions)

### 20. How do the parent and child components exchange information?

This task is generally performed with the help of functions. Actually, there are several functions that are provided to both parent and child components. They simply make use of them through props. Their communication should be accurate and reliable. The need for the same can be there anytime and therefore functions are considered for this task. They always make sure that information can be exchanged easily and in an efficient manner among the parent and child components.

### 21. What is a State in React and How is it used?

In React, State is an object that represents how the component renders and behaves. States are the sources of data and allow you to create dynamic and interactive components. They are accessed using this.state(). For changing a value in the state object, call it using this.setState() method.

### 22. What are the differences between the Class component and the Functional component?

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Class Component** | **Functional Component** |
| Syntax | This component requests you to extend from React. Component to create render function that in turn returns a react element | It is just a plain JavaScript function that accepts props as their arguments and returns the react element. |
| Life cycle hooks | Lifecycle hooks are created from the React Component. This class component makes lifecycle hooks available in it. | We cannot use lifecycle hooks in a functional component. |
| Readability | They are very difficult to test and read | They are much easier to test and read |

Class Component Coding

class App extends Component {

render () {

return (

<Text>Hello World!</Text>

)

}

}

Functional  Component Coding

const PageOne = () => {

return (

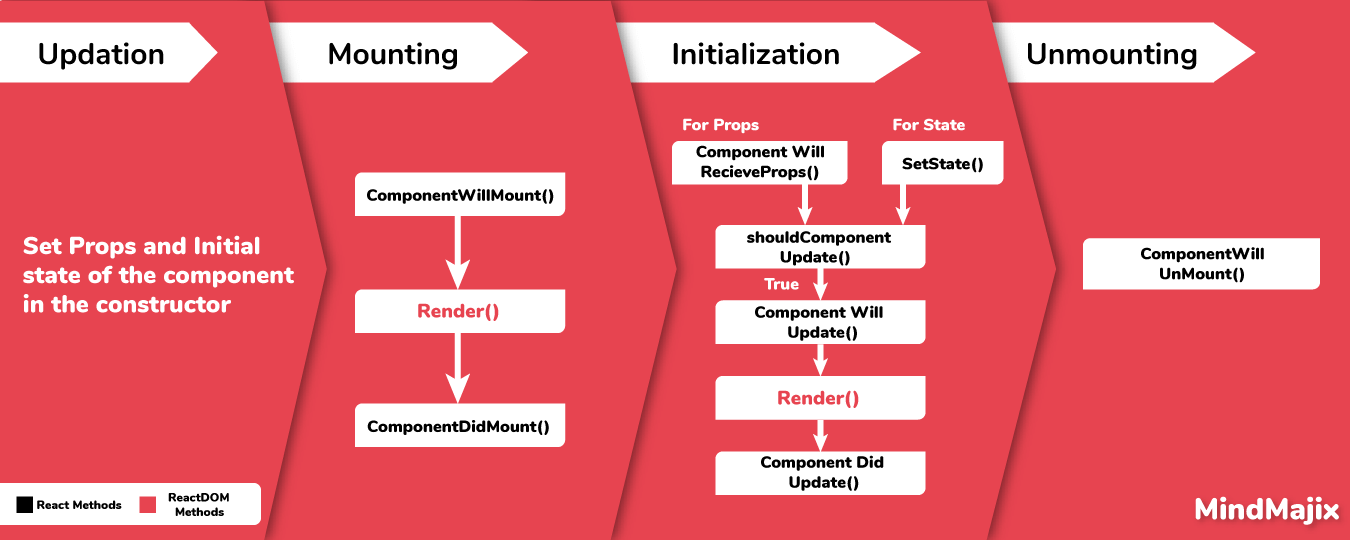
<h1>Page One</h1>

);

}

### 23. What are props in React?

Props stand for properties in React and used for passing the information from one component to another. But the data with the Props are passed in a unidirectional flow, i.e., one way from parent to child. Further, they are read-only data, which means child components cannot change data coming from the parent.



### 24. What are the significant differences between state and props?

The difference between state and props are as follows:

|  |  |
| --- | --- |
| **State** | **Props** |
| The state is completely managed within a component for internal communication. | Props are directly passed to its parents with child component. |
| State can be modified using setState() method. | A particular component should never modify its own props. |
| State changes can be asynchronous | Props are read-only |

### 25. What is the higher-order component?

In ReactJS high order component can be defined as the function that is mainly used to collect the component and returns a new component. These components are the patterns that are extracted from React’s compositional nature. One important aspect of this component is that it is used as a reusable component logic in React. It provides us with the best way to share behavior between different React components.

### 26: How to embed two components in One component?

import React from 'react';

class App extends React.Component{

render(){

return(

<div>

<Header/>

<Content/>

</div>

);

}

}

class Header extends React.Component{

render(){

return(

<div>

<h1> Header</h1>

</div>

)

}

}

class Content extends React.Component{

render(){

return(

<h2>Content</h2>

<p>The Content Text!!!</p>

</div>

)

}

}

export default App;

### 27. What are Synthetic events in React?

React implements Synthetic events to improve the consistency and performance of applications and interfaces. The synthetic event is a cross-browser wrapper around the browser’s native event. It combines the behavior of multiple browsers into a single API to make sure events have the same properties across different browsers and platforms.

## React JS Interview Questions for Experienced

### 28. Give one basic difference between pros and state?

Pros are immutable while the state is mutable. Both of them can update themselves easily.

### 29. How do you tell React to build in Production mode and what will that do?

Ordinarily, you'd utilize Webpack's DefinePlugin strategy to set NODE\_ENV to production. This will strip out things like prototype approval and additional notices. Over that, it's likewise a smart thought to minify your code in light of the fact that React utilizes Uglify's dead-code end to strip out advancement just code and remarks, which will radically diminish the measure of your package.

### 30. What do you understand with the term polling?

The server needs to be monitored for updates with respect to time. The primary aim in most cases is to check whether novel comments are there or not. This process is basically considered pooling. It checks for updates approximately every 5 seconds. It is possible to change this time period easily. Pooling help keep an eye on the users and always make sure that no negative information is present on the servers. Actually, it can create issues related to several things, and thus pooling is considered.

Learn [*React JS Online Training in Hyderabad*](https://mindmajix.com/react-js-training-hyderabad)

### 31. When would you use a Class Component over a Functional Component?

If your component has a state or a lifecycle method(s), use a Class component. or else, use a Functional component.

### 32. What do you mean by virtual DOM?

For all the available DOM objects in ReactJS, there is a parallel [virtual DOM](https://reactjs.org/docs/faq-internals.html) object. It is nothing but can be considered as the lighter version of the true copy and is powerful in eliminating the complex code. It is also used as a Blue Print for performing several basic experiments. Many developers also use it while practicing this technology.

### 33. Compare MVC with Flux?

MVC approaches are presently considered as outdated. Although they are capable to handle data concerns, controllers as well as UI, many developers found that it doesn’t properly work when application size increases. However, they are capable to handle some of the key issues such as eliminating the lack of data integrity as well as managing the data flow which is not properly defined. On the other side, Flux works perfectly with all the sizes irrespective of their size.

### 34. What’s the difference between an Element and a Component in React?

Basically, a React component describes what you need to see on the screen. Not all that basically, a React element is a protest portrayal of some UI.

A React component is a function or a class that alternatively acknowledges input and returns a React component (ordinarily by means of JSX which gets transpiled to a createElement invocation).

### 35. Tell us three reasons behind the success of ReactJS?

ReactJS is a technology that can be trusted for complex tasks. While performing any task through it, developers need not worry about the bugs. It always ensures error-free outcomes and the best part is it offers scalable apps. It is a very fast technology and can simply be trusted for quality outcomes.

### 36. In which lifecycle event do you make AJAX requests and why?

AJAX solicitations ought to go in the componentDidMount lifecycle event.   
There are a couple of reasons behind this,

Fiber, the following usage of React's reconciliation algorithm, will be able to begin and quit rendering as required for execution benefits. One of the exchange offs of this is componentWillMount, the other lifecycle event where it may bode well to influence an AJAX to ask for will be "non-deterministic". This means React may begin calling componentWillMount at different circumstances at whenever point it senses that it needs to. This would clearly be a bad formula for AJAX requests.

You can't ensure the AJAX request won't resolve before the component mounts. In the event that it did, that would imply that you'd be attempting to setState on an unmounted component, which won't work, as well as React will holler at you for. Doing AJAX in componentDidMount will ensure that there's a component to update.

### 37. What is the difference between createElement and cloneElement?

createElement is the thing that JSX gets transpiled to and is the thing that React uses to make React Elements (protest representations of some UI). cloneElement is utilized as a part of a request to clone a component and pass it new props. They nailed the naming on these two.

### 38. What is meant by event handling?

To capture the user’s information and other similar data, the event handling system is considered. It is generally done through DOM elements that are present in the code. This task is simple to accomplish. Two-way communication is considered in this approach.

### 39. What is the second argument that can optionally be passed to setState and what is its purpose?

A callback work will be conjured when setState has completed and the part is re-rendered.   
Something that is not talked about a great deal is that setState is asynchronous, which is the reason it takes in a moment callback function. Ordinarily, it's best to utilize another lifecycle strategy instead of depending on this callback function, however, it's great to know it exists.

Class Training extends Course

{

this.state = {

sampleItem: 'learn',

}

handleChange = (event) => {

console.log(this.state.sampleItem)

this.setState({

sampleItem: event.target.value //event.target.value = Welcome

}, () => console.log(this.state.sampleItem))

};

**Output:**

Learn

Welcome

### 40. How many outermost elements can be there in a JSX expression?

It must have one JSX element present so that the task can be accomplished easily. Having more than one expression is not an issue but probably it will slow down the process. There are also chances of confusion with more than one expression if you are new to this technology.

### 41. What are controlled and uncontrolled components?

There are components in the ReactJS that maintain their own internal state. They are basically considered uncontrolled components. On the other side, the components which don’t maintain any internal state are considered as controlled components in ReactJS. Controlled components can easily be controlled by several methods. Most of the React components are controlled components.

### 42. Mention the key benefits of Flux?

Applications that are built on Flux have components that can simply be tested. By simply updating the store, developers are able to manage and test any react component. It cut down the overall risk of data affection. All the applications are highly scalable and suffer no compatibility issues.

### 43. What's wrong with the following code?

this.setState{(prevState, props)=>

{

return {

streak: prevState.streak+props.count

}

})

Nothing isn't right with it. It's once in a while utilized and not outstanding, but rather you can likewise pass a function to setState that gets the past state and props and returns another state, similarly as we're doing above. Furthermore, is nothing amiss with it, as well as effectively recommended in case you're setting state in light of the previous state.

### 44. Why browsers cannot read JSX?

Actually, JSX is not considered proper JavaScript. Browsers cannot read them simply. There is always a need to compile the files that contain JavaScript Code. This is usually done with the help of the JSX compiler which performs its task prior to the file entering the browser. Also, compiling is not possible in every case. It depends on a lot of factors such as the source or nature of the file or data.

### 45. What are pure Functional Components?

Traditional React Components as we have seen so far are making a class with class Example extends React.Component or React.createClass(). These make stateful components on the off chance that we at any point set the state (i.e. this.setState(), getInitialState(), or this.state = {} inside a constructor()).

In the event that we have no expectation for a Component to require state, or to require lifecycle methods, we can really compose Components with a pure function, consequently the expression "pure function Component":

function Date(props)

{

let {msg="The date is:"} = props

let now = new Date()

return <div>

<span> {msg}</span>

<time> {now.toLocaleDateString()}</time>

</div>

}

This function that returns a React Element can be used wherever we see fit:

DOM.render(<div> <Date msg="Today is"/></div>)

You might notice that also takes a prop – we can still pass information into the Component.

### 46. What is the difference between Real DOM and virtual DOM?

* DOM stands for Document Object Model. It allows scripts and programs to dynamically access and updates the content, structure, and style of a document. DOM is an abstraction of a structured code called HTML, also described as HTML DOM.
* Virtual DOM
* is a lightweight Javascript object, which is the copy of the representation of a DOM object. It is an abstraction of HTML DOM. Virtual is quite faster compared to DOM, performs its tasks reliably.

|  |  |
| --- | --- |
| **Virtual DOM** | **Real DOM** |
| Updates faster | Updates slower |
| No memory wastage | Excess memory wastage |
| Can't update HTML directly | Update HTML directly |
| DOM manipulation is easy | DOM manipulation costly |

### 47. What happens during the lifecycle of a React component?

A standout amongst the most valuable parts of React is its segment lifecycle — so seeing precisely how segments components after some time is instrumental in building a viable application.

### 48. What exactly you can do if the expression contains more than one line?

In such a situation, enclosing the multi-line JSX expression is an option. If you are a first-time user, it may seem awkward but later you can understand everything very easily. Many times it becomes necessary to avoid multi-lines to perform the task reliably and for getting the results as expected.

### 49. Is it possible to use the word “Class” in JSX. Why or why not?

No, it is not possible in the JSX. This is because the word “Class” is a reticent (occupied) word in JavaScript. However, you can use you are free to use the word “ClassName”. If you use the word “Class” the JSX will be translated to JavaScript immediately.

a) High-Level Component Lifecycle:

At the highest level, React components have lifecycle events that fall into 3 general classifications:

1. Initialization
2. State/Property Updates
3. Destruction

Each React component defines these events as a system for dealing with its properties, state, and rendered output. Some of these events just happen once, others happen more as often as possible; understanding these 3 general classes should help you clearly visualize when certain logic required to be applied.

For instance, a component may need to add an event audience to the DOM when it initially mounts. In any case, it ought to likely expel those event listeners when the component unmounts from the DOM with the goal that not relevant handling doesn't occur.

class MyComponent extends React.Component{

//when the component is added to the DOM...

componentDidMount(){

window.addEventListener('resize',this.onResizeHandler);

}

//when the component is removed from the DOM...

componentWillmount(){

window.addEventListener('resize',this.onResizeHandler);

}

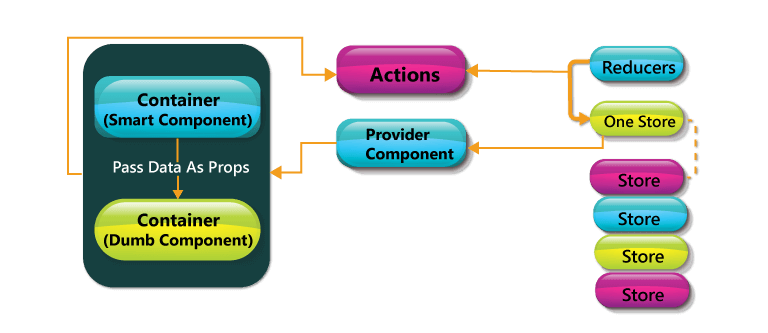
onResizeHandler(){

console.log('The window has been resized!');

}

}

b) Low-Level Component Lifecycle:



Inside these 3 general buckets exist various particular lifecycle hooks — basically unique techniques - that can be used by any React component to all the more precisely manage updates. Seeing how and when these hooks fire is vital to building stable components and will empower you to control the rendering procedure (enhancing execution).

Observe the diagram above. The events under "Initialization" just happen when a component is first initialized or added to the DOM. Thus, the events under "Devastation" just happen once (when the component is expelled from the DOM). However, the events under "Update" happen each time the properties or state of the component change.

For instance, components will naturally re-render themselves whenever their properties or states change. However, at times a component should not update - so keeping the component from re-rendering may enhance the execution of our application.

class MyComponent extends React.Component{

//only re-render if the ID has changed!

shouldComponentUpdate(nextProps, nextState;

}

}

### 50. What do you know about React Router?

Rendering the components is an important task in ReactJS. React router is used to decide which components are to be rendered and which ones should not. It also performs dictation during several activities.

### 51. Compare Flux vs MVC

Conventional MVC designs have functioned admirably to separate the worries of data (Model), UI (View) and logic (Controller) — however many web engineers have found impediments with that approach as applications develop in measure. In particular, MVC architectures as often as possible experience 2 primary issues:

**Ineffectively defined data flow:** The cascading updates which happen crosswise over perspectives frequently prompt a tangled web of events that are hard to debug.

**Lack of data integrity:** Model data can be changed from any place, yielding erratic results over the UI.

With the Flux pattern complex, UIs never again experience the ill effects of cascading updates; any given React component will have the capacity to recreate its state in light of the information given by the store. The flux pattern likewise upholds data integrity by limiting direct access to the shared data.

While a technical interview, it is awesome to talk about the contrasts between the Flux and MVC configuration designs inside the setting of a particular illustration:

For instance, imagine we have a "master/detail" UI in which the client can choose a record from a rundown (master view) and alter it utilizing an auto-populated form (detail view).

With an MVC architecture, the data contained inside the Model is shared between both the master and detail views. Each of these perspectives may have its own particular Controller assigning updates between the Model and the View. Anytime the information contained inside the Model may be updated — and it's hard to know where precisely that change happened. Did it occur in one of the Views sharing that Model, or in one of the Controllers? Since the Model's information can be transformed by any performing artist in the application, the danger of information contamination in complex UIs is more prominent than we'd like.

With a Flux architecture, the Store data is correspondingly shared between different Views. However this data can't be straightforwardly changed — the greater part of the solicitations to update the data must go through the Action > Dispatcher chain first, eliminating the risk of arbitrary data pollution. At the point when refreshes are made to the data, it's presently significantly less demanding to find the code requesting for those progressions.

Visit here to learn [*React JS Online Training in Bangalore*](https://mindmajix.com/react-js-training-bangalore)

### 52. What are the stateless components?

On the off chance that Reacts components are basically state machines that produce UI markup, at that point what are stateless segments?

Stateless components (a kind of "reusable" components) are simply pure functions that render DOM construct exclusively with respect to the properties given to them.

As you can see, this component has no requirement for any internal state — not to mention a constructor or lifecycle handlers. The yield of the component is absolutely a function of the properties gave to it.

### 53. What is one of the core types in React?

ReactNode

### 54. What is redux?

A method of handling the state (or data) of an application.

### 55. Is it possible to display props on a parent component?

Yes, it is possible. The best way to perform this task is by using the spread operator. It can also be done by listing the properties but this is a complex process.

### 56. In ReactJS, why there is a need to capitalize on the components?

It is necessary because components are not the DOM element but they are constructors. If they are not capitalized, they can cause various issues and can confuse developers with several elements. At the same time, the problem of integration of some elements and commands can be there.

### 57. What are Synthetic events in React?

React implements Synthetic events to improve the consistency and performance of applications and interfaces. The synthetic event is a cross-browser wrapper around the browser’s native event. It combines the behavior of multiple browsers into a single API to make sure events have the same properties across different browsers and platforms.

### 58. Explain DOM diffing?

When the components are rendered twice, Virtual Dom begins checking the modifications elements have got. They represent the changed element on the page simply. There are several other elements that don’t go through changes. To cut down the changes to the DOM as an outcome of user activities, DOM doffing is considered. It is generally done to boost the performance of the browser. This is the reason for its ability to perform all the tasks quickly.

### 59. Is it possible to nest JSX elements into other JSX elements?

It is possible. The process is quite similar to that of nesting the HTML elements. However, there are certain things that are different in this. You must be familiar with the source and destination elements to perform this task simply.