React Interview Question

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 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.

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.

9. Differentiate between Angular and React.

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

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.

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) } />
);
}
```

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 native 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

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.

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.

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.

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!

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.

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.

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.