**1) What is Single Page Application ?**

Single page applications work within a single browser session and allow users to jump between parts of the system without requiring the entire page to reload.

**2) Difference between npm & npx ?**

• Node package Manager is a tool: used to install, maintain, uninstall javascript modules (packages)

• NPX: (Node package Execution) If we want directly run a package for once.

• NPM is for management -vs- NPX is one time execution.

**3) What is babel & Web-pack?**

• Bable is a transpiler & also a compiler.

• It will take ES6 input & converts it to ES5 for our browsers to understand.

• It also converts JSX (JavaScript Extended) language to JS.

• Web-pack is basically bundling tool used by react:

• It will bundle all the CSS & html files & will add to Index.html file. (Using <link> tags)

**4) Explain Class based components vs functional components?**

|  |  |
| --- | --- |
| **Functional Components** | **Class Components** |
| Functional components are written  as a JavaScript function. | Class components are written  as a JavaScript class. |
| Functional components do not have a state or lifecycle methods. | Class components have a state  and can implement lifecycle  methods like  componentDidMount  and componentDidUpdate. |
| Faster performance as they do  not have state and lifecycle,  react needs to do less work to  render these components. | Slower performance as they  have state and lifecycle, react  needs to do comparatively  more work to render  these components. |
| Functional components code tend  to be shorter and more concise. | Class components require  the boilerplate code, such as a constructor method and the use  of “this” to access props and state. |

**5) What is Hooks? What are different types of Hooks?**

Hooks are functions that let you “hook into” React state and lifecycle features from function components. Hooks don't work inside classes — they let you use React without classes.

Types of Hooks are:

• ‘useState’: It is the most commonly used React Hook. It allows functional components to have state variables. It takes an initial state value as the argument and returns an array with two elements — the current state value and a function to update that state.

• ‘useEffect’: It enables performing side effects, such as data fetching, subscriptions, or DOM manipulations after the component has been rendered.

• ‘useRef’: Provides a way to create mutable references to values or DOM elements that persist across renders. Often used for accessing or manipulating DOM elements.

**6) What is a state in React?**

State is an Object, and we use the state to store data, that will be used by our component.A component's state can change over time; whenever it changes, the component re-renders.

**7) What does setState() method do?**

setState() method will modify value of the state.

Then it will Schedule a call to our render method.

**8) What is an Event Handler?**

Event Handler are functions which will be called (invoked) by the browsers when an event occurs.

**9) What are Synthetic events?**

Synthetic events are representative of real events a.k.a wrapper around realevents. Because we have to deal with different implementation of eventfunctions in different browsers, a synthetic event acts as a specification for asingle event access all browsers.

**10) What is a target?**

A target inside a synthetic event object is the element on which the event has occurred. As we cannot use DOM functions like document.getElementById() etc inside react, we use event object to fetch the element present inside ‘target variable’.

**11) What is Virtual DOM?**

The virtual DOM (VDOM) is a programming concept where an ideal, or “virtual”,representation of a UI is kept in memory and synced with the “real” DOM by a library such asReact DOM. This process is called reconciliation.

**12) What is the difference between state-vs-props?**

● Props are used to pass data from one component to another.

● The state is a local data storage that is local to the component only and cannot be passed to other components.

**13) Why is event.preventDefault() used?**

● Ideally when a form is submitted it will be taken to the next page (consecutive page).

event.preventDefault(); method when invoked avoids it from happening.

● We do this because, we do not want to lose our current state variable values.

**14) What is ReactJS?**

React is a JavaScript library that makes building user interfaces easy. It was developed by Facebook.

**15) Does React use HTML?**

No, It uses JSX, which is similar to HTML.

**16) Define the term Redux in React.**

Redux is a library used for front end development. It is a state container for JavaScript applications which should be used for the applications state management. You can test and run an application developed with Redux in different environments.

**17) What is the ‘Store’ feature in Redux?**

Redux has a feature called ‘Store’ which allows you to save the application’s entire State at one place. Therefore all it’s component’s State are stored in the Store so that you will get regular updates directly from the Store.

**18) Name the important features of React.**

Here, are important features of React.

• Allows you to use 3rd party libraries

• Time-Saving

• Faster Development

• Simplicity and Composable

• Fully supported by Facebook.

• Code Stability with One-directional data binding

**19) Explain React Router.**

React Router is a routing library which allows you to add new screen flows to your application, and it also keeps URL in sync with what’s being shown on the page.

**20) What is meant by callback function? What is its purpose?**

A callback function should be called when setState has finished, and the component is re-rendered. As the setState is asynchronous, which is why it takes in a second callback function.

**21) What are Props in React JS?**

Props mean properties, which is a way of passing data from parent to child. We can say that props are just a communication channel between components. It is always moving from parent to child component.

**22) What is the use of a super keyword in React?**

The super keyword helps you to access and call functions on an object’s parent.

**23) Name two types of React component.**

Two types of react Components are:

• Function component

• Class component

**24) How can you update state in React JS?**

A state can be updated on the component directly or indirectly.

**25) Explain the use of the arrow function in React?**

The arrow function helps you to predict the behavior of bugs when passed as a callback. Therefore, it prevents bug caused by this all together.

**26) What are the lifecycle steps of React?**

Important lifecycle steps of React js are:

• Initialization

• State/Property updates

• Destruction are the lifecycle of React

**27) What is ‘create-react-app’?**

‘create-react-app’ is a command-line tool which allows you to create one basic react application.

**28) Explain the use of ‘key’ in react list ?**

Keys allow you to provide each list element with a stable identity. The keys should be unique.

**29) What are children prop?**

Children props are used to pass component to other components as properties. You can access it by using

{props.children}

**30) What is the use of empty tags <> </>?**

Empty tags are used in React for declaring fragments.

**31) Explain strict mode.**

Strict Mode allows you to run checks and warnings for react components. It runs only on development build. It helps you to highlight the issues without rendering any visible UI.

**32) Can we update props in react?**

We can’t update props in react js because props are read-only. Moreover, we can not modify props received from parent to child.

**33) Name any five predefined prototypes used in React.**

Most important protoype used in React js are:

• number

• string

• array

• object

• element

**34) What is JSX?**

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

**35) Explain how lists work in React.**

We create lists in React as we do in regular JavaScript. Lists display data in an ordered format. The traversal of lists is done using the map() function.

**36) Explain the lifecycle methods of components.**

• componentDidMount(): Is executed when the component gets rendered and placed on the DOM.

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

• componentDidUpdate(): Is invoked immediately after rendering takes place.

• componentWillUnmount(): Is invoked immediately before a component is destroyed and unmounted permanently.

**37) What are the features of React?**

Major features of React are listed below:

• It uses the virtual DOM instead of the real DOM.

• It uses server-side rendering.

• It follows uni-directional data flow or data binding.

**38) What are the biggest limitations of React?**

Following is the list of the biggest limitations of React:

• React is just a library. It is not a complete framework.

• It has a huge library which takes time to understand.

• It may be difficult for the new programmers to understand and code.

• React uses inline templating and JSX, which may be difficult and act as a barrier. It also makes the coding complex.

**40) List down the components of Redux.**

Redux is composed of the following components:

• Action – It’s an object that describes what happened.

• Reducer – It is a place to determine how the state will change.

• Store – State/ Object tree of the entire application is saved in the Store.

• View – Simply displays the data provided by the Store.

**41) What is Lifting State Up in React?**

When several components need to share the same changing data then it is recommended to lift the shared state up to their closest common ancestor. That means if two child components share the same data from its parent, then move the state to parent instead of maintaining local state in both of the child components.

**42) Why React uses className over class attribute?**

The attribute class is a keyword in JavaScript, and JSX is an extension of JavaScript. That's the principal reason why React uses className instead of class. Pass a string as the className prop.

render()

{

return <spanclass Name = {'menunavigationmenu'} > {'Menu'} </span>

}

**43) Why fragments are better than container divs?**

Below are the list of reasons to prefer fragments over container DOM elements,

• Fragments are a bit faster and use less memory by not creating an extra DOM node. This only has a real benefit on very large and deep trees.

• Some CSS mechanisms like Flexbox and CSS Grid have a special parent-child relationships, and adding divs in the middle makes it hard to keep the desired layout.

**44) How to use styles in React?**

The style attribute accepts a JavaScript object with camelCased properties rather than a CSS string. This is consistent with the DOM style JavaScript property, is more efficient, and prevents XSS security holes.

Example :

const divStyle = {

color: "blue",

backgroundImage: "url(" + imgUrl + ")",

};

function HelloWorldComponent() {

return <div style={divStyle}>Hello World!</div>;

}

**45) How to loop inside JSX?**

You can simply use Array.prototype.map with ES6 arrow function syntax.

For example, the items array of objects is mapped into an array of components:

<body>

{items.map((item) => (

<SomeComponent key={item.id} name={item.name} />

))}

</body>

**46) Why is it necessary to start component names with a capital letter?**

In React, it is necessary to start component names with a capital letter. If we start the component name with lower case, it will throw an error as an unrecognized tag. It is because, in JSX, lower case tag names are considered as HTML tags.

**47) Mention one difference between Props and State.**

State is mutable while Props are immutable. This means that the state is internal and managed by the component while props are external and managed by anything that renders the component.

**48) What does ‘key’ do in React list?**

Keys are used for providing identity to list elements.

**49) How will you update props in React?**

It is not possible as we can’t update props in ReactJS as they are read-only.

**50) Can JavaScript urls be used in React 16.9?**

Yes, but it will give a warning in the console.

**51) Explain the ‘Store’ feature**.

It is a feature of Redux that allows the application to save the entire State at one place. This allows you to get regular updates from the store directly.

**52) What is the Presentational segment?**

It is a component in React that just renders HTML and its only function is presentation markup.

**53) Explain Props in ReactJS.**

Props in React mean properties. They act as a communication channel from parent to child.

**54) In Redux, what is an action?**

In Redux, an action is an object that contains information about an event that has occurred in an application.

**55) What is React Router?**

It is a library for routing in React.

**56) Can you list some animation packages in React?**

Some animation packages are as follows:

React Motion

React Transition Group

React Spring

**57) What is the most advised way for naming components?**

By using reference rather than using displayName because it provides a clear and concise name for the component.

**58) What are pure components in ReactJS?**

A React component is considered to be pure if it produces the same output when given the same set of state and props.

**59) Explain children's prop.**

Children prop are used to pass data from the parent component to the children component.

**60) What is React Fiber?**

It is the new reconciliation engine in React 16 and is used to enable incremental rendering of virtual DOM.

**61) What can you do to spill the reducers?**

We can use event actions.

**62) How do you implement routing in ReactJS?**

React Router is available as an npm package, and can be added to your application with a single line of code. Once installed, you can define your routes in your application and use the component to link between them.

**63) Can you attach a JSX element to other JSX components?**

Yes, you can add JSX elements to other JSX components.

**64) How can you fetch data with React Hooks?**

We can use useEffect that fetches the data through axios from the API.

**65) What is one-way data flow in React?**

React uses unidirectional data flow which means you can transfer data from parent to child and not the other way around.

**66) Why do we use keys in Lists?**

The main reason why we use keys in lists are as follows

It is an identifier to find which items have changed, updated, or deleted from lists.

Find out which items need to be re-rendered

**67) Explain React forms?**

React forms allow users to interact with web applications. They help users to enter the information when required. They contain buttons, text fields, checkboxes, etc.

**68) What methods are invoked during error handling?**

componentDidCatch()

**69) What is the purpose of callback function as an argument of setState()?**

The callback function is invoked when setState finished and the component gets rendered. Since setState() is asynchronous the callback function is used for any post action.

**70) How to pass a parameter to an event handler or callback?**

You can use an arrow function to wrap around an event handler and pass parameters:

<button onClick={() => this.handleClick(id)} />

This is an equivalent to calling .bind:

<button onClick={this.handleClick.bind(this, id)} />

**71) What are inline conditional expressions?**

You can use either if statements or ternary expressions which are available from JS to conditionally render expressions. Apart from these approaches, you can also embed any expressions in JSX by wrapping them in curly braces and then followed by JS logical operator &&.

<h1>Hello!</h1>

d{

messages.length > 0 && !isLogin?

<h2>

You have {messages.length} unread messages.

</h2>

:

<h2>

You don't have unread messages.

</h2>

}

**72) What are controlled components?**

A component that controls the input elements within the forms on subsequent user input is called Controlled Component, i.e, every state mutation will have an associated handler function.

For example, to write all the names in uppercase letters, we use handleChange as below,

handleChange(event) {

this.setState({value: event.target.value.toUpperCase()})

}

**73) What are Higher-Order Components?**

A higher-order component (HOC) is a function that takes a component and returns a new component. Basically, it's a pattern that is derived from React's compositional nature.We call them pure components because they can accept any dynamically provided child component but they won't modify or copy any behavior from their input components.

const

EnhancedComponent = higherOrderComponent (Wrapped Compone nt)

**74) What is context?**

Context provides a way to pass data through the component tree without having to pass props down manually at every level.

**75) What is reconciliation?**

When a component's props or state change, React decides whether an actual DOM update is necessary by comparing the newly returned element with the previously rendered one. When they are not equal, React will update the DOM. This process is called reconciliation.

**76) What are stateless components?**

If the behaviour of a component is independent of its state then it can be a stateless component. You can use either a function or a class for creating stateless components. But unless you need to use a lifecycle hook in your components, you should go for function components.

**77) What are stateful components?**

If the behaviour of a component is dependent on the state of the component then it can be termed as stateful component. These stateful components are always class components and have a state that gets initialized in the constructor.

**78) How to fetch data with React Hooks?**

The effect hook called useEffect is used to fetch the data with axios from the API and to set the data in the local state of the component with the state hook’s update function.

**79) What is a switching component?**

A switching component is a component that renders one of many components. We need to use object to map prop values to components.

**80) Is it possible to use async/await in plain React?**

If you want to use async/await in React, you will need Babel and transform-async-to-generator plugin. React Native ships with Babel and a set of transforms.

**81) What are render props?**

Render Props is a simple technique for sharing code between components using a prop whose value is a function. The below component uses render prop which returns a React element.

**82) What is flux?**

Flux is an application design paradigm used as a replacement for the more traditional MVC pattern. It is not a framework or a library but a new kind of architecture that complements React and the concept of Unidirectional Data Flow. Facebook uses this pattern internally when working with React.

**83) What is the difference between React context and React Redux?**

You can use Context in your application directly and is going to be great for passing down data to deeply nested components which what it was designed for. Whereas Redux is much more powerful and provides a large number of features that the Context API doesn't provide. Also, React Redux uses context internally but it doesn't expose this fact in the public API.

**84) How to re-render the view when the browser is resized?**

It is possible to listen to the resize event in componentDidMount() and then update the width and height dimensions. It requires the removal of the event listener in the componentWillUnmount() method.

**85) What is the main purpose of constructor?**

The constructor is mainly used for two purposes,

* To initialize local state by assigning object to this.state
* For binding event handler methods to the instance

For example, the below code covers both the above cases

constructor(props) {

super(props);

// Don't call this.setState() here!

this.state = { counter: 0 };

this.handleClick = this.handleClick.bind(this); }

**86) What are the basic rules of hooks?**

Hooks have some rules: They must be used only at the top level of a function component, hooks mustbe called in the same order every time the component renders, and they cannot be called conditionally.

**87) What are reducers in Redux?**

Reducers in Redux are pure functions that specify how the application's state changes in response to actions. They take the current state and an action as input and return a new state based on that action.

**88) What is React Native?**

React Native is a framework for building native mobile applications using React. It allows developers to write mobile apps using JavaScript.

89) What is the purpose of the React.Fragment component?

React.Fragment is a built-in component in React that allows you to group multiple elements without adding an extra node to the DOM.

90) What is the useReducer hook in React?

The useReducer hook is a built-in hook in React that allows you to manage state using a reducer function. It is an alternative to useState and is useful for managing more complex state logic or state transitions.

91) What is the purpose of the componentWillUnmount() method in React class components?

The componentWillUnmount() method is a lifecycle method in React class components that is called just before a component is unmounted and removed from the DOM. It allows for performing cleanup tasks such as removing event listeners or cancelling subscriptions.

92) Explain Flux Concept in React.

It is a new kind of architecture used for client-side web applications. Supports the concept of Unidirectional Data Flow.

93) What is yield catchphrase in JavaScript?

It is used to delay and resume a generator work.

94) What are error boundaries?

They help catch errors in child components, log the error and show fallback UI.

95) What are refs?

They are an attribute of DOM elements. It is a function provided by React to access these elements. You might want to use them in case when you want to change the value of a child component without use of props.