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.

1. **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

1. **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)

1. **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. |
| **Performance** is Faster as they do not have state and lifecycle, react needs to do less work to render these components. | **Performance** is Slower as they have state and lifecycle, react needs to do comparatively more work to render these components. |
| Functional components codes tend to be shorter and more concise | Class components code require the boilerplate code, such as a constructor method and the use of “this” to access props and state. |

1. **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.

1. **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.

1. **What does setState() method do?**

* **s**etState() method will modify value of the state.
* Then it will Schedule a call to our render method.

1. **What is an Event Handler?**

Event Handler are functions which will be called (invoked) by the browsers

when an event occurs.

1. **What are Synthetic events?**

Synthetic events are representative of real events a.k.a wrapper around real

events. Because we have to deal with different implementation of event

functions in different browsers, a synthetic event acts as a specification for a

single event access all browsers.

1. **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’.

1. **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 as ReactDOM. This process is called reconciliation.

1. **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.

1. **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.

1. **What is ReactJS?**

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

1. **Does React use HTML?**

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

1. **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.

1. **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.

1. **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

1. **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.

1. **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.

1. **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.

1. **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.

1. **Name two types of React component.**

Two types of react Components are:

* Function component
* Class component

1. **How can you update state in React JS?**

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

1. **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.

1. **What are the lifecycle steps of React?**

Important lifecycle steps of React js are:

* Initialization
* State/Property updates
* Destruction are the lifecycle of React

1. **What is ‘create-react-app’?**

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

1. **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.

1. **What are children prop?**

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

{props.children}

1. **What is the use of empty tags <> </>?**

Empty tags are used in React for declaring fragments.

1. **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.

1. **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.

1. **Name any five predefined prototypes used in React.**

Most important prototype used in React js are:

* number
* string
* array
* object
* Element

1. **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.

1. **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.

1. **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.

1. **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.

1. **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.

1. **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.

1. **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.

1. **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.