**1. What is ReactJS ?**

React is a **JavaScript** library created by **Facebook**

React is a **User Interface** (UI) library

React is a tool for building **UI components**

**2. What is JSX in React?**

JSX stands for **JavaScript XML**. JSX allows us to write HTML in React. JSX makes it easier to write and add HTML in React.

**3. What is Hooks in ReactJS?**

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

**4. What is props in ReactJS?**

The props are **a type of object where the value of attributes of a tag is stored**. The word “props” implies “properties”, and its working functionality is quite similar to HTML attributes. Basically, these props components are read-only components.

**5. What is useEffect in React?**

The useEffect Hook **allows us to perform side effects in your components**. Some examples of side effects are: fetching data, directly updating the DOM, and timers. useEffect accepts two arguments. The second argument is optional.

**6. Why state is used in React?**

The state is a built-in React object that is used **to contain data or information about the component**. A component's state can change over time; whenever it changes, the component re-renders.

**7. What is Babel in React?**

Babel is **a toolchain that is mainly used to convert ECMAScript 2015+ code into a backwards compatible version of JavaScript in current and older browsers or environments**.

**8. What is NPM in React?**

npm is an abbreviation used for the **node package manager**. It is a package manager for JavaScript. It is the default package manager that comes with NodeJS when you install it.

**9. What is ES6 in React?**

ES6 stands for **ECMAScript 6**. ECMAScript was created to standardize JavaScript, and ES6 is the 6th version of ECMAScript, it was published in 2015, and is also known as ECMAScript 2015.

**10. Why Webpack is used in React?**

Webpack is a popular module bundling system built on top of Node. js. It can handle not only combination and minification of JavaScript and CSS files, but also other assets such as image files (spriting) through the use of plugins.

. Your project will bundle its own copy of react and react-dom with webpack, and ReactJS.NET will be used only for server-side rendering.

**11. What is lazy loading in React?**

In essence, lazy loading means that **a component or a part of code must get loaded when it is required**. It is also referred to as code splitting and data fetching . Talking about React specifically, it bundles the complete code and deploys all of it at the same time.

**12. What is the main function of React?**

React allows developers to create large web applications that can change data, without reloading the page. The main purpose of React is **to be fast, scalable, and simple**. It works only on user interfaces in the application. This corresponds to the view in the MVC template

**13. What is useState Hook?**

useState Hook **allows us to track state in a function component**. State generally refers to data or properties that need to be tracking in an application.

**14. Features of React:**

* JSX (JavaScript Syntax Extension)
* Virtual DOM
* One-way data binding
* Performance
* Extensions
* Conditional statements
* Components
* Simplicity

**15. Can web browsers read JSX directly?**

* Web browsers cannot read JSX directly. This is because they are built to only read regular JS objects and JSX is not a regular JavaScript object
* For a web browser to read a JSX file, the file needs to be transformed into a regular JavaScript object. For this, we use Babel

### ****16 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.

### ****17 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 | one-way binding |

**18 What is an event in React?**

An event is an action that a user or system may trigger, such as pressing a key, a mouse click, etc.

* React events are named using camelCase, rather than lowercase in HTML.
* With JSX, you pass a function as the event handler, rather than a string in HTML.

|  |
| --- |
| <Button onPress={lightItUp} /> |

### ****19 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 |

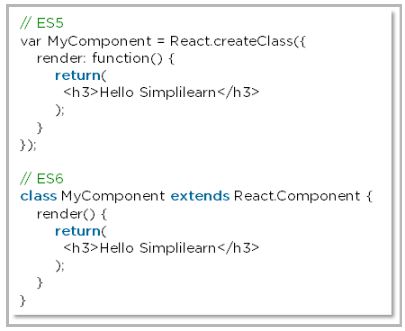
### ****20 What is the difference between props and states?****

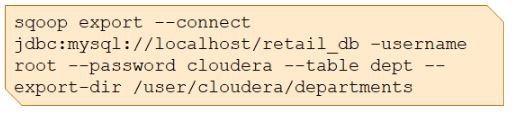
|  |  |  |
| --- | --- | --- |
| **Condition** | **Props** | **States** |
| Changes in child components | Yes | No |
| Parent component changing values | Yes | No |
| Changes inside components | No | Yes |

**21. What is the difference between the ES6 and ES5 standards?**

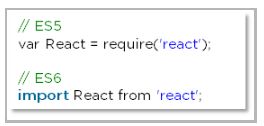
These are the few instances where ES6 syntax has changed from ES5 syntax:

* Components and Function





* require vs import

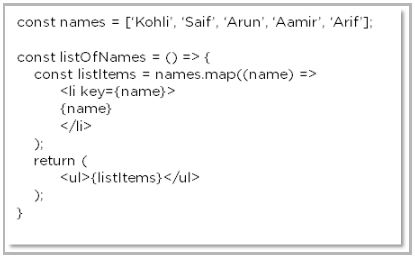


**22. What are synthetic events in React?**

* Synthetic events combine the response of different browser's native events into one API, ensuring that the events are consistent across different browsers.
* The application is consistent regardless of the browser it is running in. Here, preventDefault is a synthetic event.
* function ActionLink(){
* function handleClick(e){
* e.preventDefault();
* console.log("you justed clicked link")
* }
* return(
* <a href="#" onClick= {handleClick} > Click Me</a>
* );
* }
* export default ActionLink;

**23.Explain how lists work in React**

* We create lists in React as we do in regular JavaScript. using the map() function

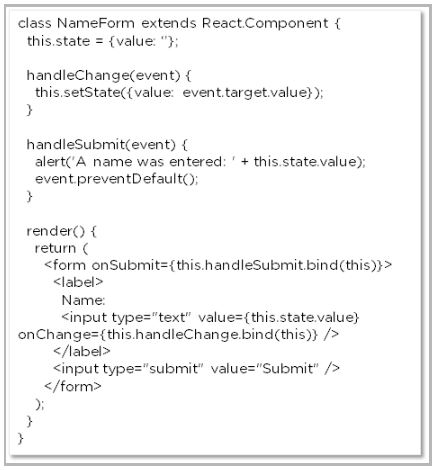


**24. What are forms in React?**

* User can interact with the application and enter the required information whenever needed. Form contain certain elements, such as text fields, buttons, checkboxes, radio buttons, etc

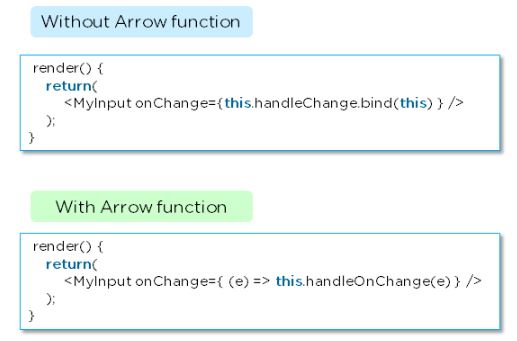
### ****25. How do you create forms in React?****

We create forms in React by doing the following:



**26. What is an arrow function and how is it used in React?**

* An arrow function is a short way of writing a function to React.
* It is unnecessary to bind ‘this’ inside the constructor when using an arrow function. This prevents bugs caused by the use of ‘this’ in React callbacks.



### ****27. How is React different from React Native?****

|  |  |
| --- | --- |
| React | React Native |
| Release | 2013 | 2015 |
| Platform | Web | Mobile – Android, iOS |
| HTML | Yes | No |
| CSS | Yes | No |
| Prerequisites | JavaScript, HTML, CSS | React.js |

### ****What are the components in React?****

### Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and return HTML. Components come in two types, Class components and Function components

### Functionl Component:

### A functional component is just a plain JavaScript pure function that accepts props as an argument and returns a React element(JSX).

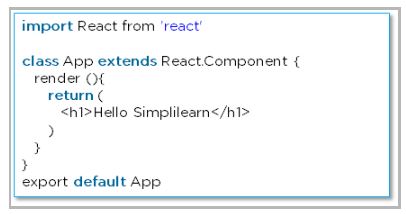
|  |
| --- |
| function Greeting(props) {    return <h1>Welcome to {props.name}</h1>;  } |

* **Class Components:** A class component requires you to extend from React. Component and create a render function which returns a React element.

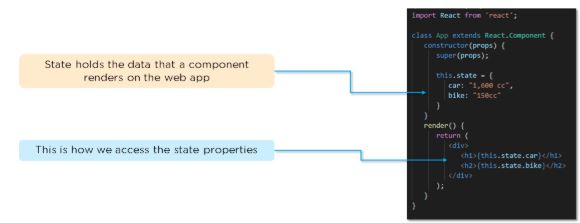
|  |
| --- |
| class Greeting extends React.Component {    render() {      return <h1>Welcome to {this.props.name}</h1>;    }  } |

**28. What is the use of render() in React?**

* It is required for each component to have a render() function. This function returns the HTML, which is to be displayed in the component.
* If you need to render more than one element, all of the elements must be inside one parent tag like <div>, <form>.

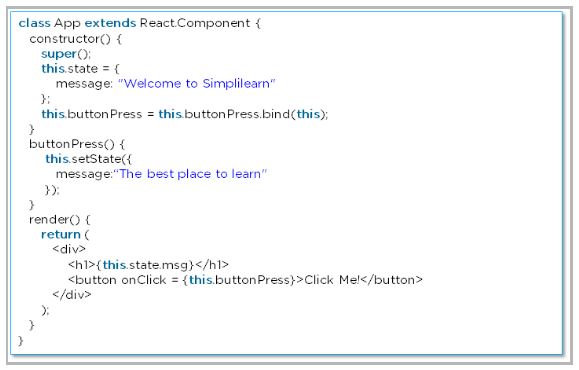


### ****29. How do you implement state in React?****

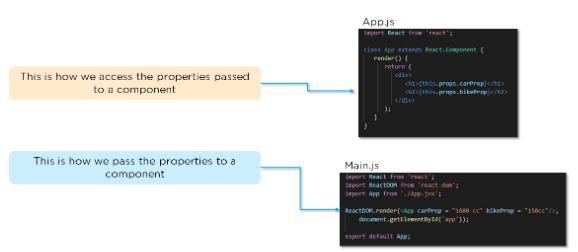


### ****30. How do you update the state of a component?****

We can update the state of a component by using the built-in **‘setState()’**method:



### ****31. How do you pass props between components?****

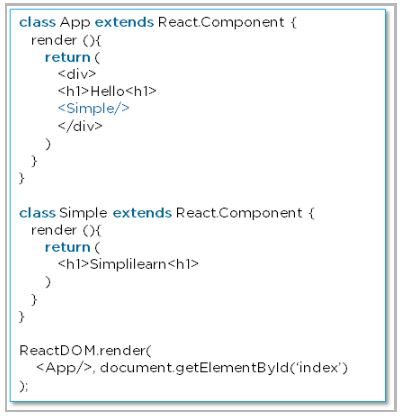


### ****32. What is a higher-order component in React?****

A higher-order component acts as a container for other components. This helps to keep components simple and enables re-usability. They are generally used when multiple components have to use a common logic.

### ****33. How can you embed two or more components into one?****

We can embed two or more components into one using this method:



**34. Explain the lifecycle methods of components**.

* getInitialState(): This is executed before the creation of the component.
* 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.

### **ReactJS Router Questions**

### ****39. How is React routing different from conventional routing?****

|  |  |  |
| --- | --- | --- |
| **SN** | **React Routing** | **Conventional routing** |
| 1. | Single HTML page | Each view is a new HTML file |
| 2. | The user navigates multiple views in the same file | The user navigates multiple files for each view |
| 3. | The page does not refresh since it is a single file | The page refreshes every time user navigates |
| 4. | Improved performance | Slower performance |

### ****40. What is React Router?****

React Router is a routing library built on top of React, which is used to create routes in a React application.

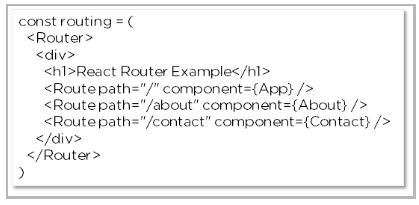
### ****36. Why do we need to React Router?****

* It maintains consistent structure and behaviour and is used to develop single-page web applications.
* Enables multiple views in a single application by defining multiple routes in the React application.

### ****41. How do you implement React routing?****

We can implement routing in our React application using this method:

Considering we have the components **App**, **About**, and **Contact** in our application:

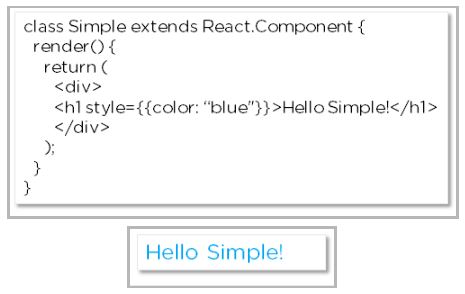


## **ReactJS Styling Questions**

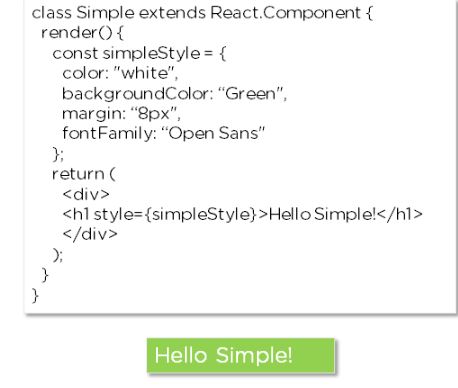
### ****42. How do you style React components?****

There are several ways in which we can style React components:

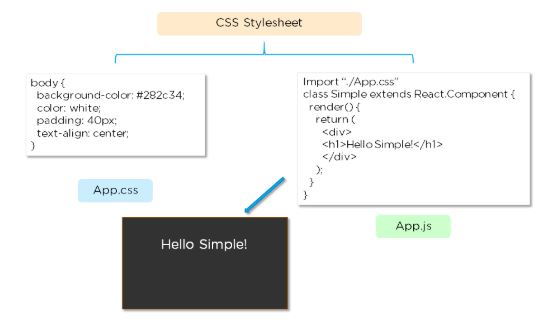
* **Inline Styling**



* **JavaScript Object**



* **CSS Stylesheet**



### ****43. Explain the use of CSS modules in React.****

* The CSS module file is created with the **.module.css** extension
* The CSS inside a module file is available only for the component that imported it, so there are no naming conflicts while styling the components.



**44. What are fragments?**

In was introduced in React 16.2 version. In React, Fragments are used for components to return multiple elements. It allows you to group a list of multiple children without adding an extra node to the DOM.

**Example**

1. render() {
2. **return** (
3. <React.Fragment>
4. <ChildA />
5. <ChildB />
6. <ChildC />
7. </React.Fragment>
8. )
9. }

There is also a shorthand syntax exists for declaring Fragments, but it's not supported in many tools:

1. render() {
2. **return** (
3. <>
4. <ChildA />
5. <ChildB />
6. <ChildC />
7. </>
8. )
9. }

**45.Why are fragments better than container divs?**

* Fragments are faster and consume less memory because it did not create an extra DOM node.
* Some CSS styling like CSS Grid and Flexbox have a special parent-child relationship and add <div> tags in the middle, which makes it hard to keep the desired layout.
* The DOM Inspector is less cluttered.

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

In React, a stateful component is a component that holds some state

**47. 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

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

In React, Controlled Components are **those in which form's data is handled by the component's state**. It takes its current value through props and makes changes through callbacks like onClick,onChange, etc. A parent component manages its own state and passes the new values as props to the controlled component

**49. What are pure components in React?**

A React component is considered pure **if it renders the same output for the same state and props**. For this type of class component, React provides the PureComponent base class. Class components that extend the React. PureComponent class are treated as pure components

### ****50. 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.

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

In a controlled component, form data is handled by a React component. The alternative is uncontrolled components, where form data is handled by the DOM itself. To write an uncontrolled component, instead of writing an event handler for every state update, you can use a ref to get form values from the DOM.

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

### createElement is the code that JSX gets compiled or converted into and is used by reacting to create elements.

### cloneElement is used for cloning elements and passing them new props

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

StrictMode is a React Developer Tool, primarily used to identify potential issues in a web application

**54. 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

**.**

### 57. 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.

### 58. 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

### 59. 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.

### 60. 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.

**useRef()**

The useRef Hook allows you to persist values between renders.

It can be used to store a mutable value that does not cause a re-render when updated.

It can be used to access a DOM element directly.

**useContext()**

useContext” hook is used **to create common data that can be accessed throughout the component hierarchy without passing the props down manually to each level**. Context defined will be available to all the child components without involving “props”.

**useReducer()**

The useReducer hook is used for complex state manipulations and state transitions.

The useReducer Hook accepts two arguments. useReducer(<reducer>, <initialState>) The reducer function contains your custom state logic and the initialState can be a simple value but generally will contain an object. The useReducer Hook **returns the current state and a dispatch method**

**useCallback()**

The useCallback hook is used when you have a component in which the child is rerendering again and again without need. **Pass an inline callback and an array of dependencies**. useCallback will return a memoized version of the callback that only changes if one of the dependencies has changed.

**useMemo()**

The React useMemo Hook returns a memoized value.

The useMemo Hook only runs when one of its dependencies update.

This can improve performance.

**customHook()**

Hooks are reusable functions.

When you have component logic that needs to be used by multiple components, we can extract that logic to a custom Hook

**What is “React Fiber”?**

Fiber is the new reconciliation engine in React 16. Its main goal is to enable incremental rendering of the virtual DOM.

**Diff b/w usecallback and usememo?**

useCallback and useMemo both expect a function and an array of dependencies. The difference is that useCallback returns its function when the dependencies change while useMemo calls its function and returns the result.

**What is diff b/w contextapi and redux?**

Context API is a light-weight solution which is more suited for passing data from a parent to a deeply nested child and Redux is a more robust State Management solution.

**What is fallback used for in suspense?**

React Suspense is a React component that suspends a component('s) being render until a certain condition has been met, and will display a fallback option. This fallback option is required, and it may be a string or another React component such as a spinner.

**What is design pattern in React?**

React Design Patterns are **core enablers of React library that enable React Developers to create complex React apps in an easily maintainable manner**, which are also flexible, deliver better results and drive better performance.

**What is forward ref**

ForwardRef() is a utility function in react that let you expose a child components DOM to a parent component with a ref. Usually the parent component passes the props and data to the child component. But in some instances like when working with input or where components need to respond with user interactions.

**What is the difference between useEffect and useLayoutEffect SSR?**

useEffect runs asynchronously after the render has been committed to the screen, while useLayoutEffect runs synchronously after all DOM mutations but before the browser has a chance to paint.