REACT INTERVIEW QUESTIONS

1. What is ReactJS?

ReactJS is a JavaScript library used to build reusable components for the view layer in MVC architecture. It is highly efficient and uses a virtual DOM to render components. It works on the client side and is written in JSX.

Important Features of React:

- **Virtual DOM:** React uses a virtual DOM to efficiently update and render components, ensuring fast performance by minimizing direct DOM manipulations.
- **Component-Based Architecture:** React builds UI using reusable, isolated components, making code more modular, maintainable, and scalable.
- **Hooks**: React hooks allow functional components to manage state and side effects, making them powerful and more flexible.

2. Explain the MVC architecture.

The Model-View-Controller (MVC) framework is an architectural/design pattern that separates an application into three main logical components Model, View, and Controller. Each architectural component is built to handle specific development aspects of an application. It isolates the business, logic, and presentation layer from each other

3. Explain the building blocks of React.

The five main building blocks of React are:

- **Components:** These are reusable blocks of code that return HTML.
- **JSX:** It stands for JavaScript and XML and allows you to write HTML in React.
- **Props and State:** props are like function parameters and State is similar to variables.
- **Context:** This allows data to be passed through components as props in a hierarchy.
- **Virtual DOM:** It is a lightweight copy of the actual DOM which makes DOM manipulation easier.

4. Explain props and state in React with differences

Props are used to pass data from one component to another. The state is local data storage that is local to the component only and cannot be passed to other components.

Here is the difference table of props and state In react

PROPS	STATE
The Data is passed from one component to another.	The Data is passed within the component only.
It is Immutable (cannot be modified).	It is Mutable (can be modified).
Props can be used with state and functional components.	The state can be used only with the state components/class component (Before 16.0).
Props are read-only.	The state is both read and write.

5. What is virtual DOM in React?

The **Virtual DOM** in React is an in-memory representation of the actual DOM. It helps React efficiently update and render the user interface by comparing the current and previous virtual DOM states using a process called **diffing**.

How Virtual DOM Works

- **Efficient Rendering**: The Virtual DOM is an in-memory representation of the actual DOM that React uses to optimize the process of updating and rendering UI changes.
- **Diffing Algorithm**: React compares the current and previous versions of the Virtual DOM using a diffing algorithm, identifying the minimal set of changes required to update the real DOM.
- **Batch Updates**: Instead of updating the real DOM immediately, React batches multiple changes to reduce unnecessary re-renders, improving performance.
- **Faster Updates**: Since updating the real DOM is slow, React minimizes direct DOM manipulations by only making updates where necessary after comparing the Virtual DOM.
- **Declarative UI**: With the Virtual DOM, React allows developers to write code in a declarative style, letting React handle when and how to efficiently update the UI.

6. What is JSX?

JSX is basically a syntax extension of regular JavaScript and is used to create React elements. These elements are then rendered to the React DOM. All the React components are written in JSX. To embed any JavaScript expression in a piece of code written in JSX we will have to wrap that expression in curly braces {}.

Example of JSX: The name written in curly braces { } signifies JSX

```
const name = "Learner";

const element = (
  <h1>
    Hello,
    {name}.Welcome to GeeksforGeeks.
  </h1>
);
```

7. What are components and their type in React?

A Component is one of the core building blocks of React. In other words, we can say that every application you will develop in React will be made up of pieces called components. Components make the task of building UIs much easier.

In React, we mainly have two types of components:

- **Functional Components:** Functional components are simply javascript functions. We can create a functional component in React by writing a javascript function.
- **Class Components:** The class components are a little more complex than the functional components. The functional components are not aware of the other components in your program whereas the class components can work with each other. We can pass data from one class component to another class component.

8. How do browsers read JSX?

In general, browsers are not capable of reading JSX and only can read pure JavaScript. The web browsers read JSX with the help of a transpiler. Transpilers are used to convert JSX into JavaScript. The transpiler used is called Babel

9. Explain the steps to create a react application and print Hello World?

To install React, first, make sure Node is installed on your computer. After installing Node. Open the terminal and type the following command.
npx create-react-app <<Application_Name>>
Navigate to the folder.
cd <<Application_Name>>
This is the first code of ReactJS Hello World!
Javascript

```
import React from "react";
import "./App.css";
function App() {
   return <div className="App">Hello World !</div>;
}
export default App;
```

Type the following command to run the application
npm start

10. How to create an event in React?

To create an event in React, attach an event handler like onClick, onChange, etc., to a JSX element. Define the handler function to specify the action when the event is triggered, such as updating state or executing logic.

11. Explain the creation of a List in react?

Lists are very useful when it comes to developing the UI of any website. Lists are mainly used for displaying menus on a website, for example, the navbar menu. To create a list in React use the map method of array as follows.
Javascript

```
import React from "react";
import ReactDOM from "react-dom";
const numbers = [1, 2, 3, 4, 5];
const updatedNums = numbers.map((number) => {
   return <li>{number}</li>;
});
ReactDOM.render(<ul>{updatedNums}</ul>, document.getElementById("root"));
```

12. What is a key in React?

A “key” is a special string attribute you need to include when creating lists of elements in React. Keys are used in React to identify which items in the list are changed, updated, or deleted. In other words, we can say that keys are used to give an identity to the elements in the lists.

13. How to write a comment in React?

There are two ways to write comments in React.

- **Multi-line comment:** We can write multi-line comments in React using the asterisk format /* */.
- **Single line comment:** We can write single comments in React using the double forward slash //.

14. Explain the difference between React and Angular?

Field	React.js	Angular
Used as	React.js is a JavaScript library. As it indicates react js updates only the virtual DOM is present and the data flow is always in a single direction.	Angular is a framework. Angular updates the Real DOM and the data flow is ensured in the architecture in both directions.
Architecture	React.js is more simplified as it follows MVC ie., Model View Control.	The architecture is complex as it follows MVVM models ie., Model View-ViewModel.
Scalability	It is highly scalable.	It is less scalable than React JS.
Data Binding	It supports Uni-directional data binding which is one-way data binding.	It supports Bi-directional data binding which is two way data binding.
DOM	It has a virtual DOM.	It has regular DOM.

15. Explain the use of render method in React?

React renders HTML to the web page by using a function called render(). The purpose of the function is to display the specified HTML code inside the specified HTML element. In the render() method, we can read props and state and return our JSX code to the root component of our app.

16. What is state in React?

The state is an instance of React Component Class that can be defined as an object of a set of observable properties that control the behaviour of the component. In other words, the State of a component is an object that holds some information that may change over the lifetime of the component.

17. Explain props in React?

React allows us to pass information to a Component using something called props (which stands for properties). Props are objects which can be used inside a component
We can access any props inside from the component's class to which the props is passed. The props can be accessed as shown below:
this.props.propName;

18. What is higher-order component in React?

Higher-order components or HOC is the advanced method of reusing the component functionality logic. It simply takes the original component and returns the enhanced component. HOC are beneficial as they are easy to code and read. Also, helps to get rid of copying the same logic in every component.

19. Explain the difference between functional and class component in React?

Here we have difference table of functional and class component in React

Functional Components	Class Components
A functional component is just a plain JavaScript pure function that accepts props as an argument	A class component requires you to extend from React. Component and create a render function
No render method used	It must have the render() method returning JSX
Also known as Stateless components	Also known as Stateful components
React lifecycle methods (for example, componentDidMount) cannot be used in functional components.	React lifecycle methods can be used inside class components (for example, componentDidMount).
Constructors are not used.	Constructor is used as it needs to store state.

20. Explain one way data binding in React?

ReactJS uses one-way data binding which can be Component to View or View to Component. It is also known as one-way data flow, which means the data has one, and only one way to be transferred to other parts of the application. In essence, this means child components are not able to update the data that is coming from the parent component. It is easy to debug and less prone to errors.

React Intermediate Interview Questions

Here, we cover all intermediate level react interview questions with answers, that recommeded for freshers as well as for experienced professionals having 1 – 2 years of experience.

21. What is conditional rendering in React?

Conditional rendering in React involves selectively rendering components based on specified conditions. By evaluating these conditions, developers can control which components are displayed, allowing for dynamic and responsive user interfaces in React applications.

Let us look at this sample code to understand conditional rendering.

{isLoggedIn == false ? <DisplayLoggedOut /> : <DisplayLoggedIn />}

Here if the boolean isLoggedIn is false then the DisplayLoggedOut component will be rendered otherwise DisplayLoggedIn component will be rendered.

22. What is react router?

React Router is a standard library for routing in React. It enables the navigation among views of various components in a React Application, allows changing the browser URL, and keeps the UI in sync with the URL.

To install react router type the following command.

npm i react-router-dom

23. Explain the components of a react-router

The main components of a react-router are:

1. **Router(usually imported as BrowserRouter):** It is the parent component that is used to store all of the other components. Everything within this will be part of the routing functionality
2. **Switch:** The switch component is used to render only the first route that matches the location rather than rendering all matching routes.
3. **Route:** This component checks the current URL and displays the component associated with that exact path. All routes are placed within the switch components.
4. **Link:** The Link component is used to create links to different routes.

24. Explain the lifecycle methods of components

A React Component can go through four stages of its life as follows.

- **Initialization:** This is the stage where the component is constructed with the given Props and default state. This is done in the constructor of a Component Class.
- **Mounting:** Mounting is the stage of rendering the JSX returned by the render method itself.
- **Updating:** Updating is the stage when the state of a component is updated and the application is repainted.
- **Unmounting:** As the name suggests Unmounting is the final step of the component lifecycle where the component is removed from the page.

25. Explain the methods used in mounting phase of components

Mounting is the phase of the component lifecycle when the initialization of the component is completed and the component is mounted on the DOM and rendered for the first time on the webpage. he mounting phase consists of two such predefined functions as described below

- componentWillMount() Function: This function is invoked right before the component is mounted on the DOM.
- componentDidMount() Function: This function is invoked right after the component is mounted on the DOM.

26. What is this.setState function in React?

We use the setState() method to change the state object. It ensures that the component has been updated and calls for re-rendering of the component. The state object of a component may contain multiple attributes and React allows using setState() function to update only a subset of those attributes as well as using multiple setState() methods to update each attribute value independently.

27. What is the use of ref in React?

Refs are a function provided by React to access the DOM element and the React element that you might have created on your own. They are used in cases where we want to change the value of a child component, without making use of props and all. They have wide functionality as we can use callbacks with them.

The syntax to use ref is

const node = this.myCallRef.current;

28. What are hooks in React?

Hooks are a new addition in React 16.8. They let developers use state and other React features without writing a class. Hooks doesn't violate any existing React concepts. Instead, Hooks provide a direct API to react concepts such as props, state, context, refs and life-cycle

29. Explain the useState hook in React?

The most used hook in React is the useState() hook. Using this hook we can declare a state variable inside a function but only one state variable can be declared using a single

useState() hook. Whenever the useState() hook is used, the value of the state variable is changed and the new variable is stored in a new cell in the stack.
When you use useState(), you declare a state variable and a function to update that state. React then manages this state internally and triggers a re-render of the component when the state changes. This allows functional components to maintain and update their internal state over time.
We have to import this hook in React using the following syntax
import {useState} from 'react'

30. Explain the useEffect hook in react?

The useEffect hook in React eliminates the side effect of using class based components. It is used as an alternative to componentDidUpdate() method. The useEffect hook accepts two arguments where second argument is optional.
useEffect(function, dependency)
The dependency decides when the component will be updated again after rendering.

31. What is React Fragments?

when we are trying to render more than one root element we have to put the entire content inside the 'div' tag which is not loved by many developers. So since React 16.2 version, Fragments were introduced, and we use them instead of the extraneous 'div' tag.
The following syntax is used to create fragment in react.

```
<React.Fragment>
   <h2>Child-1</h2>
   <p> Child-2</p>
</React.Fragment>
```

32. What is a react developer tool?

React Developer Tools is a Chrome DevTools extension for the React JavaScript library. A very useful tool, if you are working on React.js applications. This extension adds React debugging tools to the Chrome Developer Tools. It helps you to inspect and edit the React component tree that builds the page, and for each component, one can check the props, the state, hooks, etc.

33. How to use styles in ReactJS?

CSS modules are a way to locally scope the content of your CSS file. We can create a CSS module file by naming our CSS file as App.modules.css and then it can be imported inside App.js file using the special syntax mentioned below.

Syntax:

```
import styles from './App.module.css';
```

34. Explain styled components in React?

Styled-component Module allows us to write CSS within JavaScript in a very modular and reusable way in React. Instead of having one global CSS file for a React project, we can use styled-component for enhancing the developer experience. It also removes the mapping between components and styles – using components as a low-level styling construct
The command to install styled components is
npm i styled-components
Using the below code we can custom style a button in React
Javascript

```
import styled from 'styled-components'
const Button = styled.div`
```

```
  width : 100px ;
  cursor: pointer ;
  text-decoration : none;
`

export default Button;
```

35. What is prop drilling and its disadvantages?

Prop drilling is basically a situation when the same data is being sent at almost every level due to requirements in the final level. The problem with Prop Drilling is that whenever data from the Parent component will be needed, it would have to come from each level, Regardless of the fact that it is not needed there and simply needed in last.

*For further reading, check out our dedicated article on **Intermediate ReactJS Intermediate Interview Questions**. Inside, you'll discover over 20 questions with detailed answers.*

React Interview Questions For Experienced

Here, we cover **advanced react interview questions with answers** for experienced professionals, who have over 5+ years of experience.

36. What is custom hooks in React?

Custom hooks are normal JavaScript functions whose names start with "use" and they may call other hooks. We use custom hooks to maintain the DRY concept that is Don't Repeat Yourself. It helps us to write a logic once and use it anywhere in the code.

37. How to optimize a React code?

We can improve our react code by following these practices:

- Using binding functions in constructors
- Eliminating the use of inline attributes as they slow the process of loading
- Avoiding extra tags by using React fragments
- Lazy loading

38. What is the difference between useref and createRef in React ?

Here is the difference table of useref and createRef in React

useRef	**createRef**
It is a hook.	It is a function.
It uses the same ref throughout.	It creates a new ref every time.
It saves its value between re-renders in a functional component.	It creates a new ref for every re-render.
It returns a mutable ref object.	It returns a read-only ref object.
The refs created using the useRef can persist for the entire component lifetime.	The refs created using the createRef can be referenced throughout the component.
It is used in functional components.	It is used in class components.

39. What is react-redux?

React-redux is a state management tool which makes it easier to pass these states from one component to another irrespective of their position in the component tree and hence prevents the complexity of the application. As the number of components in our application increases it becomes difficult to pass state as props to multiple components. To overcome this situation we use react-redux

40. What are benefits of using react-redux?

They are several benfits of using react-redux such as:

- It provides centralized state management i.e. a single store for whole application
- It optimizes performance as it prevents re-rendering of component
- Makes the process of debugging easier
- Since it offers persistent state management therefore storing data for long times become easier

41. Explain the core components of react-redux?

There are four fundamental concepts of redux in react which decide how the data will flow through components

- Redux Store: It is an object that holds the application state
- Action Creators: These are functions that return actions (objects).
- Actions: Actions are simple objects which conventionally have two properties- type and payload
- Reducers: Reducers are pure functions that update the state of the application in response to actions

42. How can we combine multiple reducers in React?

When working with Redux we sometimes require multiple reducers. In many cases, multiple actions are needed, resulting in the requirement of multiple reducers. However, this can become problematic when creating the Redux store. To manage the multiple reducers we have function called combineReducers in the redux. This basically helps to combine multiple reducers into a single unit and use them.

Syntax:

```
import { combineReducers } from "redux";
const rootReducer = combineReducers({
   books: BooksReducer,
   activeBook: ActiveBook
});
```

43. What is context API?

Context API is used to pass global variables anywhere in the code. It helps when there is a need for sharing state between a lot of nested components. It is light in weight and easier to use, to create a context just need to call React.createContext(). It eliminates the need to install other dependencies or third-party libraries like redux for state management. It has two properties Provider and Consumer.

44. Explain provider and consumer in ContextAPI?

A provider is used to provide context to the whole application whereas a consumer consume the context provided by nearest provider. In other words The Provider acts as a parent it passes the state to its children whereas the Consumer uses the state that has been passed.

45. Explain CORS in React?

In ReactJS, Cross-Origin Resource Sharing (CORS) refers to the method that allows you to make requests to the server deployed at a different domain. As a reference, if the frontend and backend are at two different domains, we need CORS there.
We can setup CORS evironment in frontend using two methods:

- axios
- fetch

46. What is axios and how to use it in React?

Axios, which is a popular library is mainly used to send asynchronous HTTP requests to REST endpoints. This library is very useful to perform CRUD operations.

- This popular library is used to communicate with the backend. Axios supports the Promise API, native to JS ES6.
- Using Axios we make API requests in our application. Once the request is made we get the data in Return, and then we use this data in our project.

To install aixos package in react use the following command.
npm i axios

47. Write a program to create a counter with increment and decrement?

Javascript

```
import React, { useState } from "react";
const App = () => {
  const [counter, setCounter] = useState(0)
  const handleClick1 = () => {
    setCounter(counter + 1)
  }
  const handleClick2 = () => {
    setCounter(counter - 1)
  }
  return (
    <div>
      <div>
        {counter}
      </div>
      <div className="buttons">
        <button onClick={handleClick1}>
          Increment
        </button>
        <button onClick={handleClick2}>
          Decrement
        </button>
      </div>
    </div>
  )
}
export default App
```

48. Explain why and how to update state of components using callback?

It is advised to use a callback-based approach to update the state using setState because it solves lots of bugs upfront that may occur in the future.We can use the following syntax to update state using callback

```
this.setState(st => {
   return(
      st.stateName1 = state1UpdatedValue,
      st.stateName2 = state2UpdatedValue
   )
})
```

49. What is React-Material UI?

React Material UI is an open-source React component library, offering prebuilt components for creating React applications. Developed by Google in 2014, it's compatible with JavaScript frameworks like Angular.js and Vue.js. Renowned for its quality designs and easy customization, it's favored by developers for rapid development.

50. What is flux architecture in redux?

Flux architecture in Redux is a design pattern used for managing application state in a unidirectional data flow. In this architecture, actions are dispatched to modify the store, which holds the entire application state. The store sends the updated state to the view (UI), and the cycle repeats when new actions are triggered. Redux follows this structure to ensure a predictable and maintainable state management system for large applications.