**Q1: What is React?**

**Ans.** React is an open-source JavaScript library used to develop complex and interactive web and mobile UI. It is used to build web browser apps, create dynamic libraries, and build UIs that are perfectly equipped to render large datasets. React follows the component-based approach to create reusable and complex UIs from small and isolated pieces of code called components.

*Learn about JavaScript, read our post –*[*What is JavaScript*](https://www.shiksha.com/online-courses/what-is-javascript-st619-tg305)*?*

***Explore Popular Courses on Shiksha Online:***

|  |  |
| --- | --- |
| [Popular Programming Courses](https://www.shiksha.com/online-courses/programming-courses-certification-training-st619) | [Popular Java Courses](https://www.shiksha.com/online-courses/java-courses-certification-training-st619-tg17) |
| [Popular JavaScript Courses](https://www.shiksha.com/online-courses/javascript-courses-certification-training-st619-tg305) | [Popular Python Courses](https://www.shiksha.com/online-courses/python-courses-certification-training-st619-tg21) |

**Q2. What are the features of React?**

**Ans.** The features of React are:

* **High Performance:** Instead of updating all the components at once, React updates only the components that have changed. Web applications become much faster as a result.
* **One-way data-binding:**One-way data binding in React keeps everything quick and modular. Due to the unidirectional nature of the data flow, embedding child components inside parent components is a common practice when creating React apps.
* **JSX:**JSX is a JavaScript syntax extension. It is utilized with React to specify how the user interface ought to seem. We can write HTML structures in the same JavaScript code file using JSX.
* **Components:**Any React application is built from components, and a single app often comprises of several components. It divides the user interface into separate, reusable components that may be handled independently.

Explore more- [Free JavaScript Courses Online](https://www.shiksha.com/online-courses/free-javascript-courses-certification-training-st619-tg305)

**Q3. What are the advantages of React?**

**Ans.** The advantages of React are:

* **Better Performance:**React employs a virtual DOM, which speeds up online apps. Like traditional web applications, the virtual DOM compares its current state to its prior state and updates only those parts of the real DOM whose states have changed.
* **Easily Develop Dynamic Applications:** React offers greater capability with less writing, whereas JavaScript apps tend to have code that becomes quite complex very rapidly. As a result, React makes it simpler to create dynamic online applications.
* **Reusable Component:** Any React application is built from components, and a single app often comprises of several components. The development time of an application is significantly shortened by these components, each of which has its own logic and controls and may be reused throughout the application.
* **Dedicated Tools for Effortless Debugging:** We can use a Chrome extension that Facebook has provided to debug React applications. This speeds up and simplifies the debugging process for React to web applications.

Must read: [Angular vs React: Understand the Difference](https://www.shiksha.com/online-courses/articles/angular-vs-react-understand-the-difference/)

**Q. 4 Explain the process of creating a React app?**

**Ans.** The following steps help in creating a React app:

1. First, you would install NodeJS on your computer since node package manager (npm) is required for installing the React library. Npm contains several JavaScript libraries, including React.
2. Install the ‘**create-react-app’** package through the command prompt/ terminal.
3. Install any suitable text editor.

**Q5. What are the disadvantages of React?**

**Ans.** The disadvantages of React are:

* While using React, you need more code for development compared to its counterparts.
* It is just a library and not a full-fledged framework.
* Not using an isomorphic approach to exploit applications can lead to SEO indexing problems.
* React’s large size library takes time to understand.
* ReactJS uses JSX, a syntax extension that allows HTML with JavaScript mixed together. For new developers, it may result in complexity in the learning curve.

**Q6. What is JSX?**

**Ans.** JSX is a syntax extension to JavaScript that comes with the full power of JavaScript. JSX stands for JavaScript XML. It enables you to write HTML in React.

JSX produces React elements and enables developers to put any JavaScript expressions within braces inside JSX. Each React element is a JavaScript object that you can store in a variable or pass around in your program. Here is an example:

render(){

    return(

<div>

<h1> Hello World </h1>

         </div>

    );

}

Learn more:[Free Full Stack Development Courses Online](https://www.shiksha.com/online-courses/free-fullstack-development-courses-certification-training-st644-tg1410)

**Q7. Why can’t web browsers read JSX directly?**

**Ans.** Web browsers cannot read JSX directly because JSX is not a regular JavaScript (JS) object and web browsers are built for reading only regular JS objects. To read a JSX file, the file must be converted into a regular JavaScript object.

**Q8. What is the difference between Real DOM and Virtual DOM?**

**Ans.** The differences between the Real DOM and the Virtual DOM are:

|  |  |
| --- | --- |
| **Real DOM** | **Virtual  DOM** |
| 1. Real DOM takes all the HTML elements and wraps them in a tree-like structured object. | 1. It has the same features as the Real DOM object but it can’t write and display things on the screen like Real DOM. |
| 2. Updates slow. | 2. Updates faster. |
| 3. Manipulation is slow. | 3. Manipulation is faster because nothing gets drawn onscreen. |
| 4. Memory wastage is more. | 4. No memory wastage. |
| 5. It represents the document as nodes and objects. | 5. A virtual DOM object is like a lightweight copy of Real DOM. |

**Q9. Explain Virtual DOM.**

**Ans.** DOM stands for “Document Object Model”. The virtual DOM (VDOM) is a virtual representation or a copy of the real DOM. It is a programming concept where an ideal, or virtual, representation of a UI is kept in memory and synced with the real DOM. Thus, whenever the state of the application changes, the virtual DOM gets updated instead of the real DOM.

*Also Read:*[*Top AngularJS Interview Questions and Answers*](https://www.shiksha.com/online-courses/articles/angularjs-interview-questions-answers/)

**Q10. What is React.createClass?**

**Ans.** Since JavaScript didn’t have classes, React has its own class system. React.createClass allows you to generate component classes. Thus, the component class uses a class system implemented by React.

With ES6, React enables you to implement component classes that use ES6 JavaScript classes. While the way of doing is different, the end result would be the same.

**Q11. Explain the lifecycle methods of ReactJS?**

**Ans.** The lifestyle methods of ReactJS are:

1. componentWillMount: This method is used for App-level configuration in the root component. It is executed before rendering
2. componentDidMount: This method is executed after first rendering only on the client-side. In this, all AJAX requests, DOM or state updates, and more take place.
3. componentWillReceiveProps is invoked when the props are updated and before the next render is called.
4. shouldComponentUpdate: It should return a true or false value and will determine if the component will be updated or not. This is set to true by default. If you are sure that a component doesn’t need to render after state or props are updated, then you can return a false value.
5. componentWillUpdate: It is invoked just before re-rendering the component.
6. componentDidUpdate: It is invoked just after rendering and is used to update the DOM in response to prop or state changes.
7. componentWillUnmount: It is called after the component is unmounted from the DOM.

**Q12. What is the significance of Keys in React?**

**Ans.** Keys are used in React to identify unique VDOM Elements with their corresponding data driving the UI and help React identify which items have changed, are added, or are removed. Keys should be a unique number or string.

*Must Read:*[*Top JavaScript Interview Questions and Answers*](https://www.shiksha.com/online-courses/articles/top-javascript-interview-questions-and-answers/)

**Q13. What are refs in React?**

**Ans.** Refs are a function in React that acts as a way to reference a DOM element or a class component from within a parent component. It enables you to read and modify the value of a child component, without making use of props. Refs can be helpful when using third party libraries as well in animations.

**Q14. What are Higher Order Components (HOC)?**

**Ans.** Higher Order Components are an advanced way of reusing the component logic. They are a pattern derived from React’s compositional nature and are not considered a part of the React API. HOC are functions that take a component and return a new component. HOCs are pure components as they can accept any dynamically provided child component, however, they will not copy or modify any behavior from their input components.

**Q 15. Explain an event in React?**

An event is a user- or system-initiated activity, like clicking a mouse or pushing a key.

* Instead of utilizing lowercase in HTML, React events’ names are written in camelCase.
* In contrast to HTML, which uses a string as the event handler, JSX uses a function.

**Q16. What are React Hooks?**

**Ans.** Hooks enable you to use state and other React features without writing a class. These in-built functions developers to use state and lifecycle methods inside functional components. They do not work inside classes. By using Hooks, you can completely avoid the use of lifecycle methods, like componentDidMount, componentDidUpdate, and componentWillUnmount.

*Also Read:*[*Top Node.js Interview Questions*](https://www.shiksha.com/online-courses/articles/frequently-asked-node-js-interview-questions-answers/)

**React Components Interview Questions**

**Q17. What is Props?**

**Ans.** Short for properties, Props are arguments passed into React components. They are like function arguments in JavaScript. Props accept arbitrary inputs and return React elements detailing what should be displayed on the screen. Props maintain a unidirectional data flow and are only passed down from the parent component to the child components throughout the application. A child component can never send a prop back to the parent component. They are immutable.

**Q18. What are States in React?**

**Ans.** States are the source of data. They are objects which determine components rendering and behavior. While props are immutable, States are mutable and can change over time. States create dynamic and interactive components and are accessed by using this.state(). While using the State component, you will set a component’s default state, access the current state, and update the state.

*Must Read:*[*Top Django Interview Questions and Answers*](https://www.shiksha.com/online-courses/articles/django-interview-questions-and-answers/)

**Q19. What is the purpose of render() in React?**

**Ans.** The render() function in React displays the specified HTML code inside the specified HTML element. It is mandatory for each React component to have a render(). This function takes 2 arguments, namely HTML code and an HTML element, and returns a single element which is the representation of the native DOM component.

If more than one HTML element is to be rendered, then they must be grouped inside one enclosing tag, such as <form>, <group>, and <div>.

**Q20. What are Pure Components in React?**

**Ans.** Pure components are those components that render the same output for the same state and props. They are the simplest and fastest components that can be written. They can replace any component which only has a render(). Thus, a function is said to be pure if:

* Its return value is determined only by its input values
* Its return value is always the same for the same input values

**Q21. What is an Event in React?**

**Ans.** React are actions based on user events. An event could be triggered by the user action or a system-generated event. React has the same events as HTML, such as click, change, and mouseover.

React events are written in camelCase syntax and are passed as functions instead of strings. Event arguments have a specific set of properties and behavior.

**Q22. Explain the different phases of the ReactJS component lifecycle?**

**Ans.** There are four different phases of the ReactJS components lifecycle: Initialization, Mounting, Updating, and Unmounting.

1. Initialization: This is the first stage in the ReactJS lifecycle. In this stage, the component is constructed with the provided properties and a default state.
2. Mounting: It is the phase that involves putting elements into the DOM. It includes componentWillMount, and componentDidMount lifecycle methods.
3. Updating: This stage updates the stage of a component and repaints the application. This phase includes, shouldComponentUpdate, componentWillUpdate lifecycle methods.
4. Unmounting: It is the final stage, in which the component is removed from the page.

*Check out the popular*[*Frontend Development Courses*](https://www.shiksha.com/online-courses/frontend-development-courses-certification-training-st644-tg417)

**Q23. What is the difference between stateful and stateless components?**

**Ans.** The difference between Stateful and Stateless Components are:

|  |  |
| --- | --- |
| **Stateful (Class) Component** | **Stateless (Functional) Component** |
| 1. Stateful Component stores information about the component’s state change in memory. | 1. It calculates the internal state of the components. |
| 2. It has complex UI Logic. | 2. It accepts properties(props) in function and return HTML (JSX). |
| 3. It contains the data related to past, current, and possible future changes in state. | 3. It does not contain the data related to the past, current, and possible future state changes. |
| 4. It must have the render() method returning HTML. | 4. There is no render method used in the Stateless component. |
| 5. Have the authority to change the state. | 5. Do not have the authority to change the state. |

**Q24. What do you know about the controlled and uncontrolled components?**

**Ans.** The differences between controlled and uncontrolled components are:

|  |  |
| --- | --- |
| **Controlled Components** | **Uncontrolled Components** |
| They do not maintain their internal state. | They maintain their internal state. |
| Data is controlled by the parent component. | Data is controlled by the DOM itself. |
| They accept current value as a prop. | These components use ref to get their current values. |
| They allow validation control. | Uncontrolled components do not allow validation control. |
| Controlled components have good control over the form elements and data. | They have limited control over the form elements and data. |

**Q25. How to create an event in React?**

**Ans.** An event can be created in React using the following:

class Display extends React.Component({

    show(msgEvent) {

    },

    render() {

        return (

          <div onClick={this.show}>Click Me</div>

        );

    }

});

*Explore the best [Fullstack Development Courses](https://www.shiksha.com/online-courses/fullstack-development-courses-certification-training-st644-tg1410" \t "_blank)*

**Q26. Explain synthetic events in React.**

**Ans.** A synthetic event is a cross-browser wrapper around the browser’s native event. It combines the behavior of different browser’s native events into one API, including stopPropagation() and preventDefault().

Browsers can recognize different types of events like MouseEvent, KeyboardEvent, etc., and will have varying levels of functionality for each of them. However, in React, there is only the SyntheticEvent object, with a common functionality that creates consistency across browsers.

**e is a Synthetic event in the below example:**

function ActionLink() {

    function handleClick(e) {

        e.preventDefault();

        console.log(‘You had clicked a Link.’);

    }

    return (

        <a href=”#” onClick={handleClick}>

              Click\_Me

        </a>

    );

}

**Q27. What is the arrow function in React?**

**Ans.** An arrow function is a type of function syntax in JavaScript. Arrow functions don’t redefine the value of ‘this’ within their function body compared to the other method that uses a bind call inside of a constructor. The scope of ‘this’ is global and not limited to any calling function. This makes it easier to predict their behavior when passed as callbacks and prevents bugs caused by the use of ‘this’ within callbacks.

**Example:**

//General way

render() {

    return(

        <MyInput onChange={this.handleChange.bind(this) } />

    );

}

//With Arrow Function

render() {

    return(

        <MyInput onChange={ (e) => this.handleOnChange(e) } />

    );

}

**Q. 28 Describe the component lifecycle methods.**

* **getInitialState():** This is carried out before the component is created.
* **componentDidMount():** is carried out when the component is rendered and added to the DOM.
* **shouldComponentUpdate():** is called whenever a component notices changes to the DOM and, depending on specific circumstances, returns a result of “true” or “false.”
* **componentDidUpdate():** is called shortly following rendering.
* **componentWillUnmount():** is called just before a component is permanently unmounted and destroyed.

**React Redux Interview Questions and Answers**

**Q29. What is Redux?**

**Ans.** Redux is a predictable state container for JavaScript applications. It is a state management tool that enables developers to:

* write applications that behave in a consistent manner
* run in a variety of environments, and
* can be tested easily

It is commonly used with React, but also works with other JavaScript libraries and frameworks.

**Q30. What are the three fundamental principles of Redux?**

**Ans.** The three fundamental principles of Redux are:

1. Single source of truth. A single store stores the state of your application as an object tree. The single state tree will make it easier to make changes and inspect the application.
2. State is read-only. It means that the only way to change the state is to trigger an action.
3. A pure function is used to make changes. You need pure functions to specify how actions transform the state tree

**Q31. What are the components of Redux?**

**Ans.** Redux has 4 components, namely:

1. Action: It is an object that tells what happened. Actions send data from the application to the store and are the only source of information for the store.
2. Reducer: Reducers specify how the state of the application will change changes in response to actions sent to the store. While Actions specify what happened, Reducers determine how the state will change in response to an action.
3. Store: It is where the State or Object tree of the entire application is saved. The Store brings Action and Reducer together
4. View: It shows the data provided by the Store.

**Q32. What is the use of Reducer?**

**Ans.**By triggering an action, a reducer describes how the state of the application changes. Reducers take the previous state and action and then return a new state.

In simple terms, if you repeat the same test with the same input as arguments, you can always expect the same output.

Explore more: [Top 10 Basic TypeScript Interview Questions](https://www.shiksha.com/online-courses/articles/typescript-interview-questions/)

**Q33. How are Actions defined in Redux?**

**Ans.** Actions are plain JavaScript objects that have a type field. There must be a type property on Actions that indicates the type of Action. Defining them as String constants allows you to add more properties to them. We can create actions using the functions called Action Creators.  We can create actions using the functions called Action Creators.

**Example:**

function addTodo(text) {

       return {

                type: ADD\_TODO,

                 text

    }

}

**Q34. What is the difference between React and Flux?**

**Ans.** The differences between React and Flux are:

|  |  |
| --- | --- |
| **Redux** | **Flux** |
| Redux is an open-source JavaScript library that can be used with any library or framework. It is used to manage the application State. | It is not a framework or a library. It is a kind of architecture that complements React and follows the concept of Unidirectional Data Flow. |
| It has only a single Store. | Flux can have multiple Stores. |
| Immutable store state. | Mutable store state. |
| Store and change logic are separate. | The store contains State and the Change logic. |
| Lacks a dispatcher. | It has a singleton Dispatcher, and all actions pass through that Dispatcher. |
| Integration with: React, Jumpsuit, and react.js boilerplate | Integration with: React, TuxedoJS, and Fluxxor |

**Q35. What are the advantages of React Redux?**

**Ans.** Some of the major advantages of React Redux are:

1. Redux makes the state predictable.
2. It makes the code easier to maintain.
3. Easy to debug an application.
4. Redux implements many performance optimizations internally.
5. Easy to test Redux apps.
6. Enables user to persist some of the app’s state to local storage and restore it after a refresh.
7. Server-side rendering is possible.

**Q36. How would you access the Redux store outside a component?**

**Ans.** To access the Redux store outside a component, we will need to export the Store from the module where it was created with createStore() method while ensuring that it will not pollute the global window space.

**Below is the code:**

store = createStore(myReducer)

export default store

**React Native Interview Questions and Answers**

**Q37. What is the difference between Hot Reloading and Live Reloading in React Native?**

**Ans.** Live reloading reloads or refreshes the entire app when a file changes.

Hot reloading only refreshes the files that were changed without losing the application’s state. It just displays the code changes without restarting the app.

*Also explore:*

* Paid and [free online courses](https://www.shiksha.com/online-courses/coursera-courses-certification-training-v475)by Coursera
* Popular online [Udemy courses](https://www.shiksha.com/online-courses/udemy-courses-certification-training-v21)
* Top online [edX courses](https://www.shiksha.com/online-courses/edx-courses-certification-training-v117)

**Q38. What is the difference between ReactJS and React Native?**

**Ans.** The differences between React JS and React Native are:

|  |  |
| --- | --- |
| **ReactJS** | **React Native** |
| 1. It focuses on Web Development. | 1. It is an extension of React and focuses on Mobile User Interfaces (Android and iOS). |
| 2. ReactJS is a JavaScript library that enables developers to create a high performing UI Layer. | 2. It is an entire framework for building cross-platform apps. |
| 3. ReactJS uses Virtual DOM to render browser code. | 3.React Native uses Native APIs to render components in mobile apps. |
| 4. It uses HTML to render apps. | 4. It doesn’t use HTML to render the app. |

**Q39. Explain the use of Flexbox in React Native?**

**Ans.** Flexbox is a layout implementation in React Native. It offers an efficient way to create UIs and control positioning in React Native. It provides 3 properties, namely flexDirection, justifyContent, and alignItems to achieve the desired layout.

**Q40. Explain the use of the ScrollView component.**

**Ans.** The ScrollView component is a generic scrolling container. It can contain multiple components and views. However, the scrollable items need not be homogeneous. It enables you to scroll both vertically and horizontally by setting the horizontal property.

**Q41. What do you mean by Component Driven Development (CDD)?**

**Ans.** Component Driven Development (CDD) is a methodology that holds the developing process which is revolving around the components. It is a process that creates UIs from the bottom up by starting at the components level and ending at the pages or screen level. It is a concept that applies to pattern libraries, design systems, and modern JavaScript frameworks.

**Q42. How to create a stackNavigator in React Native?**

**Ans.** Below is the code to create a stackNavigator in React Native

const AppNavigator = createStackNavigator({  
Home: {  
Screen: HomeScreen,  
},  
});

**React Router Interview Questions**

**Q43. What is React Router?**

**Ans.** React Router is a routing library built on top of React, which is used to create routes in a React application. It enables multiple views in a single application by defining multiple routes in the React application. React Router allows the navigation among views of various components in a React Application. On the browser, it provides the data that will appear on the web page via the synchronous URL. It maintains the standard structure and behavior of the application.

**Q44. What is the difference between React routing and conventional routing?**

**Ans.** The differences between React routing and conventional routing are:

|  |  |
| --- | --- |
| **React Routing** | **Conventional routing** |
| Single HTML page | Each view corresponds to a new HTML file |
| The user is deceived thinking he is navigating across different pages | The user navigates across different pages for each view |
| Only the history attribute changes | An HTTP request is sent to a server and a corresponding HTML page is received |
| The page does not refresh since it is a single file | The page refreshes every time user navigates |

**Q45. What is the use of a React Router?**

**Ans.** React Router helps maintain a consistent structure and behavior. It is also used for complex navigational requirements in React applications.

**React Styling Interview Questions**

**Q46. What is CSS Module styling in React?**

**Ans.** CSS Module is a CSS file but with a key difference. Every class name and animation inside a CSS module is by default scoped locally to the component that is importing it. It is available only for the component which imports it and cannot be applied to any other components without your permission. It means that one can use virtually any valid name for the classes, without worrying about conflicts with other class names in the application. You can create a CSS Module file with the .module.css extension.

**Q47. How many ways can we style the React Component?**

**Ans.** There are four ways to style React Component:

1. Inline Styling
2. CSS Module
3. Styled Components
4. CSS Stylesheet

**Q48. What are Styled Components?**

**Ans.** Styled Components is an open-source CSS-in-JS library that bridges the gap between components and styling. It is the successor of CSS Modules. Styled components are a way to write CSS that’s scoped to a single component and cannot leak to any other element in the page. It allows users to write plain CSS in their components without worrying about class name collisions.

The Styled Components provides:

1. Automatic critical CSS
2. Easier deletion of CSS
3. No class name bugs
4. Dynamic and intuitive styling

**Q. 49 Describe how React uses CSS modules.**

* The **.module.css**extension is used to build the CSS module file.
* There are no naming conflicts when styling the components because the CSS in a module file is only accessible by the component that imported it.

Also read- [How to choose the best Full Stack Development course?](https://www.shiksha.com/online-courses/articles/how-to-choose-the-best-full-stack-development-course/)

**Q. 50 What is the difference between State and Props?**

|  |  |  |
| --- | --- | --- |
| Definition | Internal data specific to a component | External data passed to a component |
| Mutability | Can be changed (mutable) | Immutable (read-only) |
| Ownership | Owned and managed within the component | Passed from a parent component |
| Access | Accessed using ***this.state*** | Accessed using ***this.props*** |
| Modification | Modified using***setState()*** method | It cannot be modified within a component |
| Updates | Triggers component re-rendering | It does not trigger re-rendering |
| Source of Data | Component-specific | Received from a parent component |
| Scope | Limited to the component | It can be passed down multiple levels |
| Data Flow | Unidirectional (top-down) | One-way flow from parent to child |
| Changes Notification | Component updates and re-renders | No notification of changes |