

Q1. What is React?

Ans: React is a front-end JavaScript library developed by Facebook in 2011.

- It follows the component-based approach which helps in building reusable UI components.
- It is used for developing complex and interactive web and mobile UI.
- Even though it was open-sourced only in 2015, it has one of the largest communities supporting it.

Q2. What are the features of React?

Ans: 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.

Q3. What are the limitations of React?

Ans: React is just a library, not a full-blown framework

- Its library is very large and takes time to understand
- It can be a little difficult for novice programmers to understand
- Coding gets complex as it uses inline templating and JSX

Q4. What is JSX?

Ans: The abbreviation for JavaScript XML is JSX. This is a specific kind of file used by React that combines HTML-like template syntax with the expressiveness of JavaScript. The HTML file is quite simple to read as a result. This file strengthens and speeds up the performance of apps.

Q5. What do you understand by Virtual DOM? Explain its works.

Ans: The virtual DOM is a compact JavaScript object that is only a duplicate of the actual DOM at first. It is a node tree that lists the components, along with their properties, attributes, and content as objects. The React render method arranges the React components into a node tree. Then, in reaction to changes in the data model brought on by various user or system events, it updates this tree.

Q6. Why can't browsers read JSX?

Ans: Only JavaScript objects can be read by browsers, however, JSX is not a typical JavaScript object. Therefore, using JSX transformers like Babel, we must first convert JSX files into JavaScript objects before passing them to the browser in order to allow a browser to read JSX

Q7. What is the purpose of render() in React?

Ans: Every React component is required to have a render() method. It provides one React element as a result, which represents the native DOM component. If more than one HTML element is to be rendered, they must be contained within a single enclosing tag, such as a "form" tag, "group" tag, "div" tag, etc. This function must be maintained pure, which means it must always return the same outcome when called.

Q8. What is a state in React and how is it used?

Ans: The core of React components is stated. States must be kept as straightforward as possible because they are the data source. In essence, states are the objects that control how components render and behave. They produce dynamic and interactive components and, unlike the props, are malleable. They are accessed via `this.state()`.

Q9. What are synthetic events in React?

Ans: The objects that operate as a cross-browser wrapper around the native event of the browser are called synthetic events. One API combines the functionality of several browsers. This is carried out to ensure that the events display uniform attributes across various browsers

Q10. What are Pure Components?

Ans: The simplest and quickest components that may be written are pure components. Any component with merely a render can be swapped out by them (). The efficiency of the program and the simplicity of the code are improved by these components.

Q11. “In React, everything is a component.” Explain.

Ans: The foundation of a React application's user interface is components. These parts divide the whole user interface into manageable, reusable chunks. Then, without altering the rest of the UI, it makes each of these components independent of the others.

Q12. What are Props?

Ans: In React, Properties are referred to as props. They must be maintained pure, or immutable, as they are read-only components. Throughout the program, they are always transferred from the parent component to the child component. Never can a child component return a prop to the parent component. This aids in preserving the one-way data flow and is typically used to present dynamically created data.

Q13. What is the arrow function in React? How is it used?

Ans: Arrow functions are more of a brief syntax for writing the function expression. They are also called ‘fat arrow’ (`=>`) functions. These functions allow binding the context of the components properly since in ES6 auto binding is not available by default. Arrow functions are most useful while working with higher-order functions.

Q14. What are the different phases of React component’s lifecycle?

Ans: There are three different phases of React component’s lifecycle:

- i. Initial Rendering Phase: This is the phase when the component is about to start its life journey and make its way to the DOM.
- ii. Updating Phase: Once the component gets added to the DOM, it can potentially update and re-render only when a prop or state change occurs. That happens only in this phase.
- iii. Unmounting Phase: This is the final phase of a component’s life cycle in which the component is destroyed and removed from the DOM.
- iv.

Q15. What do you understand by refs in React?

Ans: In React, the term "References" is abbreviated as "Refs." It is an attribute that aids in storing a reference to a specific React element or component that will be provided by the render configuration function of the component. It is used to provide references to specific elements or components that render returned (). They are useful when we need to add methods to the components or DOM measures.