

## **What are Higher-Order components**

A higher-order component (HOC) is a function that takes a component and returns a new component. Basically, it's a pattern that is derived from React's compositional nature. We call them as "pure" components because they can accept any dynamically provided child component but they won't modify or copy any behavior from their input components.

## **What do you mean by Asynchronous API**

All APIs of Node.js library are asynchronous that is non-blocking. It essentially means a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call.

## **What is Reconciliation**

When a component's props or state change, React decides whether an actual DOM update is necessary by comparing the newly returned element with the previously rendered one. When they are not equal, React will update the DOM. This process is called reconciliation.

## **How Node prevents blocking code**

By providing a callback function. The callback function gets called whenever a corresponding event is triggered.

## **What are React Hooks**

Hooks are a new addition in React 16.8. They let you use state and other React features without writing a class. With Hooks, you can extract stateful logic from a component so it can be tested independently and reused. Hooks allow you to reuse stateful logic without changing your component hierarchy. This makes it easy to share Hooks among many components or with the community.

## **What is JSX**

JSX is a syntax extension to JavaScript and comes with the full power of JavaScript. JSX produces React elements. You can embed any JavaScript expression in JSX by wrapping it in curly braces. After compilation, JSX expressions become regular JavaScript objects. This means that you can use JSX inside of if statements and for loops, assign it to variables, accept it as arguments, and return it from functions:

## **What is prop drilling**

When building a React application, there is often the need for a deeply nested component to use data provided by another component that is much higher in the hierarchy. The simplest approach is to simply pass a prop from each component to the next in the hierarchy from the source component to the deeply nested component. This is called prop drilling.

## **What is Key**

A key is a special string attribute you need to include when creating lists of elements. Keys help React identify which items have changed, are added, or are removed.