Introduction to React Hooks | Complete React Course in Hindi #38

In this article, we will be having a short introduction to React Hooks. In addition to that, we will be understanding some of the most widely used Hooks in React Application. This is so because in the upcoming videos we will be changing the class-based component into function-based components, as they are quite easy when compared with class-based components.

What are React Hooks?

- React Hooks allows us to use the features of class-based components in function-based Components.
- It even allows us to use "state" and other React features without writing a class. For example: Earlier, we were using "this.state" whenever we needed to access the state but now we can use the "UseState" hook.
- Hooks are the functions which "hook into" react state and lifecycle features from the function components.

Commonly used React Hooks

There are several react hooks to look upon. You can even create a custom, React Hook. But in this tutorial, we are covering some of the most commonly used React Hooks:

1. **UseState:** It is used to update the state and to set the initial value of the state. The 'useState' is similar to 'this.setState' in class. The useState returns a pair where the first element is the current state value/initial value, and the second one is a function that allows us to update it. For example: If we create a text variable and we want to make it a part of the state, then we can return two elements (like 'text' and 'set text') from the state hook.
2. **Use Effect:** It is used to perform the side effect. For example: If we want to perform a certain task when the content of our application is updated, then we can take the help of the UseEffect hook. In other words, effect Hooks are similar to componentDidMount(), componentDidUpdate() and componentWillUnmount() lifecycle methods.
3. **Use Context:** This hook allows us to use the context API. Context API allows us to share data within its component tree without passing through props. In short, it helps in removing prop drilling from the application. Prop drilling is a situation when the same data is being sent at every level due to requirements. We will be discussing the Context hook in detail later in this React Course.
4. **UseRef:** It returns a mutable reference object, which has a ".current" property. It can be used to point an element inside a DOM. In most simple words, it is a holder which can store an element of the DOM inside its ".current" property.