We need a component that takes an input and the input value is converted from Fahrenheit to Celsius or vice versa, the formulas are:

Formula for converting Fahrenheit to Celsius= (°F - 32) * 5/9

Formula for converting Celsius to Fahrenheit = (°C * 9/5) + 32

Later we may have another input and after the user enters a value into the input field, we need to multiply the entered value by 2 and show the value in h1 element with pink color.

Which pattern is suitable for this task? Should we create a component each time that has a state property for each input field to keep track of each input field and change its values? Imagine we need to use the Fahrenheit conversion in multiple places around our code.

Answer: Render props pattern. It is a pattern where we set a property as a callback function that returns JSX code. Imagine we pass a property called renderProps which is a callback function to another component and then this callback function takes a value back and returns a jsx code. The component from where we set the renderProps property does not implement its rendering logic, it does not have useState but it uses the state of the parent component to get the values. This way we can change the styling as well as the values returned by the parent component depending on our needs and we do not have to create multiple state variables in both of the components. You can use render props:

- Instead of HOC
- To avoid prop drilling
- To customize component's rendering based on specific use cases.
- Managing state in a centralize way.