REACT MEMO

➡ React.Memo is a higher order component which takes other component as it's parameter and apply optimization through props comparison

➡ Let's understand this concept with an example

Assume there's a Parent component & it has 3 Child components

➡Now, whenever the Parent component re-renders due to the state change then all it's Child components will re-render. This is how React rendering works

➡ What if the state change in the Parent component doesn't affect its Child components ? The Child components are still getting re-rendered, right

➡ The Child components should only re-render if any of their props changes or there's some state change in the Child component itself

➡ Here comes React.Memo to the rescue. It will compare the props passed to the Child components between re-renders

➡ It will only re-render the Child component if the props passed are different. If the props had not changed between re-renders then the Child component would not re-render. How cool is that ;)

Syntax

➡ Whenever you are exporting the Component just wrap it with React Memo. Let's assume you have to optimize the Profile Component then you should write

Example ➡ React.Memo(Profile)

Important Note ⬇⬇

React.Memo does shallow comparison of the props so it would not work with objects and functions

If you find this post helpful then please do share this post with your connections ;))