

You are given a piece of code in `memo1question` folder.  
How to prevent the Child component from being re-rendered when the button in the App component is clicked?

In React, when you create a function inside a functional component, it's recreated every time the component renders. For example:

```
function Message() {  
  // This function is recreated on every render  
  const handleClick = () => {  
    console.log("Button clicked");  
  };  
  
  return (  
    <>  
      <button onClick={handleClick} />  
    </>  
  );  
}
```

This isn't a problem in most cases, but if you have a lot of child components that receive this function as a prop, they might re-render more often than necessary.

In React when the state of a parent component changes, it triggers a re-render of the parent component and its children. This is because React follows a unidirectional data flow, where changes in the state or props of a parent component can affect the rendering of its child components.

**So what is the main problem:** Every time a component re-renders, its functions get recreated.

**Solution: useCallback** hook and **memo**. **useCallback** hook means that do not recreate me(the function) unless my dependencies change.

**memo** is a HOC to prevent a component from re-rendering unless its props have changed. memo will cause React to skip rendering a component if its props have not changed

**useCallback** returns a memoized function.