module-12_2

[module-12: look behind the scenes of react & optimization techniques]
Module-12_2: useMemo hook

a component is re-rendered only when props/state/context is changed.

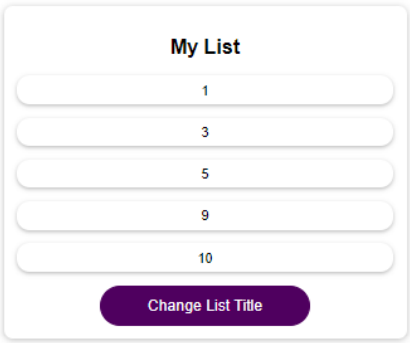
re-evaluating component !== re-rendering the DOM

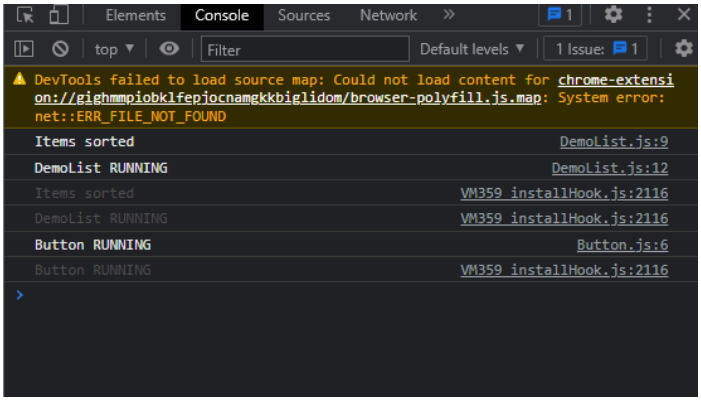
array, objects, function are non-primptive value in javascript.
string, number, boolean are primptive value in javascript.

closure??

React.memo denies the re-evaluation of a component if there's no change in the component via props.
useCallback hook saves a function in a component, which optimizes the code, it also takes a dependancy.
React.memo & useCallback hook increases the optimization of our code.

Here, useMemo hook is used in App.js





useMemo hook holds the value of a variable. If dependency changes, useMemo hook re-executes & save the value again for that variable.

So, the basic difference between useCallback & useMemo is, useCallback holds a function and useMemo holds a value.