**HOOKS [src : youtube/codeevolution]**

Hooks are just functions.

Hooks must be called only at the top level. Not inside loops/nested functions.

Hooks should be used only inside React functional components.

1.usestate hook

Why ?   
You can use **state** in functional components also.

How?

import React, { useState } from 'react'

import ReactDOM from 'react-dom'

function App() {

    const [count, setCount] = useState(0);

    let btnStyle = {

        width:'100px',

        height:'25px'

    }

    return (

        <div>

            <p>You have clicked {count} times</p>

            <button onClick={ () => setCount(count+1) }   
 style={btnStyle}>  
 Click</button>

        </div>

    )

}

ReactDOM.render(<App />, document.getElementById('root'));

2.usestate hook with previous state

What ? how to set state based on the previous state value

The setstate(count+1)

Why ? If you want to update state based on the previous state value.

How ?  
  
  
  
The problem : Even though Inc5 is clicked and you want to increment 5 times : the end result will have the count incremented by only 1 – no use even if the ***setCount(count+1)*** code is executed 5 times

function App() {

    const initialCount = 0;

    const [count, setCount] = useState(0);

    let btnStyle = {   width:'100px',  height:'25px'    }

    const Inc5 = ()=>{

        for (let index = 0; index < 5; index++) {

            setCount(count+1)

            // This setCount will be operating

// in the previous 'count' value.

            // 5 times on the same old value

        }

        //Not sure but : After this function is executed :   
 // The value of oldvalue incremented by 1 is displayed.

    }

    return (

        <div>

            <p>You have clicked {count} times</p>

            <button onClick={ () =>setCount(count+1) }         
 style={btnStyle}>  
 Increment</button>

            <button onClick={ () => setCount(count1)}  
        style={btnStyle}>  
 Decrement</button>

            <button onClick={ () => setCount(initialCount) }   
  style={btnStyle}>  
 Reset</button>

            {/\* <button onClick={ () => {setCount(count + 5)} }  style={btnStyle}>Incrementby5</button>

            The above line will work \*/}

            <button onClick={Inc5}    
 style={btnStyle}>  
 Inc5</button>

        </div>

    )

}

ReactDOM.render(<App />, document.getElementById('root'));

The Solution : Should be adopted for the increment and decrement buttons as well

import React, { useState } from 'react'

import ReactDOM from 'react-dom'

function App() {

…

    const Inc5 = ()=>{

        for (let index = 0; index < 5; index++) {

            setCount((prevCount)=>{

                return prevCount+1;

            })

        }

    }  
…

3.usestate hook with object {} as a state variable

Why ? : useState hook setter function does not automatically merge and update the object.

From the below code, when the function setName is called with an object the state object is overwritten

How ?

Problem :

const App = () => {

    const [name,setName] =   
 useState({firstName : '',lastName : ''});

    return (

        <div>

            <input type='text' value={name.firstName}

                onChange={(e)=>{  
 setName({firstName:e.target.value})}  
 }  
 />  
   
 <br/>

            <input type='text' value={name.lastName}

                onChange={(e)=>{  
 setName({lastName:e.target.value})}  
 }  
 />

            <h2>Firstname : {name.firstName}</h2>

            <h2>LastName : {name.lastName}</h2>

            {JSON.stringify(name)}

        </div>

    )

}

Solution:

const App = () => {

    const [name,setName] =   
 useState({firstName : '',lastName : ''});

    return (

        <div>

            <input type='text' value={name.firstName}

                onChange={(e)=>  
 setName({...name,firstName:e.target.value})  
 }   
 />  
 <br/>

            <input type='text' value={name.lastName}

                onChange={(e)=>  
 setName({...name,lastName:e.target.value})  
 }  
 />

            <h2>Firstname : {name.firstName}</h2>

            <h2>LastName : {name.lastName}</h2>

            {JSON.stringify(name)}

        </div>

    )

}