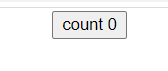
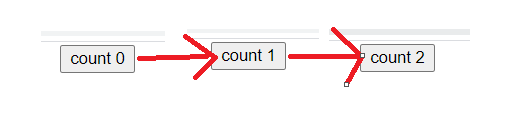
**2.1 How to use useState hook**

const [state, setstate] = useState(initialState)

**Ex. Create an app having button and when you click count on button should increase (using class component and functional component both)**



import React, { Component } from "react";

export class ClassCounter extends Component {

  constructor(props) {

    super(props);

    this.state = {

      count: 0,

    };

  }

  incrementCount = () => {

    this.setState({

      count: this.state.count + 1,

    });

  };

  render() {

    return (

      <div>

        <button onClick={this.incrementCount}>count {this.state.count}</button>

      </div>

    );

  }

}

export default ClassCounter;

import React, { useState } from "react";

const HookCounter = () => {

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

  return (

    <>

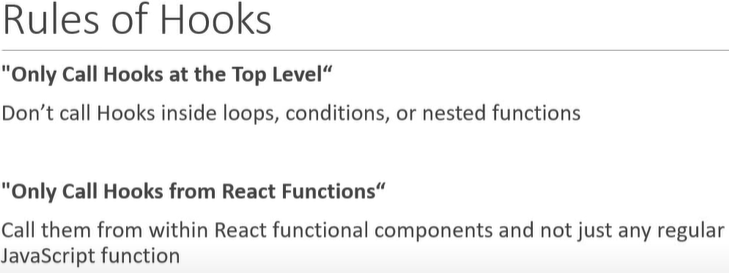
      <button onClick={() => setCount(count + 1)}> count {count}</button>

    </>

  );

};

export default HookCounter;



**2.2 How setState based on previous state value**

This the recommended way to use useState hook

import React, { useState } from "react";

const HookCounter = () => {

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

  return (

    <>

      <button onClick={() => setCount(prevState=>prevState+1)}> count {count}</button>

    </>

  );

};

export default HookCounter;

**2.3 useState hook with object**