|  |  |
| --- | --- |
| TOPIC/TITLE  useState Hook | |
| Keywords/Questions  useState : kook  using only one react function  Destructuring | Notes : |
|  |
| Our code is directly inside our index.js |
|  |
| You must use a hook inside a functional component |
|  |
| https://reactjs.org/docs/hooks-reference.html#usestate |
|  |
| Wrong way doing it |
|  |
| var count = 0;  function increase() {  count++;  ReactDOM.render(  <div className="container">  <h1>{count}</h1>  <button onClick={increase}>+</button>  </div>,  document.getElementById("root")  );  }  ReactDOM.render(  <div className="container">  <h1>{count}</h1>  <button onClick={increase}>+</button>  </div>,  document.getElementById("root")  ); |
| One rule to use useState is using it in a functional component.  In our App.jsx |
|  |
| import React{useState} from "react";  function App() {  const state = React.useState();  to enlighten yout code you import only the react function you need in this example useState |
| import React,{useState} from "react";  function App() {  const state = useState(); |
| Use State if you console it will reneder an array so we have to specify wich data wa want back by selecting its position in the array like this |
| const state = useState(567);  console.log(state[0]); |
| Allows us to destructure a complex structure. Instead of writing state [0] which has no real meaning we can write it like this |
| The complex string in js are objexts and arrays  Complex to read |
| const rgb = [41, 128, 185]  console.log = rgb[0] |
| Better |
| const [red,green,blue] = [41, 128, 185]  so red will be mapped to 41, green to 128 and blue to 185 |
| const [red,green,blue] = [41, 128, 185]  console.log(blue)  and the log will be 185. |
| const [count] = useState(increase(34233));  In this example if I log this it will return 34233instead of an array of values |
|  |
| In use State the return array first item is a value the second is a function |
|  |
| import React, { useState } from "react";  function App() {  const [count, setCount] = useState(0);  function increase() {  setCount(count + 1);  }  function decrease() {  setCount(count - 1);  }  return (  <div className="container">  <h1>{count}</h1>  <button onClick={decrease}>-</button>  <button onClick={increase}>+</button>  </div>  );  }  export default App; |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  | |