Wednesday, May 17, 2023 11:56 AM

We can add the variable to state for two reasons

1. That we want to rerender the page when the value is changed
2. That we want to survive the variable ACROSS RERENDER

Using class component - work with STATE !!!

1. In the constructor = initialize the this.state object
   this.state ={sqr:0, num1:10,num2:9}

2. whenever the value of the state property should be changed
   a. Use this.setState({sqr:9}) // this will change the sqr value but other values will be unchanged
3. Whenever the value of state property should be used
   { this.state.sqr }

Using State in Class	Using State in function

Initialize state
 	
constructor()
{
...
this.state={ sqr:0...}
}
	Initialize state
	
	function XY()
	{
	let [sqr,setsq] = useState(0)
	}

To change the state this.setState({sqr:9})	**To change the state** setsq(9)
To see the value of state {this.state.sqr}	**To see the value of state** {sqr}

____

Communication between Parent component and child component = PROPS
Parent ---------------> Child

Communication between Child Component to Parent component = Function in PROPS
Child --------------------->Parent

Ex1 --- Parent component has 1. Two textfields , <p> for result
Child component has Dropdown for computations (add, sub, power , multiply ,divide)

Data entered in the textfields MUST go from Parent ----------> Child
User can select the computation in Child , process the data and the
Result is sent back from Child --------------> Parent and rendered in the <p> of parent

Let us make the Parent and Child as Functional components.

Alternative for binding the function of a CLASS component with "this " ----

Write functions as Fat Arrow Functions instead of traditional functions

Using bind	Not using bind

constructor()
{
this.textreceiver =
this.textreceiver.bind(this)
}
	No need to bind anything in the constructor

Function Definition
class XX{
....
textreceiver(e1){
… code …
}
}
	Function Definition
	class XX{
	....
	textreceiver=(e1)=>{
	… code …
	}
	}

Conditional Renderring in JSX :

Show a component only if the condition is true !!

{hideflag?"" : <Header></Header> }

{ !hideflag && <Header></Header> }

If hideflag is true then DON'T show header !!!

Ex1 = Write a component that prints a heading

Show the component in App.js only if the checkbox is checked

Using style attribute in JSX

Style attribute In html	Style attribute in JSX
<div style="" >	<div style={} >

<dive style="background-color:'red' " >	Let obj ={backgroundColor:"red" }
	<div style={obj}
	OR
	<div style={{backgroundColor:"red" }}

The css properties have a format that uses Dash - between two words of property-name text-align ,	The css properties are written using camel case textAlign

____

List Renderring

Ex1 = Show a textfield , user enters fruit names . Onblur --add the fruitname to array
Show all the fruitnames in an ordered list