Thursday, January 12, 2023 11:37 AM

Good Morning :)

Ex - C1 discount ---- Function component
    <C2> ---- Class Component
        <C3> ----- function Component
            Apply the discount to amount and display

Pass the discount value from C1 to C3
    C1 will set the props of C2
        C2 will set the props of C3

Write Appl1.js , Use it in index.js instead of App.js
Integrate C 1 with App1

In this example Observe -

1. We used a different App1 and added it to index.js
2. We understood how props are accessed in a class component
   this.props.attribname
3. Props Drilling --- the props are passed from C1
   |
   C2
   |
   C3

___

Spread Operator -----

1. …obj
2. It will spread the properties of the object = extract all properties and laydown as single elements
3. **It is used for DEEP COPY of objects , arrays**
   1. let obj2 = {…ob1} , //OK if obj1 has only primitive properties
   2. If obj1 contains objects as attributes then add a spread for those objects also
      Let obj2= {…obj1 , containedobj : {…obj1.containedobj } }
   3 . let arr2 = […arr1]

___

How to use Inline CSS in react ?

EX 1 ---- In component C3 add inline css for printing the bill

Step1--- put all css attributes in a variable
Step2 --- all css attributes having - should be converted to camelcase
Step3 --- assign the variable to the style attribute in JSX

OR

<tag style={{backgroundColor:'red'}} >

How to use external CSS in react ?

Step 1 = create a file c1.css , add css classes to it.
Step 2 = Open the file where this CSS should be used
    import "./c1.css"

Step3 = <tag className="outline" >
OR
<tag className={clsnm} >

Event Handling in React -------

Event Handling Description -----

Source of the event (button) will generate the event (click) when acted by end user . Once the event is generated the handler is called and executed. The handler may perform validation or DOM manipulation or any logical processing .

Comparative Event Handling --------

Javascript	Jquery	React

<button onClick="handler()" >
 	
<script>
Function handler()
{
……processing logic
}
</script>
	<button>OK</button>
	
	<script>
	$(function(){
	
	$("button").click(function(){
	….processing logic
	}//end of handler
	}//end of ready
	
	</script>
		<button
		onClick={handler}
		>OK </button>
		
		Function handler()
		{
		..processing logic
		}

We write a call to handler	We register handler callback	We register handler callback

EX1 ---- Write a function component EventEx that displays hello when the button is clicked
Integrate it with App1

EX2 ---- Write a class component EventExC that displays hello when the button is clicked
Integrate it with App1

EX3 ---- write a function component ShowCaps that accepts a string in input field and displays it on the console in capital letters on button click

EX4 --- modify ShowCaps --- add two more textfields --- that accept a number each
On button click show the sum on console

EX5 ---- modify the ShowCaps on button click show the sum on the <p> tag on browser instead of console

Observe --- Component shows the INITIAL value of SUM and not the modified value
WHY ?? Because component is not RERENDERING when SUM changes !!!!
SOLUTION ---
Use the STATE variables

State in React -------------

Every component has an associated state object in it
If the value of the state changes ===== then component is RERENDERED

To use state variables

1. Function components - Use a hook = useState
2. Class components - use this.state and setState API

What is useState hook ?

It is a library function that creates a state variable and its setter method
It returns an array of two elements
Arr[0] = state variable
Arr[1] = setter method

Whenever the setter method will be called the state variable will change and lead to RERENDERING of the component

Ex 6 --- Replace the ShowCaps with ShowCapsC which is a class component .
Observe - how the state is managed in the class component

Observations ----

1. If the function in the class is written as a lambda function then only the "this" is available in that function. Otherwise the "this" is not available in the function

function component state management	Class component state management
Use a hook useState	we use this.state

let [data,setData] = useState(initialval)	constructor()
	{
	this.state={data:"initialval" }
	}

To change the state setData(newvalue) = this Rerenders	To change the state this.setState({data:"newVal"})

EX 7 ---- Write a class component ShoppingList textfield = button
When user enters value in the text field
And clicks the button
The item should added in a div seen on the browser

Practice Question ---

Write a function component IncrDecr
It had two sub components

```
<>
<Incr />
<Decr />
</>
```

Incr.js is a function component that shows a <p> having a number and an incr button
When button is clicked the number in the <p> is incremented

Decr.js is a class component that shows a <p> having a number and an decr button
When button is clicked the number in the <p> is decremented

pass the initial value of the number in <p> tag as a props to Incr and Decr component