Friday, January 13, 2023 11:57 AM

Function component IncrDecr
<input> button
Pass the input to the Incr and Decr component
class component Incr
function component Decr

Lifting State Up -----

1. This is a design implementation using props, state and event handling
2. There has to be a parent (enclosing) component and at least one or mor child (enclosed) component
3. We will keep the state of a variable in the parent component
4. We will write some modifier (like changeValue) that changes the above state also in the parent component
5. The child components will get the modifier function through props
6. On different events the child component will call the modifier function ------ this will result in the change of state in parent component, leading to RERENDERING of parent component .

CAN we call the function directly on onClick without writing a handler ? --------

YES !!!	

HOWEVER - never call a function directly in { } of event attribute !! This **may** go in infinite loop
INSTEAD ---- write a Lambda function and call your function in it

```
<button onClick={ ()=>{ props.modifier("d")} }>
```

Conditional Rendering ----------
Render the part only if the condition is true !!!!

Ex - Write a function component ConditionRenderEx
VIEW
Show LIST Checkbox
IF the Check box is checked show the list of exam centers
Else don't show

Modify the above example --- Put the List in a component ListEx

For conditional rendering we need an expression or a variable that evaluates to true or false
In the JSX use TERNERY IF expression within {}
{ booleanval ? TAG_TO_BE_RENDERED : "" or ANOTHER_TAG }

List Rendering in React …
Use arr.map() and generate JSX for each element

Ex ---- Write a Component For Shopping cart as discussed in class .

HW - DO ShowList and ShowBill components in the ShoppingCart Assignment done in class

---

Lifecycle of React Components ----------

For Class Components	**event**	For Function Components
componentDidMount() **{** **code that should run immediately after the component is rendered** **}**	**Component mounted/rendered**	useEffect(call back,[])
componentWillUnMount() **{** **code that will cleanup as component will not be seen after this** **--- send data to server** **--- reset arrays or objects** **}**	**Component is about To be removed or unmounted**	useEffect(call back this callback should **return a cleanup function** ,[])
componentDidUpdate(prevprop,prevstate) **{** **Code to do something when values of props or state changes** **}**	**When props or state changes**	useEffect(call back,[num])