

React - 2

Date 14/3/28
Page 1

~~event listeners~~ is

⇒ props is equivalent to event listeners, which starts with on.

props ke dvara hum event handling

~~ko~~ kar sakte hai. react me and every these props start with ON.

e.g. to make button clickable

`<button onClick={clickHandler}>` add to call `</button>`

function.

To make working of onclick, call the f^{unc} → function clickHandler() {

without brackets. if we use brackets then it automatically clicked the button because the f^{unc} call.

⇒ event listeners vs props onclick :-

↓

Based on Imperative approach

↓

it is declarative approach

USE STATE syntax :-

initial val.

~~useState(initial)~~

const [title, setTitle] = useState(props.title)

* ~~Ek~~ ek baas to poora UI render hove wala hai.

Normal flow ke hiab se humne poora UI render kar diya. After rendering uska state me agar hum kuch change karna chahiye hai like variable ke andar value change holeti hai but UI me soi change nahi kiya means no repaint done to it. Kahan UI update nahi hota. But if I want to update UI we have to use the STATE.

→ Before using State we have to import first.

→ import React, { useState } from 'react';

↓

React hook
utility fn. with the help of we can do different things.

→ useState (props.title)

↓
input

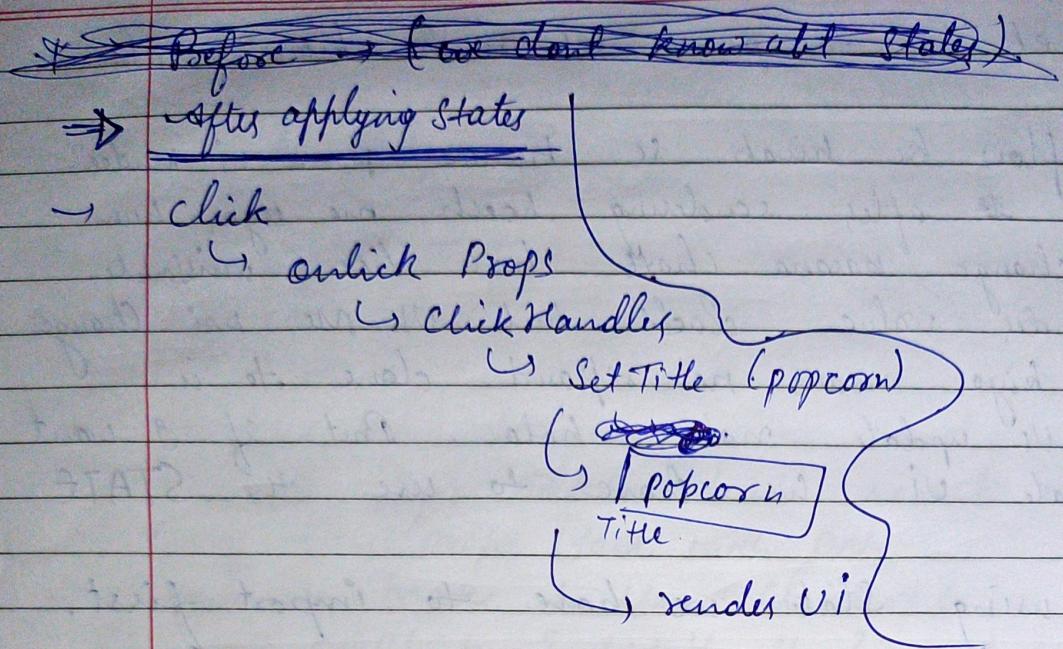
initializing val.

o/p → (fn for updating the value.)

↓
value of the variable

It is just setter

* React Hooks :- its just like some utilities fn with which we can do different kind of things.



- * use state Hook always returns an array with two elements (i) value of the variables
~~(ii) fn of return the value~~
(ii) fn for updating the value

Qn is set settled for render immediate or scheduled?

⇒ State update in react are asynchronous.
there is no guarantee that the updates will be made immediately.

the react run each action in queue, updates the final state, and return it back.

This is called lazy-computation

React will calculate the new state only when it actually need.

- ⇒ If you have data which might change and where changes to that data should be reflected on UI so we have to use State.
- ⇒ To Bhi State Change hota hai per component instance basis hota hai. means has component be instances along se handle honge.
eg → Tis button par click kara the usi me changes hote hai.
- 1:5:23
- ⇒ if i have to find the value in particular field so i need to use for that
→ event.target.value
in parameters event object is bydefault there
- Bohot Sari chejge hum handle kar sakte hai.

1:16: To set value of "hai" wo Schedule Karte hai.

* 1:17:17 is there any way to handle all the parameters at once.

→ yes! with the object.

$\Rightarrow \text{Const}(_, _) = \text{usestate}(_);$

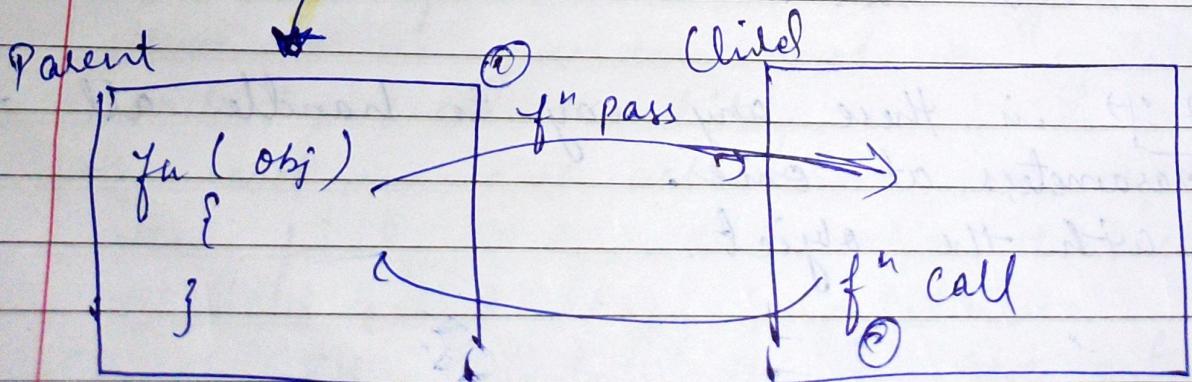
handling for multiple states using object and
performance remains same

* 1:45

How can child communicate to parent?
(Lifting up)

→ Props fⁿ

* Props :- Har ek props ke ander ek handle fⁿ se map karna padta hua. wo handle fⁿ mujhe khud se Create Karna padega us handle fⁿ Ke ander mujhe logic define Karna padega ki click Karne Ke Baad Kya hona chahiye.



in short :- Parent me ek pehle se hi
fⁿ hoga obj parameters ke sath taki
wo access kar sake.

To Sabse pehle parent fⁿ pass karega child
ko then child fⁿ call karega with
parameters of obj.