

React - 4

* useEffect hook :- managing side effect

- `useEffect (() => {
});` Tis bhi component me useEffect likha hai to wo us component pe render hone ke baad isme To bhi code hoga use execute krega.
- side effect :- it is a change that effect something outside the component being rendered.

~~eg:-~~ hum kisi component par koi action kia or us component ko chod kar baki bache hue has kisi element par change ho gya to wo use side effect consider kag sakte hu.

eg :- Changing the document title.

- Modifying the state of a component parent.
- Making an api call.
- Modifying Browser history.

→ `useEffect (() => {`

`console.log ("ui rendering done")
});`

↑
after every renders it print

* Variation 2 → for one ~~time~~ time

useEffect () => {

Console.log ("UI rendering done")
 } , []);

↑
dependency list

* Variation 3 → ~~on first render~~ whenever dependency changes

useEffect () => {
 Clg ('change observed')
 } , [test]);

↑

Tanhi test ki value change hogi to mera useEffect chalta rahega

* Variation 4 → to handle unmounting of a Component (20:00) HD

useEffect () => {
 Clg 1
 test () = {
 Clg 2
 } }
 ↓ execute
 ↓ execute

esa hi hoga hai hamisha
 return wala pehle execute
 hoga then upper wale

variable state

⇒ useState → Kisi ~~state~~ Ko hande Karna
 or ye hi value UI par bhi
 reflect ho. is liye ise useState karte
 hain.

yaani par 2 values milte hain

- 1) Variable ki value
- 2) ek fn jo variable ki value ko update
 karta hai.

⇒ useEffect hook :- agar tum apne component
 ke render hone ke basoں Ko
 task karana chahte ho wo task ko
 hum is useEffect ke andar define
 karte hai usi task ko hum side effect
 bolte hai.

Syntax :-

useEffect (callback fn, array of dependencies)

⇒ Tumhare component ke normal flow of execution
 Ke alawa tum koi task karana chahte
 ho to usko hum side effect bolenge,
 Or side effect karane ke tarasekhe kia
 bololia useEffect.

* Component Mount :-

Ki wo component render ho gya, load ho gya, wo component don po a chuka hua.

* Component Unmount :- don se remove karna.

* Class-Based Components :-

↳ Components Did Mount.

↳ Components Did update.

↳ Components will unmount.

No need to understand

instead of this we read

• use Effect Hook

→ `useEffect (() => { })` every Render
3); }

→ `useEffect (() => { })` first Render
3, []) ; }

→ `useEffect (() => { })` whenever name changes
3, [Name]); } the callback fⁿ will execute.

dependency

How print window size.

* Top course Project