



## Today's agenda

① purpose of React hooks

② useState, ~~useEffect~~, useContext

③ implementing life cycle methods componentDidMount,  
componentDidUpdate, componentWillUnmount

④ Creating custom hooks → useDebounce

useEffect() => {

⇒ Component did mount

}, [ ])

---

useEffect() => {

⇒ this will work as  
Component did update

}, [ count ])

Q what are react hooks?

React hooks are functions introduced in React 16.8

that let developers use React state & lifecycle  
features in functional components

use State ↴

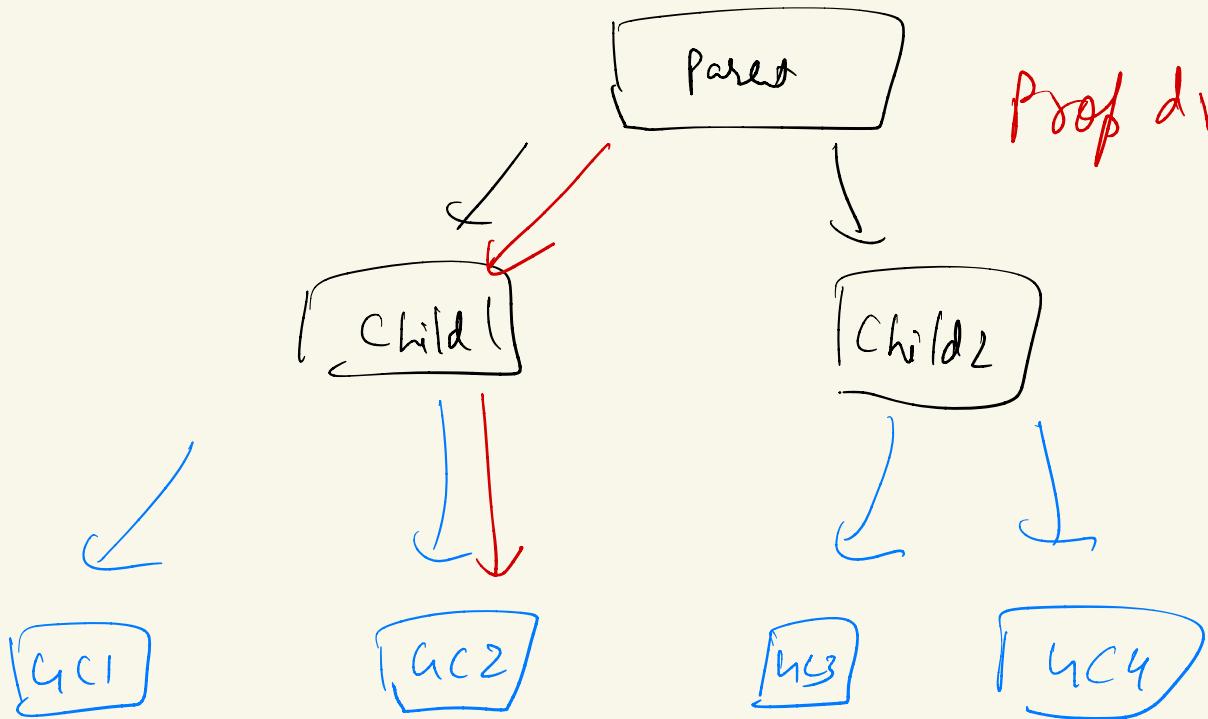
State & set State

use Effect ↴

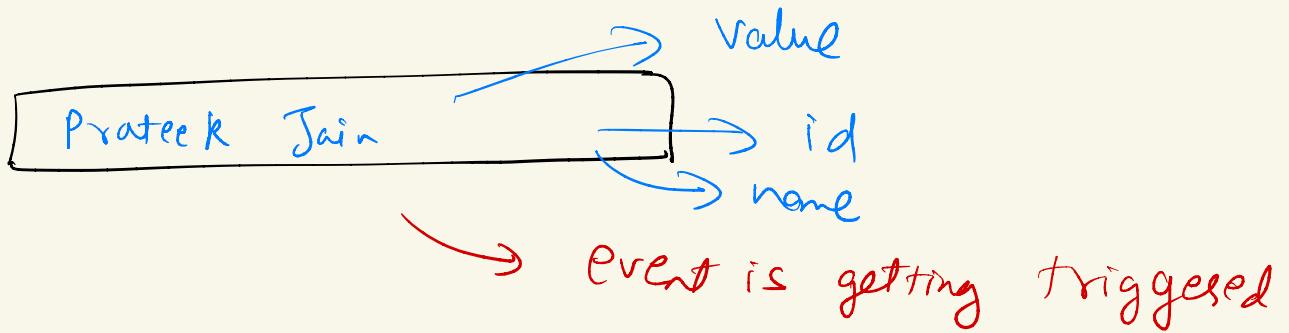
use Context ↴

Sharing State  
globally without  
passing props down  
manually

Prop drilling



Custom hooks → Custom hooks in React are user-defined functions that encapsulate reusable logic, allowing you to share & reuse stateful logic across different components



Event has every thing related

Your actions

Event. target will have all the details  
of your element

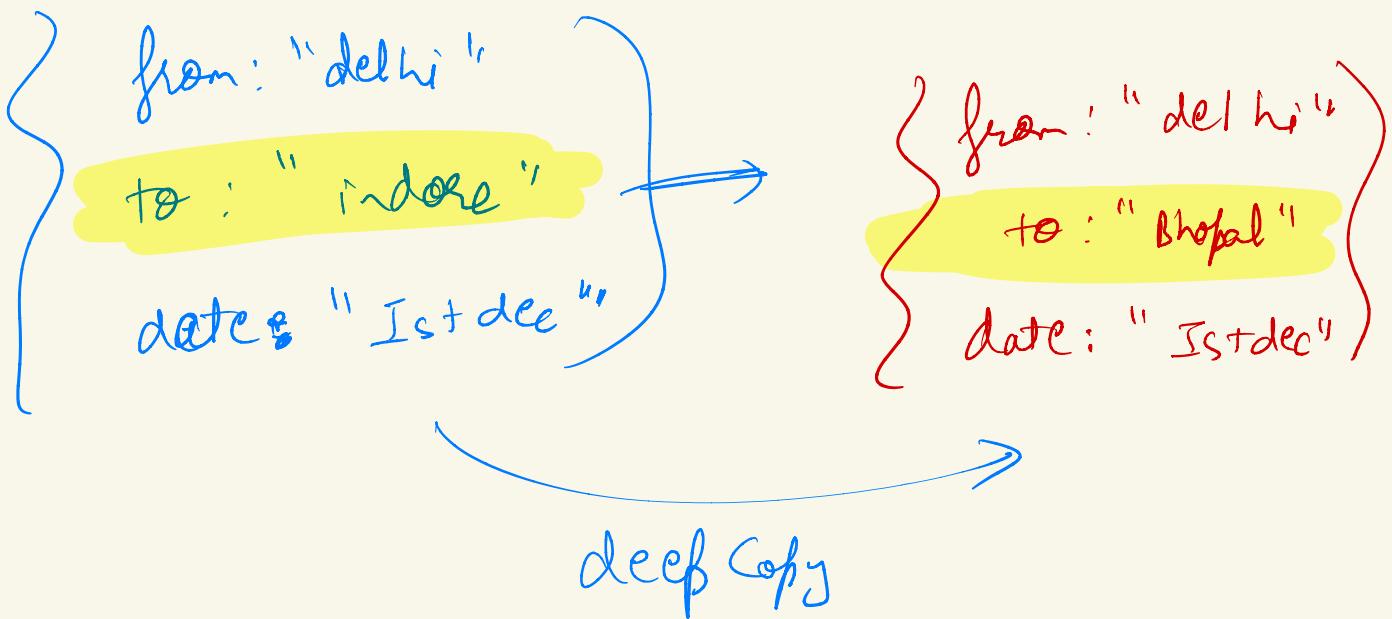
Value & id

→ which element you are

writing

{ from :  
to :  
date :

let newObj = { ... obj, to: "new value" }



Const { id, value }  $\Rightarrow$  e. taget

---

DSA

Search

bing API → results

Value →

```
useEffect(() => {
  const getData = async () => {
    try {
      setLoading(true);
      const responseObj = await fetch("https://dummyjson.com/products");
      const data = await responseObj.json();
      setProducts(data.products);
    } catch (e) {
      console.error(e);
    } finally {
      setLoading(false);
    }
  };
  getData();
}, []);
```

React batches set state  
updated

~~loading~~ → false  
products

1st render

```
const Counter = () => {
  const [count, setCount] = useState(0);
  const increaseBy5 = () => {
    setCount(count + 1);
    setCount(count + 1);
    setCount(count + 1);
    setCount(count + 1);
    setCount(count + 1);
  };
  const increase = () => {
    setCount(count + 1);
  };
  return (
    <div>
      {count}
      <button onClick={increaseBy5}>increase count by 5</button>
      <button onClick={increase}>increase count by 1</button>
    </div>
  );
}
```