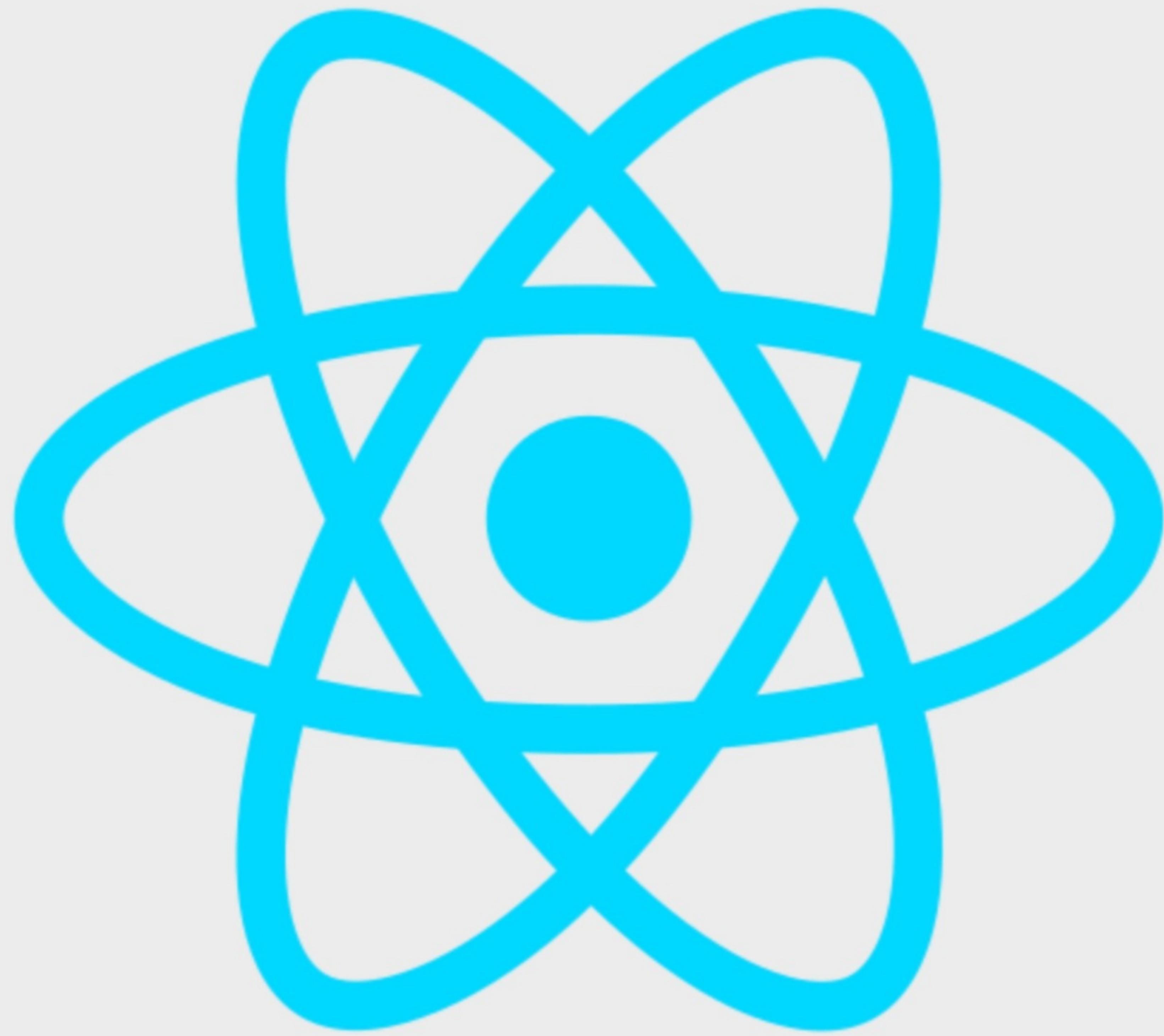


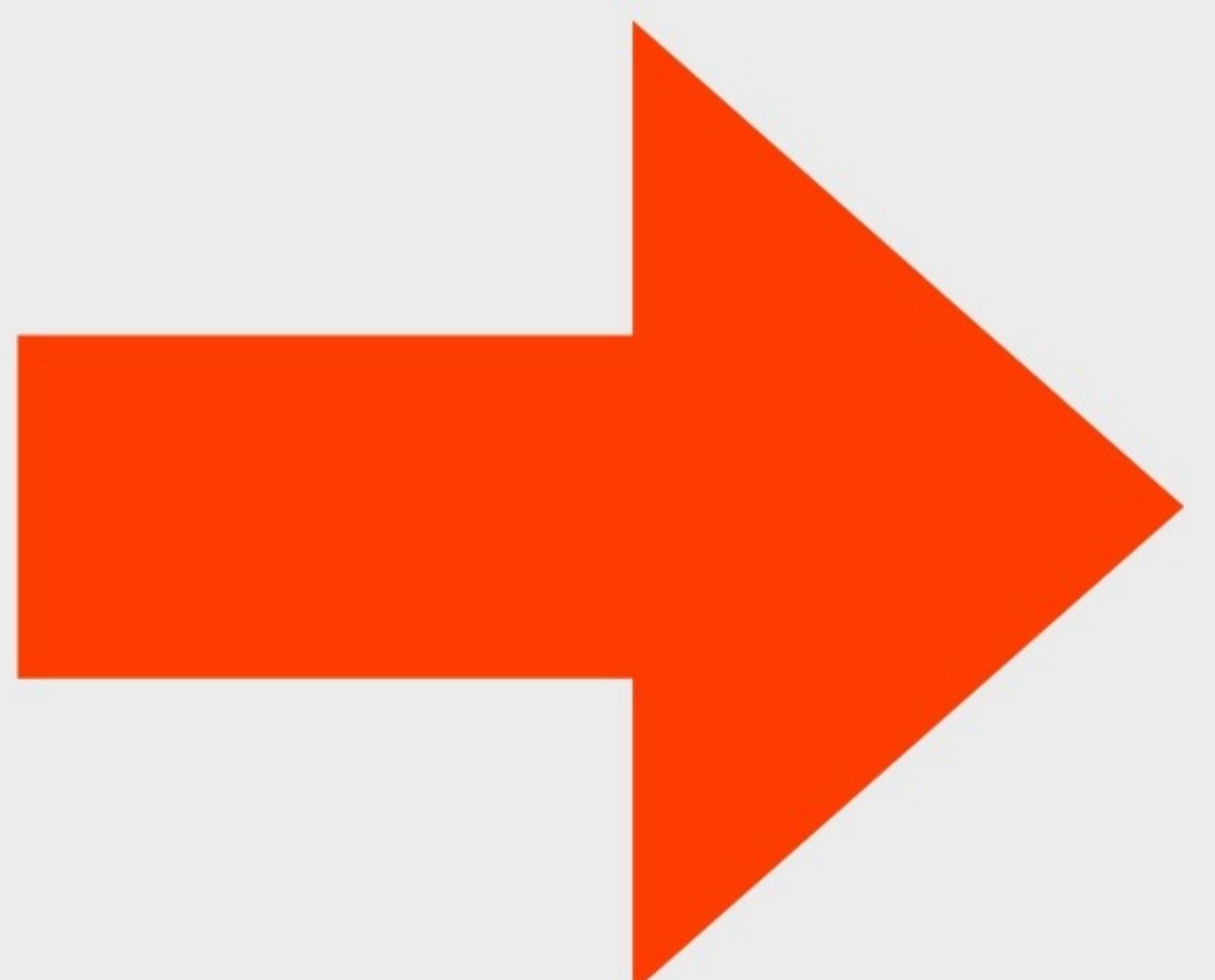
# Power of React

# 6 Hooks

# Simplified



**K RAKESH**  
*@webDeveloper*



# useState

Create and update state values.

holds the state value



```
const [count, setCount] = useState(0)
```

```
setCount(1)
```

changes the count value to 1

used to update the state value

creates the state resources and sets the initial value to 0

# useEffect

Used to run side effects in the component such as fetching data or adding listeners.

runs after the initial render

runs just before the component unmounts

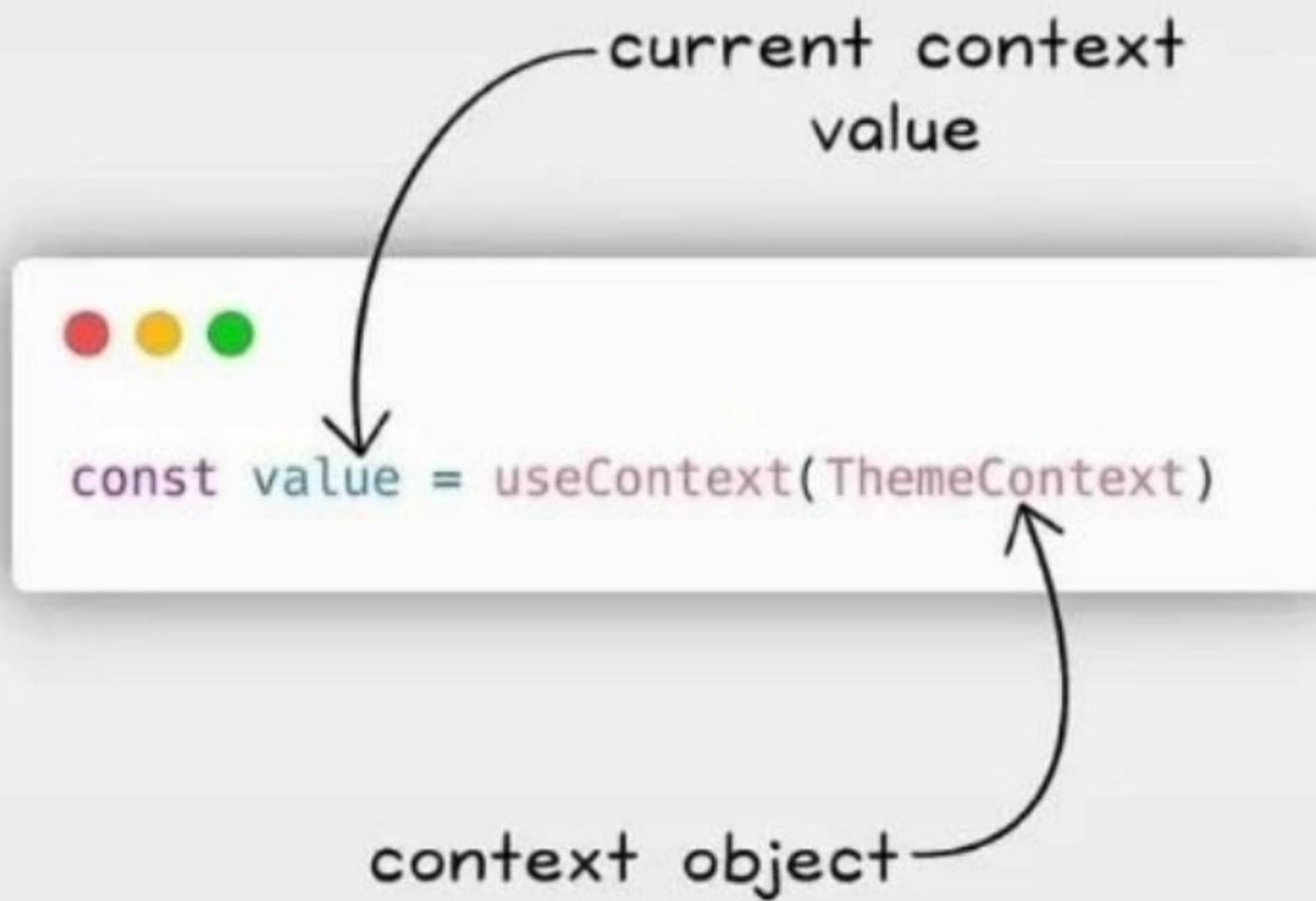
```
useEffect(() => {  
  addListeners()  
  return () => {  
    removeListeners()  
  }  
})
```

```
useEffect(() => {  
  fetchUserInfo(userID)  
}, [userID])
```

runs after the first render  
and every time userID updates

# useContext

Accepts a context object that's created using `React.createContext`, and returns the current value of that context.



# useMemo

Returns a memoized value which only gets recalculated when the defined dependencies change.

holds the cached value returned by calcSurfaceArea

```
const area = useMemo(() => {  
  return calcSurfaceArea(size)  
}, [size])
```

area updates every time size changes

# useCallback

Returns a memoized version of a callback that only changes when the dependencies change.

a memoized version of  
updateSurfaceArea



```
const handleRenderArea = useCallback(() => {  
  updateSurfaceArea(size)  
}, [size])
```

handleRenderArea  
updates when size  
changes value

# useReducer

Similar to useState but also lets you use your own update state logic.

contains state value

a function to update state

```
const [state, dispatch] = useReducer(updateCount, {count: 0})
```

```
const updateCount = (state, action) => {
  switch (action.type) {
    case 'increment':
      return {count: state.count + 1}
    case 'decrement':
      return {count: state.count - 1}
    default:
      throw new Error();
  }
}
```

```
dispatch({type: 'decrement'})
```

the action object

calls  
updateCount

a callback that takes the current state and action object and returns the new state

# Did you find it useful ?

Leave a **comment** !



**K. RAKESH**

@Web Devloper

**FOLLOW FOR MORE**

**Like**



**Comment**



**Repost**

