NAME OF
INSTITUTION OR
UNIVERSITY

2022 April 24

REACT HOOKS P2

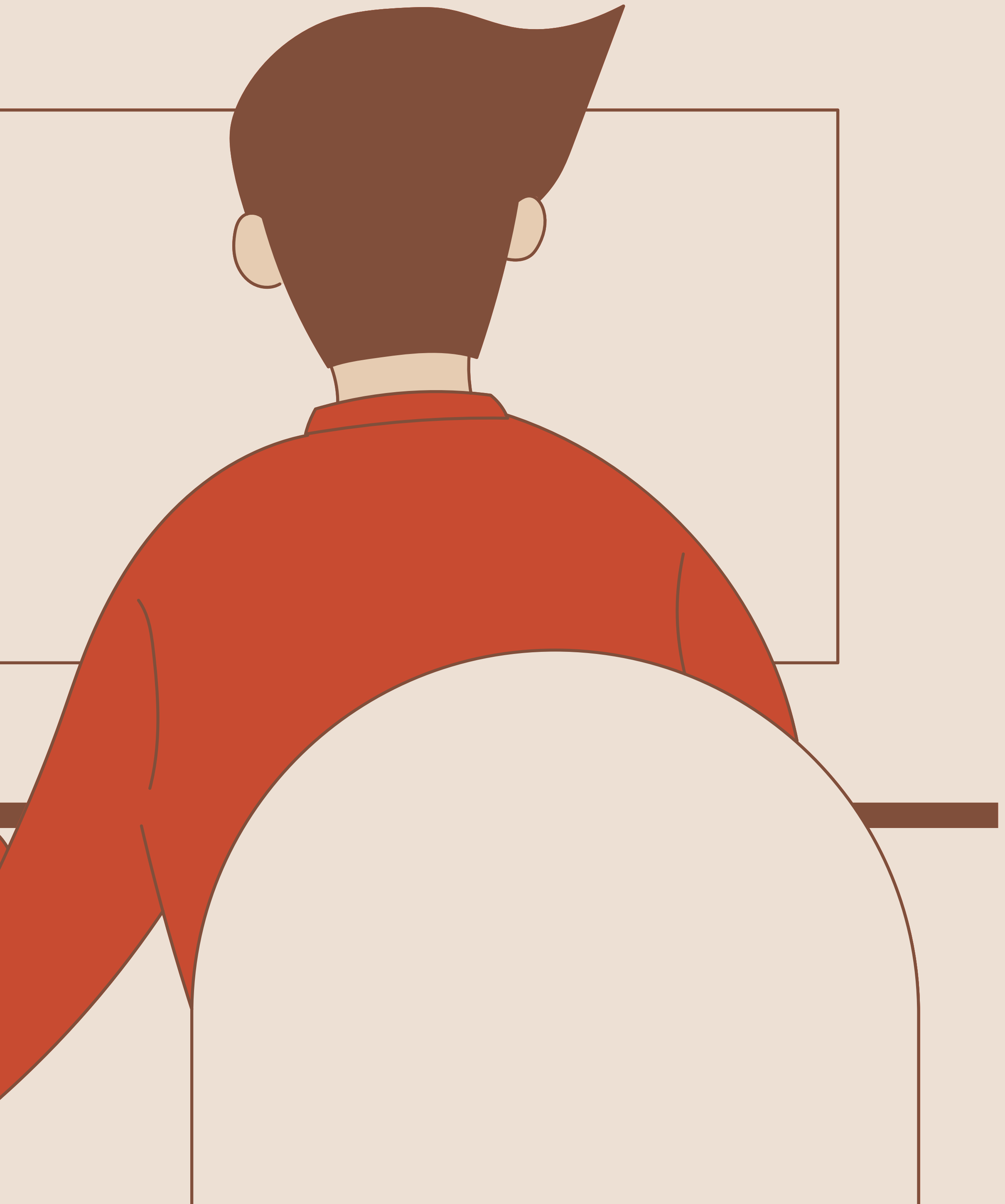
Student

hema vardhan

Final Defense

NAME OF
INSTITUTION OR
UNIVERSITY

Table of Contents



- **this is used to stop running of function unnecessarily by comparing its dependencies**
- **if dependency dosen’t change then function wont execute again “this is done to avoid delay of rendering other components”**

```
const [num,setNum] = useStae(0)
 const increment = ()=> setNum(num+1)
const iseven = useMemo( ()=>{
 let i=0; while(i<100000) i++;
if(num%2==0) return true
else false } , [num]) it runs only if num changes
there are two components
<button onClick={increment}>{num}<button/>
<span>{iseven() ? ‘even’ : 'odd'}<span/>
<other component/>
```

useMemo

it remembers or cache a value calculated once and it evaluates if and only if the dependency of the caluculation changes
The useMemo Hook can be used to keep expensive, resource intensive functions from needlessly running.

dependency

it stores the expensive function value in cache and checks if it needs to be run again.
it runs if dependency of function changes

```
const calVal = useMemo(
  function
,[dependencyValOfFxn])
```

useRef

useRef is used to access DOM nodes directly with in functional components.

most common use case is to focus on text input

use case 1: to reference an element

```
const inputRef = useRef(null)
useEffect(()=>{
inputRef.current.focus()
},▯)
```

```
<input type='text' ref={inputRef}/>
```

here ref is used to reference the element so that we can reference to input here using inputRef and manipulate the element using inputRef.curret

use case 2:

```
const [timer,setTimer] = useState(0)
useEffect(()=>{
  const interval = setInterval(()=>{
       setTimer(prev=>prev+1)
   },1000)
   return ()=>{clearInterval(invterval)}
},[])
<button onCLick=
{clearInterval(interval)}>stop<button/>
```

this dosen't have access to interval s error appears

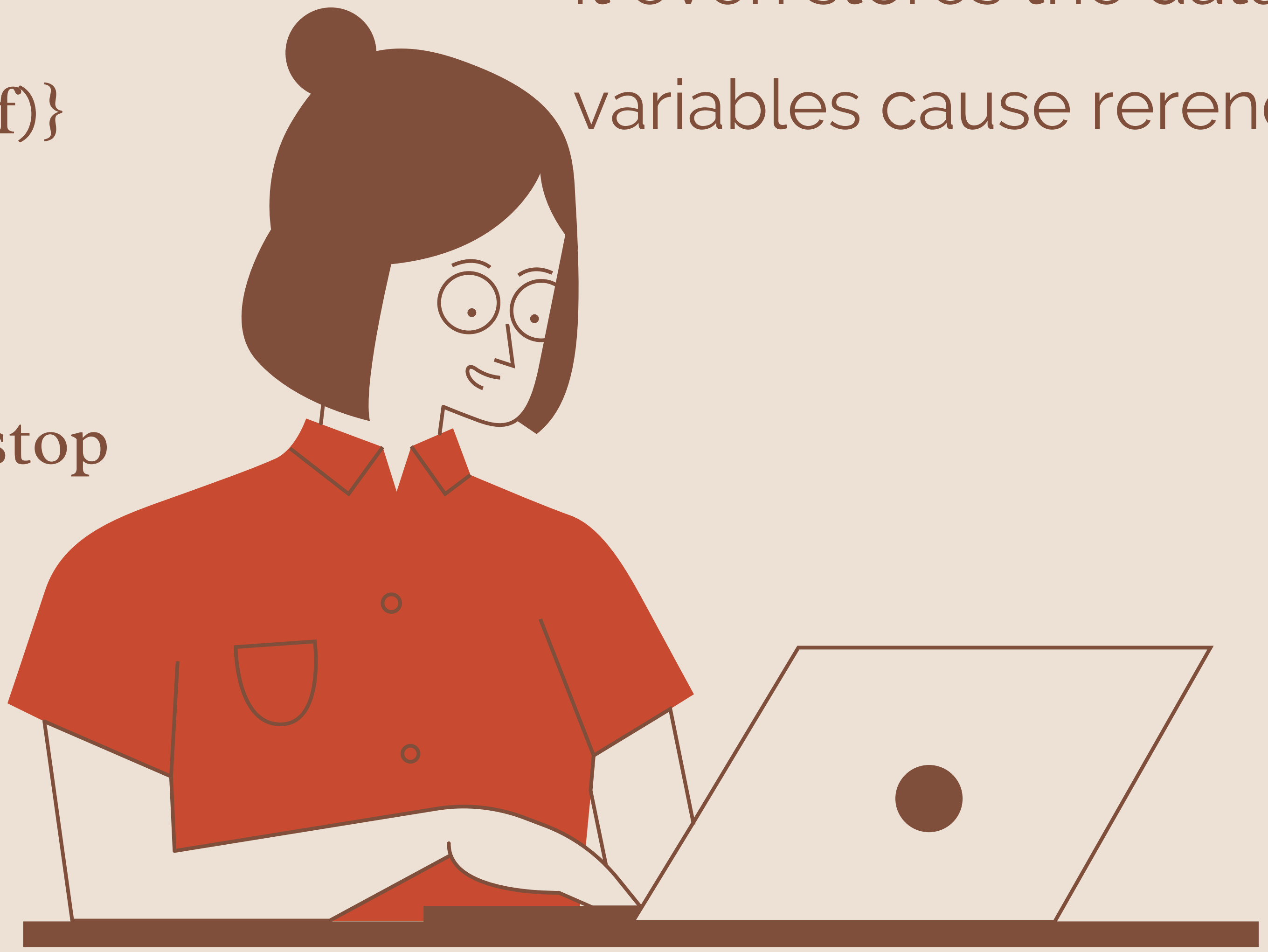
as we know useRef is used to reference any DOM it is also used to store any mutable value and also it dosen't casuse any additional rerenders

next page full details

Hypothesis 3

```
const [timer,setTimer] = useState(0)
const intervalRef = useRef()
useEffect(()=>{
 const intervalRef.current = setInterval(()=>{
 setTimer(prev=>prev+1)
 },1000)
 return ()=>{clearInterval(invtervalRef)}
},[])
<button onCLick=
{clearInterval(intervalRef.current)}>stop
<button/>
```

- this works just fine and it removes interval when ever clicked the button
- so here intervalRef is referencing the mutable variable or function which we used to clear interval
- this won't cause rerenders when value in it changes
- it even stores the data even other state variables cause rerender of that component



customHooks

custom hooks are basically js functions
whose name starts with use
a custom hook can also call other hooks
if required
to share logic we use these

useCase

```
const [count,setCount] = useState(0)
useEffect(()=>{
document.title=`title- ${count}`
},[count]) // render when count
changes
<button onClick=
{setCount(count+1)}>change<button
/>
```

if we want to implement the same logic in anyother component to update title on count change it just makes code redundent or repeate for the same logic so we can create customhook and use it here to replace the logic

- function name of cusotm hooks always start with “use”

BROCHILL

2022 April 24

Thank you for listening!

Student

hema vardhan reddy

Final Defense