

**useRef()**

# What is ref ?

Before hooks, ref was used with class components to directly access DOM elements or React elements.

# What is useRef ?

useRef is a Hook that returns a mutable ref object whose `.current` property persists across renders.

# Basic Syntax:

```
const inputRef = useRef();

const handleFocus = () => {
  inputRef.current.focus();
}

return (
  <div>
    <input ref={inputRef} type="text" placeholder="Type here..." />
    <button onClick={handleFocus}>Focus Input</button>
  </div>
);
```

- ❑ `useRef()` is like a box you can store anything inside (DOM node, number, object, etc.).
- ❑ Changes to `.current` do not cause re-render.
- ❑ It's not reactive, but persistent.

# Two Main Uses of useRef

1. Accessing DOM Elements
2. Storing Mutable Values (Without Re-render)

# 1. Accessing DOM Elements

When you need to directly manipulate a DOM element (e.g., focus, scroll, play video).

```
const videoRef = useRef();

function playVideo() {
  videoRef.current.play();
}
```



## 2. Storing Mutable Values (Without Re-render)

Sometimes you want to store data between renders without triggering updates.

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

renderCount.current += 1;

return (
  <div>
    <h3>Count: {count}</h3>
    <h4>Render Count: {renderCount.current}</h4>
    <button onClick={() => setCount(count + 1)}>Increase</button>
  </div>
);
```

## Difference between useState and useRef:

Feature	useState	useRef
Triggers re-render on update	✓ Yes	✗ No
Persistent between renders	✓ Yes	✓ Yes
Common use	Store UI data	Store DOM reference or mutable value

**Thank  
You**