



## useRef() React Hooks

- In React, the useRef hook is used to **create a mutable reference** to an element or a value. Unlike state variables managed by useState, the value of a useRef does not trigger re-renders when it changes.
- It's often used to **interact with the DOM directly** or to store values that don't need to trigger re-renders but still need to persist across renders.
- **Syntax:**

```
// Create a useRef instance
const myRef = useRef(initialValue);

// Access and modify the reference
myRef.current = newValue;
```

- myRef is the object returned by the useRef hook.
- **myRef.current** is the property you can access and modify. This property **holds the value you assigned to it**.
- You can **read or modify myRef.current** as needed within your component.

```
return (
  <div>
    <input ref={myRef} />

    <button onClick={() => console.log(myRef.current)}>
      Log Value
    </button>
  </div>
);
```

- You can use the reference in JSX attributes, event handlers, or any logic within your component.

- Remember that changes to the current property of a useRef object do not trigger re-renders in your component.
- **It's commonly used for**
  - interactions with the DOM,
  - storing previous values, or
  - holding mutable values that don't impact the rendering of your component.