

React- Day -9: useRef & useReducer:

Use Ref:

useRef to access to DOM elements, manage focus,

A function that uses the ref to **focus on the element**.

A ref is like a "**magical pointer**" that allows you to interact directly with a specific **DOM element**

useRef() only returns one item. It returns an Object called **current** (focus(), value)

don't **trigger a re-render** of the component **but state managed**.

Eg: **const ref= useRef()** => returns current object

(focus(), value) ?

ref.current.focus() -> **focus on the element**.

Ref.current.value -> input box -> value

useReducer in React:

It is similar to the useState Hook.

It allows for **custom state logic**.

If you find yourself keeping **track of a group of states** that rely on **complex logic**.

The `useReducer` Hook **returns the current state value and a `dispatch` method.**

trigger a re-render of the component **when state updated using `dispatch`.**

Eg:

```
const[state, dispatch] = useReducer(ReducerAction, statevalue)
```

State = stateValue

Dispatch = reducerAction()

Game:

Eg: Tic Tac Toe

In a Tic Tac Toe game using React,

you can use the **`useRef` hook to maintain a mutable reference to the game state without triggering re-renders.**

Additionally, you can use the **`useReducer` hook to manage complex state transitions** based on user interactions.