**What is the useReducer() hook**

The **useReducer()** hook is a React hook that provides a more robust way to manage complex state logic in functional components compared to the **useState()** hook. It is generally used when you have complex state logic that involves multiple sub-values or when the next state depends on the previous one.

**useReducer()** is particularly useful for cases where you must manage state transitions in a predictable manner. It works similarly to how reducers function in Redux if you're familiar with that library.

**Syntax**

*The* ***useReducer()*** *function takes two primary arguments:*

1. **reducer**: A function that defines how the state is updated. It takes the current state and an action object, and it returns a new state.
2. **initialState**: The initial state that the component will have when it is first mounted.

***It returns an array with two elements:***

1. **state**: The current state, based on all the applied actions.
2. **dispatch**: A function to which you can pass an action to trigger a state change.

**Example:**

import React, { useReducer } from 'react';

const initialState = { count: 0 };

function reducer(state, action) {

switch (action.type) {

case 'increment':

return { count: state.count + 1 };

case 'decrement':

return { count: state.count - 1 };

default:

throw new Error();

}

}

function Counter() {

const [state, dispatch] = useReducer(reducer, initialState);

return (

<>

Count: {state.count}

<button onClick={() => dispatch({ type: 'increment' })}>+</button>

<button onClick={() => dispatch({ type: 'decrement' })}>-</button>

</>

);

}

**Explanation**

* **initialState** is an object that has a **count** property set to 0.
* The **reducer** function takes the **state** and an **action** as arguments. Depending on the **action.type**, it returns a new state.
* Inside the **Counter** component, **useReducer()** is called with **reducer** and **initialState** as its arguments.
* **state** contains the current state, and **dispatch** is a function to update the state.
* The buttons use the **dispatch** function to send actions that either increment or decrement the count.

The **useReducer()** hook is a powerful way to manage state in your React applications, especially when your component has more complex interactions.