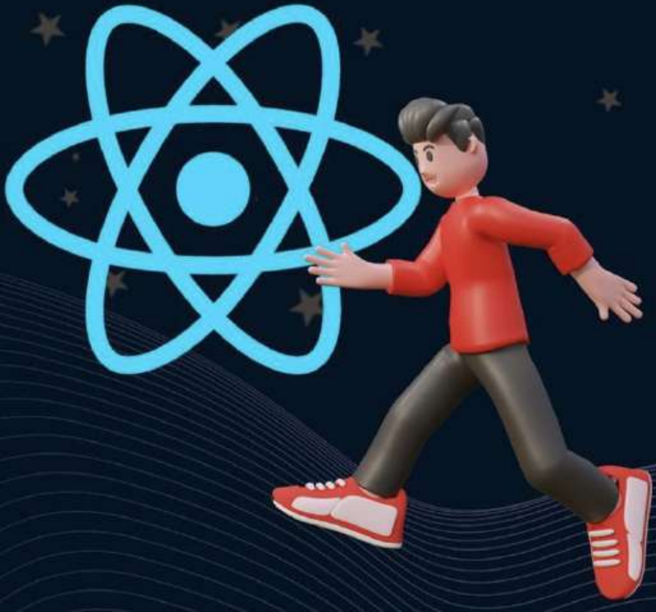
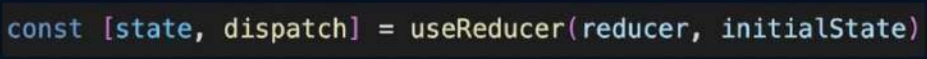
@pyplane_code

useReducer hook



Introduction

useReducer is a interesting alternative for useState in circumstances when we are dealing with complex components in order to manage their state. This react hook takes in as arguments the reducer function and initial state. It returns the state and dispatch function



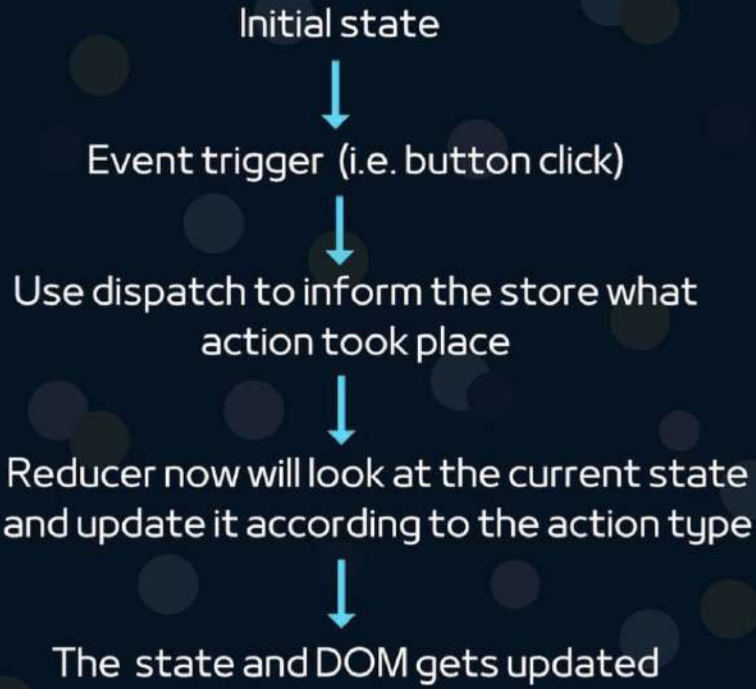
```
const [state, dispatch] = useReducer(reducer, initialState)
```

dispatch - a function for updating the state
reducer - a function that take the current state and an action as arguments and returns the new state depending on the action taken

With the dispatch function the store is informed that something happened, while the reducer takes care of the actual state update depending on the action that was passed to the dispatch function

Scheme

Before we take a look at an example, let's have a glance at a simple scheme to understand the steps taken when using the useReducer hook



Initial state

↓

Event trigger (i.e. button click)

↓

Use dispatch to inform the store what action took place

↓

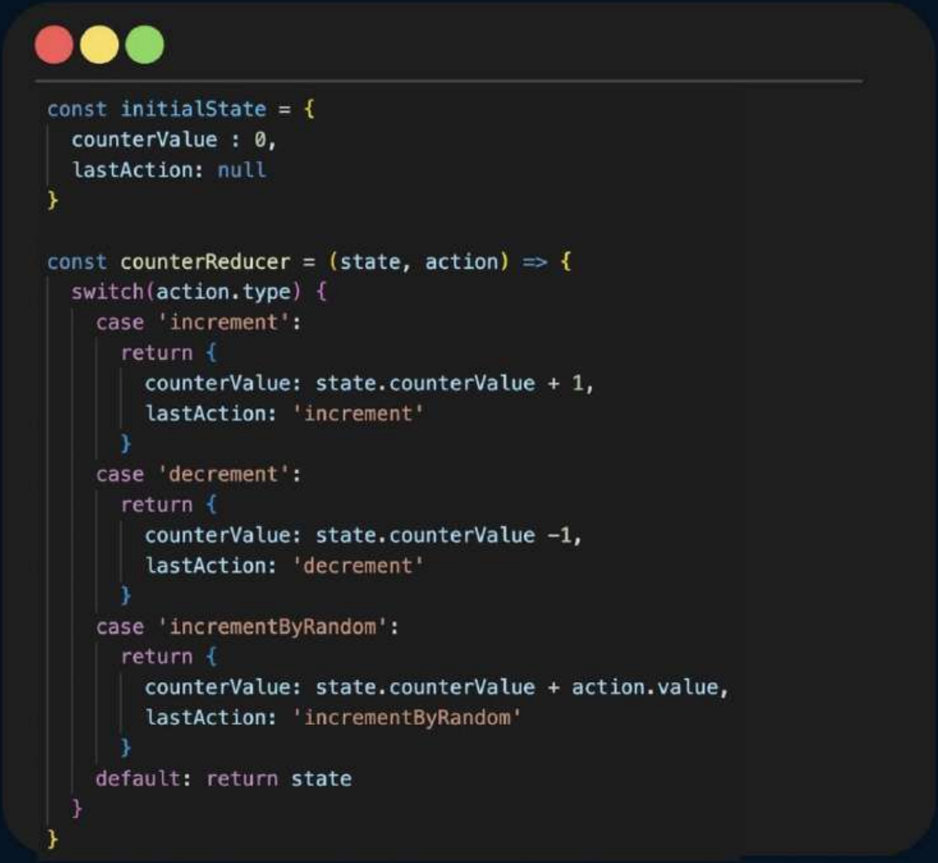
Reducer now will look at the current state and update it according to the action type

↓

The state and DOM gets updated

Reducer

Let's now take a look at a simple example starting from the reducer function

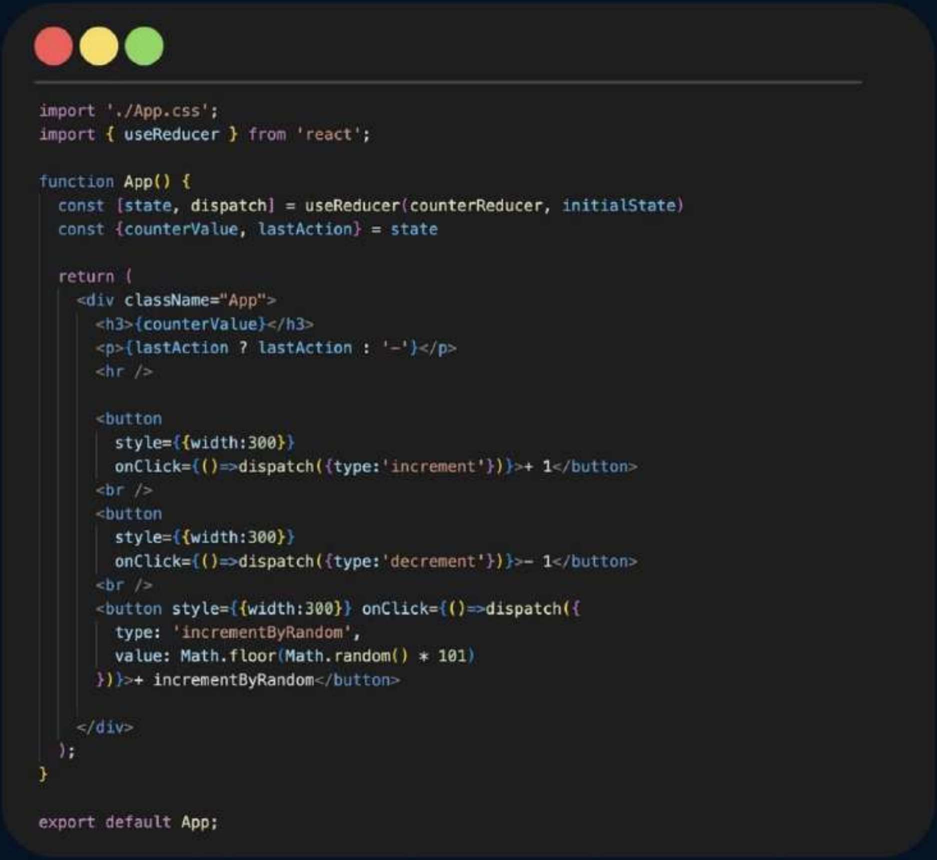


```
const initialState = {
  counterValue : 0,
  lastAction: null
}

const counterReducer = (state, action) => {
  switch(action.type) {
    case 'increment':
      return {
        counterValue: state.counterValue + 1,
        lastAction: 'increment'
      }
    case 'decrement':
      return {
        counterValue: state.counterValue -1,
        lastAction: 'decrement'
      }
    case 'incrementByRandom':
      return {
        counterValue: state.counterValue + action.value,
        lastAction: 'incrementByRandom'
      }
    default: return state
  }
}
```

App.js

Now let's see how we can use the useReducer hook with the defiened reducer and initial state



```
import './App.css';
import { useReducer } from 'react';

function App() {
  const [state, dispatch] = useReducer(counterReducer, initialState)
  const {counterValue, lastAction} = state

  return (
    <div className="App">
      <h3>{counterValue}</h3>
      <p>{lastAction ? lastAction : '-'}</p>
      <hr />

      <button
        style={{width:300}}
        onClick={()=>dispatch({type:'increment'})}>+ 1</button>
      <br />
      <button
        style={{width:300}}
        onClick={()=>dispatch({type:'decrement'})}>- 1</button>
      <br />
      <button style={{width:300}} onClick={()=>dispatch({
        type: 'incrementByRandom',
        value: Math.floor(Math.random() * 101)
      })}>+ incrementByRandom</button>

    </div>
  );
}

export default App;
```

Redux toolkit

Also take a look at redux toolkit which is an alternative for the useReducer hook

@pyplane_code

Intro to
Redux toolkit