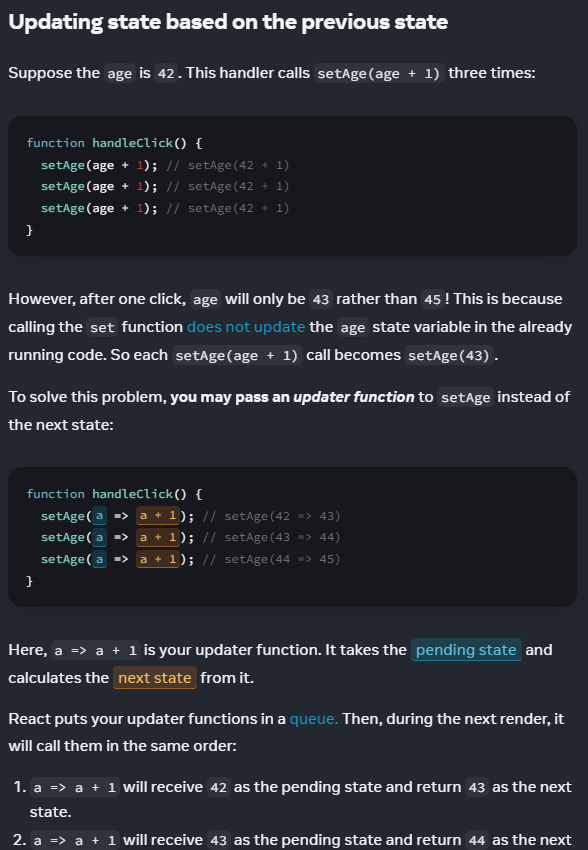
1. [Use Reducer Function](https://react.dev/reference/react/useReducer) Really well explained.
   1. const [state, dispatch] = useReducer(reducer, initialArg, init?)
   2. [Must read to understand dispatch function](https://dev.to/dustinmyers/what-even-is-a-dispatch-function-27ma)

Let me explain the useReducer hook:  
it takes two arguments --> A reducer function as defined below and the initial value which is also the state definition.  
it returns two things --> current state and a dispatch function.  
So we have 4 things to talk about - reducer and dispatch function, current state and it's definition.  
  
**reducer** 🡪 It is a function which we are supposed to provide to this hook for making updates to the state and provide  
it as a single function to the useReducer hook. The reducer function in this hook must be defined in a way such that it  
receives two JS objects as arguments : current state and [ACTION\_TYPE and some other information if need be but as a  
single JS object]. Here it is state and {type, payload} object which contains the type and some other information. It returns in the end.

**dispatch** 🡪It first checks if the state is a simple JS object and then it checks if it is defined or undefined.  
Once CONFIRM IT CALLS THE REDUCER FUNCTION WITH CURRENT STATE AND ACTION TYPE/MORE INFO AND IF A STATE CHANGE IS DETECTED BASED ON THE NEW STATE RETURNED BY THE REDUCER FUNCTION DISPATCH FUNCTION RENDERS THE DOM ELEMENT AGAIN. A lot of this is done in the backend which we do not get to know, so it is abstract.   
  
**Current state** is the state being sent to reducer function through dispatch function.

**Definition 🡪**the number of key-value pairs that you need to use in the program for the information to be managed by useReducer state hook is defined when we set the initial value, it can always be updated ofcourse, but for structure we do so there or we pass an empty JS object.

[Video on react ecosystem 2023](https://www.youtube.com/watch?v=6j9tnGMbm2c)

1. 
2. [Video example of above important issue](https://youtu.be/bkRHbbYdOqs) The gist of the matter is that, whenever we update the state WITH a callback anonymous function, IN THE BACKEND REACT injects the latest snapshot of the state for us to work with, and not some stale value.