**Use of state in React**

1) We can add event listener to elements using onClick, onChange etc.

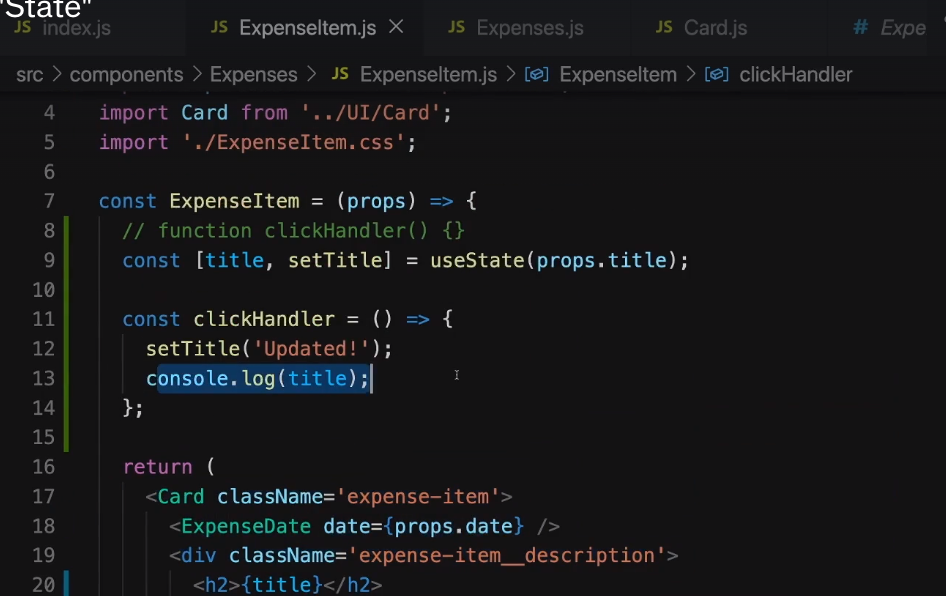
If we have any value in curly braces and if that value is changed, the UI does not update the value.

2) By using our components inside parent components, react keeps on calling the component function one after the other up until it does not find any component function that needs to be rendered.

3) Once everything is rendered, react by default does not repeat the rendering of the components. That is why react uses state to re-render the component.

We need to reinitialize the component to make sure that the updated property is visible on the DOM and that is done via useState.

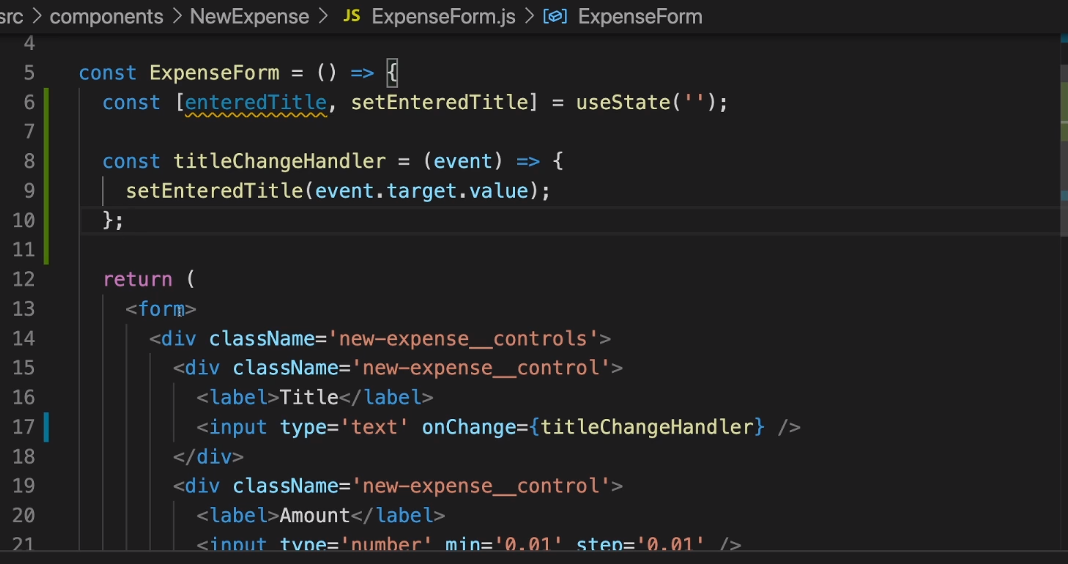
useState returns two properties first one the variable name on which we want to apply the changes and the second one is the function that we need to trigger the state change.



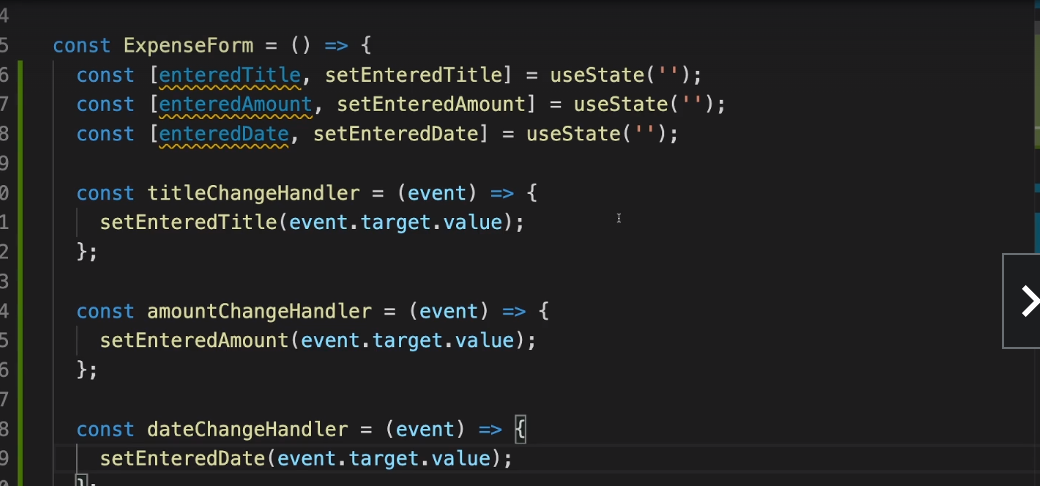
In this case setTitle is the custom function that is responsible for updating the title value. State is a per component basis.

4) Listening to user input form values:-

We can use onChange event will trigger for every key stroke. We can then manage the state of

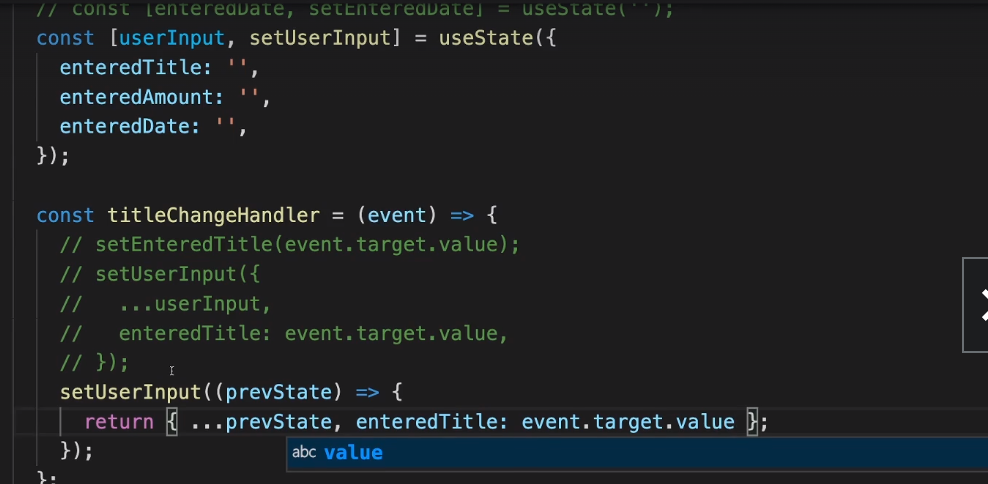


the parameters as stated above. We can also handle multiple states as follows.

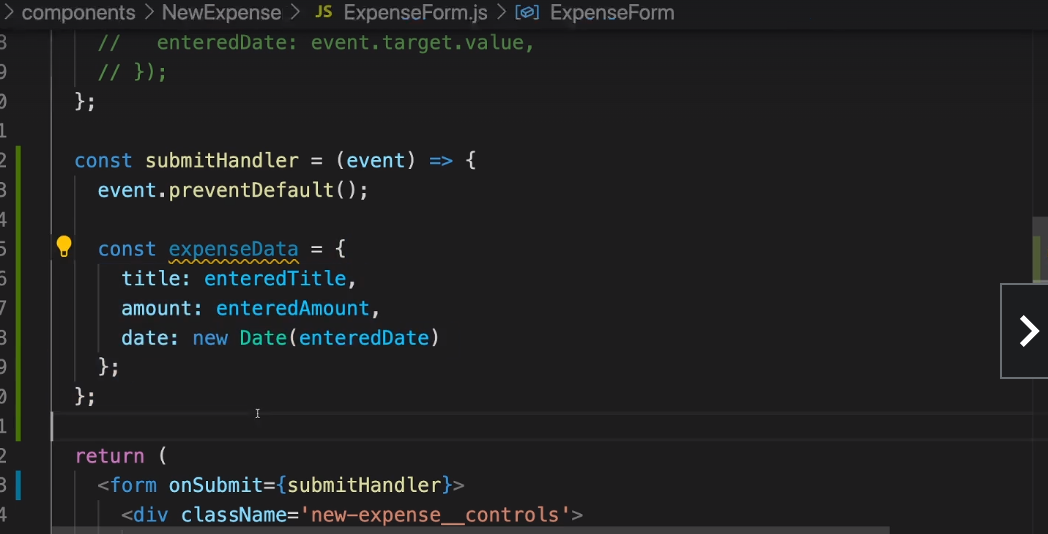


React schedules state updates it does not perform state updates instantaneously.

So if we want to handle all the state in one go, then we use object pattern as mentioned below.

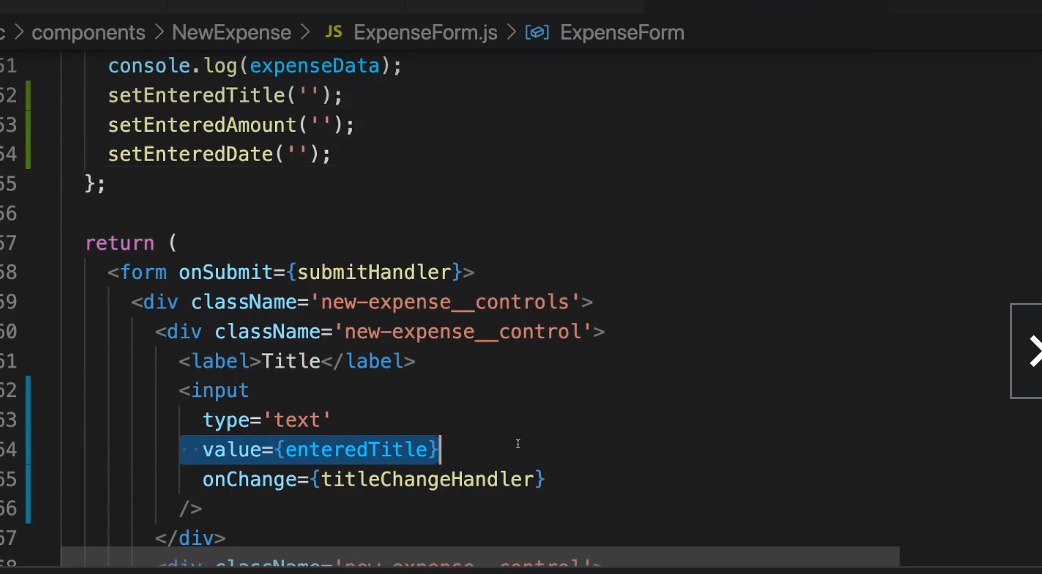


Submit handler for form submission.



Two way data binging in react:-

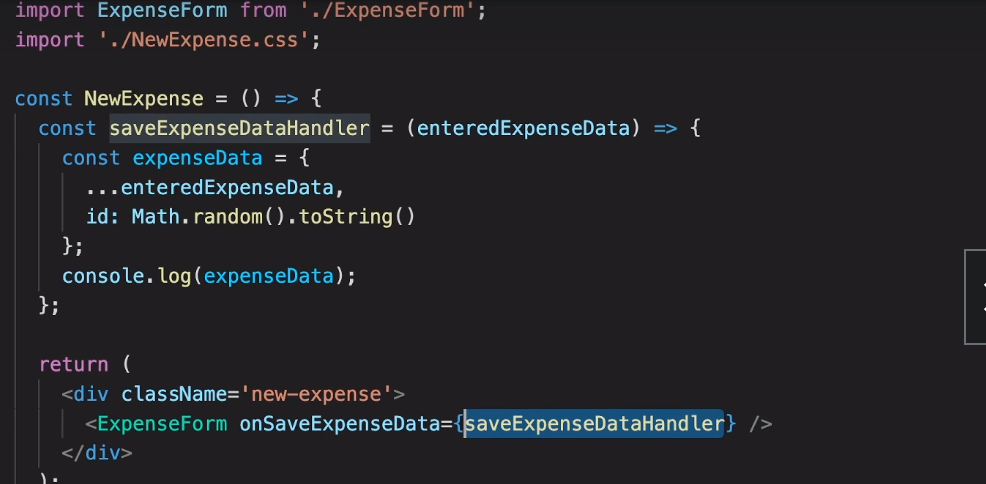
Add value properties to form input values.



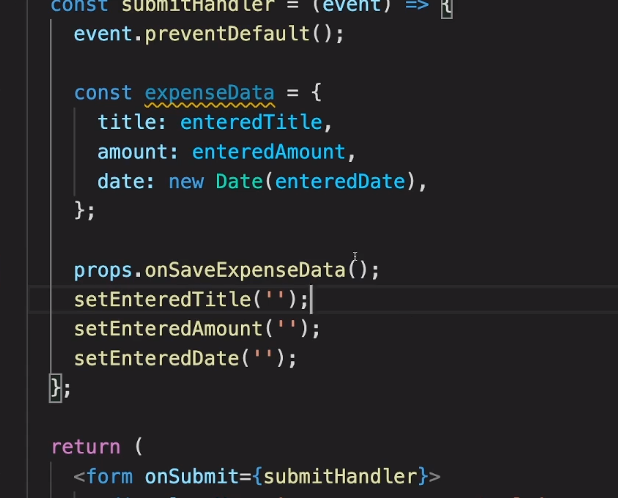
Bind the values to value property via curly braces. For example value = {enteredTitle} and later set them to empty string.

Child to parent communication in React.Props can only be passed from parent to child. We cannot skip intermediate components. In the below scenario , I want to pass function from

NewExpense (parent) to Expenseform (child)

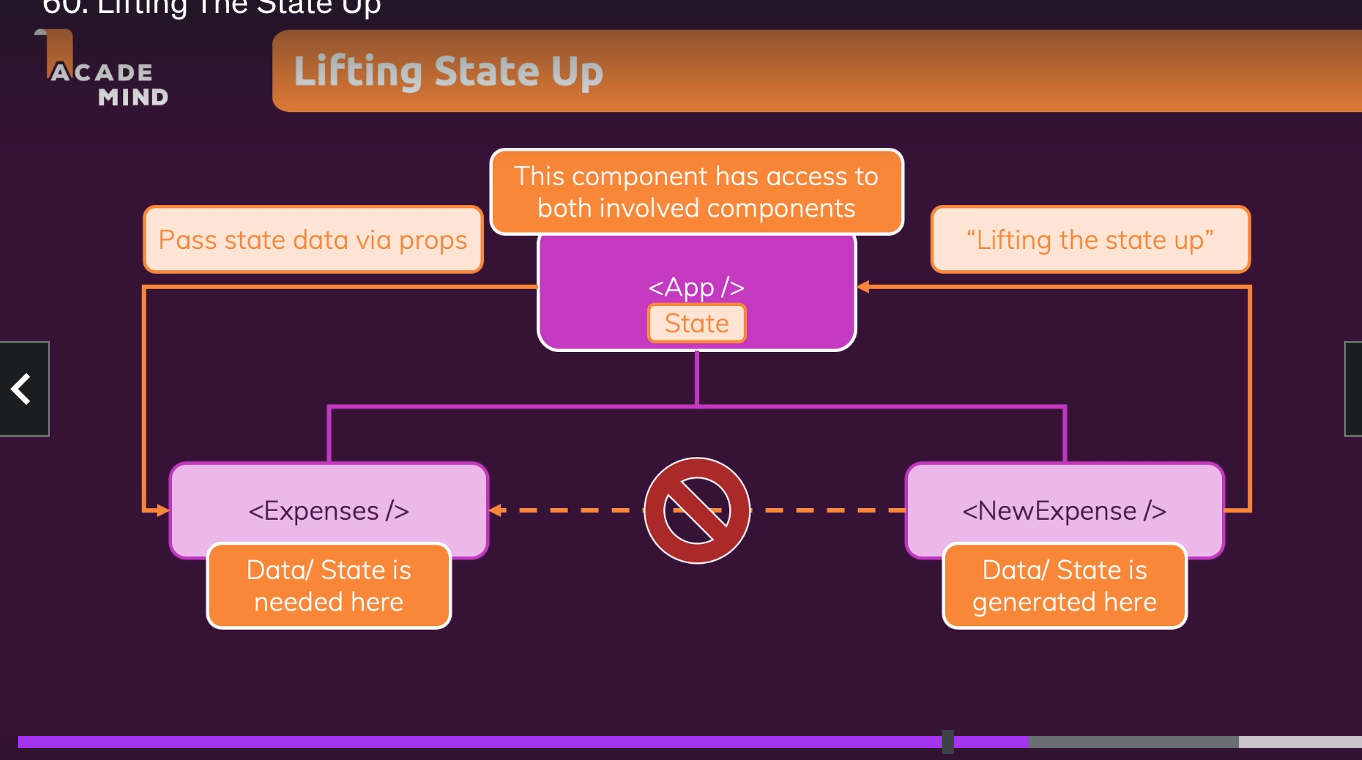


The method can be later called inside ExpenseForm component as below:-



Whenever we pass data from child to parent we call it “lifting the state up”.

This can be done as mentioned above.



State can be lifted as high as required just enough to share the data to other component.

Controlled vs uncontrolled Components & stateful Vs StateLess components:-

Controlled component:- A particular component is simply responsible for calling a function of the parent component, then that component is called as controlled component. Or a component from which we are lifting the state up to the parent component is called as parent component.

Stateful components:- Components which manage state for example wherever we use useState(). They are stateful components. Components which do not manage any state are called as stateless or dumb components