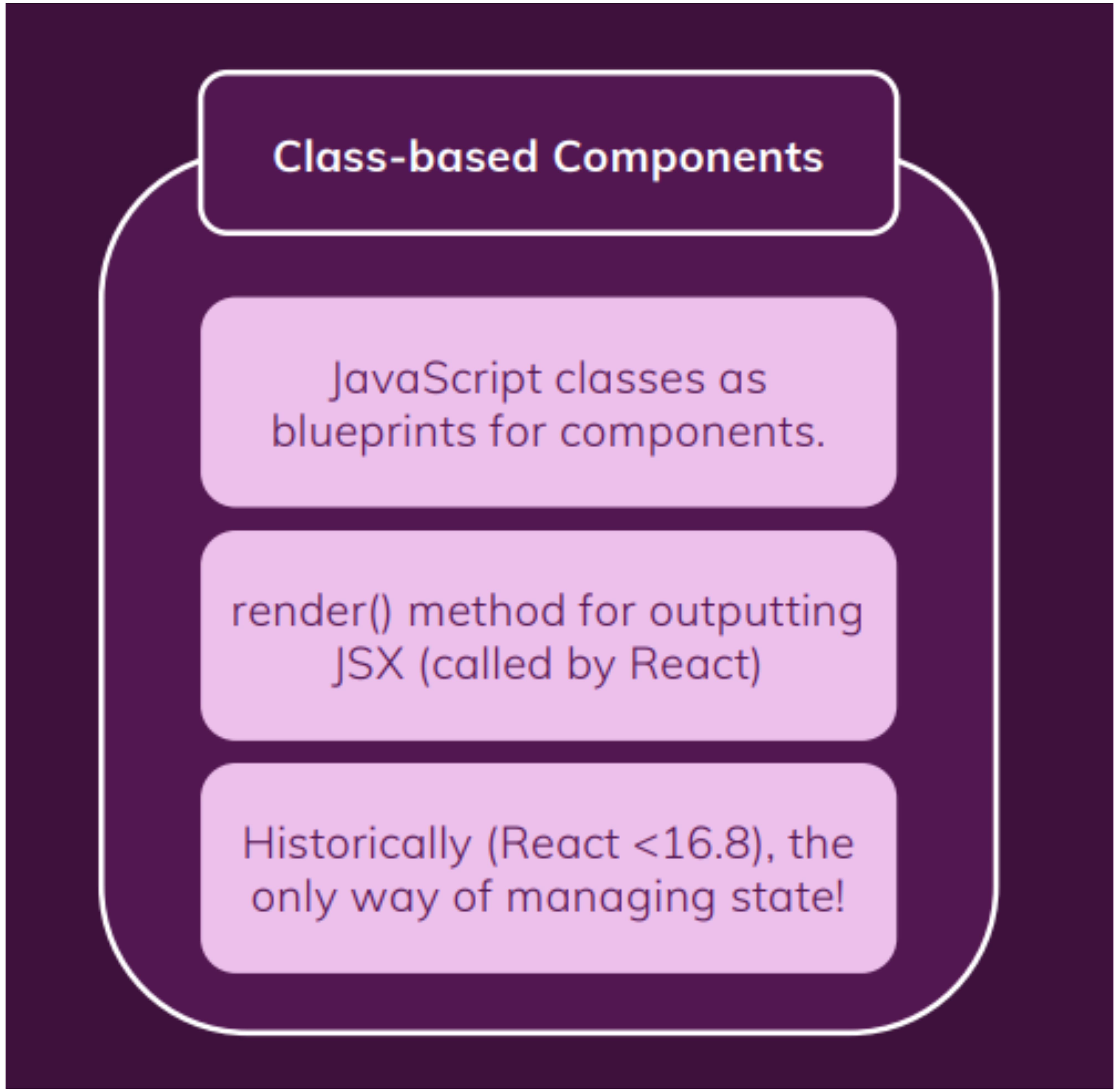
ANECO ACADEMY

REACTJS-CLASS COMPONENT

What is class component



Class-based Components

JavaScript classes as blueprints for components.

render() method for outputting JSX (called by React)

Historically (React <16.8), the only way of managing state!

Benefits of Class component

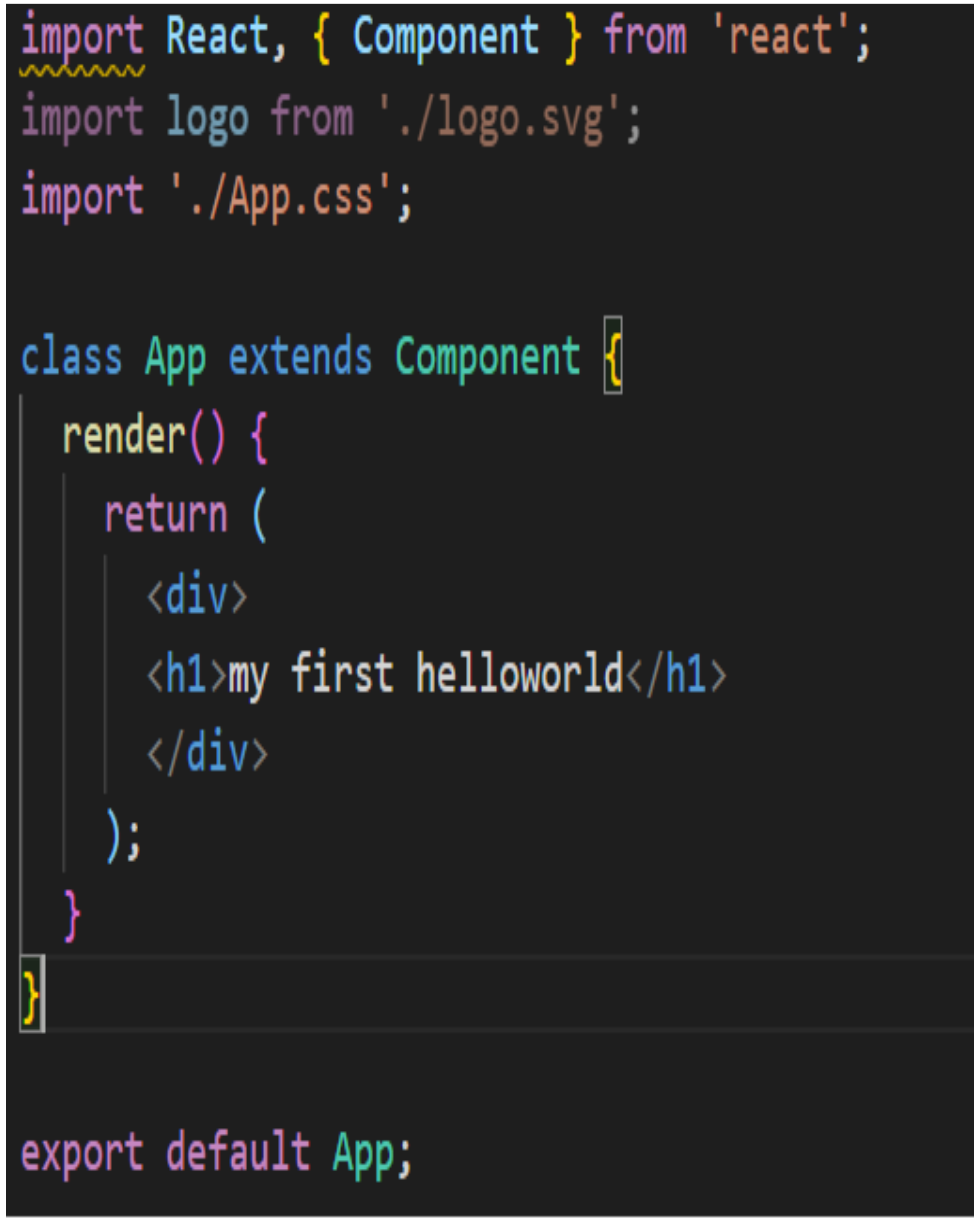
- Class components help in calling different methods at different time of DOM manipulation using lifecycle hooks
- Class components help us to manage state

Class component

Class components need to extends to Component class of react

Class component should consists of render()method inside the class

Inside class component constructor() method is optional

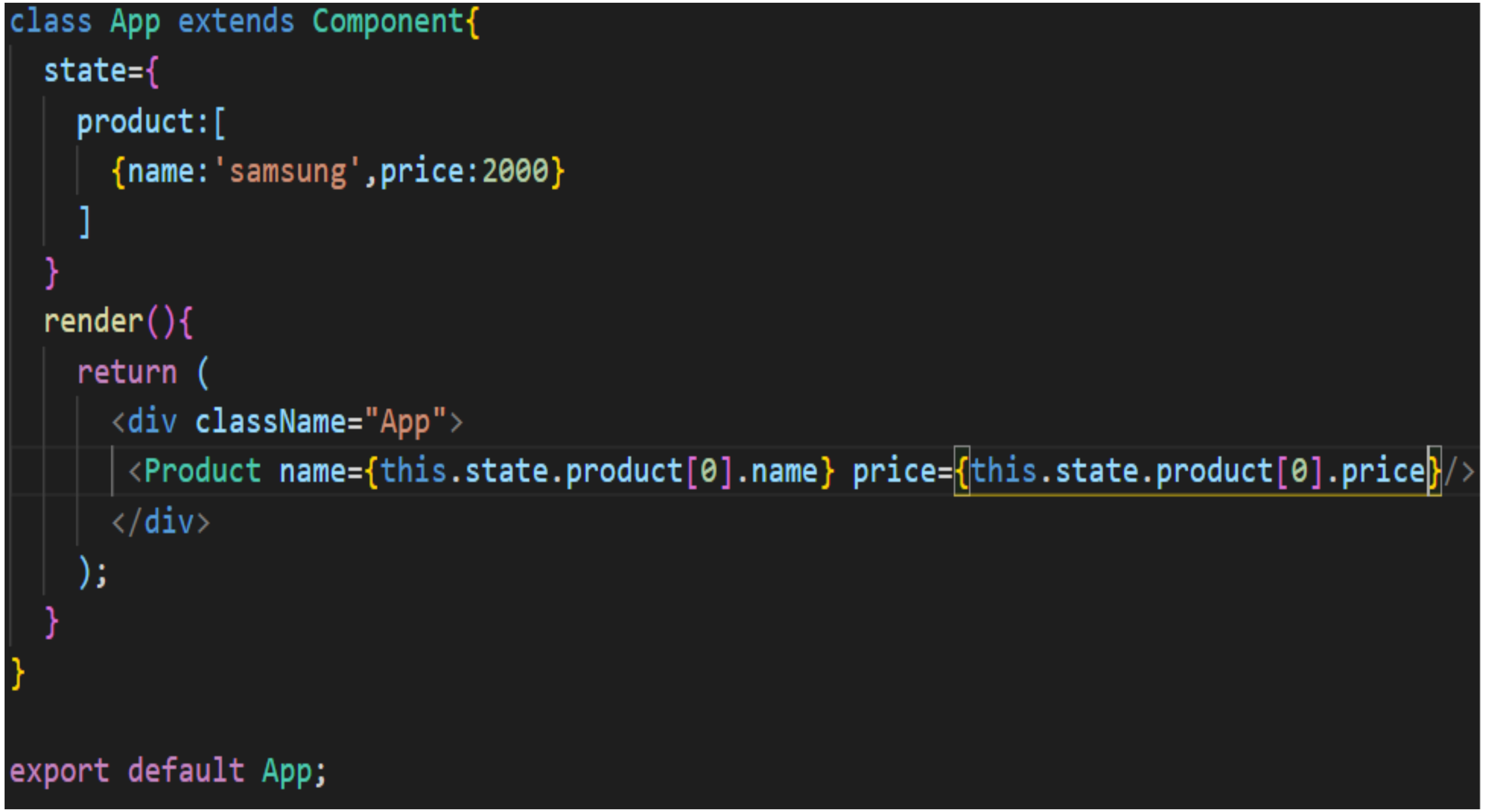


```
import React, { Component } from 'react';
import logo from './logo.svg';
import './App.css';

class App extends Component {
  render() {
    return (
      <div>
      <h1>my first helloworld</h1>
      </div>
    );
  }
}


export default App;
```

State in component class



```
class App extends Component{
  state={
    product:[
      {name:'samsung',price:2000}
    ]
  }
  render(){
    return (
      <div className="App">
       <Product name={this.state.product[0].name} price={this.state.product[0].price}/>
      </div>
    );
  }
}

export default App;
```

STATE IN CLASS

State can be declare in two ways:

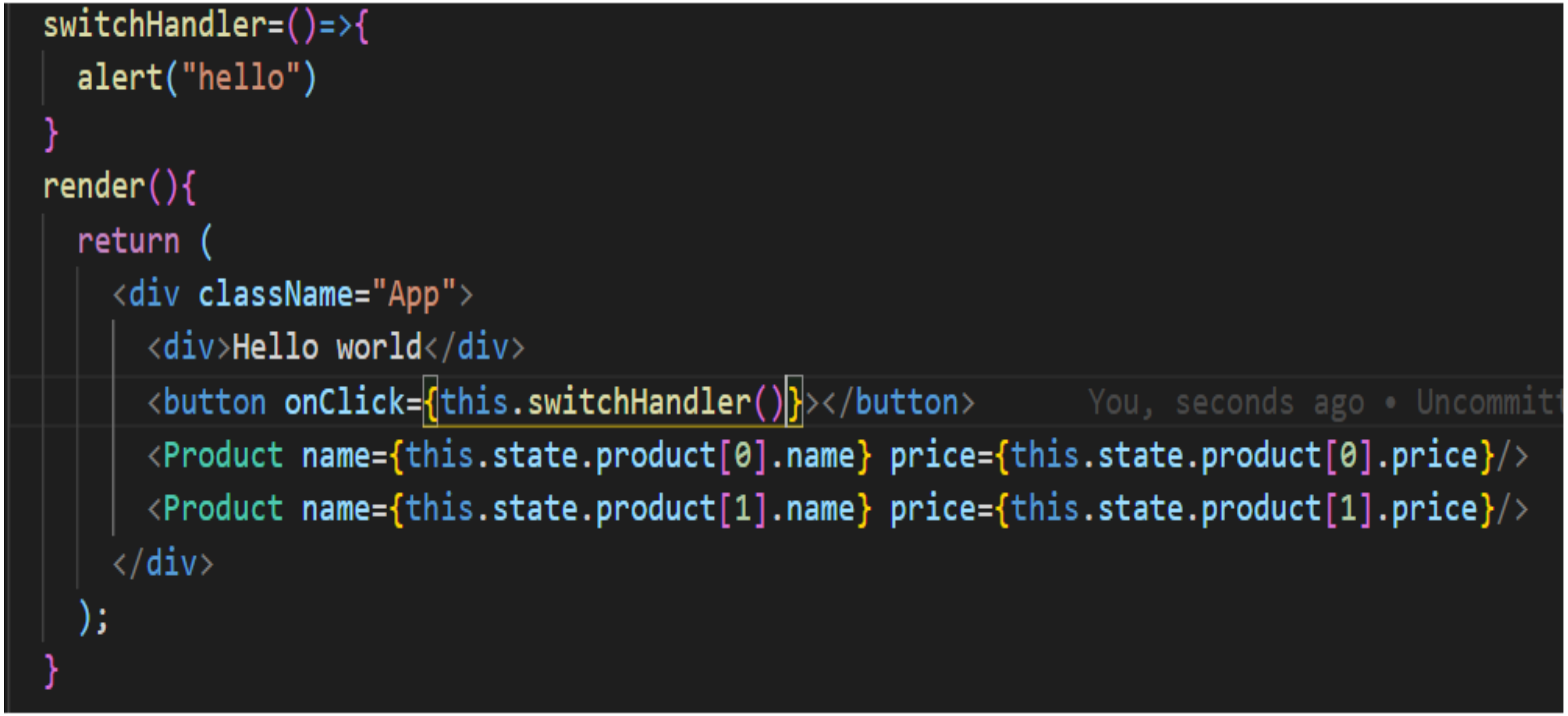
- State can be declare inside the constructor
- State can be declare outside the constructor if constructor is optional
- Whenever you declare state in class component always use keyword “state”
- Whenever you want to access the state use “this.state”

Inside the constructor declaring the state:

Example:

```
constructor(){
this.state={
name:”salman”,
Address:”chennai”
}
}
```

Handling events using state



```
switchHandler=()=>{
  alert("hello")
}
render(){
  return (
    <div className="App">
      <div>Hello world</div>
      <button onClick={this.switchHandler()}></button>       You, seconds ago • Uncommitt
      <Product name={this.state.product[0].name} price={this.state.product[0].price}/>
      <Product name={this.state.product[1].name} price={this.state.product[1].price}/>
    </div>
  );
}
```

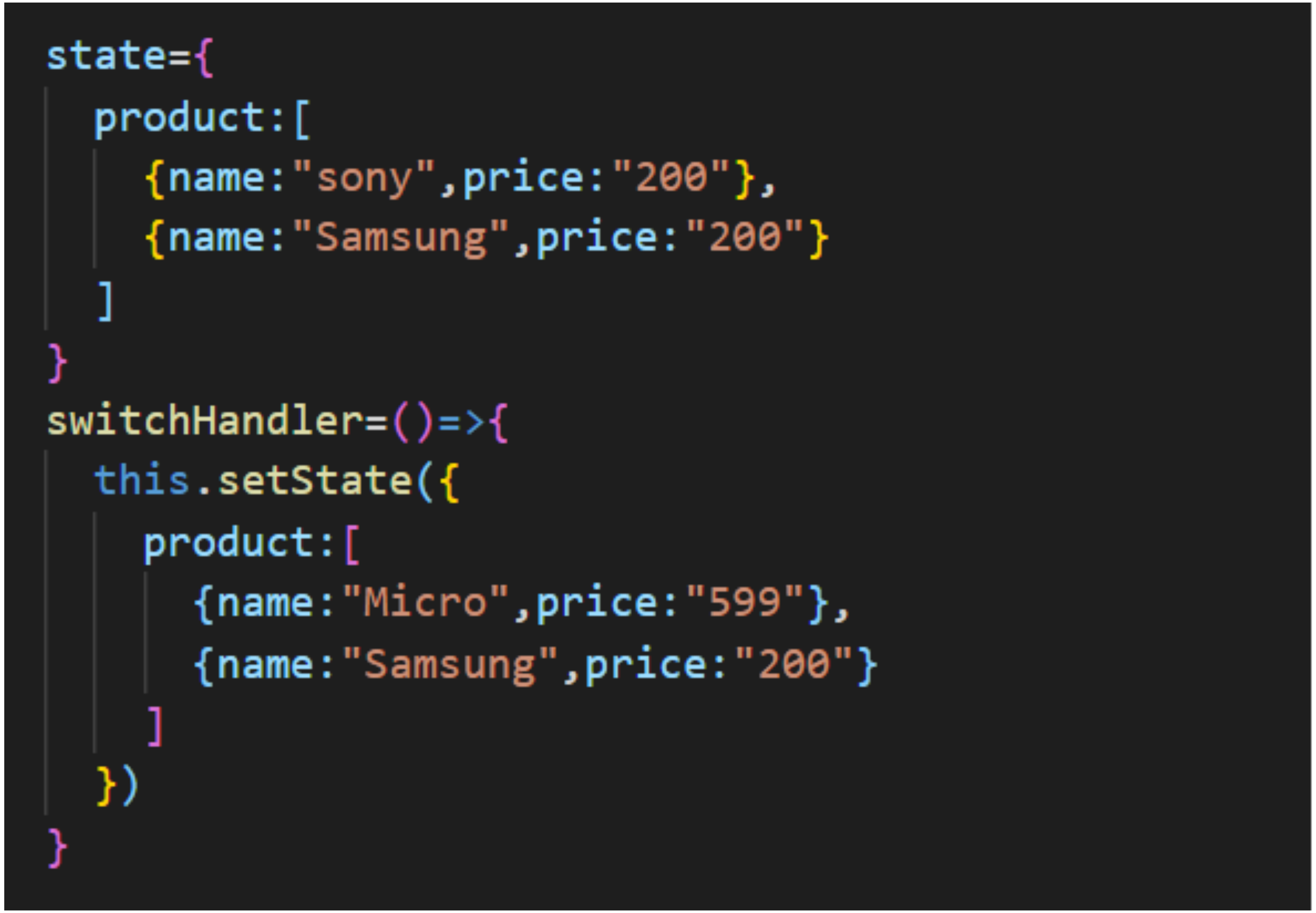
Events handling

Events can be called from JSX using the this keyword with method name

Please don't use method name with parenthesis inside the JSX because method will be called immediately

Always use arrow function to write methods inside the class components

Manipulation of state



```
state={
  product:[
    {name:"sony",price:"200"},
    {name:"Samsung",price:"200"}
  ]
}
switchHandler=()=>{
  this.setState({
    product:[
      {name:"Micro",price:"599"},
      {name:"Samsung",price:"200"}
    ]
  })
}
```

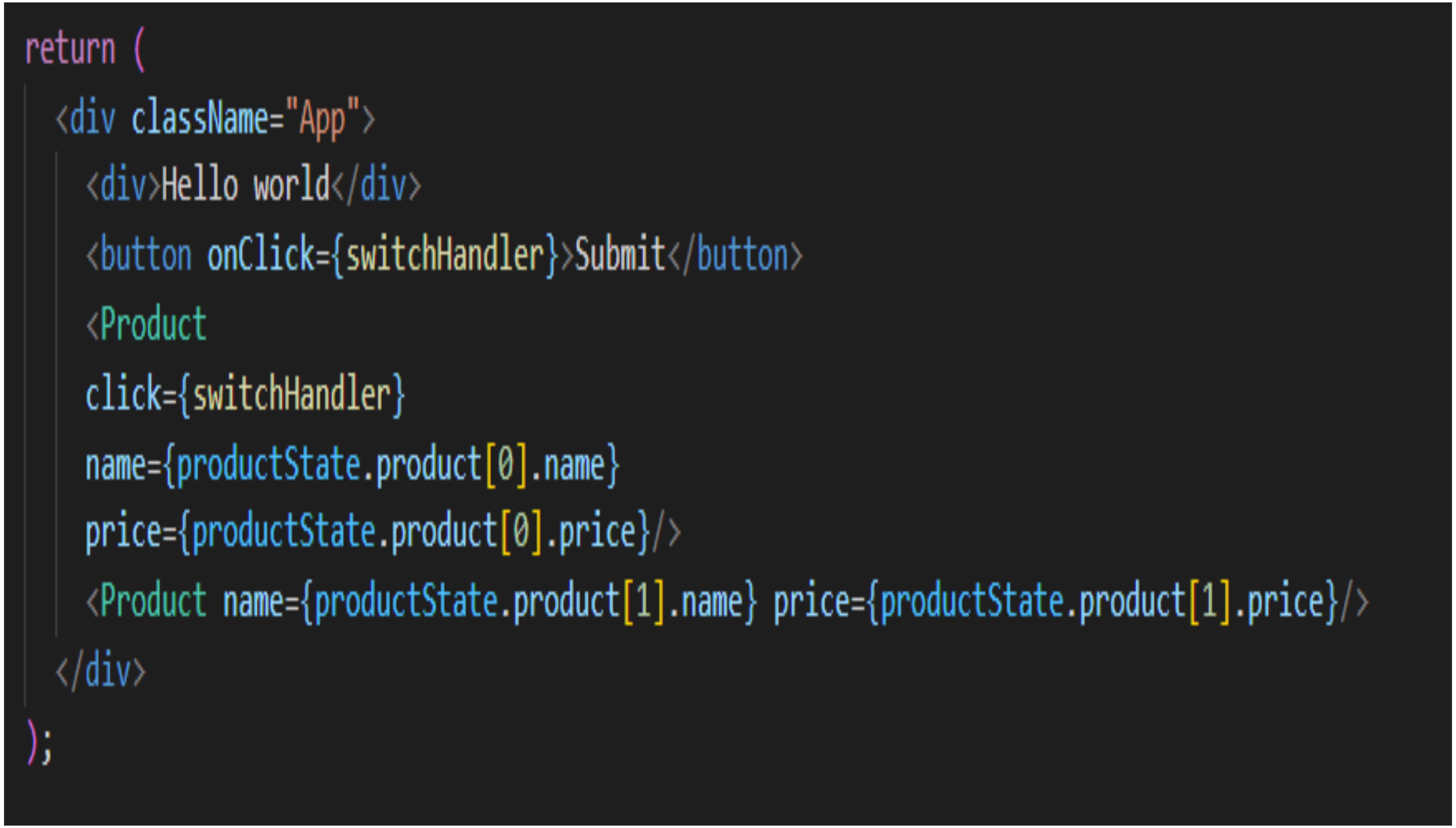
Manipulation of state

Changing state in class component always need setState() property from react components.

Inside the setState() which ever property you want to update then that particular data you can update it.

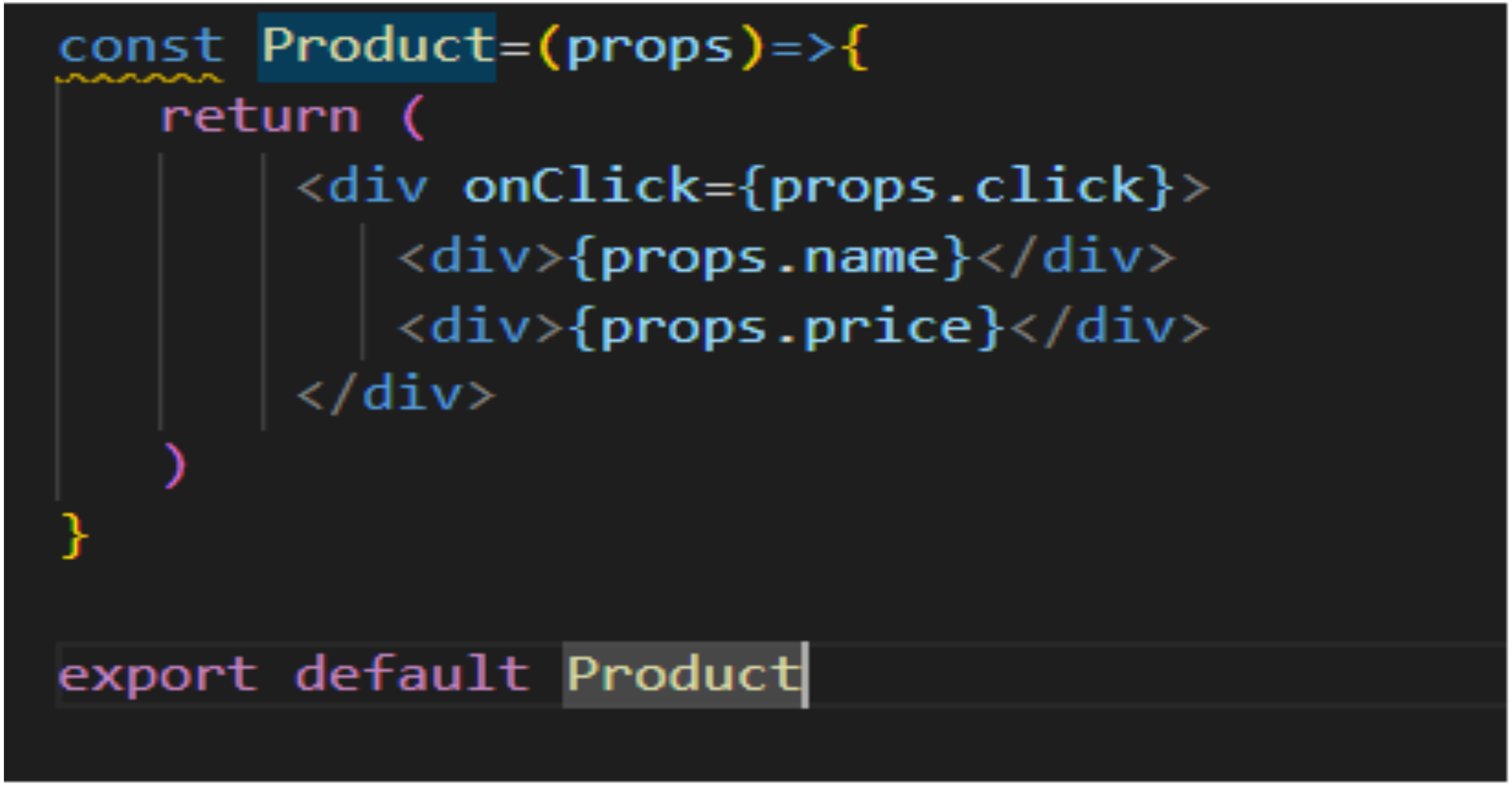
Other state data will remain as it is there will no change in data.

Passing method reference-1



```
return (
  <div className="App">
    <div>Hello world</div>
    <button onClick={switchHandler}>Submit</button>
    <Product
    click={switchHandler}
    name={productState.product[0].name}
    price={productState.product[0].price}/>
    <Product name={productState.product[1].name} price={productState.product[1].price}/>
  </div>
);
```

Passing method reference-2



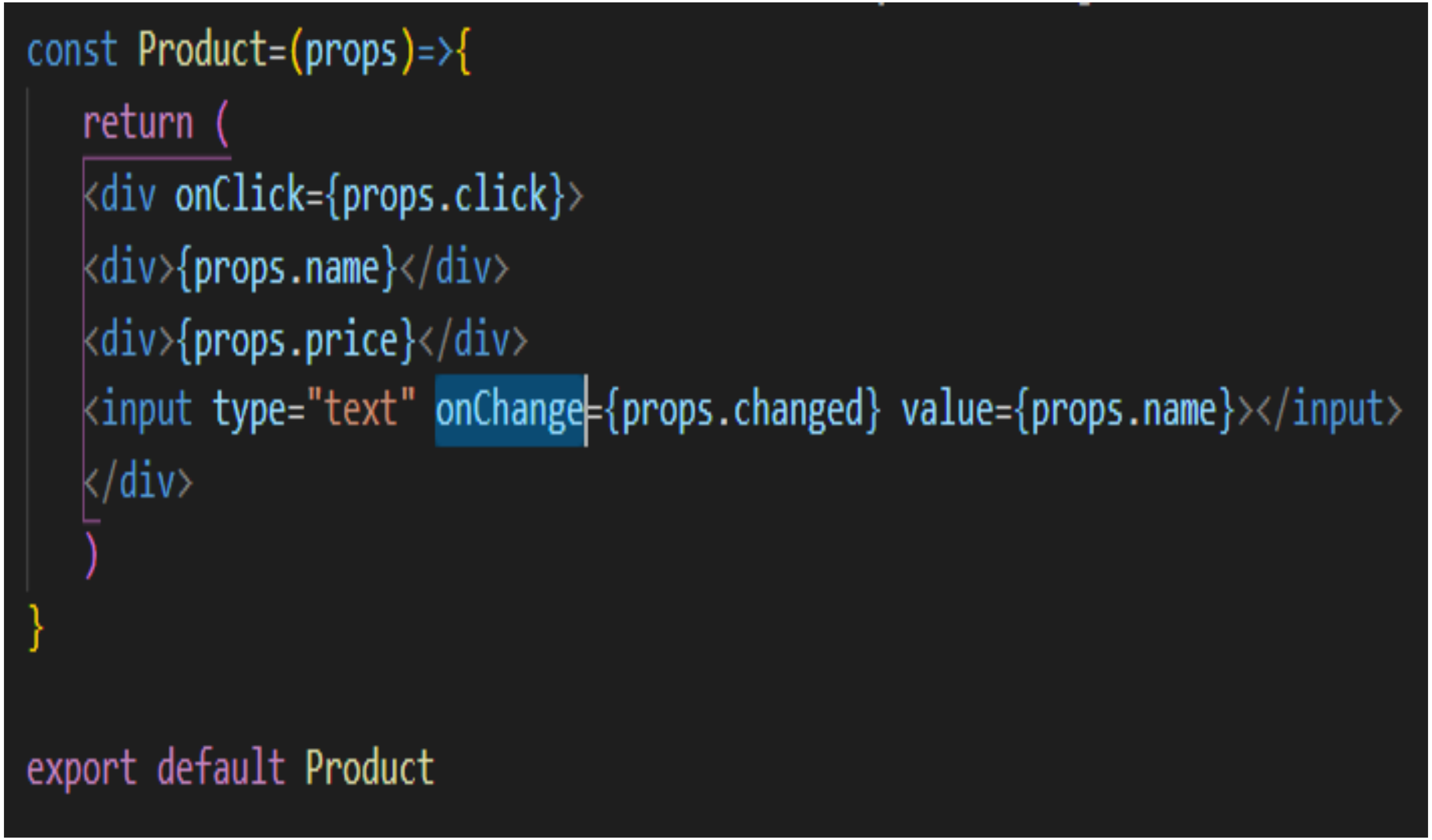
```
const Product=(props)=>{
   return (
       <div onClick={props.click}>
          <div>{props.name}</div>
          <div>{props.price}</div>
       </div>
   )
}

export default Product
```

Passing method reference-3

Calling method from child component to parent component while using method passing reference
Whenever events trigger from child component it will react to parent component via props.
Whatever methods is attached in props then particular method can be called.
We can pass a data from child to parent component using method reference
So flow of data will become bidirectional from parent to child and from child to parent component.

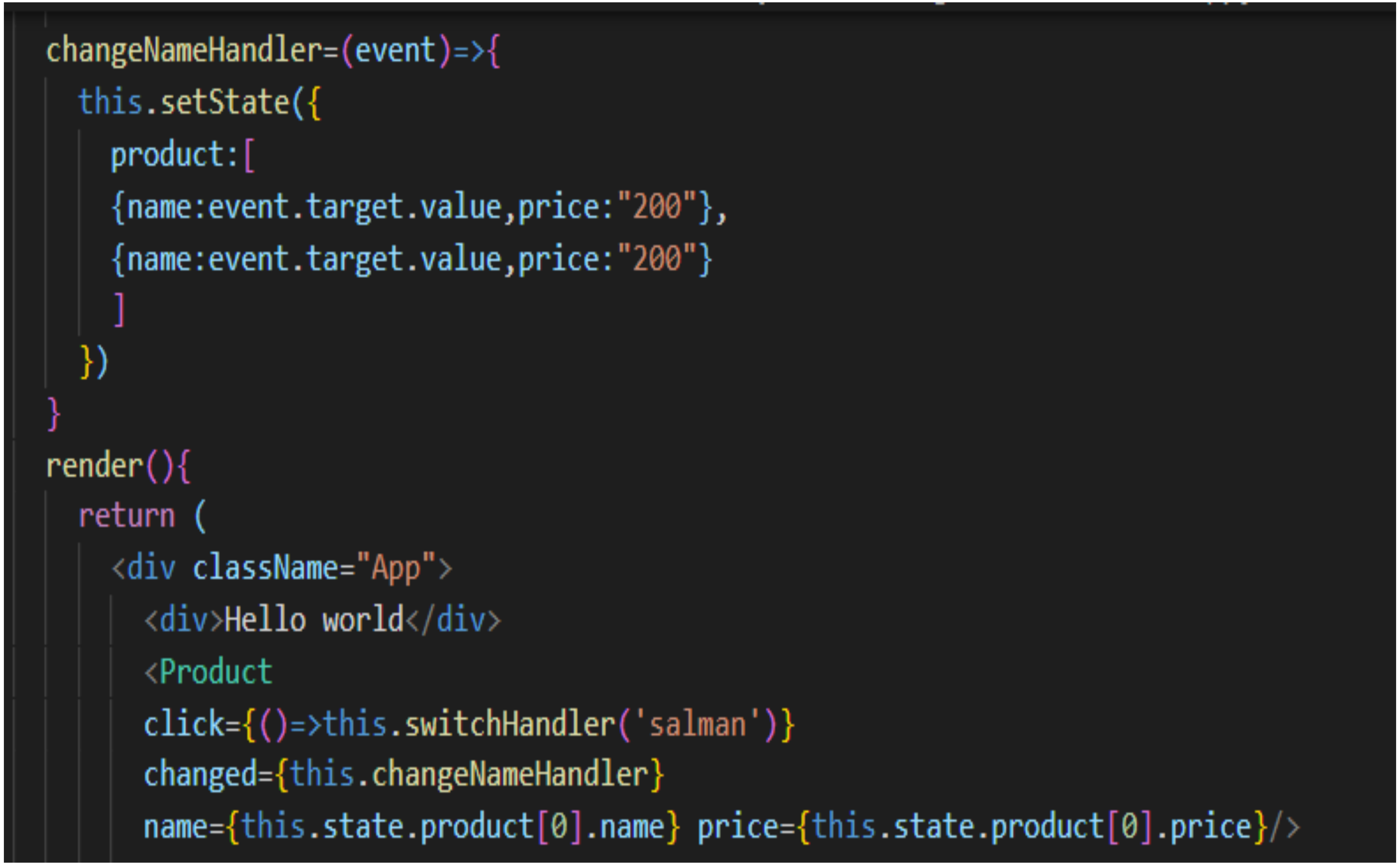
Two way binding in form - 1



```
const Product=(props)=>{
    return (
    <div onClick={props.click}>
    <div>{props.name}</div>
    <div>{props.price}</div>
    <input type="text" onChange={props.changed} value={props.name}></input>
    </div>
    )
}

export default Product
```

Two way binding form-2



```
changeNameHandler=(event)=>{
  this.setState({
    product:[
    {name:event.target.value,price:"200"},
    {name:event.target.value,price:"200"}
    ]
  })
}
render(){
  return (
    <div className="App">
      <div>Hello world</div>
      <Product
      click={()=>this.switchHandler('salman')}
      changed={this.changeNameHandler}
      name={this.state.product[0].name} price={this.state.product[0].price}/>
```

Two way binding form-3

For two way binding inside the form tag we have to use value property and onChange() events inside the input,textarea,radio,checkbox,select tags.

Value property helps us to binding the existing value and updating value from state

onChange events help us to call setState to change the value of the state.

Value property use curly braces{} to bind the value inside the JSX

onChange() call event method inside the curly braces{}don’t use circle braces inside the function call otherwise it will call immediately