Module CN5006
How to create React Class Components
Prepared by Dr.Nadeem Qazi

In this tutorial, we will develop a REACT class component that will count a number of times an emoji is clicked. We already used a functional component to develop this countrer, which is stateless component, this week we are learning a class component which keep records of the state of the component itself rather than using REACT HOOK API.

What is React Class Component:
Class components are more complex than functional components they include:
constructors

- render( ) function (**remember there is no render function involved when you did functional specification**)
- state (data) management. In this tutorial, we see how to code a class component.

Q: What is Constructor:
A constructor is a concept involved in object-oriented programming which creates an object. This is where we want to set up our state for the component

What is state of the class component:

- **State** of a component refers to an object that stores information which can change throughout the component's lifecycle. It is used for data that may alter and is reflected in the UI. For example, on Facebook, every time a user clicks the "like" button, the number of likes increases, meaning the state of the button has changed by adding one more like.

- A key distinction between **props** and **state** is that props are **immutable**—once set, they cannot be changed—whereas state is a mutable object designed to store data that can change over time and control the component's behavior based on those changes.

- While **props** are passed from a parent component, **state** is typically modified by event handlers within the component itself.

Enough theory lets create a component(last week you created the same component using the functional component) that have one one button and one image . it have following specifications:

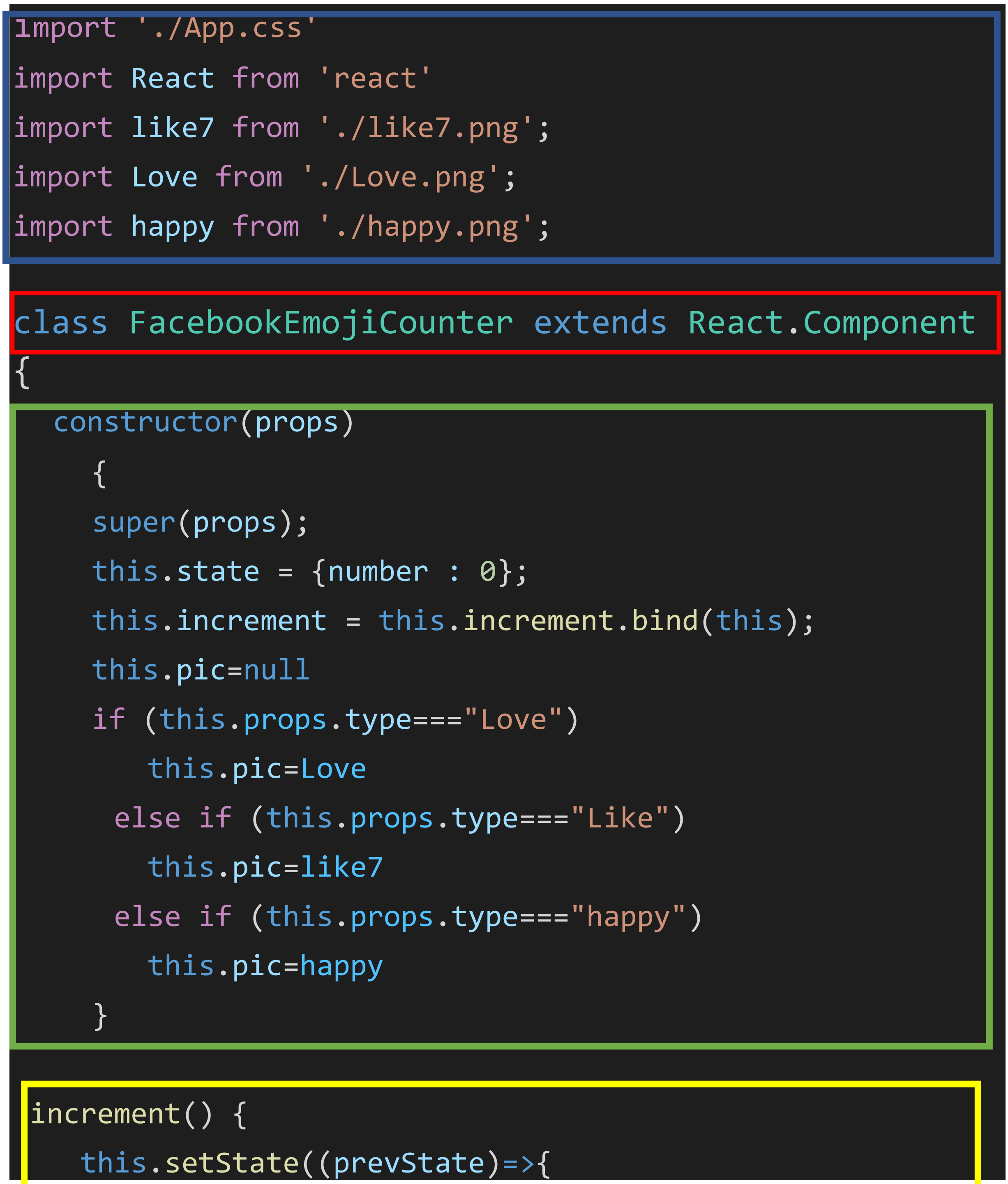
1. A property to change the type of the button i.e. like, love and happy
2. A state that maintains how many times a number is clicked
3. Images for three different types.
4. An event handler that will active when the button is clicked.

To achieve all of these specification we will define a class that will create this component. Lets Say the name of this Component is FacebookEmojis

Task1 : Creating a class component to record Facebook emotions such as like, love and happy using Class Component

Create a newfile named it as Facebookemoji.js

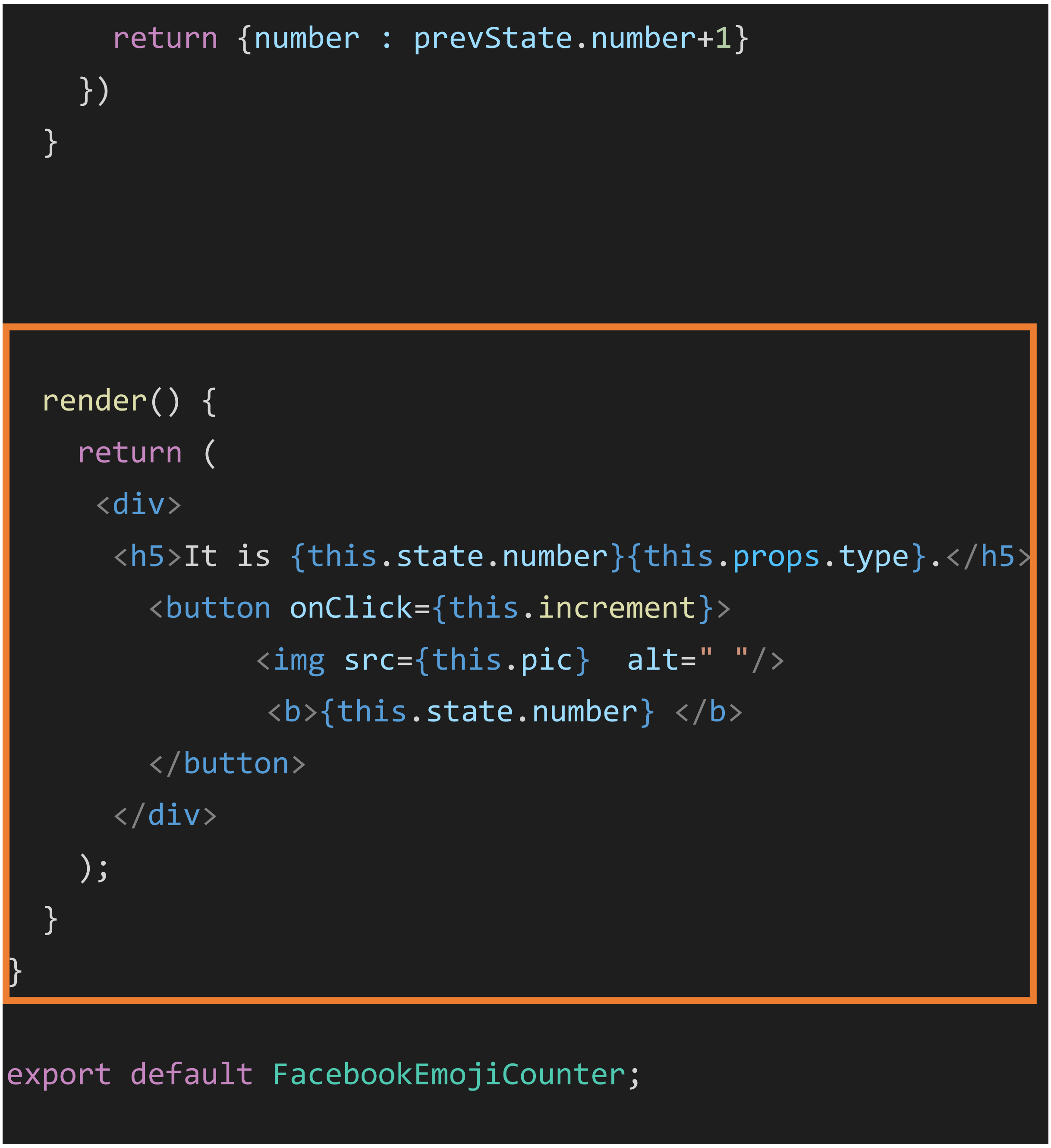
For this, we will create a new js file and name it Facebookemoji.js. Write the following code in this file: as Facebookemoji.js.



```
import './App.css'
import React from 'react'
import like7 from './like7.png';
import Love from './Love.png';
import happy from './happy.png';

class FacebookEmojiCounter extends React.Component
{
  constructor(props)
    {
    super(props);
    this.state = {number : 0};
    this.increment = this.increment.bind(this);
    this.pic=null
    if (this.props.type==="Love")
        this.pic=Love
      else if (this.props.type==="Like")
        this.pic=like7
      else if (this.props.type==="happy")
        this.pic=happy
    }

 increment() {
    this.setState((prevState)=>{
```



```
      return {number : prevState.number+1}
    })
  }



  render() {
    return (
     <div>
      <h5>It is {this.state.number}{this.props.type}.</h5>
        <button onClick={this.increment}>
              <img src={this.pic}  alt=" "/>
               <b>{this.state.number} </b>
        </button>
      </div>
    );
  }
}

export default FacebookEmojiCounter;
```

In my as Facebookemoji.js. we have created a class named FacebookEmojiCounter Once we create this class we will also export it so that it can be used outside this file. Note that the code of the class is divided into several sections. The first section marked as blue is always used for importing necessary React classes and other files used in the project. In our case, we have imported the following classes

- Line 1: import ‘./App.css’   is included because we want to use app.css in our new file for some decoration.

- Line 3,4,5 we import three png files which are provided with this turotial. Upload these files in the src directory
- **Class Definition:**
- Line6: we define a class named as FacebookEmojiCounter derived from () React.Component using the keyword extends
- It is our React Element that will be used in index.js This class contains all the logic of our components. It defines three functions :
  - **Constructor: which initializes our component when it is created**
  - **Increment: this function defined outside the constructor is used to increment a variable .**
  - **Render function: A must function for every component class withour**

Contructor function:

- **inside this class , the Line 7 calls the constructor for this class . it initializes the state of this class component to zero with the following command this.state = {number : 0}; Please note the name of state variable is number. You can give it any name.**
- **Line 8 we bind a function named as increment to this class.. Using the line this.increment = this.increment.bind(this); This function will be activated whenever the button is clicked**
- **Then we read the property object props , Note that in class component we do nt need to pass the props object as we did in functional component. The name of the property we are reading here is type which will be set in index.js when we create this element. This property is used to set the mage to be displayed in the button. If the user for example set the type property =”Love” the component will display the Love icon. The code used for this purpose given below. Note how the property is compared we have used === compare and equal**

operator of java script . it checks the value being compare along with the type of the value

```
    if (this.props.type==="Love")
      this.pic=Love
     else if (this.props.type==="Like")
      this.pic=like7
     else if (this.props.type==="happy")
      this.pic=happy
 }
```

Function increment :

This function defines the logic of our component. This function will be associated with the click event of the button. So that whenever the button is click this function will active. It like an event handler for the click event of the button. In this function we set the state of this component using the command this.setState and in doing so we first read what was the previous state through the variable prevState which holds the variable number and then we increment that variable and save this as state. Remember we cannot directly set the state of the component we must use this.setState to set the value of the state.

Then in we define the function increment s as:

```
increment() {
   this.setState((prevState)=>{
     return {number : prevState.number+1}
   })
 }
```

Render function :

This function must **be included in every class component** . In our case this functions draw a button on the web page using the HTML tag <button> it however also bind the function increment() .

function increment to its onclick event using onClick={this.increment} Note we are using {} to use java script in htlm. On the caption of the button it displays the state property. Following code is used

```
      <button onClick={this.increment}
          >
          <img src={this.pic}

          alt=" "/>
          <b>{this.state.number} </b>
    </button>
```

Calling the Component in the index.js

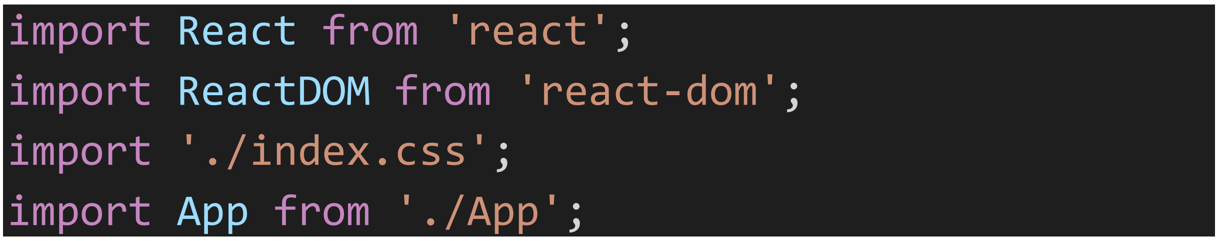
Next, In index.js include the element you have created through the line .Note this part is almost similar to calling the functional component.

```
import FacebookEmojiCounter from './Facebookemoji'
```

then clear the tag <App/> and write these lines

```
<React.Fragment>
   <FacebookEmojiCounter  type= "Like"/>
   < FacebookEmojiCounter type= "Love"/>
   < FacebookEmojiCounter type= "happy"/>
    </React.Fragment>,
```

Index.js



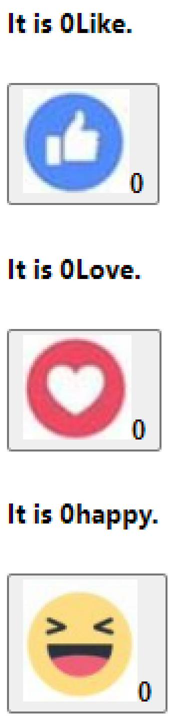
```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import FacebookEmojiCounter from './Facebookemoji
'
import * as serviceWorker from './serviceWorker';

ReactDOM.render(

   <React.Fragment>
   <FacebookEmojiCounter  type= "Like"/>
   < FacebookEmojiCounter type= "Love"/>
   < FacebookEmojiCounter type= "happy"/>
    </React.Fragment>,
  document.getElementById('root'));

// If you want your app to work offline and load
faster, you can change
// unregister() to register() below. Note this co
mes with some pitfalls.
// Learn more about service workers: https://bit.
ly/CRA-PWA
serviceWorker.unregister();
```

Save it and you will see following change in output . if you do nt see it either run it again or refresh it.



It is 0Like.

It is 0Love.

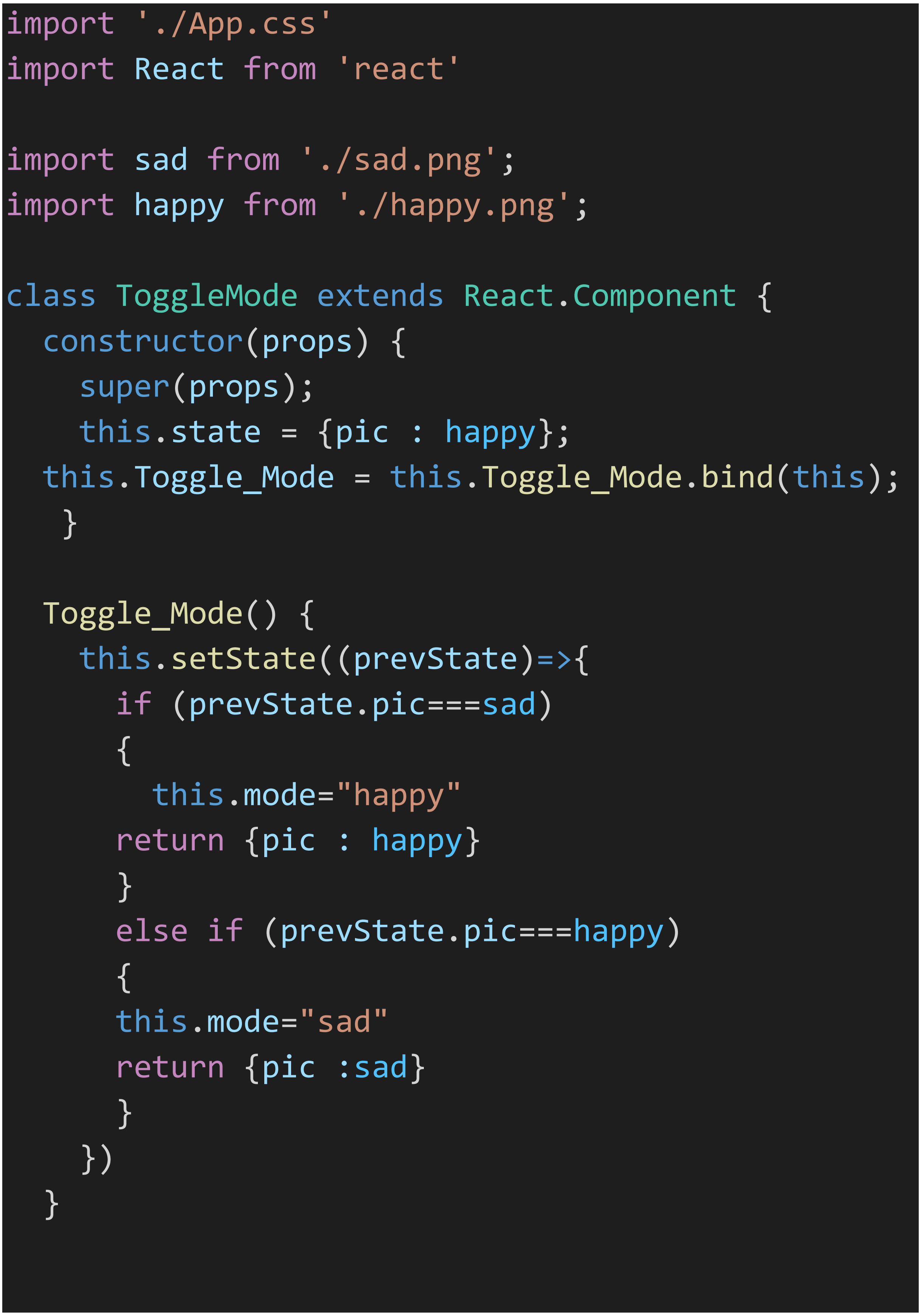
It is 0happy.

Click each of these buttons and observe what is happening, explain this when you submit your lab.

Task2: Create a class component that **<u>toggles the emoji</u>** on the button .that if the emoji is sad and the button is clicked then it turns to happy and vice versa. This task would be created similarly as we created the last component. We will define a separate class name ToggleModeComponent.js. it will have three functions as before:

1. Constructor (because we want to set the initial image on the button)
2. An event handler for the click event of the button: this time we name this as Toggle_Mode. It will be associated with the click event of the button and will toggle the picture every click event.
3. Render function.

Step 1: create a file called as ToggleModeComponent.js and add following code .



```
import './App.css'
import React from 'react'

import sad from './sad.png';
import happy from './happy.png';

class ToggleMode extends React.Component {
  constructor(props) {
    super(props);
    this.state = {pic : happy};
  this.Toggle_Mode = this.Toggle_Mode.bind(this);
   }

  Toggle_Mode() {
    this.setState((prevState)=>{
      if (prevState.pic===sad)
      {
        this.mode="happy"
      return {pic : happy}
      }
      else if (prevState.pic===happy)
      {
      this.mode="sad"
      return {pic :sad}
      }
    })
  }
  render() {
    return (

      <div>

         <h3>This is output of Task2: {this.mode}
</h3>
        <button onClick={this.Toggle_Mode}
        >
        <img src={this.state.pic}

        alt=" "/>

        </button>

      </div>
    );
  }
}
export default ToggleMode;
```

in this code we use pic property to set the image on the button. This pic property is set as state of the component. It is initialized in contrsutor using the command this.state = {pic : happy}; where happy is the image imported using the line import happy from './happy.png'; however you have to upload the happy.png file in repl.it The happy.png file is provided to you with this tutorial upload it in repl.it through clicking … three dots icon on the repl.it once it is uploaded move it in to Src directory

Note the logic of the toggle_Mode function, this time we do nt increment the number because we want to set the image on the button each time it is clicked . So we compare what is already displayed on the button using the prevState using following code

```
this.setState((prevState)=>{
if (prevState.pic===sad)
{
this.mode="happy"
return {pic : happy}
}
else if (prevState.pic===happy)
{
this.mode="sad"
return {pic :sad}
}
})
}
```

Once it is saved then open the index.js file and add following line :

import ToggleMode from './ToggleModeComponent'

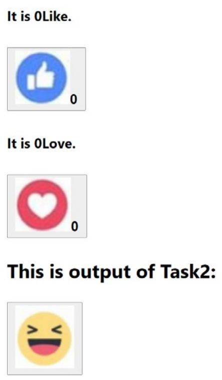
then add this line :

< ToggleMode/> to create the element from the class in the index.js. Your final code in index.js should look like this:

```
ReactDOM.render(

   <React.Fragment>
   <FacebookEmojiCounter  type= "Like"/>
   < FacebookEmojiCounter type= "Love"/>
   < ToggleMode/>
    </React.Fragment>,
  document.getElementById('root'));
```

run it if it is not automatically run. You will following



It is 0Like.

0

It is 0Love.

0

This is output of Task2:

Click the button and observe the behaviour.

Wee9 Portfolio

Complete the exercise and include the code link in your portfolio. In your reflection, discuss the differences between implementing the code in a functional component versus a class component.