## Working with Class Components:

- Class components are more complex than functional components.
- To define a React component class, we have to create a class and extend **React.Component** class.
- Output of any Class Component we create is dependent on the return value of a Method Called **render().**
- The render() method is the only required method needs to be implemented in a class component.

Eg 1:

hello-app\src\index.js

```
import React from 'react'
import ReactDOM from 'react-dom'
import './index.css'

class Employee extends React.Component {
    render() {
        return <div>
            <p>Id : {this.props.id}</p>
            <p>Name : {this.props.name}</p>
            <p>Salary : {this.props.salary}</p>
        </div>;
    }
}


const element = <Employee id="E101" name="Jones" salary="5500"/>;

ReactDOM.render(element, document.getElementById('root'))
```

**Note:**

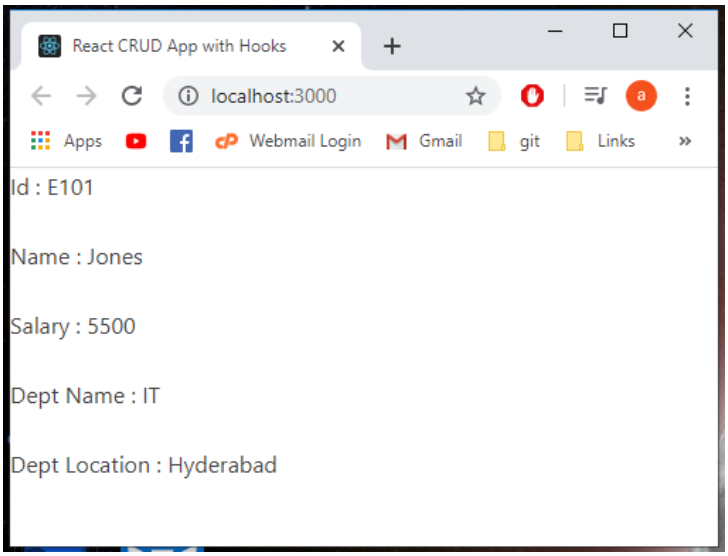
**Props** are Read-Only

Whether we declare a component as a function or a class, it must never modify its own props.

Eg 2:

hello-app\src\index.js

```
import React from 'react'
import ReactDOM from 'react-dom'
import './index.css'

class Department extends React.Component {
    render() {
        return <div>
            <p>Dept Name : {this.props.dept}</p>
            <p>Dept Location : {this.props.location}</p>
        </div>;
    }
}

class Employee extends React.Component {
    render() {
        return <div>
            <p>Id : {this.props.id}</p>
            <p>Name : {this.props.name}</p>
            <p>Salary : {this.props.salary}</p>
            <Department dept={this.props.dept} location={this.props.location}/>
        </div>;
    }
}


let element = <Employee id="E101" name="Jones" salary="5500" dept="IT" location="Hyderabad
" />;

ReactDOM.render(element, document.getElementById('root'))
```



**Function vs Class Components:**
If we are expecting features like

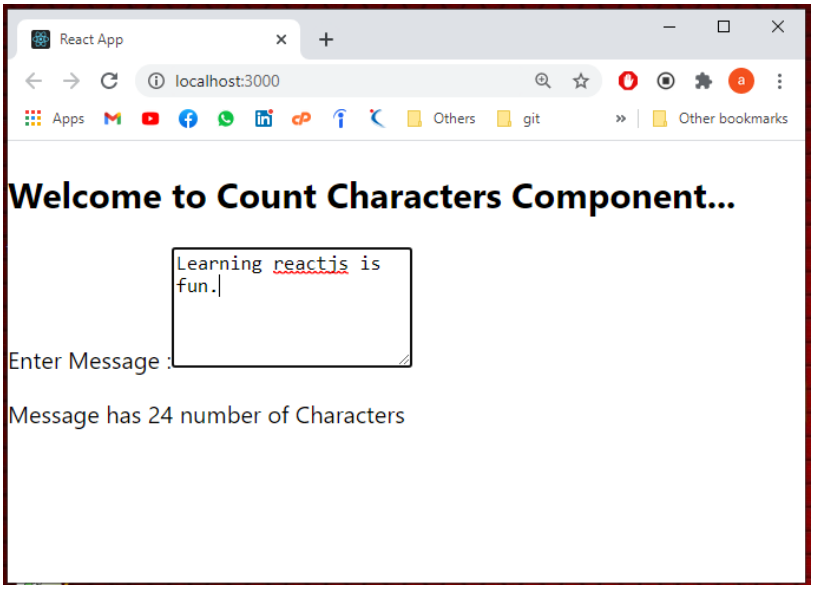
1. Managing **State** of the Components
2. Adding **Life Cycle Methods** to Components
3. Need to Write Logic for **Event Handlers**

Then we will go for **Class Component** and in rest of the cases we can go for Function Component.

## Working with state

- React introduces, a new concept named “state” which allows React components to change their output over time in response to user actions without violating this rule.
- State is similar to props, but it is private and fully controlled by the component.
- State contains data specific to a given component that may change over time.
- The state is user defined plain javascript object.
- When the state object changes, the component re-renders.

**Use Case:**



Example:

```
import React from 'react'
import ReactDOM from 'react-dom'
import './index.css'

class CountCharacters extends React.Component{
    constructor(props){
        super(props);
        this.state={
            message:'',
            counter:0
        };
    }

    onMessageChange(text){
        this.setState({
            message:'Message has '+text.length+' number of Characters',
            counter: this.state.counter+1
        });
    }

    render(){
        return <div>
            <h2>Welcome to Count Characters Component...</h2>
            <p>
                Enter Message :
                <textarea type="text" rows="5"
                            onChange={e=>this.onMessageChange(e.target.value)} />
            </p>
            <p>
                <label>{this.state.message}</label>
            </p>
            <p>
                <label>{this.state.counter}</label>
            </p>
        </div>
    }
}

const element=<CountCharacters></CountCharacters>

ReactDOM.render(element,document.getElementById("root"));
```

## Interaction between Components in React

- The UI of every React application we develop, gets broken down into Components.
- Every react application we develop will be comprising of multiple components.
- There will be one Root Component and this component can have one or more Child Components in it.
- And this nesting can go further as the Application UI gets developed.

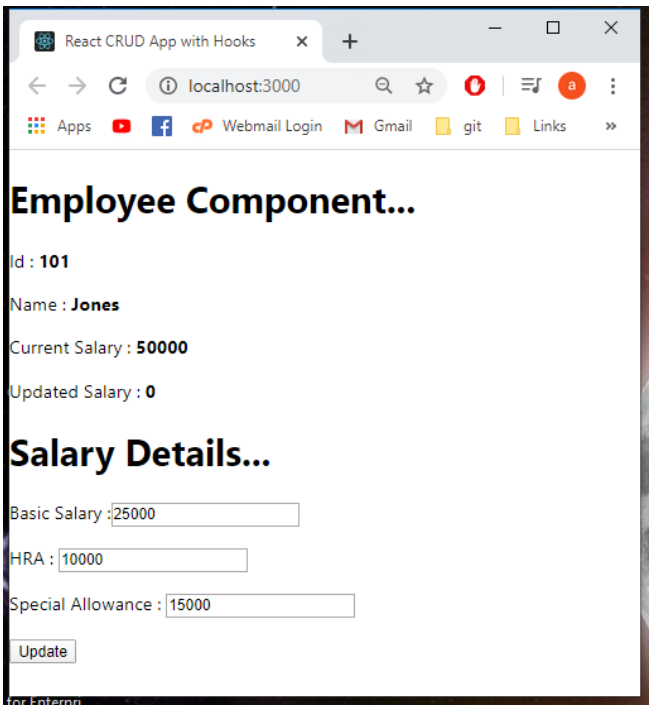
Eg:

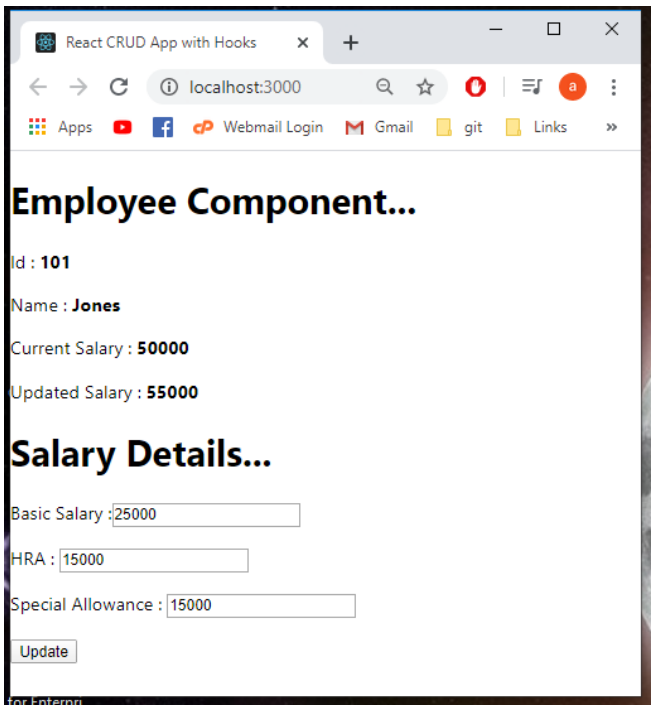
hello-app\src\index.js

Employee Component = Employee Personal Info + Salary Details

```
import React from 'react'
import ReactDOM from 'react-dom'
import './index.css'

class Employee extends React.Component {
    constructor(props) {
        super(props);
        this.state = {
            updatedSalary: 0
        };
    }

    getUpdatedSalary = (salary) => {
        this.setState({ updatedSalary: salary });
    }

    render() {
        return <div>
            <h1>Employee Component...</h1>
            <p>
                <label>Id : <b>{this.props.Id}</b></label>
            </p>
            <p>
                <label>Name : <b>{this.props.Name}</b></label>
            </p>
            <p>
                <label>Current Salary : <b>{this.props.Salary}</b></label>
            </p>
            <p>
                <label>Updated Salary : <b>{this.state.updatedSalary}</b></label>
            </p>
            <Salary BasicSalary={this.props.BasicSalary} HRA={this.props.HRA} SpecialAllow
ance={this.props.SpecialAllowance} onSalaryChanged={this.getUpdatedSalary}></Salary>
        </div>
    }
}
```

```
class Salary extends React.Component {
    constructor(props) {
        super(props);
        this.state = {
            basic: this.props.BasicSalary,
            hra: this.props.HRA,
            sa: this.props.SpecialAllowance
        };
    }

    updateSalary = () => {
        let salary = parseInt(this.refs.BasicSalary.value) + parseInt(this.refs.HRA.value)
 + parseInt(this.refs.SpecialAllowance.value);
        this.props.onSalaryChanged(salary);
    }

    render() {
        return <div>
            <h1>Salary Details...</h1>
            <p>
                <label>Basic Salary :<input type="text" defaultValue={this.state.basic} re
f="BasicSalary" /></label>
            </p>
            <p>
                <label>HRA : <input type="text" defaultValue={this.state.hra} ref="HRA" />
</label>
            </p>
            <p>
                <label>Special Allowance : <input type="text" defaultValue={this.state.sa}
 ref="SpecialAllowance" /></label>
            </p>
            <button onClick={this.updateSalary}>Update</button>
        </div>
    }
}

const element = <Employee Id="101" Name="Jones" Salary="50000" BasicSalary="25000" HRA="10
000" SpecialAllowance="15000"></Employee>

ReactDOM.render(element, document.getElementById("root"));
```


React CRUD App with Hooks
localhost:3000
Apps
Webmail Login
Gmail
git
Links
Employee Component...
Id : 101
Name : Jones
Current Salary : 50000
Updated Salary : 0
Salary Details...
Basic Salary : 25000
HRA : 10000
Special Allowance : 15000
Update


React CRUD App with Hooks
localhost:3000
Apps
Webmail Login
Gmail
git
Links
Employee Component...
Id : 101
Name : Jones
Current Salary : 50000
Updated Salary : 55000
Salary Details...
Basic Salary : 25000
HRA : 15000
Special Allowance : 15000
Update