

**Day 2 : 12 Nov 2024**

Creating function style as well as class style component.

#### Function style component creation

```
function EmployeeFunctionStyle() {  
    // coding  
    return <div>Employee component in function style</div>  
}
```

#### Class style component creation

```
class EmployeeClassStyle extends React.Component{  
    render() {  
        // coding..  
        return <div> Employee component in class style</div>  
    }  
}
```

Login to VM

Open the terminal

```
npm install -g create-react-app
```

```
create-react-app component-style-app
```

then open this project in VS code

move inside a project folder

```
npm start
```

in React inside component we can declare two types of variables. Variable is use to describe the behaviour of component.

1. state : state is a type of variable which is use to describe the behaviour of component. state variable value we can change. So state variable is known as mutable. State variable also known as local variable.
2. props : props is a type of variable which help to share the data between one component to another component. props is known as immutable variable. Using props we can share state variable value between one component to another components.

If we want to create state variable in class component we need to use constructor. Then we need to create state variable inside a constructor and those value we can access inside JSX with `{this.state.variableName}`

Before React JS 18.x version in function style component we are not able to create state variable. Old version React function style component also known as state less component.

React hooks, with help of `react hook` we can create state variable in function components. Hook concept we can use only in function component not in class component.

So if we want to create state variable in function component react provided one of the pre defined hook ie `useState()`. Those function start the pre fix use followed by name is known as hook functions.

App -> root component

EmployeeFunctionStyle -> Child component

EmployeeClassStyle -> Child component

Using props we can pass the value from App(parent component ) to EmployeeFunctionStyle or EmployeeClassStyle component.

## Class style component Vs Function style component

1. In class style user defined class must be extends `React.Component`. But in function style component we can make normal function as a component.
2. Class style component support life cycle function as well as constructor ie render as well as more etc. but function style doesn't support constructor.
3. In class style component inside render function we need to use return JSX. In function style component normal function or component return JSX.
4. In class style component inside constructor we create state variable. But in function style component with help of `useState` hook we can create state variable.

**`useState()`**: `useState` is a pre defined hook which help us to create state variable in function component.

syntax

```
let/var [variableName,function]=useState(initialValue);
```

`useState` function return type is array. 1<sup>st</sup> parameter is variable name of any type ie number, string, Boolean, array or object etc. 2<sup>nd</sup> parameter function which help to change the value. this function is type of **asynchronous** functions.

```
let [id,setId]=useState(100);
```

`setId(200);`            this code we can call insider another function and that function we can invoke base upon some event like click.

If we change `useState` variable value directly those value or updated value doesn't re-render once again on browser. If we do change through set function then only it will re-render.

**create-react-app react-use-state-app**

DOM or JS Event

```
<input type="button" value="Click Here" onClick="functionName()"/>
```

React JS Event

In React JS Event value must be JSX and that JSX takes as callback. Function the function name or function itself to another function as parameter.

```
<input type="button" value="Click Here" onClick={functionName}/>
```