**React**

**Day 19th**

**Components types**

In React we have two types of components.

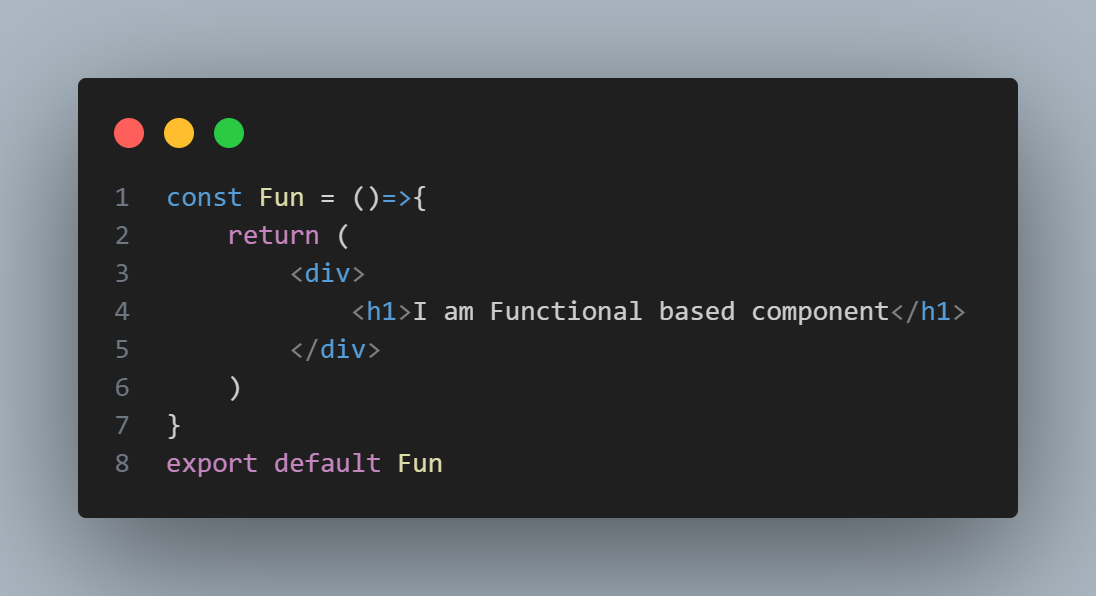
1. Class based component
2. Function based component

**Class Based Component**

It’s a traditional component, it is stateful component by default this class based component is having a state object.

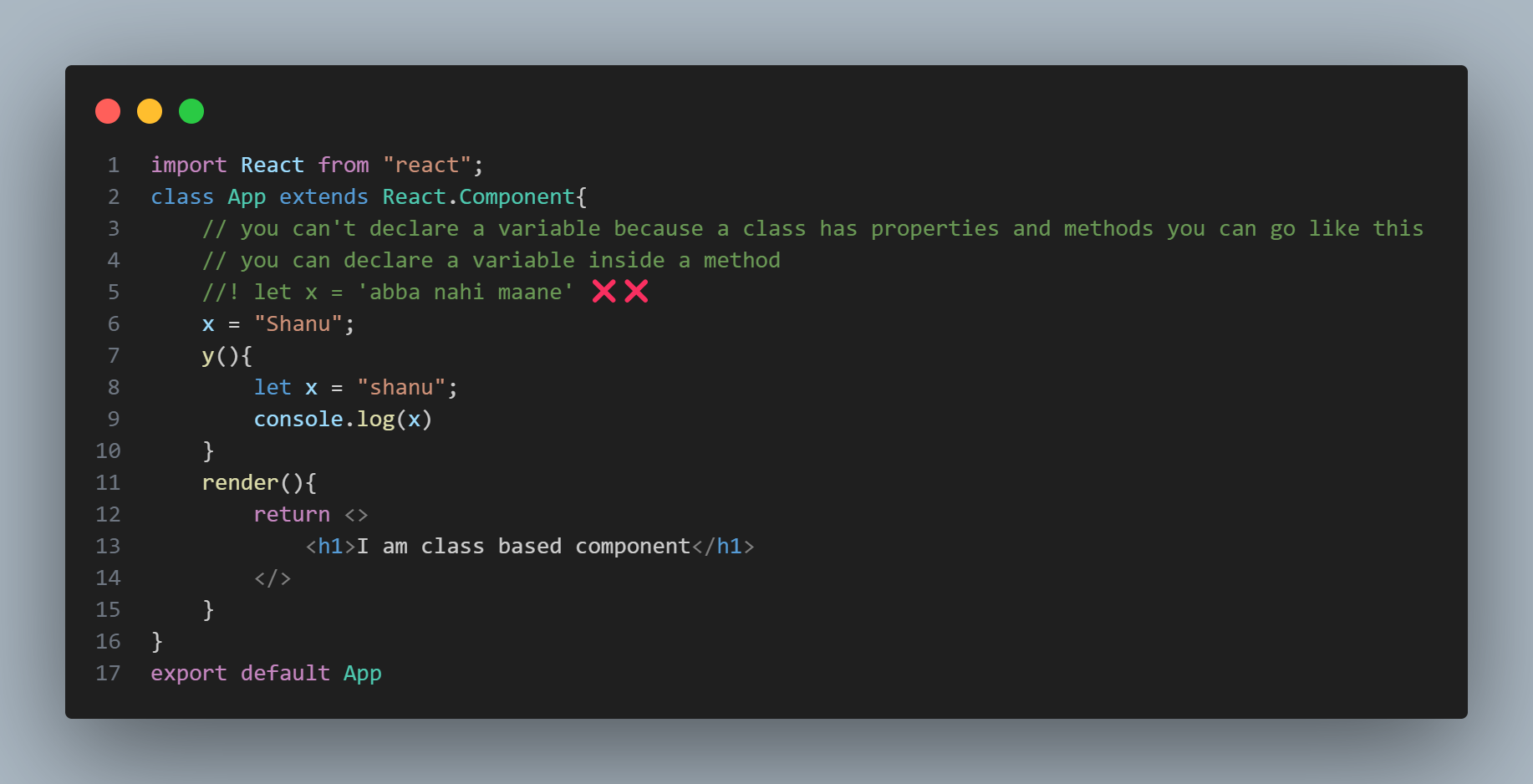
**Functional Based Component**

Function components are a means to create components in react that do not have their state and only return JSX. They are JavaScript functions that might or might not include parameters that contain data. We can write a function that accepts the props (properties) argument and outputs the displayed result.

Here is an example for FBC

**Class Based Components**

In comparison to functional components, class components are more complex. To develop class based components in react, we can use JavaScript ES6 classes. To define react component class, you need to extend React.Component. you must develop a render method that returns a react element by extending from React.Component. Data can be passed between classes and between class components. A valid class component is displayed below.

The life-cycle of a class component is accessible through specific callback APIs, which also provide access to each life cycle event.

**Handling state**

The state object, which is a built in component of react, is used to manage component actions. Re-rendering of the component will occur whenever the state object changes.

**Differences**

The most obvious one difference is the syntax. A functional component is just a plain JS function which accepts props as an argument and returns react element.

A class component requires you to extend from React.Component and create a render function which returns a react element. This requires more code but will also give you some benefits which you’ll see later on.

|  |  |  |
| --- | --- | --- |
| Differences between Function components and class components | | |
|  | Function Components | Class Components |
| Definition | JS function that returns a JSX element | JS class that extends React.Componentand has a render() method that returns a JSX element. |
| State | Uses react hooks to manage state | Has built in state management using this state |
| Lifecycle methods | Uses react hooks to handle lifecycle events like usestate, useeffect, useMemo. | Uses class lifefcycle methods like componentDidMount, componentDidUpdate and componentWillUnmount |
| Props | Props are passed as a parameter to the function | Props are accessed through this.props |
| Render | The component is defined within the function body and returned as JSX elements | The render() method is required and returns a JSX element |
| Code Clarity | Simple and concise | More verbose and harder to read |
| Performance | Sets up to follow best practices | Easier to cause performance issues. |

📝 Before 16.8 version of react only class based components were stateful. And class based component have default properties of state.

📝Functional Component does not have life cycle methods

📝Class based components supports various lifecycle methods like componentDidMount, componentDidUpdate and so on.

📝In functional based component you can achieve lifecycle methods using useEffect hooks what class based component provides

**React Hooks**

React 16.8 release in 2019 changed everything.

It introduced react Hooks.

React Hooks are functions that let functional components manage state, handle lifecycle events and access other React features. Before hooks you could only do that using class components.

**How to create a class based component**

import React,{Component} from "react";

class App extends Component{

    render(){

        return(

            <>

                <h1>I am class based component</h1>

            </>

        )

    }

}

export default App

**Constructor in React**

Constructor in react used to define or initialize state property in Class Based Components. It is the best place for initialize the state object and binding this keyword.