**COMPONENTS:**

* React Component are reusable building block of code which associated with logic.
* Using this we can build SPA
* **Types of Component :**

1. Class based component
2. Function based components

**Note :**

* Name of the component should start from uppercase and .jsx extension.

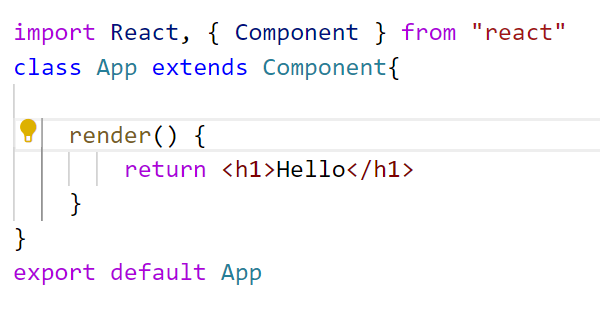
Ex : App.jsx

* We can represent component in 2 ways

1. < App/>
2. <App><App/>

**Class based component :**

**App.jsx**

****

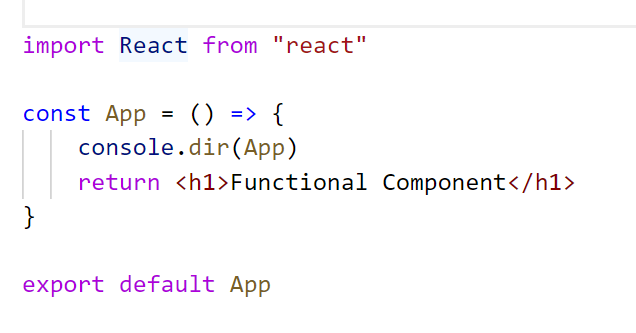
main.jsx



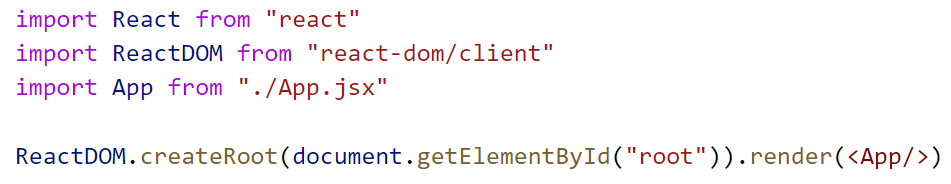
**Note :** In class based component there is one built in property called as state so class based component called as statefull Component.

**Function based component :**

App.jsx



main.jsx



**Note :** In function based component there is no built in property called as state so function based component called as stateless Component.

**Difference between Class based and Function based component**

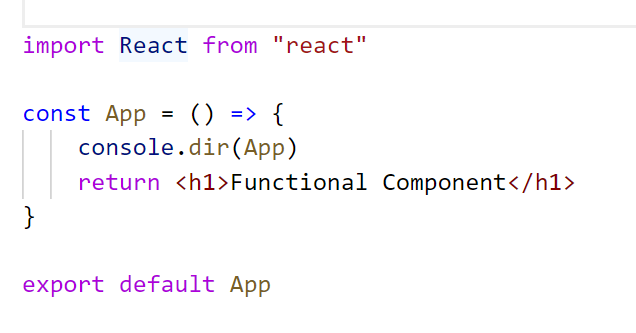
**(important interview question)**

|  |  |
| --- | --- |
| **Class Based** | **Function Based** |
| **It is called statefull** | **It is called stateless** |
| **Hooks are not supported** | **Hooks are supported** |
| **Life Cycle methods are present** | **Life Cycle methods are not present** |
| **Render method is mandatory** | **Render method is not mandatory** |
| **this keyword points to current class based component** | **no this keyword** |

**JSX: (Javascript XML) :**

* It is the combination of JavaScript and XML.
* It is an extension to JS syntax allows you to write HTML-like code when working with React.

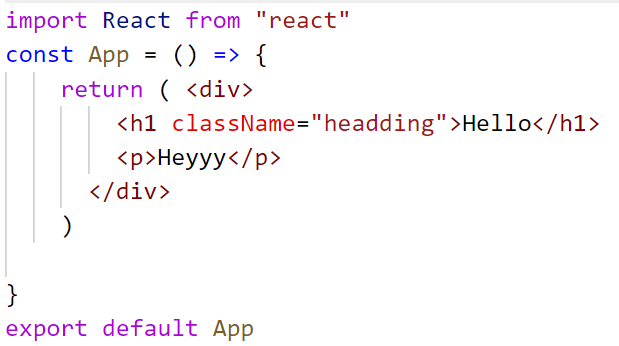
App.jsx



In this example code inside return (<h1> Functional Component</h1>) statement looks similar to HTML, but it is actually JSX.

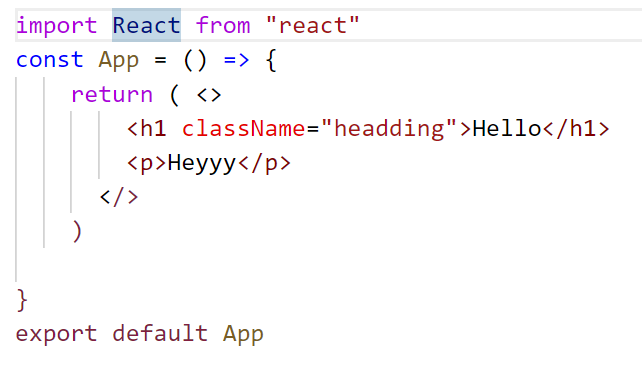
**JSX RULES :**

1. **When you want to render more than one element wrap them with one parent and ( )**

****

**Note :** In the above example we have used <div> </div> as a parent .So in this case <div></div> will create one new node. In order to avoid this we have to use fragments as a parent.

**Wrapping with fragments**



1. **In JSX every element must be closed**

* **In JSX all HTML tags must be properly closed.**

**Example:**

|  |  |
| --- | --- |
| In HTML | In JSX |
| <input type = “text “> | <input type="text" />  Or  <input type="text"></input> |
| <br> | <br />  Or  <br></br> |
|  |  |

**App.jsx**

****

1. **In JSX in order access the variable we need to wrap them with { }**

****

1. **CamelCase Conventions**

|  |  |
| --- | --- |
| **HTML** | **JSX** |
| **class** | **className** |
| **for** | **htmlFor** |

JS Events that we use here should also be in camelCase

**onclick-----------------------🡪onClick**