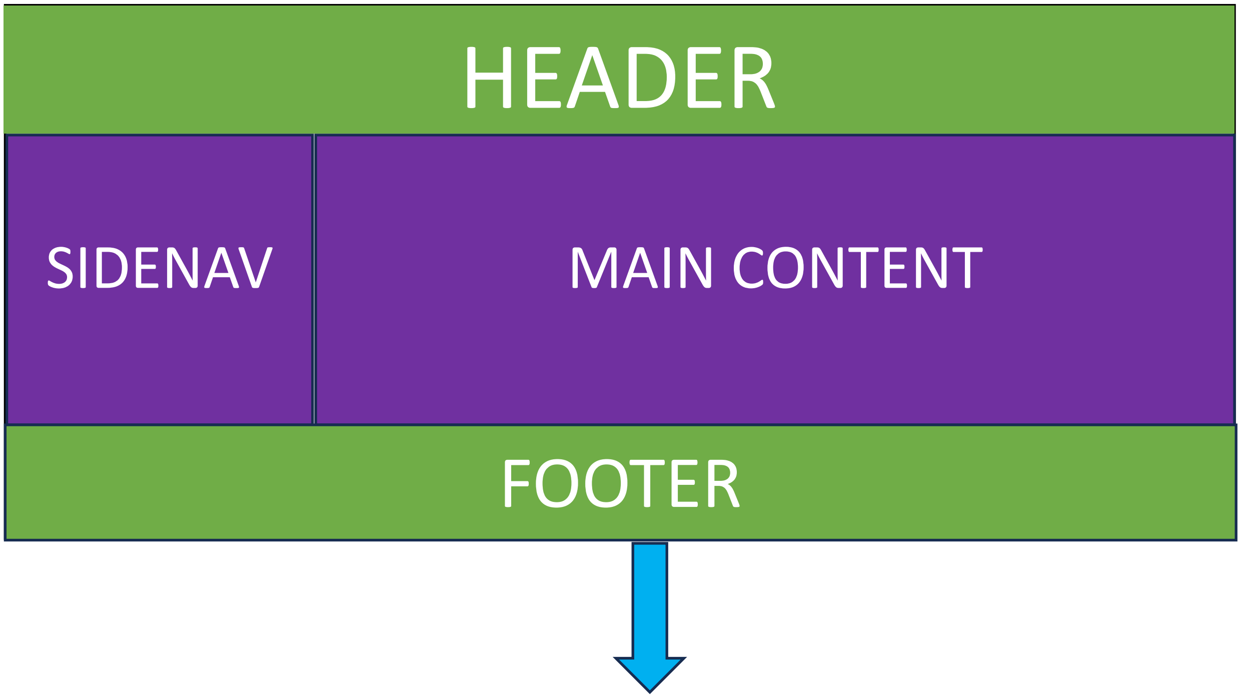
REACT

LESSON 4 – COMPONENTS

CO2214 - PRACTICAL WORK ON CO2224

COMPONENTS

- Components represents parts of UI in React.
- Our application has Five Components.



Root (App) Component

1. Header
2. SIDENAV
3. Main Content
4. Footer
5. {Root (App) Component}

   One Component to contain other Components

- Components also reusable.

- Same Components can be used with different properties to display different Information.

  Eg:- SIDENAV component can be left SIDENAV and Right SIDENAV.

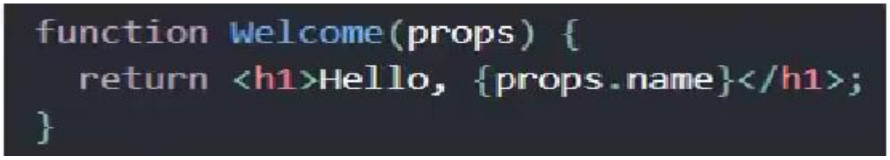
Component in Code

- Component Codes usually placed in a JavaScript File.
  Eg: - App Component is placed in App.js

- Component is a basically a code inside .js file. The code is looking different. Because It is Depend on Component Types.

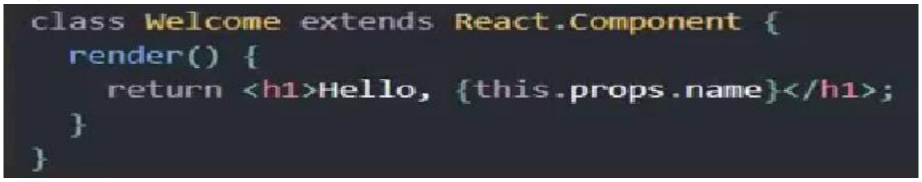
Component Types

1. Stateless Functional Components
   - Functional Components are literally JavaScript Functions. (The Written Html which describes UI)
     Eg:-



```
function Welcome(props) {
  return <h1>Hello, {props.name}</h1>;
}
```

2. Stateful Class Component
   - Regular ES6 Classes that extend Component Class from the react library.
   - They must contain the render method which return html.
     Eg:-



```
class Welcome extends React.Component {
  render() {
    return <h1>Hello, {this.props.name}</h1>;
  }
}
```

Functional Components

- Functional Components are just JavaScript Functions.

- They can optionally receive an object of properties which is refer to as props and return html which describes UI.

Now we are going to create our first Functional Component.

1. Go to App.js and clear all html without div.

2. Create a new folder called components inside the src folder with in the folder create new file Greet.js.

3. within the file the first step is import react

   ```
   import React from ‘react’
   ```

   for any component you need to create you have to import React.

4. Next let's create a new function

   ```
   function Greet ()
   {
    return <h1> Hello Inzam </h1>
   }
   ```

   That's it you have created your first Functional Components.

   But it is not going to be rendered in the browsers, because the Greet () is not connected with the rest of our application.

So, what we have to do is we need to export Greet () from Greet.js to App.js than included in the App Component

So, in Greet.js

```
export default Greet
```

5. in App.js

```
import Greet from './Components/Greet'
```

Next, we need to include Greet component inside the App Component by simply specify the Component as a Custom Html tag

```
<Great></Great>
```

if there is no any content between the tag you can simply change it in to closing tag

```
<Great />
```

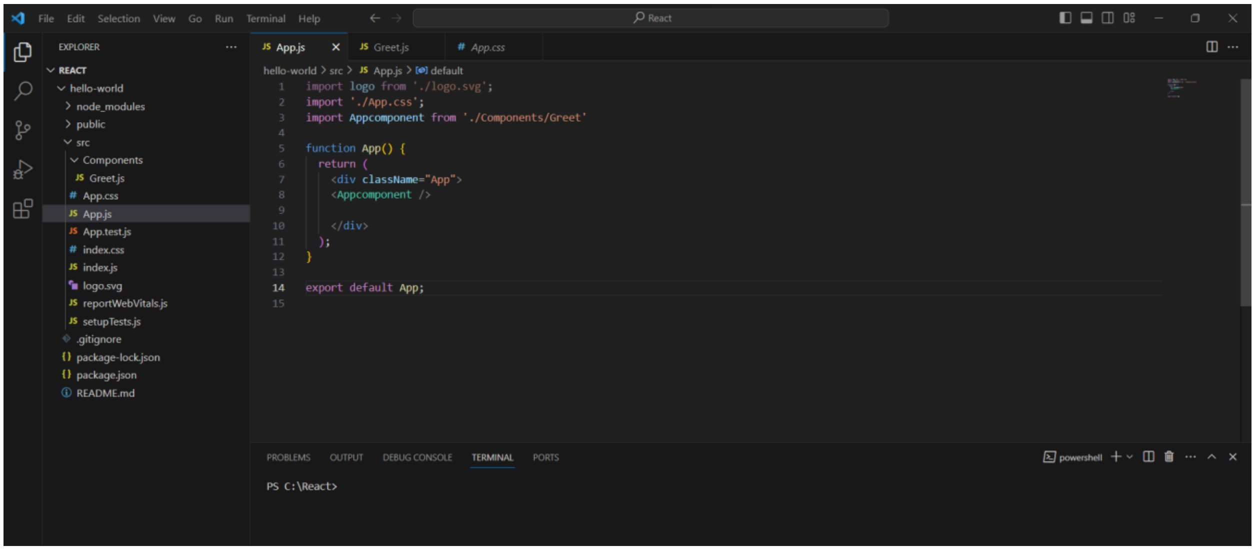
Now look at the browser and see the output.

- let's rewrite the Greet Function using ES6 Arrow Syntax

in Greet.js file

```
const Greet = () => <h1> Hello Inzam </h1>
export default Greet
```

Note the in Greet.js we are export Greet Component as default export that allow us to import the component with any name. we can change the name of import and change the tag inside the app component.



```
import logo from './logo.svg';
import './App.css';
import Appcomponent from './Components/Greet'

function App() {
  return (
    <div className="App">
    <Appcomponent />

    </div>
  );
}

export default App;

```

```
PS C:\React>
```

❖ let me comment default export and export it in ES6 now we have to give the same name otherwise it will give an error.

In Greet.js

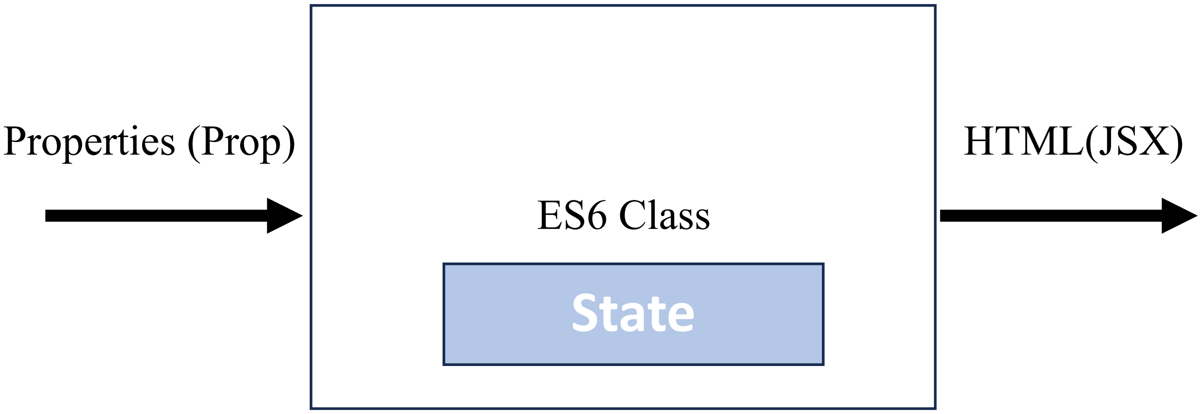
export const Greet = () => <h1> Hello Inzam </h1>

In App.js

import {Greet} from './Components/Greet'

Class Components

- Class Components are basically a ES6 Classes.
- Similar to Functional Components Class Components also receive props as input and return html.



- Apart from the props class Components can also maintain a private internal State (In simple words it can maintain some information which is private to that component and use that information to describes a UI.)

- Now we are going to create exact Greet Component using Class Component.

  1. Create a Welcome.js file inside the Components Folder.
  2. Whenever we need to create a class component, we need to include two imports.
     I. React
     II. Component Class from React

     import React, {Component} from 'react'

3. Next, we define the class

   ```
   class Welcome { }
   ```

4. This class to become a react component there are two simple steps.

   First steps it should extends Component Class from react

   ```
   class Welcome extends Component {}
   ```

   Second Step the Class has to implement render () which will return none or some html.

   ```
   class Welcome extends Component {

   render ()
       {
               return <h1>Class Component</h1>
       }

   }
   ```

5. Now, you have created your Class Component But it is not connected to rest of our application. So, we need to export Class from Welcome.js and import it in App.js

   in Welcome.js

```
class Welcome extends Component {

render ()
   {
         return <h1>Class Component</h1>
   }

}

export default Welcome
```

in App.js

```
import Welcome from './Components/Welcome'
```

within the html you have to add custom tag.

```
<Welcome />
```

Functional Vs Class Components

Functional

- ✓ Simple functions (Receiving Props and returning declaration)
- ✓ Use Functional Components as much as possible.
- ✓ Absence of 'this' keyword
- ✓ Solution without using state

(If you have number of components each with a own private state maintenance and debugging your applications is kind of difficult).

- ✓ Mainly Responsible for the UI.

Class

- ✓ More Feature rich
- ✓ Maintain their own private data – State
- ✓ Complex UI logic
- ✓ Provide lifecycle hooks

- ❖ So far, we have seen 3 components App.js, Greet.js, Welcome.js in all the three components we simply return some html, but let me tell you that is not regular html, that is something known as JSX.