**React Components:**

* + The building blocks of a React application.
  + Encapsulate UI and logic into reusable pieces.
  + Each component returns JSX which defines how the UI looks.
  + Components can have **state**, **props**, and lifecycle methods (for class components).

**Identify the differences between components and JavaScript functions**

* **JavaScript Functions:**
  + General-purpose code blocks that perform a task and return a value.
  + Do not inherently handle UI rendering or state.
* **React Components:**
  + Special JavaScript functions or classes that return JSX.
  + Manage UI, state, and props.
  + Follow React’s rules (e.g., component names must start with a capital letter).

**Identify the types of components**

* **Two main types in React:**
  1. **Class Components:**
     + ES6 classes extending React.Component.
     + Can have state and lifecycle methods.
  2. **Function Components:**
     + JavaScript functions returning JSX.
     + Originally stateless, but with **Hooks**, they can use state and lifecycle features.

**Explain class component**

* **Class Component:**
  + Defined using ES6 class syntax.
  + Extends React.Component.
  + Must have a render() method to return JSX.
  + Supports state and lifecycle methods.
  + Example:

class MyComponent extends React.Component {

render() {

return <h1>Hello, Class Component!</h1>;

}

}

**Explain function component**

* **Function Component:**
  + A JavaScript function that returns JSX.
  + Simpler and easier to write than class components.
  + With React Hooks, they can manage state and side effects.
  + Example:

function MyComponent() {

return <h1>Hello, Function Component!</h1>;

}

**Define component constructor**

* **Constructor in Class Component:**
  + A special method in class components used to initialize state and bind methods.
  + Called before the component is mounted.
  + Example:

class MyComponent extends React.Component {

constructor(props) {

super(props);

this.state = { count: 0 };

}

}

**Define render() function**

* **render():**
  + Mandatory method in class components.
  + Returns JSX to define the UI.
  + Called whenever the state or props change to update the UI.
  + Example:

class MyComponent extends React.Component {

render() {

return <p>This is rendered on the screen</p>;

}

}