**Define Props**  
Props, short for properties, are read-only inputs passed from a parent component to a child component in React. They allow data to flow in a unidirectional manner from parent to child, enabling components to be dynamic and reusable. Props are accessed within the child component using this.props in class components or directly as function arguments in function components.

**Explain Default Props**  
Default props are predefined values assigned to a component's props in case no value is provided by the parent component. This ensures that the component has a fallback behavior and does not break due to missing props. In class components, default props can be defined using the defaultProps property. In function components, they can be set by assigning default values in the function parameter list or using the Component.defaultProps syntax.

**Identify the differences between State and Props**  
Props are used to pass data from parent to child components and are immutable within the child component. They define how a component should behave based on the data received. State, on the other hand, is managed within the component itself and can be updated using methods such as setState in class components or useState in function components. While props are set externally by parent components, state is internal and controls the component’s behavior and rendering over time.

**Explain reactDOM.render()**  
The ReactDOM.render() method is used to render a React element or component into the DOM. It takes two arguments: the React element or component to render, and the DOM container where it should be mounted. This method is typically called once in a React application to load the root component. For example, ReactDOM.render(<App />, document.getElementById('root')) renders the App component inside the HTML element with the id root.