React State and Counter App

# 1. Explain React State

In React, **State** is a built-in object that allows components to **store and manage dynamic data**. It represents **information about the component that can change over time**, usually as a result of user interaction, API calls, or events.  
Key Characteristics:  
- Mutable: State can be changed after the component is rendered.  
- Local to the component.  
- Changes trigger UI updates automatically.  
  
Syntax in Class Components:  
1. Define state in constructor: this.state = { count: 0 };  
2. Update state using this.setState({ count: newValue });  
  
Difference between Props and State:

|  |  |  |
| --- | --- | --- |
| Feature | Props | State |
| Mutability | Immutable (read-only) | Mutable |
| Defined By | Parent Component | Component itself |
| Scope | Passed to child components | Local to the component |
| Updates Cause | Component re-renders if props change | Component re-renders on change |

# 2. CountPeople.js

import React, { Component } from 'react';  
import './App.css';  
  
class CountPeople extends Component {  
 constructor(props) {  
 super(props);  
 this.state = {  
 entryCount: 0,  
 exitCount: 0  
 };  
 }  
  
 updateEntry = () => {  
 this.setState({ entryCount: this.state.entryCount + 1 });  
 }  
  
 updateExit = () => {  
 this.setState({ exitCount: this.state.exitCount + 1 });  
 }  
  
 render() {  
 return (  
 <div className="container">  
 <div className="buttonSection">  
 <button onClick={this.updateEntry}>Login</button>  
 <span>{this.state.entryCount} People Entered!!!</span>  
 </div>  
 <div className="buttonSection">  
 <button onClick={this.updateExit}>Exit</button>  
 <span>{this.state.exitCount} People Left!!!</span>  
 </div>  
 </div>  
 );  
 }  
}  
  
export default CountPeople;

# 3. App.js

import React from 'react';  
import CountPeople from './CountPeople';  
  
function App() {  
 return (  
 <div>  
 <CountPeople />  
 </div>  
 );  
}  
  
export default App;

# 4. CSS Code (App.css)

.container {  
 text-align: center;  
 margin-top: 100px;  
}  
  
.buttonSection {  
 margin: 20px;  
}  
  
button {  
 background-color: green;  
 color: white;  
 border: none;  
 padding: 10px 20px;  
 margin-right: 10px;  
 font-weight: bold;  
 border-radius: 5px;  
}  
  
span {  
 font-size: 18px;  
}  
**Output:**



