







State vs Props in React

Feature	Props	State
 Definition	Props are input data passed to a component from its parent	State is local data stored with the component
 Mutability	Immutable – cannot be changed by the receiving component	Mutable – can be updated over time
 Used for	Passing data into components	Managing data within a component
 Updates?	Props are fixed for the component's lifecycle unless changed by parent	State can change , and React will re-render the component
 Who sets it?	Set by the parent component	Set by the component itself
 Usage Example	<code><Welcome name="John" /></code>	<code>const [count, setCount] = useState(0);</code>

Example to See the Difference:

jsx

Copy

Edit

```
function Greeting(props) { return <h1>Hello, {props.name}!</h1>; // using props } function Counter() { const [count, setCount] = useState(0); // using state return ( <div> <Greeting name="Alice" /> { /* props */} <p>Clicked {count} times</p> { /* state */} <button onClick={() => setCount(count + 1)}>Click me</button> </div> ); }
```

Summary (One-liner Difference):

- Props are like arguments **passed to a function**.
- State is like variables **declared inside a function**, which can change over time.

