# Difference Between useState and Props in React

In React, `useState` and Props serve different purposes. While `useState` is used for managing internal state, Props are used for passing data from a parent component to a child component. Below is a detailed comparison between the two.

## Comparison Table

|  |  |  |
| --- | --- | --- |
| Feature | Props | useState |
| Purpose | Used to pass data from parent to child components. | Used to manage the internal state of a component. |
| Type | Immutable (read-only). | Mutable (can be updated). |
| Control | Controlled by the parent component. | Controlled within the component itself. |
| Data Flow | Unidirectional: from parent to child. | Local to the component. |
| Usage | Passed as attributes to child components. | Managed using the `useState` hook. |
| Lifecycle | Exists throughout the parent-child relationship. | Exists only within the component's lifecycle. |
| Scope | Shared between components (parent and child). | Specific to a single component. |
| Responsibility | Props are controlled externally. | State is controlled internally. |

## Examples

### Using Props

Props are used to pass data from a parent component to a child component. For example:

function ProductCard({ name, price }) {  
 return (  
 <div>  
 <h2>{name}</h2>  
 <p>Price: ${price}</p>  
 </div>  
 );  
}  
  
function App() {  
 return (  
 <div>  
 <ProductCard name="Laptop" price={1000} />  
 <ProductCard name="Phone" price={800} />  
 </div>  
 );  
}

### Using useState

`useState` is used to manage dynamic data that can change within a component. For example:

import React, { useState } from 'react';  
  
function Counter() {  
 const [count, setCount] = useState(0);  
  
 return (  
 <div>  
 <p>Count: {count}</p>  
 <button onClick={() => setCount(count + 1)}>Increment</button>  
 </div>  
 );  
}  
  
export default Counter;

### Combining Props and useState

You can use both Props and `useState` together to create dynamic components. For example:

import React, { useState } from 'react';  
  
function ProductCard({ name }) {  
 const [isAvailable, setIsAvailable] = useState(true);  
  
 return (  
 <div>  
 <h2>{name}</h2>  
 <p>Status: {isAvailable ? 'Available' : 'Out of Stock'}</p>  
 <button onClick={() => setIsAvailable(!isAvailable)}>  
 {isAvailable ? 'Mark Out of Stock' : 'Mark Available'}  
 </button>  
 </div>  
 );  
}  
  
function App() {  
 return (  
 <div>  
 <ProductCard name="Laptop" />  
 <ProductCard name="Phone" />  
 </div>  
 );  
}