

# React - State

**Topics :** [React JS](#)

**Written on** [January 03, 2024](#)

In React, "state" refers to an object that represents the current condition of a component. It is used to store and manage dynamic data that can change over time as a result of user interactions, API calls, or other factors.

State is a fundamental concept in React because it allows components to be dynamic and interactive. When the state of a component changes, React automatically re-renders the component to reflect the updated state.

To work with state in React, you typically use the `useState` hook, which was introduced in React version 16.8. Here's a basic example of how to use `useState`:

```
import React, { useState } from 'react';

function Counter() {
  // useState returns an array with two elements: the current state value and a function to update it
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>Count: {count}</p>
      <button onClick={() => setCount(count + 1)}>Increment</button>
    </div>
  );
}

export default Counter;
```

In this example, the `useState` hook is used to declare a state variable `count` with an initial value of `0`. The `setCount` function is then used to update the `count` state when the button is clicked. React will automatically re-render the `Counter` component, and the updated count value will be displayed.

It's important to note that the state in React is immutable. When you update the state using a function like `setCount`, React re-renders the component with the new state, rather than modifying the existing state directly.