

# Controlled Components

In React, form inputs can be managed in two ways: **controlled components** and **uncontrolled components**. The choice between these two approaches depends on how you want to handle the state and data flow in your application.

## Uncontrolled Components

- In uncontrolled components, the **state of the input is managed by the DOM**.
- The input's value is not controlled by React but is instead retrieved directly from the DOM when needed (e.g., during form submission).
- React does not manage the input's value; it acts as a **passive observer**.

## Advantages of Uncontrolled Components

- Simpler to implement for simple use cases.
- Less boilerplate code since React does not manage the state.
- Useful for integrating with non-React code or libraries.

## Disadvantages of Uncontrolled Components

- Less control over form data
- Difficulty in implementing complex validation
- Challenges with debugging, less predictable state management
- Cannot update error validation on input change [need `onBlur` or `onSubmit` event]

## Controlled Components

- In controlled components, the **state of the input is managed by React**.
- The value of the input is tied to the component's state, and changes to the input are handled via event handlers (e.g., `onChange`).

- React acts as the **single source of truth** for the input's value.

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

function ControlledInput() {
  const [value, setValue] = useState("");

  const handleChange = (e) => {
    setValue(e.target.value);
  };

  return (
    <div>
      <label htmlFor="name">Name:</label>
      <input
        type="text"
        id="name"
        value={value}
        onChange={handleChange}
      />
      <p>You typed: {value}</p>
    </div>
  );
}

export default ControlledInput;
```

## How Controlled Components Work

1. The input's value is stored in the component's state.
2. When the user types or changes the input, an `onChange` event handler updates the state.
3. The updated state is then passed back to the input as its value.

## Advantages of Controlled Components

- Full control over the input's value and behavior.

- Easy to validate or transform input data before updating the state.
- Works well with React's declarative nature.