

Controlled Component

Definition:

A controlled component is a component where the form data is handled by the component's state using **useState** or similar hooks.

A **controlled component** is a form element (like `<input>`, `<textarea>`, `<select>`) whose **value is controlled by React state**. React fully manages the input's data using `useState`, and any change in the UI is reflected through state updates.

Example:

```
const [name, setName] = useState("");  
<input value={name} onChange={e => setName(e.target.value)} />
```

Uncontrolled Component

Definition:

An uncontrolled component is a component where the form data is handled by the DOM itself, and input values are accessed using React refs.

An **uncontrolled component** is a form element where **the DOM itself manages the input's value**. React does not directly control the form input's value. Instead, values are accessed using **ref** when needed.

Example:

```
const inputRef = useRef();  
<input ref={inputRef} />
```

Difference b/w Controlled & Uncontrolled Components

Feature	Controlled Component	Uncontrolled Component
Data Handling	Handled by React (useState)	Handled by the DOM (with ref)
Input Value Source	value prop controlled by React	defaultValue or internal DOM value
React Control	Fully controlled by React	React does not manage the value
Form Validation	Easy to handle with React logic	Harder, requires DOM access
Performance	Slightly slower for large forms (due to re-rendering)	Faster for simple inputs
Use of ref	Not required (state-driven)	Required to get the current value
Example	<code><input value={name} onChange={...} /></code>	<code><input defaultValue="John" ref={inputRef} /></code>
Use Case	Complex forms, validations, dynamic inputs	Simple forms, minimal logic, quick refs