Forms

HTML form elements work a little bit differently from other DOM elements in React, because form elements naturally keep some internal state.

In HTML, form elements such as <input>, <textarea>, and <select> typically maintain their own state and update it based on user input. In React, mutable (changeable) state is typically kept in the state property of components, and only updated with setState().

- Controlled Component
- Uncontrolled Component

Controlled Component

Form has the default HTML form behavior of browsing to a new page when the user submits the form. If you want this behavior in React, it just works. But in most cases, it's convenient to have a JavaScript function that handles the submission of the form and has access to the data that the user entered into the form. The standard way to achieve this is with a technique called "controlled components".

In a controlled component, form data is handled by a React component.

When Use Controlled Component-

You need to write an event handler for every way your data can change and pipe all of the input state through a React component.

Uncontrolled Component

In a controlled component, form data is handled by a React component. The alternative is uncontrolled components, where form data is handled by the DOM itself.

To write an uncontrolled component, instead of writing an event handler for every state update, you can use a ***ref*** to get form values from the DOM.

When Use Uncontrolled Component-

You do not need to write an event handler for every way your data can change and pipe all of the input state through a React component.

Converting a preexisting codebase to React, or integrating a React application with a non-React library.