

# 非制御コンポーネント

ほとんどの場合では、フォームの実装には制御されたコンポーネントを使用することをお勧めしています。制御されたコンポーネントでは、フォームのデータは React コンポーネントが扱います。非制御コンポーネントはその代替となるものであり、フォームデータを DOM 自身が扱います。

非制御コンポーネントを記述するには、各 state の更新に対してイベントハンドラを書く代わりに、`ref` を使用して DOM からフォームの値を取得します。

例えば、以下のコードは非制御コンポーネントで 1 つの名前を受け取ります：

```
class NameForm extends React.Component {
  constructor(props) {
    super(props);
    this.handleSubmit = this.handleSubmit.bind(this);
    this.input = React.createRef();
  }

  handleSubmit(event) {
    alert('A name was submitted: ' + this.input.current.value);
    event.preventDefault();
  }

  render() {
    return (
      <form onSubmit={this.handleSubmit}>
        <label>
          Name:
          <input type="text" ref={this.input} />
        </label>
        <input type="submit" value="Submit" />
      </form>
    );
  }
}
```

## CodePen で試す

非制御コンポーネントでは DOM に信頼できる情報源を保持するため、使用すれば React と非 React のコードの統合が簡単になることがあります。汚くても構わないので速く記述したいと思うなら少しだけコード量も減らせます。そうでなければ、通常の制御されたコンポーネントを使用するべきです。

特定の状況に対してどちらのタイプのコンポーネントを使用すれば良いかまだはっきりしない場合は、制御された入力 vs 非制御入力の記事が参考になるでしょう。

## デフォルト値

React のレンダラのライフサイクルでは、フォーム要素の `value` 属性は DOM の値を上書きします。非制御コンポーネントでは、React に初期値を指定させるが後続の更新処理には関与しないようにしたいことがよくあるでしょう。このケースを扱うために、`defaultValue` 属性を `value` の代わりに指定できます。

```
render() {
  return (
    <form onSubmit={this.handleSubmit}>
      <label>
        Name:
        <input
          defaultValue="Bob"
          type="text"
          ref={this.input} />
      </label>
      <input type="submit" value="Submit" />
    </form>
  );
}
```

同様に、`<input type="checkbox">` と `<input type="radio">` が `defaultChecked` を、そして `<select>` と `<textarea>` が `defaultValue` をサポートしています。

## ファイル input タグ

HTML では、`<input type="file">` を利用してユーザに 1 つ以上のファイルをデバイスストレージから選択させ、サーバにアップロードしたり [File API](#) を通じて JavaScript で操作したりします。

```
<input type="file" />
```

React では、`<input type="file" />` は値がユーザだけが設定できるものでありプログラムでは操作できないため、常に非制御コンポーネントです。ファイルをやり取りするのに [File API](#) を使用してください。以下の例では DOM ノードへの `ref` を作成し submit ハンドラでファイルにアクセスしています。

```
class FileInput extends React.Component {
  constructor(props) {
    super(props);
    this.handleSubmit = this.handleSubmit.bind(this);
    this.fileInput = React.createRef();
  }
  handleSubmit(event) {
    event.preventDefault();
    alert(
      `Selected file - ${
        this.fileInput.current.files[0].name
      }`
    );
  }
  render() {
    return (
      <form onSubmit={this.handleSubmit}>
        <label>
          Upload file:
          <input type="file" ref={this.fileInput} />
        </label>
        <br />
        <button type="submit">Submit</button>
      </form>
    );
  }
}

ReactDOM.render(
  <FileInput />,
  document.getElementById('root')
);
```

**CodePen で試す**

[このページを編集する](#)