JSX なしで React を使う

JSX は React を使うための必須事項ではありません。JSX なしで React を使うことは、あなたのビルド環境で JSX のコンパイルの設定をしたくない時には便利です。
各 JSX 要素は、`React.createElement(component, props, ...children)` を呼び出すための単なるシンタックスシュガーです。つまり、JSX を使ってできることは、普通の JavaScript を使ってもできます。
例えば、JSX で書かれた以下のコードは：

```
class Hello extends React.Component {
  render() {
    return <div>Hello {this.props.toWhat}</div>;
  }
}

ReactDOM.render(
  <Hello toWhat="World" />,
  document.getElementById('root')
);
```

JSX を使わない以下のコードにコンパイルできます：

```
class Hello extends React.Component {
  render() {
    return React.createElement('div', null, `Hello ${this.props.toWhat}`);
  }
}

ReactDOM.render(
  React.createElement(Hello, {toWhat: 'World'}, null),
  document.getElementById('root')
);
```

JSX から JavaScript への変換方法の例をもっと見たいなら、オンラインの Babel コンパイラで試すことができます。
コンポーネントは文字列、`React.Component` のサブクラス、もしくは（ステートレスコンポーネントの場合）プレーンな関数のいずれかで指定されます。
たくさんの `React.createElement` をタイピングするのにうんざりした場合、一般的なパターンの 1 つは以下のショートハンドを割り当てることです。

```
const e = React.createElement;

ReactDOM.render(
  e('div', null, 'Hello World'),
  document.getElementById('root')
);
```

このショートハンドを `React.createElement` に使用すれば、JSX なしで React を使うのにとても便利です。
あるいは、簡潔な構文を提供する `react-hyperscript` や `hyperscript-helpers` のようなコミュニティプロジェクトも参照してみてください。

このページを編集する