

CodeSandbox で React のフォーム部品を操作してみよう

仕事や趣味を通して JavaScript のコードを書いている方であれば、「CodeSandbox を使って React の UI コンポーネントを作る」といってピンとくる方も多いと思いますが、実際に興味はあるけれど、どう React のコードを書けばよいのかわからないという方も多いと思います。

今回は、CodeSandbox 上で React のフォーム部品を作成して操作してみる、ということ実施していきたいと思います。React に慣れる為に、手を動かして考える機会にできれば幸いです。

React の基礎から確認したいという方は以下を参考になさってください。

参考： モダン JS から始める「React」入門

http://semi3del.xsrv.jp/wp_supporter/?page_id=2984

ここでは CodeSandbox 上で UI コンポーネントの作成方法と map などの基本的な関数の確認をした後、実際に React のフォーム部品を作成していければと思います。

○ はじめに

HTML / JavaScript の開発では、適宜テキストエディタやオンラインのコードエディタを使うことで仕事も早く進めることができます。今回は、CodeSandbox を使用していきます。

- ・ オンラインのコードエディタ

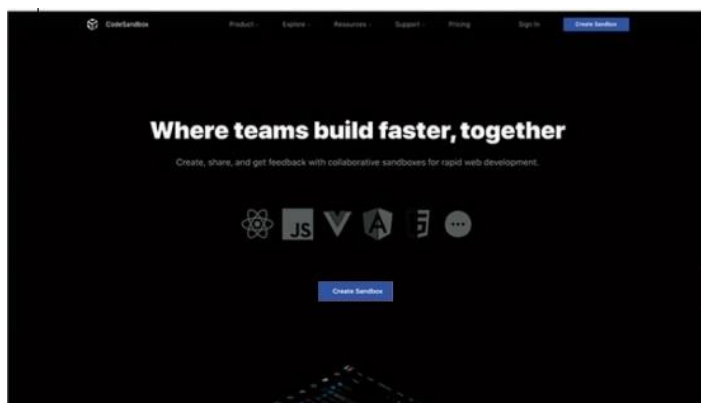
CodeSandbox

Web サイト： <https://codesandbox.io/>

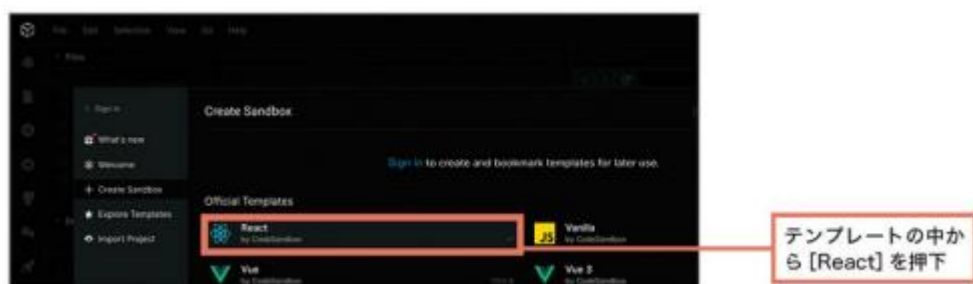
今回使用するオンラインのコードエディタです。Web エディタ上で簡単に JavaScript の作成ができて、コーディングや、URL での共有、GitHub との連携も行いうことが出来るサービスです。

開発の現場で使われるというより、個人の勉強やコードの共有の為に使われるサービスで、環境構築自体に時間を割かずにコードを書く勉強に取り組むことができます。

▶ CodeSandbox の TOP ページ



CodeSandbox メニュー画面で表示されたテンプレートの中にある[React]をクリックして、React のプロジェクトを開始します。



○ React ～ モダンな Web 開発のためのライブラリ について

JavaScript の言語仕様については、2015 年頃に ES2015 : ECMAScript（第 6 版）にて公開された仕様ークラスやモジュール、イテレータ、ジェネレータ、アロー演算子、テンプレート文字列、型付き配列などーの要素が盛り込まれました。

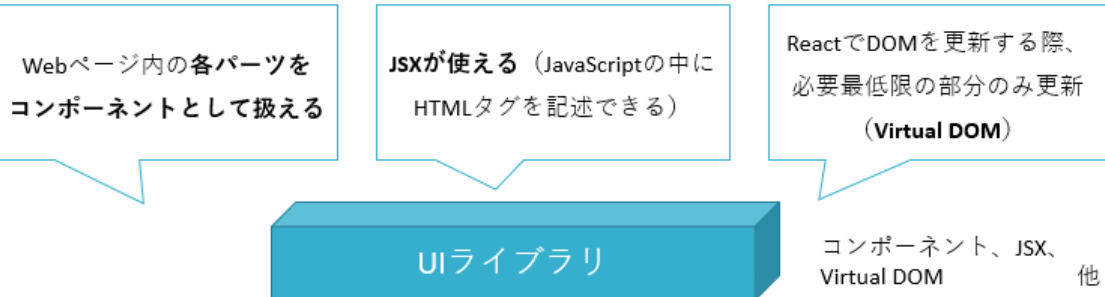
それは、言語仕様の話だけではなく、JavaScript 開発を支援するライブラリ・フレームワークも日々登場しています。その中で注目されているのが、React.js, Vue.js, Angular.js などのライブラリです。

ここでは、UI 部品のフォームを画面表示しながら確認していくことで、React の基礎事項を小テーマごとに復習していきます。

Reactとは？

- ・ Meta社とコミュニティが開発している「**UIライブラリ**」
～ HTMLをコンポーネントとして定義し、組み合わせて使うことができる

> 特徴



ここからは始める「React」～基礎事項の確認をしていきます

まずは、CodeSandbox で React のプロジェクトを作成し、Hello World を表示させてみましょう。

CodeSandbox のメニュー画面で表示されたテンプレートの中にある [React] をクリック
一度 src フォルダ内にあるファイルを全て削除し、index.js を新規に作成します。

以下のコードを記載して、「Hello World」を表示してみましょう。

JSX記法

- ・ コンポーネントを作るには、「**JSX**」で**コンテンツを返す関数を定義**する

⇒ 関数の返却値として **HTMLのタグを記述し、コンポーネントとして画面を構成**

file: src/index.js

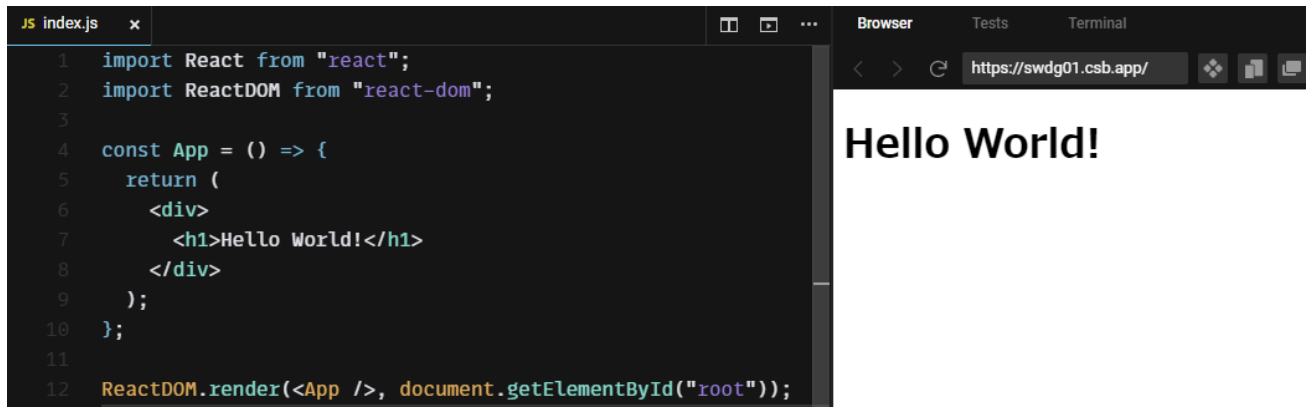
```
import React from "react";
import ReactDOM from "react-dom";

const App = () => {
  return (
    <div>
      <h1>Hello World !</h1>
    </div>
  );
};
```

ReactDOMのrender関数内
第1引数：render対象
第2引数：render箇所

関数名をHTMLのようにタグで囲う
ことでコンポーネントとして扱える

```
ReactDOM.render(<App />,
  document.getElementById("root"));
```



○ React の基礎 ～問題演習～

それでは、実際に資料をダウンロードして、React 基礎の演習問題を解いていきましょう。

React 基礎演習 ～Reactのフォーム部品を操作～

＊該当のシートに移動するには、問題番号のハイパーリンクを押下

問題番号	シート名	内容
No1	JSX記法	CodeSandboxでJSX記法を用いて、簡単なHTMLを出力する
No2	mapメソッドの使い方	配列の要素をmapメソッドを使って書き出す
No3	input, textareaの作成	フォーム部品①～inputとtextareaタグ input要素に対してはvalueと onChangeに対してStateを連携させる
No4	ラジオボタンの作成	フォーム部品②～radio ボタン ラジオボタンを画面表示するコードを書く（mapメソッドも利用可）
No5	チェックボックスの作成	フォーム部品③～checkbox ボタン チェックボックスを画面表示するコードを書く
No6	プルダウンの作成	フォーム部品④～selectとoptionタグ プルダウンを画面表示するコードを書く

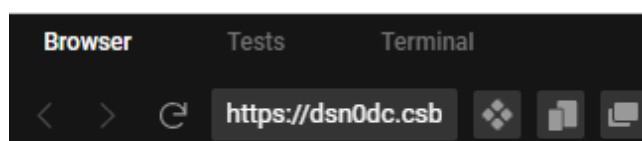
まずは、JSX 記法の復習をして Codebox 上で簡単な HTML を出力することができたら、順にフォーム部品を画面表示して確認していきましょう。

問題 1 JSX 記法 ～ CodeSandbox で JSX 記法を用いて、簡単な HTML を出力する

CodeSandbox を使って、index.js に JSX 記法を用いてコードを記載し、次の HTML が表示されることを確認してください。

▶ 表示する HTML

```
<h1>こんにちは！</h1>
<p>お元気ですか？</p>
```



こんにちは！

お元気ですか？

問題 2 map メソッドの使い方 ～ 配列の要素を map メソッドを使って書き出す

以下のように定義した配列を使って、

```
const animals = ["Dog", "Cat", "Rat"];
```

index.js 内に map メソッドを使用して、画面に「Hello, Dog Hello,Cat Hello,Rat」と配列の中身を表示させてください。

>> 画面表示

配列の操作

- Hello, Dog
- Hello, Cat
- Hello, Rat

index.js

```
src > index.js > ...
1  import React from "react";
2  import ReactDOM from "react-dom";
3
4  const animals = ["Dog", "Cat", "Rat"];
5
6  const App = () => {
7    const helloAnimals =
8
9    return (
10      <>
11        <h3>配列の操作</h3>
12        <ul>{helloAnimals}</ul>
13      </>
14    );
15  };
16
17  ReactDOM.render(<App />, document.getElementById("root"));
18
```

mapメソッドを使用して
配列の中身を「Hello, 要素」
の形式で書き出す

復習 ～ mapメソッドの使い方

▶ 押さえておきたいTips: 「mapメソッド」

map 関数は、配列に対して「**配列.map**」という形で使用します

下の例を見てみましょう

```
// 配列を定義
const nameArr = ["鈴木", "佐藤", "田中"];

// 引数で配列の値を設定、returnで返却
const nameArr2 = nameArr.map((name) => {
  return name;
});
```

「配列.map」の形

配列の要素の書き出し
(for文の代わり) をアロー関数を1行で

```
// 配列を定義
const nameArr = ["鈴木", "佐藤", "田中"];

// 引数で配列の値を設定、returnで返却
nameArr.map((name) => console.log(name));
```

問題3 input, textarea の作成 : フォーム部品①～input と textarea タグ

以下のような input タグと textarea タグを使って、

```
<label htmlFor="456">ラベル</label>
<div>
  <input
    id="123"
    placeholder="こんにちは"
    value={val}
    onChange={(e) => setVal(e.target.value)}
  />
  <textarea
    id="456"
    placeholder="こんにちは"
    value={val}
    onChange={(e) => setVal(e.target.value)}
  />
</div>
```

index.js 内で、value と onChange に対して State を連携させるようにして画面例のようなユーザ情報を表示させてください。

>> 画面表示



ユーザ情報

名前 :
名前を入力してください

メールアドレス :
メールアドレス

コメント :
コメントあれば

クリア

index.js でコンポーネントの更新する為に、useState という関数を使う場合、以下のように読み込みましょう。

```
import { useState } from "react";
```

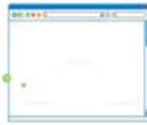
State とは

- State は、コンポーネントの状態を表す値

「状態」は全てStateとして管理し、更新処理を行うことで動的な表現をする

☞ 画面に表示するデータや、可変の状態をStateとして管理

状態管理



cf) エラーはあるか、ボタンを押せるか
テキストボックスに何を入力したか 等

- ▶ useState (ReactHooksと総称される機能群にある関数)

```
const [ num, setNum ] = useState();
```

1つ目 : State変数

2つ目 : Stateを更新するための関数

▶ useStateの import
import { useState } from "react"

問題 4 ラジオボタンの作成 : フォーム部品②～radio ボタン

以下のような input タグを使って、

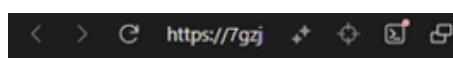
```
const [fruit, setFruit] = useState("Apple");
const onChange = (e) => setFruit(e.target.value);

return (
  <>
    <h3>React Code Practice</h3>
    <input
      type="radio"
      value="Apple"
      checked={fruit === "Apple"}
      onChange={onChange}
    />
    Apple
  </>
)
```

解答として、mapメソッドを使うことも可

index.js 内で、value と onChange に対して State を連携させるようにして画面例のようなラジオボタンを表示させてください。

>> 画面表示



React Code Practice

☐ Apple ☐ Cherry ☒ Banana

問題 5 チェックボックスの作成 : フォーム部品③～checkbox ボタン

以下のような input タグを使って、

```
<label htmlFor="my-check">チェック</label>
<input
  type="checkbox"
  id="my-check"
  value={isChecked}
  onChange={(e) => toggleChecked(e)}
/>
<div>{!isChecked ? "ON!" : "OFF!"}</div>
```

index.js 内で、value と onChange に対して State を連携させるようにして画面例のようなチェックボックスを表示させてください。

>> 画面表示

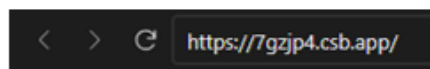
チェックなしの場合



React Code Practice

同意しますか? ☐
同意しません

チェックを付けた場合



React Code Practice

同意しますか? ☒
同意します

問題 6 プルダウンの作成 ~ フォーム部品④～select と option タグ

以下のような input タグを使って、

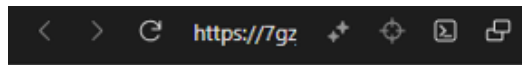
```
<select value={selected} onChange={(e) => setSelected(e.target.value)}>
  <option value="Apple">Apple</option>
  <option value="Banana">Banana</option>
  <option value="Cherry">Cherry</option>
</select>

<div>選択された果物 : {selected}</div>
```

解答として、mapメソッドを使うことも可

index.js 内で、value と onChange に対して State を連携させるようにして画面例のようなプルダウンを表示させてください。

>> 画面表示



React Code Practice

Cherry ▼

選択された果物 : Cherry

=====
今回の内容はいかがだったでしょうか。スクリプトを実行しながら動作確認できると面白いなと感じて頂けた方もいらっしゃるかもしれません。自分にできる範囲のものから少しずつ JavaScript にも挑戦してみようかなと思っていたいただければ幸いです。

以上となります。

参考文献：

- ・岡田 拓巳『モダン JavaScript の基本から始める React 実践の教科書』Informatics&IDEA（2021 年）