

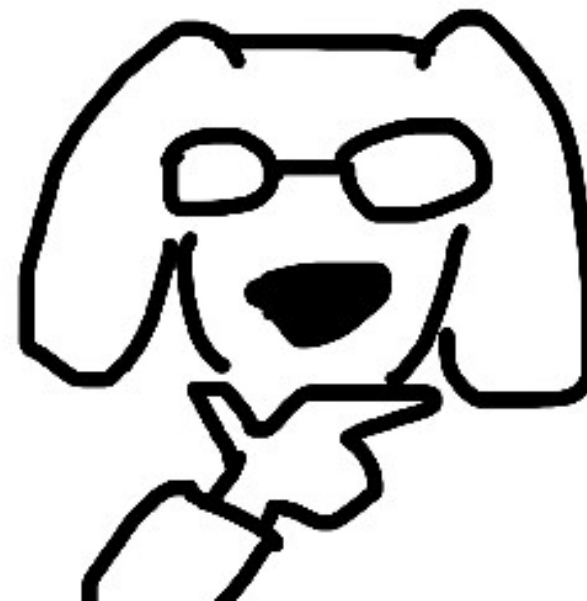
# コンポーネント設計の学び方をわかりやすく説明する

## 自己紹介

フリーランスエンジニア5年目

Github: r-sugi (経歴記載済み)

Zenn: rsugi



## 投稿済の記事

- what  
コンポーネント設計の学び方？
- how  
有名ライブラリを真似しよう！

フロントのコンポーネント設計の学び方  
を分かりやすく説明する



# 記事の一部抜粋

## 概要


React.jsでのコンポーネント設計について、チュートリアル内の下記のリストを例に考えたい。  
初めてコンポーネントを作成した際は、コンポーネントのベストプラクティスを理解せずに実装してしまっていた。

今回は可読性と再利用性についてリファクタリングしつつ説明していく。

```
const listItems = people.map(person =>
  <li key={person.id}>
    <img
      src={getImageUrl(person)}
      alt={person.name}
    />
    <p>
      <b>{person.name}</b>
      { ' ' + person.profession + ' ' }
      known for {person.accomplishment}
    </p>
  </li>
);
```

リストのレンダー – React

The library for web and native user interfaces

 [ja.react.dev](https://ja.react.dev)



## 結論


```
import { people as chemists } from "../data.js";

import ScientistProfileList from "./scientistProfileList.js";
import ScientistProfileListItem from "./scientistProfileListItem.js";
import ScientistProfileListItemIcon from "./scientistProfileListItemIcon.js";
import ScientistProfileListItemSummary from "./scientistProfileListItemSummary.js";

export default function ScientistsIndex() {
  return (
    <ScientistProfileList>
      {chemists.map((chemist) => (
        <ScientistProfileListItem key={chemist.id}>
          <ScientistProfileListItemIcon
            imageId={chemist.imageId}
            alt={chemist.name}
          />
          <ScientistProfileListItemSummary chemist={chemist} />
        </ScientistProfileListItem>
      ))}
    </ScientistProfileList>
  );
}
```

[codesandbox](#) ソースコード

# 今回のアジェンダ

- what  
コンポーネント設計で悩んだらどうするか？
- how  
有名ライブラリを真似しよう！
- **Why**   
なぜこのアプローチをとるのか？  
3パターン紹介します

## なぜこのアプローチをとるのか？(1/3)

レビュワーからの的確なコメントをもらえるとは限らないから

レビュイー（意識高い）

レビュワーからコメントもらったら改善したい！🤠

既存のコードを参考にした、レビューOKなはず！😓

↓

レビュワー

ライブラリを使ったことはあるけど、フロントエンド分からん🧑（興味もない）

既存のコードと同じ感じだし、表示が問題ないからApprove!

## なぜこのアプローチをとるのか？(2/3)

レビュワーがベストプラクティスを知らないことがあるから

レビューー

このタスクは早めに終わらせたい 🙄

既存のコードに寄せとけばレビューOKなはず！ 🙄

↓

レビュワー（意識高い）

(ベストプラクティスとか知らんけど)オレ的にOK! ✅

## なぜこのアプローチをとるのか？(3/3)

改善コメントをしやすい/理解しやすいから

レビュイー

「わかりました！」

「今回はMaterial UIのXXXコンポーネントを参考にしています」

↑ ↓

レビューワー

「このMaterial UIのXXXコンポーネントを参考にしよう」

「XXXコンポーネントと命名規則やI/Fを寄せよう」



## まとめ

- what

コンポーネント設計で悩んだらどうするか？

- how

有名ライブラリを真似しよう！

- **Why**

なぜこのアプローチをとるのか？

- レビューーからの的確なコメントをもらえるとは限らないから
- レビューーがベストプラクティスを知らないパターンがあるから
- 改善コメントをしやすい/理解しやすいから