Reactスタイリング、ライブリーなど
など

スタイリング法方

CSS

メリット

- CSSそのまま
- CSSに慣れている人にすぐ使える

デメリット

- CSS
- Tailwind CSS, Uno CSS

CSS Modules

メリット

- CSSそのまま
- classが他のファイルに混合しない

デメリット

- ほぼCSS

CSS-in-JS

- Styled Components
- Emotion
- Panda
- StyleX
- Vanila Extract
- https://danielnagy.me/posts/Post_jt4adn0o5bnr
- https://playfulprogramming.com/posts/why-is-css-in-js-slow

メリット

- 使いやすい（かも）

デメリット

- FOUC

コンポーネントライブラリー

メリット

- 統一感
- 大体ベストプラクティスがついている（アクセシビリティなど）

デメリット

- カスタマイズ性
- 使いにくい、学習する必要あり

CSSのみ

- Bootstrap
- Semantic UI
- DaisyUI

ヘッドレス（JSのみ）

- headless-ui
- react-aria
- Ark UI
- mui
- Radix UI

スタイル付きライブラリー

- React Bootstrap
- Ant Design
- shadcn/ui
- MaterialUI
- Mantine
- NextUI
- Prime React
- Fluent UI
- Tailwind Elements
- chakra

React以外にも使える

Vue/Svelte/SolidJSなど

- Park UI
- Agnostic UI
- Flowbite
- headless-ui
- Ark UI

React Native

- gluestack-ui
- Tamagui

選択するにあたって考慮する分野

実装する側

- 提供するComponentsが使い用途に充実したかどうか
- Server Componentsの対応
- Design Systemの対応
- カスタマイズ生
- 安定性
- DX・使いやすさ
- コード量
- フォームライブラリー相互性
- 学習曲線
- トークン対応
- チームメンバー

使う側

- アクセシビリティ
- 見た目・UX・雰囲気
- テーマ・ダークモード対応
- パフォーマンス

引用

- https://2023.stateofreact.com/en-US/libraries/component-libraries/#css_tools
- https://github.com/jxom/awesome-react-headless-components
- https://backlight.dev/mastery/best-react-component-libraries-for-design-systems
- https://github.com/brillout/awesome-react-components