



**G's ACADEMY**  
**TOKYO**

# Gs React 07



# React+Styled-Components

完成コード以下

[https://github.com/mitunori/  
styled-components](https://github.com/mitunori/styled-components)

# 本日のゴール

# 本日の完成図

Gsアカデミーへようこそ



[ジーズ申し込みはこちらへ](#)

Styled-Componentsを使い画面構築を行うこと

# 課題について

# Styled-Componentsを使いレイアウトする

1.Styled-Componentsを利用して画面を構築する

2.

認証を利用して、「ログインした人」と「していない人」でページを切り分ける

**※認証は別資料！**

Styled-Componentsを使い画面構築、認証を行うことがゴールです👏

# 事前設定



# ターミナルで以下を行う

```
npm install styled-components
```

**注意！！！！**

**お使いのPC（プロキシ）などの問題で誰か一人でも設定ができない場合は**

**codesandboxで対応を行います(ブラウザ上でReactを学べるもので、初回の時に触りました)**

**※一人一人の環境が違うことはこちらでは全て把握できないのでご了承ください**

**必要な機能をインストールし、package.jsonに追加されていればOKです😊**

# 特徴

# Styled-Componentsの特徴

1.記述の仕方が「今までのCSS」と似ている

- CSS部分の書き方は**ほとんど同じ**です。

2.

ファイル単位で記述できるので、**クラス名の重複が発生**しない

- 今までのCSSの場合はクラス名が重複してしまう問題を回避できます。

3.

高速なパフォーマンス

- Reactの仮想DOMと同じ仕組みでレンダリング（画面に表示）されるので**従来より高速**です。

4.

propsを活用することで、Aの場合、Bの場合といった**動的（つまり動作に応じた変化）を適用**

**しやすい**

- propsを用いて、cssを設定できる（厳密には渡すという表現）

# Styled-Componentsの少し残念なところ

## 1.HTML(JSX)が少し見づらい

- Styled-Componentsで独自の名前をつけてHTMLを書くので、初見の人は??となりやすい

## 2.CSSの使い回しがしづらい

- ファイル単位（コンポーネント）で使用するため、同じような記述を他の**パーツで使う場合は再度CSSを記述してStyled-components**の設定をしないといけない

## 3. エメットのようなショートカットで書けない

- html,cssで今まで使えていたエメットのようなショートカットができないので、自分で記述しないといけない

# コード実装

# コード記述

```
App.jsx x
src > App.jsx > ...
1  import { useState } from "react";
2  import reactLogo from "../assets/react.svg";
3  import styled from "styled-components";
4
5  import "../App.css";
6
7  const Test = styled.div`
8    background: red;
9  `;
10
11 function App() {
12   const [count, setCount] = useState(0);
13
14   return (
15     <div className="App">
16       <Test>aaa</Test>
17     </div>
18   );
19 }
20
21 export default App;
22
```

次ページに続きます👉

# コード記述

```
import styled from "styled-components";
```

**import**を用いて、Styled-Componentsを使えるようにします👏

使いたい場所で、**必ず**「import」をする必要があります👏

## ポイント！

この一行を使用したいパーツの**先頭あたりでimport**しておいただけなのでコピーで貼り付けていきましょう！（先頭と書いていますが、場所はどこでもOKです！今回は**忘れないようにという意味**で先頭と記述しているだけです）

# コード記述

```
const Test = styled.div`  
  background: red;  
`;  
;
```

## 命名のルールと特徴について

1.

const Test と書いている[Test]のTの部分は大文字（**先頭は必ず大文字！絶対に！**）

例) 先頭だけ大文字にする必要があります

```
const MoritaTag = styled.div
```

```
const OohoriTag = styled.p
```

2.

**styled.div**と書いている箇所ですが

**styled.h1**にすると⇒ h1タグの意味になります👏

**styled.p**にすると⇒ pタグの意味になります👏

つまり[styled.xx] のxxのところに[htmlのタグ]を記述するとそのタグになります👏

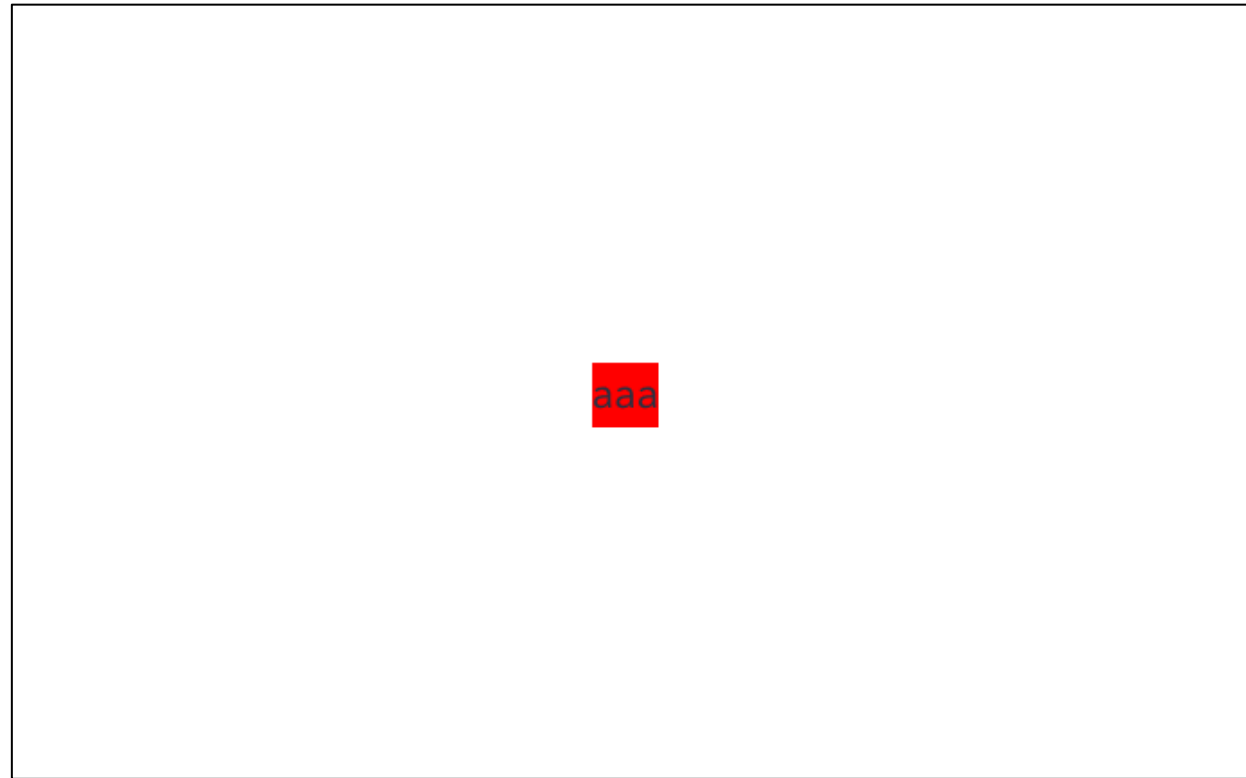


# コード記述

```
return (  
  <div className="App">  
    <Test>aaa</Test>  
  </div>  
);
```

先ほど作成した[const Test = style.div] の変数がStyled-componentsのcssが設定されているのでそれを使用したい箇所に上記のように設定してあげるとCSSが適用されます👏

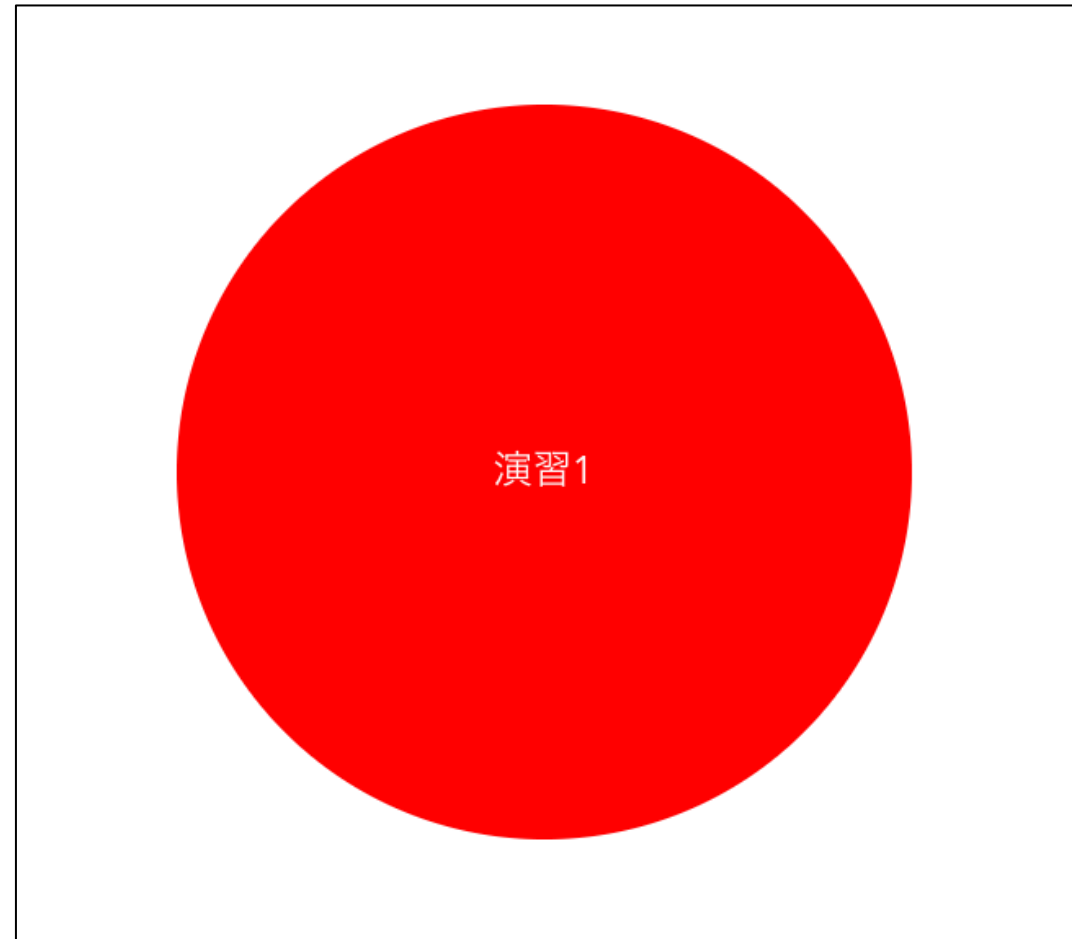
# 表示されると以下ようになります



このように「ファイル単位」= コンポーネント（パーツ）単位でCSSを設定できるので、  
その**コンポーネントを見れば**「CSSの設定」も全て一つにまとめることができるのがポイントです👏

# 演習1

# 以下のものを作成しましょう！



cssは以下で設定していますが、中央にレイアウトする方法はご自身で調べてみましょう🙌

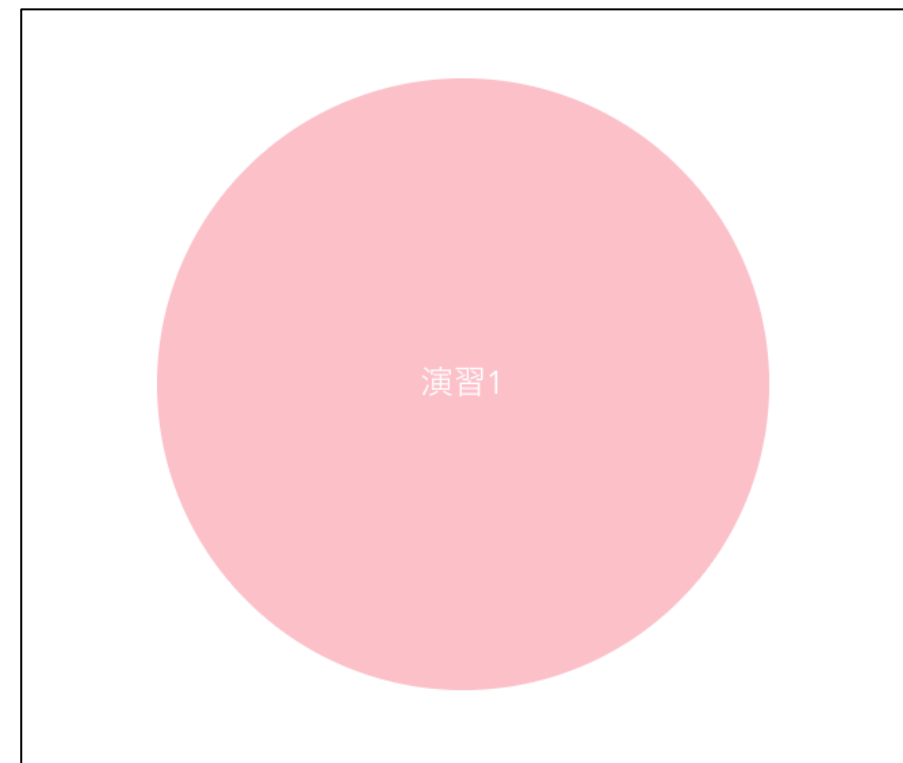
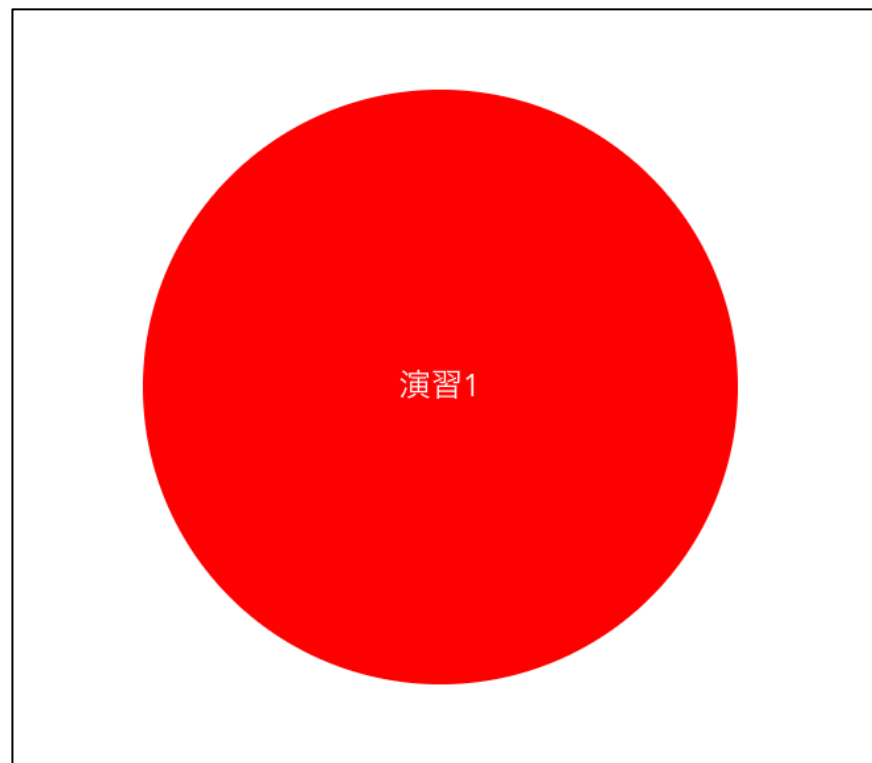
※正解は一つではなく、複数あります（中央レイアウト）

それ以外のスタイルは以下の設定（コードは正解後共有します）

```
background: red;  
color: #fff;  
width: 300px;  
height: 300px;  
border-radius: 50%;
```

propsで動的変化しよう

# 以下のものを作成しましょう！



先ほど作成したものを活用し、動的に変化をさせたいと思います👐  
次に行く

# コード記述

```
const Test = styled.div`
  color: #fff;
  width: 300px;
  height: 300px;
  border-radius: 50%;
  display: flex;
  align-items: center;
  justify-content: center;

  background: ${(props) => (props.pink ? "pink" : "red")};
`;

function App() {
  return (
    <div className="App">
      <Test pink>演習1</Test>
    </div>
  );
}
```

`$(props)} => (props.pink) ? 'pink' : 'red')};`

この **?** と **:** が続く記述のことを「**三項演算子**」といいます😊(Reactでよく見る**パターン**)

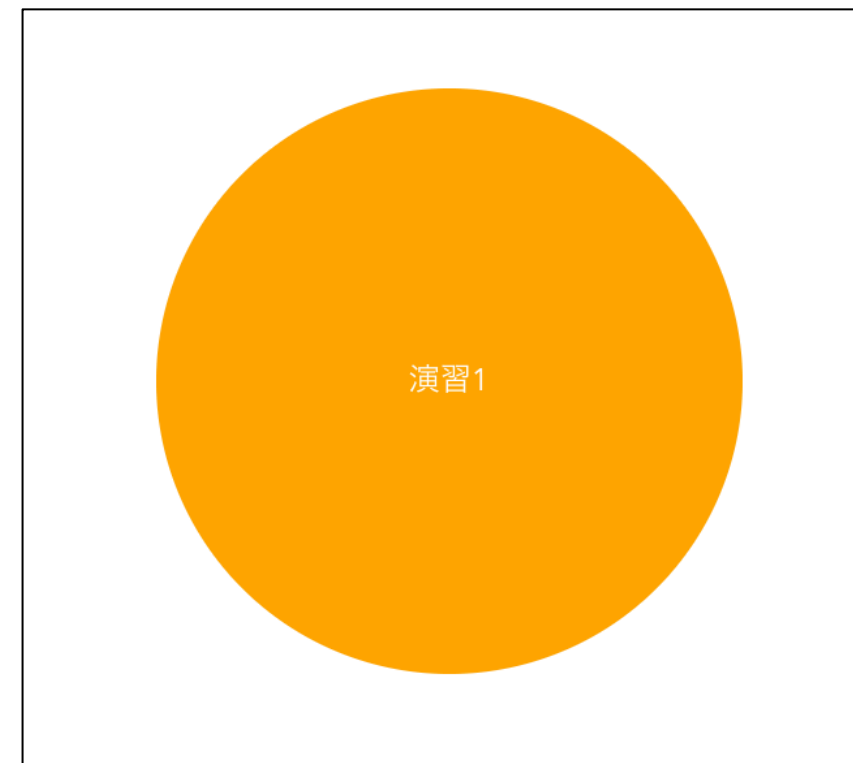
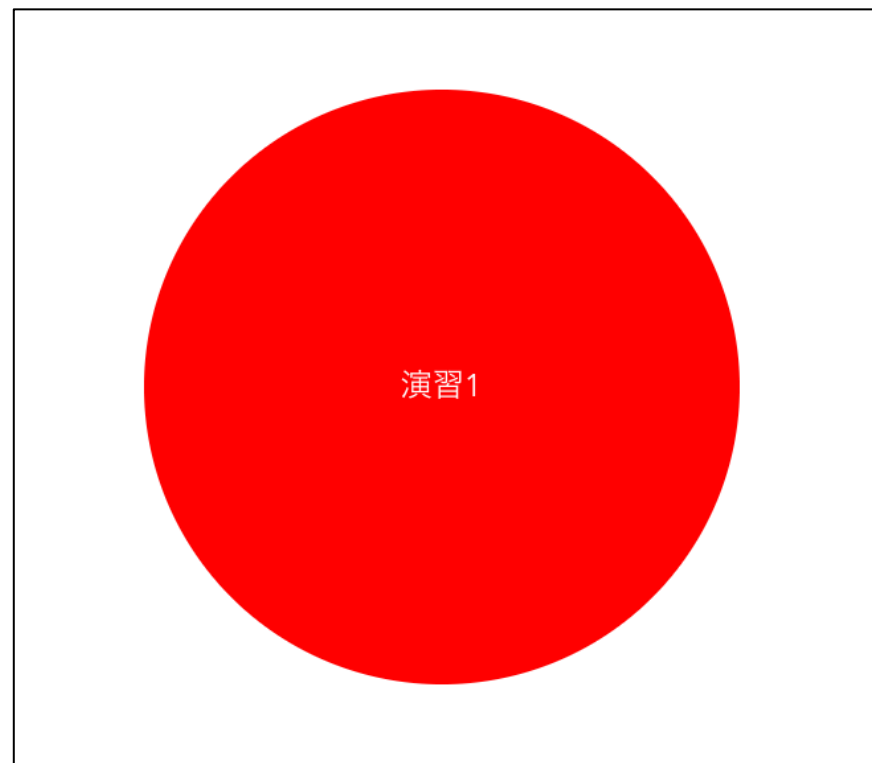
つまり⇒

「タグ」の箇所に[pink]と書いていたらそれがStyled-componentsに文字列が渡され  
props.pinkでアクセスできます😊 props.pinkがあったら⇒ [pink]、なければ[red]にする

# 演習2



# 以下のものを作成しましょう！

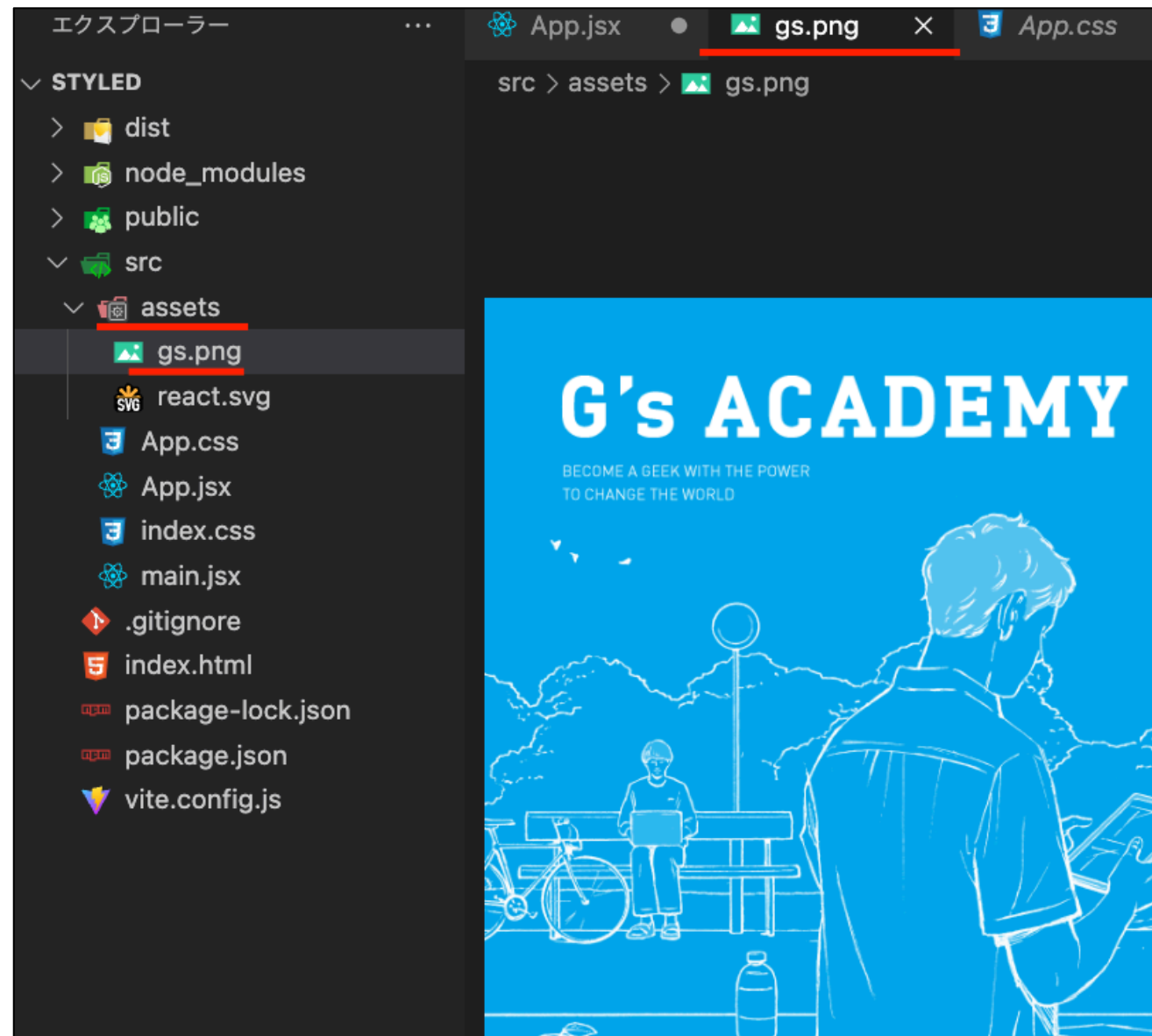


propsで動的に変化させる方法を用いて、  
タグの箇所に[orange]と書いたときに「オレンジ色」になるように設定してみましょう👉(赤線の箇所)

```
function App() {  
  return (  
    <div className="App">  
      <Test orange>演習1</Test>  
    </div>  
  );  
}
```

# 画像の表示

# コード記述



## ファイルの置き場所

src/assets/好きなファイル名

今回はgs.pngというものを置いています👉

# コード記述

```
1 import { useState } from "react";
2 import reactLogo from "./assets/react.svg";
3 import styled from "styled-components";
4 import gsBg from "./assets/gs.png";
5
6 import "./App.css";
7
8 const Test = styled.div`
9   color: #fff;
10  width: 300px;
11  height: 300px;
12  border-radius: 50%;
13  display: flex;
14  align-items: center;
15  justify-content: center;
16
17  background: ${(props) => (props.orange ? "orange" : "red")};
18 `;
19
20 function App() {
21   return (
22     <div className="App">
23       <img src={gsBg} />
24     </div>
25   );
26 }
27
28 export default App;
```

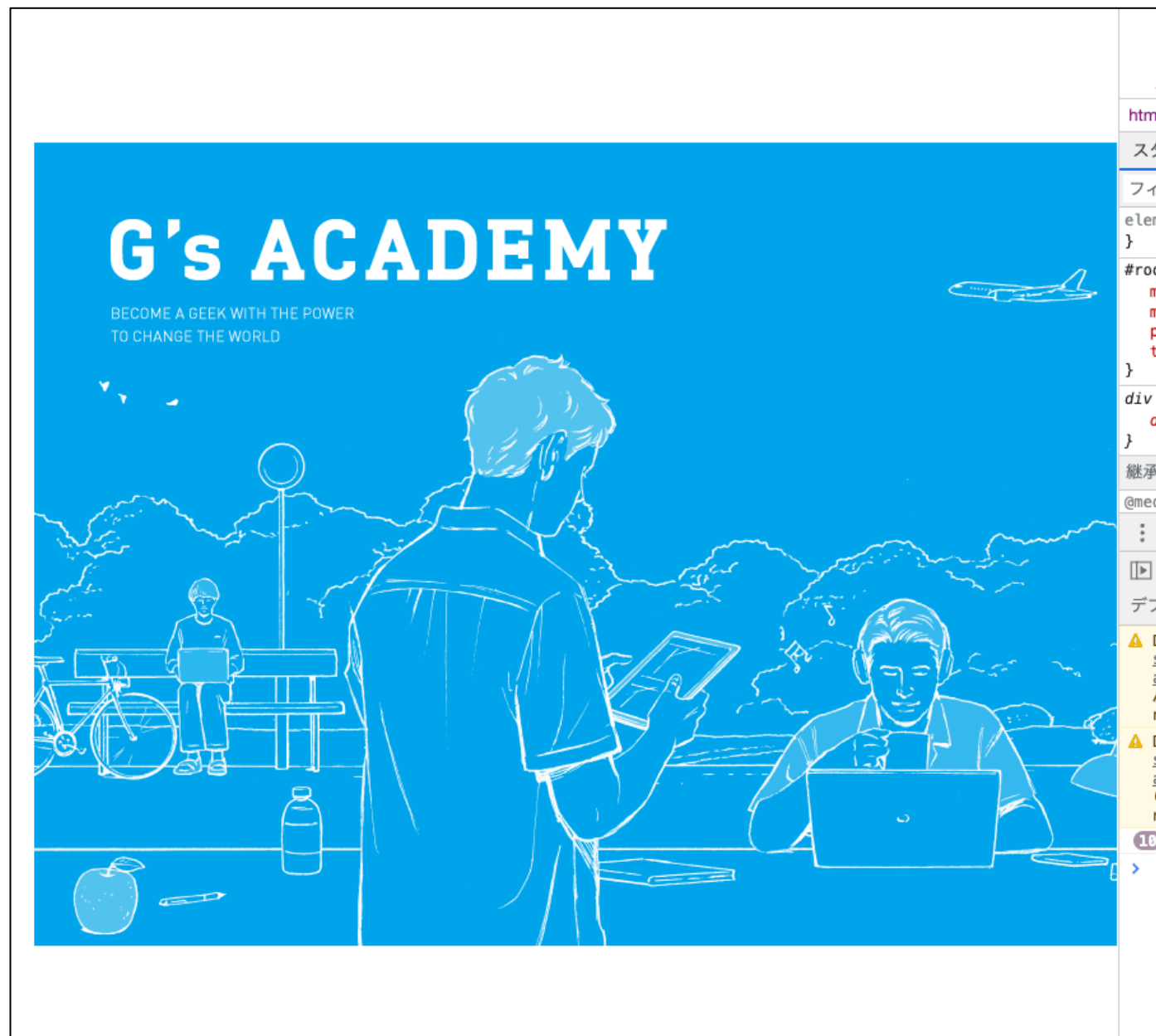
import xxx from パス

## ファイルの置き場所

gsBgと書いているところは好きな名前でもOKです👉 (ご自身でわかりやすい名前ならOK)

使用する際は<img src={gsBg} />という風に srcの中に設定してあげましょう👉

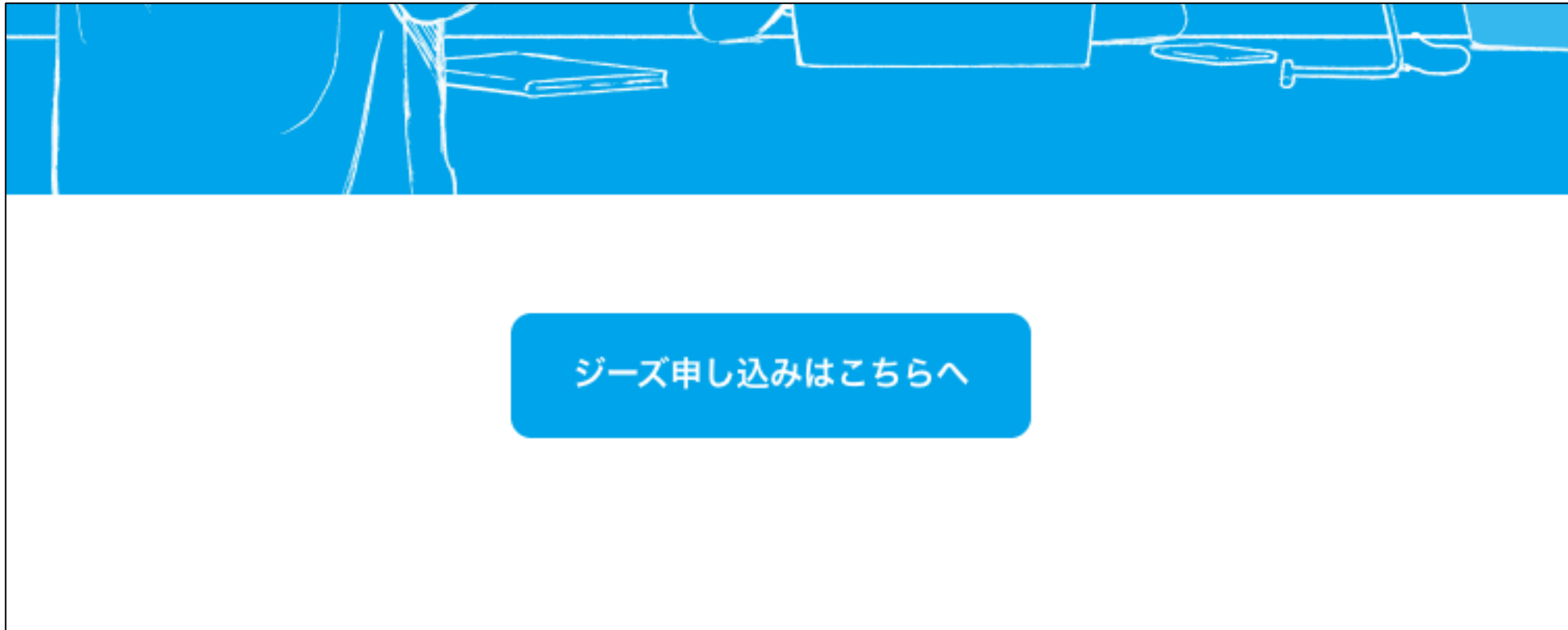
# 表示されると以下



表示されるとこのようになっています👉

# 演習3

# 以下のものを作成しましょう！



## ヒント

Styled-componentsを用いて「aタグ」を使用します👉

Cssは以下のようにしてください👉（正解は他にもありますが、一旦授業ではこちらで）

⇒書くスペースがないので次ページに貼り付けます

# CSSは以下のようにコピペで使う

```
display: flex;  
align-items: center;  
justify-content: center;  
width: 250px;  
height: 60px;  
margin: 50px auto;  
background: #03a7ea;  
color: #fff;  
border-radius: 10px;  
font-weight: bold;
```

```
&:hover {  
  color: #fff;  
  opacity: 0.5;  
}
```



まとめ

Styled-Componentsを用いて

cssを適用させる方法を理解する 🤗

ショートカットなどが効かなくて慣れるのが

大変かもしれませんが活用してみましよう 🤗

1.使用する際はimportすることを忘れない

2.説明書（ドキュメントやリファレンス）などは  
必ず見ましょう！

3.困ったら[Styled-Components]の記事を探しまし  
よう！

Styled-Componentsを体感していただきました

た😊

さまざまなCSSのやり方がありますが有効活用してみましよう😊

※なるべくわかりやすい表現やかなり噛み砕いた言い回しにしていますが、あくまでも一つの意見として参考にしてみてください  
違う方や他の記事で色々な解説や説明などもありますので、それらと見比べながらやってみると理解が深まるかと思います