

# React

## ◆ 컴포넌트 스타일링-2

정수아

# Contents

**01** Post CSS

**02** styled-components

**03** 비교



01

# PostCSS

# Post CSS

## ❖ Post CSS

- CSS를 불러와서 사용할 때 클래스 이름을 고유한 값 형태로 자동으로 만들어 컴포넌트 스타일 클래스 이름이 중복되는 현상을 방지해주는 기술

- 값 형태

```
파일이름_클래스이름_해시값
```

- 사용 방법

- 버전 2 이상부터는 별도의 설정이 필요 없고, 다음과 같은 확장자로 파일을 저장하기만 하면 적용

```
파일이름.module.css
```

# [실습] Post CSS

## ❖ App.js

```
import CSSModule from "../ComponentStyle/CSSModule";

function App() {
  return (
    <CSSModule />
  );
}

export default App;
```

# [실습] Post CSS

## ❖ CSSModule.js

```
import React from "react";
import styles from "./CSSModule.module.css";

const CSSModule = () => {
  console.log(styles);

  return (
    <div className={styles.container}>
      <span className="span">CSS Module</span> 실습
    </div>
  );
};

export default CSSModule;
```

# [실습] Post CSS

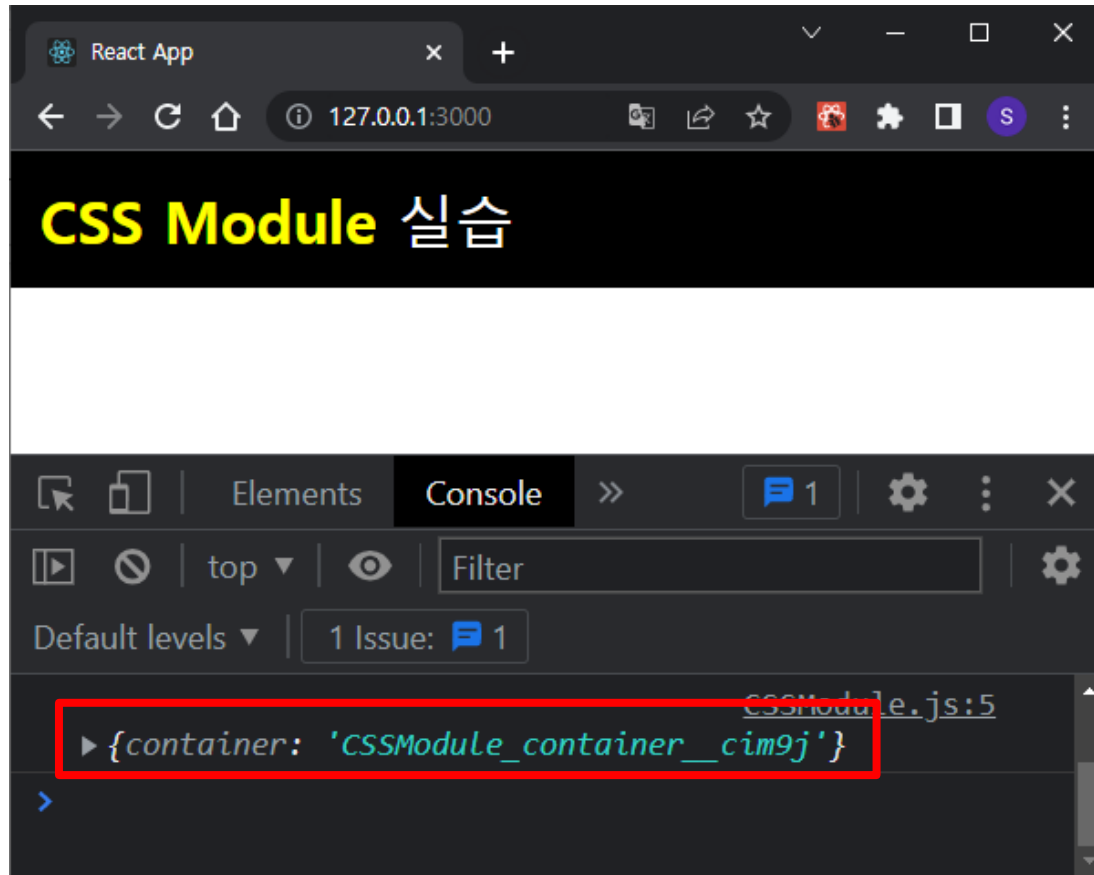
## ❖ CSSModule.module.css

```
.container {  
  background: black;  
  padding: 1rem;  
  font-size: 2rem;  
  color: white;  
}
```

```
/* 전역 스타일 */  
:global .span {  
  color: yellow;  
  font-weight: bold;  
}
```

# [실습] Post CSS

## ❖ 실행 결과





# [실습] global 스타일

## ❖ App.js

```
import GlobalStyle1 from "../ComponentStyle/GlobalStyle1";
import GlobalStyle2 from "../ComponentStyle/GlobalStyle2";

const App = () => {
  return (
    <div>
      <GlobalStyle1 />
      <GlobalStyle2 />
    </div>
  );
};

export default App;
```

# [실습] global 스타일

## ❖ GlobalStyle1.js

```
import styles from "../GlobalStyle1.module.css";

const GlobalStyle1 = () => {
  return (
    {/*CSS Module 사용할 스타일*/}
    <div className={styles.container}>

      <h1>컴포넌트1</h1>

      {/* globalBtn : 전역 스타일*/}
      <button className="globalBtn">전역스타일</button>
    </div>
  );
};

export default GlobalStyle1;
```

# [실습] global 스타일

## ❖ GlobalStyle2.js

```
const GlobalStyle2 = () => {  
  return (  
    <div className="container">  
      <h1>컴포넌트2</h1>  
      <button className="globalBtn">전역스타일</button>  
    </div>  
  );  
};  
  
export default GlobalStyle2;
```

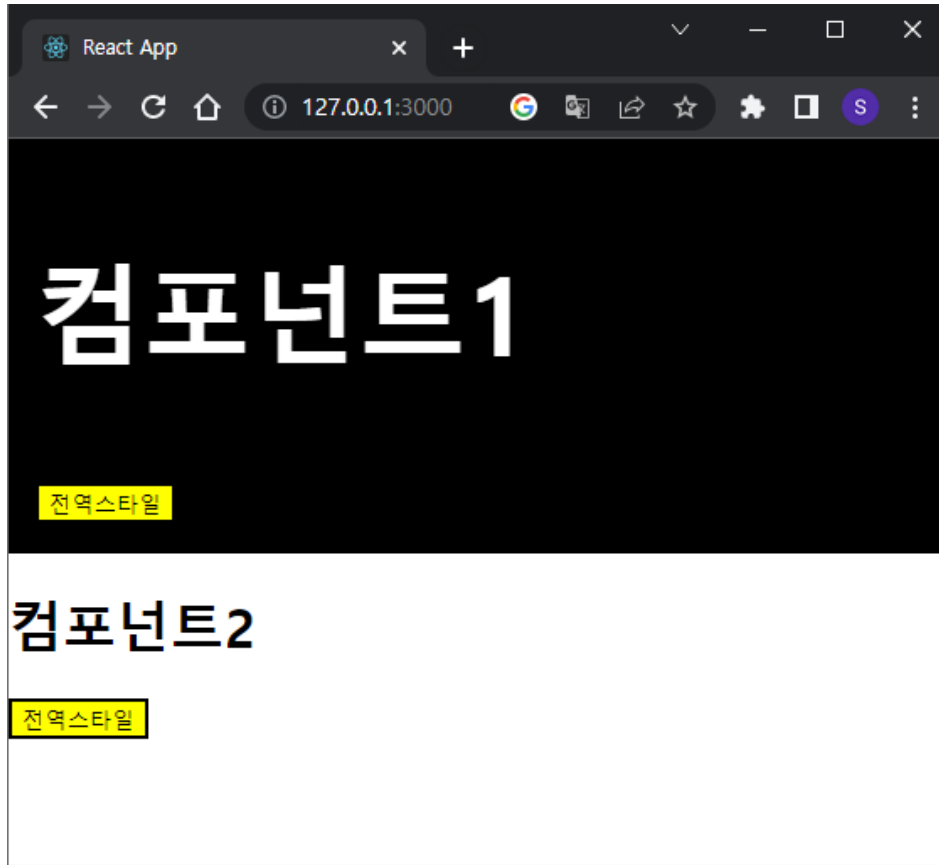
# [실습] global 스타일

## ❖ GlobalStyle1.module.css

```
.container {  
  background: black;  
  padding: 1rem;  
  font-size: 2rem;  
  color: white;  
}  
  
:global .globalBtn {  
  background-color: yellow;  
}
```

# [실습] global 스타일

## ❖ 실행 결과



# 클래스 이름을 두 개 이상 적용

## ❖ CSS Module을 사용한 클래스 이름을 두 개 이상 적용

```
<div className="container title">  
  <span className="span">CSS Module</span> 실습  
</div>
```



```
<div className={` ${styles.container} ${styles.title} `}>  
  <span className="span">CSS Module</span> 실습  
</div>
```

# [실습] 클래스 이름을 두 개 이상 적용

## ❖ CSSModule.module.css

```
.container {  
  background: black;  
  padding: 1rem;  
  font-size: 2rem;  
  color: white;  
}  
  
.title {  
  border: 5px solid aqua;  
}  
  
:global .span {  
  color: yellow;  
  font-weight: bold;  
}
```

# [실습] 클래스 이름을 두 개 이상 적용

## ❖ CSSModule.js

```
import React from "react";
import styles from "../CSSModule.module.css";

const CSSModule = () => {
  console.log(styles);

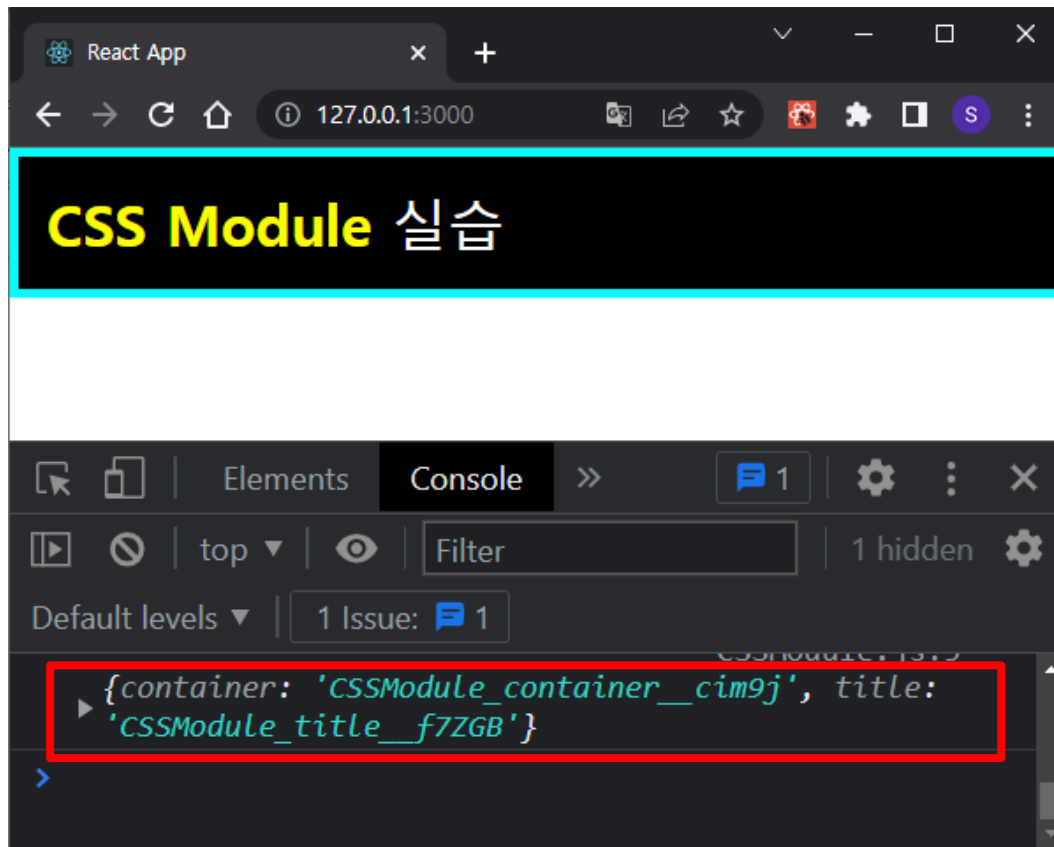
  return (
    <div className={` ${styles.container} ${styles.title}`}>
      <span className="span">CSS Module</span> 실습
    </div>
  );
};

export default CSSModule;
```



# [실습] 클래스 이름을 두 개 이상 적용

## ❖ 실행 결과



# Sass와 함께 사용하기

## ❖ CSS Module + Sass

- 사용 방법

```
파일이름.module.scss
```

# [실습] CSS Module + Sass

## ❖ CSSModule.module.scss

```
.container {  
  background: black;  
  padding: 1rem;  
  font-size: 2rem;  
  color: white;  
  
  &.title {  
    border: 5px solid greenyellow;  
  }  
}  
  
:global {  
  .span {  
    color: yellow;  
    font-weight: bold;  
  }  
}
```

# [실습] CSS Module + Sass

## ❖ CSSModule.js

```
import React from "react";
import styles from "./CSSModule.module.scss";

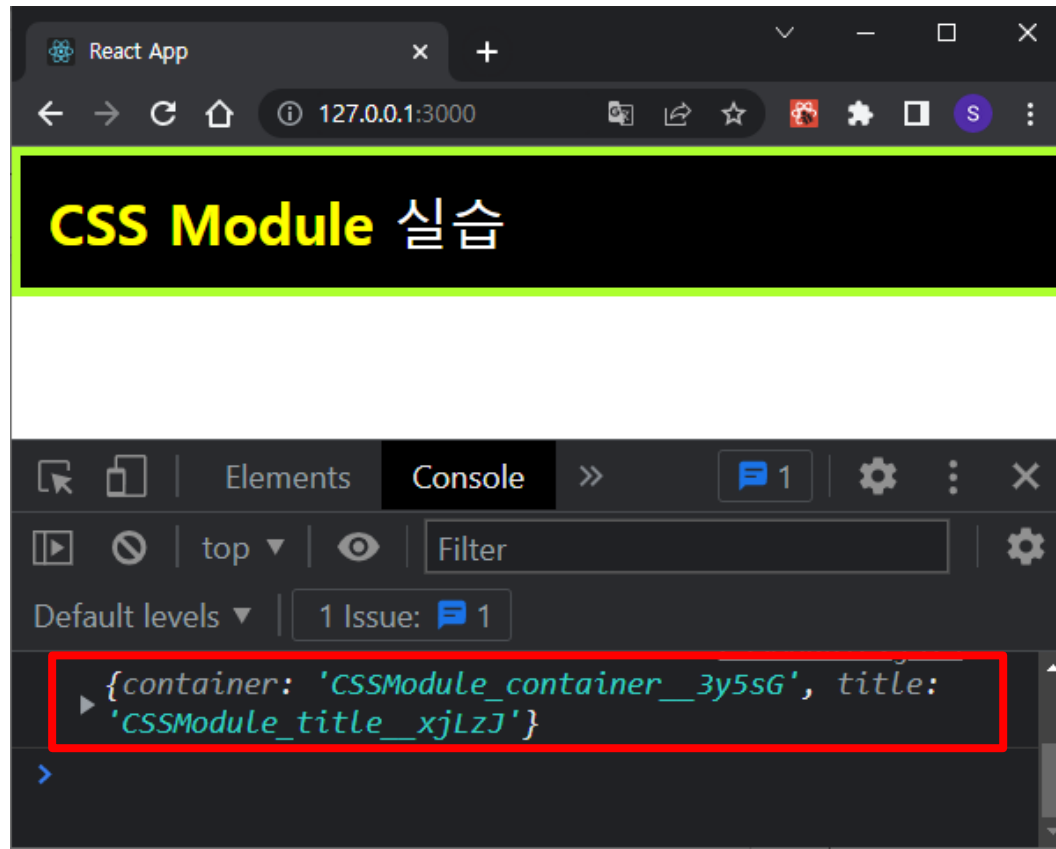
const CSSModule = () => {
  console.log(styles);

  return (
    <div className={` ${styles.container} ${styles.title}`}>
      <span className="span">CSS Module</span> 실습
    </div>
  );
};

export default CSSModule;
```

# [실습] CSS Module + Sass

## ❖ 실행 결과





02

## **styled-components**

# styled-components

## ❖ styled-components

- 자바스크립트 파일 안에 스타일을 선언하는 방식
  - CSS-in-JS
- 별도의 스타일 파일을 만들지 않아도 스타일 작성 가능
- <https://styled-components.com/>

- 설치

```
$ npm install styled-components
```

# styled-components

## ❖ vscode-styled-components 플러그인 설치

```
const Button = styled.button`
  background: ${({props) => (props.primary ? "palevioletred" : "white")});
  color: ${({props) => (props.primary ? "white" : "palevioletred")});

  font-size: 1em;
  margin: 1em;
  padding: 0.25em 1em;
  border: 2px solid palevioletred;
  border-radius: 3px;
`;
```



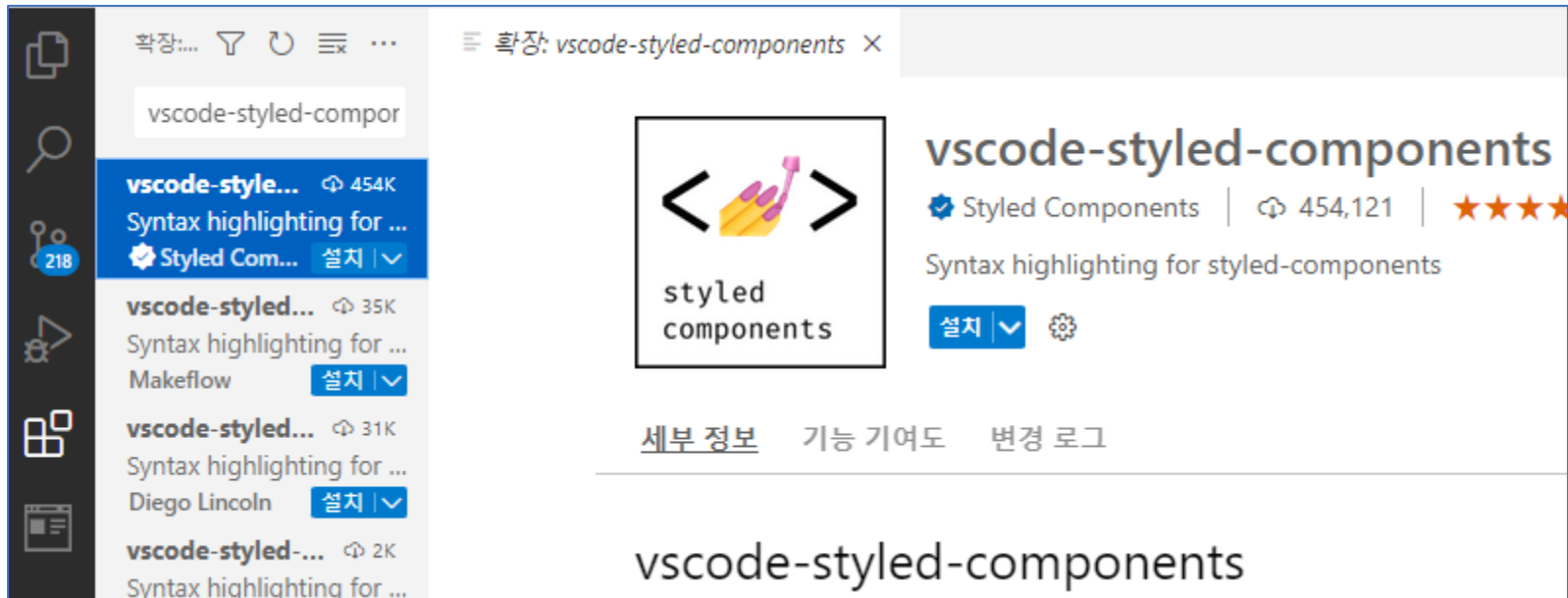
```
const Button = styled.button`
  background: ${({props) => (props.primary ? "palevioletred" : "white")});
  color: ${({props) => (props.primary ? "white" : "palevioletred")});

  font-size: 1em;
  margin: 1em;
  padding: 0.25em 1em;
  border: 2px solid palevioletred;
  border-radius: 3px;
`;
```



# styled-components

## ❖ vscode-styled-components 플러그인 설치



# styled-components

## ❖ 스타일 작성 방법

```
const 스타일_저장_변수이름 = styled.태그이름`  
  // 스타일 작성  
  font-size: 1.5em;  
  text-align: center;  
  color: palevioletred;  
`;  
;
```

- 예시

```
const Container = styled.div`  
  background: black;  
  padding: 1rem;  
  font-size: 2rem;  
  color: white;  
`;  
;
```

# styled-components

## ❖ 스타일 적용 방법

```
return (  
  <스타일컴포넌트명/>  
);
```

- 예시

```
return (  
  <Container />  
);
```

# [실습] styled-components

## ❖ App.js

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

function App() {
  const Title = styled.h1`
    font-size: 1.5em;
    text-align: center;
    color: palevioletred;
  `;

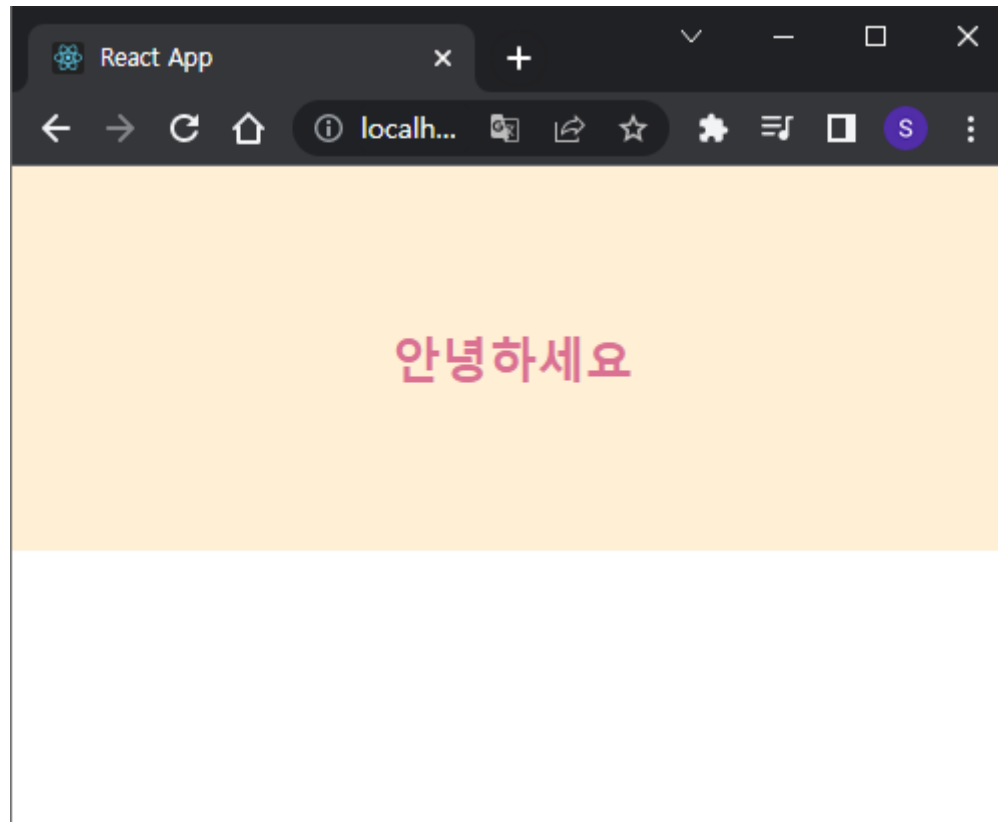
  const Wrapper = styled.div`
    padding: 4em;
    background: papayawhip;
  `;

  return (
    <Wrapper>
      <Title>안녕하세요</Title>
    </Wrapper>
  );
}

export default App;
```

# [실습] styled-components

## ❖ 실행 결과



# styled-components

## ❖ props 값에 따라 스타일 적용

- 예) props 값이 primary이면 white, 아니면 red를 적용
- 스타일 작성

```
color: ${({props}) => (props.primary ? "white" : "red")};
```

- 스타일 적용

```
<Button>Normal</Button>  
<Button primary>Primary</Button>
```

# [실습] styled-components + props

## ❖ App.js

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

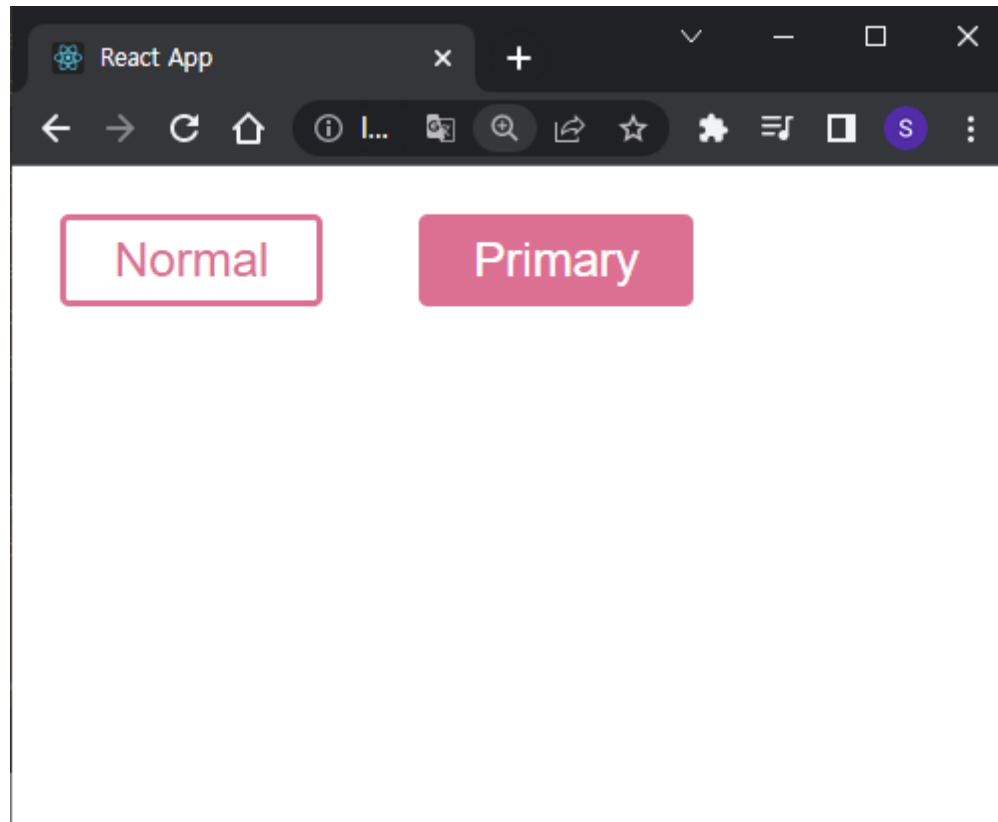
function App() {
  const Button = styled.button`
    background: ${props => (props.primary ? "palevioletred" : "white")};
    color: ${props => (props.primary ? "white" : "palevioletred")};
    font-size: 1em;
    margin: 1em;
    padding: 0.25em 1em;
    border: 2px solid palevioletred;
    border-radius: 3px;
  `;

  return (
    <>
      <Button>Normal</Button>
      <Button primary>Primary</Button>
    </>
  );
}

export default App;
```

# [실습] styled-components + props

## ❖ 실행 결과







03

비교

# CSS Module vs Styled Components

## ❖ CSS Module와 Styled Components 비교

	CSS Module	Styled Components
성능	Pure CSS	CSS in JS
사용 범위	어디서든 사용 가능	React, React-Native
특징	독립성	독립성 JS 파일에서 작성

**THANK 😊 YOU**