

# styled-components

## 1 스타일링 라이브러리

## CSS in JS

(<https://speakerdeck.com/vjeux/react-css-in-js>)

## CSS의 문제점

Global Namespace / Dependencies / Dead Code Elimination

Minification / Sharing Constants

Non-deterministic Resolution / Isolation

CSS의 문제점 (React를 쓰면서...)

Global Namespace: 글로벌 변수를 지양해야하는 JS와 대치

Dependencies: css 간의 의존 관리

Dead Code Elimination: 안쓰는 css 인지 어려움

Minification: 클래스 이름 최소화

Sharing Constants: JS의 코드와 값을 공유하고 싶음

Non-deterministic Resolution: css 파일 로드 타이밍 이슈

Isolation: 격리

스타일드 컴포넌트(styled-components)

(<https://styled-components.com/>)

(<https://styled-components.com/docs/basics>)

스타일드 컴포넌트(styled-components)

Automatic critical CSS: 자동 style injects & 코드 스플릿

No class name bugs: unique / overlap x / misspellings

Easier deletion of CSS: tied to a specific component

Simple dynamic styling: props / global theme

Painless maintenance: styling affecting your component

Automatic vendor prefixing: current standard only

## 정리

### styled-components

CSS in JS

CSS의 문제점을 해소

해결책

스타일을 style 태그로 분리

사용법(Template literals)

`styled.{element}``

styled(스타일드컴포넌트)

상속