

React 스타일링 : To do 프로젝트 스타일링

CSS의 한계를 극복하는 다양한 스타일링 방법

CSS 문제점

이름충돌

우선순위

격리 불가능

```
// a.css
.text {
  display: flex;
  color: red;
}

// b.css
.text {
  color: green;
}

.text > .icon {
  flex-grow: 1;
}
```

다양한 스타일링 방법

01

CSS Module

로컬 스코프:
각 CSS 파일이 모듈화

자동화된 클래스 이름 생성:
고유한 클래스 이름이 생성하여
이름 충돌방지

02

Emotion

코로케이션:
CSS와 JS를 함께 작성

동적 스타일링:
JS 값을 전달하여 동적 스타일링 가능

03

Tailwindcss

유틸리티-퍼스트:
사전 정의된 클래스 이름을 사용하여 빠른
개발

CSS Module

클래스 이름을 해쉬하여

이름 충돌 방지

```
// a.module.css
.text {
  display: flex;
}

// A.jsx
import styles from './a.module.css'
console.log(styles);

/*
{
  text: '_text_1I433_18'
}
*/

function Component() {
  return (
    <div className={styles.text}>
      Hello World
    </div>
  )
}
```

Emotion

CSS 스타일을 컴포넌트로
제한 하여 이름 충돌 방지

JS를 사용하여 동적 스타일
링

```
import styled from "@emotion/styled";

const size = "10px";
const Component = styled.div`
  display: flex;
  color: ${(props) => props.color};
  font-size: ${size};
`;

function App() {
  return <Component color="blue">Hello, world!
</Component>;
}
```

Tailwindcss

미리 유틸리티 클래스 제공

```
function Component() {  
  return (  
    <div className="flex w-[100px] bg-red">  
      Hello World  
    </div>  
  )  
}
```

CSS Module로 Todo List 스타일링

Emotion으로 Todo List 스타일링

Emotion 설치

```
npm install --save @emotion/react @emotion/styled //emotion 11.11.4
```

Tailwindcss로 Todo List 스타일링

Tailwindcss 설치

```
npm install -D tailwindcss postcss autoprefixer //tailwindcss 3.4.3  
npx tailwindcss init -p
```