



리액트와 함께 장고 시작하기 / 리액트

CSS를 적용하는 다양한 방법

여러분의 파이썬/장고 페이스메이커가 되겠습니다.

리액트에서의 CSS

전통적인 방법으로 CSS를 별도의 파일에 저장하고, link 태그를 통해 웹페이지에 포함을 시킬 수도 있겠지만, 이는 전역설정이 되며 컴포넌트 간에 예기치않은 스타일 적용이 발생할 수 있습니다.

컴포넌트 중심으로 생각하고, CSS도 컴포넌트 내부에서 관리/적용해봅시다.

CSS 파일을 직접 적용

```
import React from 'react';
import './Message.css';
import './Profile.css';

const Profile = () => <div className="wrapper">Profile</div>;
const Message = () => <div className="wrapper">Message</div>;

const App = () => (
  <div>
    Hello World
    <Profile />
    <Message />
  </div>
);

export default App;
```

className이 겹치기에
나중에 import한 Profile.css의 .wrapper
스타일로 덮어쓰게 됩니다. 😞

src/Message.css

```
.wrapper {
  color: pink;
  background-color: black;
}
```

src/Profile.css

```
.wrapper {
  color: gray;
  background-color: orange;
}
```

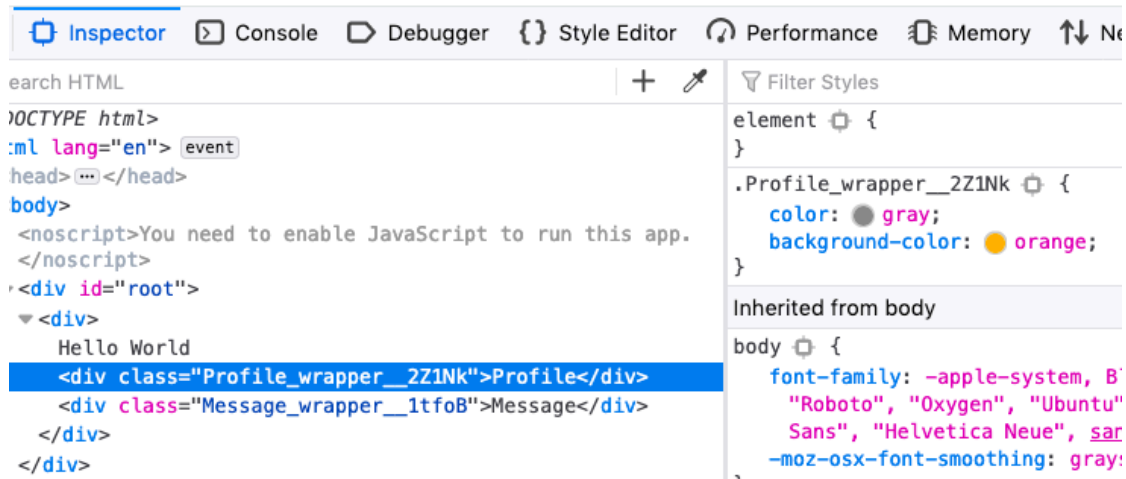
css-module 활용

```
import React from 'react';
import MessageStyle from './Message.module.css';
import ProfileStyle from './Profile.module.css';

const Profile = () => <div className={ProfileStyle.wrapper}>Profile</div>;
const Message = () => <div className={MessageStyle.wrapper}>Message</div>;

const App = () => (
  <div>
    Hello World
    <Profile />
    <Message />
  </div>
);

export default App;
```



```
src/Message.module.css
.wrapper {
  color: pink;
  background-color: black;
}
```

```
src/Profile.module.css
.wrapper {
  color: gray;
  background-color: orange;
}
```

각 클래스명에
고유한 해시값이 적용되어
클래스명 중복을 방지

Sass 활용

제가 주로 사용하는 방법

설치> yarn add --dev node-sass

```
import React from 'react';  
import MessageStyle from './Message.module.scss';  
import ProfileStyle from './Profile.module.scss';
```

나머지 동일

src/shared.scss

```
$text-color: orange;  
$background-color: blue;
```

scss/sass에서는
변수, 연산, 믹스인/인자, 중첩, 문자 보간,
임포트, 각종 함수 등을 지원

src/Message.module.scss

```
@import './shared.scss';
```

```
.wrapper {  
  color: lighten($text-color, 10%);  
  background-color: lighten($background-color, 10%);  
}
```

src/Profile.module.scss

```
@import './shared.scss';
```

```
.wrapper {  
  color: darken($text-color, 10%);  
  background-color: darken($background-color, 10%);  
}
```

<https://sass-lang.com/documentation/syntax>

css-in-js 라이브러리 중 styled-component 활용

CSS코드를 JavaScript 코드로 관리할 수 있다는 장점

설치> yarn add --dev styled-component

```
import React from 'react';
import styled from "styled-components";
```

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

Tagged Template Literals 문법

```
const Title = styled.h1`
  font-size: ${props => props.isBig ? 3 : 1.5}em;
  text-align: center;
  color: palevioletred;
`;
```

js 로직으로 CSS 스타일 결정 가능

```
const App = () => (
  <Wrapper>
    <Title>
      Hello World!
    </Title>
  </Wrapper>
);
```

```
export default App;
```

Life is short.
You need Python and Django.

I will be your pacemaker.

