



API 참고서 > 컴포넌트 >

<style>

내장 브라우저 <style> 컴포넌트를 사용하면 문서에 인라인 CSS 스타일시트를 추가할 수 있습니다.

```
<style>{` p { color: red; } `}</style>
```

- [레퍼런스](#)
 - [<style>](#)
- [사용법](#)
 - [인라인 CSS 스타일시트 렌더링하기](#)

레퍼런스

<style>

문서에 인라인 스타일을 추가하려면, [내장 브라우저 <style> 컴포넌트](#)를 렌더링하세요. 어떤 컴포넌트에서든 <style> 을 렌더링할 수 있으며, React는 [특정 경우](#)에 해당 DOM 요소를 문서의 <head> 에 배치하고 동일한 스타일을 중복 제거합니다.

```
<style>{` p { color: red; } `}</style>
```

아래 예시를 참고하세요.

Props

<style> 은 모든 공통 엘리먼트 Props를 지원합니다.

- `children`: 문자열 타입. 필수 항목. 스타일시트의 내용.
- `precedence`: 문자열 타입. 문서의 `<head>` 내 다른 요소들에 비해 `<style>` DOM 노드의 순위를 지정하여, 어떤 스타일시트가 다른 스타일시트를 덮어쓸 수 있는지를 결정합니다. React는 먼저 발견한 우선순위를 “낮게”, 나중에 발견한 우선순위를 “높게” 추론합니다. 많은 스타일 시스템은 스타일 규칙이 원자적이기 때문에 단일 우선순위 값을 사용해도 잘 작동할 수 있습니다. 동일한 우선순위를 가지는 스타일시트는 `<link>` 태그인지 인라인 `<style>` 태그인지 `preinit` 함수로 로드된 것인지와 무관하게 함께 적용됩니다.
- `href`: 문자열 타입. 동일한 `href`를 가진 스타일의 중복 적용을 제거합니다.
- `media`: 문자열 타입. 스타일시트를 특정 미디어 쿼리로 제한합니다.
- `nonce`: 문자열 타입. 엄격한 콘텐츠 보안 정책을 사용할 때 리소스를 허용하기 위한 암호화 난수입니다.
- `title`: 문자열 타입. 대체 스타일시트의 이름을 지정합니다.

다음 React 속성들은 권장하지 않습니다.

- `blocking`: 문자열 타입. "render"로 설정하면 스타일시트가 로드될 때까지 브라우저가 페이지를 렌더링하지 않도록 지시합니다. React는 Suspense를 사용하여 더 세밀하게 제어할 수 있습니다.

특별한 렌더링 동작

React는 `<style>` 컴포넌트를 문서의 `<head>`로 이동시키고, 동일한 스타일시트의 중복을 제거하며, 스타일시트가 로딩되는 동안 서스펜스 할 수 있습니다.

이 동작을 사용하려면 `href` 와 `precedence` 속성을 제공하세요. React는 동일한 `href`를 가진 스타일의 중복을 제거합니다. `precedence` 속성은 문서의 `<head>` 내 다른 요소에 비해 `<style>` DOM 노드의 순위를 지정하며, 어떤 스타일시트가 다른 스타일시트를 덮어쓸 수 있는지를 결정합니다.

This special treatment comes with three caveats:

- React will ignore changes to props after the style has been rendered. (React will issue a warning in development if this happens.)
- React will drop all extraneous props when using the `precedence` prop (beyond `href` and `precedence`).
- React may leave the style in the DOM even after the component that rendered it has been unmounted.

- 스타일이 렌더링된 후에는 React가 Props 변경을 무시합니다. (개발 중에 이 상황이 발생하면 React는 경고를 표시합니다.)
- React는 precedence Prop을 사용할 때 불필요한 Props를 제거합니다. (단, href 와 precedence 는 제외.)
- 스타일을 렌더링한 컴포넌트가 마운트 해제된 후에도 DOM에 스타일이 유지될 수 있습니다.

사용법

인라인 CSS 스타일시트 렌더링하기

컴포넌트가 올바르게 표시되기 위해 특정 CSS 스타일에 의존하는 경우, 컴포넌트 내에서 인라인 스타일시트를 렌더링할 수 있습니다.

React는 동일한 href 를 가진 스타일시트의 중복을 제거하므로 href 속성은 스타일시트를 고유하게 식별해야 합니다.

precedence Prop을 제공하면 React는 컴포넌트 트리에서 해당 값이 표시되는 순서에 따라 인라인 스타일시트의 순서를 다시 지정합니다.

인라인 스타일시트는 로딩 중이더라도 Suspense 경계를 트리거하지 않습니다.

이는, 글꼴이나 이미지와 같은 비동기 리소스를 로드하는 경우에도 마찬가지입니다.

App.js ShowRenderedHTML.js

↺ 새로고침 ✘ Clear ⌂ 포크

```
import ShowRenderedHTML from './ShowRenderedHTML.js';
import { useState } from 'react';

function PieChart({data, colors}) {
  const id = useState();
  const stylesheet = colors.map((color, index) =>
    `#${id} .color-${index}: {\`color: ${color}; \`}`
  ).join();
  return (
    <>
      <style href={"PieChart-" + JSON.stringify(colors)} precedence="medium">
        {stylesheet}
      </style>
    </>
  );
}
```

▼ 자세히 보기

이전 [**<script>**](#) 다음 [**<title>**](#)



Copyright © Meta Platforms, Inc

uwu?

React 학습하기

[빠르게 시작하기](#)

[설치하기](#)

[UI 표현하기](#)

[상호작용성 더하기](#)

[State 관리하기](#)

[탈출구](#)

API 참고서

[React APIs](#)

[React DOM APIs](#)

[커뮤니티](#)

[더 보기](#)

행동 강령

블로그

팀 소개

React Native

문서 기여자

개인 정보 보호

감사의 말

약관

