

[API 참고서 >](#) [컴포넌트 >](#)

<link>

내장 브라우저 <link> 컴포넌트는 스타일시트와 같은 외부 리소스를 사용하거나 링크 메타데이터로 문서를 주석 처리할 수 있게 해줍니다.

```
<link rel="icon" href="favicon.ico" />
```

- [레퍼런스](#)
 - <[link](#)>
- [사용법](#)
 - 관련 리소스 연결하기
 - [스타일시트](#) 연결하기
 - [스타일시트 우선순위 제어하기](#)
 - 중복이 제거된 스타일시트 렌더링
 - 문서 내 특정 항목에 링크로 주석 달기

레퍼런스

<link>

스타일시트, 글꼴, 아이콘과 같은 외부 리소스를 링크하거나 링크 메타데이터로 문서를 주석 처리 하려면, [내장 브라우저 <link> 컴포넌트](#)를 렌더링하세요. 어떤 컴포넌트에서든 <link> 를 렌더링할 수 있으며, React는 [대부분의 경우](#) 해당 DOM 요소를 <head> 에 배치합니다.

```
<link rel="icon" href="favicon.ico" />
```

아래 예시를 참고하세요.

Props

<link> 는 모든 공통 엘리먼트 Props를 지원합니다.

- rel: 문자열 타입, 필수, 리소스와의 관계를 지정합니다. React는 다른 링크와는 달리 rel="stylesheet" 링크를 특별하게 처리합니다.

다음 속성들은 rel="stylesheet" 인 경우에 적용됩니다.

- precedence: 문자열 타입. <link> DOM 노드를 문서의 <head> 내 다른 요소와 비교하여 순위를 지정해야 합니다. 이를 통해 어떤 스타일시트가 다른 스타일시트를 덮어쓸 수 있는지 결정합니다. 값은 우선순위에 따라 "reset", "low", "medium", "high" 가 될 수 있습니다. 동일한 우선순위를 가진 스타일시트는 <link> 또는 인라인 <style> 태그 또는 preload 나 preinit 함수로 로드되었는지에 관계없이 함께 적용됩니다.
- media: 문자열 타입. 스타일시트를 특정 미디어 쿼리에 제한합니다.
- title: 문자열 타입. 대체 스타일시트의 이름을 지정합니다.

다음 속성들은 rel=stylesheet 인 경우에 적용되지만, React의 스타일시트에 대한 특별한 처리를 비활성화합니다.

- disabled: 불리언 타입. 스타일시트를 비활성화합니다.
- onError: 함수. 스타일시트 불러오기에 실패했을 때 호출됩니다.
- onLoad: 함수. 스타일시트 불러오기가 완료되었을 때 호출됩니다.

다음 속성들은 rel="preload" 나 rel="modulepreload" 인 경우에 적용됩니다.

- as: 문자열 타입. 리소스의 유형을 지정합니다. 가능한 값은 audio, document, embed, fetch, font, image, object, script, style, track, video, worker 입니다.
- imageSrcSet: 문자열 타입. as="image" 인 경우에만 적용됩니다. 이미지 소스의 집합을 지정합니다.
- imageSizes: 문자열 타입. as="image" 인 경우에만 적용됩니다. 이미지의 크기를 지정합니다.

다음 속성들은 rel="icon"이나 rel="apple-touch-icon" 인 경우에 적용됩니다.

- sizes: 문자열 타입. 아이콘의 크기를 지정합니다.

다음 속성들은 모든 경우에 적용됩니다.

- href: 문자열 타입. 연결된 리소스의 URL입니다.
- crossOrigin: 문자열 타입. 사용할 CORS 정책. 가능한 값은 anonymous 와 use-credentials 입니다. as 가 "fetch" 로 설정된 경우 필수입니다.
- referrerPolicy: 문자열 타입. 리소스를 가져올 때 보낼 Referrer 헤더를 지정합니다. 가능한 값은 no-referrer-when-downgrade (기본값), no-referrer, origin, origin-when-cross-origin, unsafe-url 입니다.
- fetchPriority: 문자열 타입. 리소스를 가져오는 우선순위를 지정합니다. 가능한 값은 auto (기본값), high, low 입니다.
- hrefLang: 문자열 타입. 연결된 리소스의 언어입니다.
- integrity: 문자열 타입. 리소스의 암호 해시로 진위를 확인합니다.
- type: 문자열 타입. 연결된 리소스의 MIME 유형입니다.

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 `<link>` 컴포넌트에 해당하는 DOM 요소를 React 트리의 어디에 렌더링하든 상관없이 항상 문서의 `<head>`에 배치합니다. `<head>`는 DOM 내에서 `<link>` 가 위치할 수 있는 유일한 위치이지만, 특정 페이지를 나타내는 컴포넌트가 `<link>` 컴포넌트를 자체적으로 렌더링할 수 있다면 편리하고 구성이 용이합니다.

여기에는 몇 가지 예외가 있습니다.

- `<link>`에 `rel="stylesheet"` 속성이 있는 경우, 이 특별한 동작을 위해 반드시 precedence 속성이 있어야 합니다. 이는 문서 내 스타일시트의 순서가 중요하기 때문입니다. React는 다른 스타일시트와의 순서를 결정하기 위해 precedence 속성을 사용합니다. precedence 속성이 생략된 경우 특별한 동작이 없습니다.
- `<link>`에 `itemProp` 속성이 있는 경우, 특별한 동작이 없습니다. 이 속성은 문서 전체가 아니라 페이지의 특정 부분에 대한 메타데이터를 나타냅니다.
- `<link>`에 `onLoad` 또는 `onError` 속성이 있는 경우, 연결된 리소스의 로딩을 React 컴포넌트 내에서 수동으로 관리하기 때문입니다.

스타일시트에 대한 특별한 동작

또한, `<link>` 가 스타일시트로 연결된 경우 (즉, 속성에 `rel="stylesheet"` 가 있는 경우) React는 다음과 같은 방식으로 특별하게 동작합니다.

- 스타일시트가 로드되는 동안 `<link>` 를 렌더링하는 컴포넌트는 일시 중단됩니다.
- 여러 컴포넌트가 동일한 스타일시트에 대한 링크를 렌더링하는 경우, React는 중복된 링크를 제거하고 DOM에 단일 링크만 배치합니다. 두 링크는 `href` 속성이 동일하면 같은 것으로 간주합니다.

위 특별한 동작에는 두 가지 예외가 있습니다.

- 링크에 `precedence` 속성이 없으면 특별한 동작이 없습니다. 이는 문서 내 스타일시트의 순서가 중요하기 때문에 React는 다른 스타일시트와의 순서를 결정하기 위해 `precedence` 속성을 사용합니다.
- `onLoad`, `onError`, `disabled` 속성을 제공하는 경우 특별한 동작이 없습니다. 이러한 속성들은 스타일시트 로딩을 컴포넌트 내에서 수동으로 관리하고 있음을 나타내기 때문입니다.

위 특별한 처리에는 두 가지 주의 사항이 있습니다.

- 링크가 렌더링 된 후에 React가 속성 변경을 무시합니다. (개발 중에 경고 메시지가 표시됩니다.)
- 링크를 렌더링한 컴포넌트가 마운트 해제된 후에도 React는 링크를 DOM에 남길 수 있습니다.

사용법

관련 리소스 연결하기

아이콘, 정규화된 URL, 핑백Pingback과 같은 관련 리소스에 대한 링크로 문서에 주석을 추가할 수 있습니다. React는 이 메타데이터를 React 트리의 어디에 렌더링 되든 상관없이 문서의 `<head>`에 배치합니다.

App.js ShowRenderedHTML.js

↺ 새로고침 × Clear ⌂ 포크

```
import ShowRenderedHTML from './ShowRenderedHTML.js';

export default function BlogPage() {
  return (
    <ShowRenderedHTML>
```

```
<link rel="icon" href="favicon.ico" />
<link rel="pingback" href="http://www.example.com/xmlrpc.php" />
<h1>My Blog</h1>
<p>...</p>
</ShowRenderedHTML>
);
```

스타일시트 연결하기

컴포넌트가 올바르게 표시되기 위해 특정 스타일시트에 의존하는 경우 해당 스타일시트에 대한 링크를 컴포넌트 내에서 렌더링할 수 있습니다. 스타일시트가 로드되는 동안 컴포넌트는 일시 중단됩니다. precedence 속성을 제공해야 하며 이는 React에 이 스타일시트를 다른 스타일시트와 비교하여 어디에 배치해야 하는지 알려줍니다. 높은 우선순위의 스타일시트는 낮은 우선순위의 스타일시트를 덮어쓸 수 있습니다.

▣ 중요합니다!

스타일시트를 사용하고 싶을 때 `preinit` 함수를 호출하는 것이 유용할 수 있습니다. 이 함수를 호출하면 단순히 `<link>` 컴포넌트를 렌더링하는 것보다 브라우저가 스타일시트

를 더 빨리 가져올 수 있습니다. 예를 들어 [HTTP Early Hints 응답을 보내는 방식](#)으로 가능합니다.

App.js ShowRenderedHTML.js

↪ 새로고침 ✕ Clear ⌂ 포크

```
import ShowRenderedHTML from './ShowRenderedHTML.js';

export default function SiteMapPage() {
  return (
    <ShowRenderedHTML>
      <link rel="stylesheet" href="sitemap.css" precedence="medium" />
      <p>...</p>
    </ShowRenderedHTML>
  );
}
```

스타일시트 우선순위 제어하기

스타일시트는 서로 충돌할 수 있으며, 이 경우 브라우저는 문서에서 나중에 오는 스타일시트를 적용합니다. React는 precedence 속성을 사용하여 스타일시트의 순서를 제어할 수 있도록 합니다.

다. 이 예시에서는 세 개의 컴포넌트가 스타일시트를 렌더링하며, 동일한 precedence 값을 가진 스타일시트는 <head>에서 함께 그룹화됩니다.

App.js ShowRenderedHTML.js

↪ 새로고침 X Clear ⌂ 포크

```
import ShowRenderedHTML from './ShowRenderedHTML.js';

export default function HomePage() {
  return (
    <ShowRenderedHTML>
      <FirstComponent />
      <SecondComponent />
      <ThirdComponent/>
      ...
    </ShowRenderedHTML>
  );
}
```

▼ 자세히 보기

Note the precedence values themselves are arbitrary and their naming is up to you. React will infer that precedence values it discovers first are “lower” and precedence values it discovers later are “higher”.

중복이 제거된 스타일시트 렌더링

여러 컴포넌트에서 동일한 스타일시트를 렌더링하면 React는 문서의 <head>에 단일 <link>만 배치합니다.

App.js ShowRenderedHTML.js

↺ 새로고침 × Clear ⌂ 포크

```
import ShowRenderedHTML from './ShowRenderedHTML.js';

export default function HomePage() {
  return (
    <ShowRenderedHTML>
      <Component />
      <Component />
      ...
    </ShowRenderedHTML>
  );
}
```

문서 내 특정 항목에 링크로 주석 달기

`itemProp` 속성을 사용하여 `<link>` 컴포넌트를 문서 내 특정 항목에 관련 리소스 링크로 주석을 달 수 있습니다. 이 경우 React는 이러한 주석을 문서의 `<head>`에 배치하지 않고 다른 React 컴포넌트와 같이 배치합니다.

```
<section itemScope>
  <h3>Annotating specific items</h3>
  <link itemProp="author" href="http://example.com/" />
  <p>...</p>
</section>
```

이전

다음

<textarea>

<meta>

 Meta Open Source

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

행동 강령

팀 소개

문서 기여자

감사의 말

더 보기

블로그

React Native

개인 정보 보호

약관

