

[API 참고서 >](#) [API >](#)

# preload

## ▣ 중요합니다!

React 기반 프레임워크는 리소스 로딩을 대신 처리하는 경우가 많으므로 이 API를 직접 호출할 필요가 없을 수도 있습니다. 자세한 내용은 해당 프레임워크의 문서를 참조하세요.

preload를 사용하면 사용할 스타일시트, 글꼴 또는 외부 스크립트와 같은 리소스를 미리 가져올 수 있습니다.

```
preload("https://example.com/font.woff2", {as: "font"});
```

- [레퍼런스](#)
  - [preload\(href, options\)](#)
- [사용법](#)
  - 렌더링 시 미리 불러오기
  - 이벤트 핸들러에서 미리 불러오기

## 레퍼런스

### preload(href, options)

리소스를 미리 불러오려면 react-dom에서 preload 함수를 호출합니다.

```
import { preload } from 'react-dom';

function AppRoot() {
  preload("https://example.com/font.woff2", {as: "font"});
  // ...
}
```

아래 예시를 참조하세요.

preload 기능은 브라우저에 주어진 리소스 다운로드를 시작해야 한다는 힌트를 제공하여 시간을 절약할 수 있습니다.

## 매개변수

- href: 문자열입니다. 다운로드하려는 리소스의 URL입니다.
- options: 객체입니다. 여기에는 다음과 같은 속성이 포함되어 있습니다.
  - as: 필수 문자열입니다. 리소스의 타입입니다. 가능한 값은 audio, document, embed, fetch, font, image, object, script, style, track, video, worker입니다.
  - crossOrigin: 문자열입니다. 사용할 CORS 정책입니다. 가능한 값은 anonymous와 use-credentials입니다. as 가 "fetch"로 설정된 경우 필수입니다.
  - referrerPolicy: 문자열입니다. 가져오기 Fetching 할 때 전송할 Referrer 헤더입니다. 사용할 수 있는 값은 no-referrer-when-downgrade (기본값), no-referrer, origin, origin-when-cross-origin 그리고 unsafe-url입니다.
  - integrity: 문자열입니다. 리소스의 진위 확인을 위한 리소스의 암호화 해시입니다.
  - type: 문자열입니다. 리소스의 MIME 타입입니다.
  - nonce: 문자열입니다. 엄격한 콘텐츠 보안 정책을 사용할 때 리소스를 허용하기 위한 암호화 논스입니다.
  - fetchPriority: 문자열입니다. 리소스 페칭의 상대적 우선순위를 제안합니다. 사용할 수 있는 값은 auto (기본값), high 그리고 low입니다.
  - imageSrcSet: 문자열입니다. as: "image" 와만 함께 사용합니다. 이미지의 소스 세트를 지정합니다.
  - imageSizes: 문자열입니다. as: "image" 와만 함께 사용합니다. 이미지의 크기를 지정합니다.

## 반환값

preload 는 아무것도 반환하지 않습니다.

## 주의 사항

- preload 에 대한 여러 번의 동등한 호출은 한 번의 호출과 동일한 효과를 갖습니다. preload 에 대한 호출은 다음 규칙에 따라 동등한 것으로 간주합니다.
  - 아래 경우를 제외하고 두 호출의 href 가 같으면 두 호출은 동일합니다.
  - as 가 image 로 설정된 경우 두 호출의 href , imageSrcSet 및 imageSizes 가 같으면 두 호출은 동일합니다.
- 브라우저에서는 컴포넌트 렌더링 중, Effect, 이벤트 핸들러 등 어떤 상황에서도 preload 를 호출할 수 있습니다.
- 서버 사이드 렌더링 또는 서버 컴포넌트를 렌더링할 때 preload 는 컴포넌트를 렌더링하는 동안 또는 컴포넌트 렌더링에서 발생하는 비동기 컨텍스트에서 호출하는 경우에만 영향을 미칩니다. 다른 호출은 무시됩니다.

## 사용법

### 렌더링 시 미리 불러오기

컴포넌트 또는 그 자식 컴포넌트가 특정 리소스를 사용한다는 것을 알고 있다면 컴포넌트를 렌더링할 때 preload 를 호출하세요.

#### 미리 불러오기 예시

1. 외부 스크립트 미리 불러오기    2. 스타일시트 미리 불러오기    3. 글꼴 미리 불러오기    <    >

#### 예시 1 of 4: 외부 스크립트 미리 불러오기

```
import { preload } from 'react-dom';

function AppRoot() {
  preload("https://example.com/script.js", {as: "script"});
  return ...;
}
```

브라우저에서 스크립트 실행을 즉시 시작하려면(스크립트를 다운로드하는 대신) `preinit` 을 대신 사용하세요. ESM 모듈을 로드하려면 `preloadModule` 을 사용하세요.

다음 예시

## 이벤트 핸들러에서 미리 불러오기

외부 리소스가 필요한 페이지나 State로 전환하기 전에 이벤트 핸들러에서 `preload` 를 호출하세요. 이렇게 하면 새 페이지나 State를 렌더링하는 동안 호출하는 것보다 프로세스가 더 빨리 시작됩니다.

```
import { preload } from 'react-dom';

function CallToAction() {
  const onClick = () => {
    preload("https://example.com/wizardStyles.css", {as: "style"});
    startWizard();
  }
  return (
    <button onClick={onClick}>Start Wizard</button>
  );
}
```

이전

`preinitModule`

다음

`preloadModule`

uwu?

설치하기

React DOM APIs

UI 표현하기

상호작용성 더하기

State 관리하기

탈출구

**커뮤니티**

행동 강령

팀 소개

문서 기여자

감사의 말

**더 보기**

블로그

React Native

개인 정보 보호

약관

