

[API 참고서 >](#) [API >](#)

# preinit

## ▣ 중요합니다!

React 기반 프레임워크는 종종 리소스 로딩을 자동으로 처리하므로, 이 API를 직접 호출하지 않아도 됩니다. 자세한 내용은 사용하는 프레임워크의 문서를 참고하세요.

preinit은 스타일시트나 외부 스크립트를 미리 가져오고 실행할 수 있게 합니다.

```
preinit("https://example.com/script.js", {as: "script"});
```

- [레퍼런스](#)
  - [preinit\(href, options\)](#)
- [사용법](#)
  - [렌더링 시 preinit 하기](#)
  - [이벤트 핸들러 내에서 preinit 하기](#)

## 레퍼런스

### preinit(href, options)

스크립트나 스타일시트를 preinit 하려면 react-dom에서 preinit 함수를 호출하세요.

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

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

## 아래 예시에서 더 보기.

preinit 함수는 브라우저에게 주어진 리소스를 다운로드하고 실행하라는 힌트를 제공하여 시간 절약에 도움을 줍니다. preinit 한 스크립트는 다운로드가 완료되면 즉시 실행됩니다. 스타일시트는 문서에 삽입되어 곧바로 적용됩니다.

## 매개변수

- href: 문자열. 다운로드하고 실행할 리소스의 URL입니다.
- options: 객체. 다음 속성들을 포함할 수 있습니다.
  - as: 필수 문자열. 리소스의 유형입니다. 가능한 값은 script 와 style 입니다.
  - precedence: 문자열. 스타일시트에 필수입니다. 다른 스타일시트와의 삽입 순서를 결정합니다. 우선순위가 높은 스타일시트가 낮은 것을 덮어쓸 수 있습니다. 가능한 값은 reset, low, medium, high 입니다.
  - crossOrigin: 문자열. 사용할 CORS 정책입니다. 가능한 값은 anonymous 와 use-credentials 입니다.
  - integrity: 문자열. 리소스의 무결성을 검증하기 위한 암호화 해시입니다.
  - nonce: 문자열. 엄격한 콘텐츠 보안 정책을 사용할 때, 리소스를 허용하기 위한 암호화된 nonce to allow the resource 입니다.
  - fetchPriority: 문자열. 리소스를 가져오는 데 사용할 상대적인 우선순위를 제안합니다. 가능한 값은 auto (기본값), high, low 입니다.

## 반환값

preinit 은 아무것도 반환하지 않습니다.

## 주의 사항

- 동일한 href 로 preinit 을 여러 번 호출해도, 한 번 호출한 것과 동일한 효과만 발생합니다.
- 브라우저에서는 컴포넌트를 렌더링할 때, Effect 내부에서, 이벤트 핸들러 안에서 등 어떤 상황에서든 preinit 을 호출할 수 있습니다.

- 서버 사이드 렌더링 또는 서버 컴포넌트를 렌더링할 때는, 컴포넌트를 렌더링하면서 또는 컴포넌트 렌더링에서 시작된 비동기 컨텍스트 내에서만 `preinit` 호출이 효과를 가집니다. 그 외의 호출은 무시됩니다.

## 사용법

### 렌더링 시 `preinit` 하기

특정 컴포넌트나 그 자식 컴포넌트가 특정 리소스를 사용할 것을 알고 있고, 해당 리소스가 다운로드되자마자 바로 실행되거나 적용되는 것이 괜찮다면, 컴포넌트를 렌더링할 때 `preinit` 을 호출하세요.

#### preinit 사용 예시

1. 외부 스크립트 `preinit` 하기    2. 스타일시트 `preinit` 하기



#### 예시 1 of 2: 외부 스크립트 `preinit` 하기

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

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

스크립트를 다운로드하되 즉시 실행하지 않으려면 `preload` 를 사용하세요. ESM 모듈을 로드하려면 `preinitModule` 을 사용하세요.

다음 예시

### 이벤트 핸들러 내에서 `preinit` 하기

외부 리소스가 필요한 페이지나 상태로 전환하기 전에 이벤트 핸들러에서 preinit을 호출하세요. 이렇게 하면 새 페이지나 상태 렌더링 시점보다 더 일찍 리소스 다운로드가 시작됩니다.

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

function CallToAction() {
  const onClick = () => {
    preinit("https://example.com/wizardStyles.css", {as: "style"});
    startWizard();
  }

  return (
    <button onClick={onClick}>Start Wizard</button>
  );
}
```

이전

[prefetchDNS](#)

다음

[preinitModule](#)

 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

더 보기

행동 강령

블로그

팀 소개

React Native

문서 기여자

개인 정보 보호

감사의 말

약관

