

# html 안에 script 태그의 위치 🤔

## ✓ 브라우저가 웹 사이트를 로드 할 때 일어나는 일

1. HTML 페이지 가져오기 (예 : index.html)
2. HTML 구문 분석을 시작합니다
3. 파서는 외부 스크립트 파일을 참조하는 `<script>` 태그를 발견합니다.
4. 브라우저가 스크립트 파일을 요청합니다. 한편 파서는 페이지의 다른 HTML을 구문 분석하는 것을 차단합니다.
5. 잠시 후 스크립트가 다운로드 된 후 실행됩니다.
6. 파서는 나머지 HTML 문서를 계속 파싱합니다.

- 4번은 좋지 않은 사용자 환경을 유발합니다. 웹 사이트는 기본적으로 모든 스크립트를 다운로드 할 때까지 로드를 중지하기 때문입니다.

왜 이런 일이 일어날까요?

모든 스크립트는 `document.write()` 또는 다른 DOM 조작을 통해 자체 HTML을 삽입할 수 있습니다. 이는 구문 분석기가 나머지 문서를 안전하게 구문 분석하기 전에 스크립트가 다운로드되고 실행될 때까지 기다려야 함을 의미합니다. 결국 스크립트는 자체 HTML을 문서에 삽입할 수 있습니다. 그러나 대부분의 개발자는 문서가 로드되는 동안 더 이상 DOM을 조작하지 않습니다. 대신 문서를 수정하기 전에 문서가 로드 될 때까지 대기합니다.

## ✓ 고전적인 방법

- 이 문제를 해결하기위한 이전 접근법은 `<body>` 맨 아래에 `<script>` 태그를 두는 것이었습니다. 이렇게하면 파서가 맨 끝까지 차단되지 않기 때문입니다.

이 접근 방식에는 자체적인 문제가 있습니다. 브라우저는 전체 문서가 구문 분석 될 때까지 스크립트 다운로드를 시작할 수 없습니다. 큰 스크립트 및 스타일 시트가있는 더 큰 웹 사이트의 경우 가능한 한 빨리 스크립트를 다운로드 할 수있는 것이 성능 향상에 매우 중요합니다. 웹 사이트가 2초 내에 로드되지 않으면 사람들은 다른 웹 사이트로 이동할 것 입니다.

최적의 솔루션에서 브라우저는 가능한 한 빨리 스크립트 다운로드를 시작하는 동시에 문서의 나머지 부분을 파싱합니다.

## ✓ 현대적인 방법

- 오늘날 브라우저는 스크립트의 비동기 및 지연 특성을 지원합니다. 이러한 속성은 스크립트가 다운로드되는 동안 계속 구문 분석을 하는 것이 안전하다는 것을 브라우저에 알려줍니다.

### >> async

```
<script type="text/javascript" src="path/to/script1.js" async></script>  
<script type="text/javascript" src="path/to/script2.js" async></script>
```

- async 속성이 있는 스크립트는 **비동기**적으로 실행됩니다. 즉, 스크립트가 다운로드 되는 즉시 브라우저가 차단되지 않고 실행됩니다. 이는 스크립트 2가 스크립트 1 이전에 다운로드 및 실행될 수 있음을 의미합니다. <http://caniuse.com/#feat=script-async>에 따르면 모든 브라우저의 97%가 이를 지원합니다.

### >> defer

```
<script type="text/javascript" src="path/to/script1.js" defer></script>  
<script type="text/javascript" src="path/to/script2.js" defer></script>
```

- defer 속성을 가진 스크립트는 순서대로 (즉, 첫 번째 스크립트 1 다음에 스크립트 2) 실행됩니다. 이것은 또한 브라우저를 차단하지 않습니다. defer는 비동기 스크립트와는 다르게 지연 스크립트는 전체 문서가 로드된 후에만 실행됩니다. <http://caniuse.com/#feat=script-defer>에 따르면 모든 브라우저 중 97%가 이를 지원합니다.

## ✓ 결론

현재의 최신 기술은 스크립트를 <head> 태그에 넣고 **async** 또는 **defer** 속성을 사용하는 것입니다. 이렇게 하면 브라우저를 차단하지 않고도 스크립트를 최대한 빨리 다운로드 할 수

있습니다.