

# htmx 빠른 참조

속성, 트리거, 스왑, 이벤트, 확장

## 기본

### 설치

```
<script src="https://unpkg.com/htmx.org@2"></script>

<!-- Or via npm -->
npm install htmx.org
```

### 동작 방식

<b>HTML-driven</b>	HTML 속성으로 AJAX 발행, JS 불필요
<b>Swap content</b>	서버가 HTML 조각 반환, htmx가 교체
<b>Progressive</b>	표준 HTML 향상, 점진적 지원
<b>REST-friendly</b>	모든 서버 사이드 프레임워크와 호환

## 속성

### 핵심 요청 속성

<b>hx-get="/url"</b>	URL로 GET 요청 발행
<b>hx-post="/url"</b>	URL로 POST 요청 발행
<b>hx-put="/url"</b>	URL로 PUT 요청 발행
<b>hx-patch="/url"</b>	URL로 PATCH 요청 발행
<b>hx-delete="/url"</b>	URL로 DELETE 요청 발행

### 동작 속성

<b>hx-trigger</b>	요청을 트리거하는 이벤트
<b>hx-target</b>	콘텐츠를 교체할 요소
<b>hx-swap</b>	응답 콘텐츠를 교체하는 방법
<b>hx-select</b>	응답 HTML의 일부 선택
<b>hx-vals</b>	요청에 추가 값 포함
<b>hx-confirm</b>	요청 전 확인 대화상자 표시
<b>hx-disable</b>	요소의 htmx 처리 비활성화

## 트리거

### 트리거 문법

```
<!-- Default: natural event (click for buttons) -->
<button hx-get="/data">Load</button>

<!-- Custom trigger event -->
<div hx-get="/news" hx-trigger="every 5s">Feed</div>

<!-- Multiple triggers -->
<input hx-get="/search" hx-trigger="keyup changed delay:300ms" />
```

### 트리거 수정자

<b>changed</b>	값이 변경된 경우에만 실행
<b>delay:Ns</b>	N초 대기 후 실행
<b>throttle:Nms</b>	N ms당 한 번으로 제한
<b>once</b>	처음 한 번만 트리거
<b>from:selector</b>	다른 요소에서 이벤트 수신
<b>every Ns</b>	N초마다 폴링
<b>load</b>	요소 로드 시 실행
<b>revealed</b>	요소가 뷰포트에 나타날 때 실행

## 타겟

### 타겟 선택

```
<!-- Target another element -->
<button hx-get="/data" hx-target="#result">Load</button>
<div id="result"></div>

<!-- Target closest ancestor -->
<button hx-get="/row" hx-target="closest tr">Update</button>

<!-- Target with CSS selector -->
<button hx-get="/info" hx-target="next .output">Go</button>
```

## 타겟 키워드

<b>this</b>	요소 자체 (기본값)
<b>closest &lt;sel&gt;</b>	선택자와 일치하는 가장 가까운 조상
<b>find &lt;sel&gt;</b>	선택자와 일치하는 첫 번째 자손
<b>next &lt;sel&gt;</b>	선택자와 일치하는 다음 형제
<b>previous &lt;sel&gt;</b>	선택자와 일치하는 이전 형제

## 스왑

### 스왑 전략

<b>innerHTML</b>	내부 콘텐츠 교체 (기본값)
<b>outerHTML</b>	타겟 요소 전체 교체
<b>afterbegin</b>	타겟 안 맨 앞에 삽입
<b>beforeend</b>	타겟 안 맨 뒤에 추가
<b>beforebegin</b>	타겟 앞에 삽입
<b>afterend</b>	타겟 뒤에 삽입
<b>delete</b>	타겟 요소 제거
<b>none</b>	교체 없이 이벤트만 발생

### 스왑 수정자

```
<!-- Swap with transition delay -->
<div hx-get="/data" hx-swap="innerHTML swap:300ms">

<!-- Settle delay for CSS transitions -->
<div hx-get="/data" hx-swap="innerHTML settle:500ms">

<!-- Scroll to top after swap -->
<div hx-get="/page" hx-swap="innerHTML scroll:top">
```

## 헤더

### 요청 헤더 (htmx가 전송)

<b>HX-Request</b>	htmx 요청이면 항상 "true"
<b>HX-Target</b>	타겟 요소의 ID
<b>HX-Trigger</b>	트리거된 요소의 ID
<b>HX-Trigger-Name</b>	트리거된 요소의 name
<b>HX-Current-URL</b>	브라우저 현재 URL
<b>HX-Prompt</b>	hx-prompt에 대한 사용자 응답

### 응답 헤더 (서버가 전송)

<b>HX-Redirect</b>	URL로 클라이언트 사이트 리다이렉트
<b>HX-Refresh</b>	"true"이면 전체 페이지 새로고침
<b>HX-Retarget</b>	CSS 선택자로 hx-target 재정의
<b>HX-Reswap</b>	hx-swap 전략 재정의
<b>HX-Trigger</b>	클라이언트 사이트 이벤트 트리거
<b>HX-Push-Url</b>	브라우저 히스토리에 URL 추가

## 이벤트

### 생명 주기 이벤트

<b>htmx:configRequest</b>	요청 전; 파라미터/헤더 수정
<b>htmx:beforeRequest</b>	AJAX 호출 직전
<b>htmx:afterRequest</b>	요청 완료 후
<b>htmx:beforeSwap</b>	콘텐츠 교체 전
<b>htmx:afterSwap</b>	콘텐츠 교체 후
<b>htmx:afterSettle</b>	스왑 후 DOM이 안정화된 후
<b>htmx:responseError</b>	서버가 에러 상태 반환

## 이벤트 수신

```
document.body.addEventListener("htmx:afterSwap", (e) => {
  console.log("Swapped:", e.detail.target);
});

// Cancel a request
document.body.addEventListener("htmx:configRequest", (e) => {
  if (!confirm("Proceed?")) e.preventDefault();
});
```

## 확장

### 확장 사용

```
<script src="https://unpkg.com/htmx-ext-json-enc"></script>

<div hx-ext="json-enc">
  <form hx-post="/api/data">
    <input name="title" />
    <button>Submit as JSON</button>
  </form>
</div>
```

### 주요 확장

<b>json-enc</b>	요청 본문을 JSON으로 인코딩
<b>loading-states</b>	로딩 상태 CSS 클래스 관리
<b>head-support</b>	응답의 <head> 태그 병합
<b>preload</b>	mousedown/hover 시 링크 미리 로드
<b>sse</b>	Server-Sent Events 지원
<b>ws</b>	WebSocket 지원
<b>response-targets</b>	에러 응답에 다른 타겟 지정

## 인디케이터

### 로딩 인디케이터

```
<button hx-get="/slow" hx-indicator="#spinner">
  Load Data
</button>
<span id="spinner" class="htmx-indicator">Loading...</span>
```

### 인디케이터 CSS

```
.htmx-indicator { display: none; }
.htmx-request .htmx-indicator { display: inline; }
.htmx-request .htmx-indicator { display: inline; }
```

htmx는 요청 중 요소에 htmx-request 클래스를 추가

### 요청 중 비활성화

```
<button hx-post="/submit" hx-disabled-elt="this">
  Submit
</button>
```

hx-disabled-elt는 요청 중 disabled 속성 추가

## 일반 패턴

### 실시간 검색

```
<input type="search" name="q"
  hx-get="/search"
  hx-trigger="keyup changed delay:300ms"
  hx-target="#results" />
<div id="results"></div>
```

### 무한 스크롤

```
<tr hx-get="/rows?page=2"
  hx-trigger="revealed"
  hx-swap="afterend">
  <td>Loading more...</td>
</tr>
```

# htmx 빠른 참조

---

## 확인 후 삭제

```
<button hx-delete="/item/42"
        hx-confirm="Delete this item?"
        hx-target="closest .item"
        hx-swap="outerHTML">
  Delete
</button>
```

## 인라인 편집

```
<div hx-get="/edit/1" hx-trigger="click"
      hx-swap="outerHTML">
  Click to edit this text
</div>
```