htmx クイックリファレンス

属性、トリガー、スワップ、イベント、拡張

基本

インストール

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

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

動作の仕組み

HTML-driven	HTML 属性から AJAX を発行、JS 不要
Swap content	サーバーが HTML フラグメントを返し、htmx がそれを差し込む
Progressive	標準 HTML を拡張；グレースフルデグラデーション対応
REST-friendly	任意のサーバーサイドフレームワークで動作

属性

コアリクエスト属性

hx-get="/url"	URL へ GET リクエストを発行
hx-post="/url"	URL へ POST リクエストを発行
hx-put="/url"	URL へ PUT リクエストを発行
hx-patch="/url"	URL へ PATCH リクエストを発行
hx-delete="/url"	URL へ DELETE リクエストを発行

動作属性

hx-trigger	リクエストを起動するイベント
hx-target	コンテンツを差し込む対象要素
hx-swap	レスポンスコンテンツの差し込み方法
hx-select	レスポンス HTML の一部を選択
hx-vals	リクエストに追加の値を付加
hx-confirm	リクエスト前に確認ダイアログを表示
hx-disable	要素の htmx 処理を無効化

トリガー

トリガー構文

```
<!-- 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" />
```

トリガー修飾子

changed	値が変化した場合のみ発火
delay:Ns	発火前に N 秒待機
throttle:Nms	N ミリ秒に 1 回にスロットル
once	最初の 1 回のみトリガー
from:selector	別の要素でイベントをリッスン
every Ns	N 秒ごとにポーリング
load	要素ロード時に発火
revealed	要素がスクロールで表示されたときに発火

ターゲット

ターゲット選択

```
<!-- 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>
```

ターゲットキーワード

this	要素自身（デフォルト）
closest <sel>	セレクターにマッチする最も近い祖先
find <sel>	セレクターにマッチする最初の子孫
next <sel>	セレクターにマッチする次の兄弟
previous <sel>	セレクターにマッチする前の兄弟

スワップ

スワップ戦略

innerHTML	内部コンテンツを置換（デフォルト）
outerHTML	対象要素全体を置換
afterbegin	対象の内側の先頭に挿入
beforeend	対象の内側の末尾に追加
beforebegin	対象の前に挿入
afterend	対象の後に挿入
delete	対象要素を削除
none	スワップなし、イベントのみ発火

スワップ修飾子

```
<!-- 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">
```

ヘッダー

リクエストヘッダー（htmx が送信）

HX-Request	htmx リクエストの場合は常に "true"
HX-Target	対象要素の ID
HX-Trigger	トリガーされた要素の ID
HX-Trigger-Name	トリガーされた要素の name 属性
HX-Current-URL	ブラウザの現在の URL
HX-Prompt	hx-prompt へのユーザー入力

レスポンスヘッダー（サーバーが送信）

HX-Redirect	クライアントサイドで URL にリダイレクト
HX-Refresh	"true" の場合にページ全体をリフレッシュ
HX-Retarget	CSS セレクターで hx-target をオーバーライド
HX-Reswap	hx-swap 戦略をオーバーライド
HX-Trigger	クライアントサイドのイベントをトリガー
HX-Push-Url	ブラウザ履歴に URL をプッシュ

イベント

ライフサイクルイベント

htmx:configRequest	リクエスト前；パラメーターやヘッダーを変更可能
htmx:beforeRequest	AJAX 呼び出し直前
htmx:afterRequest	リクエスト完了後
htmx:beforeSwap	コンテンツ差し込み前
htmx:afterSwap	コンテンツ差し込み後
htmx:afterSettle	スワップによる DOM 安定化後
htmx:responseError	サーバーがエラーステータスを返した

イベントのリッスン

```
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();
});
```

拡張

拡張の使用

```
<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>
```

主な拡張

json-enc	リクエストボディを JSON としてエンコード
loading-states	ローディング状態の CSS クラスを管理
head-support	レスポンスの <head> タグをマージ
preload	マウスダウン / ホバー時にリンクをプリロード
sse	サーバー送信イベント（SSE）のサポート
ws	WebSocket のサポート
response-targets	エラーレスポンス用に別のターゲットを指定

インジケーター

ローディングインジケーター

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

インジケーター CSS

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

htmx はリクエスト中に要素へ htmx-request クラスを付与する

リクエスト中に無効化

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

hx-disabled-elt はリクエスト中に disabled 属性を付与する

htmx クイックリファレンス

よく使うパターン

アクティブサーチ

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

無限スクロール

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

確認付き削除

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

インライン編集

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