Engineer

hreflang

the attribute pairing a page with its translations

What it is

Written on a link, `hreflang` states which language the page at the other end is in. HTML allows it on `a`, `area` and `link` elements, but the use that made it well known is a set of `link rel="alternate"` elements in a page's head, one per language version, through which a site tells search engines that several addresses hold the same content in different languages.

```
<link rel="alternate" hreflang="en" href="https://example.com/guide/" />
<link rel="alternate" hreflang="da" href="https://example.com/da/guide/" />
<link rel="alternate" hreflang="x-default" href="https://example.com/guide/" />
```

The HTML standard calls the attribute "purely advisory". Its value "must be a valid BCP 47 language tag", and a browser has to take a fetched page's language from the page itself, never from the link that led there. Combined with `rel="alternate"`, an `hreflang` that differs from the document's own language marks the target as a translation.

Search engines turned that advisory note into a protocol. Google documents three equivalent ways to supply the set: link elements in the head, an HTTP `Link` header for PDFs and other files without one, and entries in an XML sitemap. Each version lists itself as well as every other version, with fully qualified addresses. A pair counts only when both pages name each other, so that another site cannot declare itself a translation of yours. Codes are an ISO 639-1 language, optionally followed by an ISO 3166-1 region, and a region on its own is invalid. The reserved value `x-default` names the page for readers whose language matches none of the others, and Google notes it was designed for language selector pages.

Two neighbours are easy to confuse with it. The `lang` attribute says what language the text in front of the reader is in; `hreflang` says what language a linked page is in. Google states that it uses neither to detect a page's language, so `hreflang` cannot make a page Danish; it tells an engine that a Danish page it already recognises belongs with an English one. And a canonical URL merges duplicates into one indexed address, while `hreflang` identifies language alternatives and helps a search engine choose which version to show.

Use the concept in a decision

Put "What is hreflang?" to work in a real decision. We can help relate the concept to the system you are building or buying. Software that reads naturally in every market you sell in, including the dates, the currencies and the way names sort.

Explore our services Get in touch →

Test the assumptions behind the term

Use the explanation and linked work to frame questions for your project. We can examine the assumptions with you before a tool is chosen. Built an English/Danish i18n system — 12,027 messages a locale, in English and Danish

Built an English/Danish i18n system Get in touch →

Why it matters

The decision `hreflang` records is whether the language versions of a page form one cluster or are judged as separate pages. Left separate, each version competes on its own merits, and an engine may show the English page to a Danish reader, or list both versions in the same results. Declared as a cluster, the versions are understood as one piece of content, and the engine can offer the version that matches the reader's language settings.

The cluster only makes sense when the versions really are the same content. A site in one language needs no `hreflang` at all, and two language sections that differ in substance are not a cluster either. Regional variants with small differences, such as `en-GB` and `en-US` pages that differ in prices, form a cluster too.

The cost is bookkeeping, and it grows with the square of the number of languages. Every version of every page carries the full list, so adding, moving or withdrawing one translation changes the head of every sibling, and a list that is right on most pages and wrong on a few fails quietly: the pairs that point both ways are used and the rest are dropped, with no visible error. That makes `hreflang` a poor fit for hand editing and a natural fit for a generator that derives every page's set from one list. What `hreflang` changes is which of several versions a searcher is shown.

What it looks like in practice

This site builds its language editions from one content tree, so each page has a set of sibling translations to declare.

One template decides which translations are announced, and several places read it: the `hreflang` set in the head, the `og:locale:alternate` tags used by link previews, the alternates in the sitemap, the `inLanguage` of the site's structured data, and, after one further subtraction, the language strip a reader sees. The list includes the page itself, which supplies the self-reference the rules ask for. The codes name languages only, such as `en`, `da` and `uk`, so each version is offered to every reader of that language wherever they are, and the English entry is written a second time as `x-default`. The head and the sitemap both carry the set, rendered from that one list, so they cannot drift apart.

The split between machines and readers is deliberate. At an earlier implementation stage, the Ukrainian version was announced to crawlers and listed in the sitemaps, while the top-bar language strip omitted it. One configuration flag controlled that distinction: crawlers received reciprocal Ukrainian alternates while only the reader's language strip changed. A second flag, for a language that is built but not ready, removes it from both lists and marks its pages `noindex`. A page in such a language declares no siblings either, because an `hreflang` from an unannounced page to an announced one is a pair an engine sees from only one end.

Copies follow the same reasoning. A page served on one of the focused subdomains, whose canonical points back to the main site, emits no `hreflang` at all: the set is read on the page an engine has accepted as canonical, and a copy naming its own host's translations would contradict the line above it. Its language switcher still works, because a reader who wants Danish should still get Danish.

A check over the built pages asserts that every page with alternates includes `x-default` and its own language, and that every alternate is a built page, is in an announced language and points back. The self-reference rule earns its place. The generator offers two similarly named lists, `.Translations`, which leaves the current page out, and `.AllTranslations`, which includes it, and the one-word slip removes each page's own language from the head and from the language strip at once, leaving the two in agreement. Against a deliberately broken build, that single assertion was the only one in the check to catch 528 of the 792 pages.

Where it goes wrong

Searchers are shown the version in the wrong language. A Danish reader is sent to the English page, or both versions turn up side by side. The pairs point one way: the Danish page names the English one, but the English page still lists only the languages it had before, and engines ignore a pair that is missing its return link. Build every page's set from one list that includes the page itself, and test reciprocity on the built output.

A whole language's annotations do nothing. The markup is present and valid HTML, yet the versions are treated as unrelated. The codes name the wrong thing: `ua` or `dk` where the languages are `uk` and `da`, `be` meant as Belgium when it is the code for Belarusian, or `en-UK` where `en-GB` was meant, since `UK` is a reserved region code that Google ignores. Use a language from ISO 639-1, add a region from ISO 3166-1 only where the content differs by country, and remember that a valid BCP 47 tag can still be one Google does not support.

Translations drop out of the index. Only the original language appears in search, and the translations are treated as its duplicates. Each translation names the original as its canonical, which asks for the translation to be merged away while its `hreflang` asks for it to be shown. Make every language version its own canonical; Google's guidance is that the canonical should be a page in the same language.

A page that asked to stay out of search is announced anyway. A draft language marked `noindex` appears among the alternates of published pages, or lists them from its own head. The announcement and the directive were set in different places, so the site gives a crawler two answers about the same address. Decide per language, announced in both directions or in neither, put that decision in one setting that every list reads, and check the built pages against it.

Further reading

- HTML Standard: the hreflang attribute — what the attribute means on a link, and why a browser must not trust it.
- HTML Standard: link type "alternate" — how `hreflang` and `type` describe a translation or another format of the same document.
- RFC 5646: Tags for Identifying Languages — the BCP 47 syntax the attribute's value has to follow.

- Google Search Central: localized versions of your pages — the three delivery methods, the reciprocity rule and `x-default` .

Glossary

Choose for your actual constraints

Describe the workflow, the existing interfaces and the users who need the change. We can define a testable delivery and a handover that lets your team maintain it.

See what it costs Get in touch →

Make the terminology practical

Ask how “What is hreflang?” applies to your organisation. A concrete example from your current system is enough to start the discussion.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://engineer.company/glossary/hreflang/