

quarto-index HTML index probe

Cross-references that repeat, and ones that only look alike

Properties of the generated HTML index that the LaTeX back-end has no equivalent of, so they cannot be probed through a `.tex` render.

The entry those targets can name: `ab`.

One target repeated on one key is one cross-reference, not two — the reader wants the target, not a count of how the marks were spread: `zeta` and again `zeta`.

Two targets on one key that print identically but name different things — a two-level target, and a single level that merely contains the level join — are two cross-references. Only the two-level one names an entry that exists, so only it is a link: `eta` and `eta`.

Ids the author has already used

An `id` the extension would otherwise mint for itself, written by the author on a mark of their own: `theta`. An `id` can also sit where the document tree carries no `id` at all — written inside raw HTML, which only a reading of the raw text itself can see:

The marks that follow must be given neither `id` — two elements answering to one name send one of the two locators to the wrong place: `iota` and `kappa`.

The same for an entry `id`: `lambda`.

A `span` forging this extension's own plumbing attribute must not steal a real mark's anchor: `decoy` is not a mark at all, and `nu` is a cross-reference, which never carries an anchor of its own.

Entries whose sort keys are equal

Two terms differing only in ASCII case fold to the same sort key, so the tie falls to character code and the capital comes first. They are written here in the opposite order to the one they must appear in, so the index cannot be right by having left them alone: bee and Bee.

Index

A

B, [1](#)

Bee, [2](#)

bee, [2](#)

eta, *see* A: B

iota, [1](#)

kappa, [1](#)

lambda, [1](#)

nu, *see* A: B

theta, [1](#)

zeta, *see* A: B