

quarto-index letter-group probe

Which group an entry lands in, and why

A top-level entry's group comes from the string it FILES under — its sort key where it has one, its printed text where it does not. Every entry below is written so that one of those two paths reaches one of the two outcomes, a letter group or the leading Symbols group.

Printed text, reaching a letter: zebra.

Printed text, reaching Symbols, and the first of the pair whose adjacency is the point: #hash-tag.

The second of that pair. Between # and ~ in character order sit every letter and every digit, so these two are neighbours only because the letters have been lifted out into groups of their own — an index that merely sorted by character code would print half the alphabet between them: ~tilde.

A sort key, reaching a letter: the entry prints as a symbol but files under a word, so it leaves Symbols altogether: #1 priority.

A sort key, reaching Symbols: the entry prints as a word but files under a symbol, so it leaves its letter group. The key is written !! because a single ! separates sort levels — this key is one level whose first character is a literal exclamation mark: Quixote.

An entry whose printed text begins outside ASCII files under Symbols, which is honest about a collation that only folds ASCII: éclair.

An entry whose top level is empty. The level prints nothing, so it is dropped and what was the sub-entry becomes the entry — filed under `windmill`, in W, rather than under an empty string in Symbols: sub.

Two more letters, so the alphabet's own order is visible: alpha and Mango.

Index

Quixote, [1](#)
#hashtag, [1](#)
~tilde, [1](#)

alpha, [1](#)
#1 priority, [1](#)

Mango, [1](#)

windmill, [1](#)

zebra, [1](#)

éclair, [1](#)