in-dexter

An index package for Typst

Version 0.0.6 (30. September 2023)

Rolf Bremer, Jutta Klebe

1 Sample Document to Demonstrate the in-dexter package

Using the in-dexter package in a typst document consists of some simple steps:

1. Importing the package `in-dexter`.
2. Marking the words or phrases to include in the index.
3. Generating the index page by calling the `make-index()` function.

1.1 Importing the Package

The in-dexter package is currently available on GitHub in its home repository (https://github.com/RolfBremer/in-dexter). It is still in development and may have breaking changes in its next iteration.

```
#import "./in-dexter.typ": *
```

The package is also available via Typst's build-in Package Manager:

```
#import "@preview/in-dexter:0.1.0": *
```

Note, that the version number of the typst package has to be adapted to get the wanted version.

1.2 Marking of Entries

We have marked several words to be included in an index page at the end of the document. The markup for the entry stays invisible . Its location in the text gets recorded, and later it is shown as a page reference in the index page.

```
#index[The Entry Phrase]
```

or

```
#index([The Entry Phrase])
```

or

```
#index("The Entry Phrase")
```

1.3 Advanced entries

1.3.1 Symbols

Symbols can be indexed to be sorted under `"Symbols"`, and be sorted at the top of the index like this

```
#index(initial: (letter: "Symbols", sorty-by: "#"), [$(rho)$])
```

1.3.2 Nested entries

Entries can be nested. The `index` function takes multiple arguments - one for each nesting level.

```
#index("Sample", "medical", "blood")
#index("Sample", "medical", "tissue")
#index("Sample", "musical", "piano")
```

1.3.3 Formatting Entries

Entries can be formatted with arbitrary functions that map `content` to `content`

```
#index(fmt: it => strong(it), [The Entry Phrase])
```

or shorter

```
#index(fmt: strong, [The Entry Phrase])
```

of course this can be abstracted into a helper function

```
#let index-bold(..entries) = index(fmt: strong, entries)
```

1.4 The Index Page

To actually create the index page, the `make-index()` function has to be called. Of course, it can be embedded into an appropriately formatted environment , like this:

```
#columns(3)[
    #make-index()
]
```

2 Why Having an Index in Times of Search Functionality?

A *hand-picked* or *handcrafted* Index in times of search functionality seems a bit old-fashioned at the first glance. But such an index allows the author to direct the reader, who is looking for a specific topic , to exactly the right places. Especially in larger documents and books this becomes very useful, since search engines may provide too many locations of specific words. The index is much more comprehensive , assuming that the author has its content selected well. Authors know best where a certain topic is explained thoroughly or merely noteworthy mentioned (using the `index` function). Note, that this document is not necessarily a good example of the index. Here we just need to have as many index entries as possible to demonstrate (using a custom made `index-bold` function) the functionality and have a properly filled index at the end. Even for symbols like (ρ).

---

3 Index

Here we generate the Index page in three columns: