Typst-Glossary

A Glossary Package for Typst

Version 0.1.4 (5.7.2024)

Rolf Bremer, Jutta Klebe

Sample Document to Demonstrate the Package Gloss-Awe

This package can create a glossary for a document. The glossary entries are pulled from a pool of entries using only entries, that are marked in the document. The package creates warnings for marked entries, that cannot be found in the entry pool, but this warnings can be suppressed with a parameter to the `make-glossary()` function.

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

1. Importing the package `gloss-awe`.
2. Marking the words or phrases to include in the glossary with `gls[]`.
3. Optional: Defining a showmarker for the marker.
4. Read in one or more glossary pool(s) (from file(s) or create it in the document itself).
5. Generating the glossary page by calling the `make-glossary(..glossary-pool)` function.

Figure 1: Glossary Page

Importing the Package

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

```
#import "./gloss-awe.typ": *
```

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

```
#import "@preview/gloss-awe:0.1.4": *
```

Note, that the version number ("0.1.4") have to be adapted to get the wanted version.

Marking of Entries

We have marked several words to be included in the glossary page at the end of the document. The marked entries are shown on the glossary page.

```
This is a #gls[Sample] to demonstrate the 'gloss-awe' package.
```

The previous markup marks "Sample" as reference for the glossary. If "Sample" is contained in the Glossary-Pool, it will be included into the resulting glossary page. If the Entry in the pool has a different key word, the following marker syntax can be used:

```
This is a #gls(entry: "Example")[Sample] to demonstrate the 'gloss-awe' package.
```

In this case, the entry for “Example” is taken from the glossary pool, while in the document the term “Sample” is used.

The `entry` parameter should also be used, if the entry text (display) is some rich content, like a math expression.

Complex Content

To reference a Glossary entry with a complex name, like this Complex Content containing a whitespace, it is a good idea to use the entry parameter of the `gls`-function, to map it to a non-complex entry in a glossary pool, or, create the pool entry with a string as its key (see sample code!).

Other, even more complex content, may definitely need an entry text given. The entry text controls where the entry is sorted in, and also is it used for the lookup in the glossary pools.

```
#gls(entry: "Kreisfläche", $A=pi r^2$)

#gls(entry: "E=MC2", $ E=m c^2 $)

#gls(entry:"I1" ,$ I = rho^2 * sigma^3 $)
```

$A = \pi r^2$

$$E = mc^2$$

$$I = \rho^2 * \sigma^3$$

Glossary Entries that are not visible in the Documents Content

It is also possible to reference glossary entries without having them occur in the content of the document. They will only appear in the glossary. The function `#gls-add[Keyword]` can be used to create such a reference.

Here we use different notations to add entries with `gls-add()`:

```
#gls-add[Calcium]
#gls-add("Beryllium")
#gls-add[Potassium - Hydrochloride]
#gls-add["Iron"]
```

Defining a showmarker

For review reasons, the marked entries can be made more visible in the resulting document. For example like here:

```
#let my-gls = gls.with(showmarker: w => text(fill: teal, [#w]))
```

This function can be used to mark entries that then appear colored in the typst document.

The index markers now show up in the resulting document and can easily be reviewed. To define this behavior generally, the gls function can be redefined like this:

```
#let gls = gls.with(showmarker: w => text(fill: teal, [#w]))
```

Casing

Note that the casing of the entries matters. It may sometimes be desirable to just ignore the casing while generating the glossary page, but there are cases where casing is important - especially when it

comes to trademarks and logos. An example is provided here, where “Context” as well as “ConTeXt” is contained in the glossary.

Starting with version 0.1.0, gloss-awe supports custom sorting: A function can be provided to make-glossary() to determine the sort key for the entries.

```
#make-glossary(global-glossary, sort: x => lower(x))
```

or shorter:

```
#make-glossary(global-glossary, sort: lower)
```

Hiding entries from the glossary page

It is also possible to hide entries (temporarily) from the generated glossary page without removing any markers for them from the document.

The following sample will hide the entries for “Amaranth” and “Butterscotch” from the glossary, even if it is marked with `gls[...]` or `gls-add[...]` somewhere in the document.

```
    #let hidden-entries = (
        "Amaranth",
        "Butterscotch"
    )

    #make-glossary(glossary-pool, excluded: hidden-entries)
```

The Glossary Pool(s)

The pool contains the definitions for the entries. In this sample, we read the pool(s) from one or more files – here from typst files. But they may also be XML-Files or other sources. The `make-glossary()` method can take more than one pool at once. The matching of marked entries is done in the order the pools where given in the parameters of the method. The first match wins.

The pools are typst dictionaries, where the key is the marked word. The entry under this key is itself a dictionary, containing one or more entries with well known keys:

- description
  This is the description of the marked word.
- link
  This optional entry can contain an external link (URL).

more well known entries may come in future versions.

The Glossary Page

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

```
#columns(2)[
    #make-glossary(glossary-pool, sort-key: lower, suppress-missing: false)
]
```

Note: the parameter `suppress-missing` is set to false (which is the default). So marked entries that could not be found in the provided glossary pools, are marked with a No glossary entry on the glossary page.

The next sample uses two different pools: a specific pool and a global pool.

```
#columns(2)[
    #make-glossary(specific-pool, glossary-pool, sort-key: lower)
]
```

Why Having a Glossary in Times of Search Functionality on the Internet?

A well-defined Glossary can be very helpful in documents where very specific meanings of certain Terms are used. For example, the term “Context”. In a specific document it may refer not to the general context, but may be used for a specific data structure in a system. In another document it may refer to a typesetting system with the name “ConTeXt”. A Glossary can be used to define things for the document’s context. It is used to agree on a common definition of Terms used in the document.

Test Text

In this section, we have some more sample text to have some more references for the glossary. The rest of this section is Test Text, so it may not carry much meaning in it. It is more like Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat..

The Commit-SHA of Git is a very nice key to identify specific versions. The term *Supercalifragilisticexpialigetisch* comes to mind, if one thinks about really long words, but that’s another story.

Glossary

To create the glossary page, we load the glossary pool from a file and call the `make-glossary()` function with it. Here we generate the glossary page with referenced entries in two columns:

“Iron”

No glossary entry

Beryllium

Element number 4 in the Periodic Table of Elements.

Calcium

Element number 20 in the Periodic Table of Elements.

casing

The casing in the glossary is of importance, since there are cases where the same Term has different meaning, depending on the Casing.

Commit-SHA

Derived from web search results: In Git, a Commit-SHA is a 40-character hexadecimal string that specifies a 160-bit SHA-1 hash. It is used to identify a commit.

common definition

Many word in a natural language have many meanings. Sometimes, but unfortunately not always, the intention of the writer or speaker can be derived from the context the word is used in. In documentation, it is important to have a common understanding of the word. That’s why we use glossaries to define the intended meaning.

Complex Content

No glossary entry

Context

The circumstances that form the setting for an event, statement, or idea, and in terms of which it can be fully understood.

ConTeXt

A nice, modern typesetting system derived from LaTeX.

$E = mc^2$

No glossary entry

glossary pool

A glossary pool is a collection of glossary entries. An automated tool can pull needed definitions from this pool to create the glossary pages for a specific context.

$I = \rho^2 * \sigma^3$

No glossary entry

iteration

repetition of a mathematical or computational procedure applied to the result of a previous application, typically as a means of obtaining successively closer approximations to the solution of a problem.

$A = \pi r^2$

Die Fläche eines Kreises.

Potassium - Hydrochloride

No glossary entry

Supercalifragilisticexpialigetisch

No glossary entry

Test Text

Test Text is used in this document to demonstrate the mechanism of the glossary functionality. It is usually a mixture of some arbitrary sentences containing terms that allow the demonstration of certain features and some lorem ipsum filler text. For demonstration of certain features a certain length of the *Test Text* is required.

typst

Typst is a new markup-based typesetting system for the sciences. It is designed to be an

alternative both to advanced tools like LaTeX and simpler tools like Word and Google Docs.

typst

See “Typst”.

XML

XML stands for `'eXtensible Markup Language'`. https://www.w3.org/XML

Glossary (with additional local pool)

This Glossary uses an additional glossary pool file to resolve the marked entries.

“Iron”

No glossary entry

Beryllium

Element number 4 in the Periodic Table of Elements.

Calcium

Element number 20 in the Periodic Table of Elements.

casing

The casing in the glossary is of importance, since there are cases where the same Term has different meaning, depending on the Casing.

Commit-SHA

Derived from web search results: In Git, a Commit-SHA is a 40-character hexadecimal string that specifies a 160-bit SHA-1 hash. It is used to identify a commit.

common definition

Many word in a natural language have many meanings. Sometimes, but unfortunately not always, the intention of the writer or speaker can be derived from the context the word is used in. In documentation, it is important to have a common understanding of the word. That’s why we use glossaries to define the intended meaning.

Complex Content

A Complex Content is an entry with a key consisting of more than a single word.

ConTeXt

A nice, modern typesetting system derived from LaTeX.

Context

The circumstances that form the setting for an event, statement, or idea, and in terms of which it can be fully understood.

$E = mc^2$

No glossary entry

glossary pool

A glossary pool is a collection of glossary entries. An automated tool can pull needed definitions from this pool to create the glossary pages for a specific context.

$I = \rho^2 * \sigma^3$

No glossary entry

$A = \pi r^2$

Die Fläche eines Kreises.

Potassium - Hydrochloride

No glossary entry

Supercalifragilisticexpialigetisch

No glossary entry

Test Text

Test Text is used in this document to demonstrate the mechanism of the glossary functionality. It is usually a mixture of some arbitrary sentences containing terms that allow the demonstration of certain features and some lorem ipsum filler text. For demonstration of certain features a certain length of the *Test Text* is required.

typst

Typst is a new markup-based typesetting system for the sciences. It is designed to be an alternative both to advanced tools like LaTeX and simpler tools like Word and Google Docs.

XML

XML stands for `'eXtensible Markup Language'`. This entry comes from the local pool. https://www.w3.org/XML

iteration

repetition of a mathematical or computational procedure applied to the result of a previous application, typically as a means of obtaining successively closer approximations to the solution of a problem.

typst

See “Typst”.