lemmify

Lemmify is a library for typesetting mathematical theorems in typst. It aims to be easy to use while trying to be as flexible and idiomatic as possible. This means that the interface might change with updates to typst (for example if user-defined element functions are introduced). But no functionality should be lost.

If you are encountering any bugs, have questions or are missing features, feel free to open an issue on GitHub

Basic usage

1. Import lemmify:

```
#import "@preview/lemmify:0.2.0": default-theorems, select-kind
```

2. Generate some common theorem kinds with pre-defined style:

```
#let (
  theorem, lemma, corollary,
  remark, proposition, example,
  proof, theorem-rules
) = default-theorems(lang: "en")
```

3. Apply the generated style:

```
#show: theorem-rules
```

4. Customize the theorems using show rules. For example, to add a block around proofs:

```
#show select-kind(proof): block.with(
  breakable: true,
  width: 100%,
  fill: gray,
  inset: lem,
  radius: 5pt
)
```

5. Create theorems, lemmas, and proofs:

```
#theorem(name: "Some theorem")[
   Theorem content goes here.
]<thm>

#theorem(numbering: none)[
   Another theorem.
]

#proof(link-to: <thm>)[
   Complicated proof.
]
]
@proof and @thm[theorem]
```

The result should now look something like this:

```
Theorem 1 (Some theorem) Theorem content goes here.

Theorem Another theorem.

Proof Complicated proof.
```

Examples

This example shows how corollaries can be numbered after the last theorem.

```
#import "@preview/lemmify:0.2.0": theorem-rules, theorem-kind, select-kind, reset-
counter

#let theorem = theorem-kind("Theorem")
#let corollary = theorem-kind(
    "Corollary",
    group: "CorollaryGroup",
    link-to: select-kind(theorem)
)

#show: theorem-rules
#show select-kind(theorem): it => {it; reset-counter(corollary)}

#theorem(lorem(5))
#corollary(lorem(5))
#corollary(lorem(5))
#corollary(lorem(5))
```

Theorem 2 Lorem ipsum dolor sit amet.

Corollary 2.1 Lorem ipsum dolor sit amet.

Corollary 2.2 Lorem ipsum dolor sit amet.

Theorem 3 Lorem ipsum dolor sit amet.

Corollary 3.1 Lorem ipsum dolor sit amet.

Custom style example

```
#import "@preview/lemmify:0.2.0": default-theorems, get-theorem-parameters

#let my-style-func(thm, is-proof: false) = {
    let params = get-theorem-parameters(thm)
    let number = (params.numbering)(thm, false)
    let content = grid(
        columns: (lfr, 3fr),
        column-gutter: lem,
        stack(spacing: .5em, strong(params.kind-name), number, emph(params.name)),
        params.body
    )
```

```
if is-proof {
  block(inset: 2em, content)
  } else {
    block(inset: 1em, block(fill: gray, inset: 1em, radius: 5pt, content))
  }
}
#let my-style = (
  style: my-style-func,
  proof-style: my-style-func.with(is-proof: true)
)
#let (
 theorem, proof, theorem-rules
) = default-theorems(lang: "en", ..my-style)
#show: theorem-rules
#lorem(20)
#theorem(name: "Some theorem")[
  #lorem(40)
\#lorem(20)
#proof[
  #lorem(30)
]
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat.

Theorem

Some theorem

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat.

Proof

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri.

```
assert(type(text) == str)
assert(type(text) == str)
```

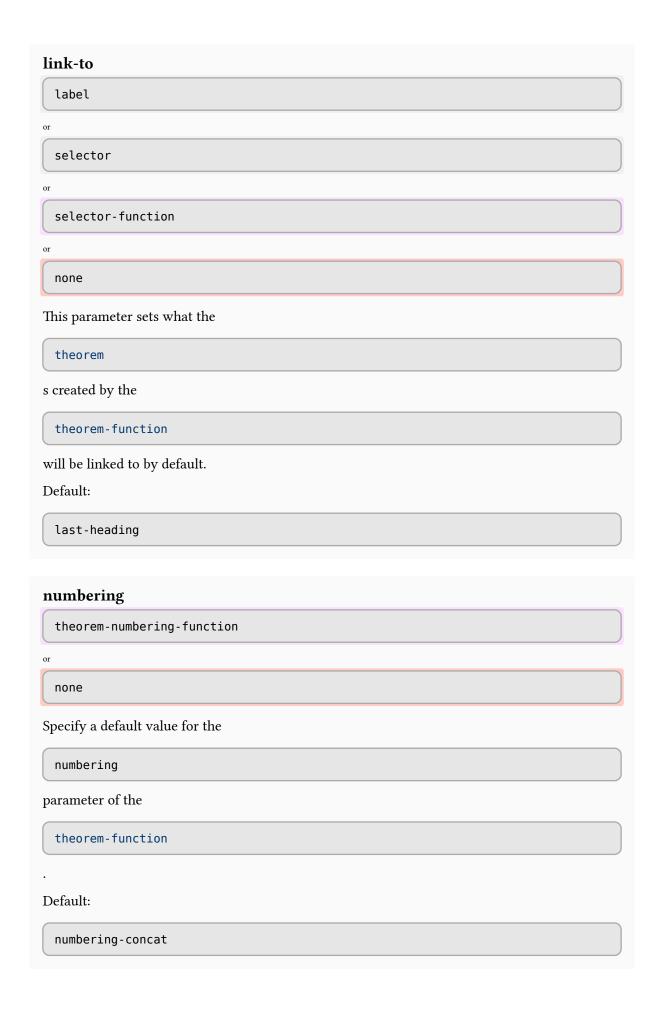
Documentation

theorem-kind

Creates a new

theorem-function **Parameters** theorem-kind(kind-name: str group: str link-to: label selector selector-function none numbering: theorem-numbering-function none subnumbering: numbering-function str none style: $\verb|style-function||$) -> theorem-function

kind-name
str
The name of the theorem kind. It also acts as an identifier together with
group
when using
select-kind
, so it should be unique.
group
str
The group identifier. Each theorem group shares one counter.
Default:
LEMMIFY-DEFAULT-THEOREM-GROUP



subnumbering numbering-function str none The subnumbering is needed to convert the theorem s counter to content, which is then used in the theorem-numbering-function Default: "1" style style-function Specifies how the theorem s will look. This will only be visible once the theorem-rules() have been applied. Default: style-simple

theorem-rules

Apply the style of every

theorem

and handle references to

theorem

s.

Parameters
theorem-rules(content:

content

->

content

default-theorems

Generate a few common theorem kinds in the specified language.

Returns a dictionary of the form

(theorem, lemma, corollary, remark, proposition, example, definition, proof, theorem-rules)

. The

theorem-rules

can be applied using a show statement. If

max-reset-level

is

none

it will be the same as

theorem-rules()

This function accepts all parameters of

theorem-kind()

once for proofs and once for all kinds except for proofs.

default-theorems(group:
str
proof-group:
str
lang:
str
style:
style-function
proof-style:
style-function
numbering:
theorem-numbering-function
none
proof-numbering:
theorem-numbering-function
none
link-to:
label
selector
selector-function
none
proof-link-to:
label
selector
selector-function
none
subnumbering:
numbering-function
str
none
max-reset-level:
int
none
)->

lang
str
The language in which the theorem kinds are generated.
Default:
"en"
max-reset-level
int
or
none
If it is not none the theorem counter will be reset on headings below
max-reset-level
. And if
link-to
is set to
last-heading
higher levels will not be displayed in the numbering.
Default:
none

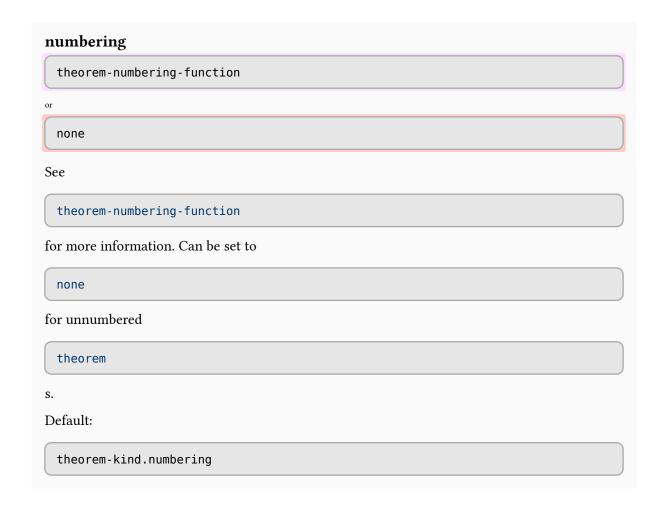
Function types

theorem-function

TODO

theorem-function(name: content str link-to: label selector selector-function none numbering: ${\tt theorem-numbering-function}$ none body: content) -> theorem name content or str The name of the theorem Default: none

link-to
label
or
selector
or
selector-function
or
none
Link the
theorem
to some other content. For
label
s and
selector
s the last match before the
theorem
is used.
Default:
theorem-kind.link-to



theorem-numbering-function

Create combined numberings from

theorem

and the content linked to it.

There are two pre-defined

theorem-numbering-function

s:

numbering-concat()

and

numbering-proof()

.

Parameters theorem-numbering-function(thm: theorem referenced: bool

thm

) ->

theorem

content

The

theorem

for which the numbering should be generated. See also

get-theorem-parameters()

referenced

bool

This is false if the numbering was requested from the

theorem

it belongs to. Otherwise it is false. See

numbering-proof()

as an example.

style-function

Defines how the

theorem

will look. Use

```
get-theorem-parameters()
to get all information stored in the
  theorem
There are two pre-defined
  style-function
s:
  style-simple()
and
  style-reversed()
Parameters
  style-function(thm:
   theorem
   content
selector-function
Useful for more advanced queries. See
  last-heading()
for an example.
Parameters
  selector-function(loc:
   location
   content
   none
```



numbering-function

A normal numbering function as described in the typst documentation.

Parameters

numbering-function(..state:

int
->

content

theorem

Α

theorem

is a

figure

with some additional information stored in one of its parameters.

is-theorem

Check if argument is

theorem

.

Parameters

is-theorem(c:

```
any
```

->

bool

get-theorem-parameters

Extract theorem parameters from figure. Returns a

```
\hbox{\tt dictionary}
```

of the form

```
(body, group, kind-name, name, link-to, numbering, subnumbering, style)
```

.

Parameters

get-theorem-parameters(thm:

theorem

->

dictionary

resolve-link

Return the

content

that is linked to the

theorem

.

Parameters

resolve-link(thm:

theorem

->

content

numbered A numbered is a heading page math.equation or figure that is already embedded in the document (that means it was obtained by a query). The numbering also has to be different from none is-numbered Check if argument is numbered **Parameters** is-numbered(n:

display-numbered

any

bool

Display the numbering of the argument at its location.

display-numbered(n:

```
numbered
->

content
```

Styles

numbering-concat If the linked content

If the linked content is numbered combine it with the numbering of the

If the linked content is numbered combine it with the numbering of the
theorem
Parameters
numbering-concat(
thm:
theorem
referenced:
bool
seperator:
content
str
seperator
content
or
str
The sepeartor is put between both numberings.
Default:
"."

numbering-proof

Copy the numbering of a linked

theorem

if referenced. Otherwise no numbering is returned.

Parameters

```
numbering-proof(
thm:

theorem

referenced:

bool
)
```

style-simple

Simple theorem style. Check the documentation for images.

Parameters

```
style-simple(
thm:

theorem

qed:

bool

ped

bool

Select if a box should be shown at the end.

Default:

false
```

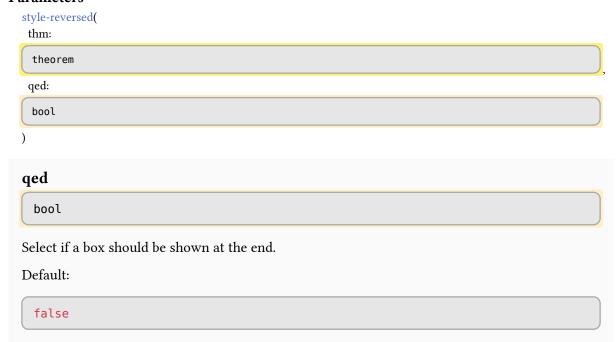
style-reversed

Reverses numbering and

```
kind-name
, otherwise the same as

style-simple()
```

.



Selectors

The selectors can be used in show-rules to customize the

theorem

s styling as well as with the

link-to

parameter.

last-heading

Selector-function which selects the last heading.

Parameters

last-heading(
ignore-unnumbered:

bool

max-level:

int

none

loc:

location
)->

heading

none

ignore-unnumbered bool Use the last heading which is numbered. Default: false max-level int or none Do not select headings above this level. Default: none

select-group

Generate selector that selects all theorems of the same group as the argument.

Parameters

```
select-group(thm-func:
    theorem-function
->
selector
```

select-kind

Generate selector that selects only theorems that were create from the

```
theorem-function
```

.

select-kind(thm-func:

```
theorem-function
->
selector
```

Resetting counters

reset-counter

Reset theorem group counter to zero. The result needs to be added to the document.

Parameters

reset-counter(thm-func:

```
theorem-function

->

content

thm-func
```

theorem-function

The group is obtained from this argument.

reset-counter-heading

Reset counter of theorem group on headings with at most the specified level.

Parameters

reset-counter-heading(
thm-func:

theorem-function

max-level:

int

content:

content

)->

thm-func

theorem-function

The group is obtained from this argument.

max-level

int

Should be at least 1.