# 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.

## Basic usage

1. Import the Lemmify library

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

2. Generate some common theorem kinds with pre-defined styling

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

3. Apply the generated styling

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

4. Customize the styling using show rules. For example, to add a red box around proofs

```
#show select-kind(proof): box.with(stroke: red + 1pt, inset: 1em)
```

5. Create theorems, lemmas, and proofs

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

#proof(link-to: <thm>)[
  Complicated proof.
]<proof>

@proof and @thm[theorem]
```

The result should now look something like this

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

**Proof** Complicated proof. □

Proof 1 and Theorem 1

## Examples

As another example we will number corollarys 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))
#theorem(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

This examples shows how custom style functions can be defined.

```
#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: (1fr, 3fr),
    column-gutter: 1em,
    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**
4
*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.

## Documentation

- default-theorems()
- theorem-kind()
- theorem-rules()

### default-theorems

Generate a few common theorem kinds in the specified language.

This function accepts all parameters of `theorem-kind` once for proofs and once for all kinds except for proofs. So for information on the missing parameters refer to `theorem-kind`.

**Parameters**

```
default-theorems(
  group: str,
  proof-group: str,
  lang: str,
  style: style-function,
  proof-style: style-function,
  numbering: theorem-numbering-function,
  proof-numbering: theorem-numbering-function,
  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
) -> dictionary
```

### 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`

## theorem-kind

Create a new theorem function. The style is only visible once the `theorem-rules` have been applied. The generated functions will be referred to as `kind-functions` and the `figures` created by it will be referred to as `theorems`.

**Parameters**

```
theorem-kind(
  kind-name: str,
  group: str,
  link-to: label selector selector-function none,
  numbering: theorem-numbering-function,
  subnumbering: numbering-function str none,
  style: style-function
)
```

### kind-name `str`

The name of the theorem kind. It 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`

**link-to** `label` or `selector` or `selector-function` or `none`

Link the `theorem` to some other content. See `numbering-concat` for how this is used. For `labels` and `selectors` the last one before the `theorem` is used. `selector-functions` are functions `(location) -> content` which can be used for more advanced queries.

Default: `none`

**numbering** `theorem-numbering-function`

A function `(theorem, bool) -> content`. The `bool` argument is true only if the numbering is requested from a reference instead of from the theorem itself (see numbering-proof).

Default: `numbering-concat`

**subnumbering** `numbering-function` or `str` or `none`

The subnumbering is needed to convert the theorems counter to content, which is then used in the `theorem-numbering-function`.

Default: `"1"`

**style** `style-function`

A function `(thm) -> content` which is used to style the theorem.

Default: `style-simple`

### theorem-rules

Apply the style of every `theorem` and handle references to `theorems`.

#### Parameters

theorem-rules(content: `content`) -> `content`

## Styles

There are a few pre-defined `style-functions` and `theorem-numbering-functions`.

- numbering-concat()
- numbering-proof()
- style-reversed()
- style-simple()

### numbering-concat
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-reversed
Reverses numbering and `kind-name`.

#### Parameters

```
style-reversed(
  thm: theorem,
  qed: bool
)
```

**qed** `bool`

Select if a box should be shown at the end.

Default: `false`

### style-simple
Simple theorem style. Check the documentation for images.

#### Parameters

```
style-simple(
  thm: theorem,
  qed: bool
)
```

**qed** `bool`

Select if a box should be shown at the end.

Default: `false`

## Selectors

The selectors can be used in show-rules to customize the `theorems` styling as well as with the `link-to` parameter.

- last-heading()
- select-group()
- select-kind()

### 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 first heading which is numbered.

Default: `false`

**max-level** `int` or `none`

TODO

Default: `none`

### select-group

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

#### Parameters

```
select-group(kind-func: kind-function) -> selector
```

### select-kind

Generate selector that selects only theorems that were create from the provided `kind-function`.

#### Parameters

```
select-kind(kind-func: kind-function) -> selector
```

## Resetting counters

- reset-counter()
- reset-counter-heading()

### reset-counter

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

#### Parameters

```
reset-counter(kind-func: kind-function ) -> content
```

**kind-func** `kind-function`

The group is obtained from this kind function.

### reset-counter-heading

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

#### Parameters

```
reset-counter-heading(
  kind-func: kind-function ,
  max-level: int ,
  content: content
) -> content
```

**kind-func** `kind-function`

The group is obtained from this kind function.

**max-level** `int`

Should be at least 1.

## Theorem utilities

The functions in the remaining two sections are only needed when defining custom style or theorem-numbering-functions.

- get-theorem-parameters()
- is-theorem()
- resolve-link()

### get-theorem-parameters

Extract theorem parameters from figure. Returns a dictionary of the form (body, group, kind-name, name, link-to, numbering, subnumbering, style).

#### Parameters

```
get-theorem-parameters(thm: theorem ) -> dictionary
```

### is-theorem

Check if argument is of type theorem.

### Parameters

is-theorem(c: `any` ) -> `bool`

### resolve-link

Return the content that is linked to the theorem.

### Parameters

resolve-link(thm: `theorem` ) -> `content`

## Numbered utilities

- display-numbered()
- is-numbered()

### display-numbered

Display the numbering of the argument at its location.

### Parameters

display-numbered(n: `numbered` ) -> `content`

### is-numbered

Check if argument is numbered. That means it is one of `heading`, `page`, `math.equation` or `figure` and its numbering is not `none`.

### Parameters

is-numbered(n: `any` ) -> `bool`