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

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
2
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.

As another example we will number corollarys after the last theorem.

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