

Theoretic

Contents

1 Summary	1	A.4 proof-fmt-prefix.	A2
Theorem 1.2 Foo	1	A.5 qed	A3
2 Setup	1	A.6 restate	A3
3 Examples	3	Proposition A.1 Funky!	A3
Example 3.1 A complicated example		A.7 show-ref	A4
showing some configuration		Theorem A.3 Pythagoras	A4
possibilities	3	Theorem Only Named	A4
Theorem 2 Name	3	A.8 solutions	A5
Example 4 Named Example	3	A.9 theorem	A5
		Theorem 2 Pythagoras	A5
A Detailed Documentation of all		A.10 toc	A8
Exported Symbols	A1	Theorem A.12 Z	A9
A.1 fmt-body.	A1	Theorem A.13 A	A9
A.2 fmt-prefix	A1	A.11 thm-counter	A10
A.3 proof	A1	B Solutions	A10

1 Summary

This package provides opinionated functions to create theorems and similar environments.

Default theorem environment and provided presets:

```
#theorem[This is a theorem]
#proof[This is a proof.]

#theorem(title: "Foo", label: <thm:foo>)[
  This is a theorem
]
#proof(title: [@thm:foo[-]])[
  This is a proof.
  - Ends with a list or enum?
    No problem.#qed()
]
```

```
Theorem 1.1 This is a theorem

Proof. This is a proof. □

Theorem 1.2 (Foo) This is a theorem

Proof of Theorem 1.2. This is a proof.
• Ends with a list or enum? No problem. □
```

References can be controlled by passing some specific supplements, see Section A.7 for more details.

2 Setup

Put the following at the top of your document:

```
#import "@preview/theoretic:0.1.0" as theoretic: theorem, proof, qed

// Otherwise, references won't work.
#show ref: theoretic.show-ref

// set up your needed presets
#let corollary = theorem.with(kind: "corollary", supplement: "Corollary")
#let example = theorem.with(kind: "example", supplement: "Example", number: none)
// ..etc
```

See Section A.9 for a detailed description of customization options.

Except for `#show ref: theoretic.show-ref`, no “setup” is necessary. All configuration is achieved via parameters on the Section A.9 function, use `theorem.with(..)` for your preset needs.

The numbering of theorems is not configurable, but can be disabled (`number: none`) or temporarily overridden (`number: "X"` or `number: 2`). If your headings are numbered, it will use top-level heading numbers as the first component, otherwise it will simply number your theorems starting with Theorem 1.

Use `#theoretic.toc()` to get a list of theorems, list of definitions, a table of contents containing theorems, etc.

Put `#theoretic.solutions()` at the end of your document to get the solutions (every theorem environment accepts a second positional argument, which gets used as the solution). (Nothing will appear unless there are solutions to show.)

Exercise 2.1 Go look for the solution of this exercise at the end of this document.¹

¹Solution in Appendix

3 Examples

Example 3.1 (A complicated example showing some configuration possibilities)

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

Theorem 1 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed.

Proof of Theorem 1. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna. *QED.*

Theorem 2 Name Lorem ipsum dolor sit amet, consectetur.

Example 3 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do.

Example 4 (Named Example) To avoid having examples and such show up in the toc, use the `toc.exclude` parameter.

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

```
#set text(font: "Besley*", size: 9pt)
#let theorem = theorem.with(
  fmt-prefix: (s, n, t) => {
    text(font: "Besley* Narrow Semi")[#s #n]
    if t!= none {
      h(2pt)
      box(fill: oklch(70%, 0.17, 307.4deg, 20%), outset: (y: 4pt), inset: (x: 2pt), radius:
2pt, text(fill: oklch(44.67%, 0.15, 307.4deg), font: "Besley* Semi", t))
    }
    h(1em)
  },
  block-args: (
    stroke: (left: 0.5pt + oklch(44.67%, 0.15, 307.4deg)),
    outset: (left: 4pt, right: 0pt, y: 4pt),
  ),
)
#let ex = theorem.with(
  kind: "example",
  supplement: "Example",
  fmt-prefix: (s, n, t) => {
    text(font: "Besley*", stretch: 85%)[#s #n]
    if t!= none [ (#t)]
    h(1em)
  },
  fmt-body: (b, _) => { emph(b) },
)
#let qed = qed.with(suffix: smallcaps[#h(1fr)_qed._])
#let proof = proof.with(fmt-suffix: qed)

#lorem(20)
#theorem(label: <e.g>)[#lorem(9)]
#proof(title: [e.g.])[#lorem(18)]
#theorem(title: "Name")[#lorem(6)]
#ex[#lorem(10)]
#ex(title: "Named Example")[
  To avoid having examples and such show up in the toc, use the `toc.exclude` parameter.
]
#lorem(20)
```

A Detailed Documentation of all Exported Symbols

A.1 fmt-body

Function to format the body

Default value of `theorem.fmt-body`.

Parameters

```
fmt-body(  
  body: content,  
  solution: content  
) -> content
```

body `content`

Theorem content.

solution `content`

A.2 fmt-prefix

Function to run at beginning of theorem.

Default value of `theorem.fmt-prefix`.

```
#fmt-prefix([Theorem], [1.34], none)...  
  
#fmt-prefix([Theorem],  
[1.34], [Pythagoras])...
```

Theorem 1.34 ...

Theorem 1.34 (Pythagoras) ...

Parameters

```
fmt-prefix(  
  supplement: content,  
  number: content none,  
  title: content none  
) -> content
```

A.3 proof

This is just `theorem()` with different defaults.

```
#proof[#lorem(5)]  
#proof(title: [@pythagoras(!)])[#lorem(6)]
```

Proof. Lorem ipsum dolor sit amet. \square

Proof of Pythagoras (Theorem A.3). Lorem
ipsum dolor sit amet, consectetur. \square

Parameters

```
proof(  
  kind: content,  
  supplement: content,  
  number: string,  
  fmt-prefix: function,  
  fmt-suffix: function,  
  ..args: arguments  
) -> content
```

kind `content`

Default: `"proof"`

supplement `content`

Default: `"Proof"`

number `string`

Default: `none`

fmt-prefix `function`

Default: `proof-fmt-prefix`

fmt-suffix `function`

Default: `qed.with(force: false)`

..args `arguments`

Same as for `theorem()`.

A.4 proof-fmt-prefix

Function to run at beginning of proof.

Default value of `proof.fmt-prefix`.

```
#proof-fmt-prefix([Proof], none, none)...  
  
#proof-fmt-prefix([Proof],  
none, [@pythagoras])...
```

Proof. ...

Proof of Theorem A.3 (Pythagoras). ...

Parameters

```
proof-fmt-prefix(  
  supplement: content,  
  number: content none,  
  title: content none  
) -> content
```

A.5 qed

Place a QED mark and clear the `_thm_needs_qed` flag, so that the theorem environment itself won't place one.

See `proof.fmt-suffix`.

Parameters

```
qed(  
  suffix: content,  
  force: boolean  
) -> content
```

force `boolean`

Whether to place suffix no matter the `_thm_needs_qed` flag.

Default: `true`

A.6 restate

Re-state a theorem.

It will reuse the original kind, supplement, number, title, and body. It will *not* re-emit the solution or label, and it will use `toctitle: none` to avoid duplicate toc entries.

It is currently not able to pick up any of the other configuration of the original theorem, therefore pass `restate.is` if you modified e.g. any of the `fmt-s`.

```
#let proposition = theorem.with(  
  kind: "proposition",  
  supplement: "Proposition",  
  fmt-body: (b, s) => { text(fill: red, {b;s})  
})  
#proposition(title: "Funky!", label: <funky>)  
[Blah _blah_ blah.]  
Restated:  
#restate("funky")  
Restated with explicit kind:  
#restate(<funky>, is: proposition)
```

*Proposition A.1 (Funky!) Blah **blah** blah.*

Restated:

*Proposition A.1 (Funky!) Blah **blah** blah.*

Restated with explicit kind:

*Proposition A.1 (Funky!) Blah **blah** blah.*

Parameters

```
restate(
  label: label string,
  is: function
) -> content
```

label label or string

Label of the theorem to restate.

is function

Theorem function to use.

Default: theorem

A.7 show-ref

Show-rule-function to be able to @ labelled theorems.

Use via ``#show ref: show-ref`` at the beginning of your document.

```
#show ref: theoretic.show-ref
#theorem(label: <fact>, supplement: "Fact")
[#lorem(2)]
#theorem(label: <pythagoras>, title:
"Pythagoras")[#lorem(2)]
#theorem(label: <zl>, title: "Only Named",
number: none)[#lorem(2)]
#theorem(label: <y>, number: "Y")[#lorem(2)]
#theorem(label: "5", number: none)[#lorem(2)]
```

As a consequence of @fact and
@pythagoras[!!]...

Fact A.2 Lorem ipsum.

Theorem A.3 (Pythagoras) Lorem ipsum.

Theorem (Only Named) Lorem ipsum.

Theorem Y Lorem ipsum.

Theorem Lorem ipsum.

As a consequence of Fact A.2 and Pythagoras
(A.3)...

The reference can be controlled via the supplement passed:

	BOTH	WITHOUT TITLE	WITHOUT NUMBER	NEITHER
@ref (Full)	Theorem A.3 (Pythagoras)	Fact A.2 / Theorem Y	Theorem (Only Named)	Theorem
@ref[-] (Compact)	Theorem A.3	Fact A.2 / Theorem Y	Theorem (Only Named)	Theorem
@ref[--] (Number)	A.3	A.2 / Y	(Only Named)	Theorem
@ref[!] (Inverted)	Pythagoras (Theorem A.3)	Fact A.2 / Theorem Y	Only Named (Theorem)	Theorem
@ref[!!!] (Compact Inverted)	Pythagoras (A.3)	Fact A.2 / Theorem Y	Only Named	Theorem
@ref[!!!!] (Name)	Pythagoras	Fact A.2 / Theorem Y	Only Named	Theorem
@ref[?] (Kind)	Theorem	Fact / Theorem	Theorem	Theorem
@ref[Custom] (Custom Supplement)	Custom A.3 (Pythagoras)	Custom A.2 / Custom Y	Custom (Only Named)	Custom

Note: the fact that references and links in this document are underlined in gray is achieved with a separate `@show` link: `it => underline(..)` rule, and not because of this function.

Parameters

`show-ref(it: ref)`

A.8 solutions

List all solutions, if any.

See Section B for how it looks. Currently not customizable, working on it.

Parameters

`solutions(title: content) -> content`

title `content`

Title/heading to use.

Default: `"Solutions"`

A.9 theorem

Theorem Environment

```
#set heading(numbering: none)

#theorem[If the headings are not numbered,
theorem numbering starts at 1.][

= Heading
#theorem(title: "Pythagoras")[
  Given a right-angled triangle, the length
  of the hypotenuse squared is equal to the
  sum of the squares of the remainig sides'
  lengths.
][[
```

Theorem 1 If the headings are not numbered, theorem numbering starts at 1.

Heading

Theorem 2 (Pythagoras) Given a right-angled triangle, the length of the hypotenuse squared is equal to the sum of the squares of the remainig sides' lengths.

Parameters

```
theorem(  
  fmt-prefix: function,  
  fmt-body: function,  
  fmt-suffix: none content,  
  block-args: dict,  
  kind: string,  
  supplement: content,  
  number: auto none integer content,  
  title: none content,  
  toctitle: auto content,  
  label: label string,  
  body: content,  
  ..solution: content  
) -> content
```

fmt-prefix function

Default: fmt-prefix

fmt-body function

Default: fmt-body

fmt-suffix none or content

Will be called at the end of the theorem if `_thm_needs_qed` hasn't been cleared. (E.g. by `qed()`)

Default: none

block-args dict

Arguments to pass to the `#block[]` containing the theorem.

Default: (:)

kind string

Used for filtering e.g. when creating table of theorems.

Default: "theorem"

supplement content

What to label the environment.

It is recommended to keep kind and supplement matching (except for “subtypes”, e.g. one might have the kind of “Example” and “Counter-Example” both as "example")

Default: "Theorem"

number `auto` or `none` or `integer` or `content`

- If `auto`, will continue numbering from last numbered theorem.
- If `integer`, it will continue the numbering of later theorems from the given number.
- If `content`, it is shown as-is, with no side-effects.

```
#let corollary = theorem.with(  
  kind: "corollary",  
  supplement: "Corollary")  
  
#corollary[#lorem(2)]  
  
#corollary(number: none)[Skip number]  
#corollary[Resume numbering]  
  
#corollary(number: "X")[Custom "number"]  
#corollary[Resume numbering]  
  
#corollary(number: 10)[Set number]  
#corollary[Continue from set number]
```

Corollary A.3 Lorem ipsum.
Corollary Skip number
Corollary A.4 Resume numbering
Corollary X Custom “number”
Corollary A.5 Resume numbering
Corollary A.10 Set number
Corollary A.11 Continue from set number

Default: `auto`

title `none` or `content`

Title of the Theorem. Usually shown after the number.

Default: `none`

toctitle `auto` or `content`

Title of the Theorem to be used in outlines. `auto` to use the title.

Default: `auto`

label `label` or `string`

Label (for references)

note: Simply putting a `<label>` after the `#theorem[]` does not work for referencing.

Default: `none`

body `content`

Theorem body

..solution `content`

Optional Solution. Pass zero or one positional arguments here.

See `solutions()`.

A.10 toc

Create an outline that includes named theorems.

Can be styled with show rules for `outline.entry()`. See the source code of this manual for an example.

```
#heading(outlined: false, level: 3)[
  Contents
]
#toc(depth: 1)
```

Contents

1 Summary	1
Theorem 1.2 (Foo)	1
2 Setup	1
3 Examples	3
Example 3.1 (A complicated example showing some configuration possibilities)	3
Theorem 2 (Name)	3
Example 4 (Named Example)	3
A Detailed Documentation of all Exported Symbols	A1
Proposition A.1 (Funky!)	A3
Theorem A.3 (Pythagoras)	A4
Theorem (Only Named)	A4
Theorem 2 (Pythagoras)	A5
Theorem A.12 (Z)	A9
Theorem A.13 (A)	A9
B Solutions	A10

Parameters

```
toc(
  depth: integer,
  exclude: list (string),
  level: integer,
  fill: content,
  sort: bool
) -> content
```

depth integer

Maximum depth of headings to consider

Default: 2

exclude `list (string)`

list of `theorem.kinds` to ignore.

```
#heading(outlined: false, level: 3)[
  Table of Examples
]
#toc(
  depth: 0,
  exclude: ("proof", "solution", "theorem")
)
```

Default: `("proof", "solution")`

Table of Examples

Example 3.1 (A complicated example showing some configuration possibilities) 3
Example 4 (Named Example) 3
Proposition A.1 (Funky!) A3

level `integer`

Fake level to use for theorems. Set this to some level greater than the depth if to avoid conflict in your show rules for `outline.entry`.

Default: `4`

fill `content`

Fill for outline entries

Default: `repeat[.]`

sort `bool`

Whether to sort the entries alphabetically. Only respected if depth is 0.

```
#theorem(title: "Z") [Blah blah.]
#theorem(title: "A") [Blah blah.]
#heading(outlined: false, level: 3)[
  Sorted Table of Theorems
]
#toc(
  depth: 0,
  sort: true,
)
```

Theorem A.12 (Z) Blah blah.

Theorem A.13 (A) Blah blah.

Sorted Table of Theorems

Theorem A.13 (A) A9
Example 3.1 (A complicated example showing some configuration possibilities) 3
Theorem 1.2 (Foo) 1
Proposition A.1 (Funky!) A3
Theorem 2 (Name) 3
Example 4 (Named Example) 3
Theorem (Only Named) A4
Theorem A.3 (Pythagoras) A4
Theorem 2 (Pythagoras) A5
Theorem A.12 (Z) A9

Default: `false`

A.11 thm-counter `counter`

Counts theorems.

In most cases, it is not necessary to reset this manually, it will get updated accordingly if you pass an integer to `theorem.number`.

B Solutions

Solution of Exercise 2.1. Yay! you found it!