

# Example Lecture

Lecture Name - Contents

I. A general overview ..... 1

1. Example ..... 1

    → thrm. I.1. The theorem's title ..... 1

    → ex. I.1. An example ..... 1

    → rmk. I.1. Warning ..... 1

    → thrm. I.2. Some Theorem with Proof ..... 1

    → thrm. I.3. A shortcut ..... 1

    → ex. I.2. .... 1

2. Setup ..... 1

    2.1. How to start ..... 1

        → thrm. I.4. This is what you just created ..... 1

        → thrm. I.5. Title ..... 2

    2.2. The Template ..... 2

II. Configuring the template ..... 2

III. Styling ..... 2

---

26.08.2025

# A general overview

I

This template offers the possibility to create theorem environments with colored boxes and titles.

## 1. Example

In this case I created the three environments ‘theorem’, ‘example’ and ‘remark’. In the ‘constructor’ (theorem-env) I gave each environment an unique body color. In the config I gave each environment a link-color, abbreviation and a bool for which environments I want to see in the outline. The environments look like this:

### Theorem I.1: The theorem’s title

The content of the theorem. This can also be a formula, like

$$E = mc^2 \quad (\text{I.1})$$

or a figure like

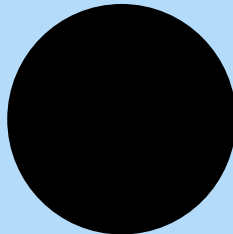


Figure I.1 A black hole.

### Example I.1: An example

Here you could give an example of something.  
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do.

### Remark I.1: Warning

Be careful

I also added a fourth one as a shortcut for a box with an extra box at the bottom. Usually you would do

### Theorem I.2: Some Theorem with Proof

This would be some kind of theorem that needs to be proven:

$$a + b = c$$

This is a usually hidden box for remarks or proofs in a theorem environment.

### Theorem I.3: A shortcut

But to make it a little bit faster when wanting to write live at a lecture I created theoremr():

Remark/Proof,...

All of these boxes are breakable too, if you want!

### Example I.2:

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

licet transferre in voluptatem, ut postea variari voluptas distinguere possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum defuturum, quas natura non depravata desiderat. Et quem ad me accedis, saluto: 'chaere,' inquam, 'Tite!' lictores, turma omnis chorusque: 'chaere, Tite!' hinc hostis mi Albucius, hinc inimicus. Sed iure Mucius. Ego autem mirari satis non queo unde hoc sit tam insolens domesticarum rerum fastidium. Non est omnino hic docendi locus; sed ita prorsus existimo, neque eum Torquatium, qui hoc primus cognomen invenerit, aut torquem illum hosti detraxisse, ut aliquam ex eo est consecutus? – Laudem et caritatem, quae sunt vitae.

## 2. Setup

I created this because I was bored but mostly because I wanted to have a nice template I can use for taking notes in lectures. In typst it is much easier to write live during the lecture due to on-the-fly compilation and the much easier commands in math mode etc.

### 2.1. How to start

To start just download the conf.typ file and create a new file (maybe main.typ) in the same folder. Then write

```
#import "conf.typ": conf, theorem-env, chapter, new-lecture, endcol
```

at the very top. I also suggest using the equate package with the following settings:

```
#import "@preview/equate:0.3.2": equate
#show: equate.with(
  breakable: true,
  sub-numbering: false,
  number-mode: "label",
)
```

Then you can already create your first theorem environment with the imported theorem-env function:

```
#let theorem(title, content, extra: "") = {
  theorem-env(
    title,
    content,
    rgb(180, 220, 250),
    kind: "theorem",
    supp: "Theorem",
    extra: extra,
  )
}
```

When creating new environments don't forget about changing the parameter kind since it controls the counter.

### Theorem I.4: This is what you just created

This box (called with #theorem[title][content]) is the result. You can write what ever you want in here. Calling it with the optional argument extra like this:

```
#theorem(extra: [This is a remark])[Title][Body]
```

adds another, darker, box at the bottom (see [thrm. I.2](#)).

You can also create another box environment with the same counter but where the bottom box is mandatory:

```
#let theoremr(title, content, extra) = {
  theorem-env(
    title,
    content,
    theorem-color,
    kind: "theorem",
    supp: "Theorem",
    extra: extra,
  )
}
```

The content produced by:

```
#theoremr[Title][Body][Extra]
```

is:

Theorem I.5: Title
Body
Extra

## 2.2. The Template

Adding

```
#show: conf
```

to the preamble after importing it, already gives a much nicer result. A lot has happened now, because all default values and settings of the template were loaded. You must change one of these settings urgently: The parameter `theorem-config` controls the numbering, styling and more of all created environments. In our case it should look like this:

```
#show: conf.with(
  theorem-config: (
    theorem: (
      link-color: theorem-link-color,
      show-in-outline: true,
      abbreviation: "thrm.",
    ),
  ),
)
```

Since ``theorem`` and ``theoremr`` share the same ``kind`` there don't need to be more environments listed.

# Configuring the template

You can configure a lot more. If you have created three environments called `theorem`, `example` and `remark` your config could look like this:

```
#show: conf.with(
  cols: 2,
  lecture-name: [Lecture Name],
  author-name: [author name],
  theorem-config: (
    theorem: (
      link-color: rgb("#0091ff"),
      show-in-outline: true,
      abbreviation: "thrm.",
    ),
    example: (
      link-color: rgb("#4be469"),
      show-in-outline: true,
      abbreviation: "ex.",
    ),
    remark: (
      link-color: rgb("#fd2e2e"),
      show-in-outline: true,
      abbreviation: "rmk.",
    ),
  ),
  img-tab-eq-link-color: (green, orange, blue),
```

```
show-dates: true,
abbreviate-links: true,
breakable-thrms: true,
chapter-color: rgb(7, 16, 73),
chapter-number-color: rgb("#80808097"),
shadow-color: rgb(0, 0, 0, int(255 * 0.3)),
)
```

The final argument is: `coverfunc`. If you want to create a cover, use a function which returns the coverpage (e.g. `#cover(args)`). Then you can include it in the template config with:

```
#show: config.with(
  ...
  cover-func: cover.with(
    arg1: ...,
    arg2: ...,
    ...,
  )
  ...
)
```

The default cover is no cover at all. The very basic cover of this document is found in `cover.typ`

By now you are probably wondering what that date on the right side means. As this is meant for notetaking I added a function

```
#new-lecture(day, month)[content].
```

This shows the given date in the page margin if the toggle argument `show-dates` is set to `true` in the config.

## Styling



29.08.2025

Also: I tried to configure this in a way that the lecture can be printed in book-format afterwards. This is the reason why there are different margins and headers for even/odd pages and why there are some "unnecessary" pagebreaks. Feel free to explore the template and to criticize it as much as you want. This is my first project and I hope you like it. I am sorry for the not-very-well-written code. This is my first typst-project and I used it to learn writing with typst.

---