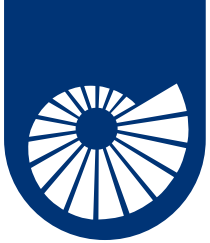


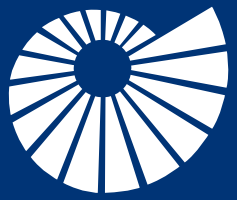
The Typst `dmunipi` theme

A guide on usage and customization of the theme

Here you should put the name of your course or degree

Francesco Baldino (025613)

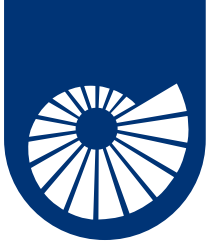
25 December 2025



Dipartimento
di Matematica
Università di Pisa

Table of Contents

- Introduction
- Basic usage
- Customization



Introduction

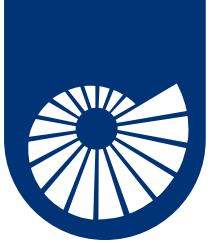
This template is a Typst porting of Fabio Durastante's `dmslide` theme for Beamer. It's made with the Touying package for creating slides and slide themes.

Some features of the original theme might be missing, but all the fundamentals are already implemented, and should be easier to use thanks to Typst.

In the following you will find a brief introduction on how to use Typst and this theme to prepare slides.

Table of Contents

2 Basic usage



Getting started with Typst

2 Basic usage

If you're new to Typst, the easiest way to start a new project is with the Typst web app, similarly to Overleaf for LaTeX. The web app is free, but it requires creating an account.

Once you're set up in the web app, you can simply create a new empty project and procede from there.

If you're more experienced or prefer to work locally, you can use the CLI typst compiler, which is open source. You can find instructions on how to install the CLI in the GitHub repo.

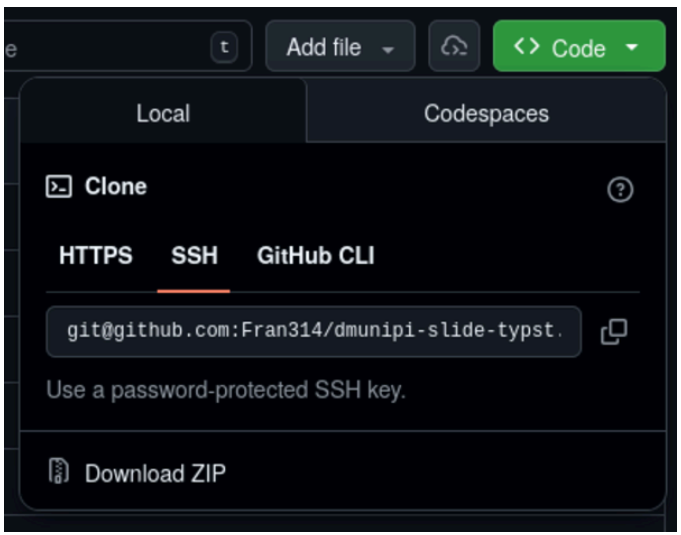
Once you've installed the CLI, creating a new project is as simple as creating a new folder and a `main.typ` file.

Getting started with this theme

2 Basic usage

To use this theme, you have to import the files of this repository in your empty project.

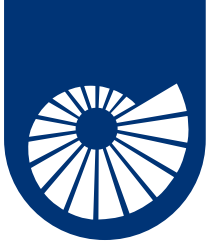
You can download the `ZIP` of this repository directly from GitHub through the green `<> Code` button in the top right.



From here you can either unzip it into your project folder if you're working locally, or unzip it and upload the resulting folder into the empty project on the web app.

Now you can start editing the `main.typ` file! Note that only the `main.typ` file and `theme` folder are necessary, you can safely delete anything else.

Tip: is the font rendering weird? Go to the troubleshooting section



The `main.typ` file

2 Basic usage

Ignoring the comments, the `main.typ` file starts with the code on the right.

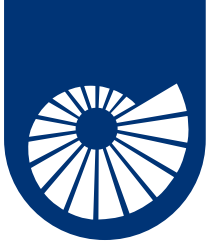
This is how the theme is imported and activated.

It's not important to understand right now what each keyword does, you just need to know that this is where the theme is activated and global informations like your title are declared.

If you want to override some of the styling of the theme, it must be done either here or below.

You can read the omitted comments to understand better what each option does and which options are available.

```
#import "./theme/lib.typ": *

#show: dmunipi-theme.with(
  config-info(
    title: [Your title],
    author: [Your name],
    short-title: [yr ttl],
    subtitle: [Your subtitle],
    course: [Name of your course],
    IDnumber: [012345],

    date: datetime.today(),
  ),
)
```



Anatomy of a Typst document

2 Basic usage

In its most basic form, a Typst document is composed of headings (sections, subsections, …) and paragraphs of text, like the following:

```
= Basic Usage

== Getting started with Typst

If you're new to Typst, the easiest way to start...
```

Within this theme, a section creates a new *“Table of Contents”* slide, and a subsection creates a new slide having the subsection title as its header, and the section title as the subheader.

Basic formatting

2 Basic usage

Simple text formatting such as bullets, numbering and text styling can be done as follows:

```
These are bullet points:
- something
- something else

The following are enumerated:
+ the first item
+ the second item

This word is *bold* while this word
is _italic_. This word is
`monospace`
```

➡️

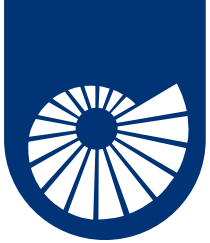
These are bullet points:

- something
- something else

And these are enumerated:

1. the first item
2. the second item

This word is **bold** while this word is *italic*.
This word is `monospace`



Math text

2 Basic usage

Writing math in Typst is much easier than LaTeX, look at the following examples:

```
$ A = pi r^2 $

$ "area" = pi dot "radius"^2 $

$ cal(A) :=
    { x in RR | x "is natural" } $

#let x = 5
$ #x < 17 $

$ x_(n+1) = (x_n + a/x_n) / 2 $
```

➡️

$$A = \pi r^2$$

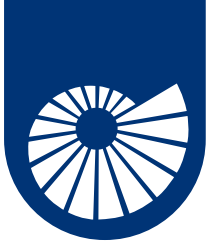
$$\text{area} = \pi \cdot \text{radius}^2$$

$$\mathcal{A} := \{x \in \mathbb{R} \mid x \text{ is natural}\}$$

$$5 < 17$$

$$x_{n+1} = \frac{x_n + \frac{a}{x_n}}{2}$$

you can find more informations here.



Coding

2 Basic usage

You can even display code with the correct syntax highlight by specifying the language:

````
Here is some rust code
```rust
fn main() {
}
```
And here is some Typst!
```typ
= Basic Usage

== Getting started with Typst

If you're new to Typst...
```
````

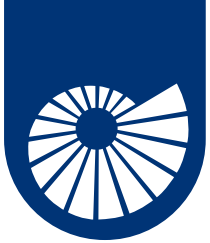
➡️

Here is some rust code

```rust
fn main() {
}
```

And here is some Typst!

```typ
= Basic Usage

== Getting started with Typst

If you're new to Typst...
```



Functions

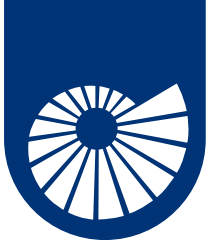
2 Basic usage

Typst also has functions, which are denoted by a `#`. Some common functions that you might find useful are:

- `#align()` to change the alignment of the given content.
- `#v()` & `#h()` to add vertical and horizontal spacing
- `#columns()` to break the given content into columns
- `#text()` to change the style of the text
- `#image()` to add images (`.png`, `.jpg`, `.gif`, `.svg`, `.pdf`, `.webp`), and `#figure()` to wrap content and add a caption
- `#table()` and `#grid()` to organize content in grids and tables

You can find all the documentation for these (and more!) functions by searching here.

There are also the extremely powerful commands `#set` and `#show` which will take a while to understand and are usually used to change the global styling.



`#basic-block` and `#title-block`

In this theme there are also two useful functions for wrapping content in blocks, which are `#basic-block()` for a standard block and `#title-block()` for a block with a title header:

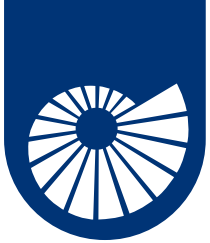
```
#basic-block[This is a simple block]

#title-block(title: "A title")[
  This block has an header with a
title
]
```

➡️

This is a simple block

A title

This block has an header with a title



Theorems and other statements

This theme comes with useful functions for statements such as theorems:

```
#theorem(title: "Euclid's Theorem")[
  There are infinitely many primes.
]

#theorem(number: "5")[
  There are infinitely many primes.
] <thm:euclid>

A reference to @thm:euclid
```

➡️

Theorem (Euclid's Theorem)

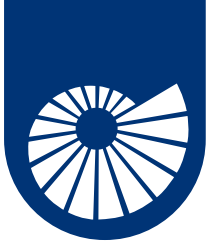
There are infinitely many primes.

Theorem 5

There are infinitely many primes.

A reference to Theorem 5

There are also: `#definition`, `#lemma`, `#corollary`, `#proposition`, `#axiom`, `#postulate`, `#assumption`, `#property` and `#conjecture`!



Animations: `#pause` and `#meanwhile`

2 Basic usage

Of course, you might want to gradually display the content of the slide!

You can use `#pause` to create a slide with only the content preceding the `#pause`.

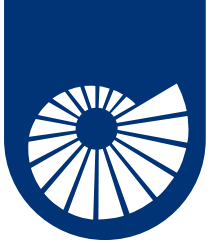
If there's some content that you want displayed despite the `#pause`, you can put it after a `#meanwhile`:

```
Show this content #pause and then
this #pause and finally this!

#meanwhile
But always show this!
```

➡️

Show this content

But always show this!



Animations: `#pause` and `#meanwhile`

2 Basic usage

Of course, you might want to gradually display the content of the slide!

You can use `#pause` to create a slide with only the content preceding the `#pause`.

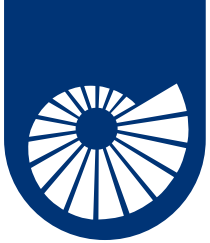
If there's some content that you want displayed despite the `#pause`, you can put it after a `#meanwhile`:

```
Show this content #pause and then
this #pause and finally this!

#meanwhile
But always show this!
```

➡️

Show this content and then this

But always show this!



Animations: `#pause` and `#meanwhile`

2 Basic usage

Of course, you might want to gradually display the content of the slide!

You can use `#pause` to create a slide with only the content preceding the `#pause`.

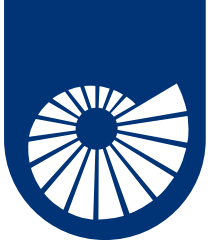
If there's some content that you want displayed despite the `#pause`, you can put it after a `#meanwhile`:

```
Show this content #pause and then
this #pause and finally this!

#meanwhile
But always show this!
```

➡️

Show this content and then this and finally this!

But always show this!



More animations: `#only` and `#uncover`

If you need more control on hiding your content, you can use `#only` and `#uncover`!

`#only` and `#uncover` only show the given content on a specified interval, with the difference being that `#uncover` reserves space for the hiddend content, while `#only` doesn't.

An interval `"n-m"` means that the content gets shown from frame `n` to frame `m`, where a missing `n` or `m` means, respectively, *"since the start"* and *"until the end"*.

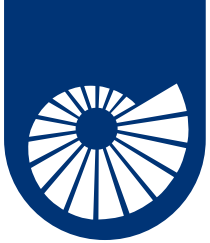
```
1… #pause 2… #pause 3… #pause Tadaa!

#meanwhile
- show #only("2-3")[#emoji.star] from 2 to 3
- uncover #uncover("3-")[#emoji.sun] from 3 to end
```

1…

➡️

- show from 2 to 3
- uncover from 3 to end

Note: depending on how you use `#only` and `#uncover`, you might get some issues with handout mode. There are some (possibly opinionated) animation guidelines if you're interested in using them.



More animations: `#only` and `#uncover`

2 Basic usage

If you need more control on hiding your content, you can use `#only` and `#uncover`!

`#only` and `#uncover` only show the given content on a specified interval, with the difference being that `#uncover` reserves space for the hiddend content, while `#only` doesn't.

An interval `"n-m"` means that the content gets shown from frame `n` to frame `m`, where a missing `n` or `m` means, respectively, *"since the start"* and *"until the end"*.

```
1… #pause 2… #pause 3… #pause Tadaa!

#meanwhile
- show #only("2-3")[#emoji.star] from 2 to 3
- uncover #uncover("3-")[#emoji.sun] from 3 to end
```

1… 2…

➡️

- show ⭐ from 2 to 3
- uncover from 3 to end

Note: depending on how you use `#only` and `#uncover`, you might get some issues with handout mode. There are some (possibly opinionated) animation guidelines if you're interested in using them.

More animations: `#only` and `#uncover`

If you need more control on hiding your content, you can use `#only` and `#uncover`!

`#only` and `#uncover` only show the given content on a specified interval, with the difference being that `#uncover` reserves space for the hiddend content, while `#only` doesn't.

An interval `"n-m"` means that the content gets shown from frame `n` to frame `m`, where a missing `n` or `m` means, respectively, *"since the start"* and *"until the end"*.

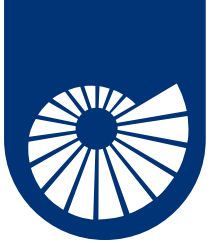
```
1… #pause 2… #pause 3… #pause Tadaa!

#meanwhile
- show #only("2-3")[#emoji.star] from 2 to 3
- uncover #uncover("3-")[#emoji.sun] from 3 to end
```

➡️

1… 2… 3…

- show ⭐ from 2 to 3
- uncover ☀️ from 3 to end

Note: depending on how you use `#only` and `#uncover`, you might get some issues with handout mode. There are some (possibly opinionated) animation guidelines if you're interested in using them.



More animations: `#only` and `#uncover`

2 Basic usage

If you need more control on hiding your content, you can use `#only` and `#uncover`!

`#only` and `#uncover` only show the given content on a specified interval, with the difference being that `#uncover` reserves space for the hiddend content, while `#only` doesn't.

An interval `"n-m"` means that the content gets shown from frame `n` to frame `m`, where a missing `n` or `m` means, respectively, *"since the start"* and *"until the end"*.

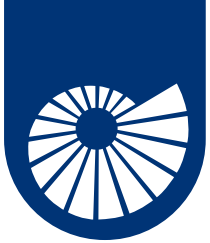
```
1… #pause 2… #pause 3… #pause Tadaa!

#meanwhile
- show #only("2-3")[#emoji.star] from 2 to 3
- uncover #uncover("3-")[#emoji.sun] from 3 to end
```

➡️

1… 2… 3… Tadaa!

- show from 2 to 3
- uncover ☀️ from 3 to end

Note: depending on how you use `#only` and `#uncover`, you might get some issues with handout mode. There are some (possibly opinionated) animation guidelines if you're interested in using them.



`#slide`, `#title-slide` and more

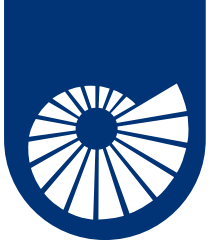
Some slides such as the title slide have their own function which needs to be called to be created. This is true for:

- `#title-slide()`, which creates the initial slide of the presentation
- `#ending-slide()`, which creates the *“Thank you for listening!”* slide
- `#focus-slide()[your content here]`, similar to an ending slide in appearence but for other type of contents

Additionally, while slides are automatically created for each subsection, you can manually declare a slide with `#slide()[slide content here]`. This is most useful if you want to override some stile only on a single slide, as you will see in the Customization section.

Table of Contents

3 Customization



Language

3 Customization

By default, some parts of these slides appear in English, such as *“Table of Contents”*, *“Theorem”* or the month in the date.

You can change this behaviour by changing the text language with a simple `#set text()` at the top of the `main.typ`. This will change the language setting for all the presentation.

```
#theorem(title: "Euclid's Theorem")[
  There are infinitely many primes.
]
```

➡️

Theorem (Euclid's Theorem)

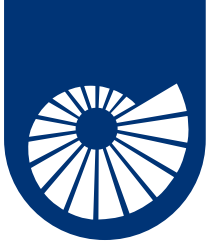
There are infinitely many primes.

```
#set text(lang: "it")
#theorem(title: "Euclid's Theorem")[
  There are infinitely many primes.
]
```

➡️

Teorema (Euclid's Theorem)

There are infinitely many primes.



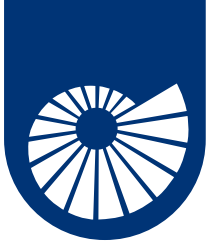
More control over slides

What?! A custom subtitle? That can't be!

Slides are automatically created for each “`== subsection`”, but you can also choose to create them manually with the `#slide()` function!

This allows you to have more control on what gets displayed on the slide, for example you can have a custom subtitle, instead of having the section title:

```
== More control over slides

#slide(subtitle: "What?! A custom subtitle? That can't be!")[
  Slides are automatically created for each...
]
```



Per-slide customizations

3 Customization

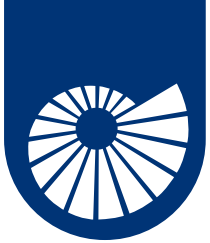
Some of the following customization happen by changing some values inside of some `config-xyz()` in the initialization of the theme.

These settings can also be changed on single slides! To do so, you must pass the config with the values you want, such as the following:

```
#slide(config: config-colors(primary: rgb("#006565")))[
  A slide with *updated primary color!*
]
```

If you need to use multiple `config-xyz()`, you can merge them using `utils.merge-dicts()`,

```
#slide(config: utils.merge-dicts(config-colors(/* … */), config-store(/* … */)))[
  A slide with many updated settings!
]
```



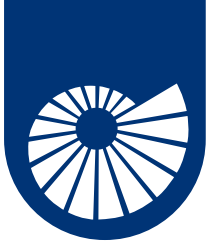
Handout mode

3 Customization

Animations like `#pause`, `#meanwhile` and `#uncover` are useful when presenting the slides, but are problematic if someone wants to print out the slides (for example to take notes on them).

To remove all the animations and only create the “final” slide for each slide, you can use the handout mode, which is activated by enabling `config-common(handout: true)`

```
#show: dmunipi-theme.with(
  // […]
  config-common(handout: true),
)
```



Headless statements

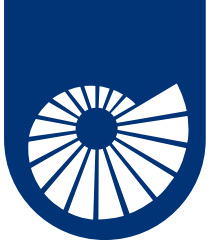
3 Customization

By default, statements such as theorems are rendered in a `#title-block` with the theorem title at the top. This might occupy too much space in some cases, and can be disabled by setting `config-store(headless-statements: true)`

```
#show: dmunipi-theme.with(
  // […]
  config-store(
    headless-statements: true,
  ),
)

#theorem(title: "Euclid's Theorem")[
  There are infinitely many primes.
]
```

➡️

Theorem (Euclid's Theorem): *There are infinitely many primes.*

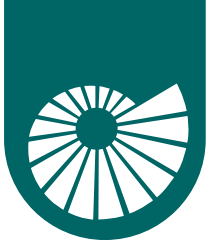


Fancy footer

3 Customization

By default, the footer of the slides only shows the slide counter. You can change the style of the footer by setting `config-store(fancy-footer: true)` to make it look like the footer in the current slide!

```
#show: dmunipi-theme.with(
  // […]
  config-store(fancy-footer: true),
)
```



Colorscheme

3 Customization

You can even change the colorscheme of the slides! The colorscheme uses:

- the `neutral-lightest` color for the background
- the `neutral-darkest` color for most of the text
- the `primary` color for most of the colored objects
- the `secondary` color for links

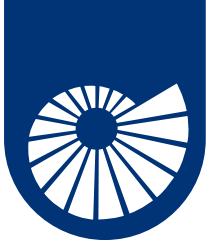
```
#show: dmunipi-theme.with(
  config-colors(
    primary: rgb("#006565"),
    secondary: rgb("#00d595"),
  ),
)
```

The Typst `dmunipi` theme

Thank you for listening!
Any question?

Table of Contents

4 Advanced usage & customization



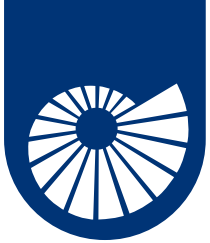
But wait… there’s more!

4 Advanced usage & customization

The part before the ending slide was the “basic” introduction of how to use Typst and this slide theme.

What you have read so far should be more than enough for your typical presentation, but there’s plenty more that can be done or customized!

The rest of this guide is dedicated to a more advanced usage of the theme. It might be useful to get comfortable with the commands `#set` and `#show`, for which you can find everything you need to start in the documentation.



Appendix

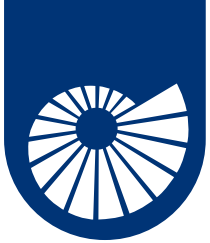
4 Advanced usage & customization

Anything after the `#ending-slide()` is countend as part of the appendix. The appendix is meant to be used as something that might be needed after the end of the presentation, but not strictly part of it (for example, some helping slides in case of questions).

The sections of the appendix will not result in the *“Table of Contents”*-s of the main part, and the slides of the appendix won’t be counted as part of the shown total.

To disable this, you can set `config-store(appendix-after-ending: false)`:

```
#show: dmunipi-theme.with(
  // […]
  config-store(appendix-after-ending: false),
)
```

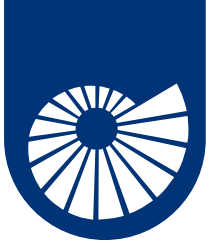


“Only-in-this-section” customizations

4 Advanced usage & customization

Any change applied through an `#config-xyz()` can also be applied to a whole block of content, instead of single slides, using the `#set-config()` command:

```
== Here a normal slide
This slide has the "normal" color

#set-config(config-colors(primary: rgb("#006565")))[
  == A slide with different colors!
  This slide has the updated color

  == Another slide with different colors!
  This slide too has the updated color!
]
```



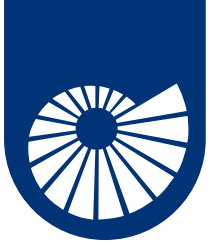
“From-here-on” customizations

4 Advanced usage & customization

Using the `#show` command, configs can also be activated *“from here on”*, meaning from the `#show` command until the end of the document, like the following example:

```
== Here a normal slide
All the slides before this one (this included) will have the usual color

#show: set-config.with(config-colors(primary: rgb("#006565")))
== A slide with different colors!
All the slides from here on will have the updated color
```

This usage of `#show` is a bit delicate, since it can be used only between slides and will throw an error otherwise.



Animation guidelines

4 Advanced usage & customization

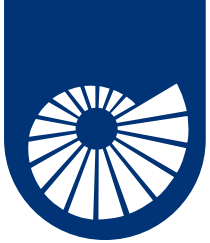
While `#pause` displays the hidden content on the last animation slide, animations like `#only` and `#uncover` (and another, see next slide) don't do so necessarily.

The issue is that in handout mode, only the last frame of each animation is shown, meaning that content that doesn't go *"until the end"* will not show up in handout mode.

I suggest to **only** use intervals of the type `"n-"` in order not to have issues with handout mode.

Similarly, with a combination of `#place` and `#pause` you can effectively end up with some background content hidden by foreground content on the last frame.

Please don't. It's possible, but don't.



Even animations: `#alternatives`

4 Advanced usage & customization

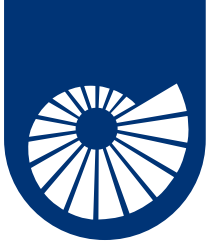
Speaking of animations that don’t behave nicely with handout mode, there is also `#alternatives`, which is be listed for sake of completeness but I don’t recommend using for the same reasons as the slide before.

`#alternatives` which switches between two or more contents depending on the frame, meaning that `#alternatives[A][B]` will render as `A` on the first frame and `B` on the second frame

You can find more informations about it in the documentation.

Table of Contents

5 Troubleshooting



Font issues

5 Troubleshooting

By default, this theme uses the `Roboto` font family.

If the font in your pdf looks like this instead of how it appears in the rest of this guide, then it's likely that you don't have the `Roboto` font installed on your system

You can either choose to download it and install it, or edit the `theme/lib.typ` file to change the default font to something installed on your system of your liking