

The **Estonia** polylux theme

Simple and customizable

Author A

Author B

Author C

Date

Table of contents

1. [Showcase](#)
2. [Usage & slide API](#)
3. [Title bar customization](#)
4. [Progress bar customization](#)
5. [Bottom bar customization](#)
6. [Layout and margins](#)

Showcase

Maths

A slide with some equations

$$\begin{aligned}\sum_{k=0}^n k &= 1 + \dots + n \\ &= \frac{n(n+1)}{2}\end{aligned}$$

more equations

$$7.32 \beta + \sum_{i=0}^{\nabla} \frac{Q_i(a_i - \varepsilon)}{2}$$

Footnotes

Footnotes¹ can be added to your Typst² slides.



Your slide content must have enough remaining vertical space. Otherwise, footnotes will not be shown.

¹<https://typst.app/docs/reference/meta/footnote/>

²Mädje, L. (2022). A Programmable Markup Language for Typesetting (Master's thesis, TU Berlin). [pdf](#)

Code blocks

You can of course show code blocks too!

```
// comment
int main()
{
    printf("Hello, world!\n");
    if (42 == 6*7 && 0+0 == 0)
        return 0;
    return 1;
}
```

Titles can contain line breaks

Such as this one

Or this one...

Title bar height should be adjusted automatically.

Usage & slide API

The slide function

This theme is essentially a `slide` function.

The slide function

This theme is essentially a `slide` function. It is called like this:

```
#slide(title: "The slide title")  
  The slide content  
]
```

The slide function

This theme is essentially a `slide` function. It is called like this:

```
#slide(title: "The slide title")[  
  The slide content  
]
```

`slide` has many parameters that enables customization. For example to hide the progress bar, show the title bar and the bottom bar with a slide number:

```
#slide(title: "Title", progress-bar: false, title-bar: true,  
  bottom-bar: true,  
  bottom-bar-args: (  
    func-args: (  
      show-slide-number: true  
    )  
  )  
)
```

The slide function: custom call

Typst makes it convenient to customize the function call for all your slides

```
#let my-slide = slide.with(/*args you want to customize*/)
```

The slide function: custom call

Typst makes it convenient to customize the function call for all your slides

```
#let my-slide = slide.with(/*args you want to customize*/)
```

You can also create your own wrapper around the function, which is very convenient to change some parameter for specific slides only. For example this wrapper that sets the font size of code blocks has been used heavily to generate the [Estonia](#) slides:

```
#let dslide(raw-font-size: 20pt, ..args) = {  
  show raw: set text(font: "Inconsolata",  
                    weight: "semibold",  
                    size: raw-font-size)  
  slide(..args)  
}
```

Title bar customization

Title bar

slide API related with title (bar):

```
#let slide(  
  title: none,  
  title-bar: auto,  
  title-bar-args: (  
    height-per-line: 1.5em,  
    func: est-title-bar,  
    func-args: (:),  
  ),  
  // (other args hidden)  
) = { /* ... */ }
```

title: none or a string

title-bar:

- auto: shown iff title is set
- true/false: always shown/hidden

inside title-bar-args dict:

- height-per-line is the height of a line. This is multiplied by the number of lines in title
- func is the function that generates the title bar (default func is presented on next slide)
- func-args are the additional arguments given to func when it is called

Title bar: default function

```
#let est-title-bar(  
  title,  
  width,  
  height,  
  bg-color: est-blue,  
  inset: 1cm,  
  alignment: left+horizon,  
  text-fill-color: white,  
  text-size: 1.2em,  
)
```

Arguments always set by `slide` at call time:

- `title`: the slide title (string)
- `width`: the page width (length)
- `height`: the title bar height (length)

The other arguments enable customization of the bar. Their names should be enough to explain what they do ;).

Title bar: customize default function call

```
#let critical-slide = slide.with(  
  title-bar: true,  
  title-bar-args: (  
    height-per-line: 3cm,  
    func: est-title-bar,  
    func-args: (  
      bg-color: red,  
      alignment: center+horizon,  
      text-fill-color: white,  
      text-size: 1.5em,  
    )  
  )  
)
```

```
#critical-slide(title: "...") [  
  /* content */  
]
```

~~~~~ TITLE BAR: YOUR OWN FUNCTION ~~~~~

```

#let my-title-bar-func(title, width, height,
                        inset: 3cm, nb-tilde: 10) = {
  show: block.with(
    width: width, height: height,
    above: 0pt, below: 0pt,
    breakable: false,
    fill: est-blue, inset: inset,
  )
  set align(horizon+center)
  set text(fill: white, size: 1.2em)
  if title == none [] else {
    for _ in range(nb-tilde) [\~]
    h(3mm) smallcaps(title) h(3mm)
    for _ in range(nb-tilde) [\~]
  }
}

#slide(title: "...",
        func: my-title-bar-func,
        func-args: (nb-tilde: 7)
)[
  /* content */
]

```

Progress bar customization

Progress bar

slide API related with progress bar:

```
#let slide(  
  progress-bar: true,  
  progress-bar-args: (  
    height: 1em,  
    func: est-progress-bar,  
    func-args: (:),  
  ),  
  // (other args hidden)  
) = { /* ... */ }
```

progress-bar:

- true/false: always shown/hidden

inside progress-bar-args dict:

- height is the height of the progress bar
- func is the function that generates the progress bar (default func is presented on next slide)
- func-args are the additional arguments given to func when it is called

Progress bar: default function

```
#let est-progress-bar(
  width, height, inset: 1cm,
  show-slide-number: true,
  bg-color: black, text-size: 0.8em,
  alignment: horizon,
  current-slide: loc =>
    {circle(radius: .15em, fill: white)},
  before-slide: loc =>
    {circle(radius: .075em,
             fill: white.darken(50%))},
  after-slide: loc =>
    {circle(radius: .075em,
             fill: white.darken(50%))},
  slide-number: loc =>
    {block(width: 1em, align(right, text(
      fill:white,
      logic.logical-slide.display()))))}
)
```

Arguments always set by `slide` at call time:

- `width`: the page width (`length`)
- `height`: the progress bar height (`length`)

The other arguments enable customization of the bar.

- `before-slide`, `current-slide` and `after-slide` are functions that generate contents inside the bar. By default they generate disks of various size/color.
- `slide-number` is a function that generate contents inside the bar. By default it simply writes the current slide number.

Progress bar: customize default function call

```
#let pacman-progress-slide = slide.with(progress-bar-args: (
  func: est-progress-bar,
  func-args: (
    show-slide-number: false,
    inset: 5mm,
    before-slide: loc => {circle(radius: .075em, fill: white.darken(60%))},
    after-slide: loc => {circle(radius: .075em, fill: white)},
    current-slide: loc => { text(size: 12pt, fill: yellow, weight: "bold")[<] }
  )
))

#pacman-progress-slide(title: "...")[
  /* content */
]
```

Progress bar: your own function

```
#let my-progress-bar(width, height) = {
  locate(loc => {
    let current_slide = logic.logical-slide.at(loc).first() - 1
    let nb_slides = logic.logical-slide.final(loc).first()
    let ratio = current_slide / nb_slides
    set align(left+horizon)
    grid(columns: (width * ratio, width * (1 - ratio)), gutter: 0mm,
      est-cell(fill: white, height: height, text(fill: est-blue, size: .8em,
        [#current_slide / #nb_slides])),
      est-cell(fill: black, height: height)
    )
  })
}
#slide(title: "...", progress-bar-args: (func: my-progress-bar))[
  /* content */
]
```

Bottom bar customization

Bottom bar


slide API related with bottom bar:

```
#let slide(  
  bottom-bar: false,  
  bottom-bar-args: (  
    height: 6mm,  
    func: est-speaker-slide-reminder-bar,  
    func-args: (:)  
  ),  
  // (other args hidden)  
) = { /* ... */ }
```

bottom-bar:

- true/false: always shown/hidden

inside title-bar-args dict:

- height is the height of the bottom bar.
 using em units here can be tedious
- func is the function that generates the title bar (default func is presented on next slide)
- func-args are the additional arguments given to func when it is called

Bottom bar: default function

```
#let est-speaker-slide-reminder-bar(  
  width, height,  
  show-slide-number: false,  
  text-size: .8em,  
  alignment: center+horizon,  
  author-text-fill-color: white,  
  author-bg-color: black,  
  presentation-title-text-fill-color: white,  
  presentation-title-bg-color: est-blue,  
  slide-number-text-fill-color: black,  
  slide-number-bg-color: white,  
  author: [Speaker],  
  presentation-title: [Presentation title],  
  slide-number:  
    {logic.logical-slide.display()}  
) = { /*...*/ }
```

Arguments always set by `slide` at call time:

- `width`: the page width (`length`)
- `height`: the bottom bar height (`length`)

The other arguments enable customization of the bar. Their names should be enough to explain what they do ;).

Bottom bar: customize default function call

```
#let struts-slide = slide.with(  
  bottom-bar: true,  
  bottom-bar-args: (  
    func: est-speaker-slide-reminder-bar,  
    func-args: (  
      author: [Jane Doe],  
      presentation-title: [Estonia],  
      show-slide-number: true,  
    )  
  )  
)  
  
#struts-slide(title: "...") [  
  /* content */  
]
```

Bottom bar: your own function

```
#let my-bottom-bar(width, height, font-size: .8em) = {
  block(width: width, height: height, breakable:false, inset: 3mm, {
    set align(horizon+right)
    set text(size: font-size)
    locate(loc => {
      let current_slide = logic.logical-slide.at(loc).first() - 1
      let nb_slides = logic.logical-slide.final(loc).first()
      [ #current_slide / #nb_slides ]
    })
  })
}

#slide(title: "...", bottom-bar: true, bottom-bar-args: (func: my-bottom-bar))[
  /* content */
]
```

Layout and margins

The layout of this theme is the one presented on this slide.
Bars can be disabled but cannot be placed at a different position for now.

The green rectangle is your content bounding box.
Your content is by default surrounded by a 1cm border on all sides.

From the Typst point of view,
everything around your content bounding box are page margins.

If you generate a slide without any bar, your content still has borders by default.

```
#slide(progress-bar: false, title-bar: false, bottom-bar: false)[  
  /* content */  
]
```

slide body & margin customization

Here are the remaining slide arguments:

```
#let slide(  
  body-args: (  
    margin: 1cm,  
    alignment: left+horizon,  
    text-fill-color: black,  
    bg-color: white),  
  body  
) = { /* ... */ }
```

margin:

- if (length): all body margins set to the given value
- if (dict): specify the body margin value for each margin
 - top, bottom, left, or right
 - rest is used if a specific value is not set

Disable all margins

```
#slide(body-args: (margin: 0cm))[  
  /* content */  
]
```

Disable a single margin

```
#slide(body-args: (margin: (top: 0cm,  
                             rest: 1cm))  
)[  
  /* content */  
]
```

Customize all margins

```
#slide(title: "...",  
  bottom-bar: true,  
  body-args: (margin: (top: 0cm,  
    bottom: 0cm,  
    left: 1cm,  
    right: 2cm)))  
)[  
  /* content */  
]
```

That's all folks!