

Example and Documentation of the HaoBook Template

by ParaN3xus

2025-04-06

By epigraph, you can add a quote or a saying at the beginning of the book.

Preface

This is a preface. You can add some information about the book here by calling `preface`.

The template is heavily inspired by the LaTeX book template [kaobook](#), and was sponsored by [@yexiang1992](#). Without the developers of kaobook and the support of [@yexiang1992](#), this template would not have been possible.

During the development of this template, I have also referenced the implementations of [project-iii-typst](#) and [tufte-memo](#). They are all excellent templates that you might want to explore if interested. The key feature of this template—sidenotes—is implemented using [marginalia](#) by [@nleanba](#). I must emphasize that [marginalia](#)'s implementation is truly outstanding. If you simply want to add sidenotes to your document, I recommend using [marginalia](#) directly, as it offers more power and customization options than this template.

This document serves both as a test document and a tutorial for the template. You can find the source code in the `example.typ` file. The template is designed to be user-friendly and customizable, allowing you to adapt it to your specific requirements.

ParaN3xus

Contents

Preface	3
Contents	4
SPECIFICATIONS	5
1 Features	6
1.1 Front Matter	6
1.2 Margin-related Features	6
1.2.1. Automatically	6
1.2.2. Manually	6
1.3 General Styles	7
2 Usage of the Template	8
2.1 Importing the Template	8
2.2 Suggested Document Structure	9
2.3 Usage of Some Functions	10
2.3.1. Side Note	10
2.3.2. Side Figure	10
2.3.3. Wide Block	10
2.3.4. Image Heading	11
2.3.5. Bibliography	11
2.3.6. Normal Page	11
3 Compatibility	12
3.1 Margin Notes Packages	12
3.2 i-figured	12
3.3 Theorems	12
3.3.1. theorion	12
3.3.2. ctheorems	12
3.3.3. thmbox	13
3.3.4. lemmify	13
3.3.5. frame-it	13
3.3.6. great-theorems	13
APPENDIX	14
A Bibliography	15
B Notes	16

SPECIFICATIONS

Features

1

In this chapter, I will show you all the features of the template.

1.1 Front Matter

As you can see, the template provides all those front matter pages before this page. You can use them to add a cover page, an epigraph, a preface, and a table of contents to your document.

1.2 Margin-related Features

1.2.1. Automatically

1.2.1.1. Margin position

You may want to publish your book in a way that the margin notes are on the left side of the page for odd pages and on the right side for even pages. This template provides you with this feature. You can enable it by using the book: `true` parameter when calling the `template` function.

1.2.1.2. Chapter outline

You may have noticed that the chapter outline is automatically generated.

1.2.1.3. References

When I'm referencing something great^[1], a detailed reference will be shown in the margin.

1.1 Front Matter 6

1.2 Margin-related Features . 6

1.3 General Styles 7

1.2.1.4. Figures

When I'm adding a figure, the caption will be shown in the margin.



[1] L. Mädje, “Typst: A Programmable Markup Language for Typesetting,” Berlin, Germany, 2022.

Figure 1-1: A rainbow

1.2.2. Manually

1.2.2.1. Side Notes

You can use `side-note` and `margin-note` to add notes to the document. Just like what¹ was shown on the right side.

A margin note.

1: A side note.

1.2.2.2. Side Figures

You can use `side-figure` to add a figure like Figure 1-2 to the document.

You might wonder why you can't simply use `figure` inside `margin-note`. This would cause conflicts with the figure caption styles shown in Section 1.2.1.4., so you need to use `side-figure` instead.



Figure 1-2: A circle

1.2.2.3. Wide Block

You can use `wideblock` to add a block that spans the whole page. This is useful for adding text or figures that need to extend across the entire page width.

Figure 1-3: A wide rainbow



1.3 General Styles

You've already seen the general styles of the template, including heading styles, page headers, etc.

You may have also noticed that figure numbering¹ follows the format `chapno-figno`. This is already implemented in the template, so there's no need to use `i-figure`.

1: Actually the counters affected are:

- `figure.where(kind: image)`
- `figure.where(kind: table)`
- `figure.where(kind: raw)`
- `math.equation`.



2 Usage of the Template

The template is designed to be easy to use. You can use it to create a book or a report with a beautiful layout.

In this chapter, I will show you how to initialize the template and use a variety of utility functions provided.

2.1 Importing the Template

To use the template, you need to import the `lib.typ` file and call the `template` function.

```
1 #import "haobook/lib.typ": *
2
3 // Pass book: true parameter to use book mode
4 #let (body-styles, normal-page, img-heading, side-
figure) = template(book: true)
```

typ

When calling the `template` function, you can pass a parameter `book` to specify whether you are creating a book with alternating margin notes position or a report with consistent margin position. The default value is `false`.

In addition to the four functions returned by `template`, functions independent of book mode are also imported. All of these functions are:

- Styles:
 - `front-matter-style(body)`: Style for front matter pages.
 - `appendix-style(body)`: Style for appendix pages.
 - `body-styles(body)`: Style for body pages.
- Pages¹:
 - `cover(title: "The Title", author: "The Author", date: datetime.today()):` Add a cover page to the document.
 - `epigraph(body)`: Add an epigraph to the document.
 - `preface(body)`: Add a preface to the document.
 - `contents`: Add a table of contents to the document.
 - `bib(body)`: Add a bibliography to the document.
 - `part(body)`: Add a part page to the document.

2.1 Importing the Template .	8
2.2 Suggested Document Structure	9
2.3 Usage of Some Functions	10

1: Some of them like `contents` and `normal-page` are not necessarily limited to one page.

- ▶ `normal-page(body)`: Add normal pages without side notes to the document.
- Tools:
 - ▶ `side-note(body, dy: 0em)`: Add a side note¹ to the document.
 - ▶ `margin-note(body, dy: 0em)`: Add a margin note to the document.
 - ▶ `side-figure(figure, label: none, dy: 0em)`: Add a side figure to the document.
 - ▶ `wideblock(body)`: Add a wide block to the document.
 - ▶ `img-heading(body, img, label: none)`: Add a heading of level 1 with a banner image to the document.

1: `side-note` has numbering.
`margin-note` doesn't have numbering.

2.2 Suggested Document Structure

Overall, your document should be structured like this:

```
typ
1 #import "haobook/lib.typ": *
2 #let (body-styles, normal-page, img-heading, side-
  figure) = template(book: true)
3
4 // frontmatter
5 #cover(
6   // title: "The Title",
7   // author: "The Author",
8   // date: datetime.today(),
9 )
10 #epigraph[
11   // Add an epigraph to the document.
12 ]
13 #preface[
14   // Add a preface to the document.
15 ]
16 #contents
17
18 // body
19 #show: body-styles
20
21 ...
22
23 // appendix
24 #part[Appendix]
25 #show: appendix-style
26
27 ...
```

For the body of the document, the template provides you with at least three hierarchical levels of structure:

- `part2`: The part will create a single page with a title. You can use it to create a new chapter or a new part in your document.
- `= Heading` or `img-heading`: The level 1 heading will create a chapter starting from a new page³. Specifically, the `img-heading` will create a heading with an image banner.
- `== Section`: The level 2 heading will create a section.
- `...:` You can use more `=` to create deeper levels of headings. The template won't style them for you.

2: A part is optional. You can create a level 1 heading without any part just like Section 2.

3: With a chapter outline just like the one at the beginning of this chapter

For the appendix, it's almost the same as the body. But their heading numbering is "A.1".

2.3 Usage of Some Functions

In this chapter, I will show you how to use some functions, mainly tool functions, as well as some page functions.

2.3.1. Side Note

Two types of notes are provided in the template:

- `side-note`: The `side-note` will create a side note with numbering¹. Its numbering will be reset for each page.
- `margin-note`: The `margin-note` will create a margin note without numbering. And thus, there won't be any mark in the main text.

1: Like this

Like this

For both types of notes, you can use the `dy` parameter to specify the vertical position of the note. The default value is `0em`, which means the note will be placed at the same vertical position as the main text. You can use a negative value to move the note up, or a positive value to move it down.

2.3.2. Side Figure

The `side-figure` will create a side figure with a caption. You can use it to create a figure with a caption on the side of the page. The caption will be numbered according to the chapter.

Here is an example of how to use the `side-figure`:

```
typ
1 #side-figure(
2   figure(
3     rect(),
4     caption: "A rectangle",
5   ),
6   label: <fig-rect>,
7 )
```

As you can see, the `side-figure` will create a figure with a caption on the side of the page, just like Figure 2-1.

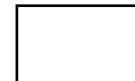


Figure 2-1: A rectangle

You can also use the `dy` parameter to specify the vertical position of the figure.

2.3.3. Wide Block

Sometimes, you may want to create a wide block of content that spans the whole page. You can use the `wideblock` function to do this:

```
typ
1 #wideblock(
2   figure(
3     rect(
4       width: 100%,
5       height: 1cm,
6       fill: gradient.linear(..color.map.rainbow),
7     ),
8     caption: [Another wide rainbow],
9   ),
10 )
```



Figure 2-2: Another wide rainbow

2.3.4. Image Heading

The `img-heading` function will create a level 1 heading with an image banner.

Actually, it's more like an example of alternative styles for level 1 headings. If you don't like it, you can modify it to your own style.

```
typ
1 #img-heading(
2   [Usage of the Template],
3   read("assets/banner.png", encoding: none),
4   label: <chapter-usage>,
5 )
```

The effect is shown in Section 2.

Note that you should use the `label` parameter to specify the label of the heading, and you have to use the `read` function to read the image file, otherwise the root of the path will be inside the template folder¹.

1: Which is not what we want

2.3.5. Bibliography

```
typ
1 #bib(read("bib.bib"))
```

The `bib` function will create a bibliography page with the references in the `bib.bib` file.

Note that you should use the `read` function to read the `bib.bib` file, for the same reason as explained for `img-heading`.

2.3.6. Normal Page

The `normal-page` function² will create a normal page without any side notes. You can use it to create a page with only the main text.

2: It's just a wrapper of `set_page(...)`

Actually it creates normal "pages" since if you pass too much content to it, it will create multiple pages.

Compatibility

3

3.1 Margin Notes Packages

The template is based on the `marginalia` package, so it should be compatible with the `marginalia` package.

Regarding `marge`, `drafting`, or other margin-related packages, I'm not certain about their compatibility. However, it's generally not recommended to use them together with the `marginalia` package.

3.2 i-figured

Don't use `i-figured` with this template. It will cause conflicts with the numbering styles.

If you want to customize the numbering styles, you can examine `chapter-fig-eq-no` and its usage inside `body-styles` in `haobook/styles.typ`.

3.3 Theorems

3.3.1. theorion

After a simple test¹, I believe the `theorion` package is compatible with this template.

1: A simple “define and reference” test.

It's an excellent package for creating theorems and proofs. You can use it to create theorems, lemmas, corollaries, propositions, definitions, examples, remarks, and proofs.

It's recommended to use `#set-inherited-levels(1)` to make its numbering consistent with the template.

3.3.2. ctheorems

After a simple test, I believe the `ctheorems` package is compatible with this template.

It's recommended to set `base_level: 1` when defining thmboxes to make its numbering consistent with the template. Also, to ensure the style of `ref2` to theorems is consistent with the template, you are recommended to use the following show rule:

typ

```
1 #show ref: it => {
2   if it.element == none or it.element.func() != figure
3     or it.element.kind != "thmenv" {
4     return it
5   }
6   show link: set text(black)
7   show underline: x => x.body
```

2: The default style of `ctheorems` `ref` is affected by the `show` rule of `link`, so it will be blue and underlined without action.

```
8   it
9 }
```

3.3.3. thmbox

After a simple test, I believe the [thmbox](#) package is compatible with this template.

It's recommended to set fonts for thmboxes to make them consistent with the template:

```
typ
1 #import "@preview/thmbox:0.2.0": *
2
3 #let (thmbox, proof, theorem, proposition, lemma,
       corollary, definition, example, remark, note, exercise,
       algorithm, claim, axiom) = (thmbox, proof, theorem,
       proposition, lemma, corollary, definition, example,
       remark, note, exercise, algorithm, claim, axiom).map(x
=> x.with(
4   title-fonts: "Libertinus Serif",
5   sans-fonts: "Libertinus Serif",
6 ))
7 #show: thmbox-init()
```

3.3.4. lemmify

It works fine, but due to its own bug¹, the numbering is not consistent with the template.

3.3.5. frame-it

It's completely² incompatible with this template. It causes conflicts with the figure caption styles.

3.3.6. great-theorems

It's completely incompatible with this template. It causes conflicts with the heading numbering³ styles.

1: I'm not entirely sure, but once I've set some heading numbering, the `max-reset-level` parameter stopped working.

2: The theorem body simply disappears.

3: Actually the conflicting part is inside `rich-counters`, so that may be another incompatible package.

APPENDIX

Bibliography | A

- [1] L. Mädje, “Typst: A Programmable Markup Language for Typesetting,” Berlin, Germany, 2022.

Notes

B

I hate writing notes.