Minimal Books

Manual

Simple and complete books without introducing new syntax.

---

min-book	1.0.0	MIT

Maycon F. Melo*

QUICK START

```
#import "@preview/min-book:1.0.0": book
#show: book.with(
  title: "Book Title",
  subtitle: "Complementary subtitle, not more than two lines long",
  authors: "Author",
)
```

DESCRIPTION

Generate complete and complex books, without any annoying new commands or syntax, just good old pure Typst. This package works by manipulating the standard Typst elements, adapting them to the needs of a book structure. All of this is managed behind the scenes, so that nothing changes in the Typst code itself.

The commands that `min-book` provides are only included for the sake of completeness, and offers some fancy optional features like horizontal rules or end notes; but it's perfectly possible to write entire books without them.

While it is possible to play with complex structures, such as parts and chapters and creative numbering, this package comes with several ready to use default values; so its really up to you to customize it your way or ride along the defaults and just start writing: both ways are possible and encouraged.

This manual will be updated only when new versions break or modify something; otherwise, it will be valid to all newer versions starting by the one documented here.

---

*https://github.com/mayconfmelo

OPTIONS

Those are the full list of options available and its default values:

```
#import "@preview/min-book:1.0.0": book
#book(
  title: none,
  subtitle: none,
  edition: 0,
  volume: 0,
  authors: none,
  date: datetime.today(),
  cover: auto,
  titlepage: auto,
  catalog: none,
  errata: none,
  dedication: none,
  acknowledgements: none,
  epigraph: none,
  toc: true,
  part: auto,
  chapter: auto,
  cfg: auto,
  body
)
```

Seems like an awful lot to start with, but let's just break down all this to understand it better, shall we?

`title:` `string` `content` (*required*)

The main title of the book.

`subtitle:` `string` `content` `none`

The book subtitle; generally two lines long or less.

`edition:` `int`

The book publication number; used when the content is cjanged or updated after the book release.

`volume:` `int`

The book series volume number; used when the same story is told through multiple books, in order.

`authors:` string array (*required*)

A string or array of strings containing the names of each author of the book.

`date:` array dictionary

The book publication date; can be a `(yyyy, mm, dd)` array or a `(year: yyyy, month: mm, day: dd)` dictionary, and defaults to current date if not set.

`cover:` auto content none

The book cover content; when `auto`, generates an automatic cover.

`titlepage:` auto content

The presentation page content, shown after the cover; when `auto`, generates an automatic title page.

`catalog:` toml yaml dictionary

Set the data for the “cataloging in publication” board.

`errata:` content

A text that corrects important errors from previous book editions.

`dedication:` content

A brief text that dedicates the book in honor or in memorian of someone important; can accompany a small message directed to the person.

`acknowledgements:` content

A brief text to recognize everyone who helped directly or indirectly in the process of book creation and their importance in the project.

`epigraph:` quote content

A short citation or excerpt of other works used to introduce the main theme of the book; can suggest a reflection, a mood, or idea related to the text.

`toc:` boolean

Defines if the book will have a table of contents.

`part:` auto string none

The name of the main divisions of the book; when `auto`, try to retrieve the translation for “Part” in `#text.lang` from `#book(lang-data)`, or fallback to English.

`chapter:` string none

The name of the sections of the book; when `auto`, try to retrieve the translation for “Chapter” in `#text.lang` from `#book(lang-data)`, or fallback to English.

`cfg:` dictionary

Set custom advanced configurations; used to fine-tune some aspects of the book, mostly aesthetic formatting. If your focus is to just write a book in English without worry about code, just ignore it; otherwise, refer to ADVANCED CONFIGURATIONS to check the supported configurations.

`body` content

The book content.

ADVANCED CONFIGURATIONS

```
#import "@preview/min-book:1.0.0": cfg
#cfg(
  numbering-style: auto,
  page-cfg: "a5",
  lang: "en",
  lang-data: toml("assets/lang.toml"),
  justify: true,
  line-space: 0.5em,
  par-margin: 0.65em,
  first-line-indent: 1em,
  margin: (x: 15%, y: 14%),
  font: ("Book Antiqua", "Times New Roman"),
  font-math: "Asana Math",
  font-mono: "Inconsolata",
  font-size: 11pt,
  heading-weight: auto,
)
```

These configurations allows to modify certain aspects of the book and better control its appearence. They are built with rebust defaults in mind, so that a casual writer can safely ignore it and *just write* , and in most cases it will be used for simple tasks such as change the language or fonts.

`numbering-style:` array string none

Defines a custom heading numbering; can be a standard numbering string, or a *numbly*[1] numbering array.

`page-cfg:` dictionary string

Set page configuration, acting as `#set page(..page-cfg)`; when string, act as `#set page(paper: page-cfg)`.

`lang:` string

Defines the language of the written text (`text.lang`).

`lang-data:` toml

A TOML translation file; the file structure can be found in the default `src/assets/lang.toml` file.

`justify:` boolean

Defines if the text will have justified alignment.

`line-space:` length

Defines the space between lines in the document.

`par-margin:` length

Defines the margin space after each paragraph. Set it the same as `#book(line-space)` to get paragraphs without additional space between them.

`first-line-indent:` length

Defines the first line indentation of all paragraphs but the first one, in a sequence of paragraphs.

`margin:` length

Defines the document margins.

`font:` string array

Defines the font families used for the text; the default font is specified in `README.md` file.

[1] https://typst.app/universe/package/numbly

`font-math:` string array

Defines the font families used for the math and equations; the default font is specified in `README.md` file.

`font-mono:` string array

Defines the font families used for the monospaced text; the default font is specified in `README.md` file.

`font-size:` length

Defines the size of the text in the document.

`heading-weight:` string

Defines the font weight of headings; by default, headings level 1–5 are `"regular"` and levels above it are `"bold"`, but `#book(cfg.heading-weight)` apply the same weight for all headings.

ADVANCED NUMBERING

The book headings can be numbered two ways: using a standard numbering string, or a *numbly*[2] numbering array. While numbering strings are indicated for simpler cases, the numbly arrays are used in more complex book numbering.

By default, when `#book(part)` is enabled, the following numbering is used:

```
(
  "{1:I}:\n",
  "{2:I}.\n",
  "{2:I}.{3:1}.\n",
  "{2:I}.{3:1}.{4:1}.\n",
  "{2:I}.{3:1}.{4:1}.{5:1}.\n",
  "{2:I}.{3:1}.{4:1}.{5:1}.{6:a}. ",
)
```

But when `#book(part: none)`, the following numbering is used:

```
(
  "{1:I}.\n",
  "{1:I}.{2:1}.\n",
  "{1:I}.{2:1}.{3:1}.\n",
  "{1:I}.{2:1}.{3:1}.{4:1}.\n",
  "{1:I}.{2:1}.{3:1}.{4:1}.{5:1}.\n",
```

[2] https://typst.app/universe/package/numbly

```
  "{1:I}.{2:1}.{3:1}.{4:1}.{5:1}.{6:a}. ",
)
```

Additionally, a custom `#book(numbering-style)` can be set to override the default numbering behavior above.

BOOK PARTS

Some larger books are internally divided into multiple *parts*. This structure allows to better organize and understand a text with multiple sequential plots, or tales, or time jumps, or anything that internally differentiate parts of the story. While the name used here is *part*, different books uses different names for it: parts, subjects, books, acts, units, modules, etc.

By default, every *level 1 heading* is a part named as *"Part"* in `text.lang` document language; setting `#book(part)` change this name, or disable parts if set to `none`.

```
#show: book.with(
  part: "Act",
)
= This heading is the Act 1 part
```

BOOK CHAPTERS

In most cases, books are divided into smaller sections called chapters. Generally, each chapter contains a single minor story, or event, or scene, or any type of subtle plot change.

Chapters are smart: by default, every *level 2 heading* is a chapter named *"Chapter"* in `text.lang` document language, when parts are enabled; but when parts are disabled, every *level 1 heading* will be a chapter. Setting `#book(chapter)` change the default chapter name, or disable chapters if set to `none`.

```
#show: book.with(
  chapter: "Scene",
)
== This is Scene 1 chapter
```

```
#show: book.with(
  part: none,
  chapter: "Scene",
)
= This is Scene 1 chapter
```

CATALOGING IN PUBLICATION

```
#import "@preview/min-book:1.0.0": catalog
#catalog(
  id: none,
  place: none,
```

```
    publisher: none,
    isbn: none,
    subjects: (),
    access: (),
    ddc: none,
    udc: none,
    before: none,
    after: none,
  )
```

`id:` string content

A *Cutter-Sanborn identification code*[3], used to identify the book author.

`place:` string content

The city or region where the book was published.

`publisher:` string content

The organization or person responsible for releasing the book.

`isbn:` string content

The *International Standard Book Number*, used to indentify the book.

`subjects:` array

A list of subjects related to the book; must be an array of strings.

`access:` array

A list of access points used to find the book in catalogues, like by `"Title"` or `"Series"`; must be an array of strings.

`ddc:` string content

A *Dewey Decimal Classification*[4] number, which corresponds to the specific category of the book.

`udc:` string content

An *Universal Decimal Classification*[5] number, which corresponds to the specific category if the book.

---

[3] http://www.cutternumber.com/
[4] https://www.oclc.org/content/dam/oclc/dewey/ddc23-summaries.pdf
[5] https://udcsummary.info/php/index.php

`before:` content

Content showed before (above) the cataloging in public ation board; generally editorial data like publisher, editors, reviewers, copyrights, etc.

`after:` content

Content showed after (below) the cataloging in publication board; generally additional information that complements the board data.

ADDITIONAL COMMANDS

As said before, this package does not requires any new syntax nor commands to write a complete book. But, well… it does offers some additional commands. These are completely optional and exists only to facilitate the book writing process; it is perfectly possible to write an entire book without them.

NOTE COMMAND

```
#import "@preview/min-book:1.0.0": note
#note(
  numbering-style: auto,
  content,
)
```

Adds end notes to the book. End notes are more common than footnotes in books, and while footnotes appear at the footer of the current page, end notes appears at its own page at the end of the current section, right before the next heading.

`numbering-style:` auto array string

Defines a custom note numbering from this note onwards; can be a standard numbering string, or a *numbly*[6] numbering array.

`content` content (*required*)

The content of the end note.

APPENDICES AND ANNEXES COMMAND

```
#import "@preview/min-book:1.0.0": appendices
#appendices(
  type: "appendix",
  title: auto,
  numbering-style: (
    "",
```

[6] https://typst.app/universe/package/numbly

```
    "{2:A}.\n",
    "{2:A}.{3:1}. ",
    "{2:A}.{3:1}.{4:1}. ",
    "{2:A}.{3:1}.{4:1}.{5:1}. ",
    "{2:A}.{3:1}.{4:1}.{5:1}.{6:a}. ",
  ),
  body
)
```

Creates an special ambient to write or include multiple appendices. An appendix is any important additional data left out of the main document for some reason, but directly referenced or needed by it.

`title:` auto array

An array of strings with singular and plural titles for appendices, respectively; when `auto`, try to retrieve the translations for “Appendix” and “Appendices” in `#text.lang` from `#book(lang-data)`, or fallback to English.

`numbering-style:` array string

Defines a custom heading numbering for appendices; can be a standard numbering string, or a *numbly*[7] numbering array.

`body` content

The appendices content; every *level 1 heading* will be treated as a new appendix.

ANNEXES COMMAND

```
#import "@preview/min-book:1.0.0": annexes
#annexes(
  type: "annex",
  title: auto,
  numbering-style: (
    "",
    "{2:A}.\n",
    "{2:A}.{3:1}. ",
    "{2:A}.{3:1}.{4:1}. ",
    "{2:A}.{3:1}.{4:1}.{5:1}. ",
    "{2:A}.{3:1}.{4:1}.{5:1}.{6:a}. ",
  ),
  body
)
```

[7] https://typst.app/universe/package/numbly

Creates an special ambient to write or include multiple annexes. An annex is any important third-party data directly cited or referenced in the main document.

`title:` auto array

An array of strings with singular and plural titles for annexes, respectively; when `auto`, try to retrieve the translations for “Annex” and “Annexes” in `#text.lang` from `#book(lang-data)`, or fallback to English.

`numbering-style:` array string

Defines a custom heading numbering for annexes; can be a standard numbering string, or a *numbly*[8] numbering array.

`body` content

The annexes content; every *level 1 heading* will be treated as a new annex.

HORIZONTAL RULE COMMAND

```
#import "@preview/min-book:1.0.0": horizontalrule
#horizontalrule(
  symbol: [#sym.ast.op #sym.ast.op #sym.ast.op],
  spacing: 1em,
  line-size: 15%,
)
```

Adds a horizontal rule, visual separators used to distinguish subtle changes of subject in extensive texts.

`symbol:` content

Defines the content of a decoration in the middle of tue horizontal rule; defaults to three asterisks.

`spacing:` length

The vertical space before and after the horizontal rule.

The same `#horizontalrule` command is also as the smaller `#hr` alias.

BLOCK QUOTE COMMAND

```
#import "@preview/min-book:1.0.0": blockquote
#blockquote(by: none, ..args)
```

[8] https://typst.app/universe/package/numbly

Simple alias to add a `#quote(block:true)` command with smaller and semantic `#quote(attribution)` option, as `#blockquote(by)`.

COPYRIGHT

The logo was obtained from Flaticon website.