Minimal Books
Manual

Simple and complete books without introducing new syntax.

min-book	1.1.0	MIT

Maycon F. Melo[*]

QUICK START

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

DESCRIPTION

Generate complete and complex books, without any annoying new commands or syntax, just good old pure Typst. This package manipulates the standard Typst elements as much as possible, adapting them to the needs of a book structure in a way that there's no need to learn a whole new semantic just because of *min-book*.

For some fancy book features there is no existing compatible Typst element to re-work and adapt; in those cases, this package do provide additional commands that are completely optional, for the sake of completeness.

This package comes with some thoughful ready-to-use defaults but also allows you to play with highly customizable options if you need them, so it's really up to you: customize it your way or ride along the defaults — both ways are possible and encouraged.

---

[*]https://github.com/mayconfmelo

OPTIONS

These are all the options and its defaults used by *min-book*:

```
#import "@preview/min-book:1.1.0": book
#show: book.with(
  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
)
```

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

   Main title.

subtitle: `string` `content` `none`

   Subtitle, generally two lines long or less.

edition: `int`

   Publication number, used when the content is changed or updated in a new release after the original publication.

volume: `int`

   Series volume number, used when one extensive story is told through multiple books, in order.

authors: `string` `array`   (*required*)

   Author (`string`) or authors (`array` of `strings`).

`date:`   `datetime` `array` `dictionary`

Publication date — an array or dictionary (`year, monty, day`) or `datetime`.

`cover:`   `auto` `content` `none`

Cover — generated automatically (`auto`) or manually (`content`), or no cover (`none`).

`titlepage:`   `auto` `content` `none`

Title page, shown after cover — generated automatically (`auto`) or manually (`content`).

`catalog:`   `dictionary` `toml` `yaml`

Cataloging in publication board, with library data — see CATALOGING IN PUB-LICATION.

`errata:`   `content` `string`

A text that corrects errors from previous book editions.

`dedication:`   `content` `string`

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` `string`

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`

Generate table of contents.

`part:`   `auto` `string` `none`

Name of the main divisions of the book — set manually (`string`) or defaults to the word "Part" in current book language (`auto`).

chapter: string none

The name of the sections of the book — set manually (`string`) or defaults to the word "Chapter" in current language (`auto`).

cfg: dictionary

Custom advanced configurations, used to fine-tune some aspects of the book — see ADVANCED CONFIGURATIONS.

body content

The book content.

ADVANCED CONFIGURATIONS

```
cfg: (
  numbering-style: auto,
  page: "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,
  cover-bgcolor: rgb("#3E210B"),
  cover-txtcolor: luma(200),
  cover-fonts: ("Cinzel", "Alice"),
  cover-back: true,
  toc-indent: auto,
  toc-bold: true,
  chapter-numrestart: false,
  two-sided: true,
)
```

These `#book(cfg)` configurations allows to modify certain aspects of the book and manage its appearence and structure. Built with some thoughful ready-to-use defaults that make its use optional, so that beginners and casual writers can safely ignore it and *just write*.

`numbering-style:`   `array` `string` `none`

Custom heading numbering — a standard numbering (`string`) or a *numbly*[1] numbering (`array`).

`page:`   `dictionary` `string`

Page configuration — act as `#set page(..cfg.page)` when `dictionary` or as `#set page(paper: cfg.page)` when `string`.

`lang:`   `string`

Book language.

`lang-data:`   `toml`

Translation file — see the `src/assets/lang.toml` file.

`justify:`   `boolean`

Text justification.

`line-space:`   `length`

Space between each line in a paragraph.

`par-margin:`   `length`

Space after each paragraph.

`first-line-indent:`   `length`

indentation of the first line of each paragraph in a sequence, except the first one.

`margin:`   `length`

Page margin.

`font:`   `string` `array`

Text font family, and fallback options (`array`).

`font-math:`   `string` `array`

Math font family, and fallback options (`array`) — see the `README.md` file to download the default font, if needed.

---

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

`font-mono:` `string` `array`

Monospaced font family, and fallback options (`array`) — see the `README.md` file to download the default font, if needed.

`font-size:` `length`

Text font size.

`heading-weight:` `string` `auto`

Heading font weight (thickness) — set as `"regular"` or `"bold"`, defaults to regular in levels 1–5 then bold (`auto`).

`cover-bgcolor:` `color`

Cover background color when `#book(cover: auto)`.

`cover-txtcolor:` `color`

Cover text color when `#book(cover: auto)`.

`cover-fonts:` `array`

Cover font families when `#book(cover: auto)` — an array (`TITLE, TEXTS`) for title and text fonts, respectively.

`cover-back:` `boolean`

Generate a back cover at the end of the document when `#book(cover: auto)`

`toc-indent:` `length` `auto`

Indentation of each table of contents entry — by default, all entries of level 2+ are equally indented in 1.5em (`auto`).

`toc-bold:` `boolean`

Allows bold fonts in table of contents entries.

`chapter-numrestart:` `boolean`

Make chapter numbering restart after each book part.

`two-sided:` `boolean`

Otimizes the content to be printed on both sides of the page (front and back), with important elements always starting at the next front side (oddly numbered) — inserts blank pages in between, if needed.

ADVANCED NUMBERING

The book headings can be numbered two ways: using a *standard*[2] numbering string or a *numbly*[3] numbering array. Strings are more simple and easy to use, while arrays are more complete and customizable.

By default, *min-book* uses slightly different numbering when `#book(part)` is enabled or disabled, that's why *parts* and *chapters* appear to have independent numbering when used. The `#book(cfg.numbering-style)` option allow to set a custom numbering used whether `#book(part)` is enabled or disabled.

BOOK PARTS

```
#show: book.with(
  part: "Act",
)
= This is a part!  // Act 1
```

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. Each book can set different names for them, like parts, subjects, books, acts, units, modules, etc; by default, *min-book* tries to get the word for "Part" in `#book(cfg.lang)` language as its name (fallback to English).

When set a value (`string`), all level 1 headings become *parts*: they occupy the entire page and are aligned at its middle; some decorative frame also appear when `#book(cover: auto)`.

BOOK CHAPTERS

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

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

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. Each book can set different names for them, like chapters, sections, articles, scenes, etc; by default, *min-book* tries to get the word for "Chapter" in `#book(cfg.lang)` language as its name (fallback to English).

---

[2]https://typst.app/docs/reference/model/numbering/
[3]https://typst.app/universe/package/numbly

Chapters are smart: when set a value (`string`), if `#book(parts: none)` all level 1 headings become chapters; otherwise, all level 2 headings become chapters — since the level 1 are parts.

CATALOGING IN PUBLICATION

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

These `#book(catalog)` options set the data used to create the "cataloging in publication" board. Other needed informations are automatically retrieved from the book data, but at least one of these options must be explicitly set to generate the board; otherwise it will be just ignored.

`id:` `string` `content`

A *Cutter-Sanborn identification code*[4], 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.

---

[4]http://www.cutternumber.com/

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*[5] number, which corresponds to the specific category of the book.

udc: `string` `content`

An *Universal Decimal Classification*[6] number, which corresponds to the specific category if the book.

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

These commands are provided as a wa6 to access some fancy book features that cannot be implemented by re-working and adapting existing Typst elements. They are completely optional and is perfectly possible to write an entire book.

NOTE COMMAND

```
#import "@preview/min-book:1.1.0": note
#note(
  numbering-style: auto,
  content,
)
```

Adds an end note, an alternative for footnotes but placed inside of the page instead of its margins. End notes appear at its own page at the end of the current section, right before the next heading.

---

[5]https://www.oclc.org/content/dam/oclc/dewey/ddc23-summaries.pdf
[6]https://udcsummary.info/php/index.php

`numbering-style:`   auto   array   string

Custom note numbering — a standard numbering (`string`) or a *numbly*[7] numbering (`array`).

`content`   content   (*required*)

The content of the end note.

APPENDICES AND ANNEXES COMMAND

```
#import "@preview/min-book:1.1.0": appendices
#appendices(
  type: "appendix",
  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
)
```

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. Inside this ambient, all level 1 heading is a new appendix.

`title:`   array   auto

The name of the appendices — an array (`SINGULAR`, `PLURAL`) to each "Appendix" and the part "Appendices" names, respectively, or defaults to these same words in current language (`auto`).

`numbering-style:`   array   string

Custom heading numbering for appendices — a standard numbering (`string`), or a *numbly*[8] numbering (`array`).

---

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

body   `content`

The appendices content.

ANNEXES COMMAND

```
#import "@preview/min-book:1.1.0": 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
)
```

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. Inside this ambient, all level 1 heading is a new annex.

title: `auto`   `array`

The name of the annexes — an array (`SINGULAR, PLURAL`) to each "Annex" and the part "Annexes" names, respectively, or defaults to these same words in current language (`auto`).

numbering-style: `array`   `string`

Custom heading numbering for annexes — a standard numbering (`string`), or a *numbly*[9] numbering (`array`).

body   `content`

The annexes content.

HORIZONTAL RULE COMMAND

```
#import "@preview/min-book:1.1.0": horizontalrule
#horizontalrule(
  symbol: [#sym.ast.op #sym.ast.op #sym.ast.op],
  spacing: 1em,
```

---

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

```
    line-size: 15%,
)
```

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

symbol: `content`

Decoration in the middle of the horizontal rule — defaults to 3 asterisks.

spacing: `length`

Vertical space before and after the horizontal rule.

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

BLOCK QUOTE COMMAND

```
#import "@preview/min-book:1.1.0": blockquote
#blockquote(by: none, ..args)
```

Adds a block of quotation, a simple alias to `#quote(block: true)`, with a smaller and more semantic `#quote(attribution)` option as `#blockquote(by)`.

COPYRIGHT

Copyright © 2025 Maycon F. Melo.
This manual is licensed under MIT.
The manual source code is free software: you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law.

The logo was obtained from Flaticon website.