



Minimal Articles

Manual

Simple and easy way to write ABNT-compliant articles

min-article

0.1.0

MIT

Maycon F. Melo.*

CONTENTS

| | |
|--------------------------------------|----------|
| QUICK START | 2 |
| DESCRIPTION | 2 |
| OPTIONS | 2 |
| ABSTRACT COMMAND | 5 |
| ABBREVIATIONS COMMAND | 5 |
| GLOSSARY TERM COMMAND | 6 |
| APPENDIX COMMAND | 6 |
| ANNEX COMMAND | 7 |
| ACKNOWLEDGMENTS COMMAND | 7 |
| FIGURE COMMAND | 7 |
| A NOTE ON BIBLIOGRAPHY | 8 |
| TRANSLATIONS | 8 |
| COPYRIGHT | 9 |

*<https://github.com/mayconfmelo>

QUICK START

```
#import "@preview/min-article:0.1.0": article
#show: article.with(
  title: "Main Title",
  subtitle: "Complementary subtitle",
  foreign-title: "Título Principal",
  foreign-subtitle: "Subtítulo complementar",
  authors: (
    ("Main Author", "Doctor in Procrastination, etc, etc."),
    ("Main Collaborator", "Graduate in Doing Nothing, etc, etc."),
    ("Collaborator", "Student of Procrastination, etc, etc.")
  ),
  lang-foreign: "pt"
)
```

DESCRIPTION

Generate a authentic, structured, and standard articles, compliant with the Brazilian Association of Technical Norms (ABNT) requirements. The main differential of this package, apart from following the ABNT standards set by the NBR 6022 and NBR 14724 norms, is being able to manage by itself almost all of that mind-frying document structuration and its little rules: just input the data at any place, and *min-article* will find where it belongs in the article, and will put it there.

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.

OPTIONS

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

```
#show: book.with(
  title: none,
  foreign-title: none,
  subtitle: none,
  foreign-subtitle: none,
  authors: none,
  abstract: none,
  foreign-abstract: none,
  acknowledgments: none,
  date: auto,
  paper: "a4",
  lang: "en",
  lang-foreign: none,
  lang-data: toml("assets/lang.toml"),
  justify: true,
  line-space: 0.3em,
  par-margin: 1.5em,
```

```
margin: (
  top: 3cm,
  bottom: 2cm,
  left: 3cm,
  right: 2cm
),
font: ("Book Antiqua", "Times New Roman"),
font-size: 12pt,
)
```

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

title: string content (required)

The article main title in native language.

foreign-title: string content

The article title in a foreign language — generally a *lingua franca*: a language known and used universally (today it is English).

subtitle: string content none

The article subtitle; generally two lines long or less.

foreign-subtitle: string content none

The article main subtitle in a foreign language.

authors: array (required)

An array of arrays containing the name and biref résumé of each author, in format ((NAME, RÉSUMÉ), (NAME, RÉSUMÉ)). If a plain array is set, its first element will be treated as NAME and its second one, RÉSUMÉ.

abstract: content

The ugly way to define the abstract — a formal resume of everything the document talks about. It is prettier and more intuitivd to use the `#abstract("main")` command instead.

foreign-abstract: content

The ugly way to define the foreign abstract. It is prettier and more intuitive to use the `#abstract("foreign")` command instead.

acknowledgments: content

The ugly way to define the acknowledgments — the text with final thanks. It is prettier and more intuitive to use the `#acknowledgments` command instead.

date: array auto

The article publication date, in format (yyyy, mm, dd). Fallback to today's date if not set, or set to auto.

paper: string

Defines the page paper type — and its size therefore.

lang: string

Defines the main language of the written text.

foreign-lang: string

Defines the foreign language use on the secondary title and abstract.

lang-data: file

Defines a custom *linguify*¹ file with translations for the automatically created headings. The default file officially has full support for English and Portuguese, as well as some other languages through AI translation.

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 `line-space` to remove the paragraphs without additional space in between.

margin: length

Defines the document margins.

font: string array

Defines the font families used for the text: a principal font and its fallbacks. `font` too.

font-size: length

Defines the size of the text in the document.

¹<https://typst.app/universe/package/linguify>

ABSTRACT COMMAND

```
#import "@preview/min-article:0.1.0": abstract
#abstract(
  type,
  body
)
```

This is the best way to define the article abstracts. This commands just captures the abstract content and feed it to the `article` command where the abstracts are actually generated. Because of that, you can put this command anywhere inside the document body and it will be generated in the right place, right after the titles, as the ABNT NBR 6022 determines.

type `string`

Defines which abstract this is: the "main" abstract, or the "foreign" abstract; only those two string values are accepted.

body `content`

The abstract content.

ABBREVIATIONS COMMAND

```
#import "@preview/min-article:0.1.0": abbrev
#abbrev(
  abbreviation,
  long,
  definition
)
```

This command manages abbreviations as the ABNT NBR 14724 requires: The first time it is used, will print the full long name, and its abbreviation between parenthesis; from then on, when is used again with the same abbreviation, it will print just the abbreviation. Additionally, every new abbreviation is collected to be used in a glossary automatically generated, along with a optional longer definition of the abbreviation and its long name.

abbreviation `string` `content` *(required)*

The abbreviation itself. Does not support any stylization os quotes — even apostrophes. Is recommended to use just lowercase letters — automatically uppercased, — and do any needed stylization outside the `#abbrev` command.

long `string` `content`

The long name represented by the abbreviation.

definition string content

A optional definition to be used in the glossary. When no definition is set, the long name is used as the definition.

GLOSSARY TERM COMMAND

```
#import "@preview/min-article:0.1.0": gloss
#gloss(
  term-name
  definition
)
```

This commands works alongside `#abbrev` collecting terms for a automatically generated glossary. While `abbrev` only collects abbreviations, `gloss` can collect any word or expression. It just retrieves the data and then prints the term in the place where the command is written; later, the `article` command use this data to generate a automatic glossary after the man text body, as the ABNT NBR 6022 determines. If not data is collected by neither `gloss` nor `abbrev`, no glossary is generated.

term-name string content *(required)*

The name of the term in the glossary; it is what will be defined. If the name, have any fancy characters like apostrophes or quotes, is recommended to use a string instead of content.

definition string content *(required)*

The definition of `term-name`; a brief text that describes and explains it.

APPENDIX COMMAND

```
#import "@preview/min-article:0.1.0": appendix
#appendix(
  appendix
)
```

This commands just captures appendices given to it along the text body and feed them to the `article` command where the appendices are actually generated. Because of that, you can put this command anywhere inside the document body and it will be generated in the right place, right after the glossary, if any, as the ABNT NBR 6022 determines. It is possible to use the command multiple times to collect multiple appendices. Appendices are additional data and information that was not included in the document itself, but was cited or is related by it.

appendix content

(required)

The appendix content itself. In longer documents, is recommended to use separated *typ* file for appendices and include it here. Inside the content, each level 1 heading is treated as a new appendix.

ANNEX COMMAND

```
#import "@preview/min-article:0.1.0": annex
#annex(
  annex
)
```

This commands just captures the annexes given to it along the text body and feed them to the `article` command, where the annexes are actually generated. Because of that, you can put this command anywhere inside the document body and it will be generated in the right place, right after the appendices, if any. It is possible to use the command multiple times to collect multiple annexes. Annexes are documents made or data collected by another person that was cited or is related to the actual document.

annex content

(required)

The annex content itself. In longer documents, is recommended to use separated *typ* file for annexes and include them here. Inside the content, each level 1 heading is treated as a new annex.

ACKNOWLEDGMENTS COMMAND

```
#import "@preview/min-article:0.1.0": acknowledgments
#acknowledgments(
  thanks
)
```

This commands just captures the acknowledgments content and feed it to the `article` command, where the acknowledgments section is actually generated. Because of that, you can put this command anywhere inside the document body and it will be generated in the right place, right at the very end of the document as the ABNT NBR 6022 determines. Acknowledgments is a text to thank and recognize the worth of prople with important roles in the creation of the article.

FIGURE COMMAND

```
#import "@preview/min-article:0.1.0": figure
#figure(
  source: none,
  alignment: center,
```

```

    ..figure-arguments
)

```

This commands shadows the default `#figure` command to add two arguments to it. As the ABNT NBT 6022 requires, all figures must have a caption on its too and a source on its bottom, and this commands fulfills it by adding the `source` argument — the other argument, `aglnment`, is just miscelanea to align the figure using less code.

source: `string content` (*required*)

The source of the information displayed in the figure — even if is the author itself. Generally, is a bibliographical reference or just “Author” and the year of the figure criation.

alignment: `alignment` (*required*)

A sintatic sugar to align the figure in the page with less code.

..figure-arguments `argument` (*required*)

Any of the default original `#figure` arguments, all of them are supported in this command.

A NOTE ON BIBLIOGRAPHY

Unfortunately, there is no way to load the bibliography file to let *min-article* deal with it and generate the Bibliography in the right place. So this is up to you. This means if you insert the command in some place, it will appear in the place where it was written and not in the right place automatically. I’m looking foward to workaround this issue, but seems like this is a matter of Typst’s limitation.

Because of that, as it is necessary to manually set where the bibliography will be, write the `bibliography` command after the end of all text body content and it will appear at the right place as ABNT NBR 6022 requires: after the text and before the glossary, if any. It can be written before any of the collector commands described here, but any other content writen after it will appear after it.

TRANSLATIONS

This package uses *linguify*² to translate the headings automatically generated. Currently, it offers support to English and Portuguese, and some other languages by IA translations. If your language is not supported, or wrongly translated by AI, consider contrivuting to this project by helping in the translation.

²<https://typst.app/universe/package/linguify>

COPYRIGHT

Copyright © 2025 Maycon F. Melo.

This manual is licensed under MIT terms and rights.

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.

This project does not have any association with the Associação Brasileira de Normas Técnicas, and the — costly — norms cited here are all property of ABNT. This is not an official project.