



# 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 .....	8
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", "PhD in Procrastination, etc."),
    ("Main Collaborator", "Degree in Doing Nothing, etc."),
    ("Collaborator", "Procrastination Student, etc.")
  ),
  lang-foreign: "pt"
)
```

## DESCRIPTION

Generate authentic, structured, and standardized articles, compliant with the requirements of the Brazilian Association of Technical Standards (ABNT, in Portuguese). The main advantage of this package, apart from the ABNT standard, is being able to manage, all by itself, almost all the mind-frying document structuration and its rules: just input the data anywhere and *min-article* will find where it belongs, 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 awful lot to start with, but let's just break down all this to understand it better, shall we?

**title:** string content (required)

The article title in the main language.

**foreign-title:** string content

The article title in a foreign language — generally a *lingua franca*.

**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 — generally a *lingua franca*.

**authors:** array (required)

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

**abstract:** content

The ugly way to define the abstract — a formal resume of everything the document talks about. It is prettier and more intuitive 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` `datetime`

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, used on the full text.

**foreign-lang:** `string`

Defines the foreign language, used on the foreign title and abstract.

**lang-data:** `file`

Defines a custom *linguify*<sup>1</sup> file with translations for the automatically created headings. The default file have officially 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 get 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-size:** `length`

Defines the size of the text in the document.

---

<sup>1</sup><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` *(required)*

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

**body** `content` *(required)*

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, 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 automatically generated glossary, along with a optional definition of the abbreviation and its long name.

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

The abbreviation itself. Does not support any stylizationz quotes, or 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 that the abbreviation represents.

**definition** string content

An optional definition, used in the glossary. When no definition is set, the long name is used as definition.

## GLOSSARY TERM COMMAND

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

This command works alongside `#abbrev` collecting terms for an 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 uses this data to generate an automatic glossary after the main text body, as the ABNT NBR 6022 determines. If no data is collected by neither `gloss` nor `abbrev`, no glossary is generated.

**term-name** string content *(required)*

The name of the glossary term; 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)*

A concise and descriptive text about the `term-name`; it is what defines the glossary term.

## APPENDIX COMMAND

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

This command 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 not included in the document itself, but cited or related to it.

#### **appendix** content

(*required*)

The appendix content itself. In longer documents, is recommended to use separated *typ* files for appendices. 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 command 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, as the ABNT NBR 6022 requires. It is possible to use the command multiple times to collect multiple annexes. Annexes are third-party documents or data cited or related to the actual document.

#### **annex** content

(*required*)

The annex content itself. In longer documents, is recommended to use separated *typ* file for annexes. 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 command 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 used to thank and recognize the worth of people 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 command shadows the default `#figure` command to add two arguments to it. The ABNT NBR 6022 requires that all figures have a caption above it and its source below, and this command allows this by adding the source argument; the other argument, `alignment`, is just syntax sugar 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.

**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 limitation of Typst itself.

Because of this limitation, write the `bibliography` command after the end of all text content and it will appear at the right place, as ABNT NBR 6022 requires: after the text and before the glossary, if any.

## TRANSLATIONS

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

---

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



## **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.

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.