Minimal Writings

Manual

Simple and basic general purpose writings

min-writing 0.1.0 MIT

Maycon F. Melo[*]

QUICK START

```
#import "@preview/min-writing:0.1.0": writing, pagebreak
#show: writing.with(
  title: "Main Title",
  subtitle: "Subtitle, generally two lines long or less",
  authors: (
    "Text Author <mailto:author@email.com>",
    "Collaborator <https://collab.site.com>"
  ),
  mode: "mobile",
)
```

DESCRIPTION

Generate everyday texts, writings, and anotations that can be confortably read in smartphone screens, bigger screens, or printed paper. This package also provides some useful tools and helpers that simplify the writing process, to make it easier and faster. The features can work in a all-parts-detachable way: you can import each feature separately and use only whatever is useful at the moment instead of everything that `#writing` offers.

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-writing:0.1.0": writing
#show: writing.with(
  title: none,
  subtitle: none,
  authors: none,
  mode: "mobile",
  pages: false,
  glossary-title: "Glossary",
  unnumbered-heading-markup: true,
  date: datetime.today(),
  paper: "a4",
  lang: "en",
  justify: true,
  line-space: 0.5em,
  par-margin: 0.5em,
  first-line-indent: 1em,
  margin: (
    top: 3cm,
    bottom: 2cm,
    left: 2.5cm,
    right: 2.5cm
  ),
  font: ("Book Antiqua", "Inconsolata"),
  font-size: 12pt,
  body
)
```

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

`#show: writing.with()` → content

Pass the entire document through `#writing` rules and stylizations with given arguments, and then show it.

`title: none` string content none

The title of the text, if any.

`subtitle: none` string content none

The text subtitle, if any; generally two lines long or less.

`authors: none` string array

The author or authors of what is being documented — not the manual. When more than one author, is an array of strings, in format `("NAME <URL>", "NAME <URL>")`, where `<URL>` is optional.

`mode: "mobile"` string

The document layout mode: if `"mobile"`, the document will be suitable for reading on smartphones; if `"screen"`, it will be suitable to be read in bigger screns; and if `"print"`, it will be suitable to be read printed on paper.

`pages: false` boolean

Set multiple pages on in *mobile* and *screen* modes — by default, these modes use a singular page with automatic height.

`glossary-title: "Glossary"` string content

Set the title for the automatically created glossary, if any.

`unnumbered-heading-markup: true` boolean

Turn on/off special `|=` markup for unnumbered headings.

`date: datetime.today()` array datetime

The text writing or publication date, in an array `(yyyy, mm, dd)`, or a proper `#datetime`.

`paper: "a4"` string

Defines the page paper type — and its size, therefore.

`lang: "en"` string

Defines the text language.

`justify: true` boolean

Defines if the text will have justified aligment.

`line-space: 0.5em` length

Defines the space between text lines in paragraphs.

`par-margin: 0.5em` length

Defines the margin space after each paragraph. Set it the same as `line-space` to get no additional space between paragraphs.

first-line-indent: 1em length

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

margin: (top: 3cm, bottom: 2cm, x: 2.5cm) length

Defines the document margins.

font: ("Book Antiqua", "Inconsolata") array string

Defines the font families for the text: a main font and its falbacks.

font-size: 12pt length

Defines the size of the text in tue document.

body content

The entire document content (automatically set in `#show` rules).

PAGEBREAK COMMAND

```
#import "@preview/min-writing:0.1.0": pagebreak
#pagebreak(..args)
```

This is a wrapper that shadows the default `#pagebreak()`, to nullify it in `mobile` and `screen` modes — otherwise the document will be printed only until the first `#pagebreak()`.

ABBREVIATIONS COMMAND

```
#import "@preview/min-writing:0.1.0": abbrev
#abbrev(
  abbreviation,
  ..definitions
)
```

This command manages abbreviations this way: 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 stylization or quotes. Lowercase letters are automatically uppercased.

..definitions arguments

Receives two positional arguments: a LONG full name of the abbreviation, and an optional short DEFINITION to be used in the glossary.

When no DEFINITION is set, LONG is used as the DEFINITION.

The ..definitions is an argument sink that makes it possible to get optional positional arguments. This way you can use #abbrev either way:

```
#abbrev[idk][I do not know]
```

or else

```
#abbrev[idk][I do not know][
  Response to a question whose answer is unknown to you.
]
```

GLOSSARY TERM COMMAND

```
#import "@preview/min-writing: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 writen; later, the article command use this data to generate a automatic glossary after the main text body, as the. If no 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 definution of term-name; a brief text that describes and explains it.

HORIZONTAL RULE COMMAND

```
#import "@preview/min-writing:0.1.0": horizontalrule
#horizontalrule(
  symbol: [#sym.ast.op #sym.ast.op #sym.ast.op],
  spacing: 1em,
  line-size: 15%,
)
```

Adds horizontal rules, used to separate suble changes of subject in book texts. Can be called by its name or its alias `#hr`.

`symbol: [#sym.ast.op #sym.ast.op #sym.ast.op]` content

Defines the content at the center of the horizontal rule. By default, the line is struck by three ∗ in its center.

`spacing: 1em` length

Defines the vertical space before and after the horizontal rule.

`line-size: 15%` length

Defines the size of the horizontal rule line.

BLOCKQUOTE COMMAND

```
#import "@preview/min-writing:0.1.0": blockquote
#blockquote(by: none, ..args)
```

Adds a block version of the `quote` command. In fact, it is just a simple wrapper of `quote(block: true)` with some minor modifications. The `by` argument is an alias for the original `attribution` argument, so that a blockquote can be written:

```
#blockquote(by: "Einstein")[
  Don't believe everithing you read on the internet.
]
```

ARGUMENT COMMAND

```
#import "@preview/min-writing:0.1.0": arg
#arg(
  title,
  body
)
```

This command offers a convenient way to document the arguments — or parameters, or options, or whatever they are called; and even structures can be easily explained.

title string

Defines the argument NAME, TYPEs, and if it is REQUIRED using the following syntaxes:

```
"NAME <- TYPE | TYPE | TYPE <REQUIRED>"
```

```
"NAME -> TYPE | TYPE | TYPE"
```

The name can be ```LANG NAME``` to get syntax highlight. The <- arrow indicates that NAME receives any of the value TYPEs, and -> indicates that NAME returns one of the value TYPEs. A special nothing type is used when nothing is received and/or returned. For optional arguments, just don't write the <REQUIRED>.

body content string

A brief description of what the argument does.

UNNUMBERED HEADINGS COMMAND

```
#import "@preview/min-writing:0.1.0": unnum-headings
#show: unnum-headings.with(body)
```

Activates only the unnumbered headings syntax, without the need to use #writing and all its other features included. By importing and using this command, you can write unnumbered headinds using a special syntax:

```
#show: unnum-headings

|= This is a unnumbered level 1 heading
|== This is a unnumbered level 2 heading
```

COPYRIGHT

This is glossary