markdown to pdf typst template

2024/12/21

Contents

1. Description 1
   - 1.1. Supported styles 2
   - 1.2. Version of the tools 2
   - 1.3. Contents of the template package directory 3
   - 1.4. Default design in the template 5
     - 1.4.1. PDF page 5
     - 1.4.2. Fonts 5
     - 1.4.3. Default font size 8
     - 1.4.4. Indentation 8
     - 1.4.5. Code highlight color theme 9
     - 1.4.6. Style modification 9
   - 1.5. How to use the pandoc template program 10
     - 1.5.1. Run 11
   - 1.6. How to use themekit 12
     - 1.6.1. Run 13
     - 1.6.2. Integration with Sublime Text 13

2. front matter 14
   - 2.1. Article parameters 14
   - 2.2. Page layout style parameters 16
   - 2.3. Content insertion parameters 18

3. Limitations 19
   - 3.1. Pandoc limitations 19
     - 3.1.1. Duplicated footnote references 19
   - 3.2. typst limitations 19
     - 3.2.1. First line indent 19
     - 3.2.2. Image files 19
     - 3.2.3. Style theme file 20
     - 3.2.4. Italic 21
     - 3.2.5. Table row number 21

4. Tricks 21
   - 4.1. Reduce the size of PDF files 21
   - 4.2. typst exporting png images are not clear enough 21
   - 4.3. Hiding Heading Numbers 22

4.4. Hiding Page Numbers 22
4.5. No Pagination 22
5. Demonstrations 23
5.1. Paragraph and line break 23
5.2. Emphasis 24
5.3. Ordered lists 25
5.4. Unordered lists 27
5.5. Task lists 28
5.6. Ignore Markdown Formatting 28
5.7. emojis 28
5.8. Links 29
5.9. Images 30
5.10. Footnotes 33
5.11. Definition lists 34
5.12. Mathematical Equation 35
5.13. Code and syntax highlighting 36
5.14. Tables 38
5.14.1. Table title table caption 41
5.14.2. simple tables 42
5.14.3. Multi-line table 42
5.14.4. grid tables 43
5.15. alerts 46
5.16. Table width and column width 48
5.17. Quotes 51
5.18. alerts 52
5.19. Horizontal Rule 54
5.20. Inline HTML 55
5.21. Title 55
5.22. Alt-H2 56
6. (only for demonstration) h1 title 56
6.1. h2 title 56
6.1.1. h3 title 56
7. (only for demonstration) Alt-H1 56
7.1. Alt-H2 56

1. Description

This template is used to convert markdown to pdf documents using pandoc and typst. Therefore, this template is somewhat prefabricated. This template will affect the typesetting layout in the generated pdf document.

1. This template is suitable for: The preset styles in the template are suitable for individuals or teams to write technical documents and instruction documents for communication or sharing, reflecting personalization. It can also be used to learn how to program the typesetting and formatting in typst.
2. This template is not suitable for: It is not for academic and paper fields with strict formats, such as lack of special types such as abstract, citation, scientific and mathematical charts. However, you can modify it based on this template. (You need to understand typst syntax)

Advantages

1. Used to quickly convert markdown to pdf. Required tools Pandoc and typst are two command line tools.
2. Support common styles, used to replace word or other heavyweight writing tools to write lightweight documents such as (work summaries, personal diaries, team task plans, and other common short and medium-length articles)
3. Use typst to replace latex to avoid using too many tools. typst is currently the lightest and most efficient solution for generating PDF documents
4. No need to convert markdown to html or docx first, and then use a browser or word/wps to convert it to pdf.
5. Offline use, when converting to generate PDF files..

Disadvantages

Due to the use scenario setting, it does not support professional writing requirements, and cannot generate papers, academic reports, and book publishing that have higher requirements for typesetting and formatting. Because the template is set to be used in an offline environment, the template does not use the typst universe extension package. However, this extension can be completed by yourself.

> 🚧 Caution
>
> The original intention of this template is to quickly generate a self-righteous pdf document after writing a markdown document in the sublime text editor for information exchange or sharing. Therefore, it does not exist as a writing tool. It only defines the typst layout and rendering layout of some commonly used document structure types, and does not cover all the markdown properties of pandoc. For the current specific supported styles, see Supported Styles

If you need more comprehensive writing support, you can use quarto (supports users in multiple fields such as Scholarly Writing, Presentations, Dashboards, Books, Manuscripts, and Publishing).

1.1. Supported styles

- Paragraphs and line breaks
- Emphasis
- Unordered lists
- Ordered lists
- Task lists
- Emojis
- Links
- Images
- Footnotes
- Definition lists
- Math equation
- Code highlight fenced code block
- Tables
- Table width and column width
- Quote quote|blocktext
- alerts
- Inline HTML
- Horizontal rule
- Title

1.2. Version of the tools

1. pandoc `3.5` Download and installation instructions, https://pandoc.org/installing.html
2. typst `0.12.0` Download and installation instructions, https://github.com/typst/typst/releases/tag/v0.12.0

First download pandoc and typst, and put them in the system environment variable `PATH` where they can be found.

- Macos/Linux operating system: You can use command line software tools such as brew, apt, mise, asdf, etc. to install.
- Windows operating system: You can use command line software tools such as scoop, Chocolatey, etc. to install. Or after downloading, add the directories where the pandoc and typst executable programs are located to the system variable `PATH`.

1.3. Contents of the template package directory

```
├──── pandoc
│     ├──── example.sh
│     ├──── pandoc
│     │     ├──── some pandoc typst tempaltes
│     │     ├──── stanalone with difference pdf page size
│     │     ├──── font color and highlighting theme
│     └──── themes
│           ├──── tmTheme files, define highlighting
├──── themekit
│     └──── 0.4.0
│           ├──── example.markdown.sublime-build
│           ├──── example.sh
│           ├──── fonts.typst
│           ├──── lib.typst
│           ├──── pandoc
│           │     ├──── Some pandoc typst template files
│           │     ├──── font color, code highlight theme
│           │     └──── ...
│           ├──── styled
│           │     ├──── typst rules file for presetting
│           │     ├──── page size, font color, highlight color
│           │     ├──── Each file is independent of each other
│           │     ├──── ...
│           │     └──── themes
│           │           ├──── ayu-dark.tmTheme
│           │           ├──── ...
│           └──── typst.toml
```

```
├──── pandoc.vs.themekit.pdf
├──── userguide.dark.ayu-dark.pdf
├──── userguide.light.ayu-dark.pdf
├──── pandoc.vs.themekit.pdf
├──── howto.pandoc.pdf
├──── howto.themekit.pdf
```

The template contains two distribution forms, `pandoc` and `themekit`, which are two different organizational methods, but the pdf files and png images converted by them have the same look.

- Using the template in the `pandoc` directory, you can generate a typst standalone file, which is convenient for independent debugging or modifying the typst result file. However, there are requirements for the location of the themes code highlight directory.
- `themekit` requires to be installed as a typst local package first. For installation methods, see the instructions in `howto.themekit.pdf`. The pandoc typst template in the `themekit` directory uses various rules definitions by referencing the typst package, which is convenient for generating target format files.

`pandoc` template directory:

1. `pandoc` directory contains independent pandoc typst templates. Each template file sets different colors and themes.
2. `.tmTheme` file in `themes` directory is the programming language highlight color theme file used by sublime text 3.1 and before. It is in XML format. typst uses it to define the color theme of code highlighting. These theme files are referenced in pandoc typst templates.

`themekit` template directory:

1. `styled` stores the rules of typst `show` or `set` according to page size, preset color, and highlight theme. Each file is independent and is re-exported in `lib.typst` of themekit.
2. `pandoc` directory stores references to the rules in `styled`, which can be used in the `--template` parameter of pandoc command line. Each file corresponds to a highlight theme.

Instructions or help documents:

1. `pandoc.vs.themekit.pdf`, which introduces the differences between the two template packages.
2. `howto.pandoc.pdf` and `howto.themekit.pdf` introduces how to use the template package.
3. `userguide`, which is the user help document. Compared with `howto`, this document is more detailed.

> i Tip
>
> The template file is named [phone.]`MODE`.`THEMES`.typst. If it starts with `phone`, the generated pdf page is 300 pt wide, which is convenient for reading on mobile phones.

> `MODE` is dark or light, corresponding to the dark or light background of the page, and `THEMES` is the code highlight color theme,

1.4. Default design in the template

1.4.1. PDF page

In this template package, there are two sizes of PDF page size, namely `A4` and `phone`. `A4` is used for normal documents, and `phone` size can be used for easy reading when sharing on mobile devices. The specific sizes are as follows:

Page	mode	size	border margin
a4	dark/ light	595.28 pt × 841.89 pt	2.5cm on top, bottom, left and right
phone	dark/ light	300 pt × 650 pt	12pt on top, left and right, 24 pt on bottom

The dark page background color value is `#10141c`, the font color value is `#bfbdb6`, the light page background color value is `#fcfcfc`, the font color value is `#5c6166`.

1.4.2. Fonts

There are two type of fonts in the document, one is the unified font used throughout the text, and the other is a monospaced font used to display code, called mono font.

- Main font: The template uses `Xiawu Wenkai GB` as the main font, which displays Chinese more elegantly.

- Monospaced font: This template package uses `Xiawu Wenkai GB` as the monospaced font, and the Western effect is consistent with the Chinese effect. If you like programming fonts such as fira code, you can modify it.

Xiawu Wenkai is a multi-version series font. The font of this template version is LXGW WenKai GB, and the corresponding English name is `LXGW WenKai GB`. Before using this template, you need to download these fonts and install them to the system. Download address of Xiawu Wenkai GB GitHub Project Network, Maoken Network

> Xiawu Wenkai GB The font weight is divided into light, regular, and medium. The overall style is relatively slender. And there is no built-in italic font, so the italic style cannot be displayed in the PDF generated by typst.

If the above fonts do not meet the requirements, you can modify them as needed. `pandoc` template package, the modification method is as follows:

```
// // for windows
// // fonts for emoji
// #let emmoji_fonts=("Segoe UI Emoji", "Noto Color Emoji")
// #let mainfont = ("Segoe UI",)
// #let monofont = ("Consolas",)

// for macos
// fonts for emoji
#let emmoji_fonts=("Apple Color Emoji", "Noto Color Emoji")
#let mainfont = ("SF Pro Text",)
#let monofont = ("SF Mono",)

// // for linux
// // fonts for emoji
// #let emmoji_fonts=("Noto Color Emoji")
// #let mainfont = ("Menlo",)
// #let monofont = ("DejaVu Sans Mono")

#let lang = "zh"
#let region = "CN"
#if lang == "zh" {
    mainfont=("PingFang SC", )
    // mainfont=("Microsoft YaHei UI", )
}

// fonts defined
#let fonts=(
    normal: (
        // typst 0.12: typst
        // normal font for whole doc
        font: (name: ("LXGW WenKai GB", ..mainfont, "Libertinus Serif", ..emmoji_fonts))
    ),
    codeblock: (
        // mono font for code snippets
        font: (name: ("LXGW WenKai Mono GB", ..monofont, "DejaVu Sans Mono", ..emmoji_fonts)),
    ),
    math: (
        // typst built-in math font, New Computer Modern Math
        font: (name: ("New Computer Modern Math", "LXGW WenKai Mono GB", ..emmoji_fonts)),
    )

)
```

Change `mainfont`, `monofont`, `LXGW WenKai GB`, `LXGW WenKai Mono GB` to your operating system or other default fonts.

The font settings in the `themekit` template are located in the `fonts.typst` file in the package, with the same content as above.

You can also set the mainfont and monofont parameters in the front matter. For specific parameters, see front matter.

i Tip

1. If the running prompt font does not exist, `unknown font family`, typst will fall back to the default system font, will not be interrupted. But not elegant.
2. The font mechanism of typst searches from the first font family of the set font group. If it cannot be found, it searches in the next font family. There may be inconsistencies when displaying emojis. For example, the symbol ⚠ uses the wireframe symbol style in `LXGW WenKai GB` instead of the emoji style in `Apple Color Emoji`. This is not a program error of typst. You can adjust the order of font families or use other fonts to solve it.

1.4.3. Default font size

- Text: 12 pt
- Title
  - Article title: 28pt
  - Heading 1: 20pt
  - Heading 2: 16pt
  - Heading 3: 14pt
  - Heading 4: 12pt
  - Heading 5: 12pt
  - Heading 6: 12pt
- Code block: 11pt
- Quote text block: 12pt

1.4.4. Indentation

1. The first line of a text paragraph is indented by four English characters.
2. The entire quotation (text block) is indented by four English characters to the right.
3. [Ordered list] and [Unordered list] are indented by two Chinese characters to the right in the body.
4. In alerts, quotes, and Definition lists, multiple text paragraphs, except for the first line of the first paragraph, cannot be indented*[1], the first line of other subsequent paragraphs is indented four English

[1] These situations occur in elements such as alerts, quotes, and Definition lists that use blocks to wrap content. It is related to the implementation mechanism of typst itself. Most elements and text layout and typesetting control use blocks. The body paragraph after the

characters wide. When only ordered lists and unordered lists appear alone, in order to maintain visual compactness, ordered lists and unordered lists are not indented.

⚠ Warning

typst 0.12 still does not solve the problem of marking the first paragraph of multiple consecutive paragraphs, so the first line of the first paragraph of the text segment cannot be indented.

1.4.5. Code highlight color theme

This template package currently includes modern and beautiful code highlight themes:

- ayu:
  - ayu-mirage
  - ayu-dark
  - ayu-light
- Catppuccin
  - Catppuccin Frappe
  - Catppuccin Latte
  - Catppuccin Macchiato
  - Catppuccin Mocha

In the pandoc typst template file, `#set raw(theme: )` specifies.

1.4.6. Style modification

In front of each independent pandoc typst template file, global variables that can be used by the entire document are defined. Most of the variable values in the template are concentrated here, such as font size, font name, color, etc.

1. Font: See the [Font] section above
2. Code highlight theme: In this template package, all color themes are listed and each corresponds to a template. You can select the corresponding template file according to your preference. If you have a custom `*.tmTheme` style file, you can modify it according to the method in Code highlight theme.

block can be implemented through special means, but there is no way to do it at the beginning of the block.

1.5. How to use the pandoc template program

This template program directory does not have an interface operation method. The template package is specified when using the pandoc command. The problems of image reference and code highlight theme reference have been described in typst limitations. If you want to be compatible with both problems, the `themes` directory in the template needs to exist in the same main directory as the image directory and markdown file, or have the same ancestor directory above.

The required directory structure is as follows Case 1:

```
my_notes
├── images
│   ├── a.png
│   └── ...
├── themes
│   ├── some tmTheme files
│   └── ...
├── notes.md
```

Or Case 2

```
my_notes
├── content
│   ├── images
│   │   ├── a.png
│   │   └── ...
│   ├── notes.md
│   ├── ...
├── themes
│   ├── some tmTheme files
│   └── ...
├── typst
│   ├── some pandoc typst template files
│   └── ...
```

Start pandoc in the directory `my_notes` , typst can access both `images` and `themes` directories at the same time.

But usually our directories are like this （images outside of pandoc_typst):

```
top
├── my_notes
│   ├── images
│   │   ├── a.png
│   │   └── ...
│   ├── notes.md
├── pandoc_typst
│   ├── themes
│   │   ├── some tmTheme files
│   │   └── ...
│   ├── typst
│   │   ├── some pandoc typst template files
│   │   └── ...
```

Of course, we can run pandoc in the top directory, but the following conditions must be met:

1. The image reference path in `notes.md` is `my_notes/images/xxx.png`
2. `#set raw(theme: )` in the typst template file must be `#set raw(theme: "pandoc_typst/themes/xxx.tmTheme")`

It looks weird no matter how you look at it.

You can create a symbolic link in the `my_notes` directory point to the `pandoc_typst/themes` directory. This is the simplest and most direct method. The method is described in the following section

1.5.1. Run

Command:

```
# linux / macos
cd CONTENT/DIRECTORY
template_dir=absolute path of template file package directory
template=$template_dir/pandoc/pandoc.dark.catppuccin-frappe.typst
md_args="+emoji+strikeout+definition_lists+implicit_figures+fancy_lists+yaml_metadata_block"
# How to create soft links without themes directory
[ ! -d themes ] && ln -s $template_dir/themes
cat markdownfile.md | pandoc -f gfm"$md_args" -t pdf --pdf-engine typst \
--template=$template -o OUTPUT.pdf
```

```
# After pdf is generated, delete the link
[ -L themes ] && rm themes;
```

1. `themes` The directory is linked to the content directory, so that for typst, the images, tmTheme files and other resources that need to be referenced are all in the same directory.
2. Use `cat markdown.md | pandoc -f gfm` or `pandoc markdown.md -f gfm`, because the markdown.md file may need to be preprocessed. If you need to use multimarkdown to preprocess the markdown file, you can do `multimarkdown -tmmd markdown.md | pandoc`

1.6. How to use themekit

The pdf file generated by `themekit` has the same visual effect as the `pandoc` template package. It is referenced in the form of typst package. When using the `themekit` package, there is no themes tmTheme highlighting theme file location, because it can exist as a component of the typst package. Just install `themekit` as a typst local package first.

The directory specification of the typst local package is: `{data-dir}`/typst/packages/`{namespace}`/`{name}`/`{version}`. Different operating systems have different `{data-dir}`, as follows:

- linux: `$XDG_DATA_HOME` or `~/.local/share`
- maxos: `~/Library/Application Support`
- whidowns: `%APPDATA%`

According to convention, `{namespace}` is `local`, the name of this template package is `themekit`, and `{version}` corresponds to the latest version number released by themekit.

Copy or link the themekit directory to the `{data-dir}`/typst/packages/local directory, such as:

```
mkdir -p "{data-dir}/typst/packages/local"
cp -R themekit "{data-dir}/typst/packages/local"
# or
ln -s themekit "{data-dir}/typst/packages/local/themekit"
```

Change `{data-dir}` to the directory corresponding to your operating system.

1.6.1. Run

Command:

```
cd markdown/document directory
template="{data-dir}/typst/packages/local/themekit/version number/pandoc/any
pandoc template in the directory.typst"

pandoc_args="markdown_mmd+emoji+strikeout+fancy_lists+task_lists+implicit_
figures+link_attributes+inline_notes+tex_math_single_backslash+simple_tables+
grid_tables+multiline_tables+table_captions+yaml_metadata_block-blank_before_
blockquote"
cat markdown document.md |
pandoc --from $pandoc_args \
-t pdf --pdf-engine typst --template="$template" \
-o "output file name.pdf"
```

Change `{data-dir}` to the directory corresponding to your operating system. Any typst template file in the pandoc directory, distinguished by page size, dark/light background, code highlighting theme.

There are example shell scripts in the directory, see `{data-dir}`/typst/packages/themekit/version number/`examples.sh`

1.6.2. Integration with Sublime Text

By integrating the above command line into the build system of Sublime Text, you can quickly perform conversions without switching to the command line terminal.

1. Step 1: Create a new sublime text build system as follows: A. Menu `setting/Brawse Packages...`, create a blank text file `markdown.sublime-build` under `Packages/User` B. `cmd` + `shift` + `p`, open the command panel, enter `new build system`, save the file as `markdown.sublime-build`
2. Step 2: In the `markdown.sublime-build` file, save the following content:

```
{
    "selector": "text.html.markdown",
    "variants": [
        {
            "name": "md to pdf (by typst)",
            "working_dir": "$file_path",
            "env": {
                // 修改为 themekit 所在目录`
                "THEMEKIT_DIR": "path/to/themekit",
                "PANDOC_MD_ARGS": "--from=markdown_mmd+emoji+strikeout+fancy_
lists+task_lists+implicit_figures+link_attributes+inline_notes+tex_math_
single_backslash+simple_tables+grid_tables+multiline_tables+table_captions+
yaml_metadata_block-blank_before_blockquote",
                "PANDOC_TARGET": " -t pdf --pdf-engine typst ",
                // 修改为对应高亮主题模板
                "PANDOC_TEMPLATE": "themekit/{version number}/pandoc/
dark.catppuccin-frappe.typst"
            },
            "shell_cmd": "cat \"$file\" | pandoc \\$PANDOC_MD_ARGS \\$PANDOC_
TARGET --template=\\$THEMEKIT_DIR/\\$PANDOC_TEMPLATE -o $file_path/$file_base_
name.pdf",
        },
    ]
}
```

Set the `THEMEKIT_DIR` parameter to the local directory path of the template package, then edit a markdown file in sublime text at will, open the command palette with `cmd` + `shift` + `p`, enter `pdf typst`, and press `Enter`. A pdf file is generated in the same directory as the markdown file. The `PANDOC_TEMPLATE` parameter specifies which template to use, and each template has a unique preset color.

You can also use pandoc and typst to generate png images with the same pdf style.

`Themekit` contains the example file `example.markdown.sublime-build` for sublime text markdown build.

2. front matter

When pandoc interprets markdown documents, it supports interpreting the parameters in front-matter. We can use these parameters to modify the document template style.

The parameters supported by this document template are:

2.1. Article parameters

title

Article title, type is `text`.

subtitle

Article subtitle, type is `text`.

author

Article author. Type is `text`, `object`, or `text array`, `object array`. When there are multiple signatures, use `,` to separate them, and use `[]` to wrap multiple names. When using an object to describe the author, the three properties of `name`, `affiliation`, and `email` are supported. For example, Multiple author names:

```yaml
# Single name
author: Author name
# Multiple names
author: [Author 1, Author 2]
```

Use objects to describe authors:

```yaml
# Single name
author:
name: Author name
affiliation: nullable
email: Email, nullable
# Multiple names
author:
- name: Author name
affiliation: nullable
email: Email, nullable
- name: Author name
affiliation: nullable
email: Email, nullable
```

date

Article date, type `text`

lang

Language and region parameters, format `lang-region code`, `lang` value is ISO 639-1/2/3 language code. `lang` sets the language properties in typst, affecting various parts of Typst,

- The text processing pipeline can make smarter choices
- Hyphens will use the correct pattern for the language
- smart quotes will become the correct quotes for the language
- And all other language-aware content

For example, when `lang` is `zh`, the table of contents title is Table of Contents, when `lang` is `en`, the table of contents title is Contents. `region` allows typist to make smarter choices when processing text. The value is ISO 3166-1 alpha-2 region code

abstract

Abstract, type `text`, because the abstract may have multiple lines, follow the multi-line syntax of yaml, such as

```yaml
abstract: |
This is the abstract.

It consists of two paragraphs.
```

citations

Citations, see Citation rendering

bibliographystyle/csl

The format of bibliography, type `text`, value see bibliography style

bibliography

Bibliography or parameter document, type `text`, point to `Hayagriva.yml` or The path of the `BibLaTeX .bib` file. The relative path rules comply with the path safety regulations of typst.

2.2. Page layout style parameters

mainfont

Default font, type is `text` or `text array`, enumerate the fonts used in the text. It should be noted that the listed fields already exist in the current operating system, otherwise a warning of unavailable fonts will appear. As shown in the following example, the fonts in `Fira Code` are used first, and the Chinese font is `PingFang SC`:

```yaml
mainfont: [Fira Code, PingFang SC]
```

monofont

Monospaced font, type is `text` or `text array`, specifies the code block font in the generated pdf document, and generally uses a programming font that is friendly to developers. The format is the same as `mainfont`.

mathfont

Mathematical formula font, type is `text` or `text array`, format is the same as `mainfont`. Used to set non-English fonts, such as Chinese in formulas.

fontsize

Font size, type typst length. For example, 12pt.

papersize

Page size, the name of the predefined page size in typst, common ones are "a4", "us-letter", etc., see typst page paper for details. papersize will override the original page size of the document template. When setting `pagesize`, the current `papersize` page size can be overwritten.

pagesize

Page size, see typst page width and page height parameters, which can override papersize. The actual page size below is `400pt` wide and `800pt` high, regardless of the value of `papersize`. When `pagesize: {height:auto}`, no pagination is done, which can be used to generate long pages.

```
papersize: presentation-16-9
pagesize:
width: 400pt
height: 800pt
```

margin

Page margin, see typst page margin, the default value is described in pdf page.

page-numbering:

Page number format, see typst page numbering. When set to `disable` or `none`, no page numbers are displayed.

section-numbering:

Heading numbering, see typst heeading numbering. When `section-numbering` is set to `disable` or `none`, the heading numbering can be hidden.

columns

The number of page columns, default is `1`, see typst page columns.

toc

Table of contents, type is `bool`, default is `false`. When `toc: true`, the table of contents is displayed. When `pagesize.height` is `auto`, the table of contents is not displayed.

toc-depth:

Table of contents level, controls the displayable title level, type is a number, see typst outline depth

toc-title:

Table of contents title, when `toc-title: auto`, it is automatically defined according to `lang`, such as `lang: zh-CN`, the title is `Table of Contents`; when `toc-title: none`, the table of contents title is not displayed. See typst outline title.

table-caption-position:

Sets the position of the table title relative to the table, which is located above the table by default, see typst figure caption position.

figure-caption-position:

Set the position of the image caption relative to the image. By default, it is below the image. See typst figure caption position.

2.3. Content insertion parameters

header-includes

Pandoc parameter. Text content is inserted after the directory and before the main text. Usually in the header when converting markdown to html. JavaScript, css references. See pandoc Variables set automatically. You can also use the pandoc command line parameter -H/–include-in-header to specify the content file.

include-before

pandoc parameter, text content is inserted before the table of contents, see pandoc Variables set automatically, you can also use the pandoc command line parameter -B/–include-before-body to specify the content.

include-after

pandoc parameter, text content is inserted after the body, at the end of the document, see pandoc Variables set automatically, you can also use the pandoc command line parameter -B/–include-after-body to specify the content.

i Tip

When using the pandoc command, you need to use the `--from gfm` or `--from markdown_mmd` or `--from markdown+yaml_metadata_block` parameters.

3. Limitations

This manual is not a markdown specification, nor a typst syntax specification. Therefore, this manual will not include markdown and typst syntax descriptions.

3.1. Pandoc limitations

3.1.1. Duplicated footnote references

Pandoc has a 10-year-old bug issue 1603 that cannot correctly parse repeated footnotes.

3.2. typst limitations

3.2.1. First line indent

`typst 0.12` does not support the recognition of the first paragraph, so it is not possible to indent the first paragraph. This is related to its internal implementation. Many elements in typst use block as the outermost container to achieve the overall layout. Even `set par(first-line-indent: )` refers to the first line between consecutive paragraphs, that is, it does not include the first line of the first paragraph. If it is a paragraph after a block or box, special means can be used to deal with it, but there is currently no way to indent the first paragraph of text in a block or box. In alerts and quotes, when the text content is long or has multiple paragraphs, you will see that the first line of the first paragraph is not indented.

I hope that typst can solve this problem in future versions, so that some redundant code for compatibility can be removed.

3.2.2. Image files

To ensure that the typst script is safe and harmless, typst comes with a sandbox mechanism. When the `.typst` file is compiled into a pdf, the image files it references cannot use the url path, and the images can only be located in the directory or subdirectory where the `.typst` file is located. The relative path is used to specify the image file location. This is different from the unlimited path writing method in the markdown file. Special attention should be paid.

The image file must be located under the same top-level directory as the `*.typst` content file. There is no requirement for the hierarchy, as long as the relative path hierarchy in `*.typst` corresponds correctly. For example:

```
my_notes
├── images
│   ├── a.png
│   └── ...
├── readme.typst
├── readme.md
```

Assuming that the image file is located in the images directory, the path of the image file relative to the readme.typst content file is `./themes`, that is, when specifying the image path in readme.typst, fill in:

```
#image("./userguide/images/a.png")
```

Because pandoc is used to convert markdown, the way to specify the image reference in markdown is:

```
![Image example](./userguide/images/a.png)
```

According to this relative path, pandoc can convert markdown to pdf correctly through typst.

3.2.3. Style theme file

Similar to the image file reference, the code highlight color theme file is also referenced through a relative path. Currently there is no way to embed the theme file directly into the `*.typst` file.

The style theme file must be located in the same top-level directory as `*.typst`, regardless of the hierarchy. As long as the relative path hierarchy in `*.typst` is correct, it will be fine. For example:

```
pandoc_typst
├── themes
│   ├── some tmTheme files
│   └── ...
├── readme.typst
```

If the theme file is located in the themes directory, then the path of `.tmTheme` relative to readme.typst is `./themes`, that is, when specifying the theme path in readme.typst, fill in:

```
#set raw(theme: "themes/file name.tmTheme")
```

3.2.4. Italic

For fonts that do not have their own italic style fonts, the italic style cannot be displayed in the text.

3.2.5. Table row number

Typst's tables are very powerful and can draw tables of various forms, but there is currently a lack of a function similar to `y==table.footer.row-number`. When pandoc converts markdown's grid table, it is impossible to write typst rules to modify the style of the table footer.

4. Tricks

4.1. Reduce the size of PDF files

This template program uses typst to generate PDF documents, and does not consider the size of the PDF document after it is generated. The size of the PDF file is determined by the size of the embedded pictures, fonts, and redundant information. Before generating a PDF document, the image size should be reduced, or the image should be cropped or compressed to a size suitable for Nottingham. If high-definition pictures cannot be used without limit, after the PDF file is generated, the PDF compression tool can be used to compress the PDF. For example, macos os comes with a PDF file size reduce filter, and ghost script is used to compress PDF documents, etc.

The example pdfs in the `examples` directory of this template program release package are all compressed using ghost script. The original `4.6M` size of pdf is compressed to about `900k` as you can see, with significant compression effect. The command is as follows:

```
gs -dBATCH -dNOPAUSE -q -sDEVICE=pdfwrite -dCompatibilityLevel=1.7 \
-dNOPAUSE -dQUIET -dPDFSETTINGS=/prepress -sOutputFile="The file name was
lost" "Original pdf file"
```

4.2. typst exporting png images are not clear enough

Use `typst compile original.typst result-{0p}-of-{t}.png` to convert a typst document with a multi-page layout into multiple png images. When exporting, png uses twice the size of the pdf file. If the pdf page size is set too small, the image effect is not clear enough. For example, in this template program, the pandoc typst template starting with `phone.` has a page size of $300pt \times 650pt$, and the exported image is `600pt × 1300pt`. When the font is small, the image quality is not good. At this time, set typst to generate images with higher resolution. The example command is:

```
pandoc --from markdown... --typst --template=pandoc or pandc template under themekit |
typst compile --ppi 200 - output-{0p}-of-{t}.png
```

The key parameter is `--ppi 200`, which can be set to `--ppi 300` to meet the printing quality requirements. Because of the `--ppi` parameter option, we don't need to use the pdf to png image tool to generate higher resolution images.

4.3. Hiding Heading Numbers

Set the markdown front matter parameter `section-numbering` to `disable` or `none` to hide the heading numbers.

```
section-numbering: disable
```

4.4. Hiding Page Numbers

Set the markdown front matter parameter `page-numbering` to `disable` or `none` to hide page numbers.

```
page-numbering: disable
```

4.5. No Pagination

No Pagination , that is keep the whole doc in one long page, set `pagesize.height` to `auto`, then the table of contents and page numbers will not be displayed, even if `toc` is `true`.

```
# toc will be ignored
toc: true

pagesize:
height: auto
# or
pagesize: { height: auto }
```

5. Demonstrations

5.1. Paragraph and line break

> Note: This section makes conventions for paragraphs and line breaks.

Paragraphs, also known as "lines link", have soft and hard line breaks in markdown. Line breaks are not segmented, line breaks are automatically spliced, and two spaces are added to the end. They are very personalized. A markdown file only supports one convention.

However, in order to simplify writing, maintain the original writing habits, and reduce the memory and confusion of rules, this template recommends that you use the following rules to write markdown. You are required to write in the following form:

```
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,

Lorem ipsum dolor sit amet,
This is another line, but it still belongs to the same line as the previous
line.

To start a new paragraph, enter a blank line first.
This sentence still follows the previous sentence.

This is another new paragraph.
```

Rendered as:

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,

Lorem ipsum dolor sit amet, This is another line, but it still belongs to the same line as the previous line.

To start a new paragraph, enter a blank line first. This sentence still follows the previous sentence.

This is another new paragraph.

🚧 Caution

When using this template, according to the line break conventions in this template, it is not recommended to add the `+hard_line_breaks` extension to `pandoc -f gfm` or `pandoc -f markdown`. When writing long paragraphs, a lot of line breaks will be generated, affecting the layout of the complete paragraph.

If a new paragraph is really needed, it is implemented by a blank line. It is easy to read and write. Ordinary line breaks are considered to be convenient for writing, or the return key is pressed inadvertently, which does not affect the actual effect.

5.2. Emphasis

todo 字体、斜体

```
Emphasis, also known as italics, uses an *asterisk* or _underline_.

Emphasis, also known as bold, uses an **asterisk** or __underline__.

Use a combination of **asterisk and _underline_** for emphasis.

Strikethrough uses two tildes. ~~Strike this out.~~

**This is bold text**

__This is bold text__

*This is italic text*

_This is italic text_

~~Strikethrough~~
```

Rendered to:

Emphasis, also known as italics, uses an *asterisk* or *underline*.

Emphasis, also known as bold, uses an **asterisk** or **underline**.

Use a combination of **asterisk and *underline*** for emphasis.

Strikethrough uses two tildes. ~~Strike this out.~~

This is bold text

This is bold text

This is italic text

This is italic text

~~Strikethrough~~

The italic in this example is not displayed as italic style because the Xiawuwenkai GB font family used in this template does not contain italic style glyphs, not a template problem or a program error of typst. For details, see the instructions in Fonts.

5.3. Ordered lists

```
1. First ordered list item
2. Another item
  * Unordered list
1. The actual number doesn't matter, as long as it's a number
  1. The numbers will reorder
4. Another item.
   You can also properly indent paragraphs in list items.
5. Extra blank lines

   In addition to properly indenting paragraphs, adding an extra blank line before a paragraph also forms a new indented paragraph with an indented first line.

   In addition to properly indenting paragraphs, adding an extra blank line before a paragraph also forms a new indented paragraph with an indented first line.
7. One more item

1. Make changes
  1. Fix bugs
  2. Improve formatting
    - Make headings bigger
2. Push my commit to GitHub
3. Open a pull request
  * Describe my changes
  * Mention all members of my team
    * Ask for feedback
```

The ordered list above renders to:

---

1. First ordered list item

2. Another item

- Unordered list

   1. The actual number doesn't matter, as long as it's a number
   2. The numbers will reorder
   3. Another item. You can also properly indent paragraphs in list items.
   4. Extra blank lines

      In addition to properly indenting paragraphs, adding an extra blank line before a paragraph also forms a new indented paragraph with an indented first line.

      In addition to properly indenting paragraphs, adding an extra blank line before a paragraph also forms a new indented paragraph with an indented first line.

   5. One more item
   6. Make changes
   7. Fix bugs
   8. Improve formatting - Make headings bigger
   9. Push my commit to GitHub
   10. Open a pull request

- Describe my changes
- Mention all members of my team
   - Ask for feedback

---

pandoc also supports `fancy_lists`, use `#.` to replace numbers, pandoc will automatically calculate the sequence number, such as:

```
1. one
#. two
#. three
```

Renders to:

---

1. one
2. two
3. three

5.4. Unordered lists

```
* Unordered lists can be made with an asterisk (`*`)
- or a minus sign (`-`)
+ or a plus sign (`+`)

+ Create a list by starting a line with `+`, `-`, or `*`
+ Create a sublist by indenting 2 spaces:
  - A change in the marker character forces the start of a new list:
    * Ac tristique libero volutpat at
    + Facilisis in pretium nisl aliquet
    - Nulla volutpat aliquam velit
+ Very simple!

+ Unordered lists can also be
+ mixed with numbered lists
  1. Indent and create a numbered list item
  2. Another numbered item
+ Another unordered list item
```

Renders to:

- Unordered lists can be made with an asterisk (`*`)
- or a minus sign (`-`)
- or a plus sign (`+`)
- *Create* a list by starting a line with `+`, `-`, or `*`
- *Create* a sublist by indenting 2 spaces:
  - A change in the marker character forces the start of a new list:
    - Ac tristique libero volutpat at
    - Facilisis in pretium nisl aliquet
    - Nulla volutpat aliquam velit
- Very simple!
- Unordered lists can also be
- mixed with numbered lists
  1. Indent and create a numbered list item
  2. Another numbered item
- Another unordered list item

5.5. Task lists

Use GitHub-Flavored Markdown Syntax, use pandoc parameter `pandoc --from markdown+task_lists` or `pandoc -f gfm` to parse task lists.

```
- [x] Finish my changes
- [ ] Push my commits to GitHub
- [ ] Open a pull request
- [x] Support \@mentions, #refs, [links](), **formatting**, and <del>tags</
del>
- [x] Require list syntax (supports any unordered or ordered list)
- [ ] This is a complete project
- [ ] This is an incomplete project
- ✅ This line is not a task list item, but it is also a list item
```

Renders to:

- ☒ Finish my changes
- ☐ Push my commits to GitHub
- ☐ Open a pull request
- ☒ Support @mentions, #refs, links, formatting, and tags
- ☒ Require list syntax (supports any unordered or ordered list)
- ☐ This is a complete project
- ☐ This is an incomplete project
- ✅ This line is not a task list item, but it is also a list item

---

5.6. Ignore Markdown Formatting

You can tell GitHub to ignore (or enable escapes for) Markdown formatting by using before Markdown characters.

```
Let's rename \*our-new-project\* to \*our-old-project\*.
```

Let's rename *our-new-project* to *our-old-project*.

5.7. emojis

You can use emoji fonts, or use emoji markers

```
*emoji marks*: :smile: :rage: :sweat: :point_down:

*emoji symbols*: 😀😃😄😁🐶🐱🐭🐹
🍏🍐🍉🍇🍓
⚽🏐🏈🏀
🚗🚕🚙🚌
⌚ 📱 📲💻⌨
💗❤🧡💛🉐㊙㊗🈴🉑☢☣
🏳🏴☠🏁
⚠🌟🔥👍✅
```

Renders to:

---

emoji marks: 😄 😡 😓 👇

emoji symbols: 😀😃😄😁🐶🐱🐭🐹 🍏🍐🍉🍇🍓 ⚽🏐🏈🏀 🚗🚕🚙🚌 ⌚ 📱 📲💻⌨ 💗❤️🧡💛🉐㊙㊗🈴🉑☢️☣️ 🏳️🏴🏴‍☠️🏁 ⚠️🌟🔥👍✅

If some of the emoji fonts above are displayed as wireframe outline symbol style, it is not a problem with the typst template. It is caused by the font mechanism of typst, not a program error. For details, see the description in the Fonts section.

> pandoc parameter: `pandoc --from gfm` or `pandoc --from markdown+emoji`

5.8. Links

```
[I am an inline-style link](https://www.google.com)

[I am an inline-style link with a title](https://www.google.com "Google's
Homepage")

[I am a reference-style link][Any case-insensitive reference text]

[I am a relative reference to a repository file](../blob/master/LICENSE)

[You can use a number for a reference-style link definition][1]

Or leave it blank and use [the link text itself].
```

```
URLs and URLs in angle brackets are automatically converted to links. http://
www.example.com or <http://www.example.com>, and sometimes example.com (but
not on Github).

Some text indicating that the reference link can follow later.

[Any case-insensitive reference text]: https://www.mozilla.org
[1]: http://slashdot.org
[The link text itself]: http://www.reddit.com
```

Rendered to:

I am an inline-style link

I am an inline-style link with a title

I am a reference-style link

I am a relative reference to a repository file

You can use a number for a reference-style link definition

Or leave it blank and use the link text itself.

URLs and URLs in angle brackets are automatically converted to links. http://www.example.com or http://www.example.com, and sometimes example.com (but not on Github).

Some text indicating that the reference link can follow later.

5.9. Images

```
This is the logo:

Inline style:
![Alternative text](./userguide/images/icon48.png "Logo Title Text 1")

Quote style:
![Alternative text][Logo]

[Logo]: ./userguide/images/icon48.png "Logo Title Text 2"

No caption below the image, and the image is right next to the text:
![Minion](./userguide/images/minion.png){width=20%}
![Stormtroopocat](./userguide/images/stormtroopocat.png "The Stormtroopocat")
{width=20%}
```

```
You can also add a caption below the image, which needs to be a separate paragraph, that is, a blank line before and after:

![Image title, some explanation][id]

Define the image location in the quote later in the document:

[id]: ./userguide/images/dojocat.jpg "The Dojocat"

You can also specify the image size:

![Minion](./userguide/images/minion.png){width=5%} ![Minion](./userguide/images/minion.png){width=10%}
![Minion](./userguide/images/minion.png){width=20%}![Minion](./userguide/images/minion.png){width=30%}
```

Rendered to:

This is the logo:

Inline style:

Quote style:

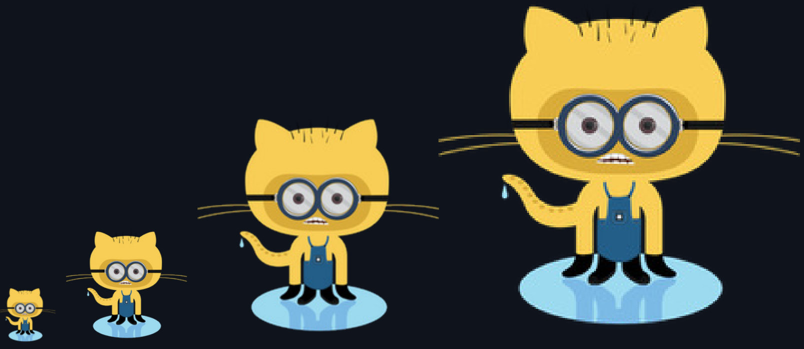
No caption below the image, and the image is right next to the text:

You can also add a caption below the image, which needs to be a separate paragraph, that is, a blank line before and after:

Figure 1: Image title, some explanation

Define the image location in the quote later in the document:

You can also specify the image size:



> 1. To add annotations below the image, use pandoc parameters:
>   `pandoc --from gfm+implicit_figures` or `pandoc --from markdown`
> 2. To control the image size, use pandoc parameters:
>   `pandoc --from gfm+attributes` or
>   `pandoc --from markdown+link_attributes`

When the image is a separate paragraph, the image is centered and the image size is set according to the page size. When the page width is less than 480 points, the image is flat without stretching. When it is an A4 page, the maximum width of the image does not exceed 11 cm. Inline images do not control the width.

5.10. Footnotes

The previous Indentation section used footnotes, which occupied the footnote number. Therefore, in the following markdown example, the numbering will continue from `2`, even if the example uses `Footnote 1 Link [^1]` The corresponding footnote number automatically continues from `2`, please pay attention to this detail.

```
Footnote 1 link [^1].

Footnote 2 link [^2].

~Inline footnote ^[text of inline footnote] definition.

Repeated footnote reference [^2].

[^1]: Footnotes **can have tags** and multiple paragraphs.

[^2]: Footnote text.
```

Renders to:

Footnote 1 link[2].

Footnote 2 link[3].

Inline footnote[4] definition.

Repeated footnote reference[5].

⚠ Warning

pandoc has a 10-year-old bug issue 1603 that cannot correctly parse repeated footnotes.

To correctly parse footnote syntax with inline references using pandoc, you need to use the following parameters.

[2] Footnotes **can have tags** and multiple paragraphs.

[3] Footnote text.

[4] text of inline footnote

[5] Footnote text.

```
pandoc --from markdown
# or
pandoc --from markdown_mmd+inline_notes
```

Footnotes in the form of inline references are only supported by pandoc's `--from markdown`, or `markdown_mmd` with the `inline_notes` option enabled.

5.11. Definition lists

```
Term
: The term name is on a separate line, and the second line starts with a colon
(`: `), followed by a description of the term definition

Another term
: It starts with a colon (`: `), followed by a description of the term
definition. *Definition lists* are often used in documents to unify the
definitions of professional terms, business terms, and nouns.

Term entry line break
: It is related to the line break interpretation of markdown. If it is `hard_
linkbreak`, a line needs to be left blank between entries.
This is not a definition term
: The definition term is not separated from the previous entry by a blank line
and cannot become a new entry.

Term definition paragraph
: The term content is indented to the right as a whole paragraph.
```

Renders to:

Term

> The term name is on a separate line, and the second line starts with a colon (`:`), followed by a description of the term definition

Another term

> It starts with a colon (`:`), followed by a description of the term definition. *Definition lists* are often used in documents to unify the definitions of professional terms, business terms, and nouns.

Term entry line break

It is related to the line break interpretation of markdown. If it is `hard_linkbreak`, a line needs to be left blank between entries. This is not a definition term

The definition term is not separated from the previous entry by a blank line and cannot become a new entry.

Term definition paragraph

The term content is indented to the right as a whole paragraph.

You need to use the pandoc parameter `--from markdown` or `--from gfm+definition_lists` to parse the definition list.

5.12. Mathematical Equation

When writing mathematical equation in pandoc markdown, use symbols such as `$$`, `\( \)`, or `\[ \]` to wrap the formula. There is also a GitHub flavored Markdown format.

```
Display on a single line
: $$
e=mc^2
$$

Inline
: This is a bunny, followed by a formula, $e=mc^2$. There can be text after
it.
```

Use pandoc parameter `pandoc --from gfm` or `pandoc --from markdown+tex_math_dollars` to render:

Display on a single line

$$e = mc^2$$

Inline

This is a bunny, followed by a formula, $e = mc^2$. There can be text after it.

If you want to use `pandoc --from gfm`, you can also support,

```
gfm-specific:
```

~~~
$`e=mc^2`$

``` math
e=mc^2
```
~~~

You can also use `pandoc --from markdown+tex_math_single_backslash` to parse the following format:

```
gfm cannot interpret:

\( e=mc^2 \)
\[ e=mc^2 \]
```

5.13. Code and syntax highlighting

For code highlighting color themes, see [Code highlight color themes]. Each theme is set in the corresponding pandoc tyspt template template file.

```
Inline `code` has `backticks` around it.
```

Inline `code` has `backticks` around it.

Code blocks use a pair of triple backticks ```

css sample:

~~~
```css
@font-face {
  font-family: Chunkfive; src: url('Chunkfive.otf');
}

body, .usertext {
  color: #F0F0F0; background: #600;
  font-family: Chunkfive, sans;
}

@import url(print.css);
@media print {
  a[href^=http]::after {
    content: attr(href)
  }
}
```
~~~

css highlighting effect:

```css
@font-face {
  font-family: Chunkfive; src: url('Chunkfive.otf');
}

body, .usertext {
  color: #F0F0F0; background: #600;
  font-family: Chunkfive, sans;
}

@import url(print.css);
@media print {
  a[href^=http]::after {
    content: attr(href)
  }
}
```

---

javascript sample:

~~~
```javascript
function $initHighlight(block, cls) {
  try {
    if (cls.search(/\bno\-highlight\b/) != -1)
      return process(block, true, 0x0F) +
             ` class="${cls}"`;
  } catch (e) {
    /* handle exception */
  }
  for (var i = 0 / 2; i < classes.length; i++) {
    if (checkCondition(classes[i]) === undefined)
      console.log('undefined');
  }
}

export  $initHighlight;
```
~~~

javascript highlighting effect:

```javascript
function $initHighlight(block, cls) {
  try {
```

```
    if (cls.search(/\bno\-highlight\b/) != -1)
      return process(block, true, 0x0F) +
             ` class="${cls}"`;
  } catch (e) {
    /* handle exception */
  }
  for (var i = 0 / 2; i < classes.length; i++) {
    if (checkCondition(classes[i]) === undefined)
      console.log('undefined');
  }
}

export  $initHighlight;
```

⚠ Warning

The typst `#highlight()` function does not control the line height as well as `#box()`, so typst ·#box()` is used in the template to implement inline code. If the inline code format is used, when the line needs to be broken beyond the end of the line, it cannot be split, which may cause too much white space in the previous line, or more than one line. At this time, you can adjust the description of the statement, or shorten the `` statement. No better implementation method has been found yet.

5.14. Tables

```
*Colons* can be used to align columns.

| Table         | is very       | Cool  |
| ------------- |:-------------:| -----:|
| Column 3 is   | Right-aligned | $1600 |
| Column 2 is   | Centered      | $12   |
| Zebra Stripes | Neat          | $1   |

There must be at least 3 dashes (`-`) between each header cell.

The outer pipes (`|`) are optional and you don't need them to make the raw
Markdown line up nicely. You can also use inline Markdown.

Table example, outer pipe characters (|) can be omitted:
```

```
Markdown | Less | Pretty
--- | --- | ---
*Still* | `Renders` | **Very well**
1 | 2 | 3

| First heading | Second heading |
| ------------- | -------------  |
| Content cell  | Content cell   |
| Content cell  | Content cell   |

Table example, cell content can be marked up:

| Command | Description |
| --- | --- |
| git status | List all new or modified files |
| git diff | Show differences in files that have not yet been staged |

| Command | Description |
| --- | --- |
| `git status` | List all *new or modified* files |
| `git diff` | Show differences in files that have not yet been staged |

Table example, columns can be freely aligned:

| Left alignment | Center alignment | Right alignment |
| :---           | :---:            | ---:            |
| git status     | git status       | git status      |
| git diff       | git diff         | git diff        |

Table example, special symbols in cells:

| Name     | Character |
| ---      | ---       |
| Backtick | `         |
| Pipe     | \|        |
```

Renders to:

Colons can be used to align columns.

Table	is very	Cool
Column 3 is	Right-aligned	$1600
Column 2 is	Centered	$12
Zebra Stripes	Neat	$1

There must be at least 3 dashes (`-`) between each header cell.

The outer pipes (`|`) are optional and you don't need them to make the raw Markdown line up nicely. You can also use inline Markdown.

Table example, outer pipe characters (|) can be omitted:

Markdown	Less	Pretty
Still	`Renders`	**Very well**
1	2	3

First heading	Second heading
Content cell	Content cell
Content cell	Content cell

Table example, cell content can be marked up:

Command	Description
git status	List all new or modified files
git diff	Show differences in files that have not yet been staged

Command	Description
`git status`	List all new or modified files
`git diff`	Show differences in files that have not yet been staged

Table example, columns can be freely aligned:

Left alignment	Center alignment	Right alignment
git status	git status	git status
git diff	git diff	git diff

Table example, special symbols in cells:

Name	Character
Backtick	`
Pipe	|

The table syntax above is called `pipe tables`. pandoc markdown also supports `simple_tables` and `grid_tables` styles, as well as multi-line cells and table captions. Use `pandoc --from markdown+simple_tables+grid_tables+multiline_tables+table_captions` to enable the extension.

5.14.1. Table title table caption

`table_captions` can be used to add captions to tables:

```
Column 1 | Column 2
---------|---------
first    | Second

: Define the caption after the table

table: Define the table caption before the table

Column 1 | Column 2
---------|---------
first    | Second
```

Start with `Table:`, `table:` or `:`, either before or after the table, remember to leave a blank line between the table and the caption. The table above is rendered as:

Table 9: Define the caption after the table

Column 1	Column 2
first	Second

Table 10: Define the table caption before the table

Column 1	Column 2
first	Second

5.14.2. simple tables

`simple_tables`, looks simple, but is actually more difficult to edit than pipe tables. It has strict format requirements

```
  Right Left     Center   Default
------- ------ ---------- -------
    12  12          12      12
   123  123        123      123
    1   1          1        1

Table: Show simple table syntax.
```

Use `-----` to separate the title and the cell. The relative position of the title and the `------` line is used to set the cell content to the left, center, or right. To be close to the actual layout, `-----`, title, and cell content are best aligned as needed. The table above is rendered as:

Table 11: Show simple table syntax.

Right	Left	Center	Default
12	12	12	12
123	123	123	123
1	1	1	1

5.14.3. Multi-line table

The syntax of `multiline_tables` is similar to `simple_tables`.

```
-------------------------------------------------------------
  Center   Default          Right Left
  Title    Aligned        Aligned Aligned
---------- ------- -------------- -------------------------
   First    row               12.0 Example of a row that
                                  spans multiple lines.

  Second    row                5.0 Here's another one. Note
                                  the blank line between
                                  rows.
-------------------------------------------------------------

Table: Here's the caption. It, too, may span multiple lines.
```

Format requirements for `multiline_tables`:

1. There is a `------` line above and below the table, and there must be a blank line after the table
2. The alignment of the title and the `-----` separator line is the same as `simple_tables`
3. Use a blank line to separate each row of content

The above cross-row table is rendered as:

Table 12: Here's the caption. It, too, may span multiple lines.

Center Title	Default Aligned	Right Aligned	Left Aligned
First	row	12.0	Example of a row that spans multiple lines.
Second	row	5.0	Here's another one. Note the blank line between rows.

5.14.4. grid tables

`grid_tables` is the most complex, but also the most functional table:

```
: grid table example

+---------------+---------------+--------------------+
| Fruit         | Price         | Advantages         |
+===============+===============+====================+
| Bananas       | $1.34         | - built-in wrapper |
|               |               | - bright color     |
+---------------+---------------+--------------------+
| Oranges       | $2.10         | - cures scurvy     |
|               |               | - tasty            |
+---------------+---------------+--------------------+

Cells can span multiple rows or columns:

+---------------------+----------+
| Property            | Earth    |
+=============+=======+==========+
|             | min   | -89.2 °C |
| Temperature +-------+----------+
| 1961-1990   | mean  | 14 °C    |
|             +-------+----------+
|             | max   | 56.7 °C  |
+-------------+-------+----------+

Table with multiple lines of headers:

+---------------------+-----------------------+
| Location            | Temperature 1961-1990 |
|                     | in degree Celsius     |
|                     +-------+-------+-------+
|                     | min   | mean  | max   |
+=====================+=======+=======+=======+
| Antarctica          | -89.2 | N/A   | 19.8  |
+---------------------+-------+-------+-------+
| Earth               | -89.2 | 14    | 56.7  |
+---------------------+-------+-------+-------+

Similar alignment syntax to *pipe tables*, use `:` to set the title separator:

+---------------+---------------+--------------------+
| Right         | Left          | Centered           |
+==============:+:==============+:==================:+
| Bananas       | $1.34         | built-in wrapper   |
+---------------+---------------+--------------------+

You can also use `===========` to set the table footer:

+---------------+---------------+
| Fruit         | Price         |
+===============+===============+
| Bananas       | $1.34         |
+---------------+---------------+
| Oranges       | $2.10         |
+===============+===============+
| Sum           | $3.44         |
+===============+===============+
```

Renders to:

Table 13: grid table example

Fruit	Price	Advantages
Bananas	$1.34	- built-in wrapper - bright color
Oranges	$2.10	- cures scurvy - tasty

Cells can span multiple rows or columns:

Property		Earth
Temperature 1961-1990	min	-89.2 °C
	mean	14 °C
	max	56.7 °C

Table with multiple lines of headers:

Location	Temperature 1961-1990 in degree Celsius		
	min	mean	max
Antarctica	-89.2	N/A	19.8
Earth	-89.2	14	56.7

Similar alignment syntax to pipe tables, use `:` to set the title separator:

Right	Left	Centered
Bananas	$1.34	built-in wrapper

You can also use `============` to set the table footer:

Fruit	Price
Bananas	$1.34
Oranges	$2.10
Sum	$3.44

`typst 0.12` Currently there is no way to get the last row of a table, so when the table footer is used, it is not possible to add styles to the table footer by typst rules.

5.15. alerts

alerts is a special style in GitHub-Flavored markdown. This template has been processed for compatibility. In addition to the note, tip, important, caution, and warning specified by gfm itself, it also supports idea, error, success, goal, and notification. Both uppercase and lowercase are acceptable.

```
> [!NOTE]
> This is a note

> [!TIP]
> This is a tip

> [!IMPORTANT]
> This is important

> [!CAUTION]
> Be careful with the problem described here

> [!WARNING]
> Warning
>


>[!Idea]
> This is an idea
>
>

> [!Error]
> This is an error
>

> [!Success]
> When successful

> [!Goal]
> This is the goal
>

> [!Notification]
> There is a notification here
>
```

Renders to:

> **📝 Note**
>
> This is a note

> **i Tip**
>
> This is a tip

> **❗ Important**
>
> This is important

> **🚧 Caution**
>
> Be careful with the problem described here

> **⚠ Warning**
>
> Warning

> **💡 Idea**
>
> This is an idea

> **❌ Error**
>
> This is an error

> **✅ Success**
>
> When successful

> **🏁 Goal**
>
> This is the goal

> **📣 Notification**
>
> There is a notification here

If the alerts style is too much, you can use a quoted text block instead:

```
> **Note:**
> Block quotes can be used to highlight important ideas.
```

Renders to:

> **Note:** Block quotes can be used to highlight important ideas.

When the text content in alerts exceeds one line or multiple paragraphs, you cannot see the indentation of the first line of the first paragraph. Currently, tyspt does not support this. See the instructions in Indentation for details.

5.16. Table width and column width

pandoc's `markdown+grid_tables` or `markdown+multiline_tables` supports specifying the table width and column width. When the table content is small and the page layout is unnatural, you can use grid tables or multiline tables to adjust it.

The table width and column width can be intervened respectively.

There are two factors affecting the table width:

1. pandoc `--columns` parameter value, the default is `72`
2. When defining a grid table, the number of characters between the `+` sign in the upper left corner of the top frame and the `+` sign in the upper right corner, that is, the total number of characters in the top frame, is the relative width of the table. Relative to the width of the text, its $relativewidth \approx totallength \div 72$, when $relativewidth > 1$, it is equal to 1. $actualwidth \approx relativelength \times textwidth$.

The table column width is also relative. The number of characters in the list title separator `====` or `---- (including trailing space)` is the ratio of the relative width of the entire table, which is the relative width of the column.

Take `grid_tables` as an example:

```
: 2 columns 72 characters

+-----------------------------------+----------------------------------+
| Fruit           Price             | Advantages                       |
+===================================+:=================================+
| Oranges         $2.10             | - cures scurvy                   |
|                                   | - tasty                          |
+-----------------------------------+----------------------------------+

: 3 columns 72 characters

+---------------+-------------------+----------------------------------+
| Fruit         | Price             | Advantages                       |
+===============+===================+:=================================+
| Oranges       | $2.10             | - cures scurvy                   |
|               |                   | - tasty                          |
+---------------+-------------------+----------------------------------+

: 2 columns 37 characters

+---------------+-------------------+
| Fruit         | Price             |
+===============+===================+
| Oranges       | $2.10             |
|               |                   |
+---------------+-------------------+
```

Renders to:

Table 18: 2 columns 72 characters

Fruit Price	Advantages
Oranges $2.10	• cures scurvy • tasty

Table 19: 3 columns 72 characters

Fruit	Price	Advantages
Oranges	$2.10	• cures scurvy • tasty

Table 20: 2 columns 37 characters

Fruit	Price
Oranges	$2.10

Take `multiline_tables` as an example:

```
: 4 columns, 72 characters

------------------------
 Centered  Default           Right Left
  Header   Aligned         Aligned Aligned
---------- ------- --------------- -------------------------------------
   First   row                12.0 Example of a row that
                                   spans multiple lines.
------------------------

: 3 columns, 34 characters

------------------------
 Centered  Default           Right
  Header   Aligned         Aligned
---------- ------- ---------------
   First   row                12.0
------------------------


: 2 columns 72 characters

------------------------
Centered   Default     Right         Left
Header     Aligned   Aligned         Aligned
-----------------------------        ------------------------------------
   First     row          12.0       Example of a row that
                                     spans multiple lines.
------------------------
```

Rendered to:

Table 21: 4 columns, 72 characters

Centered Header	Default Aligned	Right Aligned	Left Aligned
First	row	12.0	Example of a row that spans multiple lines.

Table 22: 3 columns, 34 characters

Centered Header	Default Aligned	Right Aligned
First	row	12.0

Table 23: 2 columns 72 characters

Centered Default Right Header Aligned Aligned	Left Aligned
First row 12.0	Example of a row that spans multiple lines.

5.17. Quotes

```
> Blockquotes are really handy in emails to simulate reply text.
This line is part of the same quote.

Quote separator.

> This is a really long line that will still quote correctly when it wraps.
Oh, let's keep writing it to make sure it's long enough to actually wrap for
each person. Oh, and you can *put* **Markdown** into a blockquote.

> Blockquotes can also be nested...
>> ...by using additional greater-than signs right next to each other...
> > > ...or with spaces between the arrows.
```

Renders to:

> Blockquotes are really handy in emails to simulate reply text. This line is part of the same quote.

Quote separator.

> This is a really long line that will still quote correctly when it wraps. Oh, let's keep writing it to make sure it's long enough to actually wrap for each person. Oh, and you can *put* **Markdown** into a blockquote.

> Blockquotes can also be nested…
>
>> …by using additional greater-than signs right next to each other…
>>
>>> …or with spaces between the arrows.

When the text in the quote exceeds one line or has multiple paragraphs, you cannot see the first line indent of the first paragraph. Currently tyspt does not support this. See Indentation for details.

5.18. alerts

alerts is a special style in GitHub-Flavored markdown. This template has been processed for compatibility. In addition to the note, tip, important, caution, and warning specified by gfm itself, it also supports idea, error, success, goal, and notification. Both uppercase and lowercase are acceptable.

```
> [!NOTE]
> This is a note

> [!TIP]
> This is a tip

> [!IMPORTANT]
> This is important

> [!CAUTION]
> Be careful with the problem described here

> [!WARNING]
> Warning
>

>[!Idea]
> This is an idea
>
>

> [!Error]
> This is an error
>
```

```
> [!Success]
> When successful

> [!Goal]
> This is the goal
>

> [!Notification]
> There is a notification here
>
```

Renders to:

> **📝 Note**
>
> This is a note

> **ℹ Tip**
>
> This is a tip

> **❗ Important**
>
> This is important

> **🚧 Caution**
>
> Be careful with the problem described here

> **⚠ Warning**
>
> Warning

> **💡 Idea**
>
> This is an idea

> **❌ Error**
>
> This is an error

✅ Success

When successful

🏁 Goal

This is the goal

📣 Notification

There is a notification here

If the alerts style is too much, you can use a quoted text block instead:

```
> **Note:**
> Block quotes can be used to highlight important ideas.
```

Renders to:

> **Note:** Block quotes can be used to highlight important ideas.

When the text content in alerts *exceeds one line or multiple paragraphs*, you cannot see the indentation of the first line of the first paragraph. Currently, tyspt does not support this. See the instructions in Indentation for details.

5.19. Horizontal Rule

```
Three symbols or more

Three hyphens `---`

---

Three asterisks `***`

***

Three underscores `___`

___
```

Three symbols or more

Three hyphens `---`

Three asterisks `***`

Three underscores `___`

5.20. Inline HTML

```
<dl>
<dt>Definition Lists</dt>
<dd>People sometimes use this. </dd>

<dt>Markdown in HTML</dt>
<dd>Doesn't work very well. Use the HTML <em>tag</em>. </dd>
</dl>
```

> 🚧 Caution
>
> The text content of inline html can only be displayed when using `pandoc -f markdown`, which can convert the markdown file to `typst` or convert it to `pdf` format document through `--pdf-engine typst` parameter, and falls back to displaying only text.

5.21. Title

The title presentation will affect the pdf directory structure and is placed at the end of the document.

```
Alt-H2
------

# (only for demonstration) h1 title
## h2 title
### h3 title
#### h4 title
##### h5 title
###### h6 title

Or, for H1 and H2, use the underline style:

(only for demonstration) Alt-H1
======

Alt-H2
------
```

Renders to:

5.22. Alt-H2

6. (only for demonstration) h1 title

6.1. h2 title

6.1.1. h3 title

6.1.1.1. h4 title

6.1.1.1.1. h5 title

6.1.1.1.1.1. h6 title

Or, for H1 and H2, use the underline style:

7. (only for demonstration) Alt-H1

7.1. Alt-H2
