Panovel

Contents

0.1 Introduction

The purpose of this little project is to make the process of producing a novel targeting different formats easier and more streamlined.

You just write the book in markdown, a very basic and easy to use markup language. You don't need a word processor for this, every editor will do. Then you just edit the metadata.yaml file with the options you want to use and execute the script. Voila: You have produced your book as an eBook and a pdf, ready to be printed! The main advantage of this method is, that you don't have to maintain different files for different formats.

0.1.1 How does this work?

This project is a wrapper around pandoc. It automates many steps that would be needed to convert the files manually and provides an example metadata.yaml file for easy configuration. In addition it provides many filters for pandoc to style some special elements.

0.1.2 Example

You can look at an example project here.

0.2 Installation

0.2.1 Prerequisites

Before you can use this template, you need some programs installed on your computer:

1. Python 3.6 or higher: You can get python here.
2. Pandoc: You can download pandoc here.
3. A latex distribution: If you want a pdf in addition to an epub, you need to install a latex engine. Depending on your operating system, this are some popular choices:

    - Windows: MiKTeX. I recommend the full webinstall. It is very big and takes quite some time, but you don't need to install any packages yourself this way.
    - MacOS: MacTeX
    - Linux: TeXLive via the package manager

   Additional packages for latex: If you didn't choose a complete latex distribution, you may have to install some additional packages in them. All of these distributions have a package manager you can use to download the needed packages. This template needs these three packages:

    - koma-script
    - verse
    - titletoc

   Pandoc needs some packages too. You can find the list in pandocs documentation (depending on the distribution, most or all could be installed already).

4. Kindlegen: If you want to convert your book for amazons kindle, you need this little tool from amazon. You can get it here.

0.2.2 Installing this script

Now you need to install this script itself. Open the console and enter:

```
pip3 install panovel
```

How to open the command prompt on windows: There are many ways to open the console on windows. Instead of the command prompt, you can use Powershell, if you like. The ways to access Powershell are mostly the same. Here are a few examples:

- Press the windows key and x. Now the power menu pops open. Select the command prompt from it.
- In the file explorer, enter `cmd` in the adressbar
- In the file explorer shift - right click on a folder and select `Open command window here`
- In the file explorer click on the `file` menu item and select `Open command prompt`. You can even add this command to

the menu toolbar with the right click option menu for more convinient use.

0.3 Starting a new project

Panovel can create a project folder for you, which includes anything you need to get started with your new project. Open the console in the folder you want to create your new project folder in. Then enter the following into the console:

```
panovel your_project_name
```

Per default, this creates a basic metadata.yaml file. If you need all available options, you can instead enter:

```
panovel your_project_name -c
```

Of course you can always create the full metadata file later, by opening the console in the project folder and entering:

```
panovel -c
```

0.3.1 Structure

All markdown files in the folder are used to create the book. You can therefor either have your book in just one file or in different files. All files are joined alphabetically or you can specify the exact sequence in the metadata file. In addition you can specify the files containing the front- and backmatter.

The image folder holds all images used, for example the cover for the epub/mobi file.

0.4 The command line interface

Panovel has a small list of additional options available for more advanced uses. Enter

```
panovel -h
```

in the console, to see all options available.

0.5 Formating

0.5.1 Usage

Markdown is used to format the book. You can use all functionality provided by pandocs markdown. But while markdown excels at light markup, sometimes you need special formatting. Poems, epigraphs etc. are good examples. To make formatting these elements easy, this template comes with an easy to use syntax for many such elements. Each of this styles comes with one or more filters, which formats these elements according to the output format.

The syntax is very easy:

```
~~~{.type .option key=value}
The content that should be styled.
~~~
```

Types and simple options are preceded by a dot, attributes consist of a name and a value. These options are separated by a simple space. If the value of an attribute contains spaces, you need to surround it with quotation marks.

For example, this is how you would implement a poem:

```
~~~{.poem author="William Shakespeare" title="Sonnet 18"}
Shall I compare thee to a summer's day?
Thou art more lovely and more temperate:
Rough winds do shake the darling buds of May,
And summer's lease hath all too short a date:

Sometime too hot the eye of heaven shines,
And often is his gold complexion dimmed,
And every fair from fair sometime declines,
By chance, or nature's changing course untrimmed:

But thy eternal summer shall not fade,
Nor lose possession of that fair thou ow'st,
Nor shall death brag thou wand'rest in his shade,
When in eternal lines to time thou grow'st,

|       So long as men can breathe or eyes can see,
|       So long lives this, and this gives life to thee.
~~~
```

You can find a description of all the provided styles at the end of this documentation.

0.5.2 Some additional Tips

You can use html style comments in your source files. But not inside other style elements. Such a comment looks like this:

```
<!-- This is a comment. -->
```

You can add a curly bracket with ".unnumbered" or "-" inside it at the end of a heading, to exclude it from the automatic numbering, if you chose to use that.

```
# Example Heading{.unnumbered}
# Example Heading{-}
```

0.6 Configuration

You configure how your novel is build through the "metadata.yaml" file. It consists of fields you enter your data into. Each field starts with its name and a colon. Your data comes behind the colon. Many fields have example values - just overwrite them.

Some fields are lists and can take multiple values. List items begin in the next line, indented and preceded by a dash. Just orientate yourself on the examples.

Lines beginning with a '#' are comments. They mostly explain what the field controls. Sometimes the actual field is commented out, which means it is ignored. That is an easy way to enable a feature. Just remove the '#'.

The metadata file should be rather self explanatory. There are only a couple of things you have to add to make it work:

- The title of the book
- The author
- The path to kindlegen, if you want to convert for kindle

If you have frontmatter and/or backmatter in seperate files, remember to add the filenames to the corresponding field.

0.7 Building the books

On windows you can simply use the provided bat file. Just double click it and your book is build. Otherwise the script is invoked from the command line. Just enter:

```
panovel
```

0.7.1 For advanced users

panovel accepts command line arguments. These overwrite the corresponding settings in the metadata.yaml. Just invoke it with the -h argument to see the options available.

0.8 List of styles

Below you find a description of each type provided with the template and which filter is used to render it. If no filter is given, it is named like the style. Some styles have options how they should look in the rendered book. You can change these in the metadata.yaml file.

0.8.1 align

Needs a second qualifier, one of '.center', '.right' or '.left'.

Text gets aligned accordingly inside the block.

0.8.2 copyrights

You need your copyrights surrounded by this style to be recognized properly.

0.8.3 dedication

You need your dedication surrounded by this style to be recognized properly.

0.8.4 epigraph

Adds an epigraph.

Attributes:

author The author of the epigraph

0.8.5 new_scene

This style is an exception to all others. To make a new scene, just enter three '*' on one line surrounded by blank lines. Please note, that this overrides one default markdown command for a horizontal line. Just use `---` for horizontal lines.

Example:

```
The last paragraph in the old scene.

* * *

The first paragraph in the new scene.
```

Render Options:

default A new scene is marked by blank space
text A new scene is marked by the text given, defaults to `* * *`
fleuron A new scene is marked by an image. You need to provide an image in the options

0.8.6 noindent

The paragraph is not indented.

0.8.7 poem

Styles the content as a poem. There are three render options for this style, each just formatting the poem slightly different.

Render Options:

bottom Title and author are positioned on the bottom right of the poem

one-line As above, but title and author appear on one line
top This filter inserts the title at the top of the poem

Options:

altverse This option indents every other line in a stanza
top Positions the title on the top, regardless of the render option used
bottom Positions the title on the bottom, regardless of the render option used
one-line Positions author and title in one line on the bottom, regardless of the render option used
noversewidth Per default the filter calculates the best position for the poem. If you don't want this, use this option

Attributes:

author The author of the poem
title The title of the poem
poemlines Needs a number. Adds a line number every number of lines
versewidth Needs a line of the poem, used to position the poem. Normally this is calculated automatically, use this, if you want to set it manually

0.8.8 quote

Styles the content as a quote. Of course markdown has its own syntax for quotes. You only need this style, if you want to include an author and/or a title.

There are three render options for this style, each just formating the quote slightly different.

Render Options:

bottom Title and author are positioned on the bottom right of the quote
one-line As above, but title and author appear on one line
top This filter inserts the title at the top of the quote

Options:

top Positions the title on the top, regardless of the render option used

bottom Positions the title on the bottom, regardless of the render option used

one-line Positions author and title in one line on the bottom, regardless of the render option used

Attributes:

author The author of the quote

title The title of the quote

0.9 Filter

Here are some additional filters, not used for styling elements.

0.9.1 epub_remove_fm_head

This filter is used for a technical issue. Just keep it active.

0.9.2 header_fleuron

Adds an image beneath each heading. You need to provide an image in the options.

0.9.3 remove_comments

You can use html comments to annotate your text. This filter removes this comments from the output.

0.9.4 remove_images

Removes all images from the output.