

# Document Typesetting Template Readme – Draft



June 16, 2020

## CONTENTS

<b>1</b>	<b>Preamble</b>	<b>3</b>
<b>2</b>	<b>Introduction</b>	<b>3</b>
<b>3</b>	<b>Quick Start</b>	<b>4</b>
3.1	Setup . . . . .	4
3.2	Usage . . . . .	5
3.2.1	Markdown Files . . . . .	5
3.2.2	Google Docs Links . . . . .	5
3.2.3	Miscellaneous Document Files . . . . .	6
<b>4</b>	<b>Modifying the Template</b>	<b>6</b>
<b>5</b>	<b>Further Reading</b>	<b>6</b>

## 1 PREAMBLE

The text above is a “document preamble.” It looks pretty normal within the `README.md` Markdown file, and it is invisible in the `README.pdf` generated by Pandoc. However it is sometimes visible as a table in the GitHub Markdown view for the repository. In Markdown, the YAML-formatted preamble looks like:

```
---  
title: Document Typesetting Template Readme -- Draft  
author: Genrep Software, LLC.  
date: June 16, 2020  
---
```

If reading the PDF, notice that the above has fancy syntax highlighting. Pandoc does this for us! Also note that three variables are set: `title`, `author`, and `date`. All of these must be set for the document to compile properly. If you are converting a non-Markdown document and cannot set the metadata values, run Pandoc (or the compilation script) with the `-M` or `--metadata`<sup>1</sup> option as follows, for example:

```
pandoc -M title:"Fancy title" --metadata author:"Genrep" README.md -o README.tex
```

Note that this technique of manually including metadata is particularly useful for compiling `.docx` Microsoft Word documents, which may have been exported and downloaded from Google Docs. Look in the `compile.sh` script for an example of how this is used. Read the Usage section to learn about how to properly compile `.docx` documents using the compilation script.

Unfortunately, even though Markdown syntax is similar across platforms that use it, there is no official standard. When writing Markdown documents to version control and eventually typeset using this template and script, refer to the Pandoc Markdown Reference.<sup>2</sup>

## 2 INTRODUCTION

Genrep Software specializes in, among other things, report generation. As a result, it is absolutely critical that the documents put out by the company look clean and professional. Using an advanced typesetting system is a good way to rapidly create professional-looking documents with highly consistent formatting. Furthermore, scripting the process of typesetting documents aligns with our mission of automating report generation.

For typesetting, we primarily use  $\text{\LaTeX}$  (LaTeX).  $\text{\LaTeX}$  is an advanced document typesetting system, originally created decades ago by famed computer scientist Donald Knuth. Pandoc compiles

<sup>1</sup>These two options are completely interchangeable. `-M` makes more sense for use in the terminal, whereas `--metadata` makes much more sense for scripts where readability is a concern.

<sup>2</sup><https://pandoc.org/MANUAL.html#pandocs-markdown>

Markdown to  $\LaTeX$  for us, so that we don't have to learn and remember the fancy  $\LaTeX$  commands. That being said, it is possible to inline  $\LaTeX$  commands. For example, if reading in the PDF, it will not be obvious that the weird-looking word  $\LaTeX$  is created by typing `\LaTeX{}` into the Markdown. GitHub renders the Markdown without converting it to  $\LaTeX$ , so it will look like raw commands, rather than nice formatting there.

One benefit of using  $\LaTeX$  is that we can also inline math, which will be beautifully typeset for us. The nice math below will not render correctly on GitHub unfortunately. To see how nice the math looks, view the `README.pdf` file in the repository.

$$\text{Borwein}(n) := \int_0^\infty \prod_{i=0}^n \frac{\sin(x)}{x} dx$$

The code to render the math above is:

```
$$ \mathrm{Borwein} (n) := \int_0^\infty \prod_{i = 0}^n \frac{\sin (x)}{x} dx $$
```

## 3 QUICK START

### 3.1 SETUP

Make sure that you have Pandoc and  $\LaTeX$  installed locally. There are other ways to compile `.tex` files, such as Overleaf,<sup>3</sup> but for now, it is easiest to have a local distribution running. Installation instructions can be found at the following links (respectively):

- <https://pandoc.org/installing.html>
- <http://www.tug.org/texlive/acquire-netinstall.html>

Once these are set up, clone the repository for access to the compilation script, logo image, and document template.

```
git clone https://github.com/Genrep-Software/document-template.git
cd document-template
```

To test your Pandoc/ $\LaTeX$  installation, try compiling the `README.md` file from within the `document-template` repository directory. Make sure the resulting PDF file is viewable.

```
./compile.sh README.md
```

---

<sup>3</sup><https://www.overleaf.com/>

## 3.2 USAGE

Using the template is just a matter of cloning the repository and running the `compile.sh` script. If there is no `logo.png` or `template.tex` in the working directory where the script is called, it will automatically pull the latest versions from the GitHub repository.

### 3.2.1 MARKDOWN FILES

Pandoc-flavored Markdown files can contain metadata (title, author, and date) in YAML format.<sup>4</sup> This data is used while compiling the document, and having a `title` field, an `author` field, and a `date` field is *absolutely required*. If any of these fields is not included, the document may be formatted strangely, and  $\LaTeX$  may give cryptic errors. When debugging this script and the template, the first thing to check is whether the test case has the proper metadata. This script assumes that if a Markdown file is being typeset, it will have data inside.

To compile a Markdown file with YAML metadata inside the file, use:

```
./compile.sh <filename to convert (ending in .md)>
```

### 3.2.2 GOOGLE DOCS LINKS

For comiling Google Docs links, it is necessary to include the URL and a title. There are also optional `author` and `date` arguments, but “Genrep Software, LLC.” and the date of compilation will be used if they are not provided. Make sure to surround the command-line arguments with quotation marks.

Also ensure that any Google Docs link used has the correct sharing settings. In particular, it must be set so that “Anyone with the link can view,” that way this script can access the link to export the document. If the sharing permissions are set incorrectly, there may be cryptic errors due to  $\LaTeX$  trying to compile a mostly empty document.

The general usage of the command for Google Docs links is:

```
./compile.sh <Google Docs link> "<title>" "[author]" "[date]"
```

In particular, since the `author` and `date` arguments are optional, we can do:

```
./compile.sh \  
  "https://docs.google.com/document/d/1w-AzMzVo05u5aeyMHdYXJei0wdfGFMCQxZ2Ks7_QWw8/edit" \  
  "Old To-do List"
```

This will generate a nicely-formatted, typeset file called `Old To-do List.pdf`.

---

<sup>4</sup><https://pandoc.org/MANUAL.html#metadata-blocks>

### 3.2.3 MISCELLANEOUS DOCUMENT FILES

To compile any other type of file, it is necessary to manually include the title, author, and date metadata. This can be done by specifying them as command-line arguments for the compilation script in that order. It is very important that the arguments be surrounded by quotation marks if they have spaces in them. For example:

```
./compile.sh proposal.docx "This is the title" "Genrep Software, LLC" "4/20/69"
```

The general usage of the command for non-Markdown files is:

```
./compile.sh <input file> "<title>" "<author>" "<date>"
```

The compilation script will throw an error and fail if it is given a non-Markdown file with too few arguments. If, in the generated PDF, the displayed title is only a single word and/or the author or date is only a single word, make sure to surround the command-line arguments with quotation marks.

## 4 MODIFYING THE TEMPLATE

Modifications to the template may be required for aesthetic reasons, or if there are compilation errors due to necessary packages not being imported. If there are errors because of packages, consult the `pandoc-default-template.tex` file included in this repository. The file was generated by running

```
pandoc -D latex > pandoc-default-template.tex
```

The file will include various packages based on internal Pandoc variables that are set during document compilation. Do some digging in there based on the compilation error and see if it is possible to determine what the missing package is. Google is your friend here.

## 5 FURTHER READING

This document primarily contains information specific to this script and template. To learn more about Pandoc-flavored Markdown, or to contribute effectively to this project, click the links below.

- <https://pandoc.org/MANUAL.html>
- <https://www.tug.org/begin.html>