The Bookless manual

PlainLab

2022

Contents

The Bookless book **1**
A book written in Bookless app 1

Preface **3**

1 Introduction **5**
1.1 Motivation 5
1.1.1 Distraction-free mode 5
1.1.2 Chapters explorer 5
1.1.3 Local plain text files 5
1.1.4 Nice by default 5
1.2 Get started 6
1.3 Usage 6
1.3.1 Chapters explorer 6
1.3.2 Chapter content 6

2 Components **9**
2.1 Markdown syntax 9
2.1.1 Inline formatting 9
2.1.2 Block-level elements 10
2.1.3 Math expressions 11
2.2 Figures 12
2.3 Tables 13

3 Output formats **15**
3.1 HTML 15
3.2 PDF 15
3.3 ePub 15

4 Customization **17**
4.1 Preferences 17
4.2 Themes 17
4.3 Custom CSS 17

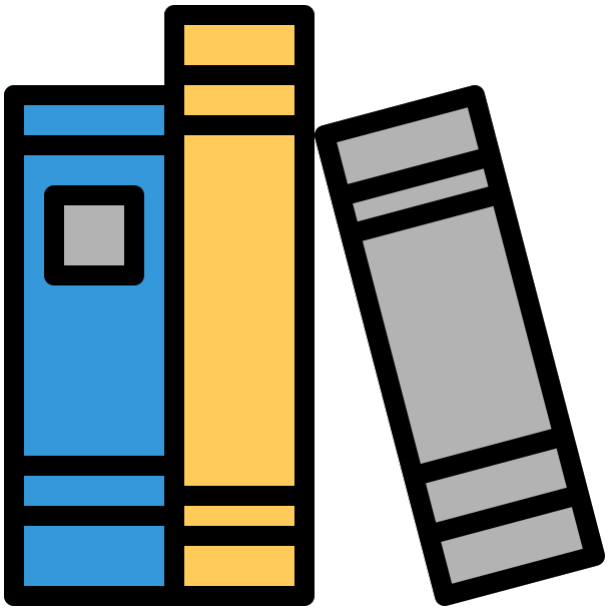
5 Editing **19**
5.1 Build the book 19
5.2 Preview 19
5.3 Spell check 19

6 Publishing **21**
6.1 GitHub Pages 21
6.2 Gumroad 21

FAQ **23**
Q: When will this book finish? 23

References **25**

The Bookless book



A book written in Bookless app

- Source: https://github.com/bookless/bookless.github.io
- HTML: https://bookless.github.io
- PDF: https://bookless.github.io/book.pdf
- ePub: https://bookless.github.io/book.epub

Preface

I love reading books, especially beautiful books. Everytime I read a nicely formatted book, I told myself I would want to make a book that looks exactly like this. But I'm yet to find an app that allows me to do that. Alright, now I'm writting it myself and call it Bookless!

This is the first book written using Bookless, about how to use it to write a book just like this one.

Bookless is built on top of Pandoc. Hence many parts of this book will come from Pandoc manual. You should check them out if you want to build a writing app. Still, this book is about how to write a book, so let's get started!

Chapter 1

Introduction

1.1 Motivation

Writing a book is hard. It's harder if we don't have a powerful enough tool for it.

1.1.1 Distraction-free mode

When onto writing something, I want to turn on the full focus mode. Concentrate only to what I am thinking and writing at the moment. Nothing else. That the first thing I want when creating Bookless: the distraction-free mode.

1.1.2 Chapters explorer

There is rarely anyone who are writing a one-chapter book. I love to organize things for the sake of self-sanity. So my book should be divided into chapters. That the second things Bookless must have: the chapters explorer.

1.1.3 Local plain text files

Plain text files are kings. Of course they are! Not every computers come with a pre-installed expensive Microsoft Word. But I assure you, every single one can edit a plain text file! So Bookless book's chapters are, not very surprisingly, just plain text files on your computer.

1.1.4 Nice by default

Every book should be beautifully formatted. That's not even for debate. But not every one can do that. So Bookless should come with some nice-looking pre-defined templates so every authors can just really get started quickly.

1.2 Get started

Just download Bookless from its GitHub, and then install it to your computer. Get started with 3 simple steps:

- Open the Bookless app
- Choose a folder to working on
- Create a new file and start writing

1.3 Usage

1.3.1 Chapters explorer

A typical book contain many chapters. They will be displayed on your left bar. Click the bookmark icon to expand it if it's hidden. Click again to enter focus mode. You can double click on the top area to toggle the mode if you like.

All chapters belong to the book will be shown on the left. You can drag and drop to re-order them. Each chapter is a Markdown file on your choosen working folder. You can rename or delete them as you see fit. Just be aware that the delete action is permanent, there is no recycle bin for you to look for deleted files.

1.3.2 Chapter content

Each chapter file must start immediately with the chapter title using the first-level heading, e.g., `# Chapter Title`. All Markdown files must be encoded in UTF-8, especially when they contain multi-byte characters such as Chinese, Japanese, and Korean. Here is an example (the bullets are the filenames, followed by the file content):

- preface.md

```
# Preface {-}

In this book, we will introduce an interesting
method.
```

- intro.md

```
# Introduction

This chapter is an overview of the methods that
we propose to solve an **important problem**.
```

- method.md

```
# Methods
```

```
We describe our methods in this chapter.
```

- application.md

```
# Applications

Some _significant_ applications are demonstrated
in this chapter.

## Example one

## Example two
```

- summary.md

```
# Final Words

We have finished a nice book.
```

The content are automatically saved so you don't have to. You should put your book on a backup system like Dropbox or iCloud to prevent any unwanted accident that may happen to your computer.

Chapter 2

Components

This chapter demonstrates the syntax of common components of a book written in **Bookless**. The approach is based on Pandoc, so we start with the syntax of Pandoc's flavor of Markdown.

2.1 Markdown syntax

In this section, we give a very brief introduction to Pandoc's Markdown. Readers who are familiar with Markdown can skip this section. The comprehensive syntax of Pandoc's Markdown can be found on the Pandoc website.

2.1.1 Inline formatting

You can make text *italic* by surrounding it with underscores or asterisks, e.g., `_text_` or `*text*`.

For **bold** text, use two underscores (`__text__`) or asterisks (`**text**`).

Text surrounded by `~` will be converted to a subscript (e.g., `H~2~SO~4~` renders H_2SO_4), and similarly, two carets (`^`) produce a superscript (e.g., `Fe^2+^` renders Fe^{2+}).

To mark text as `inline code`, use a pair of backticks, e.g., `` `code` ``.[1]

Small caps can be produced by the HTML tag `span`, e.g., `<span style="font-variant:small-caps;">Small Caps</span>` renders Small Caps.

Links are created using `[text](link)`, e.g., `[RStudio](https://www.rstudio.com)`, and the syntax for images is similar: just add an exclamation mark, e.g., `![alt text or image title](path/to/image)`.

[1] To include literal backticks, use more backticks outside, e.g., you can use two backticks to preserve one backtick inside: `` `` `code` `` ``.

Footnotes are put inside the square brackets after a caret `^[]`, e.g., `^[This is a footnote.]`.

2.1.2 Block-level elements

Section headers can be written after a number of pound signs, e.g.,

```
# First-level header

## Second-level header

### Third-level header
```

If you do not want a certain heading to be numbered, you can add `{-}` after the heading, e.g.,

```
# Preface {-}
```

This may cause the chapter doesn't have correct chapter name on the page header. You need to add a TeX command to an unnumbered header to fix this:

```
# Preface {-}
\markboth{}{}
```

Unordered list items start with `*`, `-`, or `+`, and you can nest one list within another list by indenting the sub-list by four spaces, e.g.,

```
- one item
- one item
- one item
    - one item
    - one item
```

The output is:

- one item
- one item
- one item
    - one item
    - one item

Ordered list items start with numbers (the rule for nested lists is the same as above), e.g.,

```
1. the first item
2. the second item
3. the third item
```

The output does not look too much different with the Markdown source:

1. the first item
2. the second item

3. the third item

Blockquotes are written after >, e.g.,

```
> "I thoroughly disapprove of duels. If a man should challenge me,
  I would take him kindly and forgivingly by the hand and lead him
  to a quiet place and kill him."
>
> --- Mark Twain
```

The actual output (we customized the style for blockquotes in this book):

> "I thoroughly disapprove of duels. If a man should challenge me, I would take him kindly and forgivingly by the hand and lead him to a quiet place and kill him."
>
> — Mark Twain

Plain code blocks can be written after three or more backticks, and you can also indent the blocks by four spaces, e.g.,

````
```
This text is displayed verbatim / preformatted
```
````

```
Or indent by four spaces:

    This text is displayed verbatim / preformatted
```

2.1.3 Math expressions

Inline LaTeX equations can be written in a pair of dollar signs using the LaTeX syntax, e.g., `$f(k) = {n \choose k} p^{k} (1-p)^{n-k}$` (actual output: $f(k) = \binom{n}{k} p^k (1-p)^{n-k}$); math expressions of the display style can be written in a pair of double dollar signs, e.g., `$$f(k) = {n \choose k} p^{k} (1-p)^{n-k}$$`, and the output looks like this:

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k}$$

You can also use math environments inside `$ $` or `$$ $$`, e.g.,

```
$$\begin{array}{ccc}
x_{11} & x_{12} & x_{13}\\
x_{21} & x_{22} & x_{23}
\end{array}$$
```

$$\begin{array}{ccc} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{array}$$

```
$$X = \begin{bmatrix}1 & x_{1}\\
1 & x_{2}\\
1 & x_{3}
\end{bmatrix}$$
```

$$X = \begin{bmatrix}1 & x_{1}\\ 1 & x_{2}\\ 1 & x_{3}\end{bmatrix}$$

```
$$\Theta = \begin{pmatrix}\alpha & \beta\\
\gamma & \delta
\end{pmatrix}$$
```

$$\Theta = \begin{pmatrix}\alpha & \beta\\ \gamma & \delta\end{pmatrix}$$

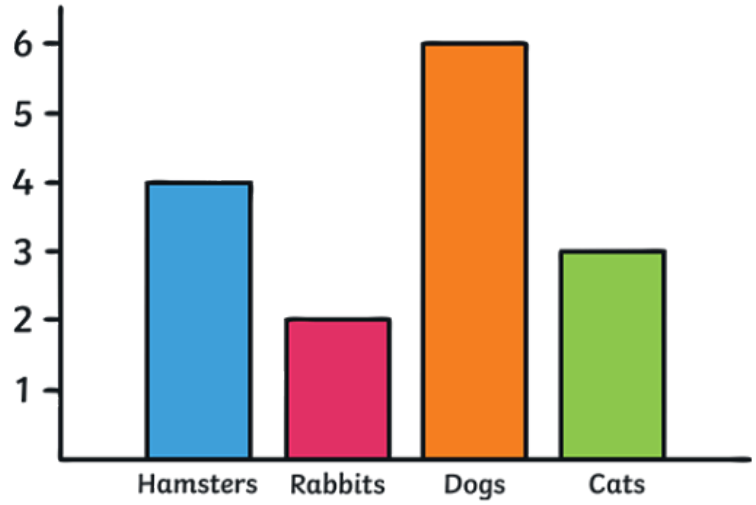
```
$$\begin{vmatrix}a & b\\
c & d
\end{vmatrix}=ad-bc$$
```

$$\begin{vmatrix}a & b\\ c & d\end{vmatrix}=ad-bc$$

2.2 Figures

```
![Bar demo](assets/CYYL-JyC-14PQ-wVnw4Rg.png)\
```

Generate this image:



If you want to create a figure, remove the last backslash:

```
![Chart demo](assets/h-ktH6YHH9w015ilzgNfs.png)
```

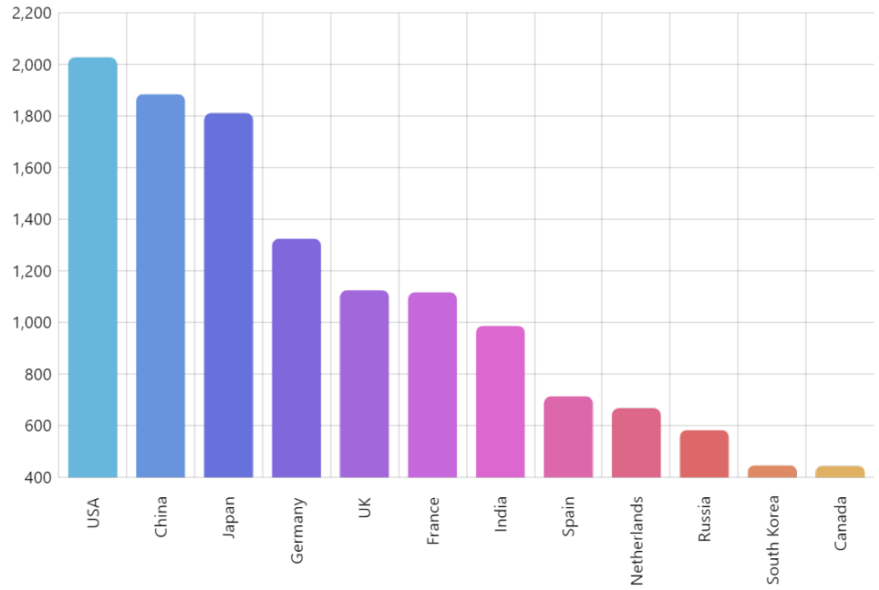


Figure 2.1: Chart demo

2.3 Tables

Table: A simple table in Markdown.

```
|Sepal.Length| Sepal.Width | Petal.Length | Petal.Width |
|------------|-------------|--------------|-------------|
|        5.1 |          3.5|           1.4|          0.2|
|        4.9 |          3.0|           1.4|          0.2|
|        4.7 |          3.2|           1.3|          0.2|
|        4.6 |          3.1|           1.5|          0.2|
|        5.0 |          3.6|           1.4|          0.2|
|        5.4 |          3.9|           1.7|          0.4|
```

Will generate below table.

Table 2.1: A simple table in Markdown.

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
5.1	3.5	1.4	0.2
4.9	3.0	1.4	0.2
4.7	3.2	1.3	0.2
4.6	3.1	1.5	0.2
5.0	3.6	1.4	0.2
5.4	3.9	1.7	0.4

Chapter 3

Output formats

Bookless supports HTML / PDF / ePub exports using embedded Pandoc.

3.1 HTML

You can export Bookless books into a single HTML file using the HTML export option.

3.2 PDF

You need to install Latex on your system before using this feature.

3.3 ePub

To be written…

Chapter 4

Customization

4.1 Preferences

Bookless allows you to customize these book preferences:

- Title
- Author
- Date
- Language
- Font family
- Font size
- Line height

All configuration will be saved on *_bookless.yaml* file. This is a valid Pandoc defaults file so you can use Pandoc command line to generate the book without Bookless.

4.2 Themes

To be written…

4.3 Custom CSS

To be written…

Chapter 5

Editing

5.1 Build the book

You can export either the whole book or each chapter using Share icons.

5.2 Preview

To be written…

5.3 Spell check

To be written…

Chapter 6

Publishing

To be written…

6.1 GitHub Pages

6.2 Gumroad

FAQ

Q: When will this book finish?

A: It would never be.

References

- Pandoc manual
- The Bookdown book