

---

# Template Device

User Manual

Martin Jäger

2019-06-27



LIBRESOLAR

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Usage</b>	<b>3</b>
2.1	Folder structure . . . . .	3
2.2	Pandoc metadata . . . . .	3
2.3	Markdown input . . . . .	3
2.4	HTML generation . . . . .	3
2.5	PDF generation . . . . .	4
<b>3</b>	<b>Math rendering support</b>	<b>4</b>
<b>4</b>	<b>Code highlighting</b>	<b>4</b>

## 1 Introduction

This document gives an example of how the Markdown Manual Template can be used.

The template expects normal markdown (with pandoc extensions) as an input and generates a nice static HTML 5 website and a PDF document.



**Figure 1:** The Creative Commons Attribution-ShareAlike logo

## 2 Usage

### 2.1 Folder structure

Everything you need is stored in the `manual` subfolder. This makes sure it is easy to integrate into existing projects. Just copy the folder from this example to your own repository.

### 2.2 Pandoc metadata

Some advanced configuration for pandoc is stored in the file `metadata.txt`.

### 2.3 Markdown input

The markdown file names should be numbered according to the order in the final document. Header numbers are generated automatically by pandoc and should not be added in the markdown files.

### 2.4 HTML generation

The HTML template is based on the great mdBook theme, which was simplified and adjusted a bit to suit the needs of a manual.

In the `manual` subfolder call:

```
pandoc metadata.txt *.md -o manual.html --template template/mdbook.html --from markdown --listings --number-sections --toc --toc-depth=2 --katex
```

## 2.5 PDF generation

PDF files are generated using LaTeX, so a working LaTeX engine needs to be installed on your system already (e.g. texlive). The manual uses the Eisvogel template.

In the `manual` subfolder call:

```
pandoc metadata.txt *.md -o manual.pdf --template template/eisvogel.tex --from markdown --listings --number-sections --toc
```

## 3 Math rendering support

Pandoc supports math rendering in HTML using different engines. We recommend to use the KaTeX engine, specified by the `--katex` option on the command-line.

In-line math equations like  $U = R \cdot I$  must be wrapped within two `$`-signs, without a space after the starting `$` and without a space before the ending `$`.

Math equations in a separate line are defined using two `$$`. They are rendered like this:

$$\rho \dot{\vec{v}} = \rho \left( \frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \nabla) \vec{v} \right) = -\nabla p + \mu \Delta \vec{v} + (\lambda + \mu) \nabla (\nabla \cdot \vec{v}) + \vec{f}.$$

## 4 Code highlighting

Here is an example of highlighted C++ code:

```
1  int main()
2  {
3      printf("Time: %d\n", time(NULL));
4  }
```