



simplivre

以极简主义风格排版你的图书

对应版本. simplivre 2021/05/23

许锦文

2021 年 5 月，北京



前言

- 1 simplivre 是 minimalist 文档类系列的成员之一, 其名称取自于法文的 simple (“简约”), 以及
- 2 livre (“书”), 由二者组合而成。整个 minimalist 系列包含用于排版文章的 minimart、einfart
- 3 以及用于排版书的 minimbook、simplivre。我设计这一系列的初衷是为了撰写草稿与笔记,
- 4 使之看上去简朴而不简陋。
- 5 simplivre 支持英文、法文、德文、中文、日文、俄文六种语言, 并且同一篇文档中这些语言
- 6 可以很好地协调。由于采用了自定义字体, 需要用 XeLaTeX 或 LuaLaTeX 引擎进行编译。
- 7 这篇说明文档即是用 simplivre 排版的, 你可以把它看作一份简短的说明与演示。

提示

多语言支持、定理类环境、未完成标记等功能是由 [ProjLib](#) 工具箱提供的, 这里只给出了将其与本文档类搭配使用的要点。如需获取更详细的信息, 可以参阅 [ProjLib](#) 的说明文档。



目录

I 说明

| | | |
|---|-------------|----|
| 1 | 初始化 | 3 |
| 1 | 1 如何载入 | 3 |
| 2 | 2 选项 | 3 |
| 2 | 关于文档类中使用的字体 | 5 |
| 3 | 使用说明 | 7 |
| 1 | 1 语言设置 | 7 |
| 2 | 2 定理类环境及其引用 | 8 |
| 3 | 3 定义新的定理型环境 | 8 |
| 4 | 4 未完成标记 | 10 |
| 5 | 5 关于行号 | 10 |
| 6 | 6 关于标题中的脚注 | 10 |
| 7 | 7 目前存在的问题 | 10 |

II 演示

| | | |
|-----|-------------------------------------|----|
| 4 | 文档示例 | 13 |
| 1 | 1 标准文档类写法 | 13 |
| 2 | 2 \mathcal{AMS} 文档类写法 | 14 |
| 5 | Heading on Level 0 (chapter) | 15 |
| 1 | 1 Heading on Level 1 (section) | 15 |
| 1.1 | 1.1 Heading on Level 2 (subsection) | 15 |

2 Lists 16

 2.1 Example for list (itemize) 16

 2.2 Example for list (enumerate) 16

 2.3 Example for list (description) 17

第 I 部分

说明

可以通过 `\parttext{text}` 在这里添加一些说明

1

初始化

/ 1 / 如何载入

1 只需要在第一行写：

```
\documentclass{simplivre}
```

2 即可使用 simplivre 文档类。

请注意

要使用 Xe_LAT_EX 或 Lua_LAT_EX 引擎才能编译。

/ 2 / 选项

3 simplivre 文档类有下面几个选项：

4 • **draft** 或 **fast**

5 – 你可以使用选项 **fast** 来启用快速但略微粗糙的样式，主要区别是：

6 * 使用较为简单的数学字体设置；

7 * 不启用超链接；

8 * 启用 ProjLib 工具箱的快速模式。

9 • **a4paper** 或 **b5paper**

10 – 可选的纸张大小。默认的纸张大小为 7in × 10in。

11 • **allowbf**

12 – 允许加粗。启用这一选项时，题目、各级标题、定理类环境名称会被加粗。

13 • **classical**

14 – 经典模式。启用这一选项时，样式会变得更加中规中矩：段落呈首行缩进样式、各
15 类下划线减少、标题样式改变、定理类环境的样式也会更贴近常见的设置。

提示

- 在文章的撰写阶段，建议使用 `fast` 选项以加快编译速度，改善写作时的流畅程度。在最后，可以把 `fast` 标记去除，从而得到正式的版本。使用 `fast` 模式时会有“DRAFT”字样的水印，以提示目前处于草稿阶段。
- 如果你喜欢更加传统的样式，`allowbf + classical` 是一个不错的搭配。

1 另外，排版图书时常用的 `oneside`、`twoside` 选项也是可以使用的。默认采用双页排版。

2

关于文档类中使用的字体

- 1 `simplivre` 默认使用 Palatino Linotype 作为英文字体，方正悠宋、悠黑 GBK 作为中文字体，
- 2 并部分使用了 Neo Euler 作为数学字体。其中，Neo Euler 可以在 <https://github.com/khaledhosny/euler-otf> 下载。其他字体不是免费字体，需要自行购买使用。可以在方正
- 3 字库网站查询详细资料：<https://www.foundertype.com>。

字体演示

- English main font. English sans serif font. English typewriter font.
- 中文主要字体，中文无衬线字体
- 数学示例： $\alpha, \beta, \gamma, \delta, 1, 2, 3, 4, a, b, c, d,$

$$\text{li}(x) := \int_2^x \frac{1}{\log t} dt$$

- 5 在没有安装相应的字体时，将采用 TeX Live 中自带的字体来代替，效果可能会有所折扣。

3

使用说明

1 接下来介绍的许多功能是由 [ProjLib](#) 工具箱提供的。这里只介绍了基本使用方法，更多细节可
2 以直接参阅其用户文档。

/ 1 /

语言设置

3 simplivre 提供了多语言支持，包括简体中文、繁体中文、英文、法文、德文、日文、俄文。可
4 以通过下列命令来选定语言：

- 5 • `\UseLanguage{<language name>}`，用于指定语言，在其后将使用对应的语言设定。
 - 6 – 既可以用于导言部分，也可以用于正文部分。在不指定语言时，默认选定“English”。
- 7 • `\UseOtherLanguage{<language name>}{<content>}`，用指定的语言的设定排版 `<content>`。
 - 8 – 相比较 `\UseLanguage`，它不会对行距进行修改，因此中西文字混排时能够保持行
9 距稳定。

10 `<language name>` 有下列选择：

- 11 • 简体中文：Chinese、chinese、SChinese、schinese、SimplifiedChinese
12 或 simplifiedchinese
- 13 • 繁体中文：TChinese、tchinese、TraditionalChinese 或 traditionalchinese
- 14 • 英文：English 或 english
- 15 • 法文：French 或 french
- 16 • 德文：German、german 或 ngerman
- 17 • 日文：Japanese 或 japanese
- 18 • 俄文：Russian 或 russian

19 另外，还可以通过下面的方式来填加相应语言的设置：

- 20 • `\AddLanguageSetting{<settings>}`
 - 21 – 向所有支持的语言增加设置 `<settings>`。
- 22 • `\AddLanguageSetting(<language name>){<settings>}`
 - 23 – 向指定的语言 `<language name>` 增加设置 `<settings>`。

1 例如, `\AddLanguageSetting(German){\color{orange}}` 可以让所有德语以橙色显示 (当
2 然, 还需要再加上 `\AddLanguageSetting{\color{black}}` 来修正其他语言的颜色)。

/ 2 /

定理类环境及其引用

3 定义、定理等环境已经被预定义, 可以直接使用。

4 具体来说, 预设的定理类环境包括: `assumption`、`axiom`、`conjecture`、`convention`、`corollary`、
5 `definition`、`definition-proposition`、`definition-theorem`、`example`、`exercise`、
6 `fact`、`hypothesis`、`lemma`、`notation`、`problem`、`property`、`proposition`、`question`、
7 `remark`、`theorem`, 以及相应的带有星号 * 的无编号版本。

8 在引用定理类环境时, 建议使用智能引用 `\cref{label}`。这样就不必每次都写上相应环境
9 的名称了。

例子

```
\begin{definition}[奇异物品] \label{def: strange} ...
```

将会生成

定义 3.1 | (奇异物品) 这是奇异物品的定义。定理类环境的前后有一行左右的间距。在
定义结束的时候会有一个符号来标记。

`\cref{def: strange}` 会显示为: 定义 3.1。

使用 `\UseLanguage{English}` 后, 定理会显示为:

THEOREM 3.2 | (Useless) A theorem in English.

默认情况下, 引用时, 定理的名称总是与定理所在区域的语言匹配, 例如, 上面的定义
在现在的英文模式下依然显示为中文: 定义 3.1 和 THEOREM 3.2。如果在引用时想让
定理的名称与当前语境相匹配, 可以在全局选项中加入 `regionalref`。

/ 3 /

定义新的定理型环境

10 若需要定义新的定理类环境, 首先要定义这个环境在所用语言下的名称 `\langle name of environ-`
11 `ment \rangle \langle language abbr \rangle`, 其中 `\langle language abbr \rangle` 是语言的简写, 分别为:

| | | | |
|----|------|----|----|
| CN | 简体中文 | DE | 德文 |
| TC | 繁体中文 | JP | 日文 |
| EN | 英文 | RU | 俄文 |
| FR | 法文 | | |

提示

如果要定义名称后带有星号 * 的环境，那么在上面的 $\langle \text{name of environment} \rangle$ 中不用写星号。

然后用下面五种方式之一定义这一环境：

- $\backslash\text{CreateTheorem}*\{\langle \text{name of environment} \rangle\}$
 - 定义不编号的环境 $\langle \text{name of environment} \rangle$
- $\backslash\text{CreateTheorem}\{\langle \text{name of environment} \rangle\}$
 - 定义编号环境 $\langle \text{name of environment} \rangle$ ，按顺序编号
- $\backslash\text{CreateTheorem}\{\langle \text{name of environment} \rangle\}[\langle \text{numbered like} \rangle]$
 - 定义编号环境 $\langle \text{name of environment} \rangle$ ，与 $\langle \text{numbered like} \rangle$ 计数器共用编号
- $\backslash\text{CreateTheorem}\{\langle \text{name of environment} \rangle\}<\langle \text{numbered within} \rangle>$
 - 定义编号环境 $\langle \text{name of environment} \rangle$ ，在 $\langle \text{numbered within} \rangle$ 计数器内编号
- $\backslash\text{CreateTheorem}\{\langle \text{name of environment} \rangle\}(\langle \text{existed environment} \rangle)$
 $\backslash\text{CreateTheorem}*\{\langle \text{name of environment} \rangle\}(\langle \text{existed environment} \rangle)$
 - 将 $\langle \text{name of environment} \rangle$ 与 $\langle \text{existed environment} \rangle$ 或 $\langle \text{existed environment} \rangle*$ 等同。
 - 这种方式通常在两种情况下比较有用：
 1. 希望定义更简洁的名称。例如，使用 $\backslash\text{CreateTheorem}\{\text{thm}\}(\text{theorem})$ ，便可以直接用名称 `thm` 来撰写定理。
 2. 希望去除某些环境的编号。例如，使用 $\backslash\text{CreateTheorem}\{\text{remark}\}(\text{remark}*)$ ，便可以去除 `remark` 环境的编号。

提示

其内部使用了 `amsthm`，因此传统的 `theoremstyle` 对其也是适用的，只需在相关定义前标明即可。

下面提供一个例子。这三行代码：

```
\def\proofideanameCN{思路}
\CreateTheorem*{proofidea*}
\CreateTheorem{proofidea}<section>
```

可以分别定义不编号的环境 `proofidea*` 和编号的环境 `proofidea` (在 `section` 内编号)，它们支持在简体中文语境中使用，效果如下所示：

思路 | `proofidea*` 环境。

思路 3.1 | `proofidea` 环境。

未完成标记

- 1 你可以通过 `\dnf` 来标记尚未完成的部分。例如：
- 2 • `\dnf` 或 `\dnf<...>`。效果为： `这里尚未完成 #1` 或 `这里尚未完成 #2: ...`。
- 3 其提示文字与当前语言相对应，例如，在法语模式下将会显示为 `Pas encore fini #3`。
- 4 类似的，还有 `\needgraph`：
- 5 • `\needgraph` 或 `\needgraph<...>`。效果为：
- 6 `这里需要一张图片 #1`
- 7 或
- 8 `这里需要一张图片 #2: ...`
- 9 其提示文字与当前语言相对应，例如，在法语模式下将会显示为
- 10 `Il manque une image ici #3`

关于行号

- 11 行号可以随时开启和关闭。`\linenumbers` 用来开启行号，`\nolinenumbers` 用来关闭行号。
- 12 标题、目录、索引等位置为了美观，不进行编号。

关于标题中的脚注

- 13 在 `\section` 或 `\subsection` 中，如果想使用脚注，只能：
- 14 • 先写 `\mbox{\protect\footnotemark}`，
- 15 • 再在后面用 `\footnotetext{...}`。
- 16 这是标题使用下划线装饰之后带来的一个缺点。

目前存在的问题

- 17 • 对于字体的设置仍然不够完善。
- 18 • 由于很多核心功能建立在 `ProjLib` 工具箱的基础上，因此 `simplivre` 自然继承了其所有问题。详情可以参阅 `ProjLib` 用户文档的“目前存在的问题”这一小节。
- 19
- 20 • 错误处理功能不完善，在出现一些问题时没有相应的错误提示。
- 21 • 代码中仍有许多可优化之处。

PART II

演示

4

文档示例

/ 1 / 标准文档类写法

- 1 如果想采用标准文档类中的写法，可以参考下面的例子：

```
\documentclass{simplivre}
\usepackage{PJLtoolkit} % Load ProjLib toolkit

\UseLanguage{French} % Use French from here

\begin{document}

\title{Le Titre}
\author{Auteur}
\date{\PJLdate{2022-04-01}}

\maketitle

\chapter{Un théorème}

%% Theorem-like environments can be used directly
\begin{theorem}\label{thm:abc}
    Ceci est un théorème.
\end{theorem}

Référence du théorème: \cref{thm:abc}
    % It is recommended to use clever reference

\end{document}
```

- 2 如果以后想切换到标准文档类，只需要将前两行换为：

```
\documentclass{book}
\usepackage[a4paper,margin=1in]{geometry}
\usepackage[hidelinks]{hyperref}
\usepackage{palatino}{PJLtoolkit} % Load ProjLib toolkit
```

- 1 如果日后有意切换到期刊模版，想采用 $\mathcal{A}\mathcal{M}\mathcal{S}$ 文档类中的写法，可以参考下面的例子：

```
\documentclass{simplivre}
\usepackage{PJLtoolkit} % Load ProjLib toolkit

\UseLanguage{French} % Use French from here

\begin{document}

\title{Le Titre}
\author{Auteur 1}
\address{Adresse 1}
\email{\href{Courriel 1}{Courriel 1}}
\author{Auteur 1}
\address{Adresse 1}
\email{\href{Courriel 2}{Courriel 2}}
\date{\PJLdate{2022-04-01}}
\subjclass{*****}
\keywords{...}

\maketitle

\section{Première section}

%% Theorem-like environments can be used directly
\begin{theorem}\label{thm:abc}
    Ceci est un théorème.
\end{theorem}

Référence du théorème: \cref{thm:abc}
    % It is recommended to use clever reference

\end{document}
```

- 2 这样，若想切换到 $\mathcal{A}\mathcal{M}\mathcal{S}$ 文档类，只需要将前两行换为：

```
\documentclass{amsbook}
\usepackage[a4paper,margin=1in]{geometry}
\usepackage[hidelinks]{hyperref}
\usepackage[palatino]{PJLtoolkit} % Load ProjLib toolkit
```

5

Heading on Level 0 (chapter)

1 Hello, here is some text without a meaning. This text should show what a printed text will
2 look like at this place. If you read this text, you will get no information. Really? Is there
3 no information? Is there a difference between this text and some nonsense like “Huardest
4 gefburn”? Kjift – not at all! A blind text like this gives you information about the selected
5 font, how the letters are written and an impression of the look. This text should contain all
6 letters of the alphabet and it should be written in of the original language. There is no need
7 for special content, but the length of words should match the language.

/ 1 /

Heading on Level 1 (section)

8 Hello, here is some text without a meaning. This text should show what a printed text will
9 look like at this place. If you read this text, you will get no information. Really? Is there
10 no information? Is there a difference between this text and some nonsense like “Huardest
11 gefburn”? Kjift – not at all! A blind text like this gives you information about the selected
12 font, how the letters are written and an impression of the look. This text should contain all
13 letters of the alphabet and it should be written in of the original language. There is no need
14 for special content, but the length of words should match the language.

1.1 | Heading on Level 2 (subsection)

15 Hello, here is some text without a meaning. This text should show what a printed text will
16 look like at this place. If you read this text, you will get no information. Really? Is there
17 no information? Is there a difference between this text and some nonsense like “Huardest
18 gefburn”? Kjift – not at all! A blind text like this gives you information about the selected
19 font, how the letters are written and an impression of the look. This text should contain all
20 letters of the alphabet and it should be written in of the original language. There is no need
21 for special content, but the length of words should match the language.

Heading on Level 3 (subsubsection)

22 Hello, here is some text without a meaning. This text should show what a printed text will
23 look like at this place. If you read this text, you will get no information. Really? Is there
24 no information? Is there a difference between this text and some nonsense like “Huardest
25 gefburn”? Kjift – not at all! A blind text like this gives you information about the selected
26 font, how the letters are written and an impression of the look. This text should contain all

1 letters of the alphabet and it should be written in of the original language. There is no need
2 for special content, but the length of words should match the language.

3 **Heading on Level 4 (paragraph)** Hello, here is some text without a meaning. This text
4 should show what a printed text will look like at this place. If you read this text, you will
5 get no information. Really? Is there no information? Is there a difference between this text
6 and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives
7 you information about the selected font, how the letters are written and an impression of
8 the look. This text should contain all letters of the alphabet and it should be written in of
9 the original language. There is no need for special content, but the length of words should
10 match the language.

/ 2 /
Lists

2.1 | Example for list (itemize)

- 11 ● First item in a list
- 12 ● Second item in a list
- 13 ● Third item in a list
- 14 ● Fourth item in a list
- 15 ● Fifth item in a list

Example for list (4*itemize)

- 16 ● First item in a list
 - 17 – First item in a list
 - 18 * First item in a list
 - 19 · First item in a list
 - 20 · Second item in a list
 - 21 * Second item in a list
 - 22 – Second item in a list
- 23 ● Second item in a list

2.2 | Example for list (enumerate)

- 24 1. First item in a list
- 25 2. Second item in a list
- 26 3. Third item in a list
- 27 4. Fourth item in a list
- 28 5. Fifth item in a list

Example for list (4*enumerate)

- 29 1. First item in a list
 - 30 (a) First item in a list
 - 31 i. First item in a list
 - 32 A. First item in a list
 - 33 B. Second item in a list
 - 34 ii. Second item in a list

- 1 (b) Second item in a list
- 2 2. Second item in a list

2.3 | Example for list (description)

- 3 **First** item in a list
- 4 **Second** item in a list
- 5 **Third** item in a list
- 6 **Fourth** item in a list
- 7 **Fifth** item in a list

Example for list (4*description)

- 8 **First** item in a list
- 9 **First** item in a list
- 10 **First** item in a list
- 11 **First** item in a list
- 12 **Second** item in a list
- 13 **Second** item in a list
- 14 **Second** item in a list
- 15 **Second** item in a list