



SIMPLIVRE

WRITE YOUR BOOKS IN
A SIMPLE AND CLEAR WAY

Corresponding to: simplivre 2022/04/03

JINWEN XU

April 2022, in Paris

PREFACE

simplivre is part of the minimalist class series. Its name is taken from French words “simple” and “livre” (for “book”). The entire collection includes minimart and einfart for typesetting articles, and minimbook and simplivre for typesetting books. My original intention in designing them was to write drafts and notes that look simple yet not shabby.

simplivre has multi-language support, including Chinese (simplified and traditional), English, French, German, Italian, Japanese, Portuguese (European and Brazilian), Russian and Spanish. These languages can be switched seamlessly in a single document. Due to the usage of custom fonts, einfart requires \LaTeX or \LuaTeX to compile.

This documentation is typeset using simplivre (with the option `use boldface`). You can think of it as a short introduction and demonstration.

TIP

Multi-language support, theorem-like environments, draft marks and some other features are provided by the [ProjLib](#) toolkit. Here we only briefly discuss how to use it with this document class. For more detailed information, you can refer to the documentation of [ProjLib](#).

TIP

This documentation has not been fully up-to-date with the new `exp13` version of this class series. Some options or commands introduced here might be obsolete.

CONTENTS

Preface i

I
INSTRUCTION

Before you start 3

1 Usage and examples 5

 1.1 How to load it 5

 1.2 Example - A complete document 5

 1.2.1 Initialization 6

 1.2.2 Set the language 6

 1.2.3 Draft marks 6

 1.2.4 Theorem-like environments 7

2 On the default fonts 9

3 The options 11

4 Instructions by topic 13

 4.1 Language configuration 13

 4.2 Theorems and how to reference them 14

 4.3 Draft mark 14

 4.4 Miscellaneous 15

 4.4.1 On the line numbers 15

 4.4.2 On the footnotes in the title 15

5 Known issues 17

II
DEMONSTRATION

6 Heading on Level 0 (chapter) 21

 6.1 Heading on Level 1 (section) 21

 6.1.1 Heading on Level 2 (subsection) 21

6.2 Lists 22

6.2.1 Example for list (itemize) 22

6.2.2 Example for list (enumerate) 22

6.2.3 Example for list (description) 23

PART I

INSTRUCTION

You can add some introductory text here via `\parttext(text)`.

BEFORE YOU START

- 1 In order to use the package or classes described here, you need to:
- 2 • install TeX Live or MikTeX of the latest possible version, and make sure that `minimalist`
- 3 and `projob` are correctly installed in your TeX system.
- 4 • download and install the required fonts, see the section “On the default fonts”.
- 5 • be familiar with the basic usage of L^AT_EX, and know how to compile your document with
- 6 pdfL^AT_EX, X_YL^AT_EX or LuaL^AT_EX.

1

USAGE AND EXAMPLES

1.1 HOW TO LOAD IT

1 One only needs to put

```
\documentclass{simplivre}
```

2 as the first line to use the simplivre class.

ATTENTION

You need to use either X_YLaTeX or LuaLaTeX engine to compile.

1.2 EXAMPLE - A COMPLETE DOCUMENT

3 Let's first look at a complete document.

```
1 \documentclass{simplivre}
2 \usepackage{ProjLib}
3
4 \UseLanguage{French}
5
6 \begin{document}
7
8 \frontmatter
9
10 \begin{titlepage}
11     <code for titlepage>
12 \end{titlepage}
13
14 \tableofcontents
15
16 \mainmatter
17
18 \part{<part title>}
19 \parttext{<text after part title>}
20
```

```

21 \chapter{<chapter title>}
22
23 \section{<section title>}
24
25 \DNF<<some hint>>
26
27 \begin{theorem}\label{thm:abc}
28     Ceci est un théorème.
29 \end{theorem}
30 Référence du théorème: \cref{thm:abc}
31
32 \backmatter
33
34 ...
35
36 \end{document}

```

If you find this example a little complicated, don't worry. Let's now look at this example piece by piece.

1.2.1 INITIALIZATION

```

\documentclass{simpivre}
\usepackage{ProjLib}

```

Initialization is straightforward. The first line loads the document class `simpivre`, and the second line loads the [ProjLib](#) toolkit to obtain some additional functionalities.

1.2.2 SET THE LANGUAGE

```

\UseLanguage{French}

```

This line indicates that French will be used in the document (by the way, if only English appears in your article, then there is no need to set the language). You can also switch the language in the same way later in the middle of the text. Supported languages include Simplified Chinese, Traditional Chinese, Japanese, English, French, German, Spanish, Portuguese, Brazilian Portuguese and Russian.

For detailed description of this command and more related commands, please refer to the section on the multi-language support.

1.2.3 DRAFT MARKS

```

\DNF<<some hint>>

```

When you have some places that have not yet been finished yet, you can mark them with this command, which is especially useful during the draft stage.

1.2.4 THEOREM-LIKE ENVIRONMENTS

```
\begin{theorem}\label{thm:abc}
  Ceci est un théorème.
\end{theorem}
Référence du théorème: \cref{thm:abc}
```

- 1 Commonly used theorem-like environments have been pre-defined. Also, when refer-
- 2 encing a theorem-like environment, it is recommended to use `\cref{<label>}` — in this way,
- 3 there is no need to explicitly write down the name of the corresponding environment every
- 4 time.

2

ON THE DEFAULT FONTS

By default, this document class uses Palatino Linotype as the English main font; Source Han Serif, Source Han Sans and Source Han Mono as the Chinese main font, sans serif font and typewriter font; and partially uses Neo Euler as the math font. You need to download and install these fonts by yourself. The Source Han font series can be downloaded at <https://github.com/adobe-fonts> (It is recommended to download the Super-OTC version, so that the download size is smaller). Neo Euler can be downloaded at <https://github.com/khaledhosny/euler-otf>. When the corresponding font is not installed, the font that comes with TeX Live will be used instead, and the effect may be discounted.

In addition, Source Code Pro is used as the English sans serif font, New Computer Modern Mono as the English monospace font, as well as some symbols in the mathematical fonts of Asana Math, Tex Gyre Pagella Math, and Latin Modern Math. These fonts are already available in TeX Live or MikTeX, which means you don't need to install them yourself.

- English main font. English sans serif font. English typewriter font.
- 简体中文主要字体，简体中文无衬线字体，简体中文等宽字体
- 繁體中文主要字體，繁體中文無襯線字體，繁體中文等寬字體
- 日本語のメインフォント、日本語のサンセリフフォント、日本語の等幅フォント
- Math demonstration: $\alpha, \beta, \gamma, \delta, 1, 2, 3, 4, a, b, c, d,$

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

3

THE OPTIONS

simplivre offers the following options:

- The language options EN / english / English, FR / french / French, etc.
 - For the option names of a specific language, please refer to $\langle language\ name \rangle$ in the next section. The first specified language will be used as the default language.
 - The language options are optional, mainly for increasing the compilation speed. Without them the result would be the same, only slower.
- `draft` or `fast`
 - The option `fast` enables a faster but slightly rougher style, main differences are:
 - * Use simpler math font configuration;
 - * Do not use `hyperref`;
 - * Enable the fast mode of [ProjLib](#) toolkit.

TIP

During the draft stage, it is recommended to use the `fast` option to speed up compilation. When in fast mode, there will be a watermark “DRAFT” to indicate that you are currently in the draft mode.

- `a4paper` or `b5paper`
 - Paper size options. The default paper size is 7in \times 10in.
- `palatino`, `times`, `garamond`, `noto`, `biolinum` | `useosf`
 - Font options. As the name suggest, font with corresponding name will be loaded.
 - The `useosf` option is used to enable the old-style figures.
- `use boldface`
 - Allow boldface. When this option is enabled, the main title, the titles of all levels and the names of theorem-like environments will be bolded.
- `classical`
 - Classic mode. An antique-looking style will be used when this option is enabled, like in the current documentation.
- `useindent`

- 1 – Use paragraph indentation instead of inter-paragraph spacing.
 - 2 • `runin`
 - 3 – Use the “runin” style for `\subsubsection`
 - 4 • `nothms`
 - 5 – Pure text mode. Do not load theorem-like environments.
 - 6 • `nothmnum`, `thmnum` or `thmnum=<counter>`
 - 7 – Theorem-like environments will not be numbered / numbered in order 1, 2, 3... / num-
8 bered within `<counter>`. If no option is used, they will be numbered within chapter
9 (book) or section (article).
 - 10 • `regionalref`, `originalref`
 - 11 – When referencing, whether the name of the theorem-like environment changes with
12 the current language. The default is `regionalref`, *i.e.*, the name corresponding to the
13 current language is used; for example, when referencing a theorem-like environment
14 in English context, the names “Theorem, Definition...” will be used no matter which
15 language context the original environment is in. If `originalref` is enabled, then the
16 name will always remain the same as the original place; for example, when referencing
17 a theorem written in the French context, even if one is currently in the English context,
18 it will still be displayed as “Théorème”.
 - 19 – In fast mode, the option `originalref` will have no effect.
- 20 In addition, the commonly used `oneside` and `twoside` options are also available. Two-
21 page layout is used by default.

4

INSTRUCTIONS BY TOPIC

4.1 LANGUAGE CONFIGURATION

simplivre has multi-language support, including Chinese (simplified and traditional), English, French, German, Italian, Japanese, Portuguese (European and Brazilian), Russian and Spanish. The language can be selected by the following macros:

- `\UseLanguage{⟨language name⟩}` is used to specify the language. The corresponding setting of the language will be applied after it. It can be used either in the preamble or in the main body. When no language is specified, “English” is selected by default.
- `\UseOtherLanguage{⟨language name⟩}{⟨content⟩}`, which uses the specified language settings to typeset `⟨content⟩`. Compared with `\UseLanguage`, it will not modify the line spacing, so line spacing would remain stable when CJK and Western texts are mixed.

`⟨language name⟩` can be (it is not case sensitive, for example, French and french have the same effect):

- Simplified Chinese: CN, Chinese, SChinese or SimplifiedChinese
- Traditional Chinese: TC, TChinese or TraditionalChinese
- English: EN or English
- French: FR or French
- German: DE, German or ngerman
- Italian: IT or Italian
- Portuguese: PT or Portuguese
- Portuguese (Brazilian): BR or Brazilian
- Spanish: ES or Spanish
- Japanese: JP or Japanese
- Russian: RU or Russian

In addition, you can also add new settings to selected language:

- `\AddLanguageSetting{⟨settings⟩}`
 - Add `⟨settings⟩` to all supported languages.
- `\AddLanguageSetting(⟨language name⟩){⟨settings⟩}`
 - Add `⟨settings⟩` to the selected language `⟨language name⟩`.

1 For example, `\AddLanguageSetting(German){\color{orange}}` can make all German
2 text displayed in orange (of course, one then need to add `\AddLanguageSetting{\color{`
3 `black}}` in order to correct the color of the text in other languages).

4.2 THEOREMS AND HOW TO REFERENCE THEM

4 Environments such as `definition` and `theorem` have been preset and can be used di-
5 rectly.

6 More specifically, preset environments include: `assumption`, `axiom`, `conjecture`, `convention`,
7 `corollary`, `definition`, `definition-proposition`, `definition-theorem`, `example`, `exercise`,
8 `fact`, `hypothesis`, `lemma`, `notation`, `observation`, `problem`, `property`, `proposition`,
9 `question`, `remark`, `theorem`, and the corresponding unnumbered version with an asterisk
10 `*` in the name. The titles will change with the current language. For example, `theorem` will
11 be displayed as “Theorem” in English mode and “Théorème” in French mode.

12 When referencing a theorem-like environment, it is recommended to use `\cref{<label>}`.
13 In this way, there is no need to explicitly write down the name of the corresponding envi-
14 ronment every time.

EXAMPLE

```
\begin{definition}[Strange things] \label{def: strange} ...
```

will produce

DEFINITION 4.1 (Strange things) — This is the definition of some strange objects.
There is approximately a one-line spacing before and after the theorem environment,
and there will be a symbol to mark the end of the environment.

`\cref{def: strange}` will be displayed as: **DEFINITION 4.1**.

After using `\UseLanguage{French}`, a theorem will be displayed as:

THÉORÈME 4.2 (Inutile) — Un théorème en français.

By default, when referenced, the name of the theorem matches the current context.
For example, the definition above will be displayed in French in the current French
context : la **DÉFINITION 4.1** et le **THÉORÈME 4.2**. If you want the name of the theorem
to always match the language of the context in which the theorem is located, you can
add `originalref` to the global options.

TIP

Please refer to the documentation of the package `create-theorem` to see how to de-
clare new theorem-like environments.

4.3 DRAFT MARK

15 You can use `\DNF` to mark the unfinished part. For example:

- 1 • `\DNF` or `\DNF<...>`. The effect is: `To be finished #1` or `To be finished #2: ...`.
2 The prompt text changes according to the current language. For example, it will be dis-
3 played as `Pas encore fini #3` in French mode.

4.4 MISCELLANEOUS

4.4.1 ON THE LINE NUMBERS

- 4 Line numbers can be turned on and off at any time. `\linenumbers` is used to enable
5 the line numbers, and `\nolinenumbers` is used to disable them. For the sake of beauty, the
6 title, table of contents, index and some other elements are not numbered.

4.4.2 ON THE FOOTNOTES IN THE TITLE

- 7 In `\section` or `\subsection`, if you wish to add footnotes, you can only:
8 • first write `\mbox{\protect\footnotemark}`,
9 • then add `\footnotetext{...}` afterwards.
10 This is a disadvantage brought about by the underline decoration of the title.

5

KNOWN ISSUES

- 1 • The font settings are still not perfect.
- 2 • Since many features are based on the [Projlib](#) toolkit, minimalist (and hence minimart,
3 einfart and minimbook, simplivre) inherits all its problems. For details, please refer to
4 the “Known Issues” section of the [Projlib](#) documentation.
- 5 • The error handling mechanism is incomplete: there is no corresponding error prompt
6 when some problems occur.
- 7 • There are still many things that can be optimized in the code.

PART II

DEMONSTRATION

6

HEADING ON LEVEL 0 (CHAPTER)

1 Hello, here is some text without a meaning. This text should show what a printed text
2 will 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.

6.1 HEADING ON LEVEL 1 (SECTION)

8 Hello, here is some text without a meaning. This text should show what a printed text
9 will 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.

6.1.1 HEADING ON LEVEL 2 (SUBSECTION)

15 Hello, here is some text without a meaning. This text should show what a printed text
16 will 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.

6.1.1.1 Heading on Level 3 (subsubsection)

22 Hello, here is some text without a meaning. This text should show what a printed text
23 will 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.

6.2 LISTS

6.2.1 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

6.2.1.1 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

6.2.2 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

6.2.2.1 Example for list (4*enumerate)

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

- 1 *b)* Second item in a list
- 2 *ii)* Second item in a list
- 3 2) Second item in a list

6.2.3 EXAMPLE FOR LIST (DESCRIPTION)

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

6.2.3.1 Example for list (4*description)

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

1 Below is a list of outstanding items, followed by the corresponding page
2 numbers.

3 #1 ⟨14⟩ :

4 #2 ⟨14⟩ : ...

5 #3 ⟨14⟩ :

6