

## 摘要

einfart 是 minimalist 文档类系列的成员之一, 其名称取自于德文的 einfach (“简约”), 并取了 artikel (“文章”) 的前三个字母组合而成。整个 minimalist 系列包含用于排版文章的 minimart、einfart 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, 使之看上去简朴而不简陋。

einfart 支持英语、法语、德语、意大利语、葡萄牙语、巴西葡萄牙语、西班牙语、简体中文、繁体中文、日文、俄文, 并且同一篇文档中这些语言可以很好地协调。由于采用了自定义字体, 需要用  $X_{\text{g}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  或  $\text{LuaL}^{\text{A}}\text{T}_{\text{E}}\text{X}$  引擎进行编译。

这篇说明文档即是用 einfart 排版的, 你可以把它看作一份简短的说明与演示。

## 目录

1	使用示例	2
1.1	如何加载	2
1.2	一篇完整的文档示例	2
2	关于默认字体	5
3	选项	5
4	具体说明	6
4.1	语言设置	6
4.2	定理类环境及其引用	7
4.3	定义新的定理型环境	7
4.4	未完成标记	9
4.5	文章标题、摘要与关键词	9
4.6	其他	10
5	目前存在的问题	10

## 开始之前

- 1 为了使用这篇文档中提到的文档类, 你需要:
- 2 ● 安装一个尽可能新版本的 TeX Live 或 MikTeX 套装, 并确保 minimalist 和 projlib 被
- 3 正确安装在你的  $\text{T}_{\text{E}}\text{X}$  封装中。
- 4 ● 熟悉  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  的基本使用方式, 并且知道如何用  $\text{pdfL}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 、 $\text{X}_{\text{g}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  或  $\text{LuaL}^{\text{A}}\text{T}_{\text{E}}\text{X}$  编译你的文
- 5 档。

## 1.1 | 如何加载

- 1 只需要在第一行写：

```
\documentclass{einfart}
```

- 2 即可使用 einfart 文档类。请注意，要使用 Xe<sub>L</sub>AT<sub>E</sub>X 或 Lua<sub>L</sub>AT<sub>E</sub>X 引擎才能编译。

## 1.2 | 一篇完整的文档示例

- 3 首先来看一段完整的示例。

```
1 \documentclass{einfart}
2 \usepackage{ProjLib}
3
4 \UseLanguage{French}
5
6 \begin{document}
7
8 \title{\langle title \rangle}
9 \author{\langle author \rangle}
10 \date{\PLdate{2022-04-01}}
11
12 \maketitle
13
14 \begin{abstract}
15     Ceci est un résumé. \dnf<Plus de contenu est nécessaire.>
16 \end{abstract}
17 \begin{keyword}
18     AAA, BBB, CCC, DDD, EEE
19 \end{keyword}
20
21 \section{Un théorème}
22
23 \begin{theorem}\label{thm:abc}
24     Ceci est un théorème.
25 \end{theorem}
26 Référence du théorème: \cref{thm:abc}
27
28 \end{document}
```

- 4 如果你觉得这个例子有些复杂，不要担心。现在我们来一点点地观察这个例子。

### 1.2.1 初始化部分

---

```
\documentclass{einfart}  
\usepackage{ProjLib}
```

---

- 1 初始化部分很简单：第一行加载文档类 `einfart`，第二行加载 `ProjLib` 工具箱，以便使用一些
- 2 附加功能。

### 1.2.2 设定语言

---

```
\UseLanguage{French}
```

---

- 3 这一行表明文档中将使用法语（如果你的文章中只出现英语，那么可以不需要设定语言）。你
- 4 也可以在文章中间用同样的方式再次切换语言。支持的语言包括简体中文、繁体中文、日文、
- 5 英语、法语、德语、西班牙语、葡萄牙语、巴西葡萄牙语、俄语。
- 6 对于这一命令的详细说明以及更多相关命令，可以参考后面关于多语言支持的小节。

### 1.2.3 标题，作者信息，摘要与关键词

---

```
\title{\langle title \rangle}  
\author{\langle author \rangle}  
\date{\PLdate{2022-04-01}}  
\maketitle  
  
\begin{abstract}  
  \langle abstract \rangle  
\end{abstract}  
\begin{keyword}  
  \langle keywords \rangle  
\end{keyword}
```

---

- 7 开头部分是标题和作者信息块。这个例子中给出的是最基本的形式，事实上你还可以这样写：

---

```
\author{\langle author 1 \rangle}  
\address{\langle address 1 \rangle}  
\email{\langle email 1 \rangle}  
\author{\langle author 2 \rangle}  
\address{\langle address 2 \rangle}  
\email{\langle email 2 \rangle}  
...
```

---

- 8 另外，你还可以采用  $\mathcal{AMS}$  文档类的写法：

---

```
\title{\langle title \rangle}  
\author{\langle author 1 \rangle}  
\address{\langle address 1 \rangle}
```

---

```

\email{\langle email 1 \rangle}
\author{\langle author 2 \rangle}
\address{\langle address 2 \rangle}
\email{\langle email 2 \rangle}
\date{\PLdate{2022-04-01}}
\subjclass{*****}
\keywords{\langle keywords \rangle}

\begin{abstract}
  \langle abstract \rangle
\end{abstract}

\maketitle

```

---

#### 1.2.4 未完成标记

```
\dnf<\langle some hint \rangle>
```

---

- 1 当你有一些地方尚未完成的时候，可以用这条指令标记出来，它在草稿阶段格外有用。

#### 1.2.5 定理类环境

```

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

```

---

- 2 常见的定理类环境可以直接使用。在引用的时候，建议采用智能引用 `\cref{\langle label \rangle}`——这样就不必每次都写上相应环境的名称了。

#### 提示

如果你之后想要切换到标准文档类，只需要把前两行换为：

```

\documentclass{article}
\usepackage[a4paper,margin=1in]{geometry}
\usepackage[hidelinks]{hyperref}
\usepackage[palatino,amsfashion]{ProjLib}

```

---

或者使用  $\mathcal{A}\mathcal{M}\mathcal{S}$  文档类：

```

\documentclass{amsart}
\usepackage[a4paper,margin=1in]{geometry}
\usepackage[hidelinks]{hyperref}
\usepackage[palatino]{ProjLib}

```

---

### 提示

如果你喜欢这个文档类，但又希望使用一种更加中规中矩的样式，那么不妨使用 `classical` 选项，就像这样：

```
\documentclass[classical]{einfart}
```

/ 2 /

## 关于默认字体

- 1 einfart 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>。
- 4
- 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 /

## 选项

6 einfart 文档类有下面几个选项：

- 7 • 语言选项 EN/english/English、FR/french/French，等等
- 8 – 具体选项名称可参见下一节的 `\language name`。第一个指定的语言将作为默认语言。
- 9 – 语言选项不是必需的，其主要用途是提高编译速度。不添加语言选项时效果是一样的，只
- 10 是会更慢一些。
- 11 • draft 或 fast
- 12 – 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：
- 13 \* 使用较为简单的数学字体设置；
- 14 \* 不启用超链接；
- 15 \* 启用 ProjLib 工具箱的快速模式。

### 提示

在文章的撰写阶段，建议使用 fast 选项以加快编译速度，改善写作时的流畅度。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。

- 16 • a4paper 或 b5paper
- 17 – 可选的纸张大小。默认的纸张大小为 7in × 10in。
- 18 • palatino、times、garamond、biolinum | useosf
- 19 – 字体选项。顾名思义，会加载相应名称的字体。
- 20 – useosf 选项用来启用“旧式”数字。

- 1 ● `allowbf`
  - 2 – 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。
- 3 ● `classical`
  - 4 – 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、各类下
  - 5 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

如果你喜欢更加传统的样式，`allowbf + classical` 是一个不错的搭配。

- 6 ● `useindent`
  - 7 – 采用段首缩进而不是段间间距。
- 8 ● `runin`
  - 9 – `\subsubsection` 采用“runin”风格。
- 10 ● `puretext` 或 `nothms`
  - 11 – 纯文本模式，不加载定理类环境。
- 12 ● `nothmnum`
  - 13 – 定理类环境不编号。

/ 4 /

### 具体说明

#### 4.1 | 语言设置

14 `einfart` 提供了多语言支持，包括英语、法语、德语、意大利语、葡萄牙语、巴西葡萄牙语、西  
15 班牙语、简体中文、繁体中文、日文、俄文。可以通过下列命令来选定语言：

- 16 ● `\UseLanguage{<language name>}`，用于指定语言，在其后将使用对应的语言设定。
  - 17 – 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”。
- 18 ● `\UseOtherLanguage{<language name>}{<content>}`，用指定的语言的设定排版 `<content>`。
  - 19 – 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距稳
  - 20 定。

21 `<language name>` 有下列选择（不区分大小写，如 `French` 或 `french` 均可）：

- 22 ● 简体中文：CN、Chinese、SChinese 或 SimplifiedChinese
- 23 ● 繁体中文：TC、TChinese 或 TraditionalChinese
- 24 ● 英文：EN 或 English
- 25 ● 法文：FR 或 French
- 26 ● 德文：DE、German 或 ngerman
- 27 ● 意大利语：IT 或 Italian
- 28 ● 葡萄牙语：PT 或 Portuguese
- 29 ● 巴西葡萄牙语：BR 或 Brazilian
- 30 ● 西班牙语：ES 或 Spanish

- 1 • 日文: JP 或 Japanese
- 2 • 俄文: RU 或 Russian

3 另外, 还可以通过下面的方式来填加相应语言的设置:

- 4 • `\AddLanguageSetting{<settings>}`
  - 5 – 向所有支持的语言增加设置 `<settings>`。
- 6 • `\AddLanguageSetting(<language name>){<settings>}`
  - 7 – 向指定的语言 `<language name>` 增加设置 `<settings>`。

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

#### 4.2 | 定理类环境及其引用

10 定义、定理等环境已经被预定义, 可以直接使用。

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

15 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  
16 的名称了。

#### 例子

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

将会生成

定义 4.1 | (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 4.1。

使用 `\UseLanguage{English}` 后, 定理会显示为:

THEOREM 4.2 | (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 4.1 和 THEOREM 4.2。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`。

#### 4.3 | 定义新的定理型环境

17 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:

- 18 • `\NameTheorem[<language name>]{<name of environment>}{<name string>}`

1 其中,  $\langle language name \rangle$  可参阅关于语言设置的小节。当不指定  $\langle language name \rangle$  时, 则会将该  
2 名称设置为所有支持语言下的名称。另外, 带星号与不带星号的同名环境共用一个名称, 因  
3 此  $\backslash NameTheorem\{envname*\}\{\dots\}$  与  $\backslash NameTheorem\{envname\}\{\dots\}$  效果相同。

4 然后用下面五种方式之一定义这一环境:

- 5 •  $\backslash CreateTheorem*\{\langle name of environment \rangle\}$ 
  - 6 – 定义不编号的环境  $\langle name of environment \rangle$
- 7 •  $\backslash CreateTheorem\{\langle name of environment \rangle\}$ 
  - 8 – 定义编号环境  $\langle name of environment \rangle$ , 按顺序编号
- 9 •  $\backslash CreateTheorem\{\langle name of environment \rangle\}[\langle numbered like \rangle]$ 
  - 10 – 定义编号环境  $\langle name of environment \rangle$ , 与  $\langle numbered like \rangle$  计数器共用编号
- 11 •  $\backslash CreateTheorem\{\langle name of environment \rangle\}\langle numbered within \rangle$ 
  - 12 – 定义编号环境  $\langle name of environment \rangle$ , 在  $\langle numbered within \rangle$  计数器内编号
- 13 •  $\backslash CreateTheorem\{\langle name of environment \rangle\}\langle existed environment \rangle$   
14  $\backslash CreateTheorem*\{\langle name of environment \rangle\}\langle existed environment \rangle$ 
  - 15 – 将  $\langle name of environment \rangle$  与  $\langle existed environment \rangle$  或  $\langle existed environment \rangle*$  等同。
  - 16 – 这种方式通常在两种情况下比较有用:
    - 17 1. 希望定义更简洁的名称。例如, 使用  $\backslash CreateTheorem\{thm\}\{theorem\}$ , 便可以直  
18 接用名称 `thm` 来撰写定理。
    - 19 2. 希望去除某些环境的编号。例如, 使用  $\backslash CreateTheorem\{remark\}\{remark*\}$ , 便可  
20 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。

21 下面提供一个例子。这三行代码:

```
 $\backslash NameTheorem[CN]\{proofidea\}\{思路\}$   
 $\backslash CreateTheorem*\{proofidea*\}$   
 $\backslash CreateTheorem\{proofidea\}\langle subsection \rangle$ 
```

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

24 思路 | `proofidea*` 环境。

25 思路 4.3.1 | `proofidea` 环境。

#### 4.4 | 未完成标记

你可以通过 `\dnf` 来标记尚未完成的部分。例如：

- `\dnf` 或 `\dnf<...>`。效果为：这里尚未完成 #1 或 这里尚未完成 #2: ...。  
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。

类似的，还有 `\needgraph`：

- `\needgraph` 或 `\needgraph<...>`。效果为：

这里需要一张图片 #1

或

这里需要一张图片 #2: ...

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

Il manque une image ici #3

#### 4.5 | 文章标题、摘要与关键词

`einfart` 同时具有标准文档类与  $\mathcal{A}\mathcal{M}\mathcal{S}$  文档类的一些特性。

因此，文章的标题部分既可以按照标准文档类 `article` 的写法来写：

---

```
\title{<title>}
\author{<author>\thanks{<text>}}
\date{<date>}
\maketitle
\begin{abstract}
  <abstract>
\end{abstract}
\begin{keyword}
  <keywords>
\end{keyword}
```

---

也可以按照  $\mathcal{A}\mathcal{M}\mathcal{S}$  文档类的方式来写：

---

```
\title{<title>}
\author{<author>}
\thanks{<text>}
\address{<address>}
\email{<email>}
\date{<date>}
\keywords{<keywords>}
\subjclass{<subclass>}
\begin{abstract}
  <abstract>
\end{abstract}
\maketitle
```

---

1 作者信息可以包含多组，输入方式为：

```
\author{\langle author 1\rangle}
\address{\langle address 1\rangle}
\email{\langle email 1\rangle}
\author{\langle author 2\rangle}
\address{\langle address 2\rangle}
\email{\langle email 2\rangle}
...
```

2 其中 `\address`、`\curraddr`、`\email` 的相互顺序是不重要的。

## 4.6 | 其他

### 4.6.1 关于行号

3 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。  
4 标题、目录、索引等位置为了美观，不进行编号。

### 4.6.2 关于标题中的脚注

5 在 `\section` 或 `\subsection` 中，如果想使用脚注，只能：

- 6 • 先写 `\mbox{\protect\footnotemark}`，
- 7 • 再在后面用 `\footnotetext{...}`。

8 这是标题使用下划线装饰之后带来的一个缺点。

### 4.6.3 关于 QED 符号

9 由于定理类环境中的字体和正文字体是一样的，为了方便看出定理类环境在哪里结束，在其  
10 结尾处放置了一个空心的 QED 符号。然而，如果你的定理是由公式或者列表结尾的，这个  
11 符号就无法自动地放在正确的位置。这时，你需要手动在公式或列表最后一个条目的后面加  
12 上 `\qedhere`，以让 QED 符号显示到这一行的最后。

/ 5 /

## 目前存在的问题

- 13 • 对于字体的设置仍然不够完善。
- 14 • 由于很多核心功能建立在 `ProjLib` 工具箱的基础上，因此 `einfart` 自然继承了其所有问题。  
15 详情可以参阅 `ProjLib` 用户文档的“目前存在的问题”这一小节。
- 16 • 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。
- 17 • 代码中仍有许多可优化之处。