ProjLib

beaulivre

以多彩的方式排版你的图书

对应版本. beaulivre 2021/05/24

许锦文

2021 年 5 月，北京

This page is intentionally left blank

前言

beaulivre 是 colorist 文档类系列的成员之一，其名称取自于法文的 beau (美丽)，以及 livre (书)，由二者组合而成。整个 colorist 系列包含用于排版文章的 colorart、lebhart 以及用于排版书的 colorbook、beaulivre。我设计这一系列的初衷是为了撰写草稿与笔记，使之多彩而不缭乱。

beaulivre 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。

这篇说明文档即是用 beaulivre 排版的 (使用了参数 `allowbf`)，你可以把它看作一份简短的说明与演示。

由于 colorist 主体是从 minimalist 系列修改而来的，因而一些页面元素还未完全重新设计，特别是目录、part 和 chapter 的样式。这些内容会在将来逐渐加入。

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。

This page is intentionally left blank

目 录

This page is intentionally left blank

1 初始化

1.1 如何载入

只需要在第一行写：

```
\documentclass{beaulivre}
```

即可使用 beaulivre 文档类。

请注意

要使用 XeLaTeX 或 LuaLaTeX 引擎才能编译。

1.2 选项

beaulivre 文档类有下面几个选项：

- `draft` 或 `fast`
  - 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：
    - 使用较为简单的数学字体设置；
    - 不启用超链接；
    - 启用 ProjLib 工具箱的快速模式。
- `a4paper` 或 `b5paper`
  - 可选的纸张大小。默认的纸张大小为 8.5in × 11in。
- `allowbf`
  - 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。

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。

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。

This page is intentionally left blank

2 关于文档类中使用的字体

beaulivre 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，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：$\alpha, \beta, \gamma, \delta, 1, 2, 3, 4, a, b, c, d,$

$$\operatorname{li}(x) := \int_2^\infty \frac{1}{\log t}\,\mathrm{d}t$$

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。

This page is intentionally left blank

3 使用说明

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，更多细节可以直接参阅其用户文档。

3.1 语言设置

beaulivre 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。可以通过下列命令来选定语言：

- \UseLanguage{⟨*language name*⟩}，用于指定语言，在其后将使用对应的语言设定。
  - 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”。
- \UseOtherLanguage{⟨*language name*⟩}{⟨*content*⟩}，用指定的语言的设定排版 ⟨*content*⟩。
  - 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定。

⟨*language name*⟩ 有下列选择 (不区分大小写，如 French 或 french 均可)：

- 简体中文：CN、Chinese、SChinese 或 SimplifiedChinese
- 繁体中文：TC、TChinese 或 TraditionalChinese
- 英文：EN 或 English
- 法文：FR 或 French
- 德文：DE、German 或 ngerman
- 日文：JP 或 Japanese
- 俄文：RU 或 Russian

另外，还可以通过下面的方式来填加相应语言的设置：

- \AddLanguageSetting{⟨*settings*⟩}
  - 向所有支持的语言增加设置 ⟨*settings*⟩。
- \AddLanguageSetting(⟨*language name*⟩){⟨*settings*⟩}
  - 向指定的语言 ⟨*language name*⟩ 增加设置 ⟨*settings*⟩。

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

3.2 定理类环境及其引用

定义、定理等环境已经被预定义，可以直接使用。

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

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。

`\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`。

下面是定理类环境的几种主要样式：

定理 3.3 Theorem style: theorem, proposition, lemma, corollary, ...

证明 | Proof style

Remark style

猜想 3.4 Conjecture style

例 Example style: example, fact, ...

问题 3.5 Problem style: problem, question, ...

为了美观，相邻的定义环境会自动连在一起：

定义 3.6 First definition.

定义 3.7 Second definition.

3.3 定义新的定理型环境

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：

- \NameTheorem[⟨*language name*⟩]{⟨*name of environment*⟩}{⟨*name string*⟩}

其中，⟨*language name*⟩ 可参阅关于语言设置的小节。当不指定 ⟨*language name*⟩ 时，则会将该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，因此 \NameTheorem{envname*}{...} 与 \NameTheorem{envname}{...} 效果相同。

然后用下面五种方式之一定义这一环境：

- \CreateTheorem*{⟨*name of environment*⟩}
  - 定义不编号的环境 ⟨*name of environment*⟩
- \CreateTheorem{⟨*name of environment*⟩}
  - 定义编号环境 ⟨*name of environment*⟩，按顺序编号
- \CreateTheorem{⟨*name of environment*⟩}[⟨*numbered like*⟩]
  - 定义编号环境 ⟨*name of environment*⟩，与 ⟨*numbered like*⟩ 计数器共用编号
- \CreateTheorem{⟨*name of environment*⟩}<⟨*numbered within*⟩>
  - 定义编号环境 ⟨*name of environment*⟩，在 ⟨*numbered within*⟩ 计数器内编号
- \CreateTheorem{⟨*name of environment*⟩}(⟨*existed environment*⟩)
  \CreateTheorem*{⟨*name of environment*⟩}(⟨*existed environment*⟩)
  - 将 ⟨*name of environment*⟩ 与 ⟨*existed environment*⟩ 或 ⟨*existed environment*⟩* 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 \CreateTheorem{thm}(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 \CreateTheorem{remark}(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：

```
\NameTheorem[CN]{proofidea}{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 3.3.1 | proofidea 环境。

3.4 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

3.5 目前存在的问题

- 对于字体的设置仍然不够完善。
- 由于很多核心功能建立在 ProjLib 工具箱的基础上，因此 beaulivre 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。
- 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。
- 代码中仍有许多可优化之处。

4 文档示例

4.1 标准文档类写法

如果想采用标准文档类中的写法，可以参考下面的例子：

```
\documentclass{beaulivre}
\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}
```

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
```

4.2 $\mathcal{AMS}$ 文档类写法

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

```
\documentclass{beaulivre}
\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}
```

这样，若想切换到 $\mathcal{AMS}$ 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)

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

5.1 Heading on Level 1 (section)

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

5.1.1 Heading on Level 2 (subsection)

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

Heading on Level 3 (subsubsection)

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

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

5.2 Lists

5.2.1 Example for list (itemize)

- First item in a list
- Second item in a list
- Third item in a list
- Fourth item in a list
- Fifth item in a list

Example for list (4*itemize)

- First item in a list
  - First item in a list
    - First item in a list
      - First item in a list
      - Second item in a list
    - Second item in a list
  - Second item in a list
- Second item in a list

5.2.2 Example for list (enumerate)

1. First item in a list
2. Second item in a list
3. Third item in a list
4. Fourth item in a list
5. Fifth item in a list

Example for list (4*enumerate)

1. First item in a list
   (a) First item in a list
      i. First item in a list
         A. First item in a list
         B. Second item in a list
      ii. Second item in a list
   (b) Second item in a list
2. Second item in a list

5.2.3 Example for list (description)

First item in a list
Second item in a list
Third item in a list
Fourth item in a list
Fifth item in a list

Example for list (4*description)

First item in a list

First item in a list

First item in a list

First item in a list

Second item in a list

Second item in a list

Second item in a list

Second item in a list