

UglyLATEX：系列 LATEX 文档类

(README.md)

Githubonline1396529

2025 年 3 月 18 日

目 录

| | | |
|-------|------------------------------|---|
| 1 | 项目简介 | 1 |
| 2 | 排版效果展示 | 2 |
| 3 | 文档类安装 | 3 |
| 3.1 | 使用 Makefile 进行自动安装 | 3 |
| 3.1.1 | 确定安装位置 | 3 |
| 3.1.2 | 指定安装目录 | 4 |
| 3.1.3 | 执行安装命令 | 5 |
| 3.2 | 手动安装 | 5 |
| 3.3 | 不安装模板直接使用 | 5 |
| 4 | 更多文档 | 5 |
| 5 | 宏包 | 5 |
| 6 | Q&A | 6 |
| 6.1 | 在 Pandoc 中启用颜色 | 6 |
| 6.2 | 在 Pandoc 中启用代码换行 | 6 |

一、项目简介

自从 [ElegantLATEX](#) 项目停更之后，我就时常感到十分的无措。尽管大家都说 [ElegantLATEX](#) 的历史使命已经完成、已经过时了，不在适合继续维护，但是我本来是很喜欢这个项目的。系列模板使用起来也特别方便，尤其是可以通过在 Markdwon 文件的

YAML Header 中使用 `documentclass` 指定文档类，再通过 Pandoc 一次性转换为 PDF via L^AT_EX 快速排版。

最初，为了满足我个人的使用需求，我自己搓了这几个模板。后来觉得比较好用，我就觉得应该发出来跟大家分享，大家一起用。但是因为我的技术比较菜，而且没有什么艺术细胞，做不到 Elegant，只能做到 Ugly，所以我把项目命名为 UglyL^AT_EX，很合理吧。

本项目旨在直接生成效果理想的排版文档，因为我想简化流程，直接生成想要的文档格式，无需在正式编写文档的时候再去调整。[ElegantL^AT_EX](#) 曾经经常被认为是将各种功能模块写得太死了，想要在排版的时候进行进一步的个性化格式修改就会很困难。这个问题在本项目中非但有之，而且更甚。总之，本项目的理念就是要确保在排版的过程中不需要引入任何额外的样式修改，直接提供足够理想的终产物样式，并适配尽可能广的应用场景。

到目前为止，本项目作为一个文档类已经可以很好地使用了，但是作为一个项目，还有很多不完善的地方。比如文档还不够齐全、代码格式和注释不严谨等。因此，本项目将在未来持续更新和完善，望各位用户能够持续跟进，或者参与本项目的开发。

二、排版效果展示

本项目总共包含三个模板类：

- **UglyNote:** 用于轻量化、便捷地编写常见的笔记、博客等。
- **UglyPaper:** 比 UglyNote 更紧凑的布局，适合编写更加正式的文档。
- **UglyRep:** 带有封面和摘要的报告，适用于更长和更正式的写作场景。

三个文档类各自的排版效果展示如下图所示。

UglyNote: 一个“丑陋”的 L^AT_EX 笔记模板

GitHub: <https://github.com/1396529/UglyNote>

2022 年 12 月 31 日

摘要

本文是 UglyNote 模板的排版效果示例及模板文件，展示了排版效果的同时简要说明了如何使用该模板及其使用方法。UglyNote 最初是我制作的 Markdown-to-PDF via L^AT_EX 的时候，其较好的 Pandoc Markdown-to-PDF via L^AT_EX 能容性，适合用于日常工作的快速排版。

关键词: L^AT_EX; 笔记; 文本类;

一、模板简介

1.1 模板介绍

自从 [ElegantL^AT_EX](#) 项目停更之后，我就对前端设计十分的无感，尽管大家都说 [ElegantL^AT_EX](#) 的历史已经过去，已经到了一个不适合继续维护的状态，但我还是决定自己动手做一个新的 L^AT_EX 笔记模板，希望是可以在 [Markdown](#) 和 [Pandoc](#) 中使用 `documentclass` 方案来实现，而不仅仅为 PDF via L^AT_EX 的通读提供。

最初，为了满足我个人的使用需求，我自己做了这几个模板，但是因为我的技术比较菜，我就觉得不如说出来给大家分享，大家一起用。但是因为我的技术比较菜，而且没有做什么艺术细胞，就不叫 Elegant，所以我就给自己命名为 UglyL^AT_EX，很合理吧。

1.2 守正创新

本模版运用了 [ElegantL^AT_EX](#) 中部分功能的实现。尽管 [ElegantL^AT_EX](#) 的部分功能（比如复合颜色和主题、可选的行内引用 `longref` 和 `long-on` 切换中英文语音模式等，目前版本基本就健健的已经有了，包括

• **语言模式切换:** 支持通过分类选项 `langca` 和 `lang-en` 切换中英文语音模式。

• **定义与公式编辑:** 支持通过分类选项 `longca` 和 `long-on` 切换中英文语音模式，并提供了 11 种不同的定理环境的选项，支持交叉引用。

3.1 来自 [ELEGANTLATEX](#) 的写作示例

3.2 写作示例

定义 3.1.1 (可积性) 设 $f(x) = \sum_{i=1}^k a_i \chi_{A_i}(x)$ 是 E 上的非负简单函数，其中 $\{A_1, A_2, \dots, A_k\}$ 是 E 上的一个可测分割， a_1, a_2, \dots, a_k 是非负实数，定义 f 在 E 上的积分值为 1 ，即

$$\int_E f dx = \sum_{i=1}^k a_i m(A_i).$$

一般情况下 $0 \leq f \leq \infty$ ，若 $\int_E f dx < \infty$ ，则称 f 在 E 上可积。

一个自然的问题是：Lebesgue 积分与我们所熟悉的 Riemann 积分有什么联系与区别？之后我们将详细讨论 Riemann 积分与 Lebesgue 积分的关系，这里只看一个简单的例子。设 $D(x)$ 是区间 $[0, 1]$ 上的 Dirichlet 函数，即 $D(x) = \chi_{Q_0}(x)$ ，其中 Q_0 表示 $[0, 1]$ 中的所有有理数的全体。根据非简单函数积分的定义， $D(x)$ 在 $[0, 1]$ 上的 Lebesgue 积分为

$$\int_0^1 D(x) dx = \int_0^1 \chi_{Q_0}(x) dx = m(Q_0) = 0$$

即 $D(x)$ 在 $[0, 1]$ 上是 Lebesgue 可积的并且积分值为零，但 $D(x)$ 在 $[0, 1]$ 上不是 Riemann 可积的。

表 3.1 燃油效率与汽车价格

| (1) | (2) | |
|-----------------|-------------|---------|
| 燃油效率 -238.90*** | -49.51 | |
| (53.09) | (88.10) | |
| 汽车重量 | 1.75*** | |
| (0.64) | | |
| 常数项 | 11253.00*** | 1946.00 |
| (1171.00) | (3597.00) | |
| 观测数 | 74 | |
| R^2 | 0.220 | 0.293 |

定义 3.1.1 (Fubini 定理) 若 $f(x, y)$ 是 $R^d \times R^e$ 上的非负可测函数，则对几乎处处的 $x \in R^d$ ， $f(x, y)$ 作为 y 是 R^e 上的非负可测函数， $g(x) =$

图 1 排版效果展示

2

10

三、文档类安装

将文档类安装到计算机可以允许您在计算机上的任何位置调用本文档类进行写作，这对于快速编排文档，尤其是与 Pandoc 协同编写文档尤其有帮助。本文档类有 2 种安装方法：

1. 使用 `Makefile` 进行自动安装；
2. 手动安装到计算机目录。

3.1 使用 `Makefile` 进行自动安装

3.1.1 确定安装位置

在安装文档类之前，您需要先确定合适的安装位置。通常，`LaTeX` 使用 `TEXMF` 目录组织所有宏包、类文件和字体资源。可用的 `TEXMF` 目录在不同的操作系统和不同的 `LaTeX` 发行版上可能会有所不同。

A. 对于 MiKTeX 用户： MiKTeX 上的 `TEXMF` 目录有 6 个，功能各不相同，分别是：

- **UserConfig:** 用于用户特定的配置文件；
- **UserData:** 用于用户特定的数据文件（格式文件、字体缓存等）；
- **UserInstall:** 用户安装的软件包的安装目标；
- **CommonConfig:** 用于共享（系统范围）配置文件；
- **CommonData:** 用于共享（系统范围）数据文件（格式文件、字体缓存等）；
- **CommonInstall:** MiKTeX 管理员安装的软件包的安装目标。

在不同的操作系统上，各个目录的具体位置请自行参考参考官方文档 [TEXMF root directories](#) 中的说明，此处不再赘述。您也可以启动 MiKTeXConsole，打开“设置”，选择“目录”选项卡，从而将任何目录设置为全局的 `TEXMF` 目录。如果您以管理员用户身份添加该目录，则该目录可以被计算机上所有用户访问；否则，该目录只能由当前用户访问。

B. 对于 TeXLive 用户： TeXLive 上可用的 `TEXMF` 目录通常包含三个：

1. **以年份命名的版本号目录：** 位于安装目录 `texlive` 下的形如 2023、2024 这样的目录名¹。保存的是 `texlive` 内置的宏包文件。不建议安装到此位置。
2. **本地用户自定义安装的宏包文件目录：** 只能被当前用户访问，在 Windows 和 Linux/-MacOS 上有所不同，一般情况下：

- **Linux/macOS:** `~/texmf`
- **Windows:** `%USERPROFILE%\texmf`

¹ 该数字即您安装的 TeXLive 年份版本号。

3. 全局用户自定义安装的宏包文件目录：能够被所有用户访问，如果您在 Linux/MacOS 上使用 TeXLive，则该路径可能是 /usr/local/texlive/texmf-local；如果您在 Windows 上使用 TeXLive，比如，如果您安装了 TeXLive 2023，则在您的 TeXLive 安装目录下会存在一个 texmf-local 目录。推荐安装到此位置。

您可以使用如下命令查询安装目录：

```
1 # 查询本地用户的安装目录  
2 kpsewhich -var-value TEXMFHOME  
3 # 查询全局安装目录  
4 kpsewhich -var-value TEXMFLOCAL
```

如果您不确定安装到哪个目录，则建议您安装到 TEXMFLOCAL 目录。

3.1.2 指定安装目录

在确定安装目录之后，您需要在 config.mk 中指定此目录。您需要编辑 TEXMF 变量，使其指向您指定的目录。config.mk 中的注释写得比较详细，可以作为参考。这里列举一些安装目录指定的操作方法：

对于 Windows 用户，假如您的 TeXLive 安装在系统的 D:\texlive 目录下，则你可以在该目录下找到名为 texmf-local 的目录。要将本文档类安装到该目录下，您需要设置 TEXMF 的值为：

```
TEXMF = /d/texlive/texmf-local
```

或者，如果您在 Windows 上使用 MiKTeX，假设您创建了一个 D:/Programs/texmf-local 目录，并已经正确添加到了 MiKTeX 的 TEXMF 列表，则可以像下面这样指定您的安装目录：

```
TEXMF = /d/Programs/texmf-local
```

Note. 在 Windows 上运行 make 命令，通常情况下需要配置 GCC 工具链。常见的 POSIX 兼容层，比如 MSYS 2，使用根目录下的 /c、/d 来指定 Windows 盘符。但如果使用的是 Cygwin，或者 WSL，则磁盘可能是被挂载在 /mnt 目录下。

```
1 TEXMF = /mnt/d/texlive/texmf-local  
2 TEXMF = /mnt/d/Programs/texmf-local
```

如果您在 Linux 或者 MacOS 上使用 TeXLive 并且希望将模板安装到本地用户的主目录：

```
TEXMF = ~/texmf
```

或者，安装到 TeXLive 的全局主目录：

```
TEXMF = /usr/local/texlive/texmf-local
```

3.1.3 执行安装命令

在正确指定了安装目录之后，只需要执行 `make install`，模板文件将会全部被安装到您的计算机上。安装完成后，执行 `mktexlsr` 命令刷新 TeX 的文件目录系统，从而使 LATEX 能够正确识别模板文件。

注意！如果您在 MiKTEX 上安装本模板，这里建议您在安装之后手动打开 MiKTEXConsole 刷新文件目录。MiKTEXConsole 的 GUI 界面在新版本中有所更新，这可能造成刷新按钮的位置有所不同，实际位置以具体版本为准。

3.2 手动安装

在一些情况下，比如，如果您的计算机上没有安装用以在 Windows 上执行 `make` 命令的兼容层环境，或者出于某些原因您不愿意使用自动安装的方案，您也可以手动安装本模板。具体的操作方法是：

1. 确认按安装位置：与自动安装时一样，安装位置必须是一个 `TEXMF` 目录。
2. 手动迁移文件：将 `./texmf/` 下的所有文件及目录复制黏贴到有待安装的目录下。
3. 执行 `mktexlsr` 命令刷新 TeX 的文件目录系统，从而使 LATEX 能够正确识别模板文件。

3.3 不安装模板直接使用

如果您不希望在您的计算机上安装本模板，而是希望直接在项目目录下使用模板，则操作方法也比较简单：直接复制本项目目录下的 `texmf/tex/latex/uglulatex/` 下的所有文件到您的项目目录下，即可直接在项目文档中指定本文档类使用。

四、更多文档

如果您想查看本文档类中的更多文档，请使用如下命令，列出所有项目文档。

```
texdoc -l uglylatex
```

五、宏包

本文档类用到的所有宏包如下表所示。在使用本文档类之前，请您先确保这些宏包已经被正确配置。

表 1 本文档类所使用到的宏包

| 1 | 2 | 3 | 4 |
|---------|-----------|----------|-------|
| ifxetex | kvoptions | etoolbox | calc |
| ctex | titling | titlesec | array |
| hologo | geometry | fontspec | float |

| 1 | 2 | 3 | 4 |
|-------------------|------------|----------|-----------|
| authblk | amsmath | amssymb | amsthm |
| mathtools | mathrsfs | cancel | tocloft |
| titletoc | hyperref | fancyhdr | enumerate |
| enumitem | metalogo | setspace | caption |
| subcaption | appendix | graphicx | booktabs |
| multirow | longtable | pdfpages | braket |
| qcircuit | mhchem | chemfig | tikz |
| tikz-network | circuitikz | pdfpages | braket |
| multirow | longtable | pgfplots | color |
| xcolor | colortbl | xpatch | verbatim |
| matlab-prettifier | bookmark | | |

六、Q&A

6.1 在 Pandoc 中启用颜色

Pandoc 默认会将所有引用、文件链接、URL 和交叉引用设置为黑色。如果需要启用颜色格式，可以在 Markdown 文件开头的 YAML Header 中加入如下内容。

```
1 # Enable document colors.
2 colorlinks: true
3 citecolor: ecolor
4 filecolor: ecolor
5 linkcolor: ecolor
6 urlcolor: ecolor
```

6.2 在 Pandoc 中启用代码换行

Pandoc 中默认使用 Verbatim 环境高亮代码，过长的代码可能会超出屏幕。在 Markdown 文件开头的 YAML Header 中使用如下命令可以确保高亮的代码块启用自动换行。

```
1 # Warp lines in code blocks
2 header-includes:
3   - \usepackage{fvextra}
4   - |
5     \DefineVerbatimEnvironment{Highlighting}{Verbatim}{
6       breaklines,commandchars=\{\}}
7 }
```