

1

L^AT_EX 简介

第1章 L^AT_EX 简介

1.1 什么是 L^AT_EX?

在谈论 L^AT_EX 之前，先说一下 T_EX。

T_EX 是高德纳 (Donald E. Knuth) 在 1978 年首次发布的排版系统。T_EX 的名字来源于 “technology” 的希腊词根 “τεχ”。T_EX 应该按照希腊字母的读法读成 [tex]，在英语环境中读成 [tek] 也可勉强接受^[10]。

T_EX 也叫做 “引擎”，它的功能类似于编程环境中的编译器：编译器接收源代码，经过编译得到可执行文件；T_EX 接收 .tex 文件，经过编译得到最终的排版结果。

T_EX 程序本身很难直接使用。高德纳用 T_EX 能直接读取的原语 (primitive) 写了一组宏 (macro)。使用了这组宏的 T_EX 叫做 plain T_EX，plain T_EX 风格的文档又叫做 plain T_EX 格式 (format)。由 Leslie B. Lamport 编写的 L^AT_EX 是目前广泛使用的格式。

L^AT_EX 文档的内容和样式是分离的。控制排版样式的代码被封装进宏包 (package) 和文档类 (document class) 中。用户无需过度关心文档的版面格式，可以更专注于文档内容。

1.2 开始使用 L^AT_EX

1.2.1 安装 T_EX 发行版

使用 L^AT_EX 需要下载 T_EX 发行版，所谓的发行版也就是包含了引擎、宏包、文档类等文件的软件包。

主流的 T_EX 发行版有 T_EX Live 和 MiK_T_EX。我们更推荐使用 T_EX Live。

安装 T_EX Live T_EX Live 可以通过光盘镜像进行安装，安装 T_EX Live 所需的光盘镜像文件可以在下面的地址中下载：

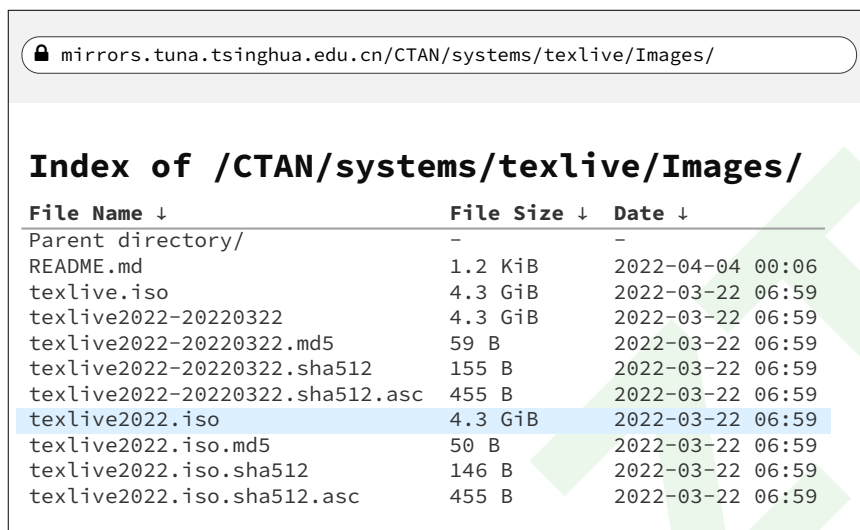
<https://mirror.ctan.org/systems/texlive/Images/>

直接下载页面中的 texlive(年份).iso (比如图 1.1 中的 texlive2022.iso) 即可。如果下载速度较慢，也可以选择下面这两个镜像站进行下载：

<https://mirrors.tuna.tsinghua.edu.cn/CTAN/systems/texlive/Images/>

<https://mirrors.ustc.edu.cn/CTAN/systems/texlive/Images/>

下载完成后，双击进入 texlive(年份).iso 开始安装。Windows 用户可以双击其中的 install-tl-windows.bat 进入安装程序。Linux、macOS 等类 UNIX 系统用户

1.2 开始使用 L^AT_EX


File Name ↓	File Size ↓	Date ↓
Parent directory/	-	-
README.md	1.2 KiB	2022-04-04 00:06
texlive.iso	4.3 GiB	2022-03-22 06:59
texlive2022-20220322	4.3 GiB	2022-03-22 06:59
texlive2022-20220322.md5	59 B	2022-03-22 06:59
texlive2022-20220322.sha512	155 B	2022-03-22 06:59
texlive2022-20220322.sha512.asc	455 B	2022-03-22 06:59
texlive2022.iso	4.3 GiB	2022-03-22 06:59
texlive2022.iso.md5	50 B	2022-03-22 06:59
texlive2022.iso.sha512	146 B	2022-03-22 06:59
texlive2022.iso.sha512.asc	455 B	2022-03-22 06:59

图 1.1: 下载 T_EX Live

可以在当前路径下打开命令行或终端，执行下面的命令开始安装：

```
./install-tl --gui
```

以 Windows 为例，进入安装程序后，安装界面如图 1.2。可以在“Installation root”一栏中修改安装路径。注意，安装路径不要出现中文，否则会影响后续使用。T_EXworks 是 T_EX Live 自带的编辑器，可以选择安装。



点击“安装”按钮，即可开始安装。由于 T_EX Live 比较大，安装时间会比较长（约 10 到 30 分钟）。安装时请耐心等待，不要在安装结束前关闭窗口。

当安装窗口显示“欢迎进入 T_EX Live 的世界！”时说明安装成功，这时就可以开始使用 T_EX Live 了。

对于 Linux 系统，安装完成后还需在 root 权限下进行以下操作：

1. 将 texlive-fontconfig.conf 复制到 /etc/fonts/conf.d/09-texlive.conf；
2. 运行 `fc-cache -fsv` 刷新字体缓存。

这与 X_YT_EX 的 fontconfig 库有关^[1]。如果遇到了 X_YT_EX 使用系统字体方面的问题，可以查阅 T_EX Live 的手册 [1]。

在 T_EX Live 的安装界面中，最下面的“Advanced”按钮用来设置高级安装选项。用户在此可以重新设置 `texmf` 等路径，不过不建议没有使用经验的用户自己设置这些路径。

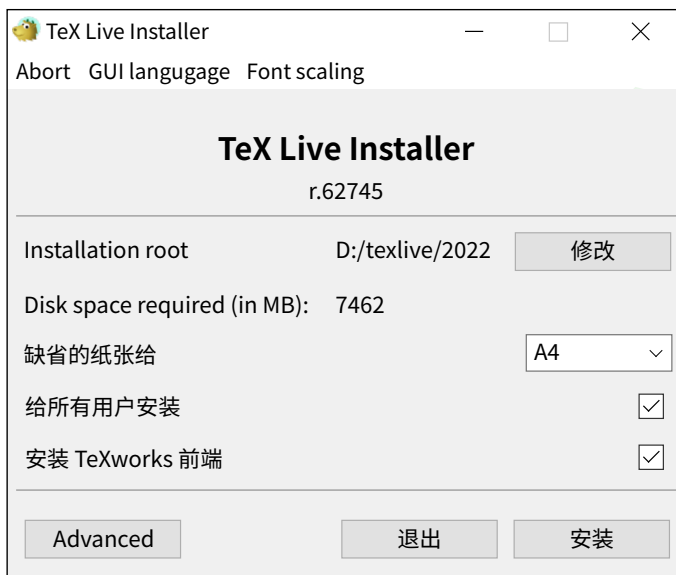


图 1.2: T_EX Live 的安装界面

更新 T_EX Live T_EX Live 每年都会更新一次，建议用户定期更新。在更新 T_EX Live 前，先要卸载旧版本的 T_EX Live。Windows 用户可以运行“开始”菜单中的“Uninstall T_EX Live”，或运行安装目录下的 `/tlpkg/installer/uninst.bat` 进行卸载（如右图中的 Uninstall T_EX Live）。Linux 和 macOS 用户可以直接删除 `texlive` 目录。卸载完成后下载新版 T_EX Live 重新安装即可更新。

只有最新的 T_EX Live 才可以使用 `tlmgr` 进行更新。

更新宏包和文档类 更新 T_EX Live 中收录的宏包、文档类需要通过命令行程序 `tlmgr`（T_EX Live manager）来实现。打开命令行，执行

```
tlmgr update --self --all
```

可以更新全部可更新的宏包和文档类。`tlmgr` 也有相应的 GUI 程序，Windows 用户在“开始”菜单中可以找到。

手动安装/宏包和文档类 手动安装宏包可以在命令行执行

```
tlmgr install <宏包名>
```

1.3 编写你的第一个 L^AT_EX 文档

移除宏包只需把上面的 `install` 换成 `remove` 即可。但是随意移除宏包可能是很危险的，因为有可能破坏有些宏包之间的依赖关系。



1.2.2 使用在线的 T_EX Live

如果有多人合作文档或文档版本管理的需求，使用在线的 T_EX 发行版会是非常好的选择。在线的 T_EX Live，主要有 Overleaf 和 T_EXPage。二者的网址分别是：

<https://www.overleaf.com>

<https://www.texpage.com>

Overleaf 支持的 T_EX Live 版本从 T_EX Live 2014 开始，T_EXPage 支持的 T_EX Live 版本从 T_EX Live 2019 开始。T_EXPage 的中文字体支持更多，网络条件也会更好一些。二者的编辑器界面分别如第 11 页图 1.5a 和 1.5b。

1.2.3 必备技能：texdoc

`texdoc` 是一个命令程序，功能是查阅 T_EX Live 中的文档。这些文档包括：发行版的说明文档、宏包/文档类手册，等等。

使用本地的 `texdoc` 在命令行中执行命令

```
texdoc <关键字>
```

即可查看对应的文档。比如，执行命令 `texdoc texlive` 即可查看 T_EX Live 的手册。

使用在线的 `texdoc` 对于 Overleaf 和 T_EXPage 用户，如果本地没有安装 T_EX Live 发行版，可以通过 T_EXdoc 网站使用 `texdoc`：

<https://texdoc.org/index.html>

1.3 编写你的第一个 L^AT_EX 文档

我们使用 T_EXworks 编写第一个 L^AT_EX 文档。T_EXworks 的界面如图第 12 页图 1.6a 所示。把以下代码输入到 T_EXworks 编辑器中：

第 1 章 L^AT_EX 简介

```
\documentclass{article}           % 使用 article 文档类

\begin{document}

Happy \TeX ing!

\end{document}
```

保存后点击界面左上角的“▶”按钮（如第 7 页图 1.3 左上角），可得到类似于下面这样的效果：

Happy T_EXing!

这个简单的例子足以说明 L^AT_EX 格式最基本的特点：

- 文档类 ▶ 1. L^AT_EX 文档需要在最前面指定一个文档类（document class），指定文档类的命令是 `\documentclass{<文档类>}`。
- 导言区和正文 ▶ 2. 位于 `\begin{document}` 和 `\end{document}` 之间的内容是真正输出到页面上的内容，这部分内容又叫正文。位于 `\begin{document}` 之前的部分叫做导言区（preamble）。

此外，还有一些需要注意的内容：

- 宏包 ▶ 3. 宏包（package）可以为文档提供功能支持。在导言区用 `\usepackage{<宏包名>}` 来使用宏包。例如，`\usepackage{amsmath}`。注意，宏包需要在指定文档类之后使用。
- 4. 使用百分号（%）来对代码进行注释（comment）。同一行内百分号后面的内容会被忽略。
- 5. 在 L^AT_EX 中，分段通过空一行来完成。详见 2.1.1 小节。

最后我们给出一个稍微复杂一点的完整的例子，详见第 149 页。

1.4 排版引擎与编译方式

在前面的例子中，将文件保存为 `file.tex`。点击“▶”按钮开始排版，实际上是在 `file.tex` 所在的路径下执行下面这行命令：

1.4 排版引擎与编译方式

```
pdflatex file.tex
```

这里的 `pdflatex` 表示使用 `pdfTeX` 作为引擎、使用 `LaTeX` 格式来编译文档。类似地，我们也可以使用其他方式来编译文档，比如 `xelatex`、`lualatex` 等。这些命令行命令通常称为“编译方式”。在本书中，编译方式写成图标的形式，如 `pdfLaTeX`、`XYLaTeX`、`LuaLaTeX` 等。

中文文档的编译方式 对于中文文档，推荐使用 `XYLaTeX` 或 `LuaLaTeX` 编译并使用 `ctex` 宏集作为中文支持。使第 5 页给出的例子支持中文最简单的办法就是把第一行的 `\documentclass{article}` 修改为

```
\documentclass{ctexart}
```

此外，还需确保文档以 UTF-8 编码保存（见本章思考题第 1 题）。在第 150 页中给出了一个完整的中文文档的例子。

在 `TeXworks` 编辑器中使用 `XYLaTeX` 作为编译方式，需要下拉“▶”按钮右侧的选项框（如图 1.3），选中 `XYLaTeX`，再点击“▶”按钮，也就是执行命令

```
xelatex file.tex
```

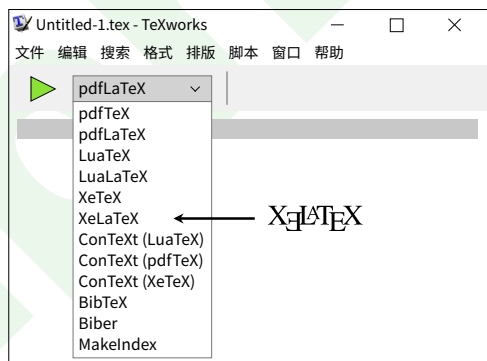


图 1.3: `TeXworks` 编辑器可选的编译方式

命令行编译 打开命令行，用 `cd` 命令将当前路径切换至源文件所在的路径下。`cd` 的对象可以是绝对路径，也可以是相对路径。例如：

在 Windows 系统中可以使用 `cmd` 或者 `Powershell` 作为命令行。首先通过 `Win + R` 组合键调出“运行”窗口，输入 `cmd` 即可打开 `cmd`，输入 `powershell` 即可打开 `Powershell`。macOS 系统中的“终端”也是命令行。

第 1 章 L^AT_EX 简介

```
cd C:/Users
```

这条命令，无论当前在哪个路径下，都会进入 C:/Users 文件夹中；在此路径下，在命令行中输入

```
cd Public
```

就会进入 C:/Users/Public 路径。

Windows 用户也可以在当前显示的文件夹中按住 Shift 同时点击右键，选择打开 PowerShell 或 Windows Terminal^①，这样可以使命令行路径位于该文件夹。

命令行编译 ▶

使用命令行编译 L^AT_EX 文档，需要在文档所在的路径下打开命令行。假如文档叫做 file.tex，就可以直接在命令行窗口输入下面这行命令：

```
xelatex file.tex
```

上面这行命令表示使用 X_LL^AT_EX 编译。编译命令也支持一些选项，比如：

```
xelatex -shell-escape file.tex
```

这个编译命令启用了 -shell-escape 选项，从而可以使用一些依赖于外部工具的宏包（比如依赖于 Python 的 minted 宏包）。

使用命令行编译，若源文件的扩展名为 .tex，则扩展名可以省略，如：

```
xelatex file
```

但是其他的扩展名（如 .ins）则不可省略。

使用命令行编译的另一个好处是比较好调试（debug）。如果文档中有一些试验性的命令或者文档使用了互相冲突的宏包，使用命令行编译可以实时查看当前的窗口报错或警告信息，以便于修改代码。

1.5 使用其他编辑器

T_EX Live 自带的 T_EXworks 编辑器在编辑较大的文档时可能出现卡顿，因此多数用户有更换编辑器的需求。本书中我们以 T_EXstudio 编辑器为例。

^① Windows Terminal 需要用户自己从 Microsoft Store 安装。

1.5 使用其他编辑器

T_EXstudio T_EXstudio 是一款免费开源的 L^AT_EX 专用编辑器，在代码补全、快捷键等方面的表现比较优秀。T_EXstudio 的官方网站是：

<https://www.texstudio.org>

由于网络原因，上面的网站可能无法访问。若无法访问，可在 GitHub 中下载：

<https://github.com/texstudio-org/texstudio/releases/>

也可以在 SourceForge 和清华大学 tuna 镜像站下载：

<http://texstudio.sourceforge.net>

<https://mirrors.tuna.tsinghua.edu.cn/github-release/texstudio-org/texstudio/>

根据操作系统下载相应的安装包，安装包的版本可以自己选择，建议选择版本较新的稳定版。T_EXstudio 在安装过程中会自动添加到路径并更新系统文件，因此，在安装前，需要先关闭所有程序，然后按照安装程序的引导完成安装。

配置 T_EXstudio 安装成功后，打开 T_EXstudio，是全英文的界面。这时需要在菜单栏的“选项 → 设置 T_EXstudio”（Options → Configure T_EXstudio）进行设置：

“常规 → 语言”（General → Language）是编辑器界面的显示语言选项，可以设置为 zh_CN。

“编辑器 → 字体”（Editor → Font Family）是编辑器的字体，建议选择等宽字体，比如 Consolas、Ubuntu Mono 等。

在“构建 → 默认编译器”（Build → Default Compiler）可以设置默认编译命令。对于中文文档，建议将“默认编译器”设置为 X_YL^AT_EX（如图 1.4）。

在“命令”（Commands）中还可以设置各个编译方式的具体行为。比如，我们可以把 X_YL^AT_EX 设置为下面这行命令：

```
xelatex -shell-escape -synctex=1 %.tex
```

其中，-shell-escape 选项开启 shell 转义，这一选项的直接应用就是可以使用 minted 宏包实现抄录代码高亮（见彩页）；-synctex=1 选项开启 SyncT_EX 程序，用于实现源代码和 PDF 之间的相互跳转。

使用 T_EXstudio 的魔法注释 魔法注释（magic comment）是可以被编辑器识别的注释，它可以控制编译命令的行为。比如说，在文档最前面加上这行代码：

```
% !TeX program = xelatex
```

第 1 章 \LaTeX 简介

这样的话，如果使用 \TeXstudio ，无论默认的编译器是什么，都会使用 \XeLaTeX 编译。编译时， \TeXstudio 下方的“消息”窗口会在第一行显示

```
%!TeX 指定使用的程序：xelatex
```

此外，拼写检查和文档编码的魔法注释也很常用：

```
% !TeX spellcheck = de_DE
```

上面的这一行魔法注释告诉 \TeXstudio 开启德语拼写检查。

```
% !TeX encoding = UTF-8
```

这一行魔法注释告诉 \TeXstudio 文档使用 UTF-8 编码。

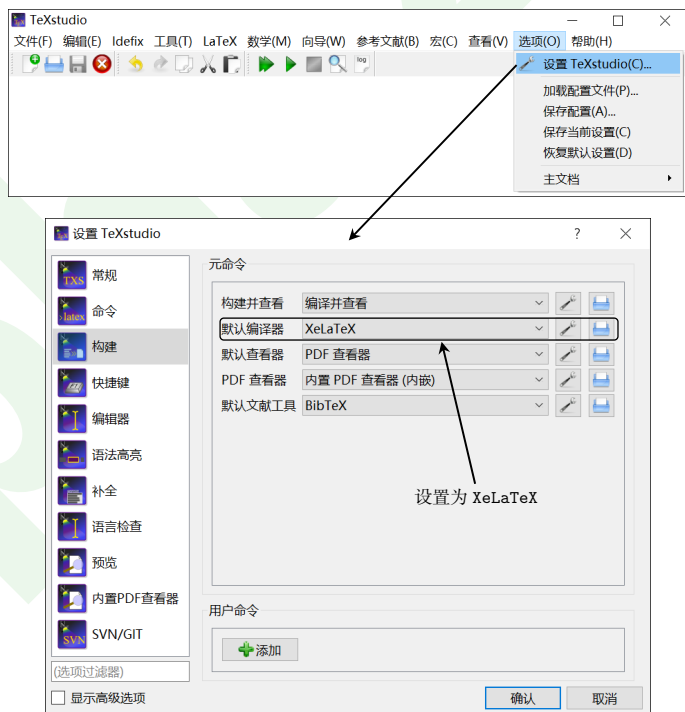
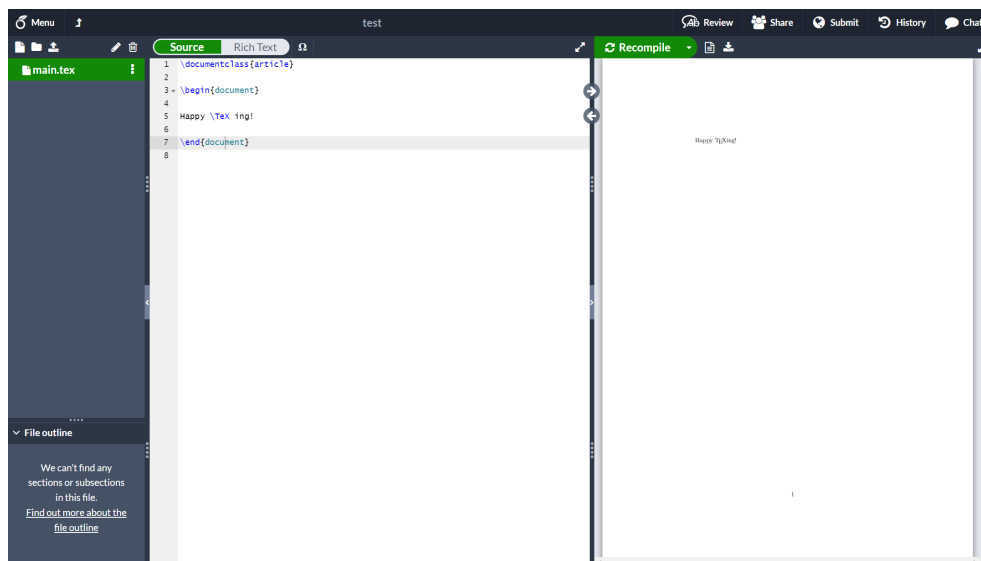
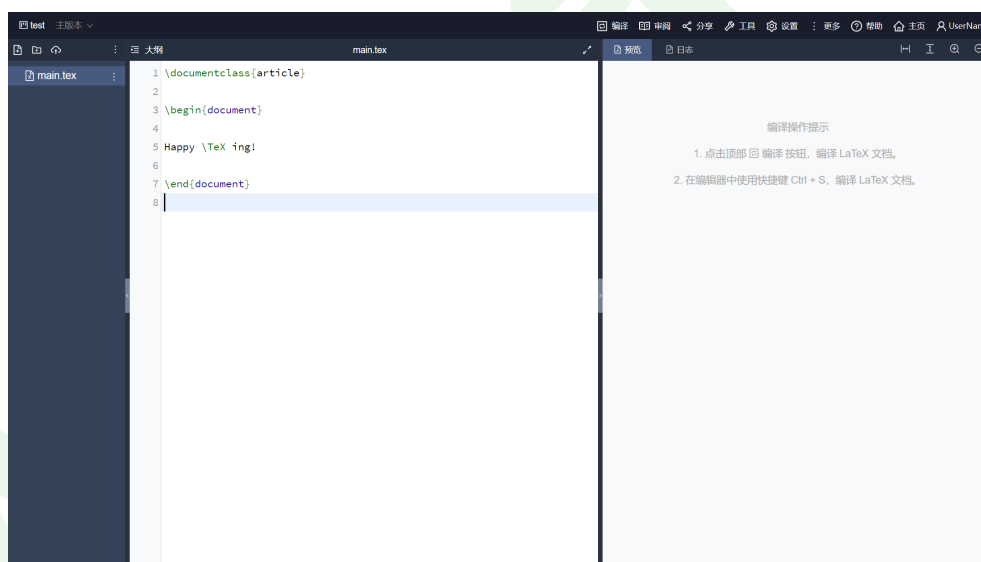


图 1.4: 设置 \TeXstudio

1.5 使用其他编辑器

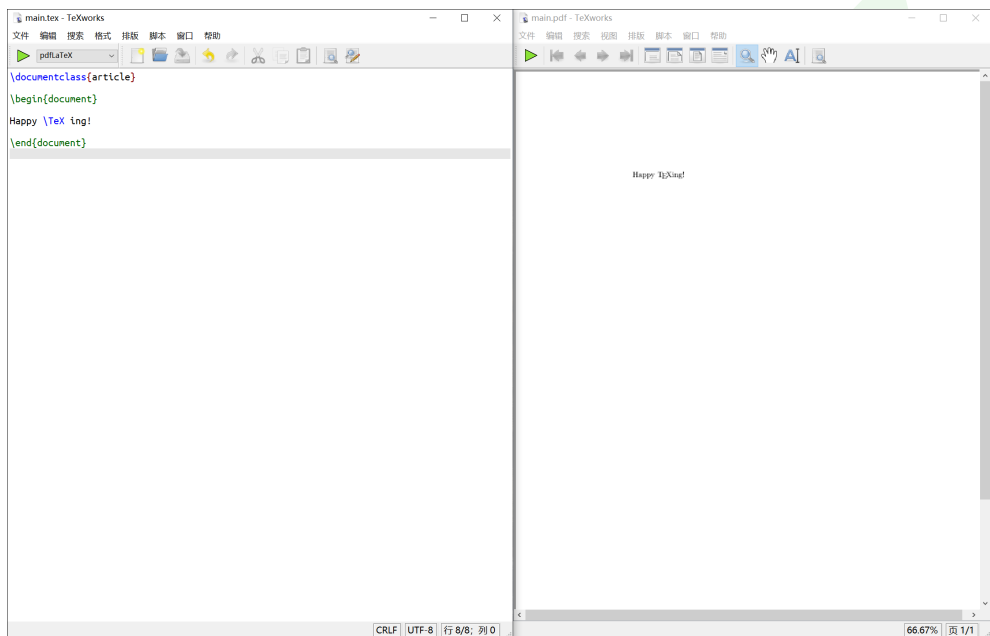
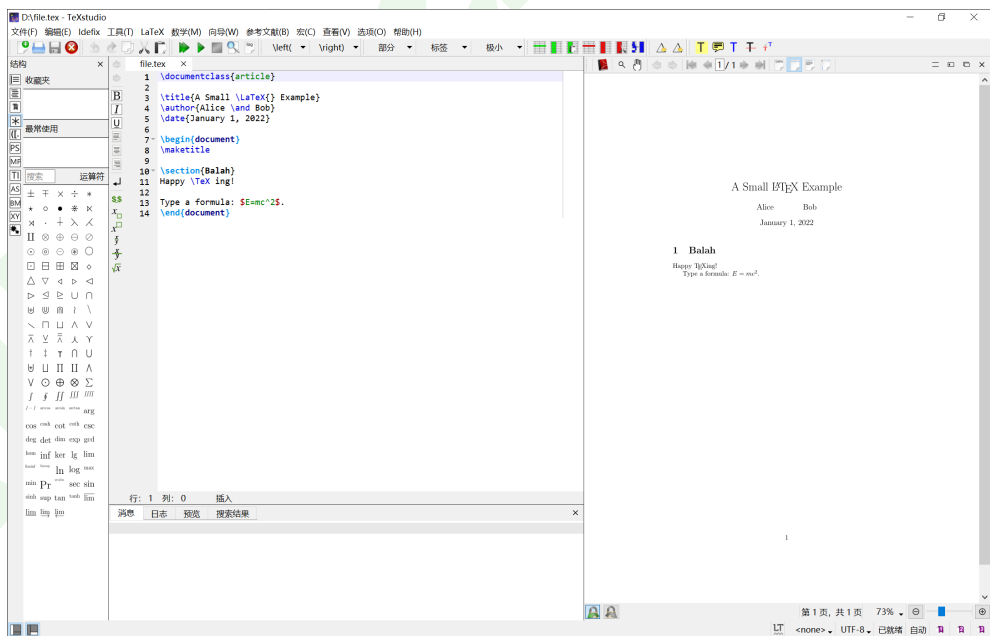


(a) Overleaf 编辑器界面



(b) TeXPage 编辑器界面

图 1.5: 在线的 $\text{T}_{\text{E}}\text{X}$ 编辑器界面

第 1 章 \TeX 简介(a) \TeX works 编辑器界面(b) \TeX studio 编辑器界面图 1.6: 本地的 \TeX 编辑器界面

本章总结

1. L^AT_EX 文档的内容和格式是分离的，作者可以专注于内容的安排。
2. 本地 T_EX 发行版推荐 T_EX Live，T_EX Live 推荐通过光盘镜像文件安装。
3. 在更新 T_EX Live 之前，先要卸载旧版的 T_EX Live。
4. 使用 T_EX Live 中的 tlmgr 来安装/更新宏包，不应当随意移除宏包。
5. 多人协作文档推荐使用在线的 Overleaf 或 T_EXPage。
6. 使用命令程序 texdoc 来查看 T_EX Live 中的各种文档。在线用户也可以访问 <https://texdoc.org/index.html>。
7. 使用命令行编译更容易调试，命令行编译也支持更加定制化的编译方式。
8. 各个编译命令的意义见表 1.1，我们在 L^AT_EX 中使用的都是 L^AT_EX 格式的编译命令。中文文档最常用的是 xelatex。

表 1.1: 几种常用的编译命令及其使用的格式和引擎

编译命令	格式	引擎	输出文件格式
latex	L ^A T _E X	T _E X	.dvi
pdflatex	L ^A T _E X	pdfT _E X	PDF
xelatex	L ^A T _E X	X _Ǝ L ^A T _E X	PDF
lualatex	L ^A T _E X	LuaT _E X	PDF
tex	plain T _E X	T _E X	.dvi
pdftex	plain T _E X	pdfT _E X	PDF
xetex	plain T _E X	X _Ǝ L ^A T _E X	PDF
luatex	plain T _E X	LuaT _E X	PDF

扩展阅读

\LaTeX 的历史 高德纳在编写他的七卷本著作 *The Art Of Computer Programming* (TAOCP) 的第二卷时, 对当时糟糕的计算机排版感到震惊, 于是用了将近十年时间编写了 \TeX 程序。

最初的 \TeX 程序本身 (也叫做 IniTeX) 只能处理其提供的原语。这无疑是相当难以使用的。于是, 高德纳按照 \TeX 的宏展开机制编写了一组宏, 叫做 plain TeX 。如果你使用的是 Windows 下的 \TeX Live, 可以在安装目录下找到 plain TeX 的内核:

```
./texmf-dist/tex/plain/base/
```

比如 `C:/texlive/2022/texmf-dist/tex/plain/base/`。Plain \TeX 的内核是非常轻量级的, 其中最基本的文件是 `plain.tex`, 它只有不到 1250 行。

Plain \TeX 的内核很小, 速度也很快。但是这也就意味着, plain TeX 对排版做出的具体设定很少, 也没有结构化的文档写作模式。Leslie B. Lamport 重构了一组宏, 也就是最早的 \LaTeX 。 \LaTeX 中的 \La 取自 Lamport。在 1985 年 Lamport 发布了 \LaTeX 2.09 后, \LaTeX 开发工作主要转移至 Frank Mittelbach 等人^[14]。目前我们使用的版本是 \LaTeX 2 ϵ 。这里的 2 ϵ 表示这一版本比 2 大一点, 但是远没有达到 3——这与数学 ϵ - δ 语言中的 ϵ 意义相同。

如今, \LaTeX 3 也正在开发中。 \LaTeX 3 主要通过宏包的形式使用, 比如 `expl3`、`xparse`、`xpatch` 等宏包。有很多宏包使用了 \LaTeX 3 的格式, 比如提供中文排版支持的 `ctex` 宏包。

除了最常用的 \LaTeX 和 plain TeX 外, 还有另一个格式—— ConTeXt 。 ConTeXt 的内核比 \LaTeX 更大, 使用时无需调用各种宏包, 但是 ConTeXt 相对比较封闭, 编译速度也比较慢。目前期刊投稿广泛使用的是 \LaTeX 格式。

更多关于 \LaTeX 的历史可以参考在线文档 [20]、[14]。

\TeX 引擎 高德纳最早的 \TeX 仅支持 256 个字符, 这也就导致了早期的 \TeX 面对多语言时需要使用一些 “dirty tricks”, 比如切换字符编码和字体编码的 `inputenc` 和 `fontenc` 宏包以及 21 世纪初的 CJK 宏包。这无疑增加了 \LaTeX 的使用难度。

好在 21 世纪初由 Jonathan Kew 发布了全新的支持 Unicode 的引擎—— XeTeX 。 XeTeX 最早仅支持 macOS 的 AAT 字体, 后来在 2005 年开始支持 OpenType 字体。在 2006 年 XeTeX 成功移植到 Windows 和 Linux 后, XeTeX 被收录进 \TeX Live 2007 中。至此, 在 \TeX 中使用 Unicode 字符变得容易起来。

后来, 自 2007 年开始开发的 LuaTeX 引擎被收录进 \TeX Live 2016 中, LuaTeX 不

目前 CJK 宏包已经过时。现在使用 \LaTeX 排版中文推荐在 XeLaTeX 下使用 `ctex` 宏集。

仅可以支持 Unicode，还可以通过 `\directlua` 命令内联 Lua。

现在，使用 L^AT_EX 排版中文主要依赖于 X_YL^AT_EX 和 LuaL^AT_EX。设置中文字体底层主要依赖于 X_YL^AT_EX 下的 `xeCJK` 宏包。使用 L^AT_EX 排版日文主要依赖于 LuaL^AT_EX 下的 `luatexja` 宏包。

驱动 最早的 T_EX 引擎只能在 Xerox 打印机上使用，后来 T_EX 的输出格式改为 `.dvi`。`.dvi` 是一种与设备无关的准页面描述语言，之所以叫“准页面描述语言”而不是页面描述语言，是因为 `.dvi` 不支持嵌入字体或插入图片，且不支持使用颜色。


那么，在 T_EX 中嵌入字体、使用颜色和图片应该通过什么方法呢？答案是使用驱动程序（driver）。DVIPS 是比较早的驱动，它可以将 `.dvi` 转换为 Adobe 的 PostScript 格式。后来，PDF 逐渐流行起来，也诞生了新的驱动——DVIPDF。后来又出现了更好的 DVIPDFM。这两个驱动都可以将 `.dvi` 转换为 PDF，但是它们最多仅支持 256 个字符，很难处理中文等超大字符集的文字。因此，全新的 DVIPDFM_x 出现了。

支持 Unicode 的 X_YL^AT_EX 引擎的输出格式是 `.xdv`（Extended DVI），又需要一个新的驱动将 `.xdv` 转换为 PDF，这就是后来的 XDVIPDFM_x。XDVIPDFM_x 也是 X_YL^AT_EX 编译方式默认的驱动。后来，XDVIPDFM_x 也被兼容进 DVIPDFM_x 中。

在 Visual Studio Code 中使用 L^AT_EX 除了 T_EX studio 之外，微软推出的 Visual Studio Code 编辑器也是不错的选择。可以在下面展示的网址下载 Visual Studio Code：

<https://code.visualstudio.com/Download>

安装后进入“设置”，点击“扩展”（或使用快捷键 `Ctrl + Shift + X`），搜索“L^AT_EX Workshop”，安装即可使用。需要注意的是，这一扩展仅支持 T_EX Live 发行版。

配置默认编译方式，可以进入“设置”→“扩展”→“L^AT_EX”，点击右上角的“”按钮，进入 `settings.json`。在 `settings.json` 中添加以下代码：

```
"latex-workshop.latex.tools": [
  {
    "name": "xelatex",
    "command": "xelatex",
    "args": [ "-synctex=1", "-interaction=nonstopmode",
      "-file-line-error", "%DOC%" ]
  },
],
```

以上设置仅对不需要复杂的参考文献列表和索引的文档有效。