

L^AT_EX 超快入门教程

李天意

2012 年 3 月 11 日

1 它到底是啥

L^AT_EX 本质来说是一种语言，但与其它编程语言不同，它写的源代码将被编译成一篇文章或一本书，而不是一个程序。可能大家更习惯用类似 Word 的软件进行文档的编写，那么它们之间的区别是什么？为什么会产生 L^AT_EX？

事实上，先产生的是 T_EX 语言。其在当时之所以诞生，是因为当时的发明者大牛 [Donald Knuth](#) 在交给出版社让其编辑出版自己写的一本书的时候，发现文章样式尤其是数学公式的排版尤其糟糕，于是就萌发了自己设计一个排版语言软件的想法，于是就产生了 T_EX 语言。T_EX 语言比较初级，作为非信息的学生可能下面这个比喻不好，就好像汇编语言（接近机器语言）那样；于是之后就产生了 L^AT_EX，相当于一个再封包，增加许多比较高级的命令，相对汇编 L^AT_EX 就类似于 C 语言。

正如刚才所说，T_EX 或 L^AT_EX 语言的诞生（下面都直接说 L^AT_EX，意指所有从 T_EX 中衍生出来的科技文档编写语言）主要是为了排版带有数学公式的文档。你可能会说 Word 也可以打数学啊，我用 Mathtype 打公式我也挺好么？的确，直接 Word（比如 2007 版的公式编辑功能）或配合其他插件（如 Mathtype）也可以打公式，但是从美观这个指标来看，远远没有 L^AT_EX 制作的数学公式在字体运用、字符各个指标的位置及各个数学量之间的间隙控制得好，这一点可以自己 Google 搜索 “Word, LaTeX” 看看，尽管我承认现在 Word 就公式而言美观程度是在逐渐随着版本的提高而改善。

我刚才说到就公式而言，是因为 L^AT_EX 设计的理念是，将排版和内容基本上分开管理：排版，指的是正文字体是啥啦、多大啦、标题是什么字体啦、每个章节是什么格式啦、章节编号如何编号啦、段与段之间间隔多少啦、公式与普通文字之间的间隔如何控制啦……而且，就我而言 60% 的 L^AT_EX 及已加载宏包（后面提到）的默认设置就已经很好了，也就是说排版并不需要占你写这篇文章或这本书很大的部分，你只要专心于内容就好了，这就是 L^AT_EX 与 Word 最本质的区别。

在讲 Word 之前，我首先要承认 Word 的确具有将排版与内容想分开管理的模式，但其管理得并不严密，因为 Word 制作文档的理念是所见即所得，也就是说作者仍然可以或需要修改或设计排版。撇开“排版与内容分开管理”不说，L^AT_EX 能最发挥其特色的地方还是体现在科技文档的系统排版上。科技文档，当然需要一个相对固定的格式（尤其是章节层次），通常定下来之后就不会再发生什么很大的变化。再加上科技文档中的数学公式比较多，L^AT_EX 可以相对较好地控制页面的留白处，做到数学与文字的和谐（这个词不用取引申含义）。

举一个 L^AT_EX 文档的大致框架的例子，就这篇文章好了：

```
1 \documentclass{tycn_art}
2 \usepackage{tymaths}
3 \usepackage{listings}
4 \palatino
5 \pdfptac{LaTeX超快入门教程}{李天意}{Blue}
6 \title{\LaTeX 超快入门教程}
7 \author{李天意}
8
9 \begin{document}
10 \maketitle
11 \section{它到底是啥}
12
13 ....
14
15 \end{document}
```

第 1 行的 `\documentclass{tycn_art}` 为定义文档的类型，即到底是一篇小文章还是一篇报告或一本书。类型 `tycn_art` 是我自己设置的专门为了写中文文章的类型，添加了许多针对中文的修改，一劳永逸，之后只要加载这个类型就可以了。

第 2 行的 `\usepackage{tymaths}` 为加载我自己设置的数学宏包。即加载了一些写数学用到的宏包和定义一些可能会用到的命令，下面会提到。第 3 行的 `listings` 是用来显示各种编程源代码的宏包，你看到的上面那个灰框框就是它做出来的。由于这个宏包不是经常用，所以我没有放在自己设置文档类型中（当然更没有放在我自设的数学宏包中，因为这个和数学没啥关系），所以我单独加载。

第 4 行的 `\palatino` 是我自己在文档类型宏包中定义的命令，意思是使用 **Palatino** 字体，一个我很喜欢的字体，本文的英文和数学字体就是它，当然各种源代码采用等款字体 **Consolas**。

第 5 行的 `\pdfptac` 也是我自己定义的一个命令，意思是设置 PDF 文档的属性，在这里设置了标题 **T**、作者 **A** 和超链接的颜色 **C**。第 6 行和第 7 行设置本文章的标题和作者信息。第 1 到第 7 行称作 L^AT_EX 文章的导言区，因为下面就开始正文了，由第 9 行的 `\begin{document}` 作为标志。

开始正文后，第 10 行的 `\maketitle` 进行制作“封面”，即打印标题、作者和时间信息。随后的 `\section` 命令就开始新建一个章节，开始正文。再之后如果要再开一个章节，就再 `\section` 即可。文档的最后用 `\end{document}` 标志。

如果上手 L^AT_EX 貌似是一个初心者要琢磨很长时间的问题，我记得当时在白天中也摸索也很久。说白天是因为网上资料真得很多，但问题是近来 L^AT_EX 发展很快，一些陈旧的方法已经不再实用，这样新老文章在一起也不知道到底应该选择什么...其实有点过了，因为其实大致 L^AT_EX 的框架已经固定下来，变的，只是一些宏包和选项。下面开始细讲。

2 名词解释

前面已经提到很多 L^AT_EX 的专有名词，下面一一来解释。

文档类型 指的是你目前写的文档到底是一个什么类型（废话），L^AT_EX 的标准类有：文章类 **article**，报告类 **report** 和书籍类 **book**，一般来说够用了，不过像我写动机信还会用到 **letter** 类等等。在 L^AT_EX 源代码中用 `\documentclass{class}` 来指明所用文档类。你既可以用已经设置好的标准类，也可以用其他宏包提供的类型（如 C_T_EX 宏包提供的专门写中文的文章类 **ctexart** 等），当然也可以自己设置一个文档类。

宏包 宏包就类似于一个包括了很多设置和新命令的大礼包，当你需要什么的时候就去加载一个合适的宏包。比如如果会用到一些数学符号比如要输入矩阵，那么就加载 **amsmath** 宏包，它提供了 **bmatrix** 环境就用来输入矩阵。我自己也做了几个宏包，其实就是把自己平时常常会用到的命令放到一个文件（宏包）之中，以后需要用到只要加载这个宏包即可。宏包的选择是入门 L^AT_EX 的一个难点...因为不知道到底什么宏包好，什么不好。调用宏包使用 `\usepackage` 命令。

环境 在正文中可能会使用到某些环境，比如你想将一段文章居中排，那么可以用到 **center** 环境

```
1 \begin{center}
2 你想要居中的东西
3 \end{center}
```

来实现。环境就是一个 **begin** 一个 **end**，放在里面的东西就处于这个环境之下，具有特殊的排版效果。事实上写这篇文章的时候，上面你看到的这个放源代码的灰底框架就是使用了 **listings** 宏包提供的 **lstlisting** 环境，在这个环境中的东西会自动会编程源代码的样子。

命令 ...所谓命令其实就是 L^AT_EX 的语法，L^AT_EX 本身以及你加载的文档类、宏包都会告诉你它都定义了哪些命令供你使用。L^AT_EX 的命令都由反斜杠 `\` 开头，且全英文不允许数字。比如我要输出 L^AT_EX 的标志，我就输入 `\LaTeX`，编译后就成了 L^AT_EX。你也可以自己定义一个命令，因为可能这个命令会经常使用。

导言区 所有在 `\begin{document}` 之前的命令都处于导言区，在导言区中我们定义文档类型，加载需要用到的宏包和设置一些属性、定义自己的命令。

3 下载 LaTeX 处理程序

L^AT_EX 只是一种编程语言，所以我们需要编辑器和编译器，好在现在 L^AT_EX 提供了很多发行版，意思就是将编译器、很多实用宏包、编辑器和其他辅助软件等捆绑在一起发行。

Windows 用户推荐大家下载 [MiKTeX](#)，其宏包管理系统“飞行下载”很实用，即当你编译的时候发现你需要调用的那个宏包还没有下载（还没有在你电脑上），它会自动帮你去下载。它默认使用 [Texworks](#) 作为编辑器，个人认为是一个小巧但实用的编译器，且默认编码为 UTF-8，适合打中英法文。

Mac 用户推荐大家下载 [MacTeX](#)，事实上它是基于 [TeXLive](#) 发行版的，但再加上了几个适合苹果的辅助软件。编辑器推荐大家用 [TeXShop](#)，很漂亮而且也很实用，事实上 [Texworks](#) 就是根据它向 Windows 迁移的。

4 处理中文

T_EX 发明得很早，在那时没有怎么考虑怎么整合亚洲文字...好在现在已经有一套完整且较完美的解决方案了。

处理中文的第一步是选择编译语言。大家可能会觉得奇怪，说了那么多 L^AT_EX 难道编译语言不是 T_A 么？事实上是又不是。因为如今基于 L^AT_EX 的新语言很多，它们往往更实用，于是直接采用 L^AT_EX 进行编译的情况基本上没有。在这里我推荐大家使用 [XeLaTeX](#) 进行编译（可以在编辑器中选择），这个语言有 L^AT_EX 提供的新命令（相比 T_EX），而且能够很好地支持多语言，比如一会中文一会法语，其默认编码也为 UTF-8。基于 [XeLaTeX](#) 也诞生了很多宏包，比如下面会提到的处理中文的 C_TE_X 宏包。

在 [XeLaTeX](#) 编译语言下，C_TE_X 宏包将配合专门用来设置中文的 [xeCJK](#) 宏包一起来调整中文文章的排版效果。如果你要写一篇有中文的文档，那么只需要加载 C_TE_X 宏包提供的 [ctexart](#)、[ctexrep](#) 和 [ctexbook](#) 文档类即可，一般它会自动调整字体并成功显示中文。不过有时候，比如在 Mac 下，可能需要自己设置，那么其实也很简单，只需要在导言区加上比如

```
1 \setCJKmainfont[BoldFont=SimHei,ItalicFont=KaiTi]{SimSun}
2 \setCJKmonofont{FangSong}
```

之类的命令即可，意思就是将亚洲语言的字体默认设置为宋体，其中粗体采用黑体、斜体采用楷体、等宽字体（源代码字体）采用仿宋。当然前提是你电脑上要有这些字体，Windows 用户可能没问题，Mac 的话可以拷.....

5 数学

正如前面所言，L^AT_EX 一个很大的优势就是数学公式。我们可以使用行内公式 $f: [0, 1] \rightarrow \mathbb{C}$ ，也可以使用独立公式

$$J_f(x) \stackrel{\text{def}}{=} [df_x] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(x) & \frac{\partial f_1}{\partial x_2}(x) & \cdots & \frac{\partial f_1}{\partial x_n}(x) \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_m}{\partial x_1}(x) & \frac{\partial f_m}{\partial x_2}(x) & \cdots & \frac{\partial f_m}{\partial x_n}(x) \end{bmatrix}$$

个人觉得比 Word 啥的好看多了，不是么.....

行内公式用 文字\$公式\$文字 来实现，即用美元符号包住公式¹。独立公式根据情况的不同可以采用不同的环境。上面的用

```
1 \[
2 J_f(x)\eqdef\mathrm{f}_x=\begin{bmatrix}
3 \mathrm{d}f_1(x_1)(x) & \mathrm{d}f_1(x_2)(x) & \cdots & \mathrm{d}f_1(x_n)(x) \\
4 \vdots & \vdots & \vdots & \vdots \\
5 \mathrm{d}f_m(x_1)(x) & \mathrm{d}f_m(x_2)(x) & \cdots & \mathrm{d}f_m(x_n)(x) \\
6 \end{bmatrix}
7 \]
```

来实现，是不会显示公式编号的。如果用

```
1 \begin{equation}
2 \sum_{n=-\infty}^{+\infty} \hat{f}(n/T) = T \sum_{n=-\infty}^{+\infty} f(nT)
3 \end{equation}
```

那么效果就是

$$\sum_{n=-\infty}^{+\infty} \hat{f}(n/T) = T \sum_{n=-\infty}^{+\infty} f(nT) \quad (1)$$

当然，还有很多情况，比如多行公式。我这里就不细讲了，大家可以参考 `math-mode` 或 `The LaTeX Mathematics Companion` 这两本书。

6 写在后面

这篇文章实在是太简略了，在下载、使用 L^AT_EX 的过程中会遇到无数问题，但本文可能一个都无法回答。不是因为我懒（可能也是），而是因为问题太多、太杂，而解决方案层出不穷。不管怎样，给出学习 L^AT_EX 旅程中下面的路。

多逛逛 [CTeX 论坛](#)，看看里面的置顶贴，如果有疑问也可以在里面发帖。如果我不小心看到你发的一个傻问题我会好心地回答的。里面也分享了很多教程，不管是一般的还是针对某个特定主题的（比如数学）。其中 `lshort` 这本书在我一开始学的时候帮了不少忙，可以看看这本书。

希望大家学会用 `Google`，上面提到的所有书都可以搜索到。如果有什么私人问题，如果相信我的回复速度的话，那么就给我发邮件，最好不要什么太傻的问题.....

tianyikillua(at)gmail.com



¹原因是发明者认为数学很贵.....