

寻求帮助

2020 年 9 月 16 日

寻求帮助

什么知识会过时？什么不会？

$\text{T}_{\text{E}}\text{X}$ 是排版语言，也是广泛使用的软件，并且不断在发展中；因此，总有一些东西会很快过时。作为学习 $\text{T}_{\text{E}}\text{X}$ 的人，免不了要看各种各样的书籍、电子文档和网络论坛上的只言片语，因此了解什么知识会迅速过时，什么知识不会是十分重要的。

最稳定的是关于 Primitive $\text{T}_{\text{E}}\text{X}$ 和 Plain $\text{T}_{\text{E}}\text{X}$ 的知识，也就是 Knuth 在他的《The TeXbook》中介绍的内容。因为 $\text{T}_{\text{E}}\text{X}$ 系统开发的初衷就是稳定性，要求今天的文档到很久以后仍可以得到完全相同的结果，因此 Knuth 限定了他的 $\text{T}_{\text{E}}\text{X}$ 语言和相关实现的命令、语

法。这些内容许多年来就没有多少变化，在未来的一些年里也不会有什么变化。Primitive $\text{T}_{\text{E}}\text{X}$ 和 Plain $\text{T}_{\text{E}}\text{X}$ 的知识主要包括 $\text{T}_{\text{E}}\text{X}$ 排版的基本算法和原理，盒子的原理，底层的 $\text{T}_{\text{E}}\text{X}$ 命令等。其中技巧性的东西大多在宏包设计中，初学者一般不会接触到很多；而基本原理则是常常被提到的，譬如， $\text{T}_{\text{E}}\text{X}$ 把一切排版内容作为盒子（box）处理。

相对稳定的是关于基本 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ 的知识，也包括围绕 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ 的一些核心宏包的知识。 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ 是自 1993 年以来的一个稳定的 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 版本，直到最近的一次修订（2005 年）都没有大的变动。 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 的下

一个计划中的版本 L^AT_EX3 遥遥无期，在可预见的将来，L^AT_EX 2_ε 不会过时。L^AT_EX 2_ε 的知识是目前大部分 L^AT_EX 书籍的主体内容。关于 L^AT_EX 的标准文档类 (article、report、book、letter、slide 等)，关于基本数学公式的输入，文档的章节层次，表格和矩阵，图表浮动体，LR 盒子与段落盒子…… 这些 L^AT_EX 的核心内容都是最常用的，相对稳定的。与 L^AT_EX 2_ε 相匹配的核心宏包，如 graphics(x)、ifthen、fontenc、doc 等，也同样是相对稳定的。还有一些被非常广泛应用的宏包，如 amsmath 系列，也可以看作是相对稳定的。

简单地说，关于基本 T_EX/L^AT_EX 的语言，都是比

较稳定的。与之对应，实现或者支持 $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 语言的软件，包括在 $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 基础上建立的新的宏，都不大稳定。

容易过时的是关于第三方 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 宏包的知识、第三方 $\text{T}_{\text{E}}\text{X}$ 工具的知识，以及新兴 $\text{T}_{\text{E}}\text{X}$ 相关软件的知识等。 $\text{T}_{\text{E}}\text{X}$ 和 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 语言是追求稳定的；但无论是宏包还是工具，作为不断更新软件，它们是不稳定的。容易过时的技术很多，而且现在广泛地出现在几乎所有 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 文档之中，因此需要特别引起注意：宏包的过时的原因可能是宏包本身的升级换代带来了新功能或不兼容，也可能是同一功能的更新更好的宏包代替了旧的

宏包。前者的典型例子比如绘图宏包 PGF/TikZ，现在的 3.00 版功能十分强大，和旧的 1.1x 版相差很大，和更旧的 0.x 版本则几乎完全不同；后者的典型例子比如 caption 宏包先是被更新的 caption2 宏包代替，后来 caption 宏包更新又使得 caption2 宏包完全过时。——安装更新的发行版可以避免使用过旧的宏包；认真阅读宏包自带的文档而不是搜索得到的陈旧片断可以避免采用过时的代码。

工具过时的主要原因也是升级换代和被其他工具替换。前者的典型例子是编辑器 WinEdt 在 5.5 以后的版本支持 UTF-8 编码，而旧版本不支持；后者的典型

例子是中文字体安装工具从 GBKFonts 到 xGBKFonts 到 FontsGen 不断被取代。图形插入是一个在 T_EX 实现、宏包与外围工具方面都更新很快的东西。在过去，最常用的输出格式是 PS (PostScript) 格式，因此插入的图像以 EPS 为主流。使用 Dvips 为主要输出工具，外围工具有 GhostScript、bmeps 等等，相关宏包有 graphics 等，相关文档如《LaTeX2e 插图指南》。

但凡提及“L^AT_EX 只支持 EPS 图形”的，就是这个过时的时代的产物。事实上 T_EX/L^AT_EX 并不限定任何图形格式，只不过是当时的输出格式 (PS) 和工具 (Dvips) 对 EPS 情有独钟而已。后来 PDF 格式成为

主流，pdfTeX、DVIPDFM、DVIPDFMx、XeTeX 等工具则主要支持 PDF、PNG、JPG 格式的图形，涉及一系列工具如 ImageMagick、ebb 等。

值得特别提出注意的就是，中文处理也一起是更新迅速、容易过时的部分。而且因为中文处理一直没有一个“官方”的“标准”做法，软件、工具、文档以及网上纷繁的笔记也就各式各样。从八十年代开始的 CCT 系统、天元系统，到后来的 CJK 方式，到近来的 XeTeX、LuaTeX 方式，中文处理的原理、软件、宏包、配置方式等都在不断变化中。目前我们推荐使用 XeTeX 处理中文，几年之后可能会转向 LuaTeX 方式。

而且新方法确实也越来越简单、越来越直接。

如何提问

使用 T_EX 过程遇到问题，可以先查阅宏包说明文档，只需要在命令行下输入

```
texdoc <package name>
```

即可。例如，

```
texdoc amsmath
```

可以得到 amsmath 宏包的使用说明。

也可以到网上搜索。若还找不到答案，可以到

<https://www.latexstudio.net/>

提问。

你可能还不了解如何提问才能把问题说清楚。基本的要求是**清晰**：

如果你在问关于代码的问题，那么你应该给出：

1. 能直接编译就产生重现这个问题的完整代码。但在完整的基础上代码最好经过删节，不要出现整页的信息。如果效果不正确，那么给出：
2. 理想中的效果之清楚描述，并且把出问题的效果截图，或将 PDF 文件上传。

如果代码还产生了错误，那么应该再给出：

3. 在 log 文件中关于编译错误的信息，为了正确说明问题，上下文完整一些比简洁更重要。其中 log 文件是在你写作文档所在目录编译时产生的一个与源文件同名、后缀名为 .log 的文件。编译信息，请把出错之前的若干行信息在正文中贴出来，如果必要再将整个 log 文件附件上传，但不要对编译错误贴图。

4. 如果对选择代码和错误信息有困难，请把所涉及的文件打包成 .zip、.rar 或 .tar.gz 等格式的文件，用附件的形式发上来。打包的内容至少应该包括源文件和 log 文件。

不好的提问

1. 我的 $\text{T}_\text{E}\text{X}$ 文件无法编译，救命啊！
2. 我一用某某宏包就报错，删除这个包就好了，是否这个包有 Bug？