

作者：milksea

新手请先读我

本帖不是一个入门教程，不讲 $\text{T}_{\text{E}}\text{X}$ 系统的配置安装，也不讲具体的 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 代码。这里仅仅试图以一些只言片语来解释：进入这个门槛之前新手应该知道的注意事项，以及遇到问题以后该去如何解决问题。

China $\text{T}_{\text{E}}\text{X}$ 整理

目 录

1 什么是 $\text{T}_\text{E}\text{X}/\text{L}\text{A}\text{T}_\text{E}\text{X}$ ，我是否应该选择它	1
2 周围的人不会，使用 $\text{L}\text{A}\text{T}_\text{E}\text{X}$ 文档不方便与人交流怎么办？	2
3 我该安装什么发行版？	2
4 我该用什么编辑器？	3
5 我该去哪里寻找答案？	7
6 我应该看什么 $\text{L}\text{A}\text{T}_\text{E}\text{X}$ 读物？	8
7 新手不建议做的几件事情 –neals	9
8 什么知识会过时？什么不会？	10

1 什么是 $\text{T}_\text{E}\text{X}/\text{L}\text{A}\text{T}_\text{E}\text{X}$ ，我是否应该选择它

$\text{T}_\text{E}\text{X}$ 是最早由高德纳 (Donald Knuth) 教授创建的一门标记式宏语言，用来排版科技文章，尤其擅长处理复杂的数学公式。 $\text{T}_\text{E}\text{X}$ 同时也是处理这一语言的排版软件。 $\text{L}\text{A}\text{T}_\text{E}\text{X}$ 是 Leslie Lamport 在 $\text{T}_\text{E}\text{X}$ 基础上按内容/格式分离和模块化等思想建立的一集 $\text{T}_\text{E}\text{X}$ 上的格式。

$\text{T}_\text{E}\text{X}$ 本身的领域是专业排版 (即方正书版、InDesign 的领域)，但现在 TeX/LaTeX 也被广泛用于生成电子文档甚至幻灯片等， $\text{T}_\text{E}\text{X}$ 语言的数学部分偶尔也在其他一些地方使用。但注意 $\text{T}_\text{E}\text{X}$ 并不适用于文书处理 (MS Office 的领域，以前和现在都不是)。

选择使用 $\text{T}_\text{E}\text{X}/\text{L}\text{A}\text{T}_\text{E}\text{X}$ 的理由包括：

- 免费软件；
- 专业的排版效果；
- 是事实上的专业数学排版标准；

- 广泛的西文期刊接收甚或只接收 LaTeX 格式的投稿;

.....

不选择使用 $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 的理由包括:

- 需要相当精力学习;
- 图文混合排版能力弱;
- 仅流行于数学、物理、计算机等领域;
- 中文期刊的支持较差;

.....

请尽量清醒看待网上经常见到的关于 $\text{T}_{\text{E}}\text{X}$ 与其他软件的优劣比较和口水战。在选择使用或离开之前, 请先考虑 $\text{T}_{\text{E}}\text{X}$ 的应用领域, 想想它是否适合你的需要。

参考:

- ✍ CTeX 主页基础知识: <http://www.ctex.org/BasicKnowledge> (是的, 有些新潮的内容有点老)
- ✍ CTeX-FAQ: <http://www.ctex.org/CTEXFAQ> (是的, 这个文档也不新)
- ✍ The UK $\text{T}_{\text{E}}\text{X}$ FAQ (英文文档。请在自己电脑上或 CTAN 寻找)

2 周围的人不会, 使用 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 文档不方便与人交流怎么办?

这个并没有什么很好的办法, 各种转换软件的效果都不是很好。

要给导师修改的文章, 可以使用 Acrobat PDF 批注的方式, 或是打印出来在纸上修改。如果环境实在不协, 不如不用 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 。

3 我该安装什么发行版?

我什么都不知道/大多数人: $\text{T}_{\text{E}}\text{X}$ Live 2011 或 $\text{C}_{\text{T}}\text{E}_{\text{X}}$ 套装 2.9 以后版本

我使用 ConTeXt: ConTeXt Minimals

我还听说有 XXX: 不妨自己试试

注 3.1

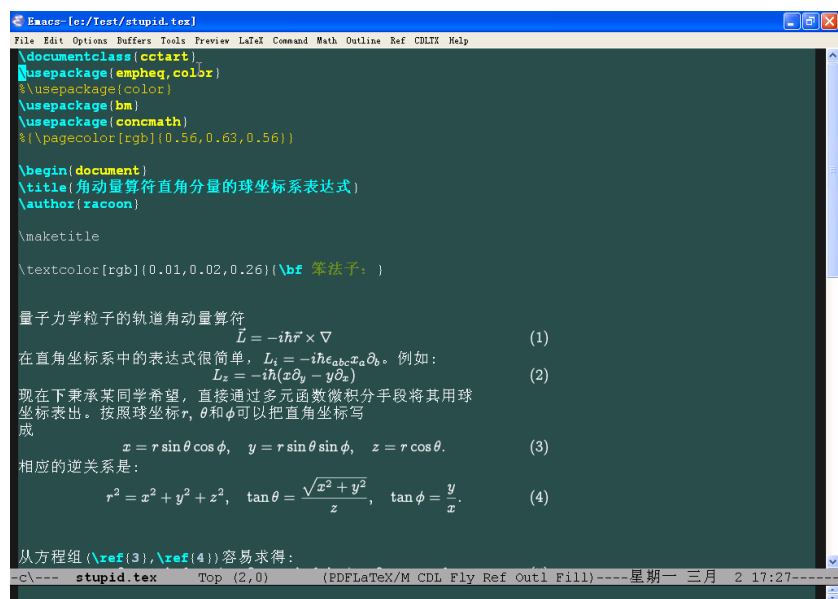
安装后建议使用 $\text{T}_{\text{E}}\text{XLive}$ Manager 或 $\text{MiK}_{\text{T}}\text{E}_{\text{X}}\text{Maintenance}$ 连接 CTAN 在线更新。部分中文支持的宏包可能需要更新后才能确保正常使用。

- ✍ $\text{T}_{\text{E}}\text{XLive}$ 的光盘镜像与在线安装文件都可以在 CTAN（后详）找到。
- ✍ TUG 发布页 (<http://www.tug.org/texlive/>)
- ✍ 中文文档 (<http://www.tug.org/texlive/doc/texlive-zh-cn/texlive-zh-cn.pdf>)
- ✍ $\text{C}_{\text{T}}\text{E}_{\text{X}}$ 套装: <http://www.ctex.org/CTeXDownload>

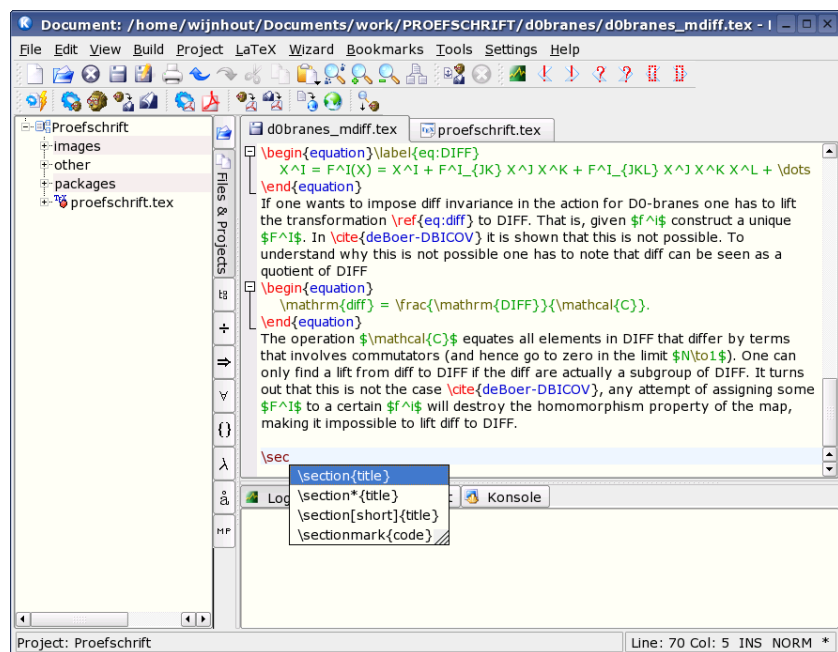
4 我该用什么编辑器？

编辑器功能有简有繁，特色不一，从简单的纯文本编辑器到繁复的 Emacs，因人而异。基本功能有语法高亮、方便编译预览就很好了，扩充功能和定制有无限的可能。初学者可以使用功能简单、使用方便的专用编辑器，如 TeXWorks、Kile、WinEdt 等，或者类似所见即所得功能的 LyX；熟悉的人可以使用定制性更强的 Notepad++、SciTE、Vim、Emacs 等。这方面的介绍很多，一开始不妨多试几种，找到最适合自己的才是最好的。

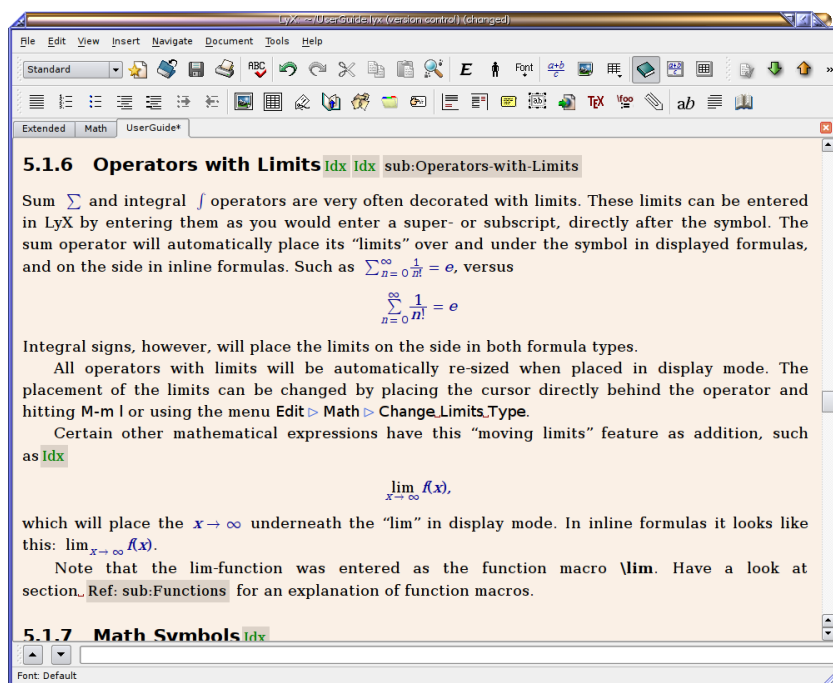
另外提醒一句，编辑器只是工作的助力，不必把它看得太重。一些编辑器有极为繁杂的功能，一些编辑器常常会引来黑客们的论战（如 Emacs 与 Vim）。为工作，别为这些浪费太多精力，适用即可。



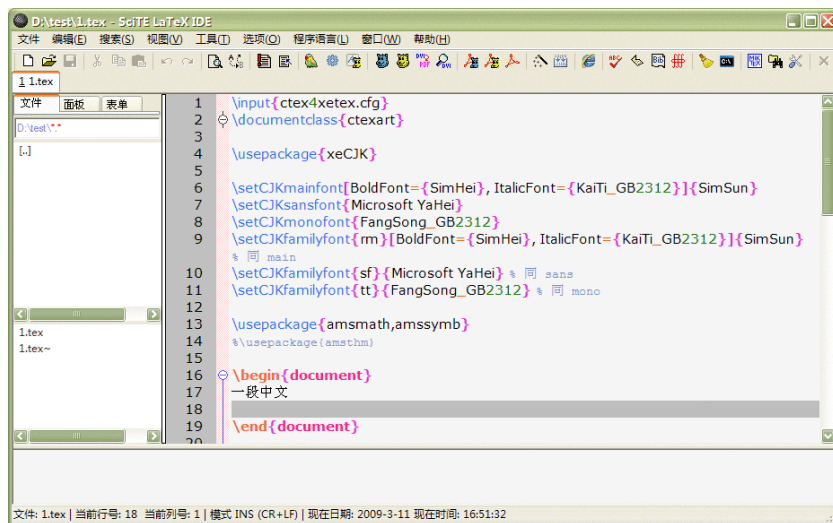
EMacs 界面 (racoon 网友截图)



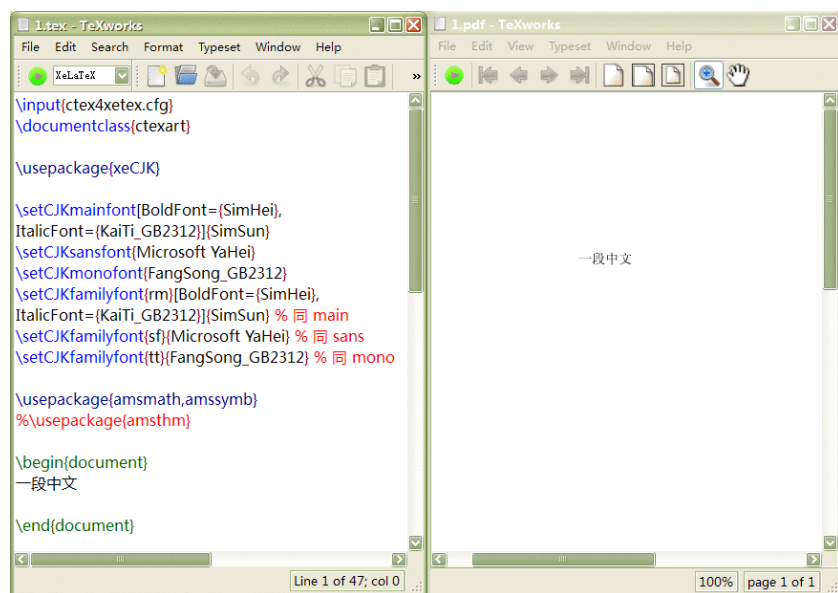
Kile 官网截图



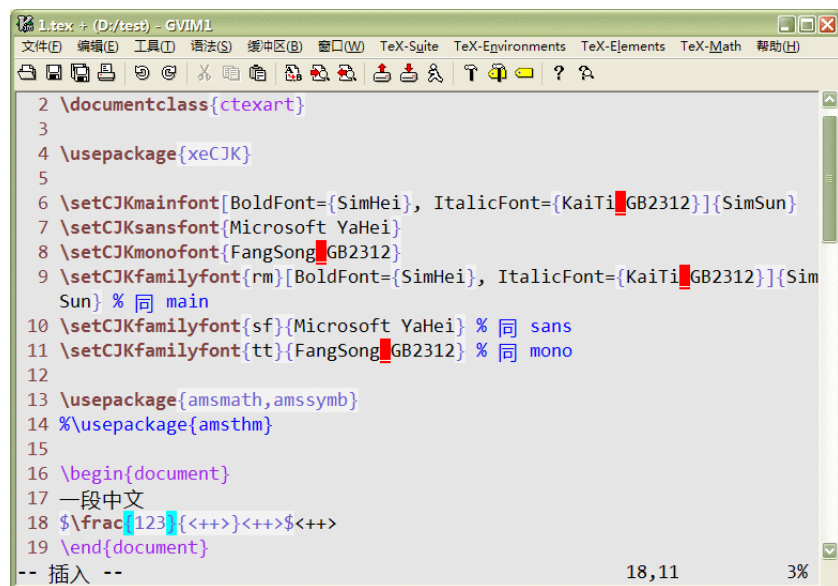
LyX 官网截图



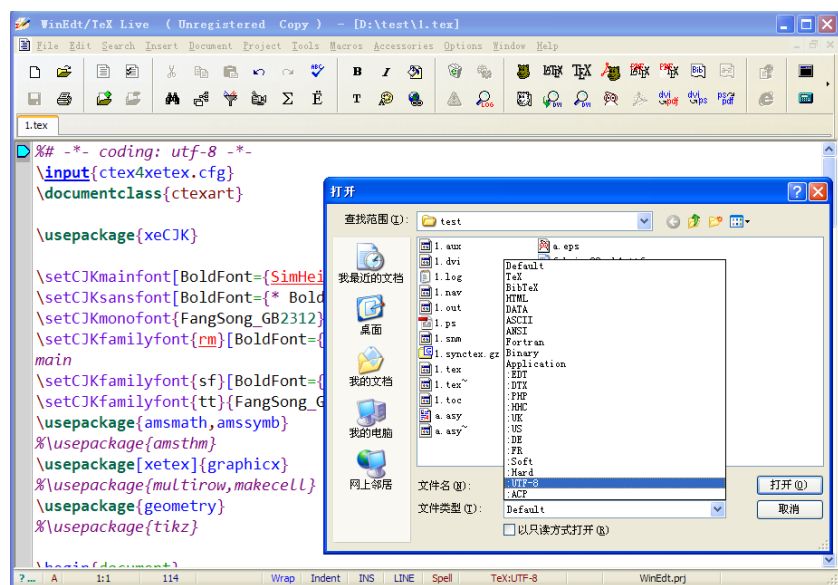
专用版 SciTE (LaTeX IDE) 截图



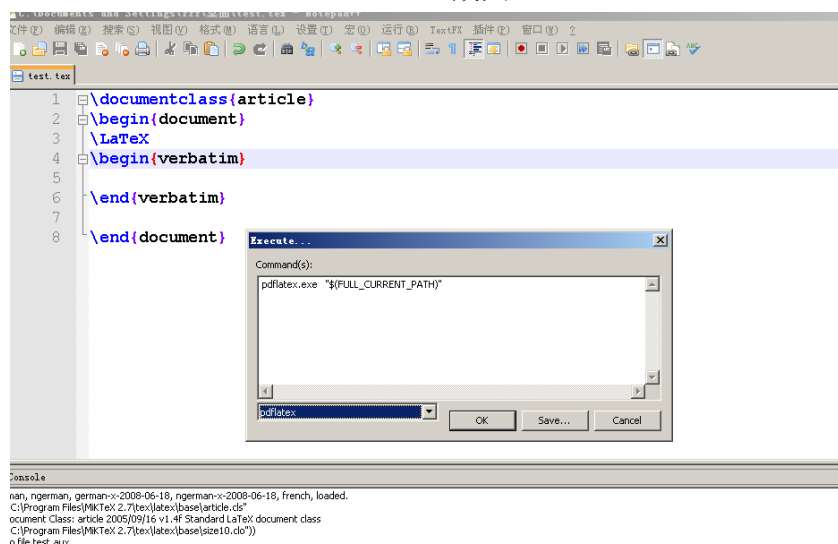
TeXWorks 截图



Vim + LaTeX suite 截图



WinEdt 截图



Notepad++ 界面（cvgmt 截图）

5 我该去哪里寻找答案？

- 0、绝对的新手，先读完一本入门读物，了解基本的知识。
- 1、**无论如何，先读文档！**绝大部分问题都是文档可以解决的。

2、再利用 Google 搜索，利用 (bbs.ctex.org) 版面搜索。

3、清楚、聪明地提出你的问题。

参考：

《提问的智慧》<http://www.beiww.com/doc/oss/smart-questions.html>

6 我应该看什么 \LaTeX 读物？

这不是一个容易回答的问题，因为有许多选择，也同样有许多不合适的选择。这里只是选出一个比较好的答案。更多更详细的介绍可以在版面和网上寻找（注意时效）。

近两年 \TeX 的中文处理发展很快，目前没有哪本书在中文处理方面给出一个最新进展的合适综述，因而下面的介绍也不主要考虑中文处理。

1. 我可以阅读英文

- (a) 我要迅速入门：ltxprimer.pdf (LaTeX Tutorials: A Primer, India TUG)
- (b) 我要系统学习：A Guide to \LaTeX , 4th Edition, Addison-Wesley 有机工业出版社的影印版（《 \LaTeX 实用教程》）
- (c) 我要深入学习：要读许多书和文档， \TeX book 是必读的
- (d) 还有呢？去读你使用的每一个宏包的说明文档
- (e) 还有许多专题文档，如讲数学公式、图形、表格、字体等

2. 我更愿意阅读中文

- (a) 我要迅速入门：lnotes.pdf (\LaTeX Notes, 1.20, Alpha Huang)
- (b) 我要系统学习：《 \LaTeX 2e 科技排版指南》，邓建松（电子版）如果不好找，看《 \LaTeX 入门与提高》第二版，陈志杰等
- (c) 我要深入学习： \TeX book.pdf (特可爱原本， \TeX book 的中译，xi-anxian)
- (d) 还有呢？英语，绝大多数 \TeX 资料还是英文的

7 新手不建议做的几件事情 –neals

- 不了解情况下复制他人代码：代码最好都是自己写，每句话啥意思要自己清楚，同时我也建议各位同学，给别人解决问题的时候，不要从自己文档里面咔嚓一段代码上来，最好是说明用哪个宏包，有啥用处，剩下的是提问者看文档，相信各位在每次帮助他人的同时更巩固了对某些常用宏包的理解
 - 有一点是特别让人不爽的，就是用 `mathtype` 还是什么然后生成 `tex` 代码，忽然有啥问题了，就划拉下来问怎么办。怎么说呢，一般有能力解决的人是不会痴迷这种代码的，好心的就让你看看 `mathmode` 好好学学，普通点的就直接看下一篇。在此忠告：公式一定要自己手敲。
- 求助模板相关问题：除了投稿，尽量不要用现成的模板，因为模板千差万别，良莠不齐，出了什么问题也较难定位。如果真的喜欢某个文档格式而用该模板，请确保你知道模板作者的联系方式，直接向他提问是最有效的解决途径，他人用这个模板而又这么巧看到了你的问题，又那么巧知道问题在哪里，又这么巧有空帮你，这就不是自己能控制的了。不用该模板的用户一般对这种问题是没有兴趣看一看的。
- 毫无基础就开始写文章：我不是吓唬大家，不看点书、做点练习就想直接写书呢不是一种主流做法。花个个把礼拜学习未必使你的文章结束得晚。我经常看到一些代码中充斥着 `\hskip \quad \sim \indent` 甚至无数的 `\` 等用来对齐或者换行，嗯，当时的情况就是这样的。这样写文章难道不累吗。我这里就不介绍应该如何写简练又有意义容易掌控的代码，看书吧。
- 试图修改 \LaTeX 的默认表现：比较有代表性的是浮动体等的标题等，一会儿说不要浮动了，一会儿说 `figure 1.2` 要改成中英文双语版而且后面的冒号要改成五角星，最麻烦的还是双栏下面一浮动咋跑下一页去了...，等等等等，不要管它们，随它们的便，这是最好的处理方

式。

8 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 $\text{T}_{\text{E}}\text{X}$ book**》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 **$\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X} 3$** 遥遥无期，在可预见的将来， **$\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$** 不会过时。 **$\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$** 的知识是目前大部分 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 书籍的主体内容。关于 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 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 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 的核心内容都是最常用的，相对稳定的。与 **$\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$** 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 $\text{T}_{\text{E}}\text{X}/\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 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 \LaTeX 宏包的知识、第三方 \TeX 工具的知识，以及新兴 \TeX 相关软件的知识等。 \TeX 和 \LaTeX 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 \LaTeX 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 2.00 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 \TeX 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“ \LaTeX 只支持 EPS 图形”的，就是这个过时的时代的产物。事实上 \TeX / \LaTeX 并无限定任何图形格式，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，pdf \TeX 、DVIPDFM、DVIPDFMx、X \TeX 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显得相当混乱。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 X_YTeX、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。