

## TeX 简介

古来圣贤皆寂寞，唯有饮者留其名

沧海桑田，UNIX 的世界在风雨飘摇中悄悄变换了容颜

从传统工具到 GNU 工具，从 4.4BSD-Lite 到 FreeBSD，从 Minix 到 Linux，从 VI 到 VIM.....往日的 UNIX 已经不复存在

唯一历久弥新的，只有 UNIX 的风骨；或者还有，TeX？

### 为什么要用 TeX

TeX 足够强大，可以满足使用者的绝大多数要求。尤其在排版质量、数学公式、图形生成、结构化控制等方面，几乎无出其右者

尽管学习 TeX 相对困难，但是学习的收益比成本要大得多

TeX 尤其适合需要打印的文档。恐怕你不能够在办公室里使用诸如 DocBook、reStructuredText 之类的格式写一份文件，然后把它拿给你的老板

### 顺流、逆流

现代文档工具，倾向于分离内容与样式。而内容部分，则用结构化的方式进行梳理，对于规模比较大的文档，这是最佳的方案。

Knuth 的 TeX 本身没什么结构控制，它就像一套排版领域的"汇编语言"，大约 300 多个标记。TeX 在处理一份文档时，就是把整篇文档当作一个很长很长的字符串一口吞下去，然后消化掉，最后排出 dvi 文件（现代的 TEX 输出的是 PDF）。

由于 TeX 支持宏扩展，为了方便排版，Knuth 做了一个 Plain TeX，对 TeX 标记进行了一些功能逻辑上的封装。不过，还是很结构化。

LaTeX 是较早尝试对 TeX 进行结构化封装的宏包，它可以明确地实现样式与内容的分离。不过，LaTeX 本身对排版方面的内建支持太少，需要许多其它宏包的支持，而那些宏包不是统一开发的，它们之间经常出现冲突。

ConTeXt 是对 TeX 进行结构化封装的新尝试，比 LaTeX 更加结构化，对排版方面的内建支持很好。而且 ConTeXt 对 XML 的支持相对完善，甚至可以直接用 ConTeXt 的 XML 语法来写文档<sup>[63]</sup>。

总体说来，LaTeX 是科技论文事实上的标准，如果要排论文，就用 LaTeX；而作个人的文档、非正式散布的文档、演示文档，推荐用 ConTeXt

---

<sup>[63]</sup> DocBook与ConTeXt之间可以无缝转换