

## Preface

Many people discover L<sup>A</sup>T<sub>E</sub>X after years of struggling with wordprocessors and desktop publishing systems, and are amazed to find that T<sub>E</sub>X has been around for over 30 years and they hadn't heard of it. It's not a conspiracy, just 'a well-kept secret known only to a few million people', as one anonymous user has put it.

Perhaps a key to why it has remained so popular is that it removes the need to fiddle with the formatting while you write. Playing around with fonts and formatting is highly attractive to new computer users, and great fun for a while, but it is completely counter-productive for the serious author or editor who needs to concentrate on actual *writing* — ask any journalist or professional writer. 'Best-guess' estimates by experts in the field of usability engineering are that average computer users spend up to 50% of their time fiddling with the formatting rather than thinking or writing — and this is with the so-called 'office productivity packages' that major manufacturers peddle to their clients!

A few years ago a new L<sup>A</sup>T<sub>E</sub>X user expressed concern on the `comp.text.tex` newsgroup about 'learning to write in L<sup>A</sup>T<sub>E</sub>X'. Some excellent advice<sup>2</sup> was posted in response to this query, which I reproduce with permission below (the bold text is my own emphasis):

No, the harder part might be *writing*, period. T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X is actually easy, once you relax and stop worrying about appearance as a be-all-and-end-all. Many people have become 'Word Processing Junkies' and no longer 'write' documents, they 'draw' them, almost at the same level as a pre-literate 3-year old child might pretend to 'write' a story, but is just creating a sequence of pictures with a pad of paper and box of *Crayolas* -- this is perfectly normal and healthy in a 3-year old child who is being creative, but is of questionable usefulness for, say, a grad student writing a Master's or PhD thesis or a business person writing a white paper, etc. For this reason, I strongly recommend *not* using any sort of fancy GUI 'crutch'. Use a plain vanilla text editor and treat it like an old-fashioned typewriter. Don't waste time playing with your mouse. Note: I am *not* saying that you should have no concerns about the appearance of your document, just that you should *write* the document (completely) first and tweak the appearance later...*not* [spend time on] lots of random editing in the bulk of the document itself.

[Heller, *New To L<sup>A</sup>T<sub>E</sub>X... Unlearning Bad Habits* (11 March 2003)]

---

<sup>2</sup> `news:comp.text.tex/MPG.18d82140d65ddc5898968c@news.earthlink.net`

## Debunking the mythology



Naturally, over all the years, a few myths have grown up around  $\LaTeX$ , often propagated by people who should know better. So, just to clear up any potential misunderstandings. . .

**MYTH: ' $\LaTeX$  has only got one font'**  $\LaTeX$  systems can use any OpenType, TrueType, Adobe (PostScript) Type1 or Type3 (METAFONT) font. This is more than any other known typesetting system.  $\LaTeX$ 's default font is Computer Modern (based on Monotype Series 8: see the table on p. 117), not Times Roman, and some people get very upset because Computer Modern looks different to Times (I'm not making this up: it's just a typeface, guys, get over it).

**MYTH: ' $\LaTeX$  isn't WYSIWYG'** Simply not true.  $\TeX$ 's DVI and PDF is generally better quality WYSIWYG than any wordprocessor and most DTP systems. What people mean is that  $\LaTeX$ 's typographic display is asynchronous with the editor window. This is only true for the default CLI implementations. See About this book on p. xx for details of synchronous versions.

**MYTH: ' $\LaTeX$  is obsolete'** Quite the opposite: it's under constant development, with new features being added weekly. Check `comp.text.tex` for messages about recent uploads to CTAN. It's arguably more up-to-date than most other systems:  $\LaTeX$  had the Euro (€) before anyone else, it had Inuktitut typesetting before the Inuit got their own province in Canada, and it still produces better mathematics than anything else.

Learning to write well can be hard, but authors shouldn't have to make things even harder for themselves by using manually-driven systems which break their concentration every few seconds for some footling adjustment to the appearance, simply because the software is incapable of doing it right by itself.

Donald Knuth originally wrote  $\TeX$  to typeset mathematics for the second edition of his master-work *The Art of Computer Programming*, and it remains pretty much the only typesetting program to include fully-automated mathematical formatting done the way mathematicians do it. But he also brought out a booklet called *Mathematical Writing* which shows how important it is to think about what you write, and how the computer should be able to help, not hinder, the author while writing.

$\TeX$  is of course much more than math: it's a programmable typesetting system which can be used for almost any formatting task, and the  $\LaTeX$  document preparation system which is built on it has made it usable by almost anyone. Professor Knuth generously placed the entire  $\TeX$  system in the public domain, which meant it is free for anyone to use, but which also meant that for many years there was little commercial publicity which

would have got  $\text{T}_{\text{E}}\text{X}$  noticed outside the technical field, because there were few commercial versions.

Nowadays, however, there are many companies selling  $\text{T}_{\text{E}}\text{X}$  software or services,<sup>3</sup> dozens of publishers accepting  $\text{\LaTeX}$  documents for publication, and hundreds of thousands of users using  $\text{\LaTeX}$  for millions of documents.<sup>4</sup>

There is occasionally some confusion among newcomers between the two programs,  $\text{T}_{\text{E}}\text{X}$  and  $\text{\LaTeX}$ , and the other versions available, so I'd like to clear this up:

**$\text{T}_{\text{E}}\text{X}$**  A typesetting program, originally written by Don Knuth at Stanford in 1978–9. It implements a macro-driven typesetters' programming language of some 300 basic operations and it has formed the core of many other desktop publishing (DTP) systems. Although it is still possible to write in the raw  $\text{T}_{\text{E}}\text{X}$  language, you need to study it in depth, and you need to be able to write macros (subprograms) to perform even the simplest of repetitive tasks.

**$\text{\LaTeX}$**  A user interface for  $\text{T}_{\text{E}}\text{X}$ , designed by Leslie Lamport at Digital Equipment Corporation (DEC) in 1985 to automate all the common tasks of document preparation. It provides a simple way for authors and typesetters to use the power of  $\text{T}_{\text{E}}\text{X}$  without having to learn the underlying language.  $\text{\LaTeX}$  is the recommended system for all users except professional typographic programmers and computer scientists who want to study the internals of  $\text{T}_{\text{E}}\text{X}$ .

**Con $\text{T}_{\text{E}}\text{X}$ t** (not 'Contest') A system similar to  $\text{\LaTeX}$ , but with its own set of commands, and a much greater emphasis on producing high-function PDF output. Documentation is less accessible than for  $\text{\LaTeX}$ , but the author, Hans Hagen provides excellent support at Pragma/ADE<sup>5</sup>.

**$\text{X}_{\text{E}}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$**  A recent reimplementaion of  $\text{T}_{\text{E}}\text{X}$  by Jonathan Kew which merges Unicode and modern font technologies. It is in common use in graphical environments such as  $\text{T}_{\text{E}}\text{X}$ shop (Mac OS X), *Kile* (GNU/Linux), and *WinEDT* (Windows). Details are at the SIL web site<sup>6</sup>.

**pdf $\text{T}_{\text{E}}\text{X}$  and pdf $\text{\LaTeX}$**  An extended version of the  $\text{T}_{\text{E}}\text{X}$  program that creates PDF directly instead of via DVI files and PostScript, written by Hàn Thê Thành. It can also enhance the result of  $\text{T}_{\text{E}}\text{X}/\text{\LaTeX}$  typesetting with the help micro-typographic extensions, native font embedding, and

<sup>3</sup> See, for example, the list of  $\text{T}_{\text{E}}\text{X}$  vendors in Table 1, and the list of consultants published by TUG.

<sup>4</sup> A guesstimate. With free software it's virtually impossible to tell how many people are using it.

<sup>5</sup> <http://www.pragma-ade.nl/>

<sup>6</sup> [http://scripts.sil.org/cms/scripts/page.php?site\\_id=nrsi&id=xetex](http://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&id=xetex)

## More mythology



**MYTH: 'L<sup>A</sup>T<sub>E</sub>X is a Unix system'** People are also heard saying it's 'a Windows system', 'a Mac system', etc., etc. ad nauseam. T<sub>E</sub>X systems run on almost every computer in use, from the biggest supercomputers right down to handhelds (PDAs like the Sharp Zaurus, the Nokia N800, and most Apple and Android devices). That includes Windows and Linux PCs, Macs, and all other mini, mainframe, and Unix systems. If you're using something T<sub>E</sub>X doesn't run on, it must be either incredibly new, incredibly old, or unbelievably obscure.

**MYTH: 'L<sup>A</sup>T<sub>E</sub>X is "too difficult"'** This has been heard from physicists who can split atoms; from mathematicians who can explain why  $\pi$  exists; from business people who can read a balance sheet; from historians who can grasp Byzantine politics; from librarians who can understand LoC and MARC; and from linguists who can decode Linear 'B'. It's complete nonsense: most people can grasp L<sup>A</sup>T<sub>E</sub>X in 20 minutes or so — it's not rocket science (or if it is, I know any number of unemployed rocket scientists who will teach it to you).

**MYTH: 'L<sup>A</sup>T<sub>E</sub>X is 'only for scientists and mathematicians''** Untrue. Although it grew up in the mathematical and computer science fields, two of its biggest growth areas are in the humanities and business, especially since the rise of XML brought new demands for automated web-based typesetting.

PDF support for hyperlinking. It can also produce DVI files, so it is currently (2011) the default T<sub>E</sub>X engine in most distributions.

**T<sub>E</sub>Xinfo** Texinfo is the official documentation format of the GNU project.<sup>7</sup> It was invented by Richard Stallman and Bob Chassell. It uses a single source file to produce output in a number of formats, both online and printed (DVI, HTML, INFO, PDF, XML, etc.). TeXinfo documents can be processed with any T<sub>E</sub>X engine.

Both T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X have been constantly updated since their inception. Knuth has now frozen changes to the T<sub>E</sub>X engine so that users and developers can have a virtually bug-free, rock-stable platform to work with.<sup>8</sup> Typographic programming development continues with the New Typesetting System (NTS), planned as a successor to T<sub>E</sub>X. The L<sup>A</sup>T<sub>E</sub>X3 project has taken over development of L<sup>A</sup>T<sub>E</sub>X, and the current version is L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>, which is what we are concentrating on here. Details of all developments can be had from the TUG web site at <http://www.tug.org>

<sup>7</sup> GNU's Not Unix (GNU) is a project to create a completely free computing system — 'free' meaning both free from encumbrances and restrictions as well as free of charge.

<sup>8</sup> Knuth still fixes bugs, although the chances of finding a bug in T<sub>E</sub>X these days approaches zero.