Preface

Many people discover LaTeX after years of struggling with wordprocessors and desktop publishing systems, and are amazed to find that TeX has been around for over 40 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 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, not just to new computer users, and it's great fun, 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 may 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 software' that the major manufacturers foist on their clients!

A few years ago a new LaTeX user expressed concern on the `comp.text.tex` newsgroup about 'learning to write in LaTeX'. Some excellent advice was posted in response to this query, which I reproduce with permission below (the bold text is my own emphasis; full link in Heller (2003) below):

> No, the harder part might be *writing*, period. TeX/LaTeX 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 2003)

More recently, an article reporting on a study of writing patterns between Microsoft Word users and LaTeX users reported that it was faster to use *Word* (Knauff and Nejasmic 2014). As a reviewer of that article, I asked the authors to make it clearer that the use of the proper templates (classes and packages) removed the need for LaTeX users to spend the time formatting that Word users do. The publication of the article upset a number of people in the TeX field, but I hope that it will spur the critical examination of how we write, and why it's better to do it in LaTeX than in other systems.

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* (Knuth 1980), and it remains pretty much the only typesetting program to include fully-automated mathematical formatting by default, done the way mathematicians do it. But he also brought out a booklet called *Mathematical Writing* (Knuth, Larrabee and Roberts 1989) 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.

But TEX is 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 TEX 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 for many years this also meant that there was little commercial publicity which would have got TEX noticed outside the technical field, because there was no great corporate marketing department to advertise its existence. Even now, some people who used it in college believe that it no longer exists!

Nowadays, however, there are several companies selling TEX software or services,[1] dozens of publishers accepting LATEX documents for publication, and hundreds of thousands of users using LATEX for millions of documents.[2]

There is occasionally some confusion among newcomers between the two programs, TEX and LATEX, and the other versions available, so the differences are explained in the list .

TEX: The underlying typesetting program, originally written by Donald Knuth at Stanford in 1978–79. 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 TEX 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.

[1] See, for example, the list of consultants published by TUG.

[2] A guesstimate. With free software it's virtually impossible to tell how many people are using it.

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

Debunking the mythology

Naturally, over all the years, a few myths have grown up around LaTeX, often propagated by people who should know better. These *canards* make it harder to explain to potential users why they should look at LaTeX, 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 list section 6.2.2.1 on page 161), 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, people, get over it).

MYTH: 'LaTeX isn't WYSIWYG': Simply not true. TeX's DeVice-Independent (DVI) and PDF output are generally better quality WYSIWYG than any wordprocessor and most DTP systems. What people mean is that the typographic display (preview) is asynchronous with the editor window. This is only true for the default CLI implementations. See the list item 'Synchronous typographic displays' on page xxxiii for details of synchronous versions.

MYTH: 'LaTeX is obsolete': Quite the opposite: it's under constant development, with new features being added or updated almost daily. Check `comp.text.tex` for messages about recent uploads to CTAN, or follow @TeXUsersGroup on Twitter. 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.

ConTEXt: (not 'Contest') A system similar to LaTeX, but with its own set of commands, and a much greater emphasis on producing high-function PDF output. The documentation is less accessible than for LaTeX, but the author, Hans Hagen, provides excellent support at Pragma/ADE.

pdfTeX and pdfLaTeX: Extended versions of the *tex* and *latex* programs that create PDF instead of DVI files, written by Hàn Thế Thành. There are also enhancements for microtypographic extensions, native font embedding, and PDF support for hyperlinking. It is currently (2022) still the default TeX engine in most distributions.

More mythology

If you come across other myths from people who should know better, please let me know — I'm collecting them here!

MYTH: 'LaTeX is a Unix system': People are also heard saying it's 'a Windows system', 'a Mac system', etc, etc *ad nauseam*. TeX systems run on almost every computer in use, from the biggest supercomputers and ancient mainframes right down to handhelds (even old PDAs like the Sharp *Zaurus* and the Nokia *N800*), and most Apple and Android smartphones). That includes Unix & GNU/Linux, including Apple Macintosh OS X, Windows, and all other desktop, mini, and mainframe systems. If you're using something TeX doesn't run on, it must be either incredibly new, incredibly old, or unbelievably obscure.

MYTH: 'LaTeX is "too difficult"': This has been heard from physicists who can split atoms; from mathematicians who can explain why π exists; from business people who can read a balance sheet; from historians who can explain Byzantine politics; from librarians who can understand Library of Congress (LOC) and MAchine-Readable Cataloging (MARC); and from linguists who can decode Linear 'B'. It's complete nonsense: most people can grasp LaTeX 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: 'LaTeX is "only for scientists and mathematicians"': Completely untrue. Although TeX grew up in the mathematical and computer science fields, because those were its author's 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.

XeTeX and XeLaTeX: A recent reimplementation of the *tex* and *latex* programs by Jonathan Kew which merges Unicode and modern font technologies. It is already in common use in editing environments such as *TeXshop* (Apple Macintosh OSX), *Kile* (Unix & GNU/Linux), and *WinEdt* (Windows). Details are at the Sourceforge web site.

XeLaTeX is used to produce the PDF edition of this book, and is the recommended system for new users as its Unicode features make the use of mnemonics for non-Latin characters largely unnecessary, and it works with any of the user's installed system fonts. Newcomers are unlikely to need the scripting facilities of LuaLaTeX but they are of course welcome to use it instead.

LuaTeX and LuaLaTeX: LuaLaTeX is a development of XeLaTeX which has a copy of the *Lua* scripting language built into it. This means you can write actual programs inside LaTeX, for example to calculate results and prepare statistics, avoiding the need for a separate set of external programs. The output is recalculated afresh each time, so if the data is rapidly changing, it can always represent the most recent values. It also implements a slightly different way of accessing TT and OT fonts.

TeXinfo: TeXinfo is the official documentation format of the GNU project.[3] 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, [GNU] INFOrmation (INFO), PDF, XML, *DocBook*, EPUBv3, etc). TeXinfo documents can be processed with any TeX engine.

Both TeX and LaTeX have been constantly updated since their inception. Knuth has now frozen changes to the underlying TeX engine so that users and developers now have a bug-free, rock-stable platform to work with.[4] Typographic programming development continues with the New Typesetting System (NTS), planned as a successor to TeX, and LaTeX3 as a long-term successor to current LaTeX. The LaTeX Project manages the ongoing development of LaTeX (see `www.latex-project.org`), and the current version is LaTeX 2_ε, which is what we are concentrating on here. Details of all developments can be had from the TUG web site at `www.tug.org`

[3] 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.

[4] Knuth still fixes bugs, although the chance of finding a bug in TeX these days approaches zero.