LaTeX for Everybody

Adrian Heathcote

(Australian MacWorld—September 2002)

If you're like me you've by now become addicted to OS X's crash-free behaviour. No more Restart messages and no more black anarchist bombs to push you to yet another reboot of the system. But what about the software that you run? A stable system is all very well, but it's in the applications that you do your work—what do you do when your apps are not quite as good as the system that they run on?

Chances are that, if you want to produce a nice essay, letter, or article, you type it and format it in Word. But Word is—how shall I put this delicately—not *optimally stable* in OS X. And if you have been grinding your teeth in frustration at its charming idiosyncrasies, you have probably started to ask: is there a better way?

Well there is. A much better way. And it is even more stable than OS X itself—so stable that it just *never* crashes. And it's free.

It is called LaTeX.

To understand what LaTeX is it is necessary to take a step back and understand the program that it is based on: TeX. TeX was developed in the late '70s and early '80s by a mathematician/computer programmer at Stanford named Donald Knuth. Knuth was tired of seeing his maths papers typeset badly by journals and set his mind to creating a better type-setting system—one that the user could control just by typing in commands at the keyboard. The resulting set of commands is called a *mark-up language*. If you have looked at HTML code then you know what a mark-up language looks like.

But the mark up that TeX uses has too many primitives and requires too much setting up, so, still in the early '80s, Leslie Lamport devised a set of short cuts that sit on top of the TeX base. This is what is known as LaTeX.

Here is an idea of how it works. Suppose you want to italisize the word 'verily' in an article. In Word you select the word and go to the menu bar and press *I* (or you press Command-I). In LaTeX you type \textit{verily}. When you run the program over the text the word is now in italics: *verily*.

In this case Word looks far simpler. But what if you want to put 'verily' into small caps or a sans serif font? Then you must go to the menu and manually change the font—if you even have the small caps or matching sans serif! In LaTeX you simply type \textsc{verily} or, for sans serif, \textsf{verily}. And when you run the processor you get the right output.

And when it comes to putting symbols into text, say a line in Greek, or even a simple mathematical equation like $x^n + y^n = z^n$, then the shortcomings of Word become painfully apparent. It is almost impossible to get such things looking right—and selecting Symbol is one of those things that has often in the past caused Word caused to crash.

LaTeX does such things far better—not surprising given that this is what it was designed to do.

In fact the right way to see LaTeX is that it is not just a replacement for Word, it is also a replacement for Quark, for it does not produce output on a par with a word processor, but rather a full type setting program—so that the result is camera-ready for sending to a print shop. In fact many mathematical books are produced in just this way—directly from the users own LaTeX file.

To get started using LaTeX you need a word editor. The full BBEdit is a good choice, though it costs money. At a pinch BBEdit Lite would do. But the other alternative is to use TeXShop—which connects directly to the LaTeX processor (in a way that I'll shortly explain).

In the word editor you type your piece, putting the declarations of your markup in the preamble, and then also marking your text. So a simple article would look like this

```
\documentclass[11pt]{article}
\title{What I Did on My Holidays}
\author{M. R. Hobbs}
\begin{document}
\maketitle

\noindent \textsc{It was} a fell, dark, night---pestilential
with \textit{vapours}---when I approached the Inn\ldots

\end{document}
```

Save this as *Something*.tex. You then need to process this document. For that you need to have a TeX system installed. Fortunately you can get that along with TeXShop for free from `darkwing.uoregon.edu/~ koch/texshop/texshop.html`. Simply download, run the installer and you're off. In TeXShop select LaTeX as the processor and your article will compile straight away (in about 1 sec.) to look like (Fig 1).

As you can see the output of this process will be a pdf (called *Something*.pdf) which will appear alongside your .tex file. That is part of the beauty of using LaTeX in OS X. The screen display of OS X is a pdf—so you are essentially looking at distilled postscript. The TeX processor is an adaptation of the standard UNIX LaTeX distribution, called tetex, and is set by default to produce pdf as output (it uses a program called pdftex, developed by Han The Thanh). So the LaTeX processor uses the same system wide structural features that the OS does. The result is very rapid processing of even complicated documents.

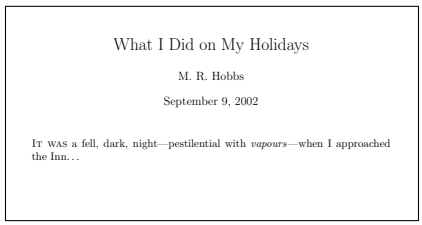


Figure 1: Your *Something*.pdf

The other part of the good news is that the LaTeX distribution that accompanies TeXShop is kept up to date and expertly maintained by its distributor Gerben Wierda. So as the OS is tweaked you can be sure that the LaTeX distribution will go on working. In fact TeXShop itself is a pure Cocoa app—and the recipient of a recent award from Apple for best ported application (an award that they probably intended to go to Wierda's accompanying distribution as well, since that is the only part that has actually been ported.)

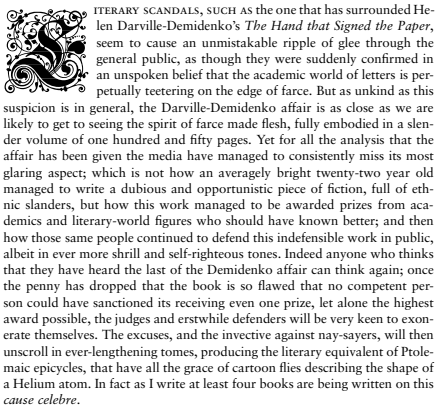
But let us get back to the How To aspect of creating documents.

It is undeniable that, compared to the average word processor, LaTeX has a steep learning curve at the beginning. This is because you need to master the mark up. But much of the pain of having to write mark up is taken away by choosing the right editor. TeXShop, for example, has a palette of LaTeX commands that makes doing much of this formatting as easy as doing it in Word. You just select, point and click. Beyond that you will need a manual for commands, options, declarations and the like. There are very good beginner's manuals, on the web in pdf format, that again are free of charge. Beyond that, the book by Kopka and Daly called *A Guide to LaTeX* (3^{rd} ed) Addison Wesley, 1999, is a comprehensive introduction.

One of the great advantages of LaTeX is that it takes care of much of the formatting for you. For example, like Word, it will produce correctly numbered footnotes, that renumber automatically if you insert another later. This gives LaTeX a decisive advantage over standard DTP programs like Quark—where producing footnotes is quite difficult manual labour.

It is also quite easy to add hyperreferences to documents, allowing the reader to jump around the text if it is viewed on screen. Likewise movies and sound can be added into documents—and text can be repurposed for the web using an add-on called *latex2html*. All things that DTP programs are still struggling toward.

LaTeX comes packaged with a very well-designed set of fonts, made by Knuth himself, that give all documents, but particularly those with mathematics in them, a nicely integrated look. But if you want to use other postscript or true type fonts then you can—after you've done a bit of converting. So if you feel that your rabble-rousing political pamphlet needs a more 17^{th} Century look and feel then you can choose your font appropriately. (See fig 2, which uses Linotype Sabon.)



THE DEMIDENKO AFFAIR

Adrian Heathcote

(Appeared in Philosopher magazine, No. 3, 1997)

LITERARY SCANDALS, SUCH AS the one that has surrounded Helen Darville-Demidenko's *The Hand that Signed the Paper*, seem to cause an unmistakable ripple of glee through the general public, as though they were suddenly confirmed in an unspoken belief that the academic world of letters is perpetually teetering on the edge of farce. But as unkind as this suspicion is in general, the Darville-Demidenko affair is as close as we are likely to get to seeing the spirit of farce made flesh, fully embodied in a slender volume of one hundred and fifty pages. Yet for all the analysis that the affair has been given the media have managed to consistently miss its most glaring aspect; which is not how an averagely bright twenty-two year old managed to write a dubious and opportunistic piece of fiction, full of ethnic slanders, but how this work managed to be awarded prizes from academics and literary-world figures who should have known better; and then how those same people continued to defend this indefensible work in public, albeit in ever more shrill and self-righteous tones. Indeed anyone who thinks that they have heard the last of the Demidenko affair can think again; once the penny has dropped that the book is so flawed that no competent person could have sanctioned its receiving even one prize, let alone the highest award possible, the judges and erstwhile defenders will be very keen to exonerate themselves. The excuses, and the invective against nay-sayers, will then unscroll in ever-lengthening tomes, producing the literary equivalent of Ptolemaic epicycles, that have all the grace of cartoon flies describing the shape of a Helium atom. In fact as I write at least four books are being written on this *cause celebre*.

Figure 2: Sabon with Haralambous Initial

There are other TEX distributions than the one that accompanies TeXShop, though they are shareware rather than free. In fact one of these is produced and maintained locally (by Adelaide University's Andrew Trevorrow). It is called OzTeX, and it is a carbon program rather than cocoa, so it will run under the classic Mac OS as well as OSX. Another, called CMacTeX, is maintained by Tom Kiffe, and produces postscript, rather than pdfs, by default.

No matter what distribution you choose you can augment it by loading in packages that increase the flexibility of your system, allowing you to produce documents that have just the features and layout that you want. There is a central repository for these packages at `www.ctan.org`. And, once again, they are all free.

But the real benefit for the user is in the stability. Once you have a system that you are in control of, that produces documents whose features turn out the way you want them to—all without crashing—well, it's hard to go back.

But why would you want to? Isn't this the OSX way?