

An Argument for Learning L^AT_EX: Benefits Beyond Typesetting

Evan J. Wessler

Abstract I discovered L^AT_EX more or less by accident, and I could not have estimated the positive impacts that would result from learning the typesetting system. Here, I argue for the benefits of L^AT_EX from perspectives apart from/stemming out of typesetting.

Introduction

As an undergraduate biology major, I had little reason and no impetus to leave the world of the WYSIWYG word processor for an advanced typesetting system. After all, the documents I was producing featured almost exclusively text, with an occasional chemical formula (e.g. CaCl₂) or simple mathematical equation (e.g. for linear regression analyses) used in laboratory reports. It was by chance that I was introduced to L^AT_EX during my sophomore year of college, when a friend who had used it to typeset a bioengineering paper happened to send me his source and output. I became immediately interested in the typesetting system, because I had recognized the appearance (i.e. the Computer Modern font and the well-formatted mathematics) as similar to that which I had seen on my calculus exams and homework sets. (Admittedly, I was always impressed with the aesthetics of these documents, and had [in retrospect, rather embarrassingly] tried to replicate their style in Microsoft Word, to no avail). I quickly learned the ins and outs of the typesetting system, and since have been a regular user, enthusiast, and unabashed proponent of L^AT_EX versus conventional word processing and presentation software.

Over the past three years, I have used L^AT_EX to produce an array of documents: analytical chemistry laboratory reports, formal business letters, physics equation sheets, charts, a symposium presentation... the list goes on. All of them were higher in quality—in terms of aesthetics—than comparable documents produced

by my peers (most of whom used Microsoft Office or Open Office) for the same purposes. However, in looking back on my experience with typesetting, I realized that there are many benefits of a *non-aesthetic* nature to learning and using L^AT_EX. In the remainder of this article, I will present and assess these advantages, and explain how learning the typesetting system has developed my skills, both in typesetting and beyond.

Problem Solving

Whether one has used it for ten years or ten minutes, all users of the L^AT_EX typesetting system are intimately familiar with L^AT_EX error messages. These notifications appear when errors in the source are encountered during typesetting. An experienced user knows they can be due to a number of things, among which are incorrectly-spelled commands, missing or extraneous brackets, failure to close environments, and other errantly typed and/or conceived text in the source. However, a new user—that is, one who is new to L^AT_EX and has no experience in dealing with code-based, debuggable source entry (e.g. in computer programming)—will be unfamiliar with the presentation and interpretation of errors, as well as with the proper action(s) that must be taken to correct them. Moreover, this process is often non-trivial, because it is not as straightforward as the new user might assume. For example, a message may indicate that there is something wrong at a certain line, whereas the actual incorrect element may be present several lines before the stated location. In addition, the very syntax of error messages may be puzzling (bewilderment with errors during creation of complex tables comes to mind). It is also the case that some commands and environments cannot be used in tandem (e.g. the `\verb` command cannot be used as-is inside `\section{}` commands). Incompatible environments and non-global commands may be unbeknownst to the author; the consequent errors are often the source of several frustrating problems that require an advanced solution.

Thus, to be able to produce a correctly typeset document using L^AT_EX, one must become proficient at troubleshooting. The process may be as simple as searching a few lines for errant symbols, or trying different commands. However, a particularly perplexing error message containing ambiguous and unhelpful language may demand more creative solutions. An especially useful approach that I discovered at some point early on is what I will call the “incremental comment-

out” strategy. This involves systematically commenting-out (i.e., marking lines with the “%” symbol, so that they are ignored by the typesetting engine) different short segments of the source in succession, and attempting to typeset each time. For an error message that reveals little to no information about the location of the error, this tactic is invaluable; one will see that the document fails to typeset every time except for when the region in which the error is contained is commented-out. In this way, the error is pinpointed and can be corrected. This method may be dismissed as inefficient by those users who produce very large documents (intermittent typesetting is always advisable) and those who are more experienced and highly-versed in the nuances of L^AT_EX warnings; however, to the novice, this is a good way to learn about error syntax, typical problems encountered, and the methods behind locating and fixing mistakes. Of course, once an error is found, solving the problem is usually a matter of referencing any decent L^AT_EX manual; but in order to get to this point, the significant work of finding and understanding the error must be performed by the user. Any exercise of this nature is bound to increase one’s capacity and ability to problem solve.

Taking Command of the Command Line

Before I started using L^AT_EX, the command-line interface was largely unknown to me; its seemingly obscure commands and cold, intimidating appearance (it is, after all, manipulated using *just* text, which is difficult for someone like myself—who started using computers when graphical user interfaces were the norm—to accept) made it esoteric at best. I had previously dabbled in the Windows “Command Prompt” program, but had no real idea of what I was doing. (Fortunately, this did not lead to any catastrophes). Upon first introduction to L^AT_EX, I edited and typeset my documents in the way that most new users probably do; I wrote the source in a user-friendly GUI front-end program (in my case, Richard Koch’s “TeXShop”) and hit the “Typeset” button. It was only when I started reading up on L^AT_EX that I discovered that editing, typesetting and previewing could be accomplished within a terminal (via the use of editors such as Emacs and Vim, and issuing commands such as `latex` and `xdvi`). This discovery led me to utilize the terminal more often, and in turn to start experimenting with its other uses.

Since then, I have become fairly proficient at the command line. But this is not just for the sake of using it as a neat or exhibitionist alternative (I have

become convinced that anyone who is unfamiliar with the terminal that sees me using it thinks I am either up to no good, or that I am performing operations too sophisticated to be relevant to the “normal” computer user); I regularly use it to perform necessary functions (e.g. secure shell, efficient exploration, creation, copying and moving of directories and their contents, elementary programming, etc.). In this way, L^AT_EX served as a sort of “gateway” into learning to use my computer to its maximum potential. I would argue that this kind of exploration would serve as a similar boon to other users, and that it is always positive when one learns more about the workings of the technology they depend on and use frequently.

An Appreciation for Formatting

As a burgeoning scientist, I have read countless papers and borne witness to a host of presentations that suffered from a major deficit: lack of logical formatting. It has become evident to me that this often has a significant, negative impact on the content of a scientific message. Blatantly incorrectly constructed outlines, disordered talking points, and poorly formatted section labels often turn what would be great papers and presentations into travesties of communication. Anyone who has tried to understand science knows that if ideas and data are not presented in an organized, logical fashion, they can be lost in a swirl of seemingly incomprehensible babble. The same can be said for material in other fields; it is a universal fact that ignoring logical structure can be disastrous.

That said, it is quite obvious that formatting is given little credence by most people who produce documents and give presentations. Much of the problem—as cited in so many pro-L^AT_EX pieces of literature—is that authors often make bad choices when detailing the aesthetic layout of their media. The consensus solution is to remove this task from the author’s responsibility; this is successfully achieved by L^AT_EX.

Of course, an inanimate typesetting system cannot absolve one from focusing on *how* to organizing their content. But at least for me, something interesting happened upon learning L^AT_EX and using it for a period of time: I began not only to realize the cruciality of logical formatting, but also to *think carefully* about it. In other words, when I use L^AT_EX, I know I don’t have to worry about the boldness or the size of my section headers; in turn, I am empowered to dedicate more

of my focus toward what I want to say, and where I want to say it. The side effects of this have transcended my use of the typesetting system. For example, when I take notes in my laboratory notebook, I notice a greater attention toward organized and systematized record-keeping; when I create a presentation, I find myself conscious of my outline, how each slide fits into it, and the efficiency with which I move between points. Thus, \LaTeX has helped me gain an appreciation for logical formatting that has extended into activities which are essential for the purveyance of information.

Synthesis

The widespread use of \LaTeX has obvious explicit utilitarian impact; it has made possible the creation of well-formatted documents, whether they have significant mathematical content or otherwise. Numerous proponents of the typesetting system seem to enjoy focusing their efforts on berating conventional productivity software programs for their inefficiency, and for the aesthetic inferiority of the documents they produce. But it is not often the case that the *learning* of \LaTeX is the topic of discussion. (There are a few pieces that deal with the learning curve of \LaTeX , but not the process and consequences themselves). Here, I have attempted to make this my focus. I have proposed that there is significant weight to the argument that learning \LaTeX not only allows you to produce great-looking documents, but also confers benefits that may not directly relate to typesetting, such as extension of problem solving skills, learning more about the technology in use, and culturing of logical planning skills.

There exists of course the potential for additional benefits to arise. Indeed, “learning” \LaTeX is not a one-shot deal; rather, it is a continuous process, in which the user constantly develops his/her skills in typesetting, and as a result finds new and better ways to produce high-quality, well-formatted documents. As a high-level typesetting system, \LaTeX demands curiosity, encourages tinkering, and promotes careful thinking, leading to positive developments in typesetting and beyond.