

Maier, Gunther

Article

LaTeX: Know what you are missing

REGION

Provided in Cooperation with:

European Regional Science Association (ERSA)

Suggested Citation: Maier, Gunther (2016) : LaTeX: Know what you are missing, REGION, ISSN 2409-5370, European Regional Science Association (ERSA), Louvain-la-Neuve, Vol. 3, Iss. 2, pp. R1-R9,
<https://doi.org/10.18335/region.v3i2.176>

This Version is available at:

<https://hdl.handle.net/10419/155695>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<http://creativecommons.org/licenses/by-nc/4.0>

L^AT_EX – Know what you are missing

Gunther Maier¹

¹ WU, Vienna, Austria (email: gunther.maier@wu.ac.at)

Received: 11 November 2016/Accepted: 15 November 2016

Abstract. This article gives a brief introduction to L^AT_EX and related tools. The aim is to give an overview, to demonstrate the flexibility and versatility of the software, and to assist the reader taking first steps using it. The article links to a number of valuable resources for further information.

1 Introduction

There are not many cases of software developed in the 1970s that are still in use today. Donald Knuth’s program T_EX is such a case. He developed it in the late 1970 in reaction to the bad quality typesetting of one of his manuscripts. Nowadays, almost forty years later, the program is still used intensively, mostly via the standard macro package L^AT_EX that builds on T_EX. This package, which was developed in the 1980s by Leslie Lamport, makes the power of T_EX much more accessible to occasional users, and opened the door for a much wider use of the typesetting program. Today, although everything builds on Donald Knuth’s T_EX, L^AT_EX is the term most widely used. L^AT_EX runs on all major computer platforms, is supported by major publishers, and forms the de-facto standard in all formal scientific disciplines. But, L^AT_EX offers more than just nicely formatted equations; actually, *a lot* more. It is the intention of this brief introduction to give you a glimpse of that.

True, L^AT_EX is not the type of program that the average user just starts up and intuitively knows how to use. On the other hand it is also far from rocket science. The support that some of the modern tools like TeXstudio and web-based versions provide reduces the learning requirement for beginners substantially. So, if you have decided some time ago that L^AT_EX is too difficult for you, now may be the time to reconsider.

In the next section we will briefly discuss what L^AT_EX is and what it is not. Then, we will turn to the flexibility of L^AT_EX and will demonstrate some of the strengths of the software system.

2 What L^AT_EX is and what L^AT_EX is not

As has been mentioned in the introduction, L^AT_EX is a typesetting system. Technically, it takes an input file, processes it, and produces some output file. There are different versions and flavors of L^AT_EX available in an installation that can generate different output formats. pdf_lat_ex, for example, writes a PDF-file. Standard L^AT_EX produces a DVI¹-file, which can be converted into various output formats in a second step. It is

¹DVI stands for “device independent”.

the task of the user to generate the necessary input file, feed the correct compiler and then handle the output file appropriately. Viewed from today's menu and window based user interfaces, this is rather archaic procedure. Recent add-on programs like TeXstudio (<http://texstudio.sourceforge.net/>), however, integrate it into a modern computing environment and support the user with all kinds of nice features.

The typesetting quality of L^AT_EX shows up in many seemingly minor features. Take the previous paragraph, for example, and look closely at the words “file” and “flavors”. You will see that “f” and “i” and “f” and “l” seem to form just one character. Such ligatures have been developed in the typesetting craft over the centuries and are by default respected by L^AT_EX. Similarly, you can see that spacing between words, which is proportionally entered in order to produce blocks of text, is slightly smaller than the space that is inserted after the period at the end of a sentence. These features make it easier to read the text and contribute to the typesetting quality of documents produced by L^AT_EX.

Nevertheless, and despite of the fact that it produces much nicer output, L^AT_EX is not a word processor. WYSIWYG – “What You See Is What You Get” – is not on the agenda of L^AT_EX. Your input file will never show you exactly how the output will eventually look. Because of that, if you occasionally want to type a letter or some short texts, it is probably not worth installing L^AT_EX in the first place. Use your pre-installed WordPad or download one of the many free word processors and do not bother to learn about packages, commands and environments. In all other cases, L^AT_EX may be your best friend. In particular, these are documents

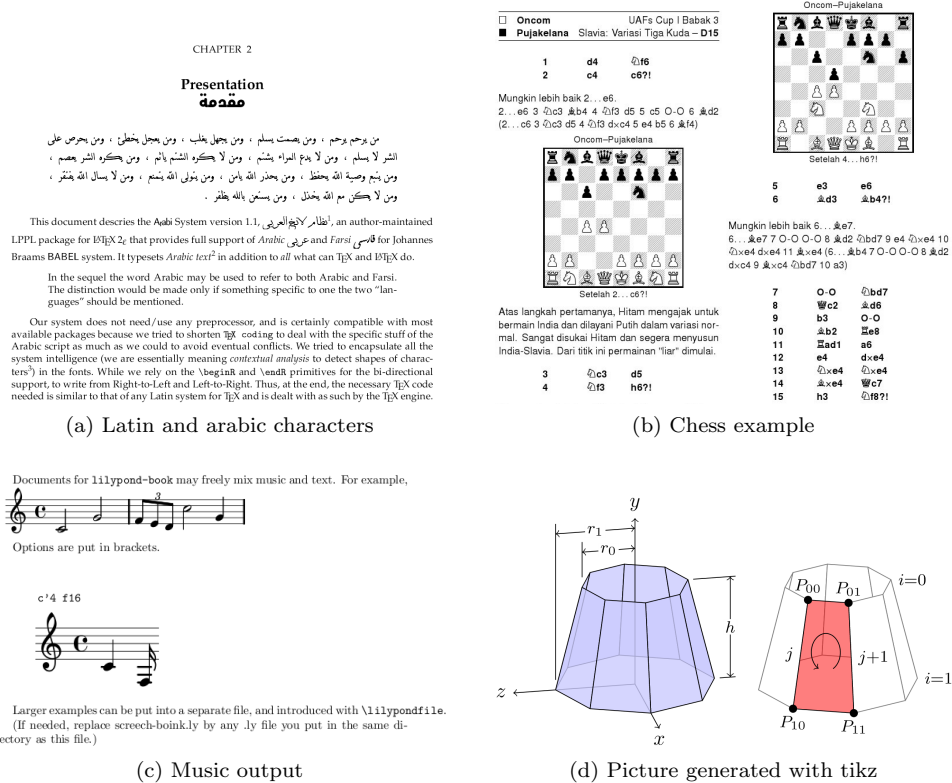
- with more than a hand full of equations;
- with more than 100 pages;
- with substantial cross-referencing (to sections, tables, figures, equations, etc.);
- with an extensive bibliography;
- with unusual content like text in foreign characters, written music, etc.;
- with close connections to statistical analyses.

In all these cases it may be worth spending a little bit of time exploring L^AT_EX. The return of this investment will probably be quite large. And, your investment will only be in terms of time and effort, because L^AT_EX is open source software and therefore freely downloadable. If you are using a Mac or a Unix-based machine, chances are that you don't even have to install the software. It may already be sitting there waiting for you to use it.

While T_EX is a very flexible system for typesetting, L^AT_EX implements many of the rules for high quality typesetting. To some extent, L^AT_EX does the formatting for you. For example, it takes care of the font used for section headings and of the space before and after the heading. It numbers them correctly and also makes sure a section heading does not show up as the last item on a page. Generally, the formatting standards of L^AT_EX make good sense. Nevertheless, most of them can be changed by redefining parameters or via the use of external packages. But, if you want to exert full control over all the formatting of your document, L^AT_EX might not be the right tool. In that case you probably should turn directly to T_EX. But, be warned: this involves substantial time and effort.

One important feature of L^AT_EX that is not immediately visible is that it is largely using plain text files. In particular, the input for your document is just plain text. Everything is in plain view and nothing is hidden or stored in a proprietary file format. This implies a high level of data security because even after years you will still be able to access your old documents. Outdated file formats are simply not an issue with L^AT_EX. Moreover, the text-based format also allows you to read the input file with a program and reformat it into a new version very easily.

This feature is important for some tools that can be viewed as extensions of L^AT_EX. For example, “knitr” (see [Xie 2015](#)) provides a link between R and L^AT_EX which allows for automatic inclusion of R-output into L^AT_EX code and thus for the generation of dynamic documents of high typesetting quality.

Figure 1: Output produced solely from \LaTeX by use of packages.

3 The versatility of \LaTeX

In this short document we cannot give an introduction to the use of \TeX or \LaTeX . Classic references are Knuth (1984) and Lammport (1994). They cover \TeX and plain \LaTeX . Mittelbach et al. (2005), en.wikibooks.org (2016), and many others also describe some of the more popular \LaTeX packages. The best starting point on the Internet is CTAN, the Comprehensive \TeX Archive Network, at <http://www.ctan.org>. It offers a wealth of information and acts as the official repository of packages and styles. They adjust and adapt plain \LaTeX in many different ways thus adding to the versatility of the system.

In the next section we will make suggestions for your first steps toward using \LaTeX and guide you through some very basic steps. Before we do this, we want to highlight the versatility of \LaTeX and show some of the unexpected things that can be done with the software – without explaining how to do that.

Figure 1 shows examples of unusual output produced by \LaTeX and publicly available packages. They demonstrate the use of different character sets in one document, the creation of complex figures via package “tikz”, as well as documents with music and chess output. None of these examples uses any other software but \LaTeX and its packages and therefore demonstrates the versatility of this software. Of course, \LaTeX can also embed graphics files that were produced by external software. With the “media9” package you can even embed videos and other interactive media content (requiring a Flash component for playback).

4 \LaTeX for the absolute beginner

Now that you are hopefully convinced that \LaTeX is worth a closer look, let us turn to some of the specifics. Again, the aim is not a complete description of all the commands and options, but to help you with the first steps. We want to lay the foundation for your own exploration of the world of \LaTeX . We do this assuming that you are an absolute

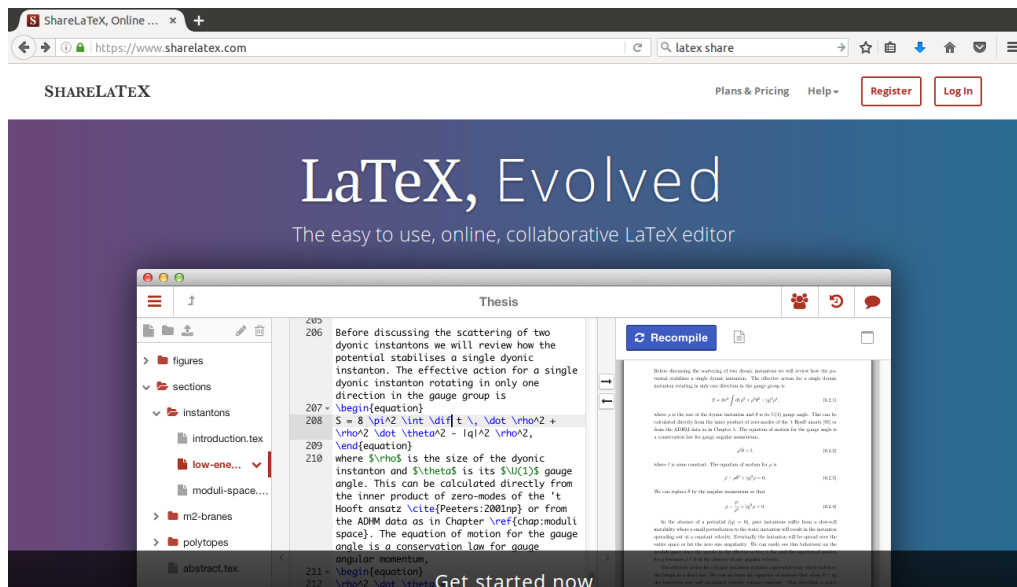


Figure 2: ShareLaTeX starting page

beginner and have not been exposed to \LaTeX before.

In subsection 4.1 we will tell you how to get access to the software. Subsection 4.2 will then describe the language basics and guide you through the creation of your first \LaTeX document.

4.1 Getting the software

If you are an absolute beginner and just want to explore \LaTeX , your best option is probably *not* to install the software on your local computer but to use \LaTeX interactively over the Internet. There are different sites available. We will use “ShareLaTeX” (www.sharelatex.com).

When you type in the URL or click the link above, you will see Figure 2. Register or log in and you are ready to run. One of the obvious advantages of using a web based version of \LaTeX is collaboration. Provided they are registered with ShareLaTeX, you can invite your coauthors to your document so that you always work on the same version.

The traditional form of using \LaTeX , however, is from an installation on your private computer. CTAN (<http://www.ctan.org/starter>) offers detailed instructions for all major operating systems. Unless it is absolutely necessary and you have expert help available, stay away from manual installation and use one of the distributions (like “ \miktex ” for Windows). In addition to the basic programs these distributions typically also install many commonly used packages and management software.

In addition to \TeX , \LaTeX , and friends in a local installation, you will also need an editor to write your documents. I use TeXstudio, which provides a lot of support both for generating documents and for compiling them.

4.2 Language basics

When you have logged in to ShareLaTeX, you will see the – initially empty – projects page (Figure 3). Click “New Project” and – for our example – then “Blank Project”. Name your project and you will get the bare minimum of a working \LaTeX document in your editing environment (Figure 4). The environment has three columns. The left one lists the involved files, the middle one the \LaTeX code, and the right one the compiled output. The middle column is a text editor where you can enter text and \LaTeX code. Whenever you want to update the output, click the blue “Recompile” button. If there is an error in your code, the respective information can be found in the log file.

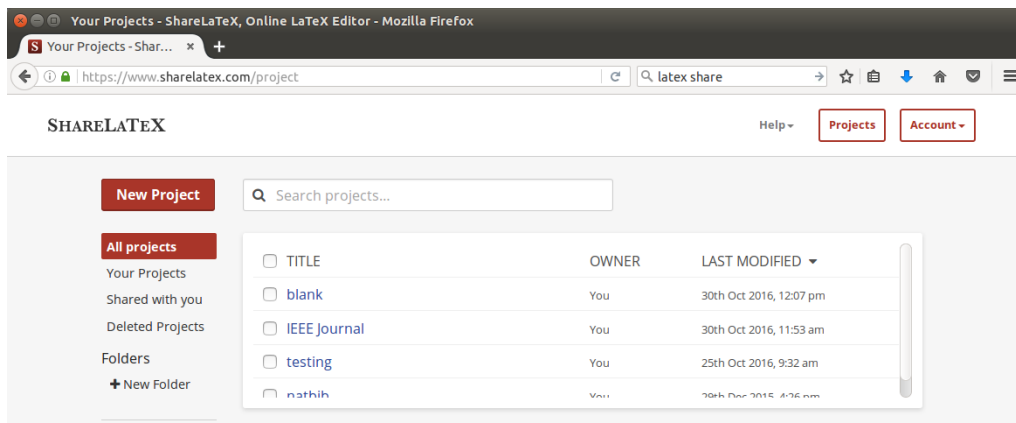


Figure 3: ShareLaTeX projects page

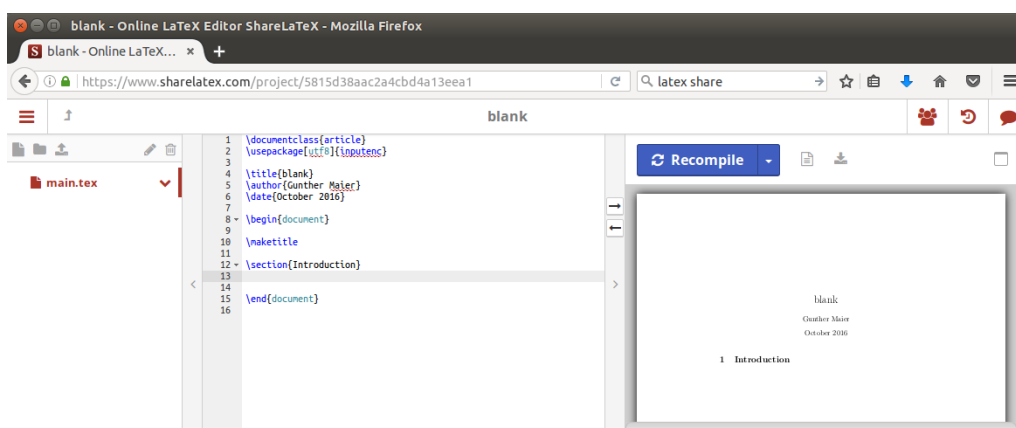


Figure 4: ShareLaTeX editing environment

Let us look more closely at the \LaTeX code in the middle column of Figure 4. All statements beginning with “\” are \LaTeX commands or environments. Environments have a `\begin{}` and an `\end{}` statement like `document` in our example. Commands – like `\section{}` or `\maketitle` – consist of only the “\”, the command name, and maybe some argument. Every \LaTeX document starts with the command `\documentclass{}`. The parameter defines the type of document that will be generated. In our example, we instruct \LaTeX to produce an article.

The command `\usepackage{}` includes external packages. In our example the package “inputenc” is included. This package allows one to use different character sets for input. Which character encoding to use is defined in an additional parameter in square brackets. For details see the documentation of “inputenc” on CTAN.

The next three commands, `\title{}`, `\author{}`, and `\date{}` are used to internally store the respective information. These commands do not produce any output. The later used command `\maketitle` takes this information and typesets the title of the article.

The most fundamental environment in \LaTeX is the `document` environment. Only text within this environment can generate output. The part before the `\begin{document}` is called preamble, statements after `\end{document}` are ignored by the compiler.

The final command in the code is `\section{}`. This command produces a section heading and numbers it accordingly. The command also enters information to an auxiliary file that can be used for generating a table of content. Depending on the type of document, additional sectioning commands are available. Books, for example, can have parts (`\part{}`) and chapters (`\chapter{}`). Sections can be structured further with `\subsection{}` and `\subsubsection{}`. After every such statement you can insert a `\label{}` statement with a unique label. They can be the target of cross referencing by use of `\ref{}` statements. \LaTeX replaces those statements with the respective numbers.

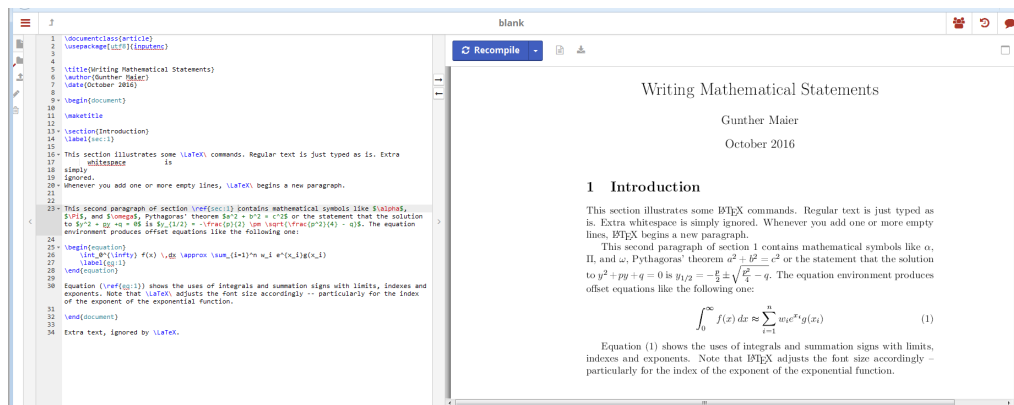


Figure 5: ShareLaTeX some mathematical commands

The most important element for structuring text is not a \LaTeX command at all, i.e. a statement that begins with a “\”, but an empty line. Empty lines – one or more – mark the end of a paragraph. Line breaks within a block of text are simply treated as blanks by \LaTeX .

There are many, many more commands and environments even in plain \LaTeX . Packages typically add additional commands and environments or redefine existing ones. For details see [Lamport \(1994\)](#). Important environments are

figure defines a floating element to display a figure. Normally, this element is not typeset at that location in the output where it is defined. By default, floating elements like ‘figure’ and ‘table’ are positioned at the top of output pages and shifted to the “best” location. As usual, this behavior can be influenced by optional parameters.

table defines a floating element to display a table. It behaves just like the **figure** environment.

itemize defines a list of bullet points. The items of the list are marked with `\item`.

enumerate defines a numbered list. Again, the items of the list are marked with `\item`.

equation defines an offset mathematical equation. Key aspects of typesetting mathematical equations are discussed in the next subsection.

4.3 Math mode in \LaTeX

\LaTeX can operate in two different modes: text mode and math mode. The two modes accept different commands and also produce different output. \LaTeX starts operation in text mode and all the statements shows in Figure 4 are in text mode.

Various commands are used to switch from one mode to the other. One is the **equation** environment. With `\begin{equation}` \LaTeX switches to math mode, with `\end{equation}` it switches back to text mode. When mathematics is needed within a block of text, the `$` character switches between the two modes. Sometimes one needs to insert text into a mathematical statement. This is done with the `\mbox{}` command.

Distinguishing between text mode and math mode is important for two reasons: first, most \LaTeX statements only work in one mode or the other; second, the two modes produce very different output even with the same input. In math mode \LaTeX typesets all input as a mathematical statement. Characters are treated as mathematical variables, a series of characters as a product of variables. For example, the text “math mode” is formatted as “*mathmode*” in math mode. Note the slanted characters and that the blank between the two words disappears. This also applies to expressions like “log”, “exp”, “sin”, etc. To format them correctly, \LaTeX offers commands like `\log`, `\exp`, and `\sin`.

\LaTeX offers a large number of mathematical symbols, operators, commands, etc. This set is extended further by the American Mathematical Society via some packages. These

statements are described in the standard literature on L^AT_EX and summarized by many pages on the Internet².

The ShareLaTeX screenshot in figure 5 illustrates some of the more common statements. This example extends the one in figure 4. Besides math mode commands, which will be described below, the example also illustrates (1) that extra whitespace does not impact the formatting, (2) that a blank line separates paragraphs, and (3) cross referencing by use of `\label{}` and `\ref{}`. Of course, you are completely free in choosing the label strings and do not have to indicate the type of label as we did in our example. This is just a convention that we apply. It makes it easier to keep track of all the labels in larger documents.

The example shows both, separate mathematical expressions (equation environment) and mathematics embedded in the text (enclosed by dollar-signs). There is no difference in the command set of the two. However, when mathematics is embedded in the text, L^AT_EX tries to format the mathematical statements in such a way that the line spacing is disturbed as little as possible. Because of that, the two options may produce slightly different output.

The use of Greek letters is quite self explanatory. Besides these, there are many other symbols that can be used in math mode. The other mathematical statements in the text illustrate subscripts (`_`) and superscripts (`^`). These characters shift just the next character. If more characters need to be shifted, they have to be enclosed by curly braces. For example, `y_{t-1}` produces the variable y at time period $t - 1$ (y_{t-1}). Without the curly braces you subtract one from the value of y at time t ($y_t - 1$).

The math in the text also demonstrates the commands `\frac` for fractions and `\sqrt` for square roots. The command `\pm` produces the plus-minus operator (\pm).

Additional math commands are used in equation (1) in the example. This equation is generated by use of the equation environment. In the style we used L^AT_EX centers the equation and places the numbering on the right margin. The sample demonstrates the use of `\int` for generating integrals and of `\sum` for summation signs. Both commands allow you to optionally add margins via subscript and superscript characters. Note that `\infty` generates the infinity character (∞). The integral on the left hand side of the equation shows an additional feature. The command `\,` adds a small amount of extra space between $f(x)$ and dx so that the differential is visually separated from the function. Usually, such explicit formatting should be avoided. In this case, however, it is necessary to add this command to ensure proper formatting. `\,` is just one of a number of commands for adding extra space in math mode.

The command `\approx` (\approx) demonstrates that L^AT_EX knows more relational operators than $=$. Other examples are `\neq` (\neq), `\geq` (\geq), and `\leq` (\leq). The right hand side of the equation also demonstrates the effect of subscripts in superscripts. Note that the index i in the exponential function is smaller than that of the w or that of the x in the function $g()$. L^AT_EX makes these adjustments automatically and can go much further than just two layers.

5 Features of L^AT_EX we did not cover

In this short paper we can only describe a few of the services that L^AT_EX can provide. There are many more important tools and some of them we at least want to mention. In section 1 we listed features of documents that may best be handled by L^AT_EX. Among those were large documents and documents with extensive bibliographies. We will touch the arguments in this context briefly below in this section.

Most word processors load the whole document into memory and do all the text manipulations there. This implies the risk of losing your work in case of a program bug or power loss. But, even when the software is bug free, the necessary memory management will slow down operations considerably once your document exceeds a certain size. L^AT_EX on the other hand only reads in the document source when compiling and reads it in chunks. Because of that compile runs in L^AT_EX can never jeopardize the source document

²See, for example, http://www.colorado.edu/physics/phys4610/phys4610_sp15/PHYS4610_sp15/Home_files/LaTeXSymbols.pdf and <ftp://ftp.ams.org/pub/tex/doc/amsmath/short-math-guide.pdf>

and can read documents of very large size. One of the disadvantages of this strategy is that usually one compile run is not enough to get all the citations and cross-referencing right. Many \LaTeX users therefore employ the habit of recompiling before they look at the result.

\LaTeX includes commands for – selectively – inserting files into a master document. This allows you to split, for example, a book manuscript into chapter files and still compile the whole, consistently formatted book in one run. It is also worth mentioning that \LaTeX can produce a *table of contents*, a *list of figures*, a *list of tables*, and an *index* automatically. Custom lists can be produced with a little bit of extra code.

A special and potentially very demanding list in a publication is the *list of references*. \LaTeX is installed with various tools for the integration of citations and lists of references into a document. Probably the most popular one is BibTeX. It allows one to build a repository³ of citations. \LaTeX draws from this repository for generating both, the citations in the text and the entries in the list of references; and both formatted accordingly. This formatting can be adjusted very flexibly by use of packages and style files. REGION uses this mechanism for managing citations and references in all of the papers it publishes.

BibTeX can save a lot of work retyping and reformatting bibliographical entries. In addition, many literature databases offer BibTeX as one of the formatting options in data export. Examples are ISI Web of Science, Google Books, and the ERSA journal archive.

6 Concluding remarks

This short article wanted to provide an introduction to \LaTeX . Rather than trying to be complete, the goal was on the one hand to demonstrate the enormous flexibility of the system and on the other hand to help the reader taking first steps in using \LaTeX . The availability of on-line installations like ShareLaTeX make such first steps much easier. They save you the sometimes tedious installation of a \LaTeX distribution. Try \LaTeX out on-line, enjoy writing your first texts, articles, slides, and then decide whether or not it is worth for you digging deeper into the topic. There are plenty of books, videos, and on-line tutorials that can help with all the tasks involved. Some of these resources are listed in the references and in the appendix.

Of course, \LaTeX is not the one and only super tool for everyone and every task. But, it is a great tool for many tasks that researchers frequently do. With this introductory article you can just click on a few links, try it out and see for yourself. So that at least you know what you are missing.

References

- en.wikibooks.org (2016) *LaTeX*. <http://en.wikibooks.org/wiki/LaTeX>
- Knuth D (1984) *The TeXbook*. Addison-Wesley Professional. <http://www.ctex.org/documents/shredder/src/texbook.pdf>
- Lamport L (1994) *LaTeX: A document preparation system*. Addison-Wesley. <http://users.softlab.ntua.gr/~sivann/books/LaTeX - User's Guide and Reference Manual-lamport94.pdf>
- Mittelbach F, Goossens M, Braams J, Carlisle D, Rowley C (2005) *The \LaTeX Companion: Tools and Techniques for Computer Typesetting* (2nd ed.). Addison-Wesley
- Xie Y (2015) *Dynamic Documents with R and knitr* (2nd ed.). CRC Press, Boca Raton, FL. [http://bacbuc.hd.free.fr/WebDAV/data/Bouquins/Dynamic Documents with R and knitr \(2nd ed\)- Yihui Xie \(2015\).pdf](http://bacbuc.hd.free.fr/WebDAV/data/Bouquins/Dynamic Documents with R and knitr (2nd ed)- Yihui Xie (2015).pdf)

³Some publications call this a “database”. This term is somewhat misleading because it is again just a plain text file that stores the information.

A Appendix: Some electronic resources

The ultimate source for $\text{T}_{\text{E}}\text{X}$ and $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ is CTAN (<https://www.ctan.org>). It provides distributions, packages, documentations, tutorials, links to user groups and much more. The content of CTAN is mirrored to about one hundred computers all over the world.

A.1 Introductory texts and tutorials

Introductory texts

- Introduction to $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ (<http://www.latex-project.org/about/>)
- A short introduction to $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ (<http://ricardo.ecn.wfu.edu/~cottrell/ecn297/latex-tut.pdf>)
- The not so short introduction to $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ (<https://tobi.oetiker.ch/lshort/lshort.pdf>)
- An Introduction to Using $\text{T}_{\text{E}}\text{X}$ in the Harvard Mathematics Department (<http://www.math.harvard.edu/texman/>)

Video tutorials

- An introduction to $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ (<https://www.youtube.com/watch?v=YMhcc9VQ6Uk>)
- A series of introductory videos (<https://www.youtube.com/watch?v=FXujG7c9p8g-&list=PLNnwglGGYoTtW7o4PHFOSWGevcdFa3v3D>)
- Bibliography in $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ (<https://www.youtube.com/watch?v=lwy6pn5vi5A>)

A.2 ShareLaTeX videos

- A series of YouTube videos (<https://www.youtube.com/user/ShareLaTeX>)
- Comparing Authorea, ShareLaTeX and Overleaf for academic writing (<https://www.youtube.com/watch?v=OcbGDSE9kaM>)
- How to use ShareLaTeX (<https://www.youtube.com/watch?v=iWlc5ufcsdk>)
- Templates in ShareLaTeX (<https://www.youtube.com/watch?v=fSgi-Mwvs6g>)

A.3 Installation and setup videos

Windows

- Windows 10 (<https://www.youtube.com/watch?v=G4uIAOPq1UE>)
- Various versions (<https://www.youtube.com/watch?v=CX-I0op0z4k>)

MacOS

- Any version (<https://www.youtube.com/watch?v=5CNmIaRxS20>)

Linux

- Ubuntu (<https://www.youtube.com/watch?v=46LHjKucDbg>)
- Linux Mint (<https://www.youtube.com/watch?v=z6HPyqvomrI>)