TEX and LATEX

Filip Marić

Faculty of Mathematics, Belgrade

Abstract: This paper will present the document preparation systems TEX and LATEX. Documents created by these systems have very high aesthetic quality. Systems are mainly used for the production of technical and scientific documentation and are *de facto* standard for the communication and publication of scientific documents. Both systems are available as free software.

1 About the Document Preparation

In order to help better understanding of the role of TEX and LATEX in digital document publishing and their relationship with other similar products, in this section the document preparation process will be briefly described.

Historical perspective Publishing and printing of articles, books and similar types of documents is an art that was used and developed centuries before digital computers and digital printing were invented. Classical publishing system usually used to function like this: Authors would submit their manuscripts to publishing companies where experienced book designers or technical editors would determine the way that the text should be printed and mark it appropriately. Designers would, based on their own experience, determine and mark section and subsection titles within the manuscript, mark the examples and quotes that should be emphasized, and mark all other elements of the *logical structure* of the document. They also needed to determine and mark the number of columns that are going to be used for laying out the document, to mark the margin sizes, spacing between paragraphs and headings, to mark the font that should be used for printing and to mark all other elements of the *visual presentation* of the document. Documents marked by the designers were given to typesetters who followed the mark-up instructions and actually typeset the document, breaking the text into pages and lines during the process.

Challenges of high quality typesetting. During the document preparation process, it is necessary to take into account many fine details in order to achieve an aesthetically pleasing appearance. For example, in a correctly typeset document the spacing between the words in each line should be constant, but it usually vary from line to line in order to achieve better aligning to both margins. The spacing between sentences should be larger then the spacing between words within a sentence. To achieve a nicer look, some combinations of letters are printed by using special symbols called *ligature*. For example, there is a subtle difference between the ligature “fi” and two separate letters “fi”. Dashes and hyphens in the text should be printed in different sizes, lengths and on different heights relative to the base line, depending on their type and role. So, in the following examples different dashes and hyphens are used and that is how it should be: “X-rays”, “pages 13–15”, “yes — or no?”, and “the number –1”. There is a difference between using three dots for … and using the special ellipsis symbol ... Also, there are different types of quotes (“,”,‘,’) and rules of the language determine the correct way to use them. For example, “Please, press the ‘x’ button”.

Document preparation on computers. Modern computers let authors become the designers and editors of their own documents. Along with advances in computer technology came systems that enable authors to prepare their documents in a way which clearly determines their final visual presentation so that the text can be directly shown on the screen or printed on the printer. Marking the text and its preparation for printing is possible to do either explicitly or implicitly and this is what distinguishes the following two approaches.

WYSIWYG – *What-You-See-Is-What-You-Get*. Tools based on this approach require their users to work on the document all the time in the form that is ready for the final output (e.g., printing on a paper). Typical examples of these kinds of tools are word-processors (e.g., *Microsoft Office*, *OpenOffice.org*), HTML editors (e.g., *Microsoft Front Page*}, *Macromedia Dreamveawer*) etc. Documents are usually edited directly relying on their visual presentation, usually by using the mouse, menus and other elements of a graphical user interface (GUI). So, for example, to mark that a certain word should be printed in bold font, it is necessary to select the word (usually by using the arrows on the keyboard, or a mouse), and then issue a command for bold font (usually by using a keyboard shortcut, menu item or an icon in the toolbox). The whole process, however done, takes some time.

Markup languages – Text is explicitly marked, by entering special *annotations* or *tags* defined by *markup languages*. Most widely used markup languages today are meta-languages SGML and XML and their numerous applications, most famous of which is HTML used to mark up hyper-textual documents, and TEX and LATEX that are going to be described in the sequel. In the markup-language approach, to mark that a certain word should be printed in bold font, a special annotation of the language should be used. So, for example, in HTML bold text is annotated as `<b>bold text</b>`, while in LATEX it is annotated as `\textbf{bold text}`. Documents marked this way are then translated using specialized programs into a human-friendly form. These programs that are similar to programming language compilers are either stand-alone programs (as it is the case with TEX and LATEX, where PDF documents are generated based on the marked input files), or integrated into other programs (e.g., web browsers which is the case with HTML).

The main advantage of using explicit document markup is that it significantly simplifies document processing. Marked documents become stored information convenient for automated processing by various applications that are only interested in textual content.

Logical structure and visual presentation of a document. Extremely important principle that should be obeyed when laying out the document is clear and explicit separation between the *logical structure* and the *visual presentation* of a document. Logical structure of a document is determined by its subdivision into smaller units (e.g., parts, sections, paragraphs), and by its special elements (e.g., examples, quotes, definitions, theorems). Visual presentation of a document determines its graphical structures i.e., the look of its elements (e.g., font size and family used for printing, spacing between lines and paragraphs, margin sizes, layout of document elements on a paper or on screen, color of document elements). Modern markup languages, by their design, emphasize the importance of the separation between the document structure and its presentation and require the document authors to prepare their documents by following this principle. This is usually achieved by describing the visual presentation using separate *style-sheets*. These are usually written in a specialized sub-language of the markup language (e.g., visual presentation of hyper-textual documents is described by using the CSS language, which is a part of the HTML standard usually in a file that is physically separated from the main document). Notice that it is

possible to obey the principle of structure and presentation separation, even when WYSIWYG tools are used. However, these tools usually do not insist on this principle as much as markup languages do, and they permit unskilled users to prepare documents in a inconsistent fashion, without clearly separating the logical from the visual aspect.

The main advantage gained when the logical structure and the visual presentation are separated is the possibility to assign different visual presentations to a single document. Documents become flexible and it is possible to reuse them a number of times, each time in a different way and in a different context. For example, an article prepared using the LaTeX system can easily be adapted to a uniform look of a journal to which it is being submitted, by assigning it the style-sheet of the journal (usually available on the Internet). If the same article is to be submitted to another journal, the new style-sheet should be used, and the only thing that should be changed in the document is the part that assigns a style-sheet, without any modifications to the content of the article itself. Style-sheets are usually prepared by typographically skilled designers, so the final results are usually very pleasing. In contrast, when WYSIWYG approach is used, authors themselves usually design their own documents which often leads to unprofessionally looking documents. It can be concluded that the separation between the logical structure and the visual presentation of a document distinguishes a high-quality professional document preparation from a low-quality amateurism.

2 "The name of the game"

Let start the description of the TEX system, by paraphrasing its author *Donald Knuth* (Kunth 1986).

Words like *technology* have their origin in the Greek word *τεχνη*. This word also means *art*. The name TEX stands for the capitalized prefix *τεχ*. TEX is correctly pronounced as *tech* (as in the German word "Ach" or in the Scottish "loch"). This linguistic exercise should explain the basic usage of the system — creating high-quality technical documentation. Emphasis is, as it was with Greeks, put on the technology and on art. The lowered letter 'E' in the name suggests that TEX is the system that does the arranging of letters — typesetting.

TEX is quite old system. In late 1970s, when preparing the second edition of his famous book *The Art of Computer Programming*, Donald Knuth got back his manuscript from the publisher, and found that its typographic quality was very poor. As a computer scientist, he realized that he had the possibility to make a system that would enable authors to prepare their text themselves, achieving equally good if not even better quality then the professional publishers do. Principles and algorithms that Knuth embedded into the new document preparation system brought revolution to the scientific publishing and remain unsurpassed even today. The system TEX, as how we use it today, was developed in 1982 with smaller additions included in 1989 which have improved the support for different natural languages. The system is extremely stable; it works on a number of computer systems and is almost bug-free. Its version slowly converges to π and currently is 3.141592.

The TEX system has a role of a typesetter in the classical publishing system described earlier. Indeed, based on a precise description of the visual presentation of a document, using very sophisticated algorithms, it splits text creating different lines, pages, boxes that contain images, figures, tables and similar graphic elements and lays them out in a way which is comfortable to read and is aesthetically pleasing.

The role of the designer in the classic document preparation system is assigned to the system LaTeX – a macro system[1] offering its users (the document authors) possibility to prepare documents by specifying only their logical structure.

LaTeX, as an experienced designer relaying on standard style descriptions (usually hidden from the document authors), determines the visual presentation of the document, marks its elements and leaves the typesetting tasks to the underlying TeX system.

Of course, LaTeX is pronounced as *lay-tech*. The first version of LaTeX appeared in 1985, and the author of this system is Leslie Lamport (Lamport 1994). The version that is currently used is LaTeXe, and the version LaTeX3 is in preparation. Similarly to TeX, the LaTeX system is extremely stable and new versions of the kernel appear relatively rarely, but the system is constantly extended with additional packages used for specialized purposes.

3 A Short LaTeX Tutorial

This section contains a short tutorial on how to use LaTeX.

The material here presented can be a starting point for further investigation of this system, but for more serious usage additional documentation should be consulted. Along with reference manuals for TeX (Knuth 1986) and LaTeX (Lamport 1994), the book "LaTeX companion" (Goossens et al, 1993) is recommended for advanced usage. For novice users, the document "The not so Short Introduction to LaTeXe" (Oetiker 2008), available online, is recommended. The TeX users group (TUG) delivers the journal *TUGBoat* dedicated to the system TeX three times a year.

There is a long tradition of using TeX and LaTeX in Serbia, and in ex-Yugoslavia. There is a number of documents about these systems published in Serbian, and we recommend the book "LaTeX za autore" (Nenadić 2003).

3.1 The TeX and LaTeX Environment

TeX and LaTeX are freely available and work under all major operating systems. In the moment when this paper is published, most wide-spread distributions are the *TeXLive* (mostly used on Unix-like systems including the GNU/Linux), *teTeX* (its maintenance stopped in 2006 and from that point the usage of *TeXLive* distribution is recommended), and *MiKTeX* (used under Microsoft Windows). All these distributions contain compilers for TeX and LaTeX which translate source documents into a format that does not depend on the output device (so called *Device Independent* or DVI format), then programs that display the DVI documents on screen, programs that convert the DVI to PostScript and PDF documents, corresponding sets of fonts, and rich collections of additional packages that expand the basic possibilities of TeX and LaTeX. It is worth noticing that listed distributions do not contain source file editors. Since TeX and LaTeX are markup languages, their input documents can be created by any text editor. Still, there are some specialized editors that help creating TeX and LaTeX documents, but they have to be independently obtained and installed. Some of these are *WinEdt* and *TexnicCenter* (running under MS Windows) and *Kile* and *Emacs* TeX mode (running under GNU/Linux). In the rest of this paper, it will be assumed that LaTeX system is used (installed from any of the listed distributions) and that no specialized editor nor integrated environment is used. The process of building the final document from the source is the following:

1. The file named *document*`.tex` should be created by using any text editor available.

2. The instruction `latex` *document*`.tex`, in command interpreter of the operating system (i.e., the shell), if there were no errors in the source file, creates the file *document*`.dvi`. If there were errors, the program will make a report and the user is recommended to fix all errors and repeat the compilation process in order to ensure that the generated `dvi` file is correct.

3. The contents of the generated `dvi` file can be shown on screen by using a specialized *dvi previewer* (e.g., *xdvi*, *kdvi*, *yap*).

4. The file *document*`.dvi` can be converted to the `ps` format by the instruction `dvips` *docu-*

ment.dvi. Also, a conversion to pdf format is possible, by the instruction dvipdfm *document*.dvi. The *document*.pdf file could also be generated directly from the LaTeX source by the instruction pdflatex *document*.tex.

3.2 Elements of the LaTeX Language

A minimal document. Let start this overview of the LaTeX language by giving an example of a short, but complete document. The top cell of the following table shows the LaTeX source, while the bottom cell shows the compiled document. The same presentation style will be used in the rest of this paper.

```
1  \documentclass[a4paper]{article}
2  \title{\LaTeX{} Example}
3  \author{Filip Mari\' c}
4  \date{January 2009.}
5  \begin{document}
6    \maketitle
7    Hello, world!
8    How     are you?
9
10   Welcome to the world of \LaTeX.
11 \end{document}
```

LaTeX Example

Filip Marić

January 2009

Hello, world! How are you?
Welcome to the world of LaTeX.

The first four lines form the *preamble*, and the remaining lines form the *main body* of the document. First line suggests that the document represents a short article (the document class is article) and that the paper size will be A4. The next three lines mark the title, author and the creation date of the document. The main body is marked within the document environment. The \maketitle command produces the title of the document, based on the information on the title, author and date provided in the preamble. After this, the ordinary text that makes the document content is specified. Notice the following important LaTeX feature — whitespace and the arrangement of the text in the source file does not affect the layout of the text in the final document. Multiple spaces are treated as single, while new paragraphs should be marked by empty lines. The command \LaTeX creates the LaTeX signature. The specifics of using the Serbian language will be discussed in the sequel.

The LaTeX syntax. All *commands* of the LaTeX language begin with the symbol \ followed by the command name, usually made only of letters. The command arguments can be specified within the braces {}, and possible optional arguments can be specified within the brackets []. One such command is the \documentclass[a4paper]{article}. If braces are omitted, the first symbol following the command name is treated as its argument. Along with commands, important syntactic elements are *environments* that begin with the command \begin{...} with the environment name given as its argument, and that end with the command \end{...}, again with the environment name specified as an argument.

The importance of document class. Notice that the source file does not contain any information about the font that should be used for the title, it is not written that the title should be center and, generally, there is no information about any aspect of the visual presentation of the document. The information that the document belongs to the article class, implicitly assigns a visual presentation to the document. This presentation is defined by the article class and its authors have precisely specified all aspects of the document presentation with the class itself. Once we assigned the article class to our document, we have accepted the visual presentation defined by it. This does not mean that it is not possible to change certain visual aspects from within the document itself, but this is not recommended to novice users and it is certainly not trivial. Although this can be regarded as a

shortcoming of the LaTeX system, it is generally accepted opinion that this, in fact, is its advantage. Namely, this way authors are encouraged to think only about the logical structure of their content, and standard classes are used for defining its visual presentation. Time has shown that these classes are very good, aesthetically pleasing and they follow the long-lasting typographic tradition. Along with the `article` document class, standard classes of LaTeX are also the class book intended for writing books, the class `report` intended for writing longer reports, the class `letter` intended for writing letter, `slides` for creating slides, etc. The task of defining high-quality visual presentation for the document is left to document class authors, and these are usually skilled graphic designers who are also better experts for TeX and LaTeX then the document authors who use these systems as their auxiliary tool. The fact that authors prepare text in a form which is "clear" from any markup of visual presentation makes possible to easily alter the visual presentation after the document has been created and adapt the document to the specific requirements of a journal.

Text encoding and special symbols. Although modern versions of LaTeX allow multi-byte character encodings (e.g., UTF-8), it is customary that the text is encoded only by using the ASCII character set and to use special commands to create missing symbols. So, for example, the command `\v` inserts a caron above its argument (the letter that follows it). This can be used for printing symbols č, š, and ž (as `\v c`, `\v s`, `\v z`). Similarly, the command `\'` inserts and acute above its argument. This can be used for printing the symbol ć (as `\' c`). Also, the command `\"` inserts the umlaut and can be used for printing German symbols ö, ë, and ü. If a better support for the Serbian language is needed, it is recommended that the multi-lingual `babel` package is included by using the `\usepackage[serbian]{babel}` in the document preamble.

Many of the previously mentioned challenges of high-quality typesetting are directly supported by special commands of LaTeX. So, for example, LaTeX supports several different kinds of dashes and hyphens -, –, and — (entered as `-`, `--` and `---`). The ellipsis sign … is printed by using the `\ldots` command. Different kinds of quotations are printed by combining the ASCII signs `, `` ` ``, `'`, and `"`. Special symbols used in commands like `\`, `{`, and `}` have to be entered by using special commands `\backslash`, `\{`, and `\}`. The symbol `%` is used to mark comments in the source LaTeX documents and to print it, a special command `\%` must be used. Also, it is possible to force LaTeX to print some content `verbatim`, without interpreting commands it contains. This is achieved by using the command `\verb"..."` and the text is then printed by using a monospaced font. If a longer content should be printed verbatim, the environment verbatim should be used. All LaTeX code presented in this article is entered this way.

Marking the parts of the document and creating tables of contents. Documents are divided into parts, chapters, sections, etc. Depending on the document class used, authors have different sectioning units on their disposal. Most standard document classes offer the commands `section{}`, `\subsection{}`, and `\subsubsection{}`. The document classes book and report assume that the text is longer and also offer the commands `\part{}` and `\chapter{}`. For example:

```
1 \section{Introduction}
2 \subsection{Historical perspective}
3 Once upon a time, \ldots
```

1 Introduction

1.1 Historical perspective

Once upon a time, …

Using these sectioning instructions, it is possible to automatically generate the table of contents for the document, by issuing the `\tableofcontents` command.

Enumerations. LaTeX supports several different kinds of numerations. The environment `itemize` denotes a bulleted list. Items of the list are annotated by the `\item` command. For example,

```
\begin{itemize}
\item First item
\item Second item
\end{itemize}
```

- First item
- Second item

Similarly, the `enumerate` environment denotes a numbered list. Environments can be nested.

```
\begin{enumerate}
\item First item
  \begin{enumerate}
  \item First sub-item
  \item Second sub-item
  \end{enumerate}
\item Second item
\end{enumerate}
```

1. First item
   (a) First sub-item
   (b) Second sub-item
2. Second item

The `description` environment is usually used to create description lists.

```
\begin{description}
\item[\TeX] - a document prepara-
tion
system
\item[\LaTeX] - a more advanced
document preparation system
\end{description}
```

TEX - a document preparation system
LATEX - a more advanced document preparation system

Tabele. Scientific documents usually contain a number of tables. Tables are usually created by using the `tabular` LaTeX environment. For example

```
\begin{tabular}{lr}
hello & world \\
good  & afternoon
\end{tabular}

\begin{tabular}{|c|c|}
\hline
hello & world \\
\hline
good  & afternoon \\
\hline
\end{tabular}
```

hello	world
good	afternoon

hellow	world
good	afternoon

The first, simpler table has two columns: one aligned on the left and the other aligned on the right. This is specified by the argument `lr` given within the table definition. Cells of the table are separated by the ampersand symbol `&` and rows are separated by `\\`. The other table contains two centered columns, while vertical lines are put around both columns. This is specified by the `|c|c|` argument. The horizontal lines are explicitly marked by the `\hline` command. Tables with much more complicated structure than this can be created, but in order to do this, the reader should consult the LaTeX documentation.

Typing Mathematical Formulae. All mathematical content in LaTeX should be surrounded with dollar signs `$`. If single dollars are used, the mathematical content is print in the same line as the ordinary text. For example

```
Indexes are entered as $x_1$, and
exponents as $x^2$.
```

Indexes are entered as x_1, and exponents as x^2.

If double dollar signs are used `$$`, the mathematical content is printed centered in separate lines. For example:

```
Euler equality:
$$e^{i\pi} + 1 = 0$$
```

Euler equality:

$$e^{i\pi} + 1 = 0$$

Notice that, if more symbols should be printed as subscripts or superscripts, they should be grouped by braces `{}`. The letters of the Greek alphabet are marked by the commands based on their English language transcription. For example

```
$\alpha$, $\beta$, $\gamma$, \
ldots,
$\mu$, $\xi$, \ldots
```

α, β, γ, $\ldots$, μ, ξ, $\ldots$

The next example illustrates the markup of a bit more complex mathematical formula employing fractions (`\frac{}{}`), a root (`\sqrt{}`), and special symbols `$\pm$` and $\cdot$ (`\pm`, `\cdot`).

```
$$x_{1/2} = \frac
{-b \pm \sqrt{b^2-4\cdot a\cdot c}}
{2a}$$
```

$$x_{1/2} = \frac{-b \pm \sqrt{b^2-4\cdot a\cdot c}}{2a}$$

Even the markup of very advanced mathematical concepts is rather simple.

```
$$\int_a^b f(x)\,dx =
\lim_{\max \Delta x_k\rightarrow 0}
\sum_{k=1}^{n}f(x_k^*)\Delta x_k$$
```

$$\int_a^b f(x)\,dx = \lim_{\max \Delta x_k\rightarrow 0} \sum_{k=1}^{n} f(x_k^*)\Delta x_k$$

Matrices are marked very similarly to tables, except that the environment `array` is used. For example

```
$$\left(
\begin{array}{ccc}
a_{11} & \ldots & a_{1n}\\
\vdots & \ddots & \vdots \\
a_{n1} & \ldots & a_{nn}\\
\end{array}
\right)$$
```

$$\left(\begin{array}{ccc} a_{11} & \ldots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \ldots & a_{nn} \\ \end{array} \right)$$

Cross-referencing. Often authors need to make a reference to a specific section, page, figure, table or an equation from within their document. Since documents usually dynamically evolve and are frequently changed, manual maintenance of these references would be extremely tiresome. LaTeX supports automatic cross-referencing in a very convenient way. The document author assigns labels to specific objects (e.g., sections, figures, equations) that should be referenced. This is done by using the command `\label`. When the reference is created, if number of the object is needed it is retrieved by the command `\ref{}`, and if the page number is needed it is retrieved by the command `\pageref{}`. In the next example, a reference to an equation is illustrated.

```
\begin{equation}\label{eq:pitagora}
a^2 + b^2 = c^2
\end{equation}
The equation (\ref{eq:pitagora}) on
the page \pageref{eq:pitagora} \ldots
```

$$a^2 + b^2 = c^2 \qquad (1)$$

The equation (1) on the page 1 ...

Bibliographies. With LaTeX creating bibliographies and citing is very convenient and it can be fully automated if the system BibTeX is used. In this case, articles that are going to be cited are described by using a specialized syntax in separate `bib` files. Citation records are usually already available in this format on the Internet.

```
@ARTICLE{pitagorin-tekst,
   author = "Pitagora",
   title = "O odnosima stranica u pra-
vouglom trouglu",
    journal = "Zbornik starogr\v cke
matematike",
   volume = 33,
   number = 3,
   pages = "389--404",
   year = "510pne"
   }
```

When the citation is needed, the command `\cite{pitagorin-tekst}` is entered. The author can choose between several formats for displaying citations and references. In this example, the `plain` format is used which uses numbers for citations.

```
1 Pythagoras has in
2 \cite{pitagorin-tekst}
3 \bibliography{mybib}{}
4 \bibliographystyle{plain}
```

Pythagoras has in [1] ...

References

[1] Pitagora. *O odnosima stranica u pravouglom trouglu.* Zbornik starogrčke matematike 33(3):389–404, 510pne.

Creating new commands and environments. Full power of LaTeX can be seen only when users are trained for programming and creating new commands and environments. This way they can adjust the system to fit their specific needs. For example, mathematicians often use the text like a_1, a_2, … a_n in their documents. Instead of typing the long sequence `a_1,a_2, \ldots a_n$`, every time when it is needed, it is possible to create a new command `sequence` which displays the necessary text.

```
1 \newcommand{\sequence}
2    {$a_1, a_2, \ldots a_n$}
3 \nizelem and another \nizelem
```

$a_1, a_2, \ldots a_n$ and another $a_1, a_2, \ldots a_n$

New commands can also have their arguments.

```
1 \newcommand{\sequence}[1]
2 {\ensuremath{#1_1, #1_2, \ldots #1_n}}
3 \sequence{a} and another \sequence{b}
```

$a_1, a_2, \ldots a_n$ and another $b_1, b_2, \ldots b_n$

New environments can be created in a very similar way. Creating new commands and environments is usually considered to be an advanced way of using LaTeX, so the user is referred to consult an additional documentation on this topic.

4 Conclusions

TEX and LaTeX are document preparation systems that have been successfully used for many years for creating scientific and technical documentation. They are mainly used and popularized by mathematicians and physicists. Documents produced by these systems have an extremely appealing visual quality. By its design, the system LaTeX enforces the separation of the logical structure and the visual presentation of documents. Since the base LaTeX system is very flexible and adjustable, its numerous additions and specialized packages make it very convenient for creating different types of documents. The main shortcoming of these systems is that the learning curve is slow, and that novice users have much more problems in the beginning, compared to using WYSIWIG systems. However, when these early problems are resolved, gains are enormous. Also, creating ill structured, unreadable and ugly documents is much harder then with WYSIWIG tools.

[1] Macro is a kind of procedure, similar to the procedures in programming languages

Bibliography

Goossens, Michel, Frank Mittlebach, and Alexander Samarin. 1993. *The LaTeX Companion*, Addison-Wesley, first edition.

Knuth, Donald E. 1986. *The TeXbook*, Addison-Wesley Professional.

Lamport, Leslie. 1994. *LaTeX: A Document Preparation System: User's Guide and Reference Manual*, Addison-Wesley Professional, second edition.

Nenadić, Goran, Predrag Janičić, and Aleksandar Samardžić. 2003. *LaTeX za autore.* Kompjuter biblioteka.

Oetiker, Tobias, Hubert Partl, Irene Hyna and Elisabeth Schlegl. 2008. *The not so Short Introduction to LaTeX*, http://tobi.oetiker.ch/lshort/lshort.pdf