

# Best L<sup>A</sup>T<sub>E</sub>X practices

Jan Gosmann

February 17, 2017

# Why is this hard?

- One of the oldest pieces of software you are using.
- T<sub>E</sub>X was released 1978, feature complete since 1989.
- L<sup>A</sup>T<sub>E</sub>X was released 1985.
- Best practices change over time.
- Many recommendations on the Internet are outdated.

- 1 The T<sub>E</sub>X environment
- 2 Markup
- 3 Special characters
- 4 Package recommendations
- 5 Grab bag

# The T<sub>E</sub>X environment

# Why and why not to use L<sup>A</sup>T<sub>E</sub>X?

## Pro

- Text based format good for version control and collaboration.
- Beautiful typography.
- Semantic markup (focus on content).

## Contra

- Horrible language.
- Compile errors hard to track down.
- Sometimes hard to get exactly what you want.

- T<sub>E</sub>X: Original “low-level” typesetting system by Donald Knuth.
- L<sup>A</sup>T<sub>E</sub>X: Abstraction level build on T<sub>E</sub>X to isolate the user from typesetting decisions.
- ConT<sub>E</sub>Xt: In some ways similar to L<sup>A</sup>T<sub>E</sub>X, but provides easy access to advanced typographic control. Uses “unified system” instead of individual packages.

Probably best to stick to L<sup>A</sup>T<sub>E</sub>X because of abstraction level and templates provided by publishers.

- `latex`: Used to be the original L<sup>A</sup>T<sub>E</sub>X compiler to DVI, but now defaults to `pdflatex`.
- `pdflatex`: Compiles to PDF and supports special PDF features.
- `xelatex`: Adds support for PDF, UTF-8, and system fonts.
- `lualatex`: Adds support for PDF, UTF-8, system fonts, and Lua scripting.

Recommendations: Use `lualatex` if you can (first stable version was released last year), `xelatex` is also a good choice. Sometimes you have to resort to `pdflatex` (e.g., for the CogSci template or specific `microtype` features). Use `latex` only if you absolutely require specific packages (e.g., `pstricks`) not supported by the other compilers.

# How to install L<sup>A</sup>T<sub>E</sub>X?

- Via the package manager of your Linux distribution.
- I prefer TeX Live available for all major OS. Yearly releases, gives you *all* the packages in the most recent versions.



## Useful tools

- `texdoc`: Quickest way to pull up package documentations.
- `chktex`: Static checker, warns you about things easy to overlook. I recommend using it as an automatic checker in your favourite editor (e.g., with the Syntastic plugin in Vim).
- `latexmk`: Best way to compile  $\text{\LaTeX}$ .

### My `.latexmkrc`

```
$lualatex = 'lualatex -synctex=1 %0 %S';  
$pdflatex = 'pdflatex -synctex=1 %0 %S';  
$xelatex = 'xelatex -synctex=1 %0 %S';  
$pdf_mode = 4; # 1 for pdflatex, 4 for lualatex, 5 for xelatex
```

## pdf<sub>l</sub>atex

```
\usepackage[utf8]{inputenc}  
\usepackage[T1]{fontenc}
```

## x<sub>e</sub>l<sub>a</sub>tex and l<sub>u</sub>a<sub>a</sub>tex

```
\usepackage{fontspec}
```

Also, consider loading the `microtype` package, but you might want to disable protrusion for the table of contents (see section 9 of the manual).

# Markup

# Semantic markup

- Mark what things are, not what they should look like.
- Allows you to easily adjust formatting.

Example: Common definitions I use

```
\newcommand{\mat}[1]{\bm{#1}} % matrix  
\newcommand{\vc}[1]{\bm{#1}} % vector
```

## Special text styles

- Use `\emph` to emphasize text because it uses italics correction and can be nested: This is `\emph{emphasized with \emph{even more} emphasis}`. produces “This is *emphasized with even more emphasis*.”.
- Consider the semantic difference between `\emph{apples}`, `\emph{oranges}`, and `\emph{bananas}` (three individual items) and `\emph{A, B, and C}` (a single entity like a proper name).
- Use `\text??` commands for formatting instead of `{\?? ...}`, e.g. `\textbf{text}` instead of `{\bf text}`. The `\text??` commands can be nested and `\textit` has italics correction.

# Equations

- Use `equation` and `amsmath` environments, not `$$` for display equations.
- Equations are part of sentences.
- Do not forget punctuation.
- Do not introduce paragraph breaks around equations by additional empty lines (if it does not end the paragraph).

## Example

The input is given by

```
\begin{equation} % no empty line before this
    J(x) = \alpha_i e_{-x}(t) \text{.}
\end{equation}
```

## Formatting in equations

- Only variables (and measured physical constants) should be in italics (default).
- Common functions and operators like `max`, `cos` should be made upright by prefixing them with `\`, e.g. `\max` (`max`), **not** `max` (`max`).
- Use `\DeclareMathOperator` from the `amsmath` package to declare new operator names.
- Subscripts etc. that are not a variable should be made upright with `\mathrm` or `\mathsf`, e.g. `\tau_{\mathrm{syn}}` ( $\tau_{\text{syn}}$ ), **not** `\tau_{syn}` ( $\tau_{syn}$ ), but `a_i`, `\ 0 < i < 4` ( $a_i$ ,  $0 < i < 4$ ).
- Latex treats each letter as a single variable. Enclose multi-letter variables with either `\mathit` or `\mathsf{it}` to ensure proper spacing.
- The `commath` package provides some useful commands and make it easy to correctly set differentials.

# Special characters



- Use non-breaking space `~` where you would never want a linebreak. For example, `Dr.~Who`, `Figure~\ref{fig:a}`, or `(a)~this`, or `(b)~that`.
- Use either non-breaking space `~` or explicit normal length space `\quad` after periods that do not end a sentence because the intersentence space in English can be larger than the interword space.
- Use intersentence spacing when sentence ends in all caps word. For example, `Independent accumulator is abbreviated with IA\@`. (Chktex warns about this.)

- Use ```text''` to produce actual quotes: “text”.
- **Not** `"text"` which gives "text".
- Same for single quotes.
- Or use `\enquote` from the `csquotes` package.

# Horizontal bars aka dashes

Hyphen	-	-	connects words, e.g., good-hearted
en dash	–	--	span or range of numbers, e.g., chapters 5–9
em dash	—	---	denotes break in sentence or sets off source of a quote
Minus	−	\$- or \textminus	
Plus	+	+	(here for comparison to minus)

- Vertical alignment comparison: − — — — +.
- Minus has same height and width as the horizontal bar of the plus.
- The em dash for sentence breaks is traditionally set without (or only tiny) spaces around it.

# Package recommendations

- Use `biblatex` package with `biber` backend if possible (most modern package, most features, UTF-8 support).
- Templates provided by journals often use older packages like `natbib` and require the old `bibtex` backend.
- Use appropriate citation commands, e.g. `\parencite` vs `\textcite` with `biblatex`.
- Avoid parenthesis in parenthesis (might require some manual piecing together with different lower level citation commands).
- Using reference management software can be a good idea. (I use Zotero with the “Better Bib(La)TeX” plugin, others are Papers for macOS and Mendeley.)

# Tables

- Never, ever use vertical rules.
- Never use double rules.
- Caption goes above table, not below (in contrast to figures).
- Align numeric columns on period.
- Use booktabs package and read its documentation.

Item		
Animal	Description	Price (\$)
Gnat	per gram	13.65
	each	0.01
Gnu	stuffed	92.50
Emu	stuffed	33.33
Armadillo	frozen	8.99

## Some further package recommendations

- `nag` warns about usage of outdated packages and commands (load with `\usepackage[l2tabu,orthodox]{nag}`, consider adding the `abort` option).
- `amsmath` provides a number of useful equation environments and related commands.
- `graphicx` (it is more modern and has more functionality than `graphics`)
- `siunitx` provides configurable formatting for numbers and quantities with units.
- `tikz` allows to create great figures, but has a steep learning curve.
- `komascript` is a collection of feature-rich document classes and packages.
- Some like `memoir` (for book-style documents).
- `komascript` vs `memoir`

Grab bag



- [overleaf.com](https://overleaf.com)
  - Pros: web interface, Git access
  - Cons: version tracking is lacking, only  $\text{\LaTeX}$  inline comments
- [github.com](https://github.com)
  - Pros: PRs allow discussion threads in code, good version tracking
  - Cons: more complicated to use?
- [sharelatex.com](https://sharelatex.com) (have not tried it yet)
- Use one sentence per line. (You might want to set your editor to soft wrap mode.)

## Forward and inverse search

- Jump from tex-file to corresponding spot in the PDF (forward).
- Jump from the PDF to the corresponding spot in the tex-file (backward).
- Requires to compile tex with `-synctex=1`.
- Further setup depends on editor and PDF viewer.

## Further reading

- <https://www.olivieverdier.com/posts/2013/07/15/modern-latex/>
- <http://practicaltypography.com/>
- `texdoc l2tabuen`
- <https://jakubmarian.com/hyphen-minus-en-dash-and-em-dash-difference-and-usage-in-english/>
- <http://tex.stackexchange.com/questions/7742/what-are-the-strengths-and-weaknesses-of-koma-script-and-memoir>