

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ – You want it!  
Here's how.  
Stolen from here and there ...

Ronni Grapenthin

Geophysical Institute, University of Alaska  
Fairbanks

June 29, 2010



CTAN lion drawing by Duane Bibby;  
thanks to [www.ctan.org](http://www.ctan.org)

# Outline

- 1 What is  $\text{\LaTeX}$  anyway?
- 2 Why use  $\text{\LaTeX}$ ?
- 3 Work Flow
- 4 Commands and Environments
- 5 Document Structure
- 6 “Hello  $\text{\LaTeX}$ ”
- 7 Packages, Bibliographies, Images
- 8 Drawbacks
- 9 Resources
- 10 Writing a Paper

# What is $\text{\LaTeX}$ anyway?

$\text{\LaTeX}$ ...

a non-interactive typesetting utility that processes source text that is marked-up with a logical structure.

Our hero,  
developer of  $\text{\TeX}$ :



Donald E. Knuth;  
<http://www-cs-faculty.stanford.edu/~uno>

# What is $\text{\LaTeX}$ anyway?

$\text{\LaTeX}$ ...

a non-interactive typesetting utility that processes source text that is marked-up with a logical structure.

- 1978: Don Knuth starts working on  $\text{\TeX}$ , powerful and flexible typesetting utility

Our hero,  
developer of  $\text{\TeX}$ :



Donald E. Knuth;  
<http://www-cs-faculty.stanford.edu/~uno>

# What is $\text{\LaTeX}$ anyway?

$\text{\LaTeX}$ ...

a non-interactive typesetting utility that processes source text that is marked-up with a logical structure.

- 1978: Don Knuth starts working on  $\text{\TeX}$ , powerful and flexible typesetting utility
- high quality output, neat math typesetting

Our hero,  
developer of  $\text{\TeX}$ :



Donald E. Knuth;  
<http://www-cs-faculty.stanford.edu/~uno>

# What is $\text{\LaTeX}$ anyway?

$\text{\LaTeX}$ ...

a non-interactive typesetting utility that processes source text that is marked-up with a logical structure.

- 1978: Don Knuth starts working on  $\text{\TeX}$ , powerful and flexible typesetting utility
- high quality output, neat math typesetting
- 1984: Leslie Lamport creates  $\text{\LaTeX}$ , uses  $\text{\TeX}$  as typesetting engine

Our hero,  
developer of  $\text{\TeX}$ :



Donald E. Knuth;  
<http://www-cs-faculty.stanford.edu/~uno>

# What is $\text{\LaTeX}$ anyway?

## $\text{\LaTeX}$ ...

a non-interactive typesetting utility that processes source text that is marked-up with a logical structure.

- 1978: Don Knuth starts working on  $\text{\TeX}$ , powerful and flexible typesetting utility
- high quality output, neat math typesetting
- 1984: Leslie Lamport creates  $\text{\LaTeX}$ , uses  $\text{\TeX}$  as typesetting engine
- with  $\text{\LaTeX}$  focus on content rather than presentation

Our hero,  
developer of  $\text{\TeX}$ :



Donald E. Knuth;  
<http://www-cs-faculty.stanford.edu/~uno>

# What is $\text{\LaTeX}$ anyway?

## $\text{\LaTeX}$ ...

a non-interactive typesetting utility that processes source text that is marked-up with a logical structure.

- 1978: Don Knuth starts working on  $\text{\TeX}$ , powerful and flexible typesetting utility
- high quality output, neat math typesetting
- 1984: Leslie Lamport creates  $\text{\LaTeX}$ , uses  $\text{\TeX}$  as typesetting engine
- with  $\text{\LaTeX}$  focus on content rather than presentation
- comes with tools that automate indexing, bibliographies, referencing, etc.

Our hero,  
developer of  $\text{\TeX}$ :



Donald E. Knuth;  
<http://www-cs-faculty.stanford.edu/~uno>



# Why use L<sup>A</sup>T<sub>E</sub>X? (1)

Poll: How many of **you** have ever lost work to MS Word?

# Why use L<sup>A</sup>T<sub>E</sub>X? (1)

Poll: How many of **you** have ever lost work to MS Word?

*“[...] I wanted to produce a pdf file of the book for the web page. Now the amazing result: without any major changes the book compiled using **pdflatex**!! I do not know of any other typesetting system that is as stable over more than 25 years.”*

Walter Gander; Writing the first L<sup>A</sup>T<sub>E</sub>X-Book;  
TUG 2010: TeX's 2<sup>5</sup> anniversary, **June 28-30, 2010**

## Why use L<sup>A</sup>T<sub>E</sub>X? (2)

- **free & sexy** – yeah!

## Why use L<sup>A</sup>T<sub>E</sub>X? (2)

- **free & sexy** – yeah!
- **cross-platform** –
  - compiler: Implementations for most OS's exist
  - user input: text file, highly portable
  - output: device independent (dvi) files (older than pdf)

## Why use L<sup>A</sup>T<sub>E</sub>X? (2)

- **free & sexy** – yeah!
- **cross-platform** –
  - compiler: Implementations for most OS's exist
  - user input: text file, highly portable
  - output: device independent (dvi) files (older than pdf)
- **separation of content and style** – Typographers exist for a reason!

## Why use L<sup>A</sup>T<sub>E</sub>X? (2)

- **free & sexy** – yeah!
- **cross-platform** –
  - compiler: Implementations for most OS's exist
  - user input: text file, highly portable
  - output: device independent (dvi) files (older than pdf)
- **separation of content and style** – Typographers exist for a reason!
- **scalable** – Word gets upset with long documents

## Why use L<sup>A</sup>T<sub>E</sub>X? (2)

- **free & sexy** – yeah!
- **cross-platform** –
  - compiler: Implementations for most OS's exist
  - user input: text file, highly portable
  - output: device independent (dvi) files (older than pdf)
- **separation of content and style** – Typographers exist for a reason!
- **scalable** – Word gets upset with long documents
- **stable & mature** – Developed since 1978 (T<sub>E</sub>X), 1985 (L<sup>A</sup>T<sub>E</sub>X), see quote above

## Why use L<sup>A</sup>T<sub>E</sub>X? (2)

- **free & sexy** – yeah!
- **cross-platform** –
  - compiler: Implementations for most OS's exist
  - user input: text file, highly portable
  - output: device independent (dvi) files (older than pdf)
- **separation of content and style** – Typographers exist for a reason!
- **scalable** – Word gets upset with long documents
- **stable & mature** – Developed since 1978 (T<sub>E</sub>X), 1985 (L<sup>A</sup>T<sub>E</sub>X), see quote above
- **safe!** – L<sup>A</sup>T<sub>E</sub>X creates output file without altering user input.



## Why use L<sup>A</sup>T<sub>E</sub>X? (2)

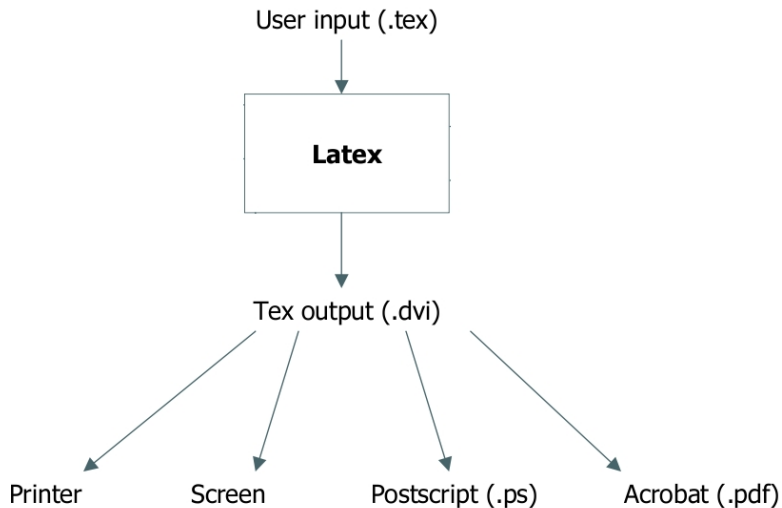
- **free & sexy** – yeah!
- **cross-platform** –
  - compiler: Implementations for most OS's exist
  - user input: text file, highly portable
  - output: device independent (dvi) files (older than pdf)
- **separation of content and style** – Typographers exist for a reason!
- **scalable** – Word gets upset with long documents
- **stable & mature** – Developed since 1978 (T<sub>E</sub>X), 1985 (L<sup>A</sup>T<sub>E</sub>X), see quote above
- **safe!** – L<sup>A</sup>T<sub>E</sub>X creates output file without altering user input.
- **control & flexibility** – the document is fully in your hands, if you want

User input (.tex)



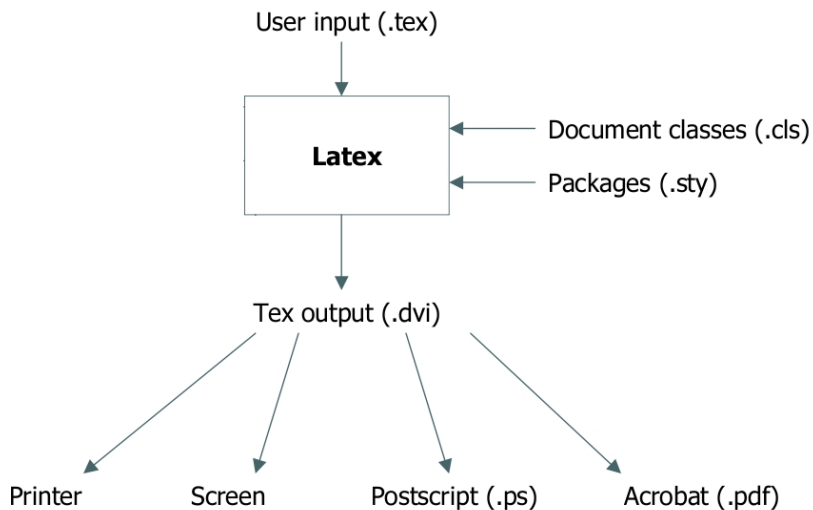
courtesy of Andrew Roberts;  
<http://www.andy-roberts.net>

# Work Flow



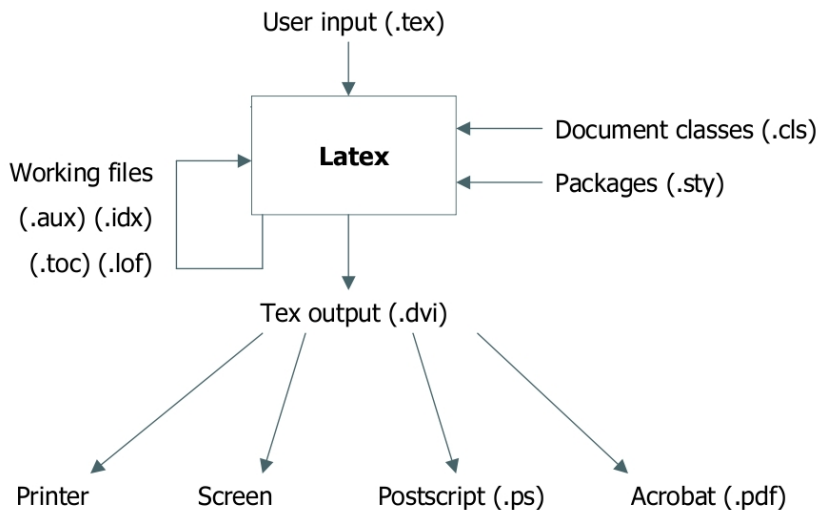
courtesy of Andrew Roberts;  
<http://www.andy-roberts.net>

# Work Flow



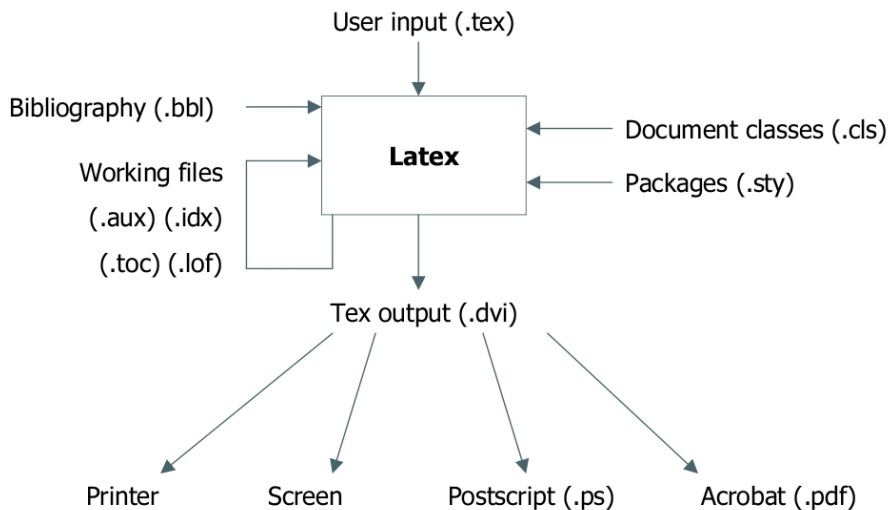
courtesy of Andrew Roberts;  
<http://www.andy-roberts.net>

# Work Flow



courtesy of Andrew Roberts;  
<http://www.andy-roberts.net>

# Work Flow



courtesy of Andrew Roberts;  
<http://www.andy-roberts.net>

## Command Format

```
\commandname [option] {argument}
```

- list of **commands** here (14 pages):

<http://www.ntg.nl/doc/biemesderfer/ltxcrib.pdf>

- **options** are optional, duh! can be flags or value assignments
- **arguments** are mandatory, for some commands empty, though

# Commands and Environments (1)

## Command Format

```
\commandname [option] {argument}
```

## Example

```
\includegraphics [width=.75\textwidth] {./ctanlion.pdf}
```

- list of **commands** here (14 pages):  
<http://www.ntg.nl/doc/biemesderfer/ltxcrib.pdf>
- **options** are optional, duh! can be flags or value assignments
- **arguments** are mandatory, for some commands empty, though



### Environments

```
\begin{environment-name}  
... contents ...  
\end{environment-name}
```

- formatting: text alignment, bullet-points, figures, tables, math, ...
- can be nested
- format for some environments more complex, check manual!
- some commands are valid in one environment, but not in others

## Commands and Environments (2)

### Environments

```
\begin{environment-name}  
... contents ...  
\end{environment-name}
```

### Example

```
\begin{document}  
All the contents of your paper  
\end{document}
```

- formatting: text alignment, bullet-points, figures, tables, math, ...
- can be nested
- format for some environments more complex, check manual!
- some commands are valid in one environment, but not in others

# Document Structure

```
\documentclass{article}  
\usepackage{babel}
```

```
\begin{document}  
\title{An Example}  
\maketitle
```

...

*Contents*

...

```
\begin{bibliography}
```

...

```
\end{bibliography}
```

```
\end{document}
```

Preamble

Front matter

Body

Back matter

# “Hello L<sup>A</sup>T<sub>E</sub>X” – A simple example

---

```
\documentclass{article}

%Here starts the document. I am a comment.
\begin{document}
  %The content
  Hello \LaTeX!
\end{document}
```

---

The result:

Hello L<sup>A</sup>T<sub>E</sub>X!

- standard document classes: article, book, letter, report, slides; determine structure of document (article has sections, no chapters)
- % is comment sign, everything that follows is ignored.
- document content is nested in document environment, that's what L<sup>A</sup>T<sub>E</sub>X prints

## Packages

```
\usepackage[options]{package}
```

- packages extend core functionality of  $\text{\LaTeX}$ , makes everything faster
- by loading packages, new environments / commands are added
- some come with  $\text{\LaTeX}$ , for complete list check [www.ctan.org](http://www.ctan.org)
- examples: `natbib` (citation), `fancyhdr` (custom header and footer), `graphicx` (...)

- bibtex is reference management / citation system
- references are stored in text file in special format
- GUIs exists to allow easier management: JabRef, mendeley desktop
- many publishers offer references in BibTeX format

- bibtex is reference management / citation system
- references are stored in text file in special format
- GUIs exists to allow easier management: JabRef, mendeley desktop
- many publishers offer references in BibTeX format

## sample.bib file

```
@article{Mogi1958,  
  author = {Mogi, K},  
  journal = {Bull. Eq. Res. Inst. Univ. Tokyo},  
  pages = {99--134},  
  title = {{Relations between eruptions of various  
  volcanoes and the deformations of the ground  
  surface around them.}},  
  volume = {36},  
  year = {1958}  
}
```

## Usage

```
\documentclass[12pt]{article}  
%include citation commands  
\usepackage{natbib}  
\begin{document}
```

As shown by `\citet{Mogi1958}` **\dots**  
We know that ... `\citep[e.g. ,][]{Mogi1958, Mogi1959}`

```
\bibliographystyle{agu04} %agu04.bst  
\bibliography{sample2}  
\end{document}
```

Execute ... (there are no typos here):

```
> pdflatex sample_bib
> bibtex sample_bib
> pdflatex sample_bib
> pdflatex sample_bib
```

To get:

As shown by *Mogi* (1958) ... We know that ... (e.g., *Mogi*, 1958, 1959)

## References

Mogi, K. (1958), Relations between eruptions of various volcanoes and the deformations of the ground surface around them., *Bull. Eq. Res. Inst. Univ. Tokyo*, 36, 99–134.

Mogi, K. (1959), Relations between eruptions of various volcanoes and the deformations of the ground surface around them., *Bull. Eq. Res. Inst. Univ. Tokyo*, 36, 99–134.



## Including Images

```
\includegraphics[options]{filename}
```

- `includegraphics` options: `width`, `height`, `scale`, `angle`
- graphic formats **latex**: PS, EPS ; **pdf<sub>l</sub>atex**: PDF, PNG, JPEG
- omit suffix in filename and  $\text{\LaTeX}$  finds fitting format with given name

## Including Images

```
\includegraphics[options]{filename}
```

## Figure environment, Example

```
\begin{figure}  
\includegraphics[width=39pc]{figurename}  
\caption{\label{source_plot} Whatever caption you want this  
figure to have ...}  
\end{figure}
```

- `includegraphics` options: `width`, `height`, `scale`, `angle`
- graphic formats **latex**: PS, EPS ; **pdflatex**: PDF, PNG, JPEG
- omit suffix in filename and  $\text{\LaTeX}$  finds fitting format with given name

## Including Images

```
\includegraphics[options]{filename}
```

## Figure environment, Example

```
\begin{figure}  
\includegraphics[width=39pc]{figurename}  
\caption{\label{source_plot} Whatever caption you want this  
figure to have ...}  
\end{figure}
```

## Referring to the above figure in text

Figure `\ref{source_plot}` clearly shows what's going on ...

- `includegraphics` options: `width`, `height`, `scale`, `angle`
- graphic formats **latex**: PS, EPS ; **pdflatex**: PDF, PNG, JPEG
- omit suffix in filename and  $\text{\LaTeX}$  finds fitting format with given name

## (possible) drawbacks

- no spellcheck – get a decent text editor
- no grammar check – oh well
- no tracking of changes – version control might help

- **L<sup>A</sup>T<sub>E</sub>X** for Windows: MikTeX – <http://www.miktex.org>
- A search engine should be the first place to ask for help!
- **Comprehensive TeX Archive Network** – <http://www.ctan.org>
- **Getting to Grips with LaTeX** –  
<http://www.andy-roberts.net/misc/latex/index.html>
- **The Not So Short Guide to LaTeX** –  
<ftp://ftp.tex.ac.uk/tex-archive/info/lshort/english/lshort.pdf>

**We'll do that in real time. File will be available at**

`http://www.gps.alaska.edu/programming/latex/paper\_template.tex`