An introduction to LaTeX, as well as Bibtex, Beamer, Tikz, and all that (Part I)

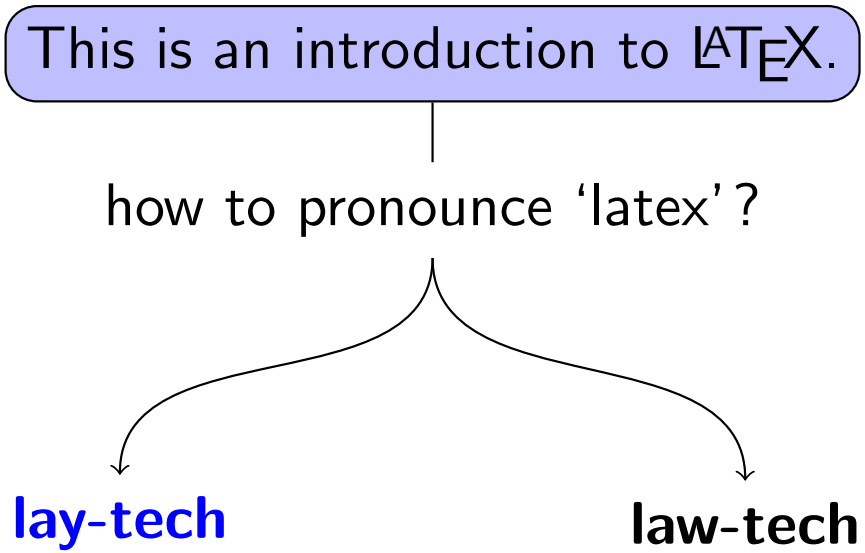
Nicolas Fillion

Department of Philosophy
Simon Fraser University

nfillion@sfu.ca

www.nfillion.com

This is an introduction to LaTeX.


This is an introduction to LaTeX.
how to pronounce ‘latex’?
lay-tech
law-tech

This is an introduction to LaTeX.

what is latex?

This is an introduction to LaTeX.

what is latex?

what latex is not like

This is an introduction to LaTeX.

what is latex?

what latex is not like

what latex is like

A LaTeX document typically looks like that:

```
\documentclass{article}
\usepackage[frenchb,russian]{babel}

\begin{document}
\tableofcontents

\section{Life is better with \LaTeX}
\LaTeX\ is \emph{absolutely brilliant}!

\bibliography{../biblio.bib}
\bibliographystyle{plain}
\end{document}
```

A LaTeX document typically looks like that:

```
\documentclass{article}
\usepackage[frenchb,russian]{babel}

\begin{document}
\tableofcontents

\section{Life is better with \LaTeX}
\LaTeX\ is \emph{absolutely brilliant}!

\bibliography{../biblio.bib}
\bibliographystyle{plain}
\end{document}
```

In LaTeX, **what you see is not what you get** ($\neg$ WYSIWYG).

```
\begin{propaganda}
```

Why using LaTeX?

Why using LaTeX?

- The open source philosophy/culture
- The technology is superior

Why using LaTeX?

- The open source philosophy/culture

  This culture promotes values such as **gratuity**, **freedom** (rejection of proprietary restrictions, rejection of non-disclosure agreements, independence, self-determination), **sharing**, **solidarity**, **contribution to the social good**, and **honesty** (no spywares).

- The technology is superior

Why using LaTeX?

The open source philosophy/culture

This culture promotes values such as **gratuity**, **freedom** (rejection of proprietary restrictions, rejection of non-disclosure agreements, independence, self-determination), **sharing**, **solidarity**, **contribution to the social good**, and **honesty** (no spywares).

The technology is superior

The word processor is a stupid and grossly inefficient tool for preparing text for communication with others.
There are much better ways of preparing paper & digital document, using a computer, than the word processor.

Objective Comparison of seven aspects

Aspect	Word	LaTeX
Speed small docs	★★★	★★☆
Speed big docs w/ graphics	★☆☆	★★★
Ease of use	★★★	★★☆
Layout quality	★★☆	★★★
Scientific features	★☆☆	★★★
Price+availability	★☆☆	★★★
Compatibility	★★☆	★★★★

Word vs. LaTeX, `http://openwetware.org/wiki/Word_vs._LaTeX`

WYSIWYG conflates **two conceptually different tasks**:

1. the **composition** of the text, *i.e.*, the choice of words and the determination of the logical structure;
2. the **typesetting** of the text, *i.e.*, the choice of fonts and of the visual representation (layout).

WYSIWYG conflates **two conceptually different tasks**:

1. the **composition** of the text, *i.e.*, the choice of words and the determination of the logical structure;
2. the **typesetting** of the text, *i.e.*, the choice of fonts and of the visual representation (layout).

The first is **the author's business**; the latter **the typesetter's business**. Conflating both is **distracting** and **suboptimal**.

Word Processor (WYSIWYG)

Examples: MS Word, Word Perfect, Open Office, etc.

VS

Text Editor

Examples: Notepad, Emacs, TEXshop, TEXnicCenter, etc.

Word Processor (WYSIWYG)

Examples: MS Word, Word Perfect, Open Office, etc.

- It has typesetting functionality (conflation).

VS

Text Editor

Examples: Notepad, Emacs, TEXshop, TEXnicCenter, etc.

- It has no typesetting functionality.

Word Processor (WYSIWYG)

Examples: MS Word, Word Perfect, Open Office, etc.

- It has typesetting functionality (conflation).
- It uses proprietary fonts (incompatibility).

VS

Text Editor

Examples: Notepad, Emacs, TEXshop, TEXnicCenter, etc.

- It has no typesetting functionality.
- It makes no use of fonts.

Word Processor (WYSIWYG)

Examples: MS Word, Word Perfect, Open Office, etc.

- It has typesetting functionality (conflation).
- It uses proprietary fonts (incompatibility).
- It is a proprietary binary data format (doc, rtf).

VS

Text Editor

Examples: Notepad, Emacs, TEXshop, TEXnicCenter, etc.

- It has no typesetting functionality.
- It makes no use of fonts.
- It only uses free and universal plain text (ASCII, UTF8).

Example of an `html` document.

Analogy with HTML

HTML		LaTeX
Text file with markups	$\Leftrightarrow$	Text file with markups
HyperText Markup Language	$\Leftrightarrow$	LaTeX Markup Language
Cascading Style Sheets	$\Leftrightarrow$	Document classes
Firefox Interpreter	$\Leftrightarrow$	LaTeX Typesetting Program
Webpage	$\Leftrightarrow$	PDF, PS, or DVI file

What you do when you use a markup language:

1. You write your text plainly and simply.
2. You indicate the **desired structure and formatting** of your document to LaTeX in the form of a **set of annotations**.

Common annotations are **intuitive, simple and easily remembered** (but there's a small learning curve). Also, you will find LaTeX-friendly text editors that will greatly help you.

TeX is the **basic typesetting engine** developed by Donald Knuth (1977-1982).

LaTeX is a large **set of macros**, initially developed by Leslie Lamport in the 1980s. These macro packages enables authors to typeset and print their work at the highest typographical quality, using a predefined, professional layout. LaTeX makes the average user's life much easier.

LaTeX also has some disadvantages:

- LaTeX doesn't work well for people who have sold their souls.
- The design of a whole new layout is difficult and takes a lot of time (but you will never do that).
- It is very hard to write unstructured and disorganized documents.
- Your hamster might, despite some encouraging first steps, never be able to fully grasp the concept of Logical Markup.

T. Oetiker, *The Not So Short Introduction to LaTeX 2ε*, `www.ctan.org`

```
\end{propaganda}
```

LaTeX distribution: the packages doing the underlying work

- Mac OS: MacTeX: `http://tug.org/mactex/`
- Windows: MikTeX: `http://miktex.org/`
- Linux: go to the software center or type "`sudo apt-get install texlive-full`"

LaTeX-friendly text editors

- Mac OS: TeXshop (built in) and TextWrangler2
- Windows: TeXnicCenter: `http://www.texniccenter.org/`
- Linux: `emacs`, `TexWorks` (built in), or TeXMaker

You now have a working version of LaTeX! The odyssey begins!

.tex LaTeX input file. Can be compiled with `latex`.

.sty LaTeX Macro package. This is a file you can load into your LaTeX document using the `\usepackage` command.

.cls Class files define what your document looks like. They are selected with the `\documentclass` command.

.dvi Device Independent File. This results from compiling with LaTeX.

.log Gives a detailed account of what happened during the last compiler run.

.toc Stores all your section headers.

.aux Another file that transports information from one compiler run to the next. Among other things, the .aux file is used to store information associated with cross-references.

.idx Contains information relative to your document' index.

LaTeX commands are case sensitive!!!

- They start with a **backslash** "\" and then have a string of characters. Command names are terminated by a space, a number or any other "non-letter."
- Some commands need a **parameter**, which has to be given between curly braces "`{ }`" after the command name. Some commands support optional parameters, which are added after the command name in square brackets "`[ ]`".

Typically, you will have:
`\command[option]{parameter}`

Your input file is the file `mydocument.tex`. Other files are generated by `latex` when you compile.

When `latex` parses your input file to compile a PDF, it has to **recognize markups**. Those characters have different meanings!

Your input file is the file `mydocument.tex`. Other files are generated by `latex` when you compile.

When `latex` parses your input file to compile a PDF, it has to **recognize markups**. Those characters have different meanings!

Special Characters

$ % ^ & _ { } ~ \

But you can type them by means of commands. Just add a "\":
`\# \$ \% \^{} \& \_ \{ \} \~`

You may want to write **comments** in your input file that `latex` will not compile. You do it by means of the symbol "%".

Ex: If you write

```
Lorem ipsum  % What is this text doing here?
dolor sit amet,consetetur sadipscing elitr,\ldots
```

The result will be

Lorem ipsum dolor sit amet, consetetur sadipscing elitr,...

You may want to write **comments** in your input file that `latex` will not compile. You do it by means of the symbol "%".

Ex: If you write

```
Lorem ipsum  % What is this text doing here?
dolor sit amet,consetetur sadipscing elitr,\ldots
```

The result will be

Lorem ipsum dolor sit amet, consetetur sadipscing elitr,...

You will fall in love with this feature. (Use "block comment".)

What your input file SHOULD contain

When `latex` compiles an input file, it expects it to follow a certain structure. Every input file **must start with** the command:

```
\documentclass[options]{classname}
```

It specifies the layout of your document (*e.g.*, `plain`, `article`, `report`, `book`, `letter`, etc).

Depending on your needs, you can use packages:

```
\usepackage[option]{packagename}
```

They add pre-programmed features. You will typically use some, but almost always the same (e.g. `fullpage`, `amssymb`, `amsmath`, etc).

Depending on your needs, you can use packages:

```
\usepackage[option]{packagename}
```

They add pre-programmed features. You will typically use some, but almost always the same (e.g. `fullpage`, `amssymb`, `amsmath`, etc).

$\Rightarrow$ That is the **preamble**. The body of the document begins **after**.

After the preamble comes the **body** of the document. The body of the document as such will follow between the commands

```
\begin{document}
⋮
\end{document}
```

This should be the last line of your input file.

After the preamble comes the **body** of the document. The body of the document as such will follow between the commands

```
\begin{document}
⋮
\end{document}
```

This should be the last line of your input file. (You can use this to leave out material.)

Multi-language Support

Package `babel`. I won't explain it here.
In the same document, you can use English, French, German, Russian, Korean, Chinese, Greek, etc. There will never be a single misinterpreted symbol.

Multi-language Support

Package `babel`. I won't explain it here.
In the same document, you can use English, French, German, Russian, Korean, Chinese, Greek, etc. There will never be a single misinterpreted symbol.

Quotation Marks

Do not use the symbol ". For multi-language support, use ‘‘ to open and ’’ to close.
LaTeX will **choose** for you the way to adjust it given the context.

Tilde

It's a special character, but we need it (e.g., in URLs).
Type `\~`.
E.g., typing "`publish.uwo.ca/\~{}mysite`" produces "publish.uwo.ca/~mysite"

Tilde

It's a special character, but we need it (e.g., in URLs).
Type \~.
E.g., typing "`publish.uwo.ca/\~{}mysite`" produces "publish.uwo.ca/~mysite"

Hyphenation

Each language has its rules. LaTeX will do it for you.
If you want to force hyphenation, just type "\-" in a word, *e.g.* "`transcen\-dental`".

Tilde

It's a special character, but we need it (e.g., in URLs).
Type \~.
E.g., typing "`publish.uwo.ca/\~{}mysite`" produces "publish.uwo.ca/~mysite"

Hyphenation

Each language has its rules. LaTeX will do it for you.
If you want to force hyphenation, just type "\-" in a word, *e.g.* "`transcen\-dental`".

Ellipsis

Don't write "...". The result will be much better with "`\ldots`".

You also have "`\vdots`" and "`\ddots`", producing "⋮" and "⋱".

Accents

\'e gives é \`e gives è \^e gives ê \"e gives ë

Accents

\'e gives é \`e gives è \^e gives ê \"e gives ë

Spacing

LaTeX ignores spaces in your input files.

- To insert an unbreakable space, use "~" (e.g., on page~3)
- To add space, use "\>", "\enskip", "\quad", "\qquad", etc.
- To begin a new paragraph, skip a line or type "\par".
- To force a line change, type "\\" or "\\[.5cm]".
- To add vertical and horizontal space, use "\vspace{1cm}", "\hspace{1cm}", "\vfill" and "\hfill".
- To change page, use "\pagebreak" or "\newpage".

Line spacing and margins

- You can use the package `setspace` and "`\doublespacing`".
- You can use the package `fullpage` and `geometry`.

Structuring your Paper (with the documentclass `article`)

- `\section{...section title...}`
- `\subsection{...}` and `\subsubsection{...}`
- `\paragraph{...}` and `\subparagraph{...}`
- For `book`, you also have `\chapter{...}` and `\part{...}`.

Line spacing and margins

- You can use the package `setspace` and "`\doublespacing`".
- You can use the package `fullpage` and `geometry`.

Structuring your Paper (with the documentclass `article`)

- `\section{...section title...}`
- `\subsection{...}` and `\subsubsection{...}`
- `\paragraph{...}` and `\subparagraph{...}`
- For `book`, you also have `\chapter{...}` and `\part{...}`.

Table of Contents

Just type "`\tableofcontents`" where you want it to appear. Use `\pagebreak` to have it on its own page.

Footnotes

Use the command "`\footnote{your text}`".
E.g.
"`Nietzche claimed that God is dead.\footnote{However, he did not specify a p-value.}`"

Footnotes

Use the command "\footnote{your text}".
E.g.
"Nietzche claimed that God is dead.\footnote{However, he did not specify a p-value.}"

Emphasis, Boldface, Italic

Use the command "\emph{your text}", "\textbf{your text}", "\textit{your text}".
E.g.
"\textbf{Nietzche} is clearly right in a sense, since \emph{Hilbert} has abandoned us, mere \textit{mortals}."

Environment

```
\begin{environment}
   text and commands
\end{environment}
```

Ex: `itemize`, `enumerate`, `flushleft`, `flushright`, `center`, `quote`, `quotation`, `verse`, `abstract`, `verbatim`, `tabular`,`figure`,`table`,...

Environment

```
\begin{environment}
   text and commands
\end{environment}
```

Ex: `itemize`, `enumerate`, `flushleft`, `flushright`, `center`, `quote`, `quotation`, `verse`, `abstract`, `verbatim`, `tabular`,`figure`,`table`,...

Some are more complex to use. You might use TEXnicCenter's and TEXshop's links. But I like to just google my questions.

Itemize

```
I have an example in many points:
\begin{itemize}
\item Text for item 1 here.
\item Text for item 2 here.
\end{itemize}
```

I have an example in many points:

- Text for item 1 here.
- Text for item 2 here.

Enumerate

```
I have an example in many points:
\begin{enumerate}
\item Text for item 1 here.
\item Text for item 2 here.
\end{enumerate}
```

I have an example in many points:

1. Text for item 1 here.
2. Text for item 2 here.

Center, flushright, flushleft

```
I have an example I want to centralize:
\begin{center}
Lorem ipsum dolor sit amet, consetetur sadipscing ...
\end{center}
```

I have an example I want to centralize:

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Quotation, quote

```
I have a great text I want to quote:
\begin{quotation}
Lorem ipsum dolor sit amet, consetetur sadipscing ...
\end{quotation}
```

I have an example I want to quote:

> *Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.*

Abstract

```
\begin{abstract}
Lorem ipsum dolor sit amet, consetetur sadipscing ...
\end{abstract}
```

Abstract

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Tabular

```
\begin{tabular}{cc|c}
p & q & p implies q\\
\hline
true & true & true\\
true & false & false\\
\end{tabular}
```

p	q	p and q
true	true	true
true	false	false

Tabular

```
\begin{tabular}{cc|c}
p & q & p implies q\\
\hline
true & true & true\\
true & false & false\\
\end{tabular}
```

p	q	p and q
true	true	true
true	false	false

Extra commands: `\cline`, `\multicolumn`. Package: `multirow`

Tabular

```
\begin{tabular}{cc|c}
p & q & p implies q\\
\hline
true & true & true\\
true & false & false\\
\end{tabular}
```

p	q	p and q
true	true	true
true	false	false

Extra commands: `\cline`, `\multicolumn`. Package: `multirow`
Eventually, the package `TikZ` might be a good alternative.

Input

For large projects, consider **splitting your document**:
`\input{filename}`
It inserts the content of the file named *filename.tex*.

```
\begin{document}

\section{Kant: the Limit of Pure Reason}
\input{/kant}
\section{Turing: the Range of Pure Reason}
\input{../church}

\end{document}
```

Hyperref

Use the package `hyperref` to have clickable links in your PDF documents. This is incredibly useful. If you do not like the default style of links, use something like this in the preamble:

```
\hypersetup{colorlinks,breaklinks,
linkcolor=black,urlcolor=black,
anchorcolor=black,citecolor=black}
```

Hyperref

Use the package `hyperref` to have clickable links in your PDF documents. This is incredibly useful. If you do not like the default style of links, use something like this in the preamble:

```
\hypersetup{colorlinks,breaklinks,
linkcolor=black,urlcolor=black,
anchorcolor=black,citecolor=black}
```

Cross-reference

Write `\label{labelname}` (in an environment, if that's what is labeled) and use `\ref{labelname}` to refer to an equation, figure, section, etc.

`\eqref` and `\pageref` are also available.

Now, a few things about **Typesetting Mathematics**...

This is just a few basic things; the rest will come as you go. For more, use google and ask!

Mathematical text **within a paragraph** is entered between $ and $ or between \begin{math} and \end{math}.

```
Add $a$ squared and $b$ squared
to get $c$ squared. Or, using
 algebraic notation:
$c^{2}=a^{2}+b^{2}$
```

Add a squared and b squared to get c squared. Or, using algebraic notation: $c^2 = a^2 + b^2$

To **set math apart** from the rest of the paragraph, it is preferable to **display it**. To do this, you can either enclose them in `\[` and `\]`, or between `\begin{displaymath}` and `\end{displaymath}`.

Other useful environments are:

- `equation` and `equation*`
- `align` and `align*`

I strongly recommend that you use **align**.

Example:

```
Add $a$ squared and $b$ squared to get $c$
 squared. Or, using a more mathematical approach:
\begin{displaymath} c^{2}=a^{2}+b^{2} \end{displaymath}
or you can type less for the same effect: \[a^2+b^2=c^2\]
```

Add a squared and b squared to get c squared. Or, using a more mathematical approach:

$$c^2 = a^2 + b^2$$

or you can type less for the same effect:

$$a^2 + b^2 = c^2$$

If you want LaTeX to enumerate your equations, you can use the align environment.

```
Now, we define a term to express the function $f$:
\begin{align}
tx_1x_2\ldots x_n & = \bigvee_{a_i\in2^n} \{t^{a_i}x_1x_2
\ldots x_n : f(a_i)=1\}\\
 & = \prod_i t^{a_i}x_1x_2\ldots x_n \in \bigvee
\end{align}
```

Now, we define a term to express the function f:

$$tx_1x_2\ldots x_n = \bigvee_{a_i\in 2^n} \{t^{a_i}x_1x_2\ldots x_n : f(a_i) = 1\} \quad (1)$$

$$= \prod_i t^{a_i}x_1x_2\ldots x_n \in \bigvee \quad (2)$$

There are differences between **math mode** and **text mode**. For example, in math mode:

- Most spaces and line breaks do not have any significance, as all spaces are either derived logically from the mathematical expressions, or have to be specified with special commands such as `\,` or `\quad` or `\qquad`.
- Empty lines are not allowed. Only one paragraph per formula.
- Each letter is considered to be the name of a variable and will be typeset as such. If you want to typeset normal text within a formula (normal upright font and normal spacing) then you have to enter the text using the `\textrm{...}` commands.

When writing documents, you often need a way to typeset "Lemmas", "Definitions", "Axioms" and similar structures.

```
\newtheorem{name}[counter]{text}[section]
```

For example:

```
\newtheorem{theo}{Murphy}[section]
\newtheorem{law}{Law}[section]
\begin{law} If there are two or more ways to do
 something, and one of those ways can result in a
catastrophe, then someone will do it.\end{law}
\begin{theo}If it could not not have gone wrong, then
 it could not have gone well.
```

There are tons of **sizes** for **parentheses and braces**. I suggest that you use `\left` and `\right` to let LaTeX manage that: For example:

```
\begin{align}
\mathcal{T}(f) = \left\{ \begin{array}{ll}
1 & f\in\left\{ g\mid \int_0^\infty g(\tau)d\tau
 \textrm{ is bounded}  \right\} \\[.25cm]
0 & f\in\left g\mid A\xrightarrow{g}{f\circ h}B \right
\end{array}\right.   \end{align}
```

$$\mathcal{T}(f) = \begin{cases} 1 & f \in \left\{ g \mid \int_0^\infty g(\tau)d\tau \text{ is bounded} \right\} \\ 0 & f \in \left\{ g \mid A \xrightarrow[f\circ h]{g} B \right\} \end{cases} \tag{3}$$

Detexify will be your friend for math symbols:

`http://detexify.kirelabs.org/classify.html`

Otherwise, there's the massive A4 symbols list (164 pages):

`tug.ctan.org/info/symbols/comprehensive/symbols-a4.pdf`

It is easy to **include figures or graphs** that you have made using another software or stolen online. First

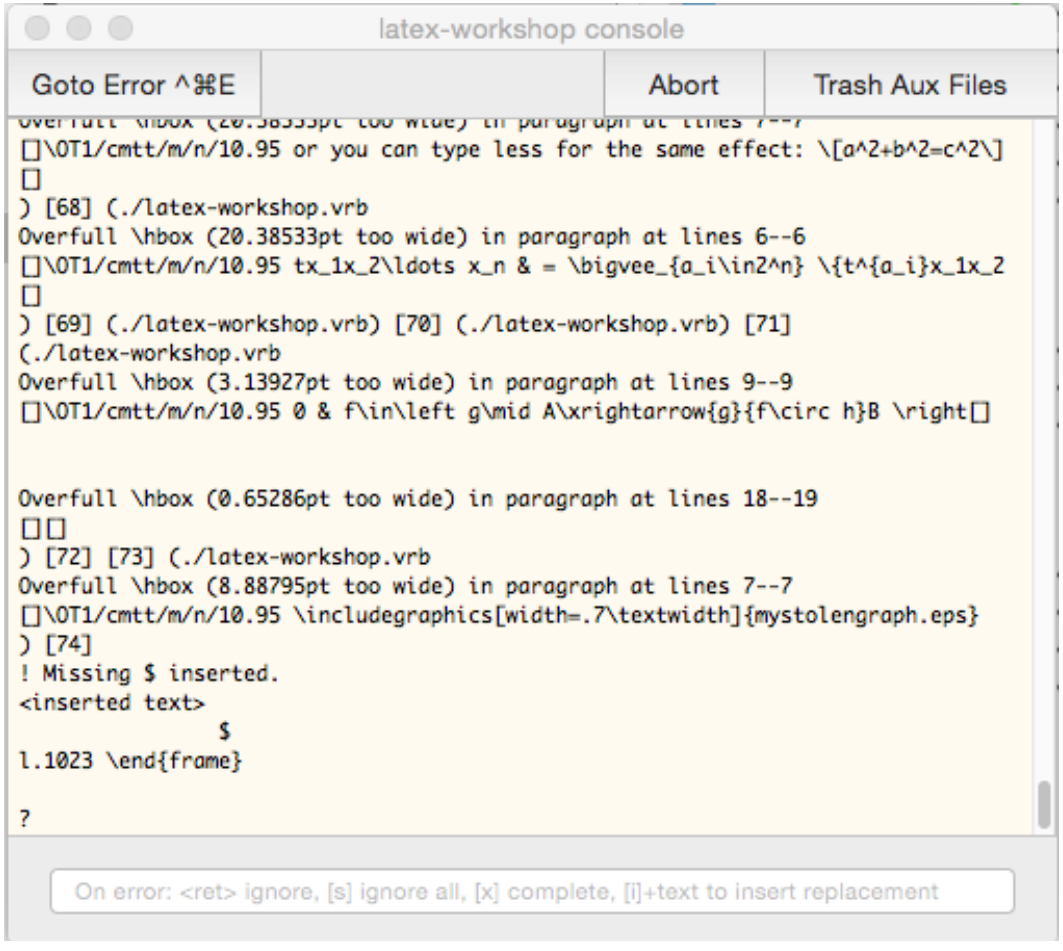
```
\usepackage{graphicx}
```

Then, simply type

```
\includegraphics[width=.7\textwidth]{mystolengraph.eps}
```

The `trim` and `clip` options are very useful!

Debugging. When you compile, there is a **console window**:



Use the error messages to fix things! **Press r + enter** to ignore.

What to do if you need help?

- Ask any LaTeX user!
- Tobias Oetiker, *The Not So Short Introduction to LATEX2*, GNU free documentation, 2007.
- The documentation available with TeXnicCenter.
- `http://www.nfillion.com/index.php/resources/latex-reference` (links to Detexify, Cheat Sheet, A4 symbols, templates for cv, term paper, thesis, etc, among other things)