

# IT Training

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

Kim Buckner

University of Wyoming

Apr. 06, 2017

- There are many distributions and LaTeX editors.
- I have successfully used two Windows distributions, TeXLive and MikTeX
- MikTeX seems to be falling behind in quality recently.
- And it only really runs on Windows.

## (more ...)

- TeXLive comes with an editor, TeXworks (on Windows)
- MiKTeX now also comes with TeXworks.
- LEd is another, installed on some IT machines with MiKTeX
- Then there is TeXnicCenter, WinShell, and others

## (more ...)

- You may have to install a spell checker, if you do, come talk to me if you have problems.
- Most of these now also run on Mac and Linux.
- Personally, I prefer GVIM/VIM on Linux and Windows.

## (more ...)

- That is because I do not like WYSIWYG editors and have used VI/VIM/GVIM for about 20 years.
- Many people like emacs. I won't comment further.
- If you are using Linux, my personal choice, LaTeX is provided with many distributions or can be installed easily.

# Why L<sup>A</sup>T<sub>E</sub>X?

- Because many academic disciplines are generally expected to be able to use it
- Because you can do much more with it and more quickly than Word, etc.
- Its free and easily installed
- Because we will be using it in this course.
- **Not** because it is faster than Word (although it can be)

# LaTeX on Windows.

- If you have your own computer, install TeX Live.
- If a department or IT machine and you are NOT an administrator, just ask, it is not difficult.
- Download the “install-tl-windows.exe” file, run it.
- This comes with TeXworks.

## (more ...)

- You must NOT use a “standard” WYSIWG editor, such as Word or Libreoffice.
- If you are creating PDF then you should be good to go and can use Acroread/Acrobat to view it.
- If making Postscript you may need to get GSView and Ghostscript.
- TexLive comes with PSview.



(more ...)

- It also comes with DVIOUT.
- Yap (yet another previewer) is a another Windows based dvi viewer that works well.  
(Comes with MiKTeX)

# Information

- A simple reference is on wikibooks, [en.wikibooks.org/wiki/LaTeX](http://en.wikibooks.org/wiki/LaTeX)
- The **Not So Short** guide is available at [www.ctan.org/tex-archive/info/lshort](http://www.ctan.org/tex-archive/info/lshort) in several languages other than English.
- There are lots of other docs on the web.
- My personal web page has these links too: [www.cs.uwyo.edu/~kbuckner](http://www.cs.uwyo.edu/~kbuckner)

- $\text{\LaTeX}$ , which we will be using, is based on Donald Knuth's  $\text{\TeX}$ .
- $\text{\TeX}$  is an engine which is based on a macro processor.
- Many people are involved in projects relating to  $\text{\TeX}$  and  $\text{\LaTeX}$ .
- I am not going to cover any explicit  $\text{\TeX}$  although you will probably see references to it in searches.

# What is in the document?

- Most editors default to a “.tex” file extension.
- Must have three (3) components
  - `\documentclass[options]{class}`
  - `\begin{document}`
  - `\end{document}`
- Text is just typed in
- Paragraphs are started by a blank line
- Extra whitespace is ignored

# Special characters

- Basic characters are [a-zA-Z0-9]
- . : ; , ? ! ' ' ( ) [ ] - / \* @
- + = -
- # \$ % & \_ { } can be escaped with a backslash
- | < > ~ ^ \ have to be printed with special commands.

# (more ...)

- \textbar \textless \textgreater
- \textasciitilde \textasciicircum  
\textbackslash

# Document classes

- *article*, journal publications
- *report*, technical reports, etc.
- *book*, theses, dissertations, things that need chapters
- And this one, *beamer* (I think better than *slide*)
- Each has its own standard (default) layout of pages and the document

# Options to the document class

- Defined by the class but ...
- 10pt | 11pt | 12pt to set the basic type size
- oneside | twoside, formats printing for one or both sides of a page
- onecolumn | twocolumn



## (more ...)

- letterpaper | legalpaper | a4paper | a5paper  
| b5paper | executivepaper
- final | draft, draft will cause a box to be printed around text that TeX does not know how to properly split on a line or page, a good thing!

# Packages

- You want to do special things, you need to load *packages*
- `\usepackage[options]{packages}`
- The *packages* can be a single package or a comma-separated list
- The *options* are given to all the listed *packages*

# Common packages

- **graphics** and **graphicx** allow you to include “jpg”, “eps”, “gif”, etc.
- **amssymb**, all the math you could ever want
- **color**, producing **color** in your document
- **xcolor**, if **color** is not enough.
- There are many more and you can pick and choose

# The document

- Consists of several components
- All of them can be adjusted
- Some should not be messed with
- –Refer to handout.–
- Also available in the WikiBook at  
[https://en.wikibooks.org/wiki/LaTeX/Page\\_Layout](https://en.wikibooks.org/wiki/LaTeX/Page_Layout)

## (more ...)

- There are lots of ways to adjust documents in LaTeX.
- The Page\_Layout section of the Wikibook has some.
- Or you can refer to the layout and then change those lengths.
- This is discussed in <https://en.wikibooks.org/wiki/LaTeX/Lengths>

# Spell Checking

- If you use one of the IDE it should be incorporated.
  - you may have to add a dictionary, ask for help
- You can use **aspell** or *hunspell* (free software)
- Can create a "special" dictionary but this is not good.
- If you go the bottom of my homepage, we can get a dictionary to install.

# Writing

# Modes

- Tex (and Latex) process input in one of three modes.
- Paragraph mode — ‘normal’ processing of text, sequence of words.
- Left-to-right (LR) mode — input is words and spaces, no line breaks **ever**.
- LR is normally used by default in tables, tabbing, and other special environments.



# Modes (cont.)

- Math mode — the input is treated as all mathematical symbols, a word like ‘is’ represents the product of variables  $i$  and  $s$ .
- Linebreaks may become an issue but not a problem.
- ‘Normal’ text input in math mode is normally italicized because all letters are assumed to be variables.

# Tables (1)

- These are similar to arrays.
- NOT used in *math* mode.
- `\begin{tabular}`
- Also has an alignment argument.
- Can additionally add | vertical bars to draw column separators.
- There are other special alignment options that can be used if needed.

# Tables (2)

- The `\hline` and `\cline` commands
- Both draw horizontal lines
- `\hline` is the width of the table
- `\cline` takes an additional 'columns' argument
- `\cline{3-5}` draws a line under columns 3, 4, and 5

# Tables (3)

- You may need text to span multiple columns
- `\multicolumn{n}{pos} {item}`
- The *n* is the number of columns to span
- The *pos* is an alignment argument just like for the original command
- The *item* is the text
- Handy if need different alignment for one column on one row.

# Tables (4)

- The special arguments for positioning include
- $p\{width\}$  which creates a `\parbox` that is *width* wide
- Parboxes are typeset in paragraph mode so lines wrap
- Saves from the “Overfull hbox” errors

- $m\{width\}$  somewhat like  $p$  but text centered vertically.
- $b\{width\}$  like  $p$  but text is flush bottom instead of top
- $>\{decl\}$  used before  $l, r, c, p, m, b$  inserts  $decl$  directly in front of the entry of the column.

- $\{decl\}$  used after l, r, c, p, m, b inserts *decl* directly right after of the entry of the column.
- $!\{decl\}$  used anywhere. Used to replace the vertical bar, |. Does not change the inter-column space.
- $@\{decl\}$  Suppresses the inter-column space and used *decl* instead.

# Tables (5)

- You may need to `\usepackage{array}` to make everything after the  $p$  work.
- Will you remember all of this? probably not.
- But now you can look it up because you know about it.



## (more ...)

- Sometimes you need really long tables, ones that extend across pages.
- *tabular* will not break on pages.
- Then you need something like `\usepackage{longtable}`.

## (more ...)

- It allows for headers (and footers) for each page.
- These are specified once, then are inserted just like page headers/footers.
- And you can specify a first header and a last footer.
- *longtable* supports its own captioning.
- Again, play with it to what is happening.

# Tabbing (1)

- Allows you to line things up
- Unlike basic **tabular**, it can be broken across pages
- `\begin{tabbing} rows \end{tabbing}`
- Each row is in LR mode
- Tab stops are numbered 0,1,2, etc.

# Tabbing (2)

- 0 is set to the prevailing left margin
- Use `\=` to set a tab stop
- Use `\>` to move to a tab stop
- Use `\\` to start a new line and go to tab stop 0
- `\kill` throws away the text and keeps the tab stop settings.

# Tabbing (3)

- `\+` increases the *left\_margin\_tab* by 1
- `\-` decreases the *left\_margin\_tab* by 1
- `\'` aligns the rest of the column right.

# Math

# Math

- Several ways to start Math mode:
- $\backslash\begin{math}\dots$
- $\backslash\begin{displaymath}\dots$
- $\backslash\begin{equation}\dots$
- $\backslash(\dots\backslash)$
- $\backslash[\dots\backslash]$
- $\$ \dots \$$

# Basic Math

- `\begin{math}...` `\( ... \)`, and `$...$` are synonymous.
- These start **math** mode.
- Spaces are ignored for layout purposes.
- Spaces may be needed to mark the ends of commands.
- Can use braces too.



# Formulae (1)

- `\[...]` and `\begin{displaymath}...`
- These are synonymous and result in an un-numbered formula.
- Same as regular math mode otherwise.
- `\begin{equation}...` begins a numbered formula.
- These are both for one-line formulae.

# Examples

This first is an un-numbered formula produced in the **displaymath** environment.

$$x^3 - 3x^2 - 5x + z = 635$$

The next type is a numbered formula produced in the **equation** environment.

$$\sum_{i=1}^{\mathbb{N}} x_i = \int_0^1 f \tag{1}$$

# (more ...)

- $\backslash[ x^3 - 3x^2 - 5x + z = 635 \backslash]$
- $\backslash\begin{equation}$   
 $\backslash\sum_{i=1}^{\mathbb{N}} x_i =$   
 $\backslash\int_0^1 f$   
 $\backslash\end{equation}$

# Formulae (2)

- By default all formulae are centered.
- Can be changed by the `\documentclass` option **fleqn**
- Causes them to be aligned left, fixed distance from the left margin
- By default all numbers are on the right side
- Use the option **leqno** to have them appear on the left instead

# Formulae (3)

- What if you want a multiline formula?
- Then use the **eqnarray** environment
- This is a basically a 3 column array
- The columns are separated by & and the ends of lines are marked by `\\`.
- Empty **columns** are possible with just a & and space

# More examples

```
\begin{eqnarray}
x &= & 17y \\
y &> & a + b + c + \nonumber \\
&& d + e + f + g
\end{eqnarray}
```

Produces something that looks like this

$$x = 17y \quad (2)$$

$$y > a + b + c + \\ d + e + f + g \quad (3)$$

## (more ...)

- Note there is no `\\` after the last line.
- If you put a pair in, you may have a blank line that is numbered.
- **`eqnarray`** is NOT used in math mode, it creates a math environment.



# Arrays

- Arrays are a 'sub' environment that can only be used in math mode
- Unlike **eqnarray** you can have as many columns as you like
- Like **eqnarray** they are formatted the same
- There is an extra argument to the environment
- $\backslash\begin{array}\{alignment\}$

# Arrays (cont.)

- The alignment is on a per column basis may be like {cclrc}
- Each character is one of center, *left*, or *right*

# Array example

```
\( \begin{array}{clr} a+b & c & 19 \\ f & R & 42 \\ a+b+c & M & 99 \end{array} \)
```

# Array example (cont.)

Which produces something like this:

$$a + b \quad c \quad 19$$

$$f \quad R \quad 42$$

$$a + b + c \quad M \quad 99$$