Introduction

- A simple introductory hands-on course. I assume that you know:
  - an editor
  - basic Unix commands
- $Latex$: a text formatter: not a WP
- Freely distributable
- written by Leslie Lamport, built upon Knuth's tex
- A division of labour
  - composition: author and editor
  - formatting & style: pre-defined by the program
- $latex$ borrows heavily from *Scribe*
  - declarative not procedural
  - descriptive commands

Document Parts

- `\documentstyle`
  - [options]: if multiple, separate options with commas
  - {style}: one of **`article`**, **`report`**, **`book`**, **`uwoletter`** or **`letter`**
- Title information
  - Title: multiple lines separated by \\
  - Author: multiple lines separated by \\
  - Date: optional — today's used if not included
- **`\begin{document}`**: mark the start of text
- **`\maketitle`**: place the title information
- Text body
- **`\end{document}`**: Always the last line.

Article Template

```
\documentstyle [11pt] {article}

\title{\bf Your Title here\\ }
\author{Your Name, DEPT\\your@email.address}
% \date{1996-03-18} % remove comment to produce the given date.

\begin{document}
\maketitle % Produces the title.

%       ...  Document Body  ...

\end{document} % Always the last command
```

Compiling

Exercise #1

1. Copy exercise files:

   ```
   tar -xf /usr/doc/courses/latex/ex.tar
   ```

2. Convert sample.tex to a .dvi file:

   ```
   latex sample.tex
   ```

3. Observe the files produced: `ls sample.*`

4. Convert the `sample.dvi` file into *PostScript*:

   ```
   dvitps -O sample.ps sample.dvi
   ```

Files Produced by *latex* and `dvitps`
(results from exercise)

Previewing: Saving time and paper.

`dvi2tty` a very rough screen approximation from the `.dvi` file:

- shows page and line breaks
- can be hard to read
- doesn't display included PostScript figures
- can be used to produce readable tty output with modifications to the input file.
- Try it: **`dvi2tty sample.dvi`**

`xdvi` A fairly accurate rendition of the `.dvi` file for X-windows based screens.

- Doesn't capture included PostScript figures

`ghostview` A rough to very good rendition (screen resolution dependent) of the `.ps` file. Useful if you have included PostScript figures.

dvi2tty Example

Replace with a printed copy of sample.text

xdvi Example

Replace with a printed copy of an xdvi screen or two.

ghostview Example

Replace with a printed copy of a ghostview screen or two.

300 dpi Laser Printer Example

Replace with a printed copy of sample.ps

Compiling & Previewing Summary

- Convert a *latex* file into Device Independent (`.dvi`) format:

  `latex` *filename*`.tex`

- Convert a `.dvi` file into PostScript

  `dvitps -O` *filename*`.ps` *filename*`.dvi`

- Convert a `.dvi` file into PostScript and send it to the printer:

  `dvitps` *filename*`.dvi | lpr -Pps`

- Preview a `.dvi` file on the screen

  `dvi2tty` *filename*`.dvi`

- Preview a `.dvi` or `.ps` file on an X-windows screen:

  `xdvi` *filename*`.dvi &`
  `ghostview` *filename*`.ps &`

Basic Formatting

The bulk of most texts is words which are formed into paragraphs and in $latex$ these are, for the most part, typed without change. There are a few things that you have to keep in mind though:

- Extra intra-word spaces are generally ignored
- Paragraphs are delimited by a blank line.
- Some characters with special meaning to $latex$:

  ```
  # $ % &  _ { } ~ ^ \
  ```

  The first seven of these can be included in your text by quoting with a backslash: \. The others require more sophisticated mechanisms. Characters like +=|<> are used primarily in mathematical formulas.

- Dashes:

  - Produces a short dash for hyphens as in part-time.

  -- Produces a longer dash used in number ranges: 45–54

  --- Produces a dash used as punctuation—like that

- *latex* commands start with a backslash followed by a number of characters—case is significant: `\section` is different from `\Section`.

- Manditory arguments to commands are enclosed in braces:

  ```
  \section{Preface}
  ```

  Optional arguments are enclosed in square brackets:

  ```
  \item [Characteristic]
  ```

- Double quotes are made using ‘‘ and ’’. Note that the editor *emacs* will generate this correctly when you type " when editing a file with the `.tex` extension.

Environments

- Environments are ways of limiting the scope of a *latex* command to a certain range.

- Environments can be delimited with curly braces:

  {*env-name ...text...*}

  or as \begin{*env-name*} ... \end{*env-name*} pairs.

Font changing environments

- {\em...*text*...} to delimit emphasised text. (Italicized, unless the surrounding text is already italicized.)

- Other common font changes are made in a similar way:

  {\bf Bold face} produces **Bold face**

  {\tt Typewriter} produces `Typewriter`

  {\sc Small Caps} produces Small Caps*

*font not available in Slide mode

Sectioning Commands

section{*Section Name*} Start a new numbered section. eg. **3. Section Name**

subsection{*Sub-Section Name*} Start a new numbered sub-section.
eg. **3.2. Sub-Section Name**

paragraph{*Paragraph Name*} Start a new numbered sub-sub-section.
eg. **3.4.2. Paragraph Name**

- Each of the sectioning commands can be followed by a star "*" to suppress the numbering.

- If a table of contents is produced section names are included in it.

Input Character Summary

The input file can contain

- Upper- and lowercase letters
- the ten digits
- the 16 punctuation characters:

  `. : ; , ? ! ‘ ’ ( ) [ ] / * @`

- the characters “`| < >`” are only available in mathematics mode. (Bracket them with $ characters.)

Input Characters, continued

- The following seven characters are special and should be quoted with a backslash to use them in running text:

  `#` Is used in defining new commands and environments

  `$` Is used to bracket mathematics mode

  `%` Introduces a comment: *latex* ignores the reset of the line

  `_` Is the subscript command

  `{}` Delimit manditory arguments and scope

- The following characters can usually only be used in a `\verb` command or a `verbatim` environment.

  `~` Marks a space that can't be split over a line break

  `^` Is the superscript command

  `\` Introduces a *latex* command

Exercise #2: Making your own Text

There are a number of *latex* "template" files on /usr/doc to help you get started:

article-template.tex The most useful and commonly used for creating short articles.

slide-root-template.tex Base file for creating a slide show.

slides-template.tex A template for the slides in the show.

figure-template.tex Gives the commands needed to include a PostScript figure or diagram into a *latex* document using the `psfig.tex` macros.

letter-template.tex *Latex* commands to help you get started on a letter to be printed on UWO letterhead. It uses the University date style: 1996 10 19.

Exercise #2, continued

1. Copy the **article-template.tex** file into your disk area and customize it for yourself. Add in your name, department and e-mail address.

2. Copy your customized **article-template.tex** into a file named **exercise2.tex**

3. Edit the file **exercise2.tex** and insert the file **raw.txt** into the buffer just before the **\end{document}** line.

4. Edit the file by adding $latex$ commands to make the output look like the following pages. You will have to add title, name, sectioning and emphasis commands to the text.

5. Send the resulting file to the laser printer for marking!

ex-prelim.ps goes here

Errors

- Tex and $latex$ error messages are notoriously hard to read.

- $Latex$ isn't too bright about files. If it can't find a file it will doggedly keep asking for it. The quick way out is to type `null` which is a file that always exists. Then type an "x" to get out of the program completely.

- Many messages are warnings:

  ```
  Overfull \hbox (2.62048pt too wide) in
    paragraph at lines 278--282
  ```

  This one means that $latex$ couldn't find a good spot to break a line:

  - Bracket the offending paragraph with `\sloppy` and `\fussy` declarations,
  - rewrite the paragraph a bit or
  - if the amount is small, don't worry that your margins aren't perfect ($2pt = \frac{1}{36}th$ of an inch).

Errors, continued

- Fatal errors stop the processing:

```
   LaTeX error.  See LaTeX manual for explanation.
       Type  H <return>  for immediate help.
! Environment verbatum undefined.
\@latexerr ...for immediate help.}\errmessage {#1}
l.120 \begin{verbatum}
?
```

  The message gives you input line number and tries to give you the context. You can try to continue by typing a Return or exit immediately to fix the problem by typing an **x**.

- The most common errors are:
  - A mis-spelt command or environment name
  - Improperly matched '{' and '}' or missing **\end** command.
  - Trying to use one of the ten special characters **# $ % & _ { } ~ ^ ** as an ordinary printing character.
  - A missing command argument.

Lists

- There are 3 standard list making environments:

  itemize bulleted points

  enumerate numbered items

  description user defined item marker

- Each is used in the same way as the example below:

```
\begin{itemize}
  \item This is the first point
  \item And this is the second
\end{itemize}
```

- For a `description`, each `\item` command is followed by a piece of text in square brackets:

```
\item [Habitat] A shoal of bass.
```

- List environments can be nested.

Display Environments

- The `quote` environment is used for a series of short (one line) quotations. Each quote is separated from the next by a blank line.

- The `quotation` environment is used to quote a paragraph or paragraphs of text. Paragraphs are separated by blank lines.

- The `verse` environment is for poetry. You mark the end of line with the \\ command.

- The `verbatim` environment is used to display text exactly as it appears in the input file. A typewriter font is used. No *latex* comands operate (except `\end{verbatim}`).

- The `verb` command is used to include short sequences exactly as typed into the running text. It uses the first character after the command as a delimiter. The text between delimiters must be kept on the same line:

```
\verb+\any old thing~$% is ok+
```

Footnotes

The `\footnote` command generates a footnote at the bottom of the page where it is encountered. Its argument is the text of the footnote. Remember to put the `\footnote` command immediately after the place that you want the footnote marker to occur. Don't add an intervening space.

```
When life\footnote{In its most literal
sense} deals you...
```

When life* deals you...

The document style determines what symbols are used to mark a footnote. Most styles use a small superscripted number.

*In its most literal sense

Exercise #3: More Fixes

1. Copy **exercise2.tex** to **exercise3.tex**.

2. Add *latex* commands to **exercise3.tex** to make it look like the following pages. You will have to use

   - itemize,
   - description,
   - verbatim,
   - quotation and
   - footnote

   to complete this edit.

3. Send the result to the laser printer for marking.

ex-final.ps goes here

Cross References

In larger documents like manuals, cross references can be an important part of the work. $Latex$ provides some commands to make this task easier:

- The **\label{***tag***}** command is used to give a name to a certain location in the document.

- The **\ref{***tag***}** command inserts the section number of the tag into the text.

- The **\pageref{***tag***}** command is used to insert the page number of the tag into the text.

To handle *forward references* you may have to run $latex$ twice to get all the **\ref** and **\pageref** commands right.

Making Tables:
The tabular environment

```
\begin{center}
\begin{tabular}{l|r|r|r}
\hline
Item                                &CCS price      &PST    & Total \$\\
\hline \hline
\multicolumn{4}{c}{Medium Cost Option} \\
\hline
SUN IPX 19'' colour, 16MB       &11,312 &904.96 &12,216.96\\
\hline \hline
Sun 1.3GB disk                  &4,310  &344.80 &4,654.80\\
\hline
CD-ROM player                   &729    &58.32  &787.32 \\
\hline \hline
Total                           &       &       &{\bf 17,659.08}  \\
\hline \hline
\end{tabular}
\end{center}
```

Item	CCS price	PST	Total $
Medium Cost Option			
SUN IPX 19" colour, 16MB	11,312	904.96	12,216.96
Sun 1.3GB disk	4,310	344.80	4,654.80
CD-ROM player	729	58.32	787.32
Total			**17,659.08**

Simple Indexes

- Useful in longer documents like **book** and **report**.
- In the preamble/headers section include the command **\makeindex**.
- Just before the **\end{document}** line add the command **\include{index}**.
- Mark words in the body of your text with the **\index{***word***}** command.
- When compiling, use the following sequence

```
latex  document.tex
makeindex -o index.tex  document.idx
latex  document.tex
```

- Note that *latex* is run twice to get the latest entries incorporated.

Including PostScript files

The following code (from `/usr/doc/figure-template.tex` can be used to include a (well behaved) PostScript file into a *latex* document:

```
\begin{figure}
\centerline{
\psfig{figure=.ps,width=4.1in}}
%              ^name of the PostScript figure
%                       (Should be created without showpage)
\vspace{.25in}
\caption{} \label{}
%       ^title   ^name (\ref{} &/or \pageref{}to refer)
\end{figure}
```

Here's a little PostScript:

Headers, another look

- Useful options
  - 12pt
  - fullpage
  - screen & dvi2tty
- style settings
- Exercise: redo exercise3.tex with fullpage and 12pt type.

Slides

Slitex is a version of *latex* that provides some special features to make the production of overhead slides easier. The slides for this course were produced using *slitex*. Here is a simple procedure for getting started into slide production:

1. Copy the template file `slide-root-template.tex` into your disk area, giving it the name of your slide show. This file is where the global declarations about your slides are stored. If you are making b&w slides, you probably don't need to edit this file.

2. Also copy in the file **`slides-template.tex`** from the **`/usr/doc`** area as **`slides.tex`**. This is where the actual contents of the slides is entered. It is like the *body* part of an ordinary *latex* document.

3. Each slide begins with a **`\begin{slide}{}`** command and ends with a **`\end{slide}`** command. Most *latex* commands can be put in between.

4. Don't forget to use the command **`slitex`** when compiling the slides.

Documentation

This course has just scratched the surface of *latex*. You should now be able do a few things, but for any serious work you should explore the documentation available.

- Cheap (on-line, on julian.uwo.ca)

  man latexintro This command will display lots of practical notes about using *latex* at UWO.

  /usr/doc/latex-local.ps This manual contains things that are not explained in the *Users Guide* since they are likely to be site dependent.

  /usr/doc/essential-latex.ps Send this to a PostScript printer for a good basic introduction too *latex*. It hasn't been customized for UWO yet.

- A few 10s of $

  Latex Users Guide and Reference Manual by *Leslie Lamport.* This book costs a bit at the campus bookstore, but is essential for doing serious work with *latex*.