

Slide 1

Administrivia

- Reminder: Homework 6 due today. Submit by e-mail.
- Reminder: Homework 7 due Monday.
- Midterm grade summaries (to be) e-mailed.

Slide 2

Minute Essay From Last Lecture

- Many/most people usually use Google Docs, though some also use MS Word. Most find the cloud-based nature of Docs appealing but find the application kind of limited. One mentioned that he also found Word somewhat limited. Another said he liked Word's ease of use. (I say "until you try something complicated". Maybe just me.) One mentioned LibreOffice (Linux fan?).
- Many but not all had used \LaTeX . (That surprised me a bit, but maybe it shouldn't — Dr. Fogarty uses it.)
Some mentioned using Web-based interfaces for it. "In the cloud" has its appeal, plus they probably provide more of a GUI-fied interface?
- One person said Word/Docs vs. \LaTeX was kind of like Python vs. Java — one's good for short simple things, the other for complex tasks.

Basic L^AT_EX Features — Review/Recap

Slide 3

- Sectioning commands make (optionally) numbered sections, subsections, etc. `\tableofcontents` generates table of contents.
- Predefined macros provide simple markup, e.g., `\textit{foo}`.
- Environments provide support for lists, etc. One worth noting for CSCI types is `verbatim`, which typesets text as is in typewriter font. Simple way to get code in your program. (A “gotcha” — it doesn’t expand tabs.)
- Math can be displayed inline, as centered and un-numbered equations, or as numbered equations.

Basic L^AT_EX Features, Continued

Slide 4

- Lots of cross-referencing features — declare symbolic label (for section, figure, etc.) with `\label{foo}`, reference with `\ref{foo}`, or `\pageref{foo}` to get page number. (The computer keeps track of numbering! Isn’t this how it ought to work?)
- Can use `\input` to pull in code from another file, like `#include` in C. Very useful for accessing your own macros. (I also use it quite a lot in preparing material for classes. Not long ago I got tired of copying and pasting text from one syllabus to another and refactored(?) to put common parts in a single place.)
`\verbatiminput` typesets included material verbatim. Simple way to include whole program listing.

Bibliographies

Slide 5

- Can sort of do by hand, but better is to use companion tool `BIBTEX`:
- You write a `.bib` file that's a sort of database of references (meaning it can contain more entries than you will use for this document). Predefined types of entries, each with a list of keywords you must/may define (author, title, etc.). Also define for each entry a symbolic name.
- In your document, reference symbolic names with `\cite`. At the point where you want the bibliography, reference the database with `\bibliography`, and select a predefined style with `\bibliographystyle` (or you can write your own!). Everything automatic from there, including formatting. (Isn't this how it should be?)
A "gotcha" — `bibtex` has its own ideas about capitalization. Sometimes this is bad, e.g., when you need all-caps in a title. Enclose in curly braces.

Tables and Graphics

Slide 6

- Support for tables with "tabular" environment. Something(?) of a pain to use but oh well (and would lend itself to being produced programmatically).
- Easy to include graphics from outside file. With traditional toolchain, must be in EPS (Encapsulated PostScript), but they scale nicely if you need that. `pdflatex` accepts input in various popular graphics formats. Not sure about scaling.
- Also there are packages for drawing figures directly.
- (More about various ways to generate figures next time.)

Slide 7

“Floats”

- Figures and tables can “float” (L^AT_EX will put them where they fit). They also can be given labels.
- In my thinking this is how you should always include these elements, with a caption explaining anything that needs explaining (within reason) and a reference in the text such as “Figure `\ref{somefig}` illustrates this point.” This avoids awkward page breaks and looks professional.

Slide 8

User-Defined Markup

- Facilities to define your own “commands” and “environments”. Makes it easy to get consistent formatting; also can provide convenient shorthand ways of doing things.
- To define a simple macro (“command”), `\newcommand`. Examples in sample document.
- To define a custom environment, `\newenvironment`.

Processing L^AT_EX Source, Revisited

- Old way is to use `latex` to generate DVI file (possibly running it several times and also using `bibtex` if needed for bibliography) and then use `dvips` to generate PostScript (and then probably convert to PDF with `ps2pdf`).

Newer way is to use `pdflatex` to go directly to PDF.

(Why run several times? as with C compiler, it's kind of a one-pass process, and it takes more than one pass to resolve crossreferences.)

- Another way — use `lAtekmk`, which runs `latex` (as many times as needed) and `bibtex` if needed. Many useful options, including:

`-outdir` to put all output in (sub)directory.

`-c` to clean up intermediate files.

Normally generates DVI file, but you can go direct to PDF, or use `dvipdf`.

- Remember on our machines to do a

Slide 9

```
module load texlive-latest.
```

Slide 10

Related Tools

- `gnuplot` integrates nicely with \LaTeX .
- Many possible ways to draw figures, but I use `xfig` — old, but nice integration with \LaTeX . (Also what it saves/loads is plain-text files.)
- Tools to convert \LaTeX source to HTML. (I use `latex2html`; there are others.)
- Tools for editing \LaTeX source. Support in both `emacs` and `vim` (`auctex` and `vimlatex` respectively). Also GUI frontends. See “useful links” page.

Slide 11

Minute Essay

- We still have many weeks left. I'm planning to do one more lecture on graphics for \LaTeX a lecture or two on miscellaneous text-mode tools, and maybe one on installing from source. But there's time for other topics. Suggestions?

Slide 12