07-131 Great Practical Ideas in CS

•••

Lecture 01: Course Introduction + LaTeX

Instructors: Eduardo Feo-Flushing & Giselle Reis

Course website: https://web2.qatar.cmu.edu/cs/07131/

Course Introduction

Goals

- To teach you about all the awesome things you can do with your computer.
- To make you super comfortable using Unix systems and the tools you will use in future courses
- ... and in future internships.
- To be a break from your other classes.

CS@CMU is not trivial...

Our job is to introduce you to the tools you need to succeed in CS...

not to make you more stressed!

Class structure

- ~30 minutes of lecture.
- Rest of the time: you work on the labs.

Labs

- Mostly puzzles using unix tools.
- Distributed through `git` (except today).
- Each lab is released at class time.
- Submission is on Autolab.
- Labs are designed to be finished during class time
- ... but you have until the end of the week to submit.
- No late days (but if you absolutely need one, ask us).

Collaboration Policy

You may:

- Consult user manual (`man`) pages for the commands in the task.
- Ask the instructors for help during class time.
- Post a question on Piazza.
- Discuss concepts with your colleagues.
- Google how to use a command or tool.

You may not:

- Ask you colleague how to solve the entire task.
- Copy code from other students or the internet.

Exams

- Two exams:
  - Midterm
  - Comprehensive Final (but not during the finals week!)

Grading

- 80% labs
  - Each lab is composed of several parts.
  - All parts must be completed for full credit.
  - Note that each lab is 8% of your grade!
- 10% midterm
- 10% final

Continuous Feedback Form

- This is our first offering of this course, so any and all feedback are welcome!
- The feedback form will be open throughout the semester at:
  https://forms.gle/JLyQorecHPAtjcr1A
  (also linked on the course website)

Questions?

...

LaTeX

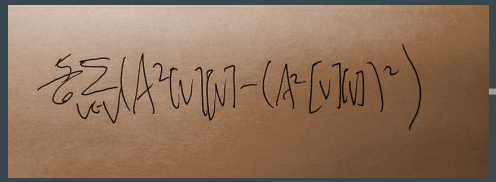
LaTeX

- LaTeX is a high-quality typesetting system.
- Pronunciation: LAH - tekh.
- Standard for communication and publication of scientific documents.
- Free software. \o/
- Why not Word? (or another WYSIWYG* document tool)
  - Typing math in Word is annoying.
  - Easy to get typesetting wrong.
  - Managing references is hard.

* What you see is what you get.

Do I really need to?

- Be nice to your reader (that can be your future self).
- Create clear and readable documents (i.e. pretty is better than ugly).
- Some courses at CMU require your homework to be typeset (for good reason!).
- Many (most?) conferences and journals require LaTeX files to publish papers.
- **Take time to learn once, and save time for the rest of your life** (which should really be the slogan of this course).



$$\frac{5}{6} \sum_{v \in V} (A^2[v][v] - (A^2[v][v])^2)$$

Installation

Local installation:

- You need to install TeX Live by hand*.
- Files are kept locally on your machine.
- Edit the files on your favorite IDE.
- You can work from a plane.
- Compilation is faster.
- Collaboration requires another tool (e.g. git).

Using Overleaf:

- You need to create a user in overleaf.com.
- Files are kept online.
- Edit the files on the website.
- You need to be online to work.
- Compilation is slower.
- Collaboration is easier.

Choose wisely... or try both if you are unsure.

* For Ubuntu-based distributions, the command is `apt-get install texlive-full`

Basics

```
\documentclass[a4paper, 11pt]{article}

\usepackage{...}

\newcommand{...}

\title{...}
\author{...}

\begin{document}
\maketitle

% Comments start with %

\section{...}

\end{document}
```

This part is called the *preamble*.

Indicates which kind of document this is.

Imported packages.

Your defined commands.

Self-explanatory.

Most of your stuff goes inside the *document* block.

Starts a new section.

Typing Text

- Bold text: `\textbf{...}`
- Italic text: `\textit{...}`
- Emphasized text: `\emph{...}` (because italic inside italic is invisible)
- Underlined text: `\underline{...}`
- New paragraph: two line breaks
- New line: `\newline` (I rarely use this...)
- New page: `\newpage`
- Quotation marks: ``` `` ... '' ``` (two backticks, two single quotes)
- Sections: `\section{...}`, `\subsection{...}`, `\subsubsection{...}`, ...
- Escape special characters: `\{` and `\}` for `{` and `}`, etc.

Typing Math

Math has to be written in "math-mode" in LaTeX. That means:

- Between `$...$` for *inline* math:
  - `The fraction $\frac{1}{2}$ is equivalent to 0.5.`
- Between `\[ ... \]` (or `$$...$$`) for a centered equation:
  - ```
    Bhaskara's formula is:
    \[ x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \]
    ```
- Inside an equation environment for numbered equations:
  - ```
    \begin{equation} \label{pyth}
    a^2 + b^2 = c^2
    \end{equation}
    Equation~\ref{pyth} is Pythagoras' theorem.
    ```

Typing Really Long Math

Math expressions can be split into multiple lines and aligned for better visualization.

Use the *align** environment (in the amsmath package), separating columns with &:

```
\begin{align*}
    \Delta &= \sqrt{5^2 - 4 \times 2 \times 2} \\
           &= \sqrt{25 - 16} \\
           &= \sqrt{25 - 16} \\
           &= \sqrt{9} \\
           &= 3
\end{align*}
```

$$
\begin{aligned}
\Delta &= \sqrt{5^2 - 4 \times 2 \times 2} \\
       &= \sqrt{25 - 16} \\
       &= \sqrt{25 - 16} \\
       &= \sqrt{9} \\
       &= 3
\end{aligned}
$$

Typing Code

- The verbatim environment ignores LaTeX commands and respect spaces: `\begin{verbatim} … \end{verbatim}`
- But for pretty code, better use package `listings`

Typing Code: Listings

Configuration in Preamble

```
\usepackage[usenames,dvipsnames]{color}
\usepackage{listings}

% Listings setting for Python code
\usepackage{listings}
\lstset{
  basicstyle=\small\ttfamily,
  breakatwhitespace=true,
  breaklines=true,
  language=Python,
  otherkeywords={self},
  commentstyle=\itshape,
  xleftmargin=10pt,
  rangeprefix=/*\ ,
  rangesuffix=\ */,
  includerangemarker=false,
  keywordstyle=\color{Red},
  identifierstyle=\color{Blue},
  commentstyle=\color{Gray},
  stringstyle=\color{Green}
}
```

Code in Text

```
\begin{lstlisting}
def f(pairs):
      d = dict()
      for pair in pairs:
      p1 = pair[0]
      p2 = pair[1]

      if p1 in d:
            d[p1] += 1
      else:
            d[p1] = 1

      if p2 in d:
            d[p2] += 1
      else:
            d[p2] = 1

      return d
\end{lstlisting}
```

Output

```
def f(pairs):
    d = dict()
    for pair in pairs:
        p1 = pair[0]
        p2 = pair[1]

        if p1 in d:
            d[p1] += 1
        else:
            d[p1] = 1

        if p2 in d:
            d[p2] += 1
        else:
            d[p2] = 1

    return d
```

Creating your own LaTeX macros

- If you find yourself typing a sequence of commands a lot, you can abbreviate them with a macro:
  - Macro definition: `\newcommand{\pyth}[3]{#1^2 + #2^2 = #3^2}`
  - Macro usage: `$\pyth{a}{b}{c}$`
  - Rendering: $a^2 + b^2 = c^2$

- Define macros in the preamble so not to lose them or redefine things.
- Be careful if the command needs to be used inside math mode or not.
- Some people have their own macros file that they use for every LaTeX project. You can do that for your CMU homeworks :)

Final fun fact: LaTeX is a programming language ;)

LaTeX References

Links on the right

- LaTeX Cheat Sheet
- Homework Template
- The Comprehensive LaTeX Symbol List
- LaTeX tutorial by Overleaf

It's smashlab time!

Lab tips

- Read the instructions before starting.
- Each task is a file describing how to modify it.
- Uncomment each task in `main.tex` to work on it.
- There may be hints around to help you.
- You must submit the final pdf and source tex files!
- The `makefile` will only work if you have a local LaTeX installation.