



# INTRODUCTION TO $\text{\LaTeX}$

DR. ISAAC GRIFFITH

IDAHO STATE UNIVERSITY

After today's lecture you will:

- Be introduced to the  $\text{\LaTeX}$  document processing system
- Understand how to create  $\text{\LaTeX}$  documents, including
  - $\text{\LaTeX}$  Preamble (i.e., document class, authorship info, and the inclusion of packages)
  - Structuring  $\text{\LaTeX}$  documents
  - Creation of tables, figures, etc.
  - Basic components of formatting  $\text{\LaTeX}$  documents
  - Mathematical equations
  - Citations and Bibliographies



# § Introduction to L<sup>A</sup>T<sub>E</sub>X

---

CS 1187



- Problems with WYSIWYG editors such as MS Office and LibreOffice
  - Author is distracted from the proper business of composing text, in favor of making typographical choices in relation to which they may have no expertise
    - i.e., fiddling with fonts and margins when they should be concentrating on content
  - Making changes to the whole document i.e., section headings, numbering of figures, references and tables is tedious
  - Does not encourage concern with document structure

# Many Reasons to use $\text{\LaTeX}$

- The typesetting of mathematical formulae is supported in a convenient and powerful always
- Users only need to learn a few easy-to-understand commands that specify the logical structure of a document.
  - They almost never need to tinker with the actual layout of the document
- Even complex structures such as footnotes, references, table of contents, and bibliographies can be generated easily
- $\text{\LaTeX}$  encourages authors to write well-structured texts, because this is how  $\text{\LaTeX}$  works: by specifying structure
- $\text{\TeX}$ , the formatting engine of  $\text{\LaTeX}2\text{e}$ , is highly portable and free
  - Therefore the system runs on almost any hardware platform available

- Rather than attempting to be all things in a single program,  $\text{T}_{\text{E}}\text{X}$  is designed with modularity in mind.
  - Thus  $\text{T}_{\text{E}}\text{X}$  itself provides only fundamental typesetting capabilities and does not incorporate editing, printing, or previewing capabilities
- $\text{\LaTeX}$  (pronounced “Lay-tech” or “Lah-tech”), is a  $\text{T}_{\text{E}}\text{X}$  macro package written by Leslie Lamport (1980’s), that allows authors to use  $\text{T}_{\text{E}}\text{X}$  easily and uses  $\text{T}_{\text{E}}\text{X}$  as its formatting engine
- $\text{\LaTeX}2_{\text{e}}$  is the name used when distinguishing the current version of  $\text{\LaTeX}$  from its predecessor,  $\text{\LaTeX}$  2.09

The typesetting operations of T<sub>E</sub>X are applied on a very low level. They address the tasks of

- Stringing characters together in words and paragraphs
- Positioning symbols properly in mathematical formulas
- Automatically finding good page breaks
- Dealing with footnotes and other floating objects (such as figures and tables)

- By design  $\text{T}_{\text{E}}\text{X}$  is almost always used in conjunction with an auxiliary piece of software called a  $\text{T}_{\text{E}}\text{X}$  format
  - The purpose of this tool is to bridge the gap between the low-level typesetting functions of  $\text{T}_{\text{E}}\text{X}$  and
  - A higher-level interface more suitable for authors
- A  $\text{T}_{\text{E}}\text{X}$  format is made by assembling all of the  $\text{T}_{\text{E}}\text{X}$  macros that define the higher-level interface and precompiling them as a unit in order to reduce start-up time
  - $\text{\LaTeX}$  and  $\text{\LaTeX}2\text{e}$  are  $\text{T}_{\text{E}}\text{X}$  formats





- 1st step is to type the file that  $\text{T}_{\text{E}}\text{X}$  reads, usually called the  $\text{T}_{\text{E}}\text{X}$  file or the input file, using a simple text editor and saved in ASCII mode without any special control characters
- The  $\text{T}_{\text{E}}\text{X}$  program then reads your input file and produces a DVI file (Device Independent)
  - This file is not readable
- The DVI file is then read by another program (called a device driver)
  - Produces the output that is readable by humans
- Why the extra file?
  - The same DVI file can be read by different device drivers to produce output on a printer, screen, or phototypesetter
- Once you have produced a DVI file that gives the right output, you can be assured that you will get the same output on all other devices

- T<sub>E</sub>X Studio – Windows
- Atom – All OS
- Kile – Linux
- Overleaf – Online

For this course we will be using Overleaf

# Document Layout

---

CS 1187

- When  $\text{\LaTeX}$ 2e processes an input file, it expects it to follow a certain structure.
  - Thus every input file just start with the command:  
`\documentclass{...}`
  - This specifies what sort of document you intend to write.
- After that, add commands to influence the style of the whole document, or load packages that add new features to the  $\text{\LaTeX}$  system
  - To load such a package you use the command  
`\usepackage{...}`

- When all the setup work is done, you start the body of the text with the command  
`\begin{document}`
- Now you enter the text mixed with some useful  $\text{\LaTeX}$  commands. You end the document with the following command  
`\end{document}`
- Anything following that command will be ignored

- The first thing  $\text{\LaTeX}$  needs to know when processing an input file is the type of document the author wants to create.
  - This is specified with the `\documentclass` command
- `\documentclass[options]{class}`
- Here `class` specifies the type of document to be created
- The `options` parameter customizes the behavior of the document class
  - options need to be separated by commas
- **Example:**

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

class	description
article	for articles in scientific journals, presentations short reports, program documentation, invitations
proc	a class for proceedings based on the article class
minimal	is as small as it can get. It only sets a page size and a base font. It is manly used for debugging purposes
report	for longer reports containing several chapters, small books, Master's and PhD theses
book	for real books
beamer	the $\LaTeX$ version of Powerpoint

# Document Class Options

option	description
10pt/11pt/12pt	Font size
letterpaper/a4paper	Paper size
twocolumn	Use two columns
twoside	Set margins for two-sided
draft	Double-space lines



- There are some functionalities that you won't be able to do with basic  $\LaTeX$
- Including graphics, color, etc. requires enhancing capabilities by including packages using the `\usepackage[options]{package}` command in the preamble of your document
- Comprehensive  $\TeX$  Archive Network (CTAN) is the central source for all packages
- <http://www.ctan.org>

- Common Packages to include:
  - `graphicx` - for graphics
  - `algorithm` and `algpseudocode` - for algorithms
  - `listings` - for code listings
  - `multirow` and `multicol` - for multiple row or multiple column cells in tables
  - `amssymb` and `amsmath` - for mathematical symbols
  - `xcolor` with option `svgnames` - for defining colors
  - `url` - to insert a url with the `\url{...}` command

Before the `\begin{document}`

- Author Information is added with the `\author{...}` directive
- Title of document is added with the `\title{...}` directive
- Date of a document is set with the `\date{...}` directive

After `\begin{document}`

- Making the title
  - Typically right after we start the body of the document with `\begin{document}`, we issue the `\maketitle` declaration
  - `\maketitle` is a macro which will take all of the information necessary and format the title components

- $\text{\LaTeX}$  commands or directives are case sensitive
- They start with a backslash `\` and then have a name consisting of letters only
- Command names are terminated by a space
- Some commands require a parameter, which has to be given between curly braces `{}` after the command name
- Some commands take optional parameters, which are inserted after the command name in square brackets `[]`
- The `%` character is used to comment out a line

- Environments define special sections of text to be formatted in a particular way.
- All environments use the following structure:
- Additionally, environments can be nested within each other as long as the correct nesting order is maintained

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

- Where `environment` is the name of the environment

```
\begin{aaa}  
text  
  \begin{bbb}  
    text  
  \end{bbb}  
text  
\end{aaa}
```

# § Document Body

---

CS 1187

- A document can be decomposed into several components:
  - `\part{title}` - sections of a book, the highest subdivision
  - `\chapter{title}` - chapters of a book
  - `\section{title}` - section of a document
  - `\subsection{title}` - subsection of a section
  - `\subsubsection{title}` - a sub-subsection of a subsection
  - `\paragraph{title}` - a titled paragraph in a section/subsection/subsubsection
  - `\subparagraph{title}` - a titled subparagraph
- Note that each of these components will be numbered. You can opt to have unnumbered items if you follow the name with a star
  - Example: `\section*{title}` will create an unnumbered titled section.

## Input:

It does not matter whether you enter one or several spaces after a word.

An empty line starts a new paragraph.

## Output:

It does not matter whether you enter one or several spaces after a word.

An empty line starts a new paragraph.

- Whitespace characters, such as blank or tab, are treated uniformly as a space by  $\text{\LaTeX}$
- Several consecutive whitespace characters are treated as one space
- Whitespace at the start of a line is generally ignored, and a single line break is treated as whitespace
- An empty line between two lines of text defines the end of a paragraph
- Several empty lines are treated the same as one empty line



- The following symbols are reserved characters that either have a special meaning in  $\text{\LaTeX}$  or are not available in all the fonts.

- If you enter them directly in your text, they will normally not print, but rather coerce  $\text{\LaTeX}$  to do things you did not intend.

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

- Insert a backslash before these characters to get the desired result

\# \\$ \% \^ \{ \} \& \\_ \sim \textbackslash \textbar

```
\flushleft
\begin{enumerate}
\item You can nest the list environments to
your taste:
\begin{itemize}
\item But it might start to look silly.
\item[-] With a dash.
\end{itemize}
\item Therefore remember:
\begin{description}
\item[Stupid] things will not become
smart because they are in a list.
\item[Smart] things, though, can be
presented beautifully in a list.
\end{description}
\end{enumerate}
```

1. You can nest the list environments to your taste:

- But it might start to look silly.
- With a dash.

2. Therefore remember:

**Stupid** things will not become smart because they are in a list.

**Smart** things, though, can be presented beautifully in a list.

- Often in LaTeX we need to add objects, which float within the document as the text changes, these include
  - Tables
  - Figures
  - Equations

## Tables

```
\begin{table}[t]
\centering
\caption{Caption}
\label{tab:label}
\begin{tabular}
...
\end{tabular}
\end{table}
```

## Figures

```
\begin{figure}[b]
\centering
\includegraphics[options]{file}
\caption{caption}
\label{fig:label}
\end{figure}
```

## Equations

```
\begin{equation}[H]
\centering
...
\caption{Caption}
\label{eqn:label}
\end{equation}
```

- Font Families
  - Roman: `\textrm{text}` or `\rmfamily text`
  - Sans serif: `\textsf{text}` or `\sffamily text`
  - Typewriter: `\texttt{text}` or `\ttfamily text`
- Font Styles
  - Medium: `\textmd{text}` or `\mdseries text`
  - Bold: `\textbf{text}` or `\bfseries text`
  - Italics: `\textit{text}` or `\itseries text`
  - Small Caps: `\textsc{text}` or `\scshape text`
  - Emphasized: `\emph{text}` or `\em text`
  - Slanted: `\textsl{text}` or `\slshape text`
  - Upright: `\textup{text}` or `\upshape text`
  - Underline: `\underline{text}`
- Note that the command forms handle spacing better than the declaration forms

- Although we can directly set the font size, it is typically best to utilize the built-in declarations
  - `\tiny`
  - `\scriptsize`
  - `\footnotesize`
  - `\small`
  - `\normalsize`
  - `\large`
  - `\Large`
  - `\LARGE`
  - `\huge`
  - `\Huge`
- As these are declarations, they will apply to everything following them, to reduce this we can use braces surrounding the section we want to change the font of. `{\small ...}`

- Default justification is fully-justified
- Center - Text can be center justified as follows
  - Environment: `\begin{center} ... \end{center}`
  - Declaration: `\centering`
- Left - Text can be left justified as follows
  - Environment: `\begin{flushleft} ... \end{flushleft}`
  - Declaration: `\raggedleft`
- Right - Text can be right justified as follows:
  - Environment: `\begin{flushright} ... \end{flushright}`
  - Declaration: `\raggedright`
- I suggest that you use the package `ragged2e` which provides some useful refinements to the above

- Quotes: ``text'`` this produces “text”,
  - Starts with two backticks, ends with two single quote marks
- Braces: curly braces denote a block in  $\text{T}_{\text{E}}\text{X}$ , thus you need to escape them as follows: `\{ \}`
  - Parentheses and square brackets can be use as is
  - angle brackets (less than or greater than) need a special command:
    - `\textless`
    - `\textgreater`

- Books are often typeset with each line having the same length.
  - $\LaTeX$  inserts the necessary line breaks and spaces between words by optimizing the contents of a whole paragraph
- If necessary, it also hyphenates words that would not fit comfortably on a line.
  - How the paragraphs are typeset depends on the document class.
  - Normally the first line of a paragraph is indented, and there is no additional space between two paragraphs
    - To prevent indent use `\noindent`
- In special cases it might be necessary to order  $\LaTeX$  to break a line
  - `\` or `\newline` starts a new line without starting a new paragraph
  - `\`\* additionally prohibits a page break after a forced line break
  - `\newpage` starts a new page



- To create a table use the `\begin{tabular}{cols} content \end{tabular}` environment
- The `cols` option has the following basic specification
  - for each column we want, we add a “l”, “c”, or “r”, for left-, center-, or right-aligned column
  - if we want vertical lines before or after a column, we add a “|” before or after the columns letter in the options
- Rows are then a line of data where each cell of the row is separated by an “&” and the line is ended with a “\”
  - if we wish to have a line between (or above/below a row) we need to use the `\hline` command

# Table Example



```
\begin{table}[!th]
\begin{tabular}{|l|c|r|}
\hline
first & row & data \\
\hline
second & row & data \\
\hline
\end{tabular}
\caption{This is the caption}
\label{tab:table}
\end{table}
```

first	row	data
second	row	data

This is the caption

- Note that we reference something that has a `\label{something}` using the `\ref{something}` command.

- Equations
  - For inline math: `\{ ... \}` or `$ ... $`
  - For numbered eqns: `\[ ... \]` or `\begin{equation}...\end{equation}`
- You can write any basic equation:
  - `$4x + 5 = 17$`
  - Superscript: `$x^{\{y\}}$` yields  $x^y$
  - Subscript: `$x_{\{y\}}$` yields  $x_y$
  - Fractions: `$\frac{\{x\}}{\{y\}}$` yields  $\frac{x}{y}$
  - Roots: `$\sqrt[n]{\{x\}}$` yields  $\sqrt[n]{x}$
- Sums:
  - `$\sum_{\{k=1\}}^{\{n\}} x$` yields  $\sum_{k=1}^n x$
  - `$\overset{\{n\}}{\underset{\{k=1\}}{\sum}} x$` yields  $\sum_{k=1}^n x$
- Products:
  - `$\prod_{\{k=1\}}^{\{n\}} x$` yields  $\prod_{k=1}^n x$
  - `$\overset{\{n\}}{\underset{\{k=1\}}{\prod}} x$` yields  $\prod_{k=1}^n x$

- We construct bibliographies using the BibTeX format.
- Bibliographies are stored in files ending in “\*.bib” extension
- We can then include these in our documents (before the `\end{document}`) as follows:
  - We first set the bibliography style using: `\bibliographstyle{style}`
  - We then set the bibliography file (using the filename without the .bib extension):  
`\bibliograph{filename}`
- The basic bibliography styles include:
  - Standard: `abbrv`, `alpha`, `plain`
  - APA: `apa`
  - Unsorted: `unsorted`
  - Alpha + Abstract: `abstract`

- BibTeX is a tool for processing citations with  $\text{\LaTeX}$ 
  - Overleaf handles processing this for you.
- The particular format of BibTeX files is a collection of BibTeX entries
  - Each entry has one of the following types:

Entry Type	Description	Entry Type	Description
@article	Journal or magazine article	@book	Book with publisher
@booklet	Book without publisher	@conference	Article in conference proceedings
@inbook	A part of a book and/or range of pages	@incollection	A part of book with its own title

Entry Type	Description	Entry Type	Description
@misc	If nothing else fits	@phdthesis	PhD Thesis
@proceedings	Proceedings of a conference	@techreport	Tech report, usually numbered in series
@unpublished	Unpublished manuscript		

- The general syntax is as follows: `@type{key, ...}`
  - Where `type` is an entry type from the table, and `key` is a unique alphanumeric key within the file

- Each BibTeX entry is then composed of several fields, not all of which must be filled in.

Field	Description	Field	Description
address	Address of publisher	author	Names of authors
booktitle	Title of book when part of it is cited	chapter	Chapter or section number
edition	Edition of a book	editor	Name of editors
institution	Sponsoring institution of tech. report.	journal	Journal name
month	Month published (3-letter abbrev.)	note	Any additional information

- Each BibTeX entry is then composed of several fields, not all of which must be filled in.

Field	Description	Field	Description
number	Number of journal or magazine	organization	Organization that sponsors a conference
pages	Page range (2,6,9--12)	publisher	Publisher's name
school	Name of school (for thesis)	series	Name of series of books
title	Title of work	type	Type of tech. report., e.g. "Research Note"
volume	Volume of a journal or book	year	Year of publication



```
@Article{WC:1953,  
  author   = {James Watson and Frances Crick},  
  title    = {A structure for Dexoxyribose Nucleic Acid},  
  journal  = N,  
  volume   = {171}  
  pages    = {737},  
  year     = 1953  
}
```

- Typically, I would suggest using a tool to manage your references, and then use it to generate the BibTeX file for you
- Such tools include
  - Zotero
  - Mendely
  - EndNote
  - RefWorks
- Each has its benefits and drawbacks and each of these integrates with Overleaf, to automate BibTeX generation

- When you need to cite a reference item in your document, you need two things:
  - A key, as defined in the BibTeX file
  - The right command
- The commands that can be used for citing items are:
  - `\cite{key}` - cites the full author list and year i.e., “(Watson and Crick 1953)”.
  - `\citeA{key}` - Full author list i.e., “(Watson and Crick)”
  - `\citeN{key}` - Full author list and year i.e., “Watson and Crick (1953)”
  - `\citeyear{key}` - cites the year only i.e., “(1953)”
- Note that different reference styles will cite differently, additionally they may use different commands.
  - Be sure to read the documentation on how to use the particular class or template you have selected

# Miscellaneous Commands



- `\today` - Generates the current date
- `$_sim$` - Generates a  $\sim$  in math mode
- `\hspace{1}` - Generates a Horizontal space of length  $l$  (e.g.,  $l = 1.5em$ )
- `\vspace{1}` - Generates a Vertical space of length  $l$
- `\rule{w}{h}` - Creates a line of width  $w$  and height  $h$

# Typical CS Document Classes



- In Computer Science we tend to use several specific formats:
  - IEEE Transactions
    - Document Class: `IEEEtran` with options (`journal`, `compsoc`, `conference`)
    - Bibliography Style: `IEEEtran`
  - ACM
    - Document Classes: `acmart` has multiple options
    - Bibliography Style: `ACM-Reference-Format`
- These are available on [Overleaf.com](https://www.overleaf.com) to be used as templates
- For this course, our homework will be using the document class `homework` from the template `Gij's Homework Template` found on [Overleaf.com](https://www.overleaf.com)

# For Next Time



- Review LaTeX Tutorial
- Review this Lecture
- Come To Lecture
- Read DMA Chapter 3





# Are there any questions?