Document Structure

Lecture 3

David Kraemer & Caleb Leedy

October 18, 2014

Outline

1. **Overview**
    - Last time
    - This time
2. **Introductory formatting**
    - Preamble for today
    - Creating a title
    - Titles in various document classes
    - Sections and paragraphs
3. **In-depth formatting**
    - Setting margins
    - Paragraph and line spacing
    - Paragraph justification environments
    - List environments

Last time

- We covered math mode and display mode
- We learned about how to use environments
- We also started looking at how commands in LaTeX work
- We were able to define our own command

This time

- Tonight we're going to talk about how to structure documents.
- We will experiment with different document types and how they affect the overall typesetting.
- We will also explore some online templates for customizing the look and feel of the documents you create.

Preamble for today

Let's create a new LaTeX document with the following code:

Document skeleton

```
\documentclass[10pt]{article}

\begin{document}



\end{document}
```

Creating a title

- LaTeX gives you an automatic way to produce titles.
- You can do so by making use of preamble commands.
- After `\documentclass`, insert

Title preamble

```latex
\title{Document Title}
\author{Author Name}
\date{\today} % or whatever you want
```

- You can use % to add comments to your code.
- To generate the title, simply add `\maketitle` inside the `document` environment.

Titles in various document classes

- The way the title is compiled depends on the document class.
- There are a few standard document classes that can be used:
  - `book` for textbooks
  - `report` for technical reports and theses
  - `proc` for conference proceedings
  - `letter` for letters
  - `beamer` for slide presentations
- You should try compiling with different document classes to see how they affect the title.

Sections and paragraphs

- Inside the `document` class, LaTeX gives you the ability to produce section headings easily.
- Each section type has its own indentation level.

Command	**Level**	**Function**
`\chapter{}`	0	book and `report` only
`\section{}`	1	not in `letter`
`\subsection{}`	2	not in `letter`
`\subsubsection{}`	3	not in `letter`
`\paragraph{}`	4	not in `letter`
`\subparagraph{}`	5	not in `letter`

- To remove the numbering, put a $*$ before the brace.

Setting margins

- The geometry package is a great way to adjust margins globally.
- We can modify the package initialization in the preamble to say

Simple margins

```
\usepackage[margin=1in]{geometry}
```

- Or, you can make specific adjustments to each margin by

Custom margins

```
\usepackage{geometry}
\geometry{ left=1in, right=1in,
           top=1in,  bottom=1in}
```

Paragraph and line spacing

There are three built-in spacing values that we can customize in the preamble.

- `\parindent`, which sets the indentation at the beginning of each paragraph.
- `\parskip`, which sets the spacing between paragraphs.
- `\linespread{}`, which sets the line-spacing.

For example, a no-indent, single-spaced preamble would look like

Preamble customization

```
\setlength{\parindent}{0in}
\setlength{\parskip}{1.0in}
\linespread{1.0}
```

- `linespread` is a bit weird, so `1.3` denotes $1\frac{1}{2}$ spacing and `1.6` denotes double spacing.

Paragraph justification environments

- The default paragraph alignment is full-justification.
- A simple way to change alignment is to use one of three environments:
  - `flushleft`, which gives left-justification
  - `flushright`, which gives right-justification
  - `center`, which gives center-justification

This is left-justified text.

This is right-justified text.

This is center-justified text.

List environments

- There are three list environments:
  1. `enumerate`, which creates numbered lists
  2. `itemize`, which creates bullet-point lists
  3. `description`, which allows you to specify the items in the list

Each environment works like:

List structure

```
\begin{enumerate}
\item Item 1
\item Item 2
\end{enumerate}
```

- `description` requires `item[Description]` for every item.
- Each of the list environments can be nested to create sub-lists.