

L^AT_EX FOR UNDERGRADUATES

PREAMBLE

Lecture Notes

Tom Schenk Jr.

Version of November 17, 2005

1 Motivation

Every L^AT_EX document is divided into two parts: the preamble and body. This lecture covers the preamble. The preamble contains all of the “meta” information for the rest of the document (i.e. font size, layout style, margins). The information typed in the preamble is not seen by the reader, instead, it sets up how the document looks. The body, on the other hand, is what appears in the document—this concept is covered in the next lecture.

2 Preamble and Body

The border between the preamble and body is the `\begin{document}` command. As it implies, this command tells L^AT_EX when to start the actual document. The information that appears after this command is considered the body. Information that appears before this command is the preamble.

2.1 documentclass Command

The first command for every document should be the `\documentclass` command. The syntax of the command is as follows:

$$\text{\texttt{\textbackslash documentclass[option]\{class\}}}$$

class tells the L^AT_EX what type of document you want. Typically, we will use the **article** class. Other classes include **book**, **report**, **slides**, and **letter**. Each one of these classes have a unique feature. For instance, when you use the **book** class, L^AT_EX will generate a title page. Likewise, both **book** and **report** allow the author to use chapters, whereas **article** (the most common class) does not. Generally speaking, there are many classes available for L^AT_EX. Anyone can create “.sty” files, which are class files. Many graduate schools have created their own .sty files to meet their requirements for Ph.D. thesis formats. Although too

advanced for this tutorial, as you become more advanced, you may want to look into writing your own .sty files that comply with common class requirements. In the meantime, we will learn other methods of changing a document layout.

An author also has a handful of options for each class. These options allow the author to manipulate the layout of the document. A common option is changing the font size. Assume that we want to type a common document (`article` class) with a font size of 12 points. The `documentclass` command would be the following:

```
\documentclass[12pt]{article}
```

Options are, well, optional. L^AT_EX uses its defaults when the author does not define options. The appendix contains a table listing common options and their actions. Likewise, since options are not required, the author does not have to include the option argument. However, she *must* include the class argument. Below is an example of acceptable and unacceptable arguments.

<i>Acceptable</i>	<code>\documentclass[legalpaper]{book}</code>
<i>Acceptable</i>	<code>\documentclass{book}</code>
<i>Not Acceptable</i>	<code>\documentclass[12pt]</code>

Additionally, the author may list several options within a single line using the following:

```
\documentclass[12pt,legalpaper]{article}
```

In the above example, L^AT_EX will use a 12 point font and a legal paper layout.

As I noted earlier, the author *always* needs to include the `documentclass` command. It does not need to be the first command, but it's practical convention to do so.

2.2 Packages

Packages are a powerful way of including more features into a document. Packages are not a unique concept, the same concept is used by Mathematica, GNU R, SAS, and other mathematical/statistical programs. You typically have to download packages and install them on your computer. I will point to some useful resources for packages later in these notes.

Typically, package commands are included right after `documentclass`. Below is an example of a basic package command:

```
\usepackage[options]{package}
```

Assume that we want to enable ourselves to make an index table at the end of our document. To do this, we must load a package with the following command:

```
\usepackage{makeidx}
```

Where `makeidx` is the package name, this will *enable* us to make an index. It actually takes a few other commands to actually generate the index (Lecture 3.2.3). That is, if you're unsure if you'll actually use the package, it doesn't hurt to include it anyways.

You may have noticed that packages also have *options*. \LaTeX will use defaults if the author does not invoke any options, but sometimes they are useful. Perhaps the easiest example is using another powerful feature of \LaTeX — multilingual capabilities. To let \LaTeX operate in another language, the author can use the *Babel* package.¹ The author then defines which language she would like to use with an option operator. Let's assume that we would like to use German or French in our document (both philosophy and mathematics majors may encounter this in graduate study), below is an example of the `usepackage` command:

```
\usepackage[german,french]{babel}
```

`german` and `french` are defined in the options field.

You may have noticed that we can separate multiple options with a comma. \LaTeX allows the author to list multiple options for a single package with commas. We may list multiple packages with a comma, as long as no options are used. Below are some acceptable uses of listing multiple packages:

<i>Acceptable</i>	<code>\usepackage{makeidx}</code>
	<code>\usepackage{pstricks}</code>
	<code>\usepackage[spanish]{babel}</code>
<i>Acceptable</i>	<code>\usepackage{makeidx,pstricks}</code>
	<code>\usepackage[spanish]{babel}</code>
<i>Not Acceptable</i>	<code>\usepackage[spanish]{makeidx,pstricks,babel}</code>

There are a copious amount of packages available. If you are using MiKTeX, you can obtain packages using the Package Manager which was installed on your computer. Since version 2.4, MiKTeX will automatically download packages from the web. If MiKTeX cannot find the package, or if you are not using MiKTeX, you may obtain packages from a variety of locations. ctan.org provides many packages as well as documentation.

Unlike `documentclass`, packages are not mandatory. They are required to do some functions, but the basic functions do not require packages. For instance, students should not need to use packages to typeset mathematical formulas.

3 Conclusion

We now know that \LaTeX is broken down into two parts: preamble and body. The preamble contains all of the information needed before we can type our document (e.g. document style, options, etc.).

¹I have given no attention to the `babel` package in this tutorial. Those interested in typesetting in other languages should read chapter 10 (pp. 301–374) in *Digital Typography Using \LaTeX* available on Drake's ebrary.

We've started to explore the two most important elements in the preamble: `documentclass` and `usepackage`. `documentclass` is absolutely necessary for our document. It lets \LaTeX know what type of document we are writing. Packages are not required, but enable us to use more advanced features. After awhile, you will realize that some of the most powerful applications of \LaTeX are enabled by packages. Although packages are not "standard" features, they are often written by the same programmers that developed \LaTeX .

So far, we can construct the two first steps for a document. Below is an example of the first two lines of a \LaTeX document:

```
\documentclass[options]{class}  
\usepackage[options]{package}
```