

Text II: Font Effects and Document Structure

Introduction

Today we are going to learn more about type setting in paragraph mode. Recall that this is our usual text mode. Specifically we'll be learning how L^AT_EX helps with writing unusual, foreign letters, type-sizes, punctuation, footnotes and some new environments like `tabular`, `itemize` and `newtheorem`. We also go over sectioning commands that help to structure your article.

If your papers contain little or no math, but lots of words, pay attention today!

Accents and Non-English Letters

You will recall that the lovely, but blasé, Ms Tuğba Güvenç has accents on her name. These were produced with some of the following commands. Whatever the accent is to operate on goes in the braces. We give some common ones here:

effect	input	name
á	<code>\' {a}</code>	acute
ḃ	<code>\u {b}</code>	breve
ĉ	<code>\r {c}</code>	circle
Ď	<code>\^ {d}</code>	circumflex
ë	<code>\" {e}</code>	umlaut
ğ	<code>\' {g}</code>	grave
h	<code>\~ {h}</code>	tilde
ç	<code>\c {c}</code>	cedilla

- Spanish speakers/writers amongst you may like to note that you can produce a `ı` and `İ` by typing `!'` and `?'` respectively

Changing the type effect

There are ways to change the actual font that L^AT_EX uses, but we aren't going to cover that in this course. Nonetheless, you should be aware of the following effects, which will probably prove useful:

effect	input
<i>italics</i>	<code>\textit{}</code>
bold	<code>\textbf{}</code>
upright	<code>\upshape{}</code>
SMALL CAPITALS	<code>\scshape{}</code>

- note that the `\upright` command sets text that is default italicized upright (i.e. not in the current paragraph mode). We'll discuss when we will use such a command in due course.
- another command you may find useful is `\ldots` which gives you three dots like this ...

Changing the Size of Text

There are ten declarations in L^AT_EX for altering the type set. Here they are:

output	input
Deniz Aksoy	<code>\tiny{Deniz Aksoy}</code>
Tugba Guvenc	<code>\scriptsize{Tugba Guvenc}</code>
Tanya Bagashka	<code>\footnotesize{Tanya Bagashka}</code>
Gretchen Helmke	<code>\small{Gretchen Helmke}</code>
Arthur Spirling	<code>\normalsize{Arthur Spirling}</code>
Mark Kayser	<code>\large{Mark Kayser}</code>
Mark Fey	<code>\Large{Mark Fey}</code>
Antoine Yoshinaka	<code>\LARGE{Antoine Yoshinaka}</code>
Jim Johnson	<code>\huge{Jim Johnson}</code>
Hein Goemans	<code>\Huge{Hein Goemans}</code>

You can use them pretty much anywhere, so `\Large{Elephants} \normalsize{and} \scriptsize{mice}` will produce **Elephants** and mice.

Punctuation

Dashes

There are three types of dashes in L^AT_EX :

- the **triple dash**, which is three consecutive hyphens, like this:

Tuesdays---because I have philosophy---are
a pure delight

Which gives you: Tuesdays—because I have philosophy—are a pure delight. Use this system when you want the reader to read the text between the dashes as if it was in brackets

- the **double dash** is made with two consecutive hyphens. Use it when you want to join names that are linked, or page numbers:

the Gibbard--Satterwaite theorem is on pages 56--67

which yields: the Gibbard-Satterwaite theorem is on pages 56-67

- the **hyphen** is one dash and is used on hyphenated words like child-*bride* and frat-*boy*

Quotation Marks

We have already discussed how to use ‘ to get ‘ and ’ to get ’. The same thing works for speech marks: “ gives “ and ” gives ”. OK?

Footnotes

Where would you be without footnotes¹? You produce footnote by using the `\footnote{}` command, where any text you want goes in the braces. Footnotes are started at 1 in the `article` environment and then automatically numbered: so you don't need to worry about chopping and changing them around. As an example of a footnote, consider the following: there is only one word in English that requires the addition of a ‘c’ to make it a plural.² Which was produced using the following command:

```
there is only one word in English that requires
the addition of a ‘c’ to make it a plural.\footnote{That word is
dice.}
```

Itemizing

For presentations, it is often helpful to use bullet points. To do so, you need to run the `itemize` environment. It has the same format as all the others, but when you want a bullet, you type `\item`. Here is an example:

The mid-Atlantic states consist of:

- New York
- Pennsylvania

¹not all the way down here, that's for sure

²That word is dice.

- New Jersey

which was produced using the following code (note the spacing is automatic):

```
\begin{itemize}
\item New York %
\item Pennsylvania%
\item New Jersey%
\end{itemize}
```

Note the use of % at the end of the lines. Depending on the text wrapping mode on your WinEdt, if you don't use these, when you write `\item`, it may not move you onto the next line. This doesn't matter to the output, but it can be annoying when working on your input file.

You can 'nest' items in a list of itemized bullets, by beginning and ending the itemize environment again within the list. For example,

```
\begin{itemize}
\item New York
\begin{itemize}
\item capital is Albany
\end{itemize} %
\item Pennsylvania%
\item New Jersey%
\end{itemize}
```

gives:

- New York
 - capital is Albany
- Pennsylvania
- New Jersey

You can nest up to four times for the same bullet.

Numbered items

Numbered items are written in the `enumerate` environment. It works almost exactly as the itemize environment works, but with the modification that `\item` gives you a number, and nested items give you letters. Thus:

1. Eat your calories, don't drink them
 - (a) soda adds sugar to your diet
 - (b) stick to water and tea

2. Exercise is the perfect partner
 - (a) get into a regular gym routine
 - (b) don't overdo it!

Was produced using

```

\begin{enumerate}
\item Eat your calories, don't drink them
\begin{enumerate}
\item soda adds sugar to your diet %
\item stick to water and tea
\end{enumerate}
\item Exercise is the perfect partner
\begin{enumerate}
\item get into a regular gym routine%
\item don't overdo it!
\end{enumerate}
\end{enumerate}

```

Theorems and Definitions

We sometimes want to formally present definitions, theorems, results and so on. You will need the `\newtheorem` command to do this. You first have to define some things so L^AT_EX knows how you want the output to look. For example, `\newtheorem{definition}{Definition}` tells L^AT_EX that there is now an environment called `definition` and that it takes the title **Definition** when it occurs. It will also be numbered in order. You can put that declaration anywhere in the input file, but it is probably best to do so at the top of the file. Now, when you type:

```

\begin{definition}
A \textbf{homogeneous equation system} is of the form
 $\boldsymbol{Ax=0}$ 
\end{definition}

```

you will get:

Definition 1 *A homogeneous equation system is of the form $Ax = 0$*

Note that we told L^AT_EX to write particular parts of the definition in bold. You can prevent L^AT_EX using italics with the command `\upshape` around the relevant parts. There are ways to write postulates, assumptions and so on in different fonts and styles, but you should look in a L^AT_EX guide book for that.

Sections, subsections, subsubsections

The sectioning commands available in L^AT_EX 's `article` environment are as follows: `\part`, `\section`, `\subsection`, `\subsubsection`, `\paragraph` and `\subparagraph`. These commands are in size order, and in the current form, will be numbered (but starting at different numbers).

As an example of how to use these commands, you write the section commands as `\section{heading title}`. So you can write `\section{Discussion}` to obtain

1 Discussion

All the other commands have exactly the same format, so write

```
\subsubsection{Relevant Literature}
```

to obtain

1.0.1 Relevant Literature

Note the numbering here. L^AT_EX knows to count from when you wrote the `\section` command, and the 1.0.1 refers to the first section, the zeroth subsection and the first subsubsection.

The effect of the `\section*`-type command is that the heading produced on the output file has no number. Generally, as you will remember from the math environments like `eqnarray`, a `*` turns off the numbering. L^AT_EX will start numbering from the first time you don't use a `*`. I have used all my sectioning commands with `*` in these notes. As an example,

```
\section*{Results}
```

gives

Results

and

```
\subsubsection*{Intervening Variables}
```

gives

Intervening Variables