Introduction to Scientific Typesetting
Lesson 11: Foreign Languages, Columns, and Section Titles

Ryan Higginbottom

January 19, 2012

An Overview

The `babel` Package

Columns

Section Titles

The babel Package

Without babel

There is a way to make accented characters without loading an extra package.

á	\'{a}		à	\`{a}
â	\^{a}		ä	\"{a}
ã	\~{a}		ç	\c{c}
ō	\={o}		o̲	\b{o}
ȯ	\.{o}		ụ	\d{u}
å	\r{a}		ŏ	\u{o}
š	\v{s}			

4

A number of things.

- All of those curly braces are **a pain**.

- LaTeX is very US-centric in terms of hyphenation patterns, numbering styles and typesetting conventions. When typing in a different language, it's not just about putting accents on letters!

- If you're using different languages, you don't want to have "Contents" or "References" show up for your Table of Contents or Bibliography—you want the appropriate word for your language.

With `babel`

The `babel` package is very easy to load:
`\usepackage[options]{babel}`

You can load several languages; your *default* one is the last one loaded.

`\usepackage[spanish,english]{babel}`
This would start me off in standard English. But, I could switch over to Spanish conventions any time I wanted.

In order to use a language in your document, it must be declared in the options of the `\usepackage{babel}` command!

There are two ways.

■   `\selectlanguage{language}`
    If I want to switch to Spanish in the middle of my document, I
    just include the command `\selectlanguage{spanish}`.

■   `\begin{otherlanguage}{language}`
    The alternative to the above:

    `\begin{otherlanguage}{spanish}`
    `...`
    `\end{otherlanguage}`

Using `babel` can support non-Latin alphabets too, like Greek,
Russian and Hebrew. (You'll have to use something different
to type Chinese or other Asian alphabets.)

If your document will be entirely in Spanish, I would suggest this in the preamble:

`\usepackage[spanish,activeacute]{babel}`

This provides a lot of shortcuts for accents. (See PDF document online.)

Other differences?

- Numbering and symbols for `itemize` and `enumerate`.
- Operators with accents: instead of $\lim_{x\to 1} f(x)$ you get $\lim_{x\to 1} f(x)$
- Different words for *Contents*, *References*, *Figure*, etc.
- `\today` gives the date like "18 de enero de 2012"
- Decimals and commas in large numbers handled differently (needs to be in math mode).

Example

Open the first example file (`.tex`), build and view.

If your document will entirely be in French, I would advise this in the preamble: `\usepackage[french]{babel}`

This provides some shortcuts for punctuation. (See PDF document online.)

Other differences?

- Symbols and spacing for `itemize` environment.
- First paragraph of a section indented.
- Different words for *Contents*, *References*, *Figure*, etc.
- `\today` gives the date like "18 janvier 2012"

Typing in German

If your document will entirely be in German, I would advise this in the preamble: `\usepackage[german]{babel}`

This provides some shortcuts for punctuation and accents. (See PDF document online.)

Other differences?

■ Different words for *Contents*, *References*, *Figure*, etc.

■ `\today` gives the date like "18. Januar 2012"

A Warning About Changing Languages

If you start off with a language in your `babel` options in the preamble and change your mind, you'll need to delete the `.aux` file (and the `.toc` file if you're making a TOC) before building again, or you'll get an error.

Example: Suppose I start with

`\usepackage[spanish,french,english]{babel}`

in `file.tex` then decide I won't be using `french` after all. If I've built `file.tex` at least once, then before building `file.tex` with just

`\usepackage[spanish,english]{babel}`

in the preamble, I'll have to delete `file.aux` and possibly `file.toc`.

Columns

The `multicol` Package

There are lots of occasions to type in multiple columns. One way to accomplish this is with the `twocolumn` option in the `\documentclass` command, but the `multicol` package can do a lot more.

Make sure to put \usepackage{multicol} in your preamble.

Very simply, use the `multicols` environment, specify your desired number of columns, and go to town. This will give you text in n equal width columns.

```
\begin{multicols}{n}
multicolumn text
\end{multicols}
```

Balancing Columns

The `multicols` environment balances the columns automatically. That is, it adjusts the vertical space to make all of the columns on the final page of the environment end at the same place.

Sometimes people don't want this. They don't mind unbalanced columns on the final page. This can be achieved simply by using the `multicols*` environment.

Example

Open the second example file (`.tex`), build and view.

More on `multicol`

The distance between columns is a length controlled by `\columnsep`. If you wanted one-half inch between columns, type this before the `multicols` environment: `\setlength{\columnsep}{.5in}`.

You can have a line show up between columns as well. This is a length (default is 0pt) controlled by `\columnseprule`. If you want a line between columns of 0.5pt thickness, type this: `\setlength{\columnseprule}{.5pt}`.

To force a column break, use `\columnbreak`.

Back to the Example

Make these adjustments in the previous example file to see the effect.

■ Adjust the number of columns on the first page from 3 to 4.

■ Force some column breaks on the first page.

■ Put 1 inch between columns only on the second page.

■ Put a 2pt thick line between columns *only on the first page*.

Section Titles

The `titlesec` package

Up until now, we have accepted the section title formatting as a given. But like just about everything in LaTeX, the appearance of these can be changed.

The key is to use the `titlesec` package. This package has two ways of operating:

- the simple interface; this is easier to learn but offers less in the way of customization; and
- the advanced interface; a lot more complicated to learn, but almost unlimited potential customizations.

We're going with the simple interface. See the package documentation for a more detailed explanation.

All of these adjustments are made in the preamble. Here is a table describing the allowed changes to the font.

Element	Options	Description
Font Family	`rm`	normal font
	`sf`	sans serif font
	`tt`	`typewriter type`
Font Series	`md`	medium weight
	`bf`	**bold**
Font Shape	`up`	upright
	`it`	*italics*
	`sl`	*slanted*
	`sc`	SMALL CAPS

More Changes

Here is a table describing some other changes possible with `titlesec`.

Element	Options	Description
Font Size	`big`	big font (default)
	`medium`	medium
	`small`	small
	`tiny`	size of text
Alignment	`raggedleft`	right justified
	`center`	centered
	`raggedright`	left justified
Vertical spacing	`compact`	reduces spacing

A First Example

What would it mean to have this in the preamble?

`\usepackage[sf,bf,small,center]{titlesec}`

My section (and subsection, etc.) titles would be

- in sans-serif font;
- bold;
- in a font size just bigger than the normal text; and
- centered.

Open the third example file (`.tex`), build and view.

Changing the Number

The \titlelabel command allows you to change the appearance of the number in your sections (and subsections, etc.).

Within this, \thetitle refers to the current sectioning number.

Example:
\titlelabel{(\thetitle)\hspace{.5in}}

This would enclose the section (or subsection) number in parentheses and then put one-half inch between that right parenthesis and the title for the section.

Changing One Element

The final command provided in this basic interface is the `\titleformat*` command. This allows you to modify just a single sectioning command. Any command given here overrides what was defined in the options in `\usepackage`.

Example:
```
\titleformat*{\subsection}
   {\bfseries\itshape\raggedright}
```

This makes my subsection titles bold, italicized, and left-justified.

In this command, use syntax like `\large` or `\LARGE` to change the font size.

Notice that I had to use `\bfseries` instead of `\textbf{}`. Technically speaking, `\textbf{}` is a *command* and `\bfseries` is a *declaration*.

Commands and Declarations

[This slide is a repeat from earlier in the course.]
You know how to use commands, but declarations can be used as either
`{\bfseries Bold}`

or as an environment: `\begin{bfseries}`.

Here's the correspondence:

Command	Declaration	Command	Declaration
`\textrm`	`\rmfamily`	`\textsf`	`\sffamily`
`\texttt`	`\ttfamily`	`\textmd`	`\mdseries`
`\textbf`	`\bfseries`	`\textup`	`\upshape`
`\textit`	`\itshape`	`\textsl`	`\slshape`
`\textsc`	`\scshape`	`\emph`	`\em`

Make these changes to the preamble of the previous example file, one at a time. See the effect they have on the document.

- Add this to the preamble:
  `\titlelabel{(\thetitle)\hspace{.5in}}`

- Add this to the preamble:
  `\titleformat*{\subsection} {\bfseries\itshape\raggedright}`

Let's practice!

Open the fourth example file (`.pdf`) and reproduce it.