

L<sup>A</sup>T<sub>E</sub>X courses  
Course 2 : A real-world document

Isabelle HURBAIN  
isabelle.hurbain@free.fr

August 4, 2009

## Contents

<b>1</b>	<b>The different types of documents</b>	<b>1</b>
<b>2</b>	<b>Accents and linebreaks</b>	<b>1</b>
<b>3</b>	<b>Real-world document HOWTO</b>	<b>2</b>
3.1	The title . . . . .	2
3.2	The sectioning commands . . . . .	3
3.3	The table of contents . . . . .	3
3.4	The starred commands . . . . .	4
3.5	The appendixes . . . . .	4
<b>4</b>	<b>Exercise</b>	<b>4</b>
<b>5</b>	<b>In the next course</b>	<b>5</b>

## 1 The different types of documents

In course 1, we made a "Hello, world" document, with only a title and a "Hello, world". We will now see how to make a real document.

There are mainly three wide-used classes of documents: book, report, and article. There are two other classes, namely slides and letters, but we won't talk about them yet.

`article` is for articles in journals, short reports, programs documentation...

`report` is for reports containing several chapters or parts.

`book` is for real books.

The rule I use (it's not a general rule, just what I do ;-)) is to use the article class below 6-8 pages, report between 8 and 50 pages, and book for more than 50 pages. Needless to say, I practically never use book :-)

## 2 Accents and linebreaks

When I first began to use L<sup>A</sup>T<sub>E</sub>X, the first things I wondered were "But why do my accents do not display?" and "How do I break a line?"

You can obtain accents by adding the line

```
\usepackage[latin1]{inputenc}
```

in the preamble of your document.

And to break a line, you will have to break the line twice in your source file, like this:

```
This is a line break  
that do not works.
```

```
But this one works.
```

```
Here is the proof.
```

To edit a document in an another language that english, you may also want to use the babel package. Babel manages the redefinitions of titles (to get Chapitre instead of Chapter in a french document for example) and some typographical rules (like the beginning of section tabulation, existing in french but not in english).

To use the babel package, add to your preamble

```
\usepackage[language]{babel}
```

where `language` is the language of your document. For example, you wille have

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

in french and

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

in german.

## 3 Real-world document HOWTO

A generic document would contain a title, a table of contents, an introduction, several sections, a conclusion, and appendixes. The parts can contain subsections.

### 3.1 The title

The title is defined for the whole document. Its parameters are given in the document's preamble (before `\begin{document}`) with the commands `\title`, `\author` and `\date`. These commands take an unique argument (between curly braces `{}`).

As you must have guessed, `\title` defines the title, `\author` defines the author (in most cases, that would be yourself), and `\date` defines the date of writing or of publishing of the document.

If `\date` is omitted, the date will be the one of the last compilation. To avoid this behaviour, you can use `\date{}` to define a void date.

Once the parameters are in the preamble, you just have to insert a `\maketitle` following `\begin{document}` to display the title.

The title will have an whole page for classes `report` and `book`, and will be in the beginning of the first page for an `article`.

### 3.2 The sectioning commands

There are 6 sectioning commands in an article (in hierarchical order):

- `\part`
- `\section`
- `\subsection`
- `\subsubsection`
- `\paragraph`
- `\subparagraph`

Report and book also have a `\chapter` command, between part and section.

These commands take one argument, the title of the sectioning item. It can also take an optional argument (between square brackets `[]`). This optional argument defines an alternative title, for example for the table of contents.

These commands manage the style of the title and its numbering. So for example, if you have a section foo before a section bar and if you decide that bar should be indeed before foo, just move your section. `LATEX` takes care of the numbering of your sections (and following subsections) and of the update of the table of contents. Fantastic, isn't it?

### 3.3 The table of contents

The table of contents can be obtained with `\tableofcontents`. It displays, in order of apparition, a title and a page number for each sectioning command lesser than `tocdepth`. `tocdepth` is an intern variable, corresponding to these values :

<code>\part</code> (book and report)	-1	<code>\part</code> (article)	0
<code>\chapter</code> (book and report)	0	<code>\section</code>	1
<code>\subsection</code>	2	<code>\subsubsection</code>	3
<code>\paragraph</code>	4	<code>\subparagraph</code>	5

The depth of the table of contents can be changed with `\setcounter{tocdepth}{n}` where n is the level of depth you want.

To update a table of contents after you modified something in the sectioning, you'll have to run `latex` twice on you document. Why? In fact, `LATEX` maintains a separate file for the table of contents, the `.toc` file. At the first compilation, this file is updated, at the second, the `.dvi` document is updated.

### 3.4 The starred commands

The so-called starred commands are `\chapter*`, `\section*`, etc. They are sectioning commands, but special ones. They do not update the table of contents, they are not numerotated, and they do not update the chapter, section, etc. counter.

So what's their interest? They are indeed mainly used for introduction and conclusion. The introduction, for example, is rarely called "Chapter 1 - Introduction" in a book.

You might want, however, that your introduction is not numerotated (so you would use the starred command) but appears in the table of contents (so you wouldn't).

To achieve this, you may do something like this:

```
\chapter*{Introduction}
\addcontentsline{toc}{chapter}{\numberline{}Introduction}
```

The `\chapter*{Introduction}` creates the title of the introduction, with no numerotation as it was desired.

The `\addcontentsline{toc}{chapter}{\numberline{}Introduction}` add a line in the table of contents to create the Introduction entry.

The same kind of trick can be used for the conclusion.

### 3.5 The appendixes

To define that you left the plain document and that you are now writing appendixes, just use the command `\appendix`. This sets the sections counters to 0 and change the numbering system to alpha. It also redefines, for book and report, the title of chapters from "Chapter" to "Appendix". So if you create a chapter foo after the command `\appendix`, you will have "Appendix A foo" as a title instead of "Chapter n+1 foo".

The appendixes are also entered in the table of contents.

## 4 Exercise

Now you have all instructions to create a basic document. Try to create one, experiment all the command described here, enjoy ;-)

Do not try to modify the aspect of what you get right now. Remember,  $\LaTeX$  uses a content-oriented markup (a logical markup), not a form-oriented markup. If you want to do a chapter, do not put it as a part because you feel it is nicer - we will see how to modify the chapter definition if you need it. But, if the logical structure of your document requires a chapter, put a `\chapter`. It will also have the advantage to make you think of your document's structure more deeply, and thus making clearer documents. And it highly simplifies the maintenance of your work.

## 5 In the next course

In the next course we will speak about tables and images and how to integrate them into a document.