

Court document sur l'utilisation de $\text{\LaTeX} 2_{\epsilon}$

par Thomas Nemeth

Fait le 10 novembre 2001

Table des matières

1	Introduction	4
2	Utiliser un Terminal et créer un répertoire	5
3	Commencer un document avec L^AT_EX	6
4	Chapitres, sections et paragraphes	8
5	Environnements	9
5.1	Tableaux	9
5.2	Listes	9
5.2.1	Listes simples d'éléments	9
5.2.2	Listes numérotées d'éléments	10
5.2.3	Listes descriptives d'éléments	10
5.3	Centrer, aligner le texte à gauche et à droite	11
5.4	Insérer des images	11
5.5	Utiliser les références aux images et tableaux	12
6	Obtenir son document	14
6.1	Compiler le document	14
6.2	Visualiser le document	14
6.3	Imprimer le document	14
6.4	Résumé des opérations	14
6.5	Autre solution	15
7	Compléments	16

Table des figures

Liste des tableaux

1	Quelques commandes	5
2	Les sections d'un document	8
3	Options de création d'un tableau	9
4	Alignement du texte	11

1 Introduction

Ce document s'adresse principalement aux débutants sous $\text{\LaTeX} 2_{\epsilon}$, mais reprends aussi quelques bases de la manipulation de fichiers sous Unix pour ceux qui ne seraient pas familier avec.

La section 2 explique quelques commandes sous Unix.

La section 3 montre comment commencer un document $\text{\LaTeX} 2_{\epsilon}$ rapidement.

La section 4 décrit les différentes manières de mettre en place les structures principales du document : chapitres, section, etc.

La section 5 liste quelques-uns des environnements les plus utilisés.

La section 6 explique comment obtenir son document final.

Finalement la section 7 contient quelques compléments sur \LaTeX et les exemples présentés.

2 Utiliser un Terminal et créer un répertoire

Un Terminal est un programme qui permet de taper des commandes et de naviguer parmi les répertoires. Généralement c'est une fenetre contenant uniquement du texte blanc — ou presque — sur fond noir ou noir sur fond blanc (quand je dis *presque*, c'est qu'il peut y avoir de la couleur, mais ça ne se voit pas de premier abord).

Une fois le Terminal ouvert, on se retrouve dans son répertoire personnel, symbolisé par un `~` juste après le texte `darkstar:` qui se trouve au début de chaque ligne (ça s'appelle l'invite de commande ou prompt).

Pour créer un répertoire permettant de stocker les fichiers d'un document, il faut taper `mkdir nom-du-répertoire`. Puis pour rentrer dans le répertoire, il faut faire `cd nom-du-répertoire`. Alors le nom du répertoire sera affiché juste après `darkstar:~` et donnera : `darkstar:~/nom-du-répertoire/\$`, le `$` signalant que l'on peut taper ses commandes. Le tableau 1 récapitule quelques commandes.

Commande	signification
<code>mkdir nom-du-répertoire</code>	créer le répertoire <code>nom-du-répertoire</code>
<code>cd nom-du-répertoire</code>	aller dans le répertoire <code>nom-du-répertoire</code>
<code>cd ..</code>	aller dans le répertoire parent
<code>ls</code>	lister les fichiers du répertoire

TAB. 1 – *Quelques commandes*

3 Commencer un document avec L^AT_EX

Commencer un document avec L^AT_EX consiste à rentrer tout un tas de commandes afin de faire comprendre à L^AT_EX d'utiliser les caractères français, ainsi que de donner un titre au document et en spécifier l'auteur.

Un document se commence donc toujours (ou presque) de la même façon, c'est-à-dire par un préambule. Ce préambule doit contenir les définitions du type de document voulu et les extensions utilisées. Vous pouvez vous référer à l'exemple fourni à la fin de cette section. La première ligne décrit les principales caractéristiques du document :

```
\documentclass[taille,a4paper]{classe}
```

La taille peut être au choix soit 10pt, soit 11pt, soit 12pt. La classe du document peut-être choisie parmi les suivantes :

report : pour faire de petits documents de 1 ou 2 pages.

letter : comme son nom l'indique c'est pour faire des lettres.

article : pour faire des documents explicatifs comme celui-ci.

book : pour faire des gros documents comme des livres ou des thèses.

Pour taper un document avec L^AT_EX 2_ε, il faut l'éditer avec *Nedit* (ou tout autre éditeur de texte, le mieux étant qu'il comporte un système de coloration syntaxique) : `nedit fichier.tex &`

Il est important de séparer les différentes parties du document à taper en plusieurs fichiers et ce pour de nombreuses raisons. En particulier pour les gros documents. D'une part parceque ça permet d'éditer de petits fichiers et nécessite moins de mémoire, d'autre part parceque ça permet d'aller plus vite à la sections désirée en éditant de fichier idoine, ensuite ça permet de n'avoir d'erreur que sur un seul fichier en cas de pb (coupure d'électricité, mauvaise manip, etc...), ensuite, ça permet de pouvoir facilement déplacer des sections dans le document en déplaçant simplement l'endroit où elles sont insérées. Pour ce faire, il faut créer des fichier T_EX contenant les chapitres ou sections choisis du document et de mettre une commande dans le fichier « principal » disant qu'il faut inclure tel ou tel fichiers. Il est à noter que le fait d'utiliser ce genre de séparation introduit l'insertion de création de nouvelles pages pour chaque fichier inclu.

Exemple: une thèse qui contient plusieurs chapitres. Il faut mettre le texte des chapitres dans des fichiers en `.tex` différents puis rajouter une ligne

`\include{fichier}` (sans le `.tex`) aux endroits où l'on veut inclure ces parties.
Ce qui donnerait :

```
\documentclass[12pt,a4paper]{book}
\usepackage[french]{babel}
\usepackage[dvips]{graphicx}
\usepackage[T1]{fontenc}
\author{par Toto Rhino}
\title{Sur la métépsychose des oursins mauve en laponie inférieure}
\date{Fait le \today}
\begin{document}

\maketitle
\pagebreak
\tableofcontents
\pagebreak
\listoffigures
\pagebreak
\listoftables
\pagebreak

\include{introduction}
\include{metempsychose}
\include{oursins}
\include{laponie}
\include{reproduction}
\include{vie}
\include{conclusion}

\end{document}
```

Chaque ligne include a un fichier `.tex` du même nom que le texte entre accolades. On remarque les commandes `\usepackage` et quelques autres avant le début du document proprement dit : le premier spécifique qu'on utilise le style *french* c'est-à-dire les règles de typographie française **par défaut**. Le suivant inclue un module d'insertion d'images Encapsuled Postscript (`.eps`). Le suivant est l'encodage à utiliser pour les textes en Français.

La création de document \LaTeX peut être facilitée par l'utilisation de logiciels adéquats comme `[X]Emacs` ou encore `glatex`. Ce dernier permet de personnaliser ses classes et de créer automatiquement le squelette du document tout en gérant la liste des fichiers `.tex` et associés.

4 Chapitres, sections et paragraphes

Le titre du document se spécifie juste en dessous du mot-clef `\author` par le mot-clef `\title{ Titre du document }` (bien qu'il n'y ait pas de place particulièrement recommandée, mais je me place ici dans le contexte de l'exemple de la section précédente). Il est alors possible de mettre juste après `\begin{document}` un ligne contenant `\maketitle` pour que la page de titre soit inclu dans le document, ainsi que `\tableofcontents` afin d'avoir la table des matières, comme récapitulé dans le tableau 2.

Chapitres	<code>\chapter{Titre du chapitre}</code>
Sections	<code>\section{Titre de la section}</code>
2 niveaux de sous-sections	<code>\subsection{Titre de la sous-section}</code> <code>\subsubsection{Titre de la sous-sous-section}</code>
Paragraphes	<code>\paragraph{Titre du paragraphe}</code>

TAB. 2 – *Les sections d'un document*

À noter que les chapitres ne fonctionnent que pour la classe `book`.

5 Environnements

5.1 Tableaux

Les tableaux se font de la façon suivante :

```
\begin{tabular}{Options}  
jusqu'à la fin :  
\end{tabular}
```

Les options déterminent comment sera fait le tableau sont présentées dans le tableau 3.

Option	signification
	mettre une barre verticale de séparation ici
c	centrer le texte de cette colonne
l	aligner à gauche le texte de cette colonne
r	aligner à droite le texte de cette colonne

TAB. 3 – Options de création d'un tableau

Exemple: `\begin{tabular}{|||c|}` créera un tableau à 2 colonnes alignées à gauche (2 l) et une colonne centrée (c). Il y aura des barres verticales de chaque côté du tableau ainsi qu'entre les colonnes.

À l'intérieur du tableau, on peut créer des barres de séparation horizontales par la commande `\hline`. Si l'on en met 2 à la suite, cela crée une séparation entre le titre des colonnes et le reste du tableau comme dans le tableau ci-dessus.

Il faut séparer les textes des différentes colonnes par des `&` et signifier la fin d'une ligne par `\\`.

Exemple: `Colonne 1 & Colonne 2 & Colonne 3 \\` remplira les 3 colonnes du tableau de l'exemple précédent.

Il est aussi intéressant de pouvoir mettre des légendes aux tableaux afin de pouvoir y faire référence dans le texte. Pour cela il faut lire la section 5.5, page 12, qui explique comment le faire avec des tableaux et des figures.

5.2 Listes

5.2.1 Listes simples d'éléments

Une liste simple d'éléments commence avec :

```
\begin{itemize}
jusqu'à la fin :
\end{itemize}
chaque élément étant spécifié par :
\item élément
```

Exemple :

- élément 1
- élément 2

5.2.2 Listes numérotées d'éléments

Une liste numérotée d'éléments commence avec :

```
\begin{enumerate}
jusqu'à la fin :
\end{enumerate}
chaque élément étant spécifié par :
\item élément
```

Exemple :

1. élément 1
2. élément 2

5.2.3 Listes descriptives d'éléments

Une liste descriptive d'éléments commence avec :

```
\begin{description}
jusqu'à la fin :
\end{description}
chaque élément étant spécifié par :
\item[élément] description
```

Exemple :

élément 1 : C'est le premier élément

élément 2 : C'est le second et dernier élément

5.3 Centrer, aligner le texte à gauche et à droite

Les débuts de blocs se commencent par :

```
\begin{type}
```

et se terminent par :

```
\end{type}
```

Avec *type* qui correspond à :

Comment faire pour	type
Centrer du texte	center
Aligner à gauche	flushleft
Aligner à droite	flushright

TAB. 4 – *Alignement du texte*

5.4 Insérer des images

Les images doivent être conforme à un certain format : le format Encapsuled PostScript (eps). Pour les obtenir, il faut les scanner ou les récupérer par tout autre moyen et les convertir. Ceci ne sera fait par moi que tant que tu ne sais pas le faire seule.

Pour insérer une image, il doit y avoir dans la partie entête du document, juste au dessus de `\usepackage[T1]{fontenc}` une ligne contenant la signalisation qu'il faut utiliser des images dans le document :

```
\usepackage[dvips]{graphicx}
```

Ensuite il faut insérer ceci juste avant `\maketitle` :

```
\newcommand{\schema}[4]  
{  
  \begin{figure}[!hbp]  
  \begin{center}  
  \includegraphics[#4]{#1}  
  \end{center}  
  \caption{#2}  
  \label{#3}  
  \end{figure}  
}
```

Enfin il faut taper à l'endroit où l'on veut mettre l'image :

`\schema{fichier.eps}{Commentaire de la figure}{mémo}{options non indispensables (peut être vide)}`

Cette image ne sera pas forcément placée là où on le veut exactement mais là où \LaTeX aura décidé qu'il est le mieux placé. En effet, c'est le programme qui sait comment sont organisées les pages et le texte qu'elles contiennent. Pour insérer à tout prix une image à un endroit voulu il faut soit la rétrécir, soit créer une page blanche juste au-dessus par la commande `\clearpage`.

Les options supplémentaires peuvent ne pas être spécifiées, mais il faudra toujours laisser les accolades. Généralement l'option qui devra être utilisée est `width=12cm`, spécifiant une largeur de 12cm pour l'image, la taille peut bien sûr varier en fonction de ce que l'on désire.

Afin de pouvoir faire références aux images insérées dans le document, il est nécessaire de faire appel à des commandes supplémentaires dans le texte (voir la section 5.5, page 12).

5.5 Utiliser les références aux images et tableaux

Il est toujours bien pratique de pouvoir faire référence aux tableaux et images insérés dans le document non seulement par le numéro qu'ils comportent mais aussi par la page sur laquelle ils se trouvent. Ces 2 valeurs sont automatiquement gérées par le système et sont remplacées par les bonnes valeurs si elles venaient à changer au cours de la modification du document.

Avant de pouvoir faire référence à un tableau il faut d'abord l'entourer de délimiteurs pour signifier que ce tableau est à numéroté. Il faut donc lui mettre avant `\begin{tabular}{Options}` les deux commandes :

```
\begin{table}[!hbp]
\begin{center}
```

Ensuite, une fois le tableau tapé comme précédemment, il faut lui donner un label pour utiliser comme référence et une légende (*caption* en anglais). Ce qui est fait par la liste de commandes suivantes :

```
\caption{légende}
\label{label référence}
\end{center}
\end{table}
```

Pour faire référence à ces éléments, il faut remplacer le numéro de l'élément par `\ref{mémo}` et le numéro de la page par `\pageref{mémo}`, ces 2 commandes devant être collées aux dernier mot. Par exemple :

«voir la figure`\ref{mémo}`, page`\pageref{mémo}`.»

Bien sûr, il sera ensuite possible de faire la liste des figures et la liste des tableaux dans le document par les commandes respectives `\listoffigures` et `\listoftables`

Il y a bien sûr tout plein d'autres options à voir dans le manuel...

6 Obtenir son document

6.1 Compiler le document

Une fois du texte tapé, modifié, il faut pouvoir le générer, car le texte doit être récompilé pour générer un document qui, lui, sera capable d'être visualisé et imprimé. La compilation est donc une étape O-B-L-I-G-A-T-O-I-R-E et c'est ce que se propose de faire \LaTeX . Dans le Terminal, il suffit de taper :

```
latex fichier.tex
```

et de laisser se dérouler la création du document.

Si une erreur survient, c'est qu'un mauvais mot clef a été entré et le fichier ainsi que la ligne où l'erreur se trouve est indiquée.

6.2 Visualiser le document

Une fois le document (re-)généré, il est possible de le visualiser avec *xdvi*. Il suffit alors de taper :

```
xdvi fichier.dvi &
```

NE PAS OUBLIER LE &! Comme ça il sera possible de le laisser fonctionner pendant qu'on s'occupe de continuer à modifier le document. D'autre part, il n'est besoin de le lancer qu'une fois, contrairement à \LaTeX qui a besoin d'être relancé à chaque fois qu'on modifie une partie, aussi infime soit-elle, du document. Pour le faire s'actualiser avec le document, une fois \LaTeX relancé, il suffit de cliquer sur le texte qu'il affiche.

6.3 Imprimer le document

Enfin, pour l'imprimer, il faut taper dans le Terminal :

```
dvips fichier.dvi
```

pour imprimer tout le document ou

```
dvips -p première-page-à-imprimer -l dernière-page-à-imprimer fichier.dvi
```

pour seulement imprimer les pages du document comprises entre les 2 spécifiées y compris elles-même.

6.4 Résumé des opérations

Pour pouvoir travailler sur un document (ex. pour la thèse dans ce cas) il faut faire les choses suivantes :

- modifier le document (ou un des fichiers dont il est composé)

- ouvrir un Terminal
- taper `cd these`
- taper `latex these.tex` (3 fois de suite – il est possible de revoir ce qu'on a tapé avec les flèches vers le haut)
- taper `xdvi these.dvi &` pour visualiser
- re-taper `latex these.tex` (3 fois) à chaque fois que l'on a modifié le document – la visualisation des modifications se fait automatiquement si on a tapé correctement le `&` pour `xdvi`.
- taper `dvips these.dvi` pour imprimer

6.5 Autre solution

Une autre solution pour compiler et visualiser facilement ses documents, est d'utiliser le logiciel précédemment cité (`glatex`) qui permet, grâce à une barre de boutons, de lancer directement les bonnes commandes. De même il permet d'imprimer — et ce, même dans un fichier PostScript ou PDF.

7 Compléments

On peut ajouter que les guillemets (« ») se font avec 2 "<" ou ">" collés l'un à l'autre (les règles de typographie françaises disent d'utiliser « et » plutôt que " et ").

De même le *texte* peut être mis en valeur par la commande `\emph{texte}`.

Ne pas oublier la typographie française :

1. Les signes simples (.,) sont collés aux mots qui les précèdent.
2. Les signes doubles ont une espace avant et après (?! « »).
3. Il faut mettre les accents sur les majuscules (Pour cela, j'ai ma propre configuration du clavier qui permet d'obtenir facilement toutes les majuscules accentuées).

D'autres utilisations sont décrites dans « Une courte (?) introduction à $\text{\LaTeX} 2_{\epsilon}$ », comme les lettres grecques et autres raffinements comme l'encadrement du texte. D'ailleurs ce document est sous-titré « *ou $\text{\LaTeX} 2_{\epsilon}$ en 88 minutes* », donc pas loin d'une heure et demie : c'est pas long à lire pour voir ce qu'on peut y trouver...

Le logiciel `glatex` présenté dans ce document est disponible à l'adresse <http://tnemeth.free.fr/projets/glatex.html>.