

Formation T_EX : programmation L^AT_EX

Denis ROEGEL

roegel@loria.fr

1999

1	Introduction	5
1	1 T _E X	5
2	2 Objectifs du cours	5
3	3 Présentation du formateur	6
2	Les sources d'information	7
3	Rapide survol de L ^A T _E X	8
1	1 Historique	8
2	2 Que fait-on avec un fichier L ^A T _E X ?	9
3	3 T _E Xlive	10
4	4 Généralités sur L ^A T _E X	11
5	5 Exemple de fichier	11
6	6 Définitions	12
7	7 Environnements	12
4	Compilation d'un exemple simple	13
5	Notion de format	15
6	Étude sommaire de la compilation de l'exemple 1	16
7	Exemple plus complexe	17
8	Lecture de code source L ^A T _E X	17
9	Premières modifications de la mise en page	19
1	1 Comment modifier ?	19
2	2 Discipline	20

3	Modification du titre	21
10	Modifications dans le fichier principal	26
11	Création d'un <i>package</i>	27
12	Création d'une classe	28
13	Dimensions : marges et polices	29
	1 Marges	29
	2 Polices	32
14	Structure du document et numérotations	37
	1 Sectionnement	37
	2 Numérotation des pages	46
	3 Notes de bas de page	46
	4 Haut et pied de page	47
	5 Listes	50
15	Parties annexes	54
	1 Pages de couverture, dédicace, etc.	54
	2 Table des matières	54
	3 Index	58
	4 Bibliographie	59
16	Utilisation de fichiers externes	61
17	Objets flottants	62
18	Quelques notions de programmation avancée	67

1	Commandes robustes et fragiles	67
2	Résolution d'interactions entre <i>packages</i>	67
3	Caractères actifs	68
4	Compteurs	69
19	Les messages d'erreur	69
20	Exemple de thloria	70
21	Problèmes de mémoire	71
22	Comprendre les documents qui sont soumis	72
23	Travail sur un livre	72
24	Guides pour l'écriture de classes	73
25	Suggestion pour aller plus loin	73

1. Introduction

1.1. T_EX

1.2. Objectifs du cours

- Ce que vous saurez faire et ce qui vous restera à apprendre à l'issue de cette formation.
- Ce qui ne sera pas traité : mathématiques, graphiques, détail des polices.
- Vous aurez un certain savoir-faire et une certaine autonomie. Mais il restera des problèmes difficiles qui ne pourront pas être résolus en cinq minutes.

1.3. Présentation du formateur

- Maître de conférences en informatique à l'université Nancy 2 ;
- Activités T_EX et expérience :
 - Utilisation de T_EX depuis 1991 ;
 - Création de divers styles et en particulier d'une classe de thèse utilisée au LORIA et ailleurs ;
 - Développement d'un site web consacré à T_EX ;
 - Écriture d'articles autour de T_EX.

2. Les sources d'information

- Donald Knuth : The T_EXbook
- Thomas Lachand-Robert : La maîtrise de T_EX et L^AT_EX.
- Raymond Séroul : Le petit livre de T_EX
- Leslie Lamport : L^AT_EX, A Document Preparation System
- Goossens, Rahtz et Samarin : The L^AT_EX Companion
- Autres livres
- Manuels des *packages*
- La distribution T_EX elle-même
- Les sites ftp : CTAN ou miroirs CTAN (ex. ftp.loria.fr)
- Les sites d'information généraux : L^AT_EX Navigator au LORIA
(<http://www.loria.fr/tex>)
- Guide local du LORIA (disponible sur le L^AT_EX Navigator)

3. Rapide survol de \LaTeX

3.1. Historique

- \TeX : Knuth 1977–1985
- \LaTeX : Lamport ca. 1982–1986 puis repris à partir de 1989 par l'équipe de Schöpf et Mittelbach pour aboutir à $\text{\LaTeX} 2_{\epsilon}$.

\LaTeX : plus de structure que « plain \TeX »

Structuration analogue au logiciel *Scribe*.

3.2. Que fait-on avec un fichier \LaTeX ?

- Édition avec un éditeur de texte quelconque, voire même un autre traitement de texte ;
- Compilation : obtention d'un fichier `dvi` ;
- Visualisation du fichier `dvi`, ou transformation en PostScript, en PDF, etc.
- Envoi, Impression, etc.

3.3. T_EXlive

Environnement complet pour T_EX et L^AT_EX pour de nombreuses plate-formes. Contient les distributions gratuites standard.

Environnement respectant la norme TDS (T_EX Directory Structure).

3.4. Généralités sur L^AT_EX

3.5. Exemple de fichier

```
\documentclass[a4paper,12pt]{book}
\usepackage[T1]{fontenc}
\usepackage[latin1]{inputenc}
\usepackage{french}
\begin{document}
\chapter{Thèse}
```

Le texte de thèse.

```
\chapter{Antithèse}
```

Le texte de l'antithèse.

```
\chapter{Synthèse}
```

Le texte de synthèse.

```
\bibliographystyle{plain}
\bibliography{ref}
```

```
\end{document}
```

3.6. Définitions

Sans paramètres :

```
\newcommand{\commande}{Ceci est l'expansion  
de la commande}
```

Avec paramètres :

```
\newcommand{\commandeII}[1]{Ceci est l'expansion  
et le parametre est #1.}
```

On peut aussi définir une commande avec un paramètre optionnel.

3.7. Environnements

Les environnements sont identifiés par `\begin` et `\end`. Exemple :

```
\begin{itemize}
```

```
...
```

```
\end{itemize}
```

```
\begin{document}
```

```
...
```

```
\end{document}
```

```
...
```

4. Compilation d'un exemple simple

Étude de la compilation de `essai.tex` :

```
\documentclass{article}
\begin{document}
\title{Essai}
\author{Denis Roegel}
\date{11 janvier 1999}
\maketitle
\end{document}
```

qui donne :

```
This is TeX, Version 3.14159 (Web2C 7.2)
(essai.tex
LaTeX2e <1998/06/01>
Babel <v3.6h> and hyphenation patterns for english,
french, german, russian, dumylang, nohyphenation, loaded.
(/usr/local/tex/texmf/tex/latex/base/article.cls
Document Class: article 1998/05/05 v1.3y
                Standard LaTeX document class
(/usr/local/tex/texmf/tex/latex/base/size10.clo))
No file essai.aux.
[1] (essai.aux) )
Output written on essai.dvi (1 page, 376 bytes).
Transcript written on essai.log.
```

- Savoir lire et comprendre ce genre de message.
- Fichiers qui interviennent ici.
- Fichiers utilisés, mais non mentionnés (polices et fichiers ouverts avec `\openin` ou `\openout`);
- Davantage d'informations se trouvent dans le fichier `.log`.
- Localisation des fichiers.
- `kpsewhich` permet aussi de localiser des fichiers.

5. Notion de format

- Exécutable : $\text{T}_{\text{E}}\text{X}$ (en fait $\text{V}_{\text{ir}}\text{T}_{\text{E}}\text{X}$, pour $\text{T}_{\text{E}}\text{X}$ vierge), environ 300 commandes connues, toutes décrites dans le $\text{T}_{\text{E}}\text{X}$ book.
- Commandes supplémentaires fournies dans un « format précompilé » (fichiers `.fmt`), par exemple $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$. Pour savoir ce qui est dans le format, regarder comment il a été créé.

```
initex latex.ini
```

ou

```
initex latex.ltx
```

Il ne faut pas modifier les fichiers utilisés à la création du format, mais il faudra parfois s'y reporter. Cela concerne essentiellement `latex.ltx`.

Un format doit être recréé dans certains cas, par exemple lorsqu'on veut ajouter des coupures pour une nouvelle langue.

6. Étude sommaire de la compilation de l'exemple 1

- ouverture du fichier `essai.tex` par $\text{T}_\text{E}\text{X}$
- affichage de « `LaTeX2e <1998/06/01>` » et autres informations sur le format.
- Commande `\documentclass`; cette commande est définie dans le format (chercher dans `latex.ltx`); appelée avec le paramètre « `article` », elle fait charger la classe `article.cls` qui contient des commandes spécifiques à la mise en page d'articles.
- La taille standard n'étant pas modifiée, `article.cls` charge `size10.clo` pour des définitions propres à une taille de caractères de 10pt.
- Commande `\begin` (chercher dans `latex.ltx`) avec le paramètre « `document` ». Elle conduit en particulier à la création du fichier auxiliaire `essai.aux`.
- Commandes `\title`, `\author`, `\date` (dans `latex.ltx`) et `\maketitle` (dans `article.cls`).
- Commande `\end` avec le paramètre « `document` ». Cette commande ferme le `\begin` précédent et fait lire le fichier `essai.aux` pour détecter d'éventuels changements.
- Résultat : `essai.dvi`

7. Exemple plus complexe

Cet exemple va servir de base à la plupart des exercices ultérieurs.

Le déroulement de la suite est d'aller toujours plus loin dans les modifications et d'introduire les concepts T_EX nécessaires au fur et à mesure des besoins.

Principe de base : *faire simple et comprendre ce que l'on fait.*

Ne pas charger des *packages* si l'on ne s'en sert pas, car cela peut réserver des surprises (interférences).

8. Lecture de code source L^AT_EX

- Primitives
- Commandes avec paramètres, sans paramètres, etc.
- Rôle du blanc et du « % ».

En général, on se sert du « % » pour supprimer la fin de ligne qui peut introduire un espace non désiré. Toutefois, « % » est inutile après un nom de commande car un nom de commande absorbe automatiquement l'espace qui suit.

Tiré de article.cls :

```
\def\@maketitle{%
  \newpage
  \null
  \vskip 2em%
  \begin{center}%
  \let \footnote \thanks
    {\LARGE \@title \par}%
    \vskip 1.5em%
    {\large
      \lineskip .5em%
      \begin{tabular}[t]{c}%
        \@author
      \end{tabular}\par}%
    \vskip 1em%
    {\large \@date}%
  \end{center}%
  \par
  \vskip 1.5em}
```

9. Premières modifications de la mise en page

9.1. Comment modifier ?

- Uniquement des lettres dans les noms des commandes.
- Mécanisme de protection pour travailler avec des commandes privées dans le format, les classes et les *packages*. Ceci évite à l'utilisateur de modifier par erreur une commande.
- Utiliser `\makeatletter` et `\makeatother`.

9.2. Discipline

- Utiliser `\newcommand` ou `\renewcommand` et éviter `\def` car c'est une commande dangereuse.
- Importance de la bonne présentation du code, des commentaires, etc. pour que la compréhension et la maintenance du code soient facilitées.
- Écriture d'un mode d'emploi.

9.3. Modification du titre

Exercice :

- Trouver la commande `\maketitle` et comprendre (au moins en partie) ce qu'elle fait.
- Mettre le titre en gras.
- Changer le positionnement du titre.
- Ajouter des filets, avant et après le titre.
- Encadrer le titre.

De latex.ltx :

```
\def\title#1{\gdef\title{#1}}  
\def\ttitle{\@latex@error{No \noexpand\ttitle given}\@ehc}  
\def\author#1{\gdef\@author{#1}}  
\def\@author{\@latex@warning@no@line{No \noexpand\author given}}  
\def\date#1{\gdef\@date{#1}}  
\gdef\@date{\today}
```


Si on ne veut pas analyser tout le code, il faut deviner ce qui se passe. Ici, `\maketitle` appelle `\@maketitle` (voir plus haut) qui met le titre en forme.

On examine trois moyens de faire la modification :

- Modifications dans le fichier principal : pour des petites modifications utiles uniquement au document traité ;
- Création d'un *package* : pour des modifications pouvant servir dans d'autres documents ;
- Création d'une classe ; comme pour un *package*, mais quand les modifications sont si importantes qu'elles définissent une nouvelle classe de documents.

Les deux dernières solutions sont simplement des enrobages de la première.

10. Modifications dans le fichier principal

Les modifications sont placées avant qu'elles n'interviennent, souvent dans le préambule (entre `\documentclass` et `\begin{document}` et entre `\makeatletter` et `\makeatother` si les commandes comportent des « @ »).

Exemple :

```
\documentclass[a4paper]{book}
\makeatletter
\@addtoreset{chapter}{part}
\makeatother
\begin{document}
...
```

11. Création d'un *package*

Un *package* est un fichier d'extension `.sty`. Il a en général la structure suivante :

```
\NeedsTeXFormat{LaTeX2e}[1994/12/01]
\ProvidesPackage{ifthen}
  [1997/11/02 v1.0n Standard LaTeX ifthen package (DPC)]
  ... commandes ...
\endinput
```

`\makeatletter` et `\makeatother` sont implicites dans un *package*.

Un *package* est chargé avec `\usepackage`, éventuellement avec des options.

Un *package* peut aussi charger d'autres *packages*. Exemple :

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{graphicx}
  [1997/06/09 v1.0d
  Enhanced LaTeX Graphics (DPC,SPQR)]
\DeclareOption*{\PassOptionsToPackage
  \CurrentOption{graphics}}
\ProcessOptions
\RequirePackage{keyval,graphics}
... commandes ...
\endinput
```

12. Création d'une classe

Une classe est construite sur le modèle suivant :

```
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesClass{article}
      [1998/05/05 v1.3y
  Standard LaTeX document class]
...
\DeclareOption{11pt}{\renewcommand\@ptsize{1}}
...
\ExecuteOptions{letterpaper,%
      10pt,oneside,onecolumn,final}
\ProcessOptions
\input{size1\@ptsize.clo}
...
\endinput
```

Si une classe a besoin d'un *package*, elle peut le charger avec

```
\RequirePackage :
\RequirePackage[T1]{fontenc}
```

Une classe peut dériver d'une autre classe :

```
...
\DeclareOption*{%
      \PassOptionsToClass{\CurrentOption}{book}}
\ProcessOptions
\LoadClass[a4paper]{book}[1995/01/31]
...
```

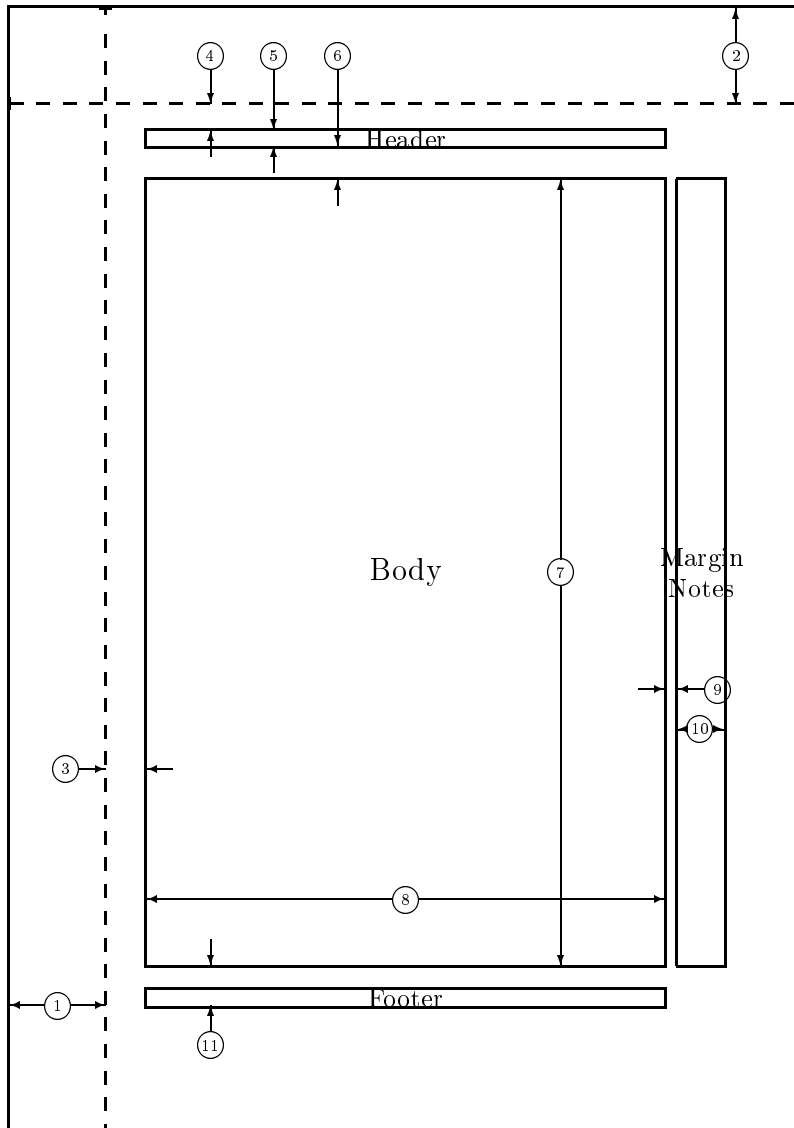
13. Dimensions : marges et polices

13.1. Marges

Les paramètres d'une page peuvent être modifiés et la mise en page observée à l'aide du *package* `layout` en exécutant la commande `\layout` :

```
\documentclass[a4paper,12pt]{article}
\usepackage{layout}
\begin{document}
\layout
\end{document}
```

dont le résultat est :



one inch	hoffset	one inch	voffset
oddsidemargin	pt	topmargin	pt
headheight	pt	headsep	pt
textheight	pt	textwidth	pt
marginparsep	pt	marginparwidth	pt
footskip	pt	marginparpush	pt (not shown)
hoffset	pt	voffset	pt
paperwidth	pt	paperheight	pt

Commandes `\newlength`, `\setlength`, `\addtolength`

Les marges peuvent aussi être modifiées via le *package* `vmargin`.

L'une des commandes est `\setmargins` et sa syntaxe est :

```
\setmargins{<leftmargin>}{<topmargin>}{<textwidth>}{<textheight>}  
{<headheight>}{<headsep>}{<footheight>}{<footskip>}
```

Par exemple, une mise en page typique de journal scientifique peut être satisfaite avec :

```
\usepackage{vmargin}  
\setpapersize{A4}  
\setmargins{1.0in}{1.25in}{6.8in}{8.8in}{0pt}{0mm}{0mm}
```

13.2. Polices

- Les dessins des caractères sont contenus dans des fichiers externes, par exemple des fichiers tels que `cmr10.600pk` ;
- Les dimensions des caractères sont contenus dans des fichiers `.tfm`, par exemple `cmr10.tfm`. $\text{T}_{\text{E}}\text{X}$ n'a besoin que des fichiers `.tfm`.
- Commande de bas niveau ($\text{T}_{\text{E}}\text{X}$) pour déclarer une police : `\font`

Exemple :

```
\font\test=cmr10
```


– Avec \LaTeX la sélection des polices est plus élaborée et fait appel à des attributs. Cinq attributs définissent une police :

- le codage
- la famille
- la série ou graisse
- l’aspect
- la taille

Les correspondances entre ces attributs et les fichiers externes sont données dans des fichiers `.fd`. Exemple :

```
\DeclareFontShape{OT1}{cmr}{m}{it}%  
  {%  
    <5><6><7>cmti7%  
    <8>cmti8%  
    <9>cmti9%  
    <10><10.95>cmti10%  
    <12><14.4><17.28><20.74><24.88>cmti12%  
  }{}
```

– Les attributs de la police courante peuvent être modifiés avec

– `\fontencoding`, par exemple `\fontencoding{T1}`

– `\fontfamily`, par exemple `\fontfamily{ptm}`

– `\fontseries`, par exemple `\fontseries{bx}`

– `\fontshape`, par exemple `\fontshape{it}`

– `\fontsize`, par exemple `\fontsize{12}{14}`

Les commandes précédentes doivent être suivies de

`\selectfont`.

– Autres commandes :

– `\bfseries`

– `\itshape`

– `\large`

– `etc.`

- `\normalfont` permet de revenir à une police droite de la famille courante.
- Le codage de la police définit la position des caractères dans la police. C'est le codage de sortie. Il y a aussi un codage d'entrée. Ces codages doivent être indiqués à \LaTeX . Si l'on écrit en codage ISO-Latin 1 avec les polices EC (au codage T1), on indiquera :

```
\usepackage[T1]{fontenc}
```

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

Avec \TeX (par opposition à $\text{MIT}\TeX$), il faut le codage T1 pour couper correctement un texte en Français.

14. Structure du document et numérotations

14.1. Sectionnement

Une commande de sectionnement, comme `\section`,

- produit un nombre reflétant la hiérarchie
- stocke le titre pour la table des matières
- stocke le titre pour l'en-tête
- formate le titre

– Compteurs :

Chaque commande de sectionnement est associée à un compteur.

Exemple : à `\section` est associé le compteur `'section'`.

Par défaut, dans la classe `book` (`book.cls`) :

```
\newcounter{part}  
\newcounter{chapter}  
\newcounter{section}[chapter]  
...
```

Ceci signifie que `part` et `chapter` sont indépendants mais que les sections sont remises à 0 lorsque l'on passe à un nouveau chapitre.

Exercice :

Faire une copie de `book.cls` et faire en sorte que les chapitres soient remis à 0 à chaque partie. Tester sur un exemple.

- Il est aussi possible de faire la même modification au sein du fichier principal, en utilisant `\@addtoreset` :
`\@addtoreset{chapter}{part}`

Exercice :

Essayer ce qui précède.

– Formatage :

Le formatage des numéros se fait via des commande préfixées par `\the`. Par défaut, le formatage des chapitres revient à faire :
`\renewcommand\thechapter{\arabic{chapter}}`

Exercice :

Cherchez ces définitions dans `book.cls` et commentez les éventuelles différences. Analysez la définition `\thesection`.

Les compteurs sont aussi utilisés pour les références et toute modification au niveau du numéro apparaissant au début de la section sera aussi répercutée là où la section est référencée.

Exercice :

Essayez de faire en sorte que le numéro de section soit encadré (avec `\fbox`) en modifiant `\thesection` et testez le mécanisme des références.

Le formatage du compteur proprement dit est fait par la commande `\@secntformat`, définie dans `latex.ltx`.

Exercice :

Modifiez cette définition de manière appropriée et comparez à la précédente solution.

- Formatage général d'un titre de section : commande `\@startsection`
- Les commandes `\section`, `\subsection`, etc., sont définies avec `\@startsection`. La syntaxe est :

```
\@startsection{<name>}{<level>}{<indent>}{<beforeskip>}{<afterskip>}{<style>}
```

Exemple :

```
\newcommand\section{%  
  \@startsection  
  {section}% nom du compteur  
  {1}% niveau  
  {\z@}% indentation par rapport a la marge  
  {-3.5ex \@plus -1ex \@minus -.2ex}% espace avant  
  {2.3ex \@plus .2ex}% % espace apres  
  {\normalfont\Large\bfseries}}% style
```

Suivant le signe du *saut avant*, le premier alinéa sera est indenté (positif) ou non (négatif).

Les parties `\@plus` et `\@minus` correspondent à une flexibilité de l'espacement.

Le signe de l'*espace après* indique s'il y a ou non un saut entre le titre de section et le premier alinéa.

Exercice :

Redéfinissez \paragroph pour qu'il y ait une ligne blanche entre le titre du paragraphe et le texte qui suit. Faites en sorte que le premier alinéa soit indenté.

Les premiers alinéas peuvent être tous indentés avec le package `indentfirst`.

– Titres de chapitres et de parties :

Ces titres ne sont pas faits avec `\@startsection`.

`\chapter` par exemple, est défini dans `book.cls` à partir de `\secdef\@chapter\@schapter`, signifiant que `\chapter` fera appel à `\@chapter` et `\chapter*` à `\@schapter` ('s' pour 'star'). La commande `\@chapter` va appeler `\@makechapterhead` pour formater le titre.

```
\def\@makechapterhead#1{% #1 = titre du chapitre
  \vspace*{50\p@}%
  {\parindent \z@ \raggedright \normalfont
    \ifnum \c@secnumdepth >\m@ne
      \if@mainmatter
        \huge\bfseries \@chapapp\space \thechapter
        \par\nobreak
        \vskip 20\p@
      \fi
    \fi
    \interlinepenalty\@M
    \Huge \bfseries #1\par\nobreak
    \vskip 40\p@
  }}

```

Exercice :

Mettez le numéro du chapitre et le titre sur la même ligne, après avoir supprimé le mot 'chapitre', et alignez le tout à droite.

- Composantes fixes des titres : `\chaptername`, `\partname`, etc. contiennent les noms 'Chapter', 'Part', etc. Ces commandes sont redéfinies avec `\renewcommand` :
`\renewcommand{\abstractname}{Summary}`

14.2. Numérotation des pages

Le compteur « page » permet de modifier le numéro d'une page :

```
\setcounter{page}{10}
```

Le style du numéro de page (arabe ou romain) peut-être modulé avec `\pagenumbering`.

14.3. Notes de bas de page

La numérotation des notes de bas de page peut être modulée en modifiant `\thefootnote`.

14.4. Haut et pied de page

Commandes de bas niveau

Le *style d'une page* consiste en les hauts et bas de page. Chaque style a un nom et il lui correspond une commande `\ps@...`. Par exemple, au style 'headings', il correspond la commande `\ps@headings`. Celle-ci est déclenchée lorsque l'on écrit `\pagestyle{headings}`.

`\ps@headings` est défini dans la classe, par exemple `book.cls` et la définition dépend du mode de mise en page (`oneside` ou `twoside`).

Une définition typique (mais simplifiée) est :

```
\def\ps@headings{%
  \let\@oddfont\@empty\let\@evenfont\@empty
  \def\@evenhead{\thepage\hfil\slshape\leftmark}%
  \def\@oddhead{{\slshape\rightmark}\hfil\thepage}%
}
```

- les bas de page impair (odd) et pair (even) sont vides ;
- le haut de page pair (even) comporte le numéro de page à gauche et la ‘marque gauche’ à droite, et en police oblique (‘slanted’) ;
- le haut de page impair (odd) comporte la ‘marque droite’ à gauche et le numéro de page à droite.

Les marques sont définies par des commandes comme

`\chaptermark`, `\sectionmark` et éventuellement d’autres. La marque gauche est typiquement le premier chapitre apparaissant sur la page et la marque droite le dernier numéro de section.

On trouvera les définitions de `\chaptermark`, etc. par exemple dans `book.cls`. Les définitions ne sont pas nécessairement les mêmes pour tous les styles de page.

Exercice :

Définissez un nouveau style appelé 'perso' sur le modèle du style 'headings', où le bas de page gauche est identique au haut de page droit du style 'headings' et où le bas de page droit est identique au haut de page gauche du style 'headings'. Testez ce style sur un fichier que vous aurez rempli de quelques chapitres et sections.

Emploi du package fancyhdr

Le *package* fancyhdr de Piet van Oostrum permet de modifier facilement les styles d'en-têtes. La documentation de ce *package* est disponible sur le LN.

14.5. Listes

Environnement *enumerate*

Il y a quatre niveaux et chaque niveau a un compteur. Les noms des compteurs sont : `enumi`, `enumii`, `enumiii`, `enumiv`. La représentation d'un compteur se fait via `\theenum...` et l'affichage via `\labelenum...`. De plus, chaque niveau est muni d'un préfixe pour construire les références. Ce préfixe est dénoté par `\p@enum...`

En résumé :

Niveau	1	2	3	4
Compteur	<code>enumi</code>	<code>enumii</code>	<code>enumiii</code>	<code>enumiv</code>
Représentation	<code>\theenumi</code>	<code>\theenumii</code>	<code>\theenumiii</code>	<code>\theenumiv</code>
Par défaut	<code>\arabic{enumi}</code>	<code>\alph{enumii}</code>	<code>\roman{enumiii}</code>	<code>\Alph{enumiv}</code>
Label	<code>\labelenumi</code>	<code>\labelenumii</code>	<code>\labelenumiii</code>	<code>\labelenumiv</code>
Par défaut	<code>\theenumi.</code>	<code>(\theenumii)</code>	<code>\theenumiii.</code>	<code>\theenumiv.</code>
Exemple de num.	1.,2.	(a),(b)	i.,ii.	A.,B.
Préfixe	<code>\p@enumi</code>	<code>\p@enumii</code>	<code>\p@enumiii</code>	<code>\p@enumiv</code>
Par défaut	{ }	<code>\theenumi</code>	<code>\theenumii</code> (<code>\theenumiii</code>)	<code>\p@enumiii</code> <code>\theenumiii</code>
Exemple de réf.	1,2	1a,2b	1(a)i,2(b)ii	1(a)iA,2(b)iiB

(tableau extrait du *L^AT_EX Companion*)

Exercice :

Faites en sorte que tous les niveaux soient « numérotés » alphabétiquement (utiliser `\Alph` comme dans le quatrième niveau) et que dans les références un niveau soit séparé du suivant par ‘-’.

Environnement *itemize*

Les items de l'environnement `itemize` sont contrôlés par les commandes `\labelitemi`, `\labelitemii`, etc.

Si les redéfinitions ne doivent être faites que pour une liste, on les mettra juste après `\begin{itemize}`.

Environnement *description*

L'environnement `description` peut être paramétré en redéfinissant la commande `\descriptionlabel` qui crée le label.

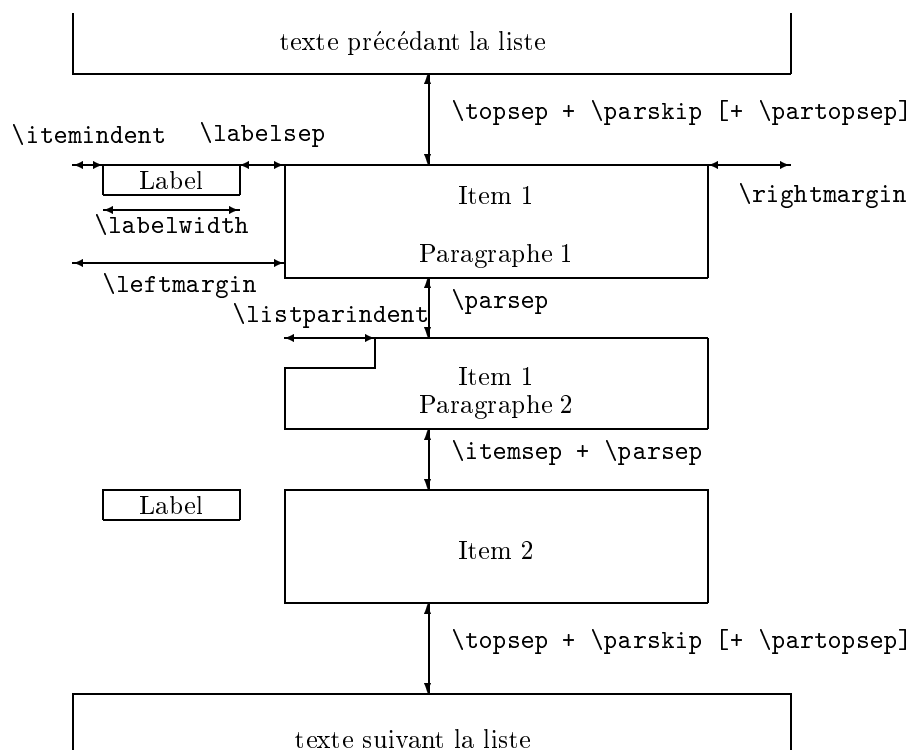
```
\renewcommand{\descriptionlabel}[1]{%  
    \hspace{\labelsep}\textsf{#1}}
```

Autres listes

On peut définir de nouvelles listes via l'environnement 'list' :

```
\begin{list}{\langle default_label \rangle}{\langle decls \rangle} \langle item_list \rangle \end{list}
```

Paramètres dans les listes



15. Parties annexes

15.1. Pages de couverture, dédicace, etc.

Certaines classes proposent des commandes pour les pages de couverture (option `titlepage` dans la classe `book` par exemple), mais cet élément doit en général être très particularisé.

15.2. Table des matières

La table des matières est générée dans un fichier `.toc`. Elle comporte des entrées de la forme

```
\contentsline{<type>}{<text>}{<page>}
```

où

- `type` vaut 'section', 'subsection', etc. Cela sert à déterminer l'indentation, le style, etc.
- `text` est ce qui apparaît.
- `page` est le numéro de la page.

Cette liste est générée automatiquement par les commandes de sectionnement qui exécutent des commandes `\addtocontents` ou `\addcontentsline`. Par exemple, la définition de `\@chapter` dans `book.cls` est :

```
\def\@chapter[#1]#2{%
  \ifnum \c@secnumdepth >\m@ne
    \if@mainmatter
      \refstepcounter{chapter}%
      \typeout{\@chapapp\space\thechapter.}%
      \addcontentsline{toc}{chapter}%
        {\protect\numberline{\thechapter}#1}%
    \else
      \addcontentsline{toc}{chapter}{#1}%
    \fi
  \else
    \addcontentsline{toc}{chapter}{#1}%
  \fi
  \chaptermark{#1}%
  \addtocontents{lof}{\protect\addvspace{10\p@}}%
  \addtocontents{lot}{\protect\addvspace{10\p@}}%
  \if@twocolumn
    \@topnewpage[\@makechapterhead{#2}]%
  \else
    \@makechapterhead{#2}%
  \afterheading
\fi}
```

Chaque entrée de la table des matières entraîne en fait un appel de `\@dottedtocline` dont la syntaxe est :

```
\@dottedtocline{<level>}{<indent>}{<numwidth>}{<text>}{<page>}
```

Les arguments 'text' et 'page' sont ceux de `\contentsline`. Les trois premiers arguments sont définis pour chaque commande de sectionnement, via une commande préfixée par `\l@` :

```
\newcommand*\l@subsection{%  
    \@dottedtocline{2}{3.8em}{3.2em}}
```

- Le niveau permet éventuellement de contrôler le nombre d'entrées affichées via le compteur 'tocdepth'.
- L'indentation est celle depuis la marge gauche.
- 'numwidth' est l'espace disponible pour le numéro de la section.

La largeur disponible pour le numéro de page est contrôlée par la valeur de `\@pnumwidth` défini ainsi dans `book.cls` :

```
\newcommand\@pnumwidth{1.55em}
```


Ajout d'une entrée après un `\chapter*`

Ces commandes ne génèrent pas d'entrée dans la table des matières. On peut y remédier de plusieurs manières :

- Redéfinir `\@schapter`

- ajouter

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

Le *package* `minitoc` permet d'avoir des mini-tables par chapitres, par parties, ... La documentation est disponible sur le LN.

15.3. Index

Un index peut être créé via un programme auxiliaire tel que `makeindex`. Des commandes `\index` sont placées dans le document et elle conduisent à l'écriture de lignes dans le fichier `.idx`. Exemple :

```
...Donald Knuth\index{Knuth!Donald}
```

conduit dans le fichier `.idx` à :

```
\indexentry{Knuth!Donald}{17}
```

Ensuite, `makeindex` prend ce fichier et produit un fichier `.ind` :

```
\begin{theindex}
```

```
  \item Knuth
```

```
    \subitem Donald, 17
```

```
\end{theindex}
```

Ce fichier est inséré dans le document principal au moyen d'une commande `\printindex`. Il faudra auparavant avoir chargé le *package* `makeidx` et indiqué `\makeindex` dans le préambule.

Les index peuvent être paramétrés au moyen de styles d'index.

15.4. Bibliographie

Les références bibliographiques peuvent être rassemblées dans un fichier `.bib` dont voici un exemple :

```
@BOOK{code-typographique93,  
  KEY = {CT},  
  TITLE = {{Code typographique. Choix de r\`e}gles \`{a}  
    l'usage des professionnels du livre}},  
  PUBLISHER = {F\`{e}d\`{e}ration C.G.C. de la Communication},  
  NOTE = {17\ieme \`{e}dition},  
  ADDRESS = {Paris},  
  YEAR = 1993,  
  MONTH = feb}  
  
@ARTICLE{desarmenien86,  
  AUTHOR = {Jacques D\`{e}sarmen\`{e}nien},  
  TITLE = {{La division par ordinateur des mots fran\c{c}ais~:  
    application \`{a} \TeX{}}},  
  JOURNAL = {Technique et Sciences Informatiques},  
  VOLUME = {5},  
  NUMBER = {4},  
  PAGES = {251--265},  
  YEAR = 1986}
```

La base de références `base.bib` est indiquée dans le fichier \LaTeX avec

```
\bibliographystyle{plain}
\bibliography{base.bib}
```

Les références sont citées avec `\cite{desarmenien86}`, etc.

L'ensemble des références est formaté par le programme auxiliaire \BIBTeX qui produit un fichier `.bbl` utilisé par \LaTeX . Exemple de contenu :

```
\begin{thebibliography}{}
...
\bibitem[]{}{desarmenien86}
Jacques D{\'}{e}sarm{\'}{e}nien.
\newblock {La division par ordinateur des mots
           fran\c{c}ais~: application \'{a} \TeX{}}.
\newblock \emph{Technique et Sciences Informatiques},
           5(4):251--265, 1986.
...
\end{thebibliography}
```

L'environnement `thebibliography` peut bien sûr être adapté.

16. Utilisation de fichiers externes

Écriture dans des fichiers pour gérer plusieurs passes. Exemple : notes marginales du bon côté.

17. Objets flottants

Paramètres de placement

Le placement des figures obéit à des contraintes régies par les paramètres suivants :

- `topnumber` : nombre maximal de flottants au haut d'une page ;
- `bottomnumber` : nombre maximal de flottants au bas d'une page ;
- `totalnumber` : nombre maximal de flottants sur une page
- `\topfraction` : fraction maximale de la page occupée par des flottants en haut de page ;
- `\bottomfraction` : fraction maximale de la page occupée par des flottants en bas de page ;
- `\textfraction` : fraction minimale de la page devant être occupée par du texte ;
- `\floatpagefraction` : fraction minimale d'une page de flottants devant être occupée par des flottants ;
- `dbltopnumber` : analogue à `topnumber` pour des pages en deux colonnes ;
- `\dbltopfraction` : analogue à `\topfraction` pour des pages en deux colonnes ;
- `\dblfloatpagefraction` : analogue à `\floatpagefraction` pour des pages de flottants en deux colonnes ;

Les valeurs par défaut des paramètres permettant le positionnement des figures sont définies dans `latex.ltx` :

```
\setcounter{topnumber}{2}
\setcounter{bottomnumber}{1}
\setcounter{totalnumber}{3}
\newcommand\topfraction{.7}
\newcommand\bottomfraction{.3}
\newcommand\textfraction{.2}
\newcommand\floatpagefraction{.5}
\setcounter{dbltopnumber}{2}
\newcommand\dbltopfraction{.7}
\newcommand\dblfloatpagefraction{.5}
```

Des valeurs moins contraignantes sont par exemple :

```
\renewcommand{\topfraction}{.8}
```

```
\renewcommand{\bottomfraction}{.8}
```

```
\renewcommand{\textfraction}{.2}
```

```
\renewcommand{\floatpagefraction}{.8}
```

Il y a au maximum 18 figures en attente à un moment donné.

Contrôle amélioré du positionnement

`\suppressfloats`

La commande `\suppressfloats` permet d'interdire des objets flottants d'un certain type sur une page donnée.

Si l'on ne souhaite pas de flottant au haut d'une page sur laquelle se trouve le début d'une section, on peut (re)définir `\section` de la manière suivante :

```
\newcommand{\section}{%  
    \suppressfloats[t]%  
    \@startsection{section}{...}{...}{...}...  
}
```

Emploi de ‘!’

L’emploi de ‘!’ en conjonction avec les paramètres `h`, `t` ou `b` relache les contraintes de placement.

Package `afterpage`

```
\afterpage{\clearpage}
```

Package `flafter`

Ce *package* a pour effet d’éviter qu’un flottant n’apparaisse avant l’endroit où il est défini dans le texte.

Package `float`

Ce *package* permet de définir de nouveaux styles de flottants. Il est documenté sur le LN.

18. Quelques notions de programmation avancée

18.1. Commandes robustes et fragiles

Problème : certaines commandes sont employées dans des conditions telles qu'elles peuvent être évaluées prématurément.

On remédie à cela en précédant la commande de `\protect` ou en déclarant une nouvelle commande avec `\DeclareRobustCommand`.

18.2. Résolution d'interactions entre *packages*

Il arrive qu'un *package* soit incompatible avec un autre *package* et que le problème n'est le fait que de quelques commandes qui ne devraient par exemple pas être redéfinies.

On peut dans ce cas sauvegarder l'ancienne valeur d'une commande le temps de l'exécution de la nouvelle.

```
\let\olddef=\cmd
\def\cmd{nouvelle definition}
...
\let\cmd=\olddef % on revient a l'ancien etat
```

18.3. Caractères actifs

Tous les caractères apparaissant dans un fichier T_EX reçoivent un code. Il y a 16 codes différents, de 0 à 15. Le code pour les lettres est 11. Celui pour la plupart des autres caractères est 12. Par défaut, les caractères ont le code que l'on attend d'eux ; par exemple, la lettre 'a' a par défaut le code 11. Mais il est possible de changer ce code et ceci peut perturber le fonctionnement de certaines commandes.

```
x % code 11
\catcode`\x=13
x % code 13
\catcode`\x=11
x % code 11
```

Typiquement, si le code de 'a' est changé, il n'est plus possible d'utiliser une commande comportant cette lettre (comme `\catcode`) puisque...ce n'est plus une lettre dans le sens du code 11.

Ainsi, les commandes bien connues `\makeatletter` et `\makeatother` sont équivalentes à :

```
\catcode`\@=12 % @ devient 'autre' et
                % ne peut plus etre utilise'
                % dans une commande
```

et

```
\catcode`\@=11 % @ devient une lettre et
                % peut etre utilise' dans une commande
```

18.4. Compteurs

Le compteur 'a' est représenté en \LaTeX par la variable entière `\c@{a}`.

- `\newcounter`
- `\setcounter`
- `\addtocounter`

19. Les messages d'erreur

Comment réagir face à un message d'erreur ?

Lire tout le message, localiser le lieu et la cause de l'erreur.

En général, c'est simple.

Si le contexte est insuffisant, augmenter `\errorcontextlines` (ou le compteur 'errorcontextlines').

Pour voir le déroulement des macros, faire `\tracingmacros=1` puis (plus loin) `\tracingmacros=0`.

20. Exemple de thloria

21. Problèmes de mémoire

À la fin de l'exécution de \LaTeX , on a des statistiques (dans le fichier `.log`) comme :

```
9830 strings out of 13690
133345 string characters out of 476858
150632 words of memory out of 263001
10000 multiletter control sequences out of 10000+0
7 words of font info for 0 fonts, out of 30000 for 1000
14 hyphenation exceptions out of 1000
37i,1n,57p,202b,63s stack positions out of 300i,100n,500p,300000b,4000s
```

On obtient quelquefois un dépassement de capacité. Souvent, le problème est une erreur dans le fichier source qui entraîne une boucle infinie. Il y a cependant des cas où il faut réellement augmenter les capacités de \TeX . On modifiera dans ce cas en général le fichier `texmf.cnf` de manière adéquate. (Cf. guide local du LORIA.)

22. Comprendre les documents qui sont soumis

- Éventuellement les transformer.
- Utiliser $\text{T}_\text{E}^\text{X}$ au lieu de $\text{L}^\text{A}^\text{T}_\text{E}^\text{X}$?

23. Travail sur un livre

Le troisième jour.

Chacun prend un livre (avec une unique colonne, pas de magazine,...) et essaie de l'analyser et de le reproduire.

Tous les problèmes ne seront pas nécessairement résolus, mais nous allons les étudier ensemble.

24. Guides pour l'écriture de classes

- Lire du code.
- Utiliser les fichiers `*guide.tex`.

25. Suggestion pour aller plus loin

- Lire le `TEXbook`
- Lire des sources de *packages*, classes, etc.