

L^AT_EX—*Débutant*¹

Emmanuel Paradis

11 septembre 2009

Table des matières

1	T_EX et L^AT_EX	2
2	Pourquoi utiliser L^AT_EX ?	3
3	Installer L^AT_EX	4
4	La compilation et les formats de fichier	4
5	Le minimum pour démarrer avec L^AT_EX	5
6	Le préambule	6
6.1	Les classes de document	7
6.2	Les « packages »	8
6.3	babel	8
7	Préparer un document L^AT_EX avec Emacs	9

¹© 2009 Emmanuel Paradis

Permission est accordée de copier et distribuer ce document, en partie ou en totalité, dans n'importe quelle langue, sur n'importe quel support, à condition que la notice © ci-dessus soit incluse dans toutes les copies. Permission est accordée de traduire ce document, en partie ou en totalité, dans n'importe quelle langue, à condition que la notice © ci-dessus soit incluse.

8	Le texte	10
8.1	Les polices de caractère	11
8.2	Contrôler les espaces	13
8.3	Contrôler les césures	15
8.4	Les accents	15
8.5	Caractères spéciaux	16
9	Le sectionnage	16
10	Les environnements	18
10.1	Les environnements <code>itemize</code> , <code>enumerate</code> , et <code>description</code> . .	18
10.2	L’environnement <code>tabular</code>	19
10.3	L’environnement <code>verbatim</code>	21
11	Expressions mathématiques	22
11.1	Principales notations mathématiques	23
11.2	Parenthèses et autres délimiteurs	24
11.3	Lettres grecques	25
11.4	Contrôle des espaces en mode math	25

1 \TeX et \LaTeX

\TeX : un langage informatique créé par Donald Knuth¹ pour la « création de beaux livres—en particulier de livres contenant beaucoup de mathématiques ».

\LaTeX : un ensemble de macros en \TeX , écrites originellement par Leslie Lamport afin de permettre une « balance entre la fonctionnalité et la facilité d’utilisation ». \LaTeX possède des classes de documents prédéfinies (articles, diaporamas, lettres, livres, . . .), ainsi qu’un grand nombre de « packages » qui rendent aisée l’accession aux fonctionnalités de \TeX . \LaTeX conserve les particularités du langage \TeX (compteurs, variables, . . .).

Autres systèmes dérivés de \TeX : \AMS-TeX , $\text{\AMS-}\text{\LaTeX}$, \ConTeXt , \TeXinfo , \Eplain , \Lollipop .

Les successeurs de \TeX : \Omega (Lambda), \e-TeX , \NTS .

¹www-cs-faculty.stanford.edu/~knuth/

2 Pourquoi utiliser \LaTeX ?

POUR

1. \LaTeX est disponible pour tous les systèmes d'exploitation.
2. Les documents produits par \LaTeX sont système-indépendants.
3. De nombreuses distributions gratuites de \LaTeX sont disponibles.
4. Les documents produits par \LaTeX sont de très bonne qualité
5. \LaTeX peut produire des documents au format PDF (Portable Document Format).
6. De nombreux types de documents peuvent être composés (livres, articles, rapports, thèses, lettres, diaporamas, partitions musicales, ...).
7. Les expressions mathématiques sont imprimées avec une excellente qualité.
8. Les définitions par défaut des formats aboutissent à des documents d'excellente qualité (en termes typographiques) avec un minimum d'apprentissage.
9. Il existe de nombreuses fonctionnalités de base (compteurs, références croisées, ...) très pratique dans la rédaction d'un document (par ex. pour la numérotation des tableaux et des figures).
10. Appréhender un document dans sa globalité, plutôt que de se focaliser sur l'aspect esthétique à l'écran, permet de meilleurs résultats quant à sa structuration et sa présentation.
11. Les formats et autres fonctionnalités (compteurs, environnements, ...) peuvent être modifiés (relativement) aisément.
12. Il existe un très grand nombre de « packages » qui étendent largement les possibilités de \LaTeX .
13. Des « environnements linguistiques » très variés sont disponibles (langues et alphabets).
14. De nombreuses maisons d'édition scientifique proposent des fichiers \LaTeX avec des formats prédéfinis (*templates*).

CONTRE

1. Il n'est pas possible de commencer un document \LaTeX sans un minimum d'apprentissage.

2. L^AT_EX n'est pas « WYSIWYG » (*what you see is what you get*).²
3. La maîtrise de L^AT_EX nécessite un apprentissage progressif.

3 Installer L^AT_EX

L'installation de L^AT_EX est bien sûr système-dépendante.

Sous Windows, on recommandera la distribution MiK_TE_X³ dont l'installation est très facile grâce à l'interface graphique. MiK_TE_X inclut de nombreux exécutables et utilitaires (compilateurs, convertisseurs de formats, ...).

L^AT_EX est disponible sur toute distribution décente de Linux. Mac_TE_X est un système T_EX complet pour MacOS X.⁴

4 La compilation et les formats de fichier

Un fichier L^AT_EX est au format ASCII et inclut le texte ainsi que les déclarations relatives à son formatage. Il est nommé avec l'extension `.tex`.

Le fichier L^AT_EX est compilé par T_EX pour produire un fichier DVI (*device independent*) nommé avec l'extension `.dvi`. Ce fichier peut être visualisé avec un logiciel approprié (*viewer*).

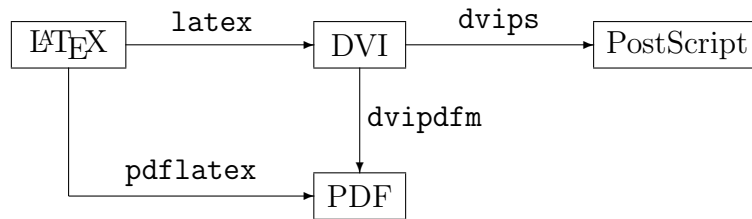
La compilation produit deux autres fichiers : `.log` qui contient un rapport de la compilation et les éventuels messages d'erreur, et `.aux` (*auxiliary*) qui est utilisé par L^AT_EX.

Les distributions de L^AT_EX incluent généralement un ensemble d'exécutables pour convertir les formats de fichier.

²Il existe des interfaces graphiques pour L^AT_EX notamment sous Windows, la plupart commerciales.

³www.miktex.org

⁴www.tug.org/mactex



Ces exécutables s'utilisent en ligne de commande. Par exemple, le présent document a été produit avec :

```
pdflatex LaTeX_debutant_E.Paradis
```

Il conviendra bien sûr d'effectuer les configurations locales nécessaires pour que l'exécutable soit accessible, par exemple avec :

```
c:\progra~1\miktex\miktex\bin\pdflatex LaTeX_debutant_E.Paradis
```

Il est possible de mettre ces commandes dans un fichier `.bat` (pour Windows) surtout s'il y en a plusieurs comme dans la première chaîne de conversion illustrée ci-dessus :

```
c:\progra~1\miktex\miktex\bin\latex LaTeX_debutant_E.Paradis.tex
c:\progra~1\miktex\miktex\bin\dvips LaTeX_debutant_E.Paradis.dvi
```

L'utilisation d'un format ou d'un autre dépend des besoins. `pdflatex` est plus direct pour créer des fichiers PDF à partir de document `LATEX` mais est incompatible avec certains packages qui nécessitent de passer par le format postscript (comme `PSTricks`). Par contre, `pdflatex` est requis par d'autres packages (comme `PPower4` pour créer des diaporamas).

5 Le minimum pour démarrer avec `LATEX`

Pour démarrer avec `LATEX` il faut :

- définir une classe de document avec `\documentclass{...}`,

- déclarer le texte avec les commandes `\begin{document}` et `\end{document}`.

Un document \LaTeX minimaliste ressemblera donc à ceci :

```
\documentclass{article}
\begin{document}
Bonjour, ça va?
\end{document}
```

Un tel document peut être compilé par \LaTeX pour produire, par exemple, un fichier DVI.

Toutefois, en français il est plus qu'utile d'utiliser le package `inputenc` avec l'option `latin1` qui précise à \LaTeX que les caractères utilisés sur un clavier AZERTY font partie du code de caractères ISO-Latin 1. Par défaut, \LaTeX considère que les caractères viennent du code ASCII et ne reconnaît donc pas é, è, ù, ç, etc. Notre document devient donc :

```
\documentclass{article}
\usepackage[latin1]{inputenc}
\begin{document}
Bonjour, ça va?
\end{document}
```

6 Le préambule

Le préambule d'un document \LaTeX est l'ensemble des commandes qui précèdent la déclaration `\begin{document}`. Il contient notamment :

- la définition de la classe du document avec les éventuelles options ;
- la déclaration des packages utilisés dans le document avec les options si besoin ;
- toute (re)définition de commandes.

Le préambule contient également la définition du titre et de(s) l'auteur(s) du document :

```
\title{...}
\author{...}
```

Un préambule peut être très long (c'est le cas des *templates*).

6.1 Les classes de document

Il existe par défaut cinq classes de document L^AT_EX :

- `article`
- `report`
- `letter`
- `book`
- `slides`

La déclaration se fait avec :

```
\documentclass[options]{class}
```

où `options` spécifie les options séparées par des virgules. La liste suivante donne les options courantes avec les valeurs possibles (la lettre `d` entre parenthèses indique la valeur par défaut).

- `a4paper`, `a5paper`, `b5paper`, `letterpaper` (`d`), `legalpaper`, `executivepaper` : taille du papier
- `landscape` : sélectionne le format paysage
- `titlepage`, `notitlepage` (`d`) : précise s'il doit y avoir une page de titre séparée
- `fleqn` : équations alignées à gauche (centrées par défaut)
- `openbib` : utiliser format de bibliographie `open`
- `draft`, `final` (`d`) : si `draft` le texte qui déborde dans les marges est indiqué par une bande noire

Les options suivantes ne marchent pas avec `slides` :

- `10pt` (`d`), `11pt`, `12pt` : taille des caractères de base
- `onecolumn` (`d`), `twocolumn` : nombres de colonnes
- `oneside` (`d`), `twoside` (`d` pour `book`) : recto simple ou recto-verso
- `openright` (`d`), `openany` : détermine si un chapitre doit commencer sur une page droite

La classe `slides` a également l'option `clock` pour imprimer l'heure au bas de chaque diapo.

6.2 Les « packages »

Les packages nécessaires sont chargés dans le préambule avec la commande `\usepackage[options]{package}`.

Une installation de \LaTeX comporte plusieurs packages dont on peut trouver une description dans (avec \MiKTeX) :

```
C:\Program Files\MiKTeX\doc\latex\base\usrguide.dvi
```

L'installation de nouveaux packages se fait, pour \MiKTeX , via le « \MiKTeX Package Manager ».

Chaque package a une fonction particulière. On a déjà vu `inputenc`. On verra d'autres exemples dans les paragraphes suivants. Le package `babel` a des fonctionnalités spéciales que l'on va détailler dans le paragraphe suivant.

6.3 babel

`babel` fournit des fonctionnalités pour le formatage du texte en fonction de la langue. Ceci concerne la typographie générale et la césure des mots (*hyphenation* en anglais).

Note : `babel` n'est pas un correcteur d'orthographe.

Les langues utilisées dans un document doivent être déclarées comme option du document ou comme option du package. Dans le premier cas, les options sont globales et donc disponibles à d'autres packages.

La commande principale de `babel` est `\selectlanguage` qui choisit la langue dans laquelle sera formaté le texte qui suit.

Le préambule de ce document est :

```
\documentclass[12pt,english,french]{article}
\usepackage[latin1]{inputenc}
\usepackage{babel}
\selectlanguage{french}
```

La commande `\selectlanguage` peut intervenir dans le texte.

La documentation détaillée de `babel` se trouve dans (toujours avec \MiKTeX) :

C:\Program Files\MiKTeX\doc\generic\Babel\babel.dvi

Ce paragraphe a été composé le 9 février 2007 : il a pour but d'illustrer certaines capacités de Babel. Dans la phrase précédente, j'ai utilisé la commande `\today` pour écrire la date. \LaTeX a beaucoup de commandes similaires !

Le même paragraphe précédé de la commande `\selectlanguage{english}` donnera le résultat suivant :

Ce paragraphe a été composé le February 9, 2007: il a pour but d'illustrer certaines capacités de Babel. Dans la phrase précédente, j'ai utilisé la commande `\today` pour écrire la date. \LaTeX a beaucoup de commandes similaires!

Il y a six différences entre ces deux paragraphes : à vous de les trouver.

7 Préparer un document \LaTeX avec Emacs

Emacs est un éditeur de texte très complet, gratuit, et disponible sous de nombreux systèmes d'exploitation.

<http://www.gnu.org/software/emacs/>

Sous Windows, on utilisera NTEmacs :

<http://www.gnu.org/software/emacs/windows/ntemacs.html>

Emacs est livré avec un package nommé RefTeX qui facilite l'édition de documents \LaTeX . On installera également le package AUCTeX qui offre des fonctionnalités supplémentaires :

<http://www.gnu.org/software/auctex/>

Ces packages doivent être chargés en mémoire au démarrage d'Emacs. On va donc configurer le fichier `.emacs`. Voici un extrait de mon fichier `.emacs` avec quelques customizations utiles (les points-virgules indiquent des commentaires) :

```
(show-paren-mode 1) ; parentheses highlighting
(global-font-lock-mode 1) ; syntax highlighting
(setq font-lock-maximum-decoration t) ; have it maximum!!
(setq frame-title-format "%b") ; put the name of the current
                                ; buffer as the frame name
(setq column-number-mode t) ; display column number
;;; RefTeX

(add-hook 'LaTeX-mode-hook 'turn-on-reftex) ;with AUCTeX LaTeX mode
(add-hook 'latex-mode-hook 'turn-on-reftex) ;with Emacs latex mode

;; -----
;; AUC TeX

(add-to-list 'load-path "D:/auctex-10.0g/")
(add-to-list 'Info-default-directory-list "D:/auctex-10.0g/doc/")
(load "tex-site")
(require 'tex-site)
```

Le fichier `.emacs` doit se trouver dans le répertoire HOME.

Autres programmes utiles : GhostScript et GSview pour convertir les formats PS, PDF, et autres, et visualiser ces fichiers :

<http://www.cs.wisc.edu/~ghost/>

8 Le texte

Dans la grande majorité des cas, le texte commencera par la commande :

```
\maketitle
```

qui imprimera le titre, l'auteur et la date. Si l'on ne veut pas imprimer la date sous le titre, celle-ci peut être redéfinie dans le préambule avec `\date{}`.

L^AT_EX va interpréter le texte compris entre les commandes `\begin{document}` et `\end{document}` comme une suite de commandes qui vont servir à imprimer le document final (DVI ou PDF). Il existe quatre types de commandes :

commande	signification pour L ^A T _E X
un caractère	imprimer ce caractère
espace(s) ou retour-chariot	insérer un espace
ligne blanche	nouveau paragraphe
<code>\commande</code>	une commande

Les trois premiers types de commandes sont exécutées en fonction de paramètres pré-définis ou définis par l'utilisateur.

Les deux textes suivants seront traités de la même façon par L^AT_EX :

Pour LaTeX, il peut y avoir un ou plusieurs espaces entre les mots.

Pour LaTeX, il peut y avoir
un ou plusieurs
espaces entre les mots.

8.1 Les polices de caractère

Par défaut, L^AT_EX utilise les polices *Computer Modern Fonts* (CMF) conçues par D. Knuth. Trois polices sont utilisables : Roman, Sans Serif et Typewriter.

Ces polices existent sous cinq formes :

- droit
- italique
- gras
- penché (*slanted*)
- petites capitales (*small caps*)

Il est possible d'utiliser d'autres polices via un package approprié, par exemple `\usepackage{times}` pour utiliser les polices d'Adobe (Times en place du Roman, Helvetica en place du Sans Serif, Courier en place du Typewriter).

Le contrôle des polices et des formes se fait via deux types de commandes (par exemple pour obtenir du texte en italique) :

- `\textit{...}` ou `\it`
- `{\it ...}`

Le premier type de commandes se cumulera éventuellement avec des définitions préalables, par exemple pour donner du gras italique ; le second type de commandes isolera la définition et le texte sera en italique simple.

<code>\rm</code>	<code>\textrm{}</code>	Roman
<code>\sf</code>	<code>\textsf{}</code>	Sans Serif
<code>\tt</code>	<code>\texttt{}</code>	Typewriter
<code>\it</code>	<code>\textit{}</code>	<i>italique</i>
<code>\bf</code>	<code>\textbf{}</code>	gras
<code>\sl</code>	<code>\textsl{}</code>	<i>penché (≠ italique)</i>
<code>\sc</code>	<code>\textsc{}</code>	PETITES CAPITALES
<code>\up</code>	<code>\textup{}</code>	droit

Les combinaisons de ces commandes ne marchent pas toujours. Cela dépend des polices installées. Voici quelques exemples qui marchent avec une installation de base :

<code>\it\bfseries</code>	<i>gras italique</i>
<code>\sf\bfseries</code>	Sans Serif gras
<code>\sf\textit{}</code>	<i>Sans Serif italique</i>
<code>\sl\bfseries</code>	<i>gras penché</i>
<code>\tt\textit{}</code>	<i>Typewriter italique</i>
<code>\tt\textsl{}</code>	<i>Typewriter penché</i>

À signaler enfin, les commandes pour insister sur du texte, `\textem{}` et `\em`, qui par défaut impriment le texte en italique.

Le contrôle de la taille des caractères se fait par des commandes qui modifient celle-ci par rapport à la taille de base définie dans la classe du document (10pt par défaut). Le tableau suivant donne les tailles absolues en points pour chaque taille de base possible.

commande	10 pt	11 pt	12 pt
<code>\tiny</code>	5	6	6
<code>\scriptsize</code>	7	8	8
<code>\footnotesize</code>	8	9	10
<code>\small</code>	9	10	11
<code>\normalsize</code>	10	11	12
<code>\large</code>	12	12	14
<code>\Large</code>	14	14	17
<code>\LARGE</code>	17	17	20
<code>\huge</code>	21	20	25
<code>\Huge</code>	25	25	25

Enfin, pour souligner du texte on utilisera la commande spéciale `\underline`.

Il est rare de voir du `\underline{texte souligné}` dans un document imprimé.

Il est rare de voir du texte souligné dans un document imprimé.

D'autres effets (double-souligné, barré, souligné ondulé, ...) sont disponibles également avec d'autres packages.

Plus de détails sur les polices (contrôles, définitions, ...) peuvent être trouvées dans :

C:\Program Files\MiKTeX\doc\latex\base\fntguide.dvi

8.2 Contrôler les espaces

L^AT_EX contrôle les espaces entre les mots en fonction :

- des paramètres en vigueur (format, langue sélectionnée, ...),
- de certains caractères.

Par défaut, le texte est justifié à gauche et à droite. L^AT_EX va donc répartir l'espace entre les mots pour que les lignes aient la même largeur.

Bien sûr, certains espaces entre mots sont remplacés par des sauts de ligne.

Pour rendre un espace insécable, il peut être remplacé par un tilde qui liera les deux mots. Il est parfois préférable d'éviter de séparer deux mots, en voici quelques exemples :

Dr.~Jekyll Fig.~1 Tableau~2 D.~Knuth

Les espaces qui suivent une commande ne sont pas pris en compte par \LaTeX sauf si cette commande comporte des accolades. Pour éviter ceci, il y a trois possibilités :

```
\LaTeX mange les espaces.    \LaTeXmange les espaces.
\LaTeX{} mange les espaces.    \LaTeX{}mange les espaces.
\LaTeX\ mange les espaces.    \LaTeX\mange les espaces.
{\LaTeX} mange les espaces.    {\LaTeX}mange les espaces.
```

La commande ‘\ ’ (antislash suivi d’un blanc) rajoute explicitement un espace mais celui-ci est sécable (au contraire du tilde).

En français, un espace est ajouté avant les ponctuations dites doubles (composées de deux caractères), même si cet espace n’a pas été tapé. La taille de cet espace est ajusté par \LaTeX si besoin, mais il est insécable.

Pour annuler cet espace, on utilisera la commande `\string` juste avant la ponctuation en question :

```
http://www...                    http://www...
http\string://www...    http://www...
```

En anglais, pour justifier un texte l’espace est ajouté de façon inégale entre les mots et les signes de ponctuation (*cf.* l’exemple ci-dessus). Notamment, l’espace est plus grand après un point de fin de phrase qu’après une virgule. Cependant, par défaut \LaTeX considère que si une majuscule précède un point, celui-ci indique une abréviation et l’espace qui suit est égal à celui qui suit un mot ou une virgule.

Il est possible d’altérer ce comportement, dans un sens comme dans l’autre :

```
We programed this in C\@. Denis et al.\ got similar results.
```

8.3 Contrôler les césures

La césure des mots, qui est effectuée par défaut avec L^AT_EX, facilite la justification du texte. Elle dépend de la langue sélectionnée. Cela ne pose aucun problème en anglais, mais en français (et d'autres langues), il est nécessaire de changer l'encodage des polices pour que L^AT_EX puisse couper les mots correctement. Cela se fait avec le package `fontenc` et l'option `T1` (à placer dans le préambule bien sûr) :

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

Si Babel ne trouve pas où couper certains mots, il est possible de lui indiquer explicitement les césures possibles :

La gé\ -no\ -mi\ -que est en plein déve\ -lop\ -pe\ -ment.

On peut aussi empêcher un mot d'être coupé :

```
\LaTeX\ n'est pas un système \mbox{WYSIWYG}.
```

Pour prévenir toute césure dans le document, on ajoutera les commandes suivantes dans le préambule (le texte restera justifié à droite) :

```
\pretolerance=2000  
\tolerance=3000
```

Le fait d'annuler la justification à droite généralement annule les césures.

8.4 Les accents

La commande `antislash` suivi d'un accent signifie « ajoute cet accent sur la lettre qui suit ».

<code>\`e</code>	<code>\`n</code>	<code>\`A</code>	<code>\`p</code>	<code>\`{}</code>	`
<code>\'e</code>	<code>\'n</code>	<code>\'A</code>	<code>\'p</code>	<code>\'{} </code>	´
<code>\~e</code>	<code>\~n</code>	<code>\~A</code>	<code>\~p</code>	<code>\~{} </code>	˜
<code>\"e</code>	<code>\"n</code>	<code>\"A</code>	<code>\"p</code>	<code>\"{} </code>	¨
<code>\~e</code>	<code>\~n</code>	<code>\~A</code>	<code>\~p</code>	<code>\~{} </code>	˜

On notera bien que le tréma (¨) est obtenu avec des guillemets (*double quotes*) et non pas la touche correspondante sur les claviers AZERTY.

Rappelons que grâce au package `inputenc`, ë à la même signification que `\"e`.

8.5 Caractères spéciaux

On a vu que certains caractères sont utilisés dans la syntaxe de \LaTeX . Ils peuvent être imprimés en étant eux-mêmes précédés d'un antislash. Une exception concerne l'antislash lui-même puisque le double antislash a une signification spéciale.

<code>\textbackslash</code>	<code>\</code>	<code>\{</code>	<code>{</code>	<code>\}</code>	<code>}</code>
<code>\&</code>	<code>&</code>	<code>\#</code>	<code>#</code>	<code>\\$</code>	<code>\$</code>
<code>_</code>	<code>-</code>	<code>\%</code>	<code>%</code>		

On signalera ces autres caractères fréquemment utilisés.

<code>\guillemotleft</code>	«	-	-	<code>\oe</code>	œ
<code>\guillemotright</code>	»	--	—	<code>\OE</code>	Œ
<code>\textdagger</code>	†	---	---	<code>\ae</code>	æ
<code>\textcopyright</code>	©	“	“	<code>\AE</code>	Æ
<code>\textregistered</code>	®	”	”	<code>\O</code>	Ø
<code>\textbullet</code>	•	‘	‘	<code>\o</code>	ø
<code>\dots</code>	...	’	’	<code>\c C</code>	Ç

En pratique, on préférera l'utilisation des commandes `\og` et `\fg` pour les guillemets.

9 Le sectionnage

Un document \LaTeX peut être sectionné en trois niveaux hiérarchiques :

- sections
- sous-sections
- sous-sous-sections

Les commandes correspondantes sont :

```
\section{...}
  \subsection{...}
    \subsubsection{...}
```

L^AT_EX numérote ces sections avec des chiffres arabes et imprime les titres avec une taille de police proportionnelle au niveau. Le numérotage se fait de la façon suivante :

- sections : 1, 2, ...
- sous-sections : 1.1, 1.2, ..., 2.1, 2.2, ...
- sous-sous-sections : 1.1.1, 1.1.2, ... 1.2.1, 1.2.2, ..., 2.1.1, 2.1.2, ...

Les versions étoilées de ces commandes produisent les mêmes niveaux de sectionnage mais sans numéro :

```
\section*{...}
  \subsection*{...}
    \subsubsection*{...}
```

Les niveaux paragraphes et sous-paragraphes ne sont pas numérotés ; le titre est juste imprimé en gras :

```
\paragraph{...}
  \subparagraph{...}
```

Pour les documents de classe `book` et `report`, deux autres niveaux sont possibles :

```
\part{...}
  \chapter{...}
```

Les parties regroupent des chapitres et sont numérotées avec des chiffres romains (I, II, ...). Les chapitres sont numérotés avec des chiffres arabes.

Quelque que soit la classe du document la table des matières peut être imprimée avec la commande :

`\tableofcontents`

L^AT_EX produira alors un fichier `.toc` avec la liste des têtes de section. Il sera éventuellement nécessaire de compiler deux fois le fichier `.tex`. Les sections (et autres sous-sections) non numérotées ne seront pas incluses dans cette table des matières.

10 Les environnements

Un *environnement* est une partie du document délimitée par les commandes `\begin{...}` et `\end{...}` entre lesquelles les paramètres de formatage sont modifiés. Il existe de nombreux environnements préprogrammés dans L^AT_EX.

L'utilisation d'Emacs facilite grandement l'accès aux environnements via le menu "LaTeX".

Les environnements ont des fonctions variées : certains modifient juste les marges (`quote`), le format simplement (`center`, `flushright`, `flushleft`), ou crée un environnement différent avec des commandes spécifiques (`picture`).

Dans la section suivante, on verra les environnements `equation` et `displaymath`.

10.1 Les environnements `itemize`, `enumerate`, et `description`

Ces trois environnements ont pour objet la présentation de listes d'articles *item*. Les environnements `itemize` et `enumerate` ont la même syntaxe :

```
\begin{...}
\item article 1
\item article 2
...
\end{...}
```

Il peut n'y avoir qu'un seul `\item`. La présentation est différente en fonction de l'environnement :

- article 1
- article 2

1. article 1
2. article 2

À signaler que le formatage de `itemize` dépend de la langue sélectionnée. La même liste formatée en anglais donnera :

- article 1
- article 2

De même pour `enumerate`, les articles sont légèrement plus espacés en anglais. Pour l'environnement `description`, un titre (optionel) entre crochets suit la commande `\item` :

```
\begin{description}
\item[itemize] les articles sont listés simplement
\item[enumerate] les articles sont listés et numérotés
\item[description] les articles sont listés avec un titre en gras
\end{description}
```

itemize les articles sont listés simplement

enumerate les articles sont listés et numérotés

description les articles sont listés avec un titre en gras

10.2 L'environnement `tabular`

L'environnement `tabular` est très utilisé dans \LaTeX car il permet de composer des tableaux.

La déclaration `\begin{tabular}` contient un argument obligatoire qui définit le formatage des colonnes et qui est composé d'une série de lettres "c", "l" ou "r" pour, respectivement, centrer, justifier à gauche ou justifier à droite.

Les différentes colonnes sont séparées par le caractère `&`.

Chaque ligne doit être terminée par un saut de ligne explicite indiqué par un double antislash. Ce saut de ligne peut intervenir même si le nombre de colonnes définies n'est pas atteint.

```

\begin{tabular}{lrc}
Espèce & poids (g) & nom scientifique\\
Souris & 10 & {\it Mus musculus}\\
Baleine & 120~000~000 & {\it Balaenoptera musculus}\\
Homme & 70~000 & {\it Homo sapiens}\\
\end{tabular}

```

Espèce	poids (g)	nom scientifique
Souris	10	<i>Mus musculus</i>
Baleine	120 000 000	<i>Balaenoptera musculus</i>
Homme	70 000	<i>Homo sapiens</i>

Il est fréquent, et souvent plus esthétique, d’inclure un tableau au sein d’un environnement `center`.

Les lignes horizontales qui servent à délimiter différentes parties du tableau, par exemple les titres des colonnes, sont ajoutées avec la commande `\hline` :

```

\begin{center}
\begin{tabular}{lrc}
\hline
Espèce & poids (g) & nom scientifique\\
\hline
Souris & 10 & {\it Mus musculus}\\
Baleine & 120~000~000 & {\it Balaenoptera musculus}\\
Homme & 70~000 & {\it Homo sapiens}\\
\hline
\end{tabular}
\end{center}

```

Espèce	poids (g)	nom scientifique
Souris	10	<i>Mus musculus</i>
Baleine	120 000 000	<i>Balaenoptera musculus</i>
Homme	70 000	<i>Homo sapiens</i>

Les lignes verticales sont ajoutées en les définissant dans la déclaration du format de colonnes à l’aide du symbole ‘|’ :

la commande <code>\verb \LaTeX </code> , <code>\dots</code>	la commande <code>\LaTeX</code> , ...
la commande <code>\verb+\LaTeX+</code> , <code>\dots</code>	la commande <code>\LaTeX</code> , ...
the command <code>\verb5\LaTeX5</code> , <code>\dots</code>	the command <code>\LaTeX</code> , ...

11 Expressions mathématiques

Il y a deux façons d'inclure des expressions mathématiques dans un document :

- au sein d'un paragraphe :
 - entre `\(` et `\)`
 - entre deux `$`
 - dans un environnement `math`
- sur des lignes séparées et centrées (par défaut) :
 - entre `\[` et `\]`
 - dans un environnement `displaymath`
 - dans un environnement `equation`

L'environnement `equation` numérote les équations séquentiellement.

Dans toutes ces situations, \LaTeX est dit en *mode math* (par opposition au *mode text*).

Le mode `math` possède ses propres commandes (qui ne marchent pas en mode `text`!) et ses propres règles de formatage.

1. Les lettres sont considérées comme des variables mathématiques et donc imprimées en italique. Pour obtenir du texte droit on utilisera les commandes `\textrm{}` ou `\mathrm{}`.
2. Les espaces tapés sont ignorés. \LaTeX ajoute automatiquement de l'espace autour des symboles mathématiques (`=`, `+`, `-`, `×`, ...). Les espaces peuvent être contrôlés avec des commandes spécifiques.
3. Les lignes blanches sont interdites.
4. Les caractères `^` et `_` servent à définir respectivement les exposants et les indices.

On fera attention que `^` et `_` ne prennent en compte que le premier caractère qui suit. Si on veut inclure plusieurs caractères dans l'exposant ou l'indice, on les groupera entre accolades.

`\x_{ij}` x_{ij} `\x_{ij}` x_{ij}

11.1 Principales notations mathématiques

<code>\sqrt{x}</code>	\sqrt{x}	<code>\overbrace{u+v+w}</code>	$\overbrace{u+v+w}$
<code>\sqrt[n]{x}</code>	$\sqrt[n]{x}$	<code>\overrightarrow{AB}</code>	\overrightarrow{AB}
<code>\surd(x^2 + y^2)</code>	$\sqrt{(x^2 + y^2)}$	<code>\overleftarrow{CDE}</code>	\overleftarrow{CDE}
<code>\overline{x+y}</code>	$\overline{x+y}$	<code>\underline{x+y+z}</code>	$\underline{x+y+z}$
<code>x', x''</code>	x', x''	<code>\underbrace{f+g}</code>	$\underbrace{f+g}$

<code>\bar{x}</code>	\bar{x}	<code>></code>	$>$	<code>\approx</code>	\approx
<code>\dot{x}</code>	\dot{x}	<code><</code>	$<$	<code>\neq</code>	\neq
<code>\ddot{x}</code>	\ddot{x}	<code>\le</code>	\leq	<code>\subset</code>	\subset
<code>\hat{r}</code>	\hat{r}	<code>\ge</code>	\geq	<code>\in</code>	\in
<code>\widehat{r}</code>	\widehat{r}	<code>\equiv</code>	\equiv	<code>\notin</code>	\notin
<code>\tilde{r}</code>	\tilde{r}	<code>\sim</code>	\sim	<code>\mid</code>	$ $
<code>\widetilde{r}</code>	\widetilde{r}	<code>\simeq</code>	\simeq	<code>\propto</code>	\propto

<code>\pm</code>	\pm	<code>\Leftarrow</code>	\Leftarrow	<code>\dots</code>	\dots
<code>\times</code>	\times	<code>\Rightarrow</code>	\Rightarrow	<code>\cdots</code>	\cdots
<code>\div</code>	\div	<code>\leftrightarrow</code>	\leftrightarrow	<code>\vdots</code>	\vdots
<code>\otimes</code>	\otimes	<code>\Leftrightarrow</code>	\Leftrightarrow	<code>\ddots</code>	\ddots
<code>\leftarrow</code>	\leftarrow	<code>\uparrow</code>	\uparrow	<code>\partial</code>	∂
<code>\rightarrow</code>	\rightarrow	<code>\downarrow</code>	\downarrow	<code>\infty</code>	∞

`{p \choose n}` $\binom{p}{n}$ `\frac{1}{x^2 + y^2}` $\frac{1}{x^2 + y^2}$

La plupart des fonctions mathématiques (qui doivent s'écrire en Roman par opposition aux variables) s'écrivent simplement précédées d'un antislash.

`\cos \sin \exp \ln \log \max \Pr \dots`

((<code>\langle</code>	<	<code>\uparrow</code>	↑
))	<code>\rangle</code>	>	<code>\downarrow</code>	↓
[[<code>\lfloor</code>	⌊	<code>\updownarrow</code>	↕
]]	<code>\rfloor</code>	⌋	<code>\Uparrow</code>	⇑
{	{	<code>\lceil</code>	⌈	<code>\Downarrow</code>	⇓
}	}	<code>\rceil</code>	⌉	<code>\Updownarrow</code>	↕
/	/	<code>\backslash</code>	\		

11.3 Lettres grecques

Les lettres grecques minuscules sont tapées avec une commande nommée d'après la lettre en question.

`\alpha` α `\beta` β `\chi` χ ...

Pour les lettres majuscules, deux cas de figures :

- la majuscule est identique à une lettre romaine (alpha, beta, epsilon, ...), dans ce cas on utilise la lettre en question (*A*, *B*, *E*, ...);
- dans le cas contraire, la commande est la même que pour la lettre minuscule mais avec la première lettre en majuscule.

Ci-dessous la liste des lettres qui tombent dans le second cas de figure.

<code>\Gamma</code>	Γ	<code>\Lambda</code>	Λ	<code>\Sigma</code>	Σ	<code>\Psi</code>	Ψ
<code>\Delta</code>	Δ	<code>\Xi</code>	Ξ	<code>\Upsilon</code>	Υ	<code>\Omega</code>	Ω
<code>\Theta</code>	Θ	<code>\Pi</code>	Π	<code>\Psi</code>	Ψ		

11.4 Contrôle des espaces en mode math

On a vu au début de cette section que les expressions mathématiques peuvent être situées au sein d'un paragraphe ou sur des lignes séparées. Pour les expressions qui prennent de la place (avec des intégrales par exemple), on utilisera de préférence la seconde solution. Si on tape de telles expressions dans un paragraphe, par exemple entre deux \$, le formatage sera plus compact.

la moyenne est `\frac{1}{n}\sum_{i=1}^n x_i`

la moyenne est $\frac{1}{n} \sum_{i=1}^n x_i$

On peut forcer de telles expressions à être imprimées de façon classique avec la commande `\displaystyle` qui précédera l'expression.

la moyenne est `\displaystyle \frac{1}{n}\sum_{i=1}^n x_i`

$$\text{la moyenne est } \frac{1}{n} \sum_{i=1}^n x_i$$

Les espaces horizontaux sont définis par défaut en mode math. Ils peuvent être contrôlés avec des commandes spéciales.

<code>x y</code>	<code>xy</code>	<code>x\; y</code>	<code>x y</code>	<code>x\quad y</code>	<code>x y</code>
<code>x\, y</code>	<code>xy</code>	<code>x\ y</code>	<code>x y</code>	<code>x\qquad y</code>	<code>x y</code>
<code>x\: y</code>	<code>xy</code>				

La commande `\!` diminue les espaces. Cette commande peut être répétée plusieurs fois.

<code>x+y</code>	<code>x + y</code>	<code>x\!+!\!y</code>	<code>x+y</code>	<code>x\!\!\!+\!\!\!y</code>	<code>x+y</code>
------------------	--------------------	-----------------------	------------------	------------------------------	------------------