

Table des matières

1 Rédiger ses documents avec L^AT_EX (semaine n° 1)	2
1.1 Ses qualités	2
1.2 Ses défauts	2
1.3 Ses concurrents	2
1.4 L ^A T _E X sous Linux	3
1.5 Premiers documents	3
1.5.1 Un simple éditeur suffit	3
1.5.2 Comment ça marche ?	3
1.6 À retenir	3
2 Architecture d'un document	4
2.1 Le préambule	4
2.1.1 Le documentclass	4
2.1.2 Les usepackage	4
2.1.3 Une page de présentation	4
2.2 Le corps du texte	4
2.2.1 Les hiérarchies de titres	4
2.2.2 La table des matières	5
3 Le texte	5
3.1 Polices	5
3.1.1 Le style	5
3.1.2 La taille	5
3.2 Les listes	6
3.2.1 Listes numérotées	6
3.2.2 Listes non numérotées	6
4 Un aperçu de la programmation avec L^AT_EX (semaine n° 2)	6
4.1 Définir de nouvelles commandes	6
4.2 Boucles	6

1 Rédiger ses documents avec \LaTeX (semaine n° 1)

\LaTeX est une surcouche d'un logiciel de traitement de texte appelé \TeX dont le projet démarra en 1977. Lors de la publication de l'un des tomes du célèbre *The Art of Computer Programming*, l'auteur Donald E. Knuth fut horrifié par la qualité des épreuves qu'il reçut de son livre. Le projet \TeX allait naître avec deux objectifs principaux

- Le premier était la qualité, il fallait que les documents produits soient les meilleurs.
- Le second était de créer un système indépendant – autant que possible – de l'évolution de la technologie.

Le projet LaTeX fut initié par Leslie Lamport en 1985, il décida d'écrire un ensemble de macros permettant de simplifier l'utilisation de \TeX . Ces macros permettent à un auteur de mettre en page son travail avec la meilleure qualité typographique possible. Elles sont ensuite insérées dans le texte à l'aide de mots clés appelés balises. Nous en verrons quelques exemples un peu plus loin.

1.1 Ses qualités

gratuité, \LaTeX est un produit entièrement gratuit sous la licence **LPPL** (LaTeX Project Public License), il fait partie du domaine public.

stabilité, la sous couche sur laquelle repose \LaTeX appelée \TeX – est achevée depuis 1990. Elle est si on peut dire le noyau qui permet à \LaTeX de fonctionner, le dernier bogue trouvé date de 1988. Nous sommes actuellement avec la version 3.14159 de \TeX . Knuth a voulu que les différentes versions empruntent leurs décimales au nombre π .

compatibilité, une portabilité absolue, la source d'un document \LaTeX donnera après compilation un résultat identique quelque soit la machine ou le système d'exploitation. Les distributions \LaTeX – \TeX live pour la plus connue – sont disponibles aussi bien pour linux, windows, macOS etc.

qualité typographique exceptionnelle en raison de l'unité de longueur utilisée qui vaut approximativement le centième de la longueur d'onde de la lumière visible. Les règles typographiques du moteur \LaTeX ont été écrites par des professionnelles.

efficacité, le système de mise en page est complètement détaché du contenu. L'utilisateur n'a pas à se préoccuper de l'esthétique visuelle de son texte. Les règles sont définies dès le départ dans un préambule standard ou personnalisé.

universalité, la communauté des utilisateurs est mondiale ce qui permet à la surcouche \LaTeX d'évoluer en fonction des besoins.

1.2 Ses défauts

les fontes qui ne sont pas aussi nombreuses que sous d'autres logiciels. Par contre celles qui sont disponibles sont d'une très grande qualité et particulièrement bien adaptées à l'écriture des mathématiques.

standardisation de la présentation. \LaTeX a été conçu pour des textes à typographe riche mais à mise en page pauvre. Les présentations personnalisées se font souvent au prix de d'efforts importants. Mais ces dernières années ont vu le développement de nouveaux packages pour faciliter la mise en forme des documents.

le balisage du texte nécessite un apprentissage plus long et moins intuitif que les logiciels WYSIWYG.

1.3 Ses concurrents

Comme tout le monde – ou presque – j'ai commencé avec Word et je m'en suis satisfait durant de longues années. Word, OpenOffice Writer ou LibreOffice plus récemment sont certainement les logiciels de traitement de texte les plus populaires, leur utilisation est intuitive pour les fonctionnalités de bases et relativement agréable à prendre en main puisqu'il fonctionne sur le principe du WYSIWYG qui est l'acronyme de la locution anglaise *What you see is what you get*, signifiant littéralement en français *ce que vous voyez est ce que vous obtenez*. Ces logiciels – pour ne citer qu'eux car il en existe beaucoup d'autres – permettent de satisfaire un grand nombre d'utilisateurs pour la rédaction de documents courts. Mais les choses se compliquent quand on veut introduire du matériel scientifique, images, équations, dessins dans le document. Dans ce contexte les logiciels de traitement de texte actuels sont relativement pauvres et l'utilisateur doit souvent faire appel à d'autres logiciels en essayant par la suite d'intégrer tant bien que mal le résultat obtenu dans le document source. Prenons l'exemple du chimiste qui aimerait dessiner sa molécule directement dans le document source, ou bien ce pourquoi \LaTeX a été développé écrire des équations mathématiques ou encore avoir la possibilité de dessiner des graphiques de grande qualité.

Mais l'écueil le plus gênant à mon avis concerne les problèmes de compatibilité existant entre les différents traitements de texte voir même entre les différentes versions d'un même logiciel. Ce qui limite considérablement la portabilité des documents. L^AT_EX offre la possibilité d'échanger les documents sans avoir à se soucier du matériel informatique.

Je tiens à préciser que mon intention n'est pas de juger de la qualité des traitements de texte cités plus haut¹, mais de soulever quelques points sensibles et d'éclairer le lecteur sur des solutions alternatives pour la rédaction de travaux scientifiques.

Le pas n'a pas été facile à franchir, car L^AT_EX avait, et a toujours je pense, la réputation d'un logiciel hermétique destiné au mordu de la ligne de commande.

1.4 L^AT_EX sous Linux

La distribution de référence est ici T_EXLive. Il suffit de l'installer depuis le gestionnaire (Synaptic) de packages de votre distribution. L'installation se résume très souvent au paquet **texlive-full**.

1.5 Premiers documents

1.5.1 Un simple éditeur suffit

Pour démarrer avec L^AT_EX il vous suffit d'ouvrir un éditeur et de sauvegarder votre fichier source avec l'extension **.tex**. Ce fichier est destiné à être compilé par le moteur L^AT_EX afin de produire un document au format **.pdf**. Pour la compilation il faut se placer avec la console dans le répertoire contenant le fichier source et utiliser la commande : `pdflatex monFichier.tex`

1.5.2 Comment ça marche ?

Pour notre premier document nous allons nous contenter du très célèbre `hello, world!`, voyons à quoi peut bien ressembler le source. La colonne de gauche correspond au texte du fichier source et celle de droite au document produit :

```

\documentclass{minimal}
\begin{document}
Hello, \textbf{world}!
\end{document}
Hello, world!
```

Le source est divisé en deux parties distinctes réduites ici à leur plus simple expression.

1. Le **préambule** correspondant à tout ce qui précède `\begin{document}`. C'est dans cette partie que l'on effectue tout les réglages qui seront appliqués à notre document et que l'on appelle les **packages**.
2. Le **corps du texte** correspondant à tout ce que l'on trouve entre `\begin{document}` et `\end{document}`.

On remarque aussi que certains mots sont précédés d'un `\`, ces mots sont appelés **commandes**. Le texte qui suit entre les accolades est alors appelé **argument** de la commande. On voit naître sur cet exemple un principe général : pour produire un effet, il faut donner une instruction au logiciel, il faut parler à L^AT_EX avec des mots qu'il comprend et sait interpréter. Donnons un autre exemple pour montrer que L^AT_EX est très fort en math...

$$\int_0^1 \sum_{k=0}^n a_k x^k$$

1.6 À retenir

L^AT_EX est un système de préparation de documents très différents d'un traitement de texte (à part la finalité commune qui est de produire un document mis en forme). Dans un traitement de texte de type Word ou OpenOffice, le texte est mis en forme en *direct* pendant que vous saisissez. En L^AT_EX, le processus est *asynchrone*, vous saisissez votre texte accompagné de balises ou commandes de mise en forme et ensuite vous demandez à L^AT_EX d'exécuter ces instructions.

1. Vous trouverez à ce sujet un article fort intéressant paru dans les cahiers GUTenberg n° 27, *Mais qu'est ce qu'ont bien pu nous apporter les systèmes WYSIWYG?* de Conrad TAYLOR

2 Architecture d'un document

2.1 Le préambule

2.1.1 Le documentclass

Plusieurs classes (ou types de présentation) prédéfinies s'offrent à vous.

Classe	Type de document
letter	une lettre
article	un article de type recherche
seminar	des transparents pour une présentation
report	un mémoire par exemple de fin de stage
book	un livre

2.1.2 Les usepackage

Juste après le `\documentclass` indiquer à \LaTeX quels packages vous souhaitez utiliser. Un package est un programme qui permet d'étendre les possibilités de \LaTeX . Appelez les quatres packages suivants dans tous vos documents. Les trois premiers permettent d'écrire en respectant les règles de typographie française et le dernier sert aux réglages des marges de votre page :

```
%francisation
\usepackage[francais]{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
%réglages des marges
\usepackage[left=2cm, right=3cm,
             top=2cm, bottom=2cm]{geometry}
```

2.1.3 Une page de présentation

\LaTeX peut même créer une page de présentation à l'aide des commandes suivantes :

Commande	Localisation	Exemple
<code>\title</code>	Préambule	<code>\title{Introduction à \LaTeX}</code>
<code>\author</code>	Préambule	<code>\author{Christophe Casseau}</code>
<code>\date</code>	Préambule	<code>\date{Le 25 avril 2013}</code>
<code>\abstract</code>	Corps du texte	<code>\abstract{Une simple introduction}</code>
<code>\maketitle</code>	Corps du texte	<code>\maketitle</code>

2.2 Le corps du texte

2.2.1 Les hiérarchies de titres

Les titres sont composés à l'aide de niveaux logiques, la taille de la police et la graisse sont gérées automatiquement par \LaTeX

Commande	Niveau de titre
<code>\part</code>	Partie
<code>\chapter</code>	Chapitre
<code>\section</code>	Section
<code>\subsection</code>	Sous-section
<code>\subsubsection</code>	Sous-sous-section

2.2.2 La table des matières

Créer une table des matières est très simple, encore une fois \LaTeX s'occupe de tout, il suffit de placer à l'endroit où doit apparaître la table la commande :

```
\tableofcontents
```

3 Le texte

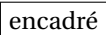

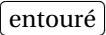
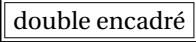
3.1 Polices

3.1.1 Le style

Les changements de style se font à l'aide des commandes suivantes en mettant le texte devant subir la modification entre accolades :

Style	Commande
<i>italique</i>	<code>\textit{...}</code>
<i>penché</i>	<code>\textsl{...}</code>
gras	<code>\textbf{...}</code>
typewriter	<code>\texttt{...}</code>
police de base	<code>\normalfont{...}</code>

Il existe bien sur un tas d'autres commandes d'enrichissement typographique :

Style	Commande	Package
<u>souligné</u>	<code>\uline{...}</code>	<code>\usepackage[normalem]{ulem}</code>
<u>double souligné</u>	<code>\uuline{...}</code>	<code>\usepackage[normalem]{ulem}</code>
<u>vague</u>	<code>\uwave{...}</code>	<code>\usepackage[normalem]{ulem}</code>
<u>barré</u>	<code>\sout{...}</code>	<code>\usepackage[normalem]{ulem}</code>
<i>HackHacké</i>	<code>\xout{...}</code>	<code>\usepackage[normalem]{ulem}</code>
Texte ^{en exposant}	<code>Texte...</code>	
1 ^{er} , 2 ^e	<code>1\ier{}</code> , <code>2\ieme{}</code>	
	<code>\fbox{...}</code>	<code>\usepackage{fancybox}</code>
	<code>\shadowbox{...}</code>	<code>\usepackage{fancybox}</code>
	<code>\ovalbox{...}</code>	<code>\usepackage{fancybox}</code>
	<code>\doublebox{...}</code>	<code>\usepackage{fancybox}</code>

3.1.2 La taille

La taille de la police peut être fixée dans le préambule à l'aide d'une option que l'on ajoute à `\documentclass` de la manière suivante :

```
\documentclass[11pt]{article}
```

Mais on peut aussi la modifier ponctuellement en signalant une variation relative par rapport à la taille normale, pour cela on utilise des *basculés* en plaçant la basculé et le texte entre accolade : `{\tiny petit mot}`

Basculé	Taille	Basculé	Taille
<code>\tiny</code>	taille	<code>\large</code>	taille
<code>\scriptsize</code>	taille	<code>\Large</code>	taille
<code>\footnotesize</code>	taille	<code>\LARGE</code>	taille
<code>\small</code>	taille	<code>\huge</code>	taille
<code>\normalsize</code>	taille	<code>\Huge</code>	taille

3.2 Les listes

3.2.1 Listes numérotées

<code>\begin{enumerate}</code>	
<code>\item Une première ligne</code>	1. Une première ligne
<code>\item Une deuxième ligne</code>	2. Une deuxième ligne
<code>\item La dernière ligne</code>	3. La dernière ligne
<code>\end{enumerate}</code>	

3.2.2 Listes non numérotées

<code>\begin{itemize}</code>	
<code>\item Une première ligne</code>	– Une première ligne
<code>\item[\textbullet] Une deuxième ligne</code>	• Une deuxième ligne
<code>\item[@] La dernière ligne</code>	@ La dernière ligne
<code>\end{itemize}</code>	
<code>\begin{description}</code>	
<code>\item[Le préambule] permet</code>	Le préambule permet d'effectuer les réglages
<code>d'effectuer les réglages</code>	
<code>\item[Le corps du texte]</code>	Le corps du texte raconte une histoire
<code>raconte une histoire</code>	
<code>\item[Les commandes] permettent</code>	Les commandes permettent de séparer
<code>de séparer le fond de la forme</code>	le fond de la forme
<code>\end{description}</code>	

4 Un aperçu de la programmation avec L^AT_EX (semaine n° 2)

4.1 Définir de nouvelles commandes

Les matières scientifiques sont le domaine de prédilection de L^AT_EX, il existe de nombreux packages permettant d'élaborer des textes scientifiques de grande qualité typographique. Mais il est parfois bien utile de créer une **commande** dédiée à un usage personnel. Prenons comme exemple une commande dont je me sers souvent pour rédiger les TP ou les contrôles, la commande `\ion{<argument>}`

Code `\newcommand{\ion}[1]{\mathrm{#1}$}`

Commande `\ion{Cu^{2+}}` `\ion{SO_4^{2-}}`

Résultats Cu^{2+} SO_4^{2-}

1. Placer le **code** dans le préambule.
2. Que représente #1 ?
3. Tester votre code à l'aide d'un exemple différent de ceux présentés.
4. Dans le même genre il est fastidieux de rédiger un texte contenant des formules d'isotopes. Renommer la commande `\ion` en `\fc` (formule chimique).
5. Tester cette nouvelle commande sur l'isotope :

$${}_{13}^{27}\text{Al}$$

4.2 Boucles

Pour réaliser une boucle L^AT_EX possède le package *multido* avec la commande du même nom. Sa syntaxe est la suivante :

$$\backslash\text{multido}\{\text{variables}\}\{\text{répétitions}\}\{\text{motif}\}$$

Code `\multido{}{10}{\textbf{ISN }}`

Résultat *ISN ISN ISN ISN ISN ISN ISN ISN ISN ISN*

Code `\multido{\n=0+1}{10}{$_{\n}X$, }`

Résultat ${}_0X, {}_1X, {}_2X, {}_3X, {}_4X, {}_5X, {}_6X, {}_7X, {}_8X, {}_9X,$

1. Ajouter dans le préambule le package *multido* à l'aide de la commande : `\usepackage{multido}`
2. Tester ces deux exemples
3. Comment afficher les isotopes suivants :
 ${}^{16}_8O, {}^{17}_8O, {}^{18}_8O,$