

Intelligence Artificielle par la Logique (AIL'12)

TD 1 : Une introduction rapide à L^AT_EX (en 3 heures)

T_EX est un système de mise en page visant à la création des documents contenant des formules mathématiques. L^AT_EX, prononcé /'leɪtɛk/ ou /'lɑ:tɛk/, est une extension de T_EX simplifiant la création d'un document avec un ensemble de commandes prédéfinies.

Aujourd'hui, il existe plusieurs distributions de L^AT_EX disponibles pour tout système d'exploitation comme Linux, Mac OS X ou Windows. Pour plus d'information sur l'installation de L^AT_EX sur votre ordinateur suivez <http://www.latex-project.org/ftp.html>.

Anatomie de fichier L^AT_EX

Tout document L^AT_EX est écrit en format de texte pur, comme c'est le cas des fichiers avec l'extension `.txt`. Des fichiers L^AT_EX ont par contre l'extension `.tex`. Pour créer des fichiers avec du texte pur il faut utiliser des éditeurs de texte simples e.g. Notepad sous Windows, Kate sous Linux, ou encore textEdit sous Mac OS X.

Nous créons notre premier document L^AT_EX contenant uniquement une phrase « Hello ! ». Pour ça, il faut sauvegarder le texte suivant dans un fichier `hello.tex` :

```
\documentclass{article}

\begin{document}
Hello!
\end{document}
```

Tout document L^AT_EX est composé de deux parties :

1. Le *préambule* qui contient des varies réglages du document. Le préambule commence dans la première ligne et finit avant `\begin{document}`. La première ligne doit avoir la déclaration du type du document `\documentclass{article}`.
2. Le *corps* qui est la partie principale du document, qui est entre `\begin{document}` et `\end{document}` et qui contient le texte propre du document.

Tout contenu après la fin du corps `\end{document}` est ignoré.

Compilation

Ensuite, nous compilons le fichier `hello.tex` pour en obtenir un document PDF. En informatique, la compilation est un processus qui prend un fichier du code *source* (`.tex`) et produit un fichier dans un format *cible* (`.pdf`).

Pour ça, il faut ouvrir Terminal, le diriger vers le répertoire où se trouve le fichier à compiler `hello.tex` et exécuter la commande `pdflatex hello.tex`. S'il y a aucun erreur, le compilateur affiche sur le Terminal le texte similaire au suivant :

```
staworko:~/work/classes/ili09/td2$ pdflatex hello.tex
This is pdfTeX, Version 3.1415926-1.40.9 (Web2C 7.5.7)
 %&-line parsing enabled.
entering extended mode
(.hello.tex
LaTeX2e <2005/12/01>
Babel <v3.81> and hyphenation patterns for english, use
nglismax, dumylang, nohyphenation, german-x-2008-06-18
, ngerman-x-2008-06-18, ancientgreek, ibycus, arabic, b
asque, bulgarian, catalan, pinyin, coptic, croatian, cz
ech, danish, dutch, esperanto, estonian, farsi, finnish
, french, galician, german, ngerman, monogreek, greek,
```

```
hungarian, icelandic, indonesian, interlingua, irish, i
talian, latin, lithuanian, mongolian, mongolian2a, bokm
al, nynorsk, polish, portuguese, romanian, russian, san
skrit, serbian, slovak, slovenian, spanish, swedish, tu
rkish, ukenglish, ukrainian, uppsorbian, welsh, load
e
d.
(/usr/local/texlive/2008/texmf-dist/tex/latex/base/arti
cle.cls
Document Class: article 2005/09/16 v1.4f Standard LaTeX
document class
(/usr/local/texlive/2008/texmf-dist/tex/latex/base/size
10.clo))
```

```
No file hello.aux.
[!{/usr/local/texlive/2008/texmf-var/fonts/map/pdftex/u
pdmap/pdftex.map}](./hello.aux) </usr/local/texlive/20
```

```
08/texmf-dist/fonts/type1/bluesky/cm/cmr10.pfb>
Output written on hello.pdf (1 page, 5620 bytes).
Transcript written on hello.log.
```

La compilation d'un fichier \LaTeX avec `pdflatex` crée 3 fichiers :

- `hello.log` qui contient les messages diagnostiques étendues. Typiquement, on n'a pas de consulter le contenu de ce fichier sauf les cas des erreurs difficiles à résoudre.
- `hello.aux` où le compilateur stock des informations utiles pour récompilations.
- `hello.pdf` qui contient la version compilée de `hello.tex`. Ce fichier doit être ouvert avec un lecteur de PDF (e.g. Kpdf ou Acrobat Reader).

Erreurs

Pour les débutants, les erreurs arrivent souvent et peuvent être très embêtants si l'on sait pas comment s'en sortir. Nous allons faire quelques erreurs exprès : dans un fichier nommé `error.tex` sauver et ensuite compiler le texte suivant :

```
\documentclass{article}
\begin{document}
I earn $10 an hour.
\end{document}
```

Ce texte contient un erreur qui fait le compilateur afficher une plainte suivante :

```
staworko:~/work/classes/ili09/td2$ pdflatex error.tex
This is pdfTeX, Version 3.1415926-1.40.9 (Web2C 7.5.7)
%&-line parsing enabled.
entering extended mode
(./error.tex
LaTeX2e <2005/12/01>
Babel <v3.81> and hyphenation patterns for english, use
nglishmax, dumylang, nohyphenation, german-x-2008-06-18
, ngerman-x-2008-06-18, ancientgreek, ibycus, arabic, b
asque, bulgarian, catalan, pinyin, coptic, croatian, cz
ech, danish, dutch, esperanto, estonian, farsi, finnish
, french, galician, german, ngerman, monogreek, greek,
hungarian, icelandic, indonesian, interlingua, irish, i
talian, latin, lithuanian, mongolian, mongolian2a, bokm
al, nynorsk, polish, portuguese, romanian, russian, san
skrit, serbian, slovak, slovenian, spanish, swedish, tu
```

```
rkish, ukenglish, ukrainian, uppsorbrian, welsh, loade
d.
(/usr/local/texlive/2008/texmf-dist/tex/latex/base/arti
cle.cls
Document Class: article 2005/09/16 v1.4f Standard LaTeX
document class
(/usr/local/texlive/2008/texmf-dist/tex/latex/base/size
10.clo))
No file error.aux.
! Missing $ inserted.
<inserted text>
$
1.4 \end{document}
?
```

Ici, le compilateur report l'erreur : `<< ! Missing $ inserted. >>`, sa location `<< 1.4 \end{document} >>` et demande quoi faire `<< ? >>`. Il y a plusieurs façons de lui répondre. Pour les toutes afficher tapez ? (point d'interrogation) suivit par la touche `< Entrée >`. Les 3 réponses suivantes sont les plus utilisées :

- `h` (`help`) affiche un descriptif court d'erreur.
- `x` (`exit`) quitte le compilateur ; tout le texte qui suit la position de l'erreur ne sera pas compilé.
- `q` (`quiet`) continue la compilation sans afficher aucun d'autre erreur. Cette réponse est conseillée, comme elle force \LaTeX à produire le fichier PDF où il est possible de visualiser le lieu d'occurrence de l'erreur.

Une autre type d'erreur est quand \LaTeX n'arrive pas à trouver de fichier. C'est souvent le résultat d'un typo dans la ligne de commande ou dans la préambule du document. Par exemple, si pour compiler le fichier `hello.tex` on lance la commande `pdflatex hell.tex`, \LaTeX se plaint de la façon suivante :

```
staworko:~/work/classes/ili09/td2$ pdflatex hell.tex
This is pdfTeX, Version 3.1415926-1.40.9 (Web2C 7.5.7)
%&-line parsing enabled.
entering extended mode
! I can't find file 'hell.tex'.
<*> hell.tex
```

(Press Enter to retry, or Control-D to exit)
Please type another input file name:

Ici, on peut taper le bon nom du fichier, mais c'est mieux de taper Ctrl+D et recommencer.

La langue française

Pour compiler le texte en français il faut indiquer le format de papier A4 et ajouter ces 2 lignes au préambule

```
\usepackage[utf8]{inputenc}
\usepackage[français]{babel}
```

Compiler le texte suivant

```
\documentclass[a4paper]{article}
```

```
\usepackage[utf8]{inputenc}
\usepackage[français]{babel}
```

```
\begin{document}
Salut, comment ça va?
\end{document}
```

Caractères spéciaux

En L^AT_EX il existent des caractères avec une signification spéciale et ils ne sont pas affichés (mais plutôt ils disent à L^AT_EX de faire quelque chose de spécial).

\$ & % # _ { } ~ ^ \

Pour afficher les 9 premiers caractères spéciaux (\$ & % # _ { } ~ ^), il faut les précéder avec un *backslash* \. Par exemple, pour produire \$ il faut utiliser \\$. Il y a plusieurs façons d'afficher le dernier symbole de la liste, par exemple \$\backslashbackslash\$. (**Attention** : La suite de deux backslash \\ a une signification encore spéciale). Éditer un fichier L^AT_EX pour obtenir le résultat suivant.

Si je gagnais \$1000, j'en déposerais 50% dans mon coffre-fort (dont le code est 191#223*). Le reste je dépenserais dans un magasin de la mode Mark&Spencer.

Mise en page

Typiquement, L^AT_EX s'occupe automatiquement de la mise en page i.e., il forme des lignes à partir des mots et des pages des lignes. L^AT_EX trait plusieurs espaces consécutives comme une et par conséquent le code suivant

```
Hello world! Hello world! Hello world! Hello world!
```

produit

```
Hello world! Hello world! Hello world! Hello world!
```

L^AT_EX ignore des occurrences singles de la caractère de nouvelle ligne (la touche « Entrée »). Pour indiquer un nouveau paragraphe, il faut utiliser au moins une ligne vide pour séparer des paragraphes. Par exemple,

C'est le premier paragraphe.
Il contient 3 phrases. Ça,
c'est sa dernière phrase!

Et ça, c'est la première
phrase de deuxième paragraphe
qui contient qu'une seule
phrase.

→

C'est le premier paragraphe. Il contient
3 phrases. Ça, c'est sa dernière phrase!

Et ça, c'est la première phrase de
deuxième paragraphe qui contient qu'une
seule ligne.

Les commandes et les blocs

Toute commande en L^AT_EX commence par un backslash \. Par exemple la commande `\today` insère la date courante, et la commande `\LaTeX` produit L^AT_EX. Les commandes ont typiquement des paramètres qui doivent être mis entre les accolades { et }. Par exemple, pour mettre le texte en gras on utilise la commande `\textbf{texte}`. Certaines commandes ne prennent pas de paramètres mais elles affectent tout le texte qui les suit (dans le même bloc). Là, on utilise les accolades pour limiter la portée de la commande. Par exemple, pour produire le texte avec une police plus grande on utilise `{\large texte}`. Plus précisément, les accolades sont utilisées pour définir un *bloc de texte* et les commandes affectent le texte qui suit jusqu'à la fin du bloc courant. Le texte de l'exemple suivant est pris de *Le Petit Nicolas*.

En sortant de `\textbf{l'école}`, j'ai
suivi un petit chien. Il avait l'air
`{\large perdu}`, le petit chien, il
était tout seul et ça m'a fait
beaucoup de peine. J'ai pensé que le
petit chien serait content de trouver
un ami et j'ai eu du mal à le
rattraper. Comme le petit chien
n'avait pas l'air d'avoir tellement
envie de venir avec moi, il devait se
méfier, je lui ai offert la moitié de
mon `\textbf{petit pain au {\large`
`chocolat}}` et le petit chien a mangé
le petit pain au chocolat et il s'est
mis à remuer la queue dans tous les
sens et moi je l'ai appelé Rex, comme
dans `{\large un film \textbf{policier}}`
que j'avais vu jeudi dernier.

→

En sortant de **l'école**, j'ai suivi un pe-
tit chien. Il avait l'air perdu, le petit
chien, il était tout seul et ça m'a fait
beaucoup de peine. J'ai pensé que le pe-
tit chien serait content de trouver un ami
et j'ai eu du mal à le rattraper. Comme
le petit chien n'avait pas l'air d'avoir tel-
lement envie de venir avec moi, il devait
se méfier, je lui ai offert la moitié de mon
petit pain au chocolat et le petit
chien a mangé le petit pain au chocolat
et il s'est mis à remuer la queue dans tous
les sens et moi je l'ai appelé Rex, comme
dans un film **policier** que j'avais vu
jeudi dernier.

Le tableau suivant contient quelques commandes utiles.

<code>\textbf{texte}</code> ou <code>{\bf texte}</code>	corps gras (boldface)
<code>\textsf{texte}</code> ou <code>{\sf texte}</code>	fonte type (sans serif)
<code>\texttt{texte}</code> ou <code>{\tt texte}</code>	fonte machine à écrire (typewriter typeface)
<code>\emph{texte}</code> ou <code>{\em texte}</code>	mis en évidence (<i>emphasis</i>)
<code>\underline{texte}</code>	texte souligné (<u>underline</u>)
<code>\small</code>	taille petite
<code>\large</code>	taille large
<code>\centerline{texte}</code>	ligne centre
<code>\og</code> et <code>\fg</code>	guillemet ouvrant « et fermant »

Rédiger un fichier L^AT_EX donnant le texte suivant

La naissance de Linux

En 1991, l'étudiant finlandais Linus Torvalds, que la faible disponibilité du serveur Unix de l'université de Helsinki indispose, entreprend d'écrire un noyau de système d'exploitation qu'on ap-

pellera plus tard *noyau Linux*. Linus désire alors surtout comprendre le fonctionnement de son ordinateur, un compatible PC basé sur un microprocesseur Intel 80386.

Le 5 octobre 1991, il annonce sur le forum Usenet la disponibilité d'une ébauche version 0.02 de son système d'exploitation, la version 0.01 ayant eu une diffusion plus que confidentielle. Le projet trouve son nom définitif grâce à Ari Lemmke, administrateur du serveur FTP `ftp.funet.fi`, qui héberge le travail de Linus Torvalds dans un répertoire nommé Linux. C'est la première apparition d'un terme composé à partir de Linus et UNIX, qui deviendra par la suite une marque déposée au nom de Linus Torvalds.

La différence essentielle de Linux par rapport à d'autres systèmes d'exploitation concurrents (comme Mac OS, Microsoft Windows et Solaris) est d'être un système d'*exploitation libre*, apportant quatre libertés aux utilisateurs, définies par la licence GNU GPL, les rendant indépendants vis-à-vis d'un quelconque éditeur, encourageant l'entraide et le partage.

Les logiciels libres sont produits de manière collaborative, souvent indépendamment les uns des autres, et peuvent être librement redistribués. Il s'ensuit une particularité du monde Linux : la séparation fréquente entre ceux qui produisent les logiciels et ceux qui les distribuent. On appelle *distribution Linux* une solution prête à être installée par l'utilisateur final. De nombreuses distributions plus ou moins spécialisées existent.

Ubuntu est une distribution spécialisée « *environnement bureau* », destinée à proposer un système convivial, ergonomique, libre et gratuit y compris pour les entreprises. Son nom provient d'un ancien mot bantou, quelqu'un d'*Ubuntu* désignant une personne sachant que ce qu'elle est, est intimement lié à ce que sont les autres, donc il est parfois traduit en l'appliquant au *je* : « *Je suis ce que je suis grâce à ce que nous sommes tous* ».

Environnements

L'environnement est une construction qui à la fois permet de structurer le texte et spécifier son traitement. L'environnement commence avec `\begin{nom-de-l'environnement}` suivi par le texte et fini par `\end{nom-de-l'environnement}`. Par exemple l'environnement `center` permet de centrer le texte e.g.,

<pre>\begin{center} \bf Le Petit Prince \end{center} Écrit par Antoine de Saint Exupéry, publié en 1943 à New York. C'est un conte poétique et philosophique sous l'apparence d'un conte pour enfants. C'est un récit qui n'a pas d'étiquette dans l'histoire littéraire.</pre>	→	<p style="text-align: center;">Le Petit Prince</p> <p>Écrit par Antoine de Saint Exupéry, publié en 1943 à New York. C'est un conte poétique et philosophique sous l'apparence d'un conte pour enfants. C'est un récit qui n'a pas d'étiquette dans l'histoire littéraire.</p>
---	---	---

Remarquons que la portée d'un l'environnement défini elle aussi un bloc de texte, i.e. commande qui change l'affichage de texte (`\bf` dans l'exemple précédent) affecte que le texte qui la suit dans l'environnement courant.

Ce qui fait l'environnement différent des blocs de textes et la possibilité d'utiliser dedans leur portée des commandes spéciales : chaque environnement a ses propres commandes spéciales et ces commandes ne peuvent pas être utilisées que dans son environnement. Par, exemple l'environnement `itemize` permet de produire des listes non-numéroté dont les éléments sont introduits avec la commande `\item`. Par exemple,

Les versions du système d'exploitation
`{\sf Mac OS X}` sont nommées avec les
nom des grand chats:
`\begin{itemize}`
`\item Cheetah (ver. 10.0)`
`\item Puma (ver 10.1)`
`\item Jaguar (ver. 10.2)`
`\item Panther (ver. 10.3)`
`\item Tiger (ver. 10.4)`
`\item Leopard (ver. 10.5)`
`\item Snow Leopard (ver. 10.6)`
`\end{itemize}`

→

Les versions du système d'exploitation Mac
OS X sont nommées avec les nom des grand
chats :

- Cheetah (ver. 10.0)
- Puma (ver 10.1)
- Jaguar (ver. 10.2)
- Panther (ver. 10.3)
- Tiger (ver. 10.4)
- Leopard (ver. 10.5)
- Snow Leopard (ver. 10.6)

Les environnements peuvent être imbriqués. Par exemple,

L'ordinateur est une machine
capable d'exécuter les programmes
et traiter des données.
Composantes d'un ordinateur:
`\begin{enumerate}`
`\item Processeur`
`\item Mémoire vive (RAM)`
`\item Mémoire externe`
`\begin{itemize}`
`\item Disque dur`
`\item CD-ROM`
`\item Clé USB`
`\end{itemize}`
`\item Periferiques d'entrée/sortie`
`\begin{itemize}`
`\item Clavier et souris`
`\item Écran`
`\item Lecteur CD/DVD`
`\item Carte Réseau`
`\end{itemize}`
`\end{enumerate}`

→

L'ordinateur est une machine capable
d'exécuter les programmes et traiter des
données. Composantes d'un ordinateur :

1. Processeur
2. Mémoire vive (RAM)
3. Mémoire externe
 - Disque dur
 - CD-ROM
 - Clé USB
4. Périphériques d'entrée/sortie
 - Clavier et souris
 - Écran
 - Lecteur CD/DVD
 - Carte Réseau

Tableaux.

Un environnement très utile est `tabular` qui permet de créer des tableaux. Nous le présentons avec un exemple

Tableau d'équivalence des lampes
`\begin{tabular}{|l|r|r|r|}`
`\hline`
No. & Fluorescente & Halogène & Classique\\
`\hline`
1 & 9 watts & 14 watts & 30 watts\\
2 & 11 watts & 20 watts & 40 watts\\
3 & 15 watts & 30 watts & 60 watts\\
4 & 20 watts & 37 watts & 75 watts\\
5 & 23 watts & 48 watts & 100 watts\\
`\hline`
`\end{tabular}`

→

	Fluorescente	Halogène	Classique
	9 watts	14 watts	30 watts
	11 watts	20 watts	40 watts
	15 watts	30 watts	60 watts
	20 watts	37 watts	75 watts
	23 watts	48 watts	100 watts

Cet environnement prends un paramètre `{|r|r|r|}` qui spécifie à la fois le nombre des colonnes (3) et comment le texte doit être positionne dans chacune des colonnes (1 – à gauche, c – centré, r – à droite). Aussi, tout `|` produit une ligne verticale. La commande `\hline` produit une ligne horizontale. Le symbole `&` est utilisé pour séparer les cellules d'une ligne du tableau et finalement `\\` indique la fin de la ligne du tableau.

Formules mathématiques

Pour produire des formules mathématiques avec \LaTeX il faut entrer dans un mode mathématique par un des deux environnements mathématiques de base. Le premier, `math`, intègre la formule dans le texte. Le deuxième, `displaymath` produit la formule dans sa propre ligne. Ces environnements sont si souvent utilisés que \LaTeX permet d'utiliser des versions abrégées : `\$formule\$` pour une formule intégrée dans le texte et `\[formule\]` pour une formule dans sa propre ligne.

<p>Une <code>\em equation quadratique</code> se présente sous la forme suivante</p> <pre>\[ax^2 + bx + c = 0 \]</pre> <p>où <code>\\$a\neq 0\\$</code>. Le <code>\em discriminant</code> de cette équation est <code>\Delta = b^2 - 4ac\\$</code>.</p>	→	<p>Une <i>équation quadratique</i> se présente sous la forme suivante</p> $ax^2 + bx + c = 0$ <p>où $a \neq 0$. Le <i>discriminant</i> de cette équation est $\Delta = b^2 - 4ac$.</p>
---	---	--

Réécrire le texte ci-dessus en utilisant les environnements `math` et `displaymath` au lieu de `\$...\$` et `\[...\]` respectivement.

Ils existent des milliers de commandes mathématiques, très utiles pour créer des formules très complexes mais toujours de qualité typographique supérieure. Deux constructions très souvent utilisées dans le mode mathématique ce sont les indices (a_i) et les exposants (x^2) obtenues avec le caractère spécial `_` et `^` respectivement. Par exemple, `\$a_{b}\$` produit a_b et `\$q^{b}\$` donne q^b . Les indices et exposants peuvent être imbriqués et leur argument peut être une formule compliquée, par exemple `\[2^{a_n+2^{n+f(n^2)}}\]` produit

$$2^{a_n+2^n+f(n^2)}$$

En utilisant **Aide-mémoire** et « *Une introduction courte à $\text{\LaTeX} 2_{\epsilon}$* » disponibles sur la page du cours, créer un document avec les formules ci-dessous. Faire ça deux fois (dans le même document) : une fois en utilisant `math` et une fois en utilisant `displaymath`. Comparer les résultats.

$\frac{1}{2} + \frac{1}{3} = \frac{3}{6} + \frac{2}{6} = \frac{5}{6},$	$x \leq 2y^3 + \sqrt{z},$	$\lim_{n \rightarrow \infty} f(x_n) = \frac{\pi^2}{6},$
$\sum_{n=1}^{20} a_n = 230,$	$\sin(2\alpha) = 2 \sin \alpha \cos \alpha,$	$\int_a^b f(x) dx = [F(x)]_a^b.$

Et maintenant quelques commandes utiles pour écrire des formules logiques (à utiliser uniquement dans un mode mathématique).

Commande	Symbole	Commande	Symbole
<code>\land</code>	\wedge	<code>\lor</code>	\vee
<code>\lnot</code>	\neg	<code>\leftarrow</code>	\leftarrow
<code>\rightarrow</code>	\rightarrow	<code>\leftrightarrow</code>	\leftrightarrow
<code>\Leftarrow</code>	\Leftarrow	<code>\Rightarrow</code>	\Rightarrow
<code>\equiv</code>	\equiv	<code>\models</code>	\models
<code>\varphi</code>	φ	<code>\phi</code>	ϕ
<code>\psi</code>	ψ	<code>\Phi</code>	Φ
<code>\Psi</code>	Ψ	<code>\vdash</code>	\vdash