

Introduction à \LaTeX

Langage et structure

Pascal Bessonneau

05/2016

Plan

Le langage \LaTeX

Type de document

La compilation

Chemins de fichiers

Le visionneur de PDF

Le langage

L^AT_EX utilise des caractères spéciaux que le compilateur va reconnaître et utiliser pour baliser le texte.

Ainsi les caractères `\`, `$`, `&`, `-`, `{`, `}`, `%`, `[`, `]`, `~` sont réservés. Ils ont une signification particulière.

Par exemple :

$$\backslash commande [options] \{ argument \}$$

commande va être reconnue comme une commande ou macro-commande L^AT_EX.

Le langage

Quand la commande prend un argument obligatoire, la syntaxe devient :

$$\backslash\text{commande}\{ \textit{argument} \}$$

Quand la commande prend plusieurs arguments obligatoires, la syntaxe devient :

$$\backslash\text{commande}\{ \textit{argument1} \}\{ \textit{argument2} \}...$$

Le langage

Lorsque qu'un argument est optionnel alors il est entre crochets :

$$\backslash\text{commande} [\textit{options}] \{ \textit{paramètre} \}$$

L'ajout d'une * permet dans certains cas de modifier un peu le comportement d'une fonction.

Le langage

Un document \LaTeX très simple :

Code

```
% Document 01
\documentclass{article}
\usepackage[latin1]{inputenc}
\usepackage[french]{babel}
\title{Introduction à \latex}
\author{Pascal Bessonneau}
\date{DEPP, 12/2010.}
\begin{document}
\maketitle
La preuve qu'il y a des êtres intelligents...
\end{document}
```

Les environnements

Les environnements vont être définis comme suit :

Code

```
\begin{nom}  
...  
\end{nom}
```

Les environnements

Les types d'environnements sont nombreux en \LaTeX : listes, code source, ... Ils permettent de spécifier un ensemble de propriétés pour les objets à l'intérieur de l'environnement.

Ils commencent par un `\begin` et se termine par un `\end` suivi dans les deux cas par le type d'environnement.

Les groupes

Une syntaxe légèrement différente permet de définir des groupes.
Cette syntaxe se rapproche de celle de R ou Perl :

Code

```
{ \mot-clef  
...  
}
```

Ce type de syntaxe se retrouve notamment pour les changements de police.

Les caractères réservés

En plus des caractères réservés vus précédemment, ils existent d'autres caractères ayant une signification pour \LaTeX comme le @ par exemple. Ils sont utilisés par \LaTeX pour l'appel de variables privées, les maths, ...

Dans le tableau ??, la liste des correspondances pour ces caractères est donné pour les inclure en tant que tel dans un texte.

Caractère	Façon de l'écrire en \LaTeX	Fonction
\$	$\backslash \$$	sert pour les mathématiques
&	$\backslash \&$	sert pour les tableaux et les maths
-	$\backslash -$	sert en mode mathématique
{	$\backslash \{$	sert pour les appels de fonction
}	$\backslash \}$	sert pour les appels de fonction
[$[$	sert pour les appels de fonction
]	$]$	sert pour les appels de fonction
%	$\backslash \%$	lance le mode commentaire

Les caractères accentués

Les caractères accentués sont gérés par L^AT_EX en spécifiant correctement l'encodage des caractères. Toutefois pour les majuscules accentuées il suffit pour le français d'utiliser les commandes suivantes :

Accents et graphèmes	code L ^A T _E X
É	<code>\'E</code>
È	<code>\'E</code>
Â	<code>\^a</code>
ï	<code>\"I</code>
Ç	<code>\c C</code>
cœur	<code>c\oe ur</code>
CŒur	<code>C\OE UR</code>
æ	<code>\ae</code>
Æ	<code>\AE</code>


Les caractères accentués

Pour une liste exhaustive des commandes disponibles, cette page est très bien faite.

Ces commandes fonctionnent également pour les minuscules. Ceci est intéressant de le savoir notamment dans le cas où l'encodage est imposé comme les fichiers bibtex (références bibliographiques).

Les caractères spéciaux

The Comprehensive \LaTeX Symbols list est un document précieux...
Il contient tous les caractères spéciaux accessibles y compris dans
certains paquets comme *amsmath*.

Il y en a plusieurs milliers : ©, †, ™, , ✓, ❄, ...

La gestion des espaces et des tabulations

Les espaces sont automatiquement gérés par le compilateur qui supprime les espaces et les tabulations excédentaires :

Code

```
La preuve qu'il y a des êtres...\nLa           preuve           qu'il y a des  êtres...\n
```

donne

La preuve qu'il y a des êtres...

La preuve qu'il y a des êtres...

Sauts de ligne et commentaires

Quand vous laissez une ligne blanche, \LaTeX suppose que vous passez au paragraphe suivant. Un paragraphe correspond donc à un blocs de lignes sans lignes vides.

En cas de besoin, vous pouvez forcer un saut de ligne en utilisant la commande `\\` comme dans l'exemple précédent.

Des commentaires peuvent être ajoutés en début de ligne ou plus loin... Tout ce qui suit un `%` est un commentaire.

documentclass

Le *documentclass* va donner l'allure générale du document :

- `article` : pour les articles scientifiques, les présentations, les rapports courts, la documentation de programme, les invitations,...
- `proc` : une classe pour les actes de conférences basé sur *article*
- `minimal` : Il contient seulement les réglages pour la dimension de la page et une police de base. Il est conçu plutôt pour du débogage
- `report` : pour les rapports plus longs contenant quelques chapitres, les petits livres, les thèses, ...
- `book` : pour les livres
- `slides` : pour les diapositives. La classe utilise des grandes lettres sans serif
- `memoir` : pour changer sensiblement la présentation du document. Ce type est basé sur le type *book* mais vous pouvez créer n'importe quel document avec
- `letter` : pour les lettres

documentclass

Il existe d'autres *documentclass* développés et qui sont disponibles sous forme de paquets ou de fichiers de styles :

- Beamer, pour les présentations
- Prosper, idem
- Modèle de revue (ex : *Journal of Statistical Software*)
- ...

Les mots-clefs

Associé au *documentclass*, des mots-clefs permettent de remplir des champs prédéfinis : auteur, titre, date, ...

Attention, les mots-clefs prédéfinis dépendent du type de *documentclass* que vous utilisez. Ils permettront d'une part de créer les titres et d'autre part de renseigner les champs des fichiers PDF créés.

Il faut mémoriser la façon dont les renseignements sont entrés qui est classique sous L^AT_EX :

`\commande { paramètre }`

La structuration des documents

il est plus difficile dur d'écrire un document non structuré avec L^AT_EX. En effet vous pouvez jouer un peu avec la typographie mais en fait la mise en forme des titres passe par une commande qui d'une part :

- mets la mise en forme adaptée au niveau hiérarchique
- ajoute une référence dans la table d'index afin de l'inclure dans la table des matières

La structuration : book

Les niveaux hiérarchiques définis par défaut sont :

- `\part{}`
- `\chapter{}`
- `\section{}`
- `\subsection{}`
- `\subsubsection{}`
- `\paragraph{}`
- `\subparagraph{}`

La structuration : report

Les niveaux hiérarchiques définis par défaut sont :

- `\part{}`
- `\chapter{}`
- `\section{}`
- `\subsection{}`
- `\subsubsection{}`
- `\paragraph{}`
- `\subparagraph{}`

La structuration : article

Les niveaux hiérarchiques définis par défaut sont :

- `\part{}`
- `\section{}`
- `\subsection{}`
- `\subsubsection{}`
- `\paragraph{}`
- `\subparagraph{}`

La structuration : letter

La structure est un peu différente et on ne retrouve que les commandes suivantes :

- `\signature{}`
- `\address{}`
- `\opening{}`
- `\closing{}`

La table des matières

Une fois que le document contient la structure, il suffit de taper la commande...

Code

```
\tableofcontents
```

... pour imprimer la table des matières où on le souhaite.

Eviter le référencement

Si on ne veut pas qu'une *section*, *chapter*, ... apparaisse dans la table des matières il faut utiliser une * :

Code

```
\section*{Mon titre de section}
```

Ce titre apparaitra dans le même style qu'un titre de section classique mais pas dans la table des matières.

La compilation...

Entre le fichier texte et le fichier final, il y a une phase de compilation.

Le principe est que le fichier va être interprété puis traduit au format PostScript ou PDF.

Théoriquement cette phase se fait en ligne de commande. En pratique avec MiK_TE_X et T_EXnicCenter sur vos postes, il suffira d'un raccourci clavier pour appeler les programmes nécessaires.

Comprendre la compilation

- le compilateur définit la classe de document
- le compilateur charge les macros associées à chaque paquet (vérifie leur compatibilité)
- éventuellement le compilateur mémorise les macros utilisateurs
- le compilateur commence la lecture de la partie texte proprement dite
- il met en page au fil du document et crée des fichiers annexes avec la position des sections, figures,...
- Quand il arrive au `\end{document}`, il arrête la compilation

Comprendre la compilation

Lors de la compilation, le compilateur crée une série de fichiers. Le plus important est le fichier log. Il est automatiquement ouvert dans T_EXMaker. Il permet de voir si la compilation s'est bien déroulée et de déterminer l'erreur dans le cas contraire.

Dans T_EXMaker, la compilation s'arrête à la première erreur. Le compilateur attend alors une réponse de l'utilisateur d'où le " ?" qui s'affiche. En pratique, il vaut mieux modifier le document et recommencer la compilation que d'utiliser les commandes de " déboguage" .

La compilation...

Comme sous Word pour les longs documents, la création de la table des matières impose la pagination du document.

Ici, l'équivalent est que plusieurs compilations sont parfois nécessaires pour obtenir le fichier final.

1. Le fichier est transformé en différents formats et un fichier *index* qui contient la position de chaque éléments de la structure est créé
2. Le fichier est recompilé avec le fichier *index* correct et la table des matières est remplie.

En effet il est nécessaire pour le compilateur de faire la mise en page finale pour déterminer à quelle position se trouve chaque élément.

Plusieurs phases sont aussi nécessaires pour la bibliographie dont nous parlerons plus tard.

Les chemins pour les fichiers

\LaTeX est assez capricieux pour les chemins de fichiers :

1. déjà il utilise des / à la place des \ pour délimiter les répertoires (comme UNIX)
2. la présence d'accents ou de blancs peut parfois être bloquant.
Il est bon de prendre l'habitude d'éviter les blancs et les accents quand on travaille avec \LaTeX

Acrobat Reader vs. SumatraPDF

Sur la plupart des postes, c'est Acrobat Reader (ou Acrobat) qui est installé. Si le fichier PDF est ouvert dans Acrobat alors vous ne pourrez pas compiler le fichier. Vous serez obligé de fermer le fichier.

Par contre si vous utilisez SumatraPDF (c'est un logiciel libre !), vous pouvez laisser ouvert le PDF, la compilation se fera et vous verrez une fois la compilation terminée les modifications directement sans avoir à ouvrir de nouveau le PDF.