



## L<sup>A</sup>T<sub>E</sub>X, un peu, beaucoup

Pascal Dupont

Mots clés : L<sup>A</sup>T<sub>E</sub>X, traitement de texte



### 3. Le béhaba

Fin décembre, vous avez suivi le mode d'emploi proposé dans *Losanges* et vous avez installé sur votre ordinateur les outils nécessaires (un éditeur et un compilateur) à l'utilisation de L<sup>A</sup>T<sub>E</sub>X. Depuis lors, vous piaffez d'impatience, dans l'attente de les utiliser.

Dans cet article, nous ne tournerons pas autour du pot : nous vous donnerons un exemple utilisable tout de suite, un exemple à partir duquel vous pourrez broder — et expérimenter ! — pour créer vos propres documents.

Mais il y a tout de même quelques notions à expliquer avant d'aller plus loin.

Le document source, celui que vous tapez, est divisé en deux parties fondamentalement différentes :

1. Le *préambule*, qui contient les instructions nécessaires au bon formatage du document.
2. Le *corps du texte*, qui contient le texte proprement dit.

Dans le préambule, rien ne peut donc se trouver qui produirait une « sortie » de texte ; seulement des réglages de paramètres (comme la langue utilisée, la largeur de ligne, etc.) et des définitions de commandes qui seront utilisées dans la suite. Par exemple, pour un article, le titre et le nom de l'auteur seront définis dans le préambule, mais l'instruction donnant l'ordre de les imprimer se trouvera dans le corps du texte.

L'exemple promis arrive très bientôt (je vous sens impatients) ; mais je dois encore expliquer deux choses.

1. Les *commandes*. Dans le document source, vous ne pouvez pas les manquer : chacune débute par le symbole « \ », que nous appellerons *controbligue* pour éviter un anglicisme. Ce caractère spécial, qui ouvre le nom de la commande, est suivi soit d'un unique caractère autre qu'une lettre, soit d'une ou

plusieurs lettres ; dans ce second cas, le nom de la commande se termine avec le premier caractère autre qu'une lettre. Par exemple, si je tape « `\$abc` », cela se comprend comme la commande `\$` suivie des trois lettres `a`, `b` et `c`. Si au contraire je tape « `\abc$` », cela se comprend comme la commande `\abc` suivie du caractère spécial `$`. On notera aussi que les blancs suivant un nom de commande sont ignorés. Donc, « `\abc$` », « `\abc $` » voire « `\abc $` » sont parfaitement équivalents. Une commande peut, ou non, avoir des arguments.

2. Les caractères de groupement sont les accolades `{` et `}`. Ils sont utilisés pour définir des blocs de texte, qui serviront par exemple d'arguments plus ou moins longs à des commandes.

Encore une chose que je suis bien obligé de mentionner tout de suite : certaines commandes ont des arguments optionnels ; ceux-ci sont alors délimités par des crochets droits `[` et `]`.

#### Un exemple commenté

Dans l'encadré ci-après, se trouvent côte à côte un document source assez bref mais déjà représentatif de bon nombre de caractéristiques essentielles et le document composé correspondant.

Examinons d'abord sa structure globale. Le préambule est formé du premier bloc de lignes. Le corps du texte s'étend de `\begin{document}` à `\end{document}`.

Le préambule définit d'abord la *classe* du document : ici, un article ; d'autres classes possibles auraient été *book*, pour un livre, *report*, pour un rapport, *letter* pour du courrier, etc.

Les deux lignes suivantes définissent l'encodage du document ; sans elles, il aurait été interdit d'employer dans le document source des lettres accentuées ; nous aurions dû nous contenter d'un jeu restreint de caractères — ce qui a des avantages et des inconvénients, nous traiterons ces questions dans un article ultérieur.



```
\documentclass[a4paper]{article}
\usepackage[T1]{fontenc}
\usepackage[applemac]{inputenc}
\usepackage[french]{babel}
\title{Sur l'irrationalité de
 $\sqrt{2}$ }
\author{Piet~\textsc{Agore}}
\date{}
% Article envoyé à Losanges
% le 29/02/2014
```

```
\begin{document}
```

```
\maketitle
```

```
Nous nous proposons, dans cette
brève note, de prouver le résultat
suivant :  $\sqrt{2} \in \mathbf{R} \setminus \mathbf{Q}$ .
Autrement dit, quels que soient les
naturels (non nuls)  $m$  et  $n$ ,
 $\left(\frac{m}{n}\right)^2 \neq 2$ .
```

```
En effet, s'il existait des naturels
 $m$ ,  $n$  pour lesquels le carré de
 $m/n$  vaut  $2$ , nous aurions
 $m^2 = 2n^2$ .
```

```
Or, ceci ne se peut, car le
membre de gauche, comme tout carré
parfait, contient un nombre pair de
facteurs  $2$ , tandis que le membre de
droite en contient un nombre impair.
```

```
\hfill\rule{1ex}{1ex}
```

```
\end{document}
```

Le document source de cet exemple est disponible  
(dans les trois codages) sur le site de la SBPMef.

## Sur l'irrationalité de $\sqrt{2}$

Piet AGORE

Nous nous proposons, dans cette brève note,  
de prouver le résultat suivant :  $\sqrt{2} \in \mathbf{R} \setminus \mathbf{Q}$ .  
Autrement dit, quels que soient les naturels  
(non nuls)  $m$  et  $n$ ,

$$\left(\frac{m}{n}\right)^2 \neq 2.$$

En effet, s'il existait des naturels  $m$ ,  $n$  pour  
lesquels le carré de  $m/n$  vaut  $2$ , nous aurions

$$m^2 = 2n^2.$$

Or, ceci ne se peut, car le membre de gauche,  
comme tout carré parfait, contient un nombre  
pair de facteurs  $2$ , tandis que le membre de  
droite en contient un nombre impair. ■

Ce qu'il faut savoir, pour le moment, c'est qu'il existe différents standards pour le codage des caractères ; celui qui est utilisé par défaut sous *Mac-OS X* est dénommé *applemac*, passé en option à la commande `\usepackage` qui appelle le module *inputenc* (un module (*package*) est un fichier de commandes supplémentaires). Si nous avons travaillé sous *Windows*, nous aurions probablement utilisé le codage *latin1*, tandis que si nous étions des fanatiques d'*unicode* — comme beaucoup d'utilisateurs de *Linux* — nous aurions utilisé, par exemple, *utf-8*. Vient ensuite l'instruction `\usepackage[french]{babel}`. Elle indique au compilateur que le texte est rédigé en français. Ceci est important parce que, par exemple, les règles régissant les espaces autour des signes de ponctuation, ou encore les règles de

coupure des mots en bout de ligne, diffèrent d'une langue à l'autre. Si nous avions omis cette ligne du préambule, L<sup>A</sup>T<sub>E</sub>X aurait imaginé que notre texte était en anglais et l'aurait (mal)traité comme tel.

Les trois dernières instructions du préambule définissent le titre de l'article, son auteur, et la date de rédaction. Ici, cette dernière est déclarée vide, au moyen de deux accolades immédiatement juxtaposées. Sans cette précaution, une date par défaut aurait été utilisée : la date du jour de la compilation, connue, en principe, du système d'exploitation. On notera que le patronyme de l'auteur est composé en petites capitales, grâce à l'instruction `\textsc`, où « **sc** » signifie *small capitals*.



Le préambule se termine par deux lignes que le compilateur va négliger, parce qu'elles débutent par le symbole % : dans le préambule ou dans le corps du texte, tout ce qui suit ce symbole, sur la même ligne, est ignoré. Ici, l'auteur du texte a utilisé cette latitude pour laisser dans le document source des notes personnelles ; ce pourraient être aussi, par exemple, des variantes non retenues du texte.

Vient ensuite le corps du texte. Celui-ci commence par l'instruction `\maketitle`, qui compose le titre sur base, d'une part des *données* fournies dans le préambule, et d'autre part des instructions de formatage prédéfinies par la classe *article* ; en particulier, titre et auteur sont centrés sur deux lignes consécutives, le titre en caractère plus grand que l'auteur.

Pour la suite, document source et document composé se ressemblent davantage. Nous pouvons cependant épingler les remarques suivantes. Pour les mathématiques d'abord :

- Les « maths » (tout ce qui est équations, relations, variables, ...) sont encadrées soit de symboles `$...$`, lorsqu'il s'agit de mathématiques dans le texte, soit de `\[...\]` pour obtenir une équation hors-texte, centrée entre deux lignes.
- Les constructions et symboles d'opération classiques des mathématiques s'obtiennent à l'aide de commandes ; on aura déjà repéré dans ce bref exemple `\sqrt{...}` pour une racine carrée, `\frac{...}{...}` pour une fraction, `\in` pour le symbole d'appartenance, `\setminus` pour la différence ensembliste. Il s'agit donc là, respectivement, de commandes à 1, 2, 0 et 0 arguments.
- La racine carrée de 2 a été obtenue, simplement, par `\sqrt{2}`. S'il s'était agi de la racine de 17, nous aurions dû utiliser des accolades pour grouper les deux caractères constituant l'argument de la fonction : `\sqrt{17}` ; pour obtenir  $\sqrt{a}$ , plus besoin d'accolades, mais par contre il faut séparer le nom de la commande, `\sqrt`, de son argument *a*, puisque celui-ci est une lettre : il faut donc taper `\sqrt{a}`, avec un blanc, ou alors `\sqrt{a}`. Si nous avons `\sqrt{a}` dans le document source, le compilateur croira avoir affaire à la commande `\sqrt{a}` et nous indiquera, dans un message d'erreur, qu'il ne la connaît pas.

- `\left(` et `\right)`, appariés, fournissent des parenthèses dont la taille s'adapte automatiquement à leur contenu.

- Pour obtenir un exposant, nous avons utilisé le caractère spécial `^` (accent circonflexe). Ici aussi, pour obtenir un exposant constitué de plusieurs caractères, on utilise les accolades : `$2^{10}=1024$` fournit  $2^{10} = 1024$ , incorrect ; c'est `$2^{\{10\}}=1024$` qu'il faut encoder pour obtenir  $2^{10} = 1024$ .

Il est bien sûr possible d'avoir des puissances dans l'exposant :  $a^{b^c} = a^{(b^c)} \neq (a^b)^c = a^{bc}$  est, très logiquement, codé comme ceci : `$a^{\{b^c\}}=a^{\{(b^c)\}}\neq(a^b)^c=a^{\{bc\}}$` (`\neq`, pour *not equal*, fournit le symbole  $\neq$ ).

Les mêmes conventions fonctionnent avec les indices, que fournit le trait de soulignement : `$a_{ij}$` donne  $a_{ij}$ .

Pour ce qui est des particularités de l'encodage du texte, voici les principales remarques :

- Plusieurs blancs consécutifs dans le document source équivalent à un seul ; il faut ici comprendre par blanc aussi bien le caractère d'espacement, qu'une tabulation ou un « retour chariot » (passage à la ligne).
- Un retour chariot est donc interprété comme un blanc : passer à la ligne dans le document source ne provoque pas de passage à la ligne dans le document composé. La commande d'alinéa est une ou plusieurs lignes vides (autrement dit, deux retours chariot consécutifs, au moins). Dans l'exemple, c'est le cas avant « En effet ».
- La commande `\hfill` fournit un blanc extensible ; ici, il repousse à la marge droite le symbole suivant.
- Celui-ci est une `\rule`, une « règle », c'est-à-dire une ligne (en fait, un rectangle) dont la largeur est d'un « ex » et la hauteur aussi ; le « ex » est une unité de longueur utilisée par T<sub>E</sub>X, égale à la hauteur du caractère « x » minuscule dans la police en vigueur ; il s'agit donc d'une longueur relative. Ici, le but est d'obtenir un carré noir qui aura pour côté exactement la hauteur des minuscules.
- Le tilde est un blanc insécable ; il interdit donc au compilateur d'aller à la ligne entre le nom et le prénom de l'auteur (ce qui est de toute manière fort improbable, ici) ou entre « facteurs » et « 2 ».