

## 0.1 Introduction

### 0.1.1 1. Au sujet de ces didacticiels :

Le premier point que je signalerai est le suivant : en ce jour (08/09/2003) la série de didacticiels consacrés au formateur de texte LaTeX est inachevée, et, conséquemment, certains points ne sont pas encore traités. J'ai tout de même décidé de mettre ceux finis en ligne dans la mesure qu'il me faudra encore beaucoup de temps pour achever ce travail, et que ceux déjà existants permettent une première approche de LaTeX pour un non-initié. Le second point concerne le format choisi. J'ai très longtemps hésité à faire ces didacticiels directement avec LaTeX. Finalement, j'ai décidé tout de même d'utiliser le html pour leur rédaction. Ceci implique qu'une majorité de code donné ne montre pas l'effet visuel. C'est donc à vous de le tester pour en voir son effet. Ceci dit, une version LaTeX verra le jour prochainement. Lors de la rédaction de ces didacticiels, j'ai voulu donner les bases, ce qui est à la portée de tout débutant en LaTeX. Mais j'ai aussi voulu pousser un peu plus loin en essayant de trouver des solutions à quelques problèmes auxquels on peut être confronté, ce qui est réservé déjà plus à des personnes initiées. En conséquence, dans chaque didacticiel, ce sera à vous de ne prendre que ce qui correspond à votre niveau. Et le conseil que je vous donne est de n'utiliser que ce que vous comprenez. En guise de dernier point, sachez qu'il existe une source importante d'informations relative à LaTeX et Google donne en général de très bons résultats. Pour ma part, cinq documents sont mes références et m'ont d'ailleurs servi pour la rédaction des tutoriaux :

- Une courte introduction à LaTeX<sup>1</sup> : c'est pour moi la Bible de LaTeX. En plus d'apprendre les syntaxes nécessaires, ce recueil vous donne les règles de l'art pour la rédaction d'un document. Il est, je crois, indispensable.
- Joli manuel pour LaTeX<sup>2</sup> : ce guide est déjà plus poussé. Très intéressant pour trouver des solutions en cas de difficultés.
- Introduction à LaTeX<sup>3</sup> par Jean-Marc Lichtlé : en plus de donner les bases, ce guide vous expliquera la classe lettre qui n'aura alors plus aucun secret pour vous.
- FAQ LaTeX<sup>4</sup> : du même auteur que le Joli manuel, cette documentation en ligne va droit au coeur des problèmes et en donne des solutions faciles à mettre en oeuvre.
- Aide-memoire-LaTeX.ps<sup>5</sup> : comme son nom l'indique, un résumé de quelques pages sur les commandes de LaTeX, très complet. Attention tout de même, un aide-mémoire n'est jamais véritablement un cours.

### 0.1.2 2. Un bref historique :

Au commencement fût TeX. Son auteur, Donald Ervin Knuth<sup>6</sup>, lors de la rédaction de l'un de ses chefs d'oeuvre, *The Art of Computer Programming*<sup>7</sup>, trouva les résultats obtenus sur les épreuves peu convaincants. À partir de 1974, il arrêta d'écrire ses articles pour des raisons d'esthétisme puis s'attela en 1977 à écrire un programme pour mettre en forme ses textes. Ce dernier vit le jour en 1978 sous le nom de TeX (se prononce tech). Le numéro

<sup>1</sup> [http://www.laas.fr/~sim\\$matthieu/cours/latex2e/](http://www.laas.fr/~sim$matthieu/cours/latex2e/)

<sup>2</sup> <http://www.ctan.org/tex-archive/info/JMPL.ps.gz>

<sup>3</sup> [http://lea-linux.org/software/latex\\_intro.php3](http://lea-linux.org/software/latex_intro.php3)

<sup>4</sup> <http://www.grappa.univ-lille3.fr/FAQ-LaTeX/>

<sup>5</sup> <http://www.via.ecp.fr/formations/1999-00/>

<sup>6</sup> [http://www-cs-faculty.stanford.edu/~sim\\$knuth/](http://www-cs-faculty.stanford.edu/~sim$knuth/)

<sup>7</sup> [http://www-cs-faculty.stanford.edu/~sim\\$knuth/taocp.html](http://www-cs-faculty.stanford.edu/~sim$knuth/taocp.html)

de version de TeX tend vers le nombre  $\pi$  et en est aujourd'hui à 3,14159. Avec plus de vingt ans d'existence, celui-ci se montre extrêmement fiable : le dernier bogue découvert le fut par Peter Breitenlohner et date de 1995. Donald Ervin Knuth<sup>8</sup> aimant la perfection, il en récompensa Peter Breitenlohner par le modique chèque d'un montant de 327,68 \$ ; et au passage, la récompense pour le prochain bogue découvert est élevé à plus de 600 \$ ! En réalité TeX et ces récompenses ne sont pas une exception : l'auteur récompense quiconque découvre une faute dans ses ouvrages par un chèque de 2,56 \$ (il faut bien avouer que la plupart des personnes ne les déposent pas ;-)). L'utilisation de TeX n'étant pas des plus aisées, un ensemble de macros fut créé en 1992 par Leslie Lamport<sup>9</sup> afin de faciliter la rédaction de documents pour les utilisateurs. Cette ensemble de macros se trouve sous l'appellation LaTeX, TeX étant toujours utilisé pour la mise en page. En 1994, des améliorations sont apportées à LaTeX, ainsi que le fusionnement de différentes variantes, par Frank Mittelbach et son équipe LaTeX3<sup>10</sup>. Un nouveau nom est alors donné : LaTeX2&epsilon, c'est de celui-ci que nous parlerons tout au long des différents didacticiels. La dernière version de LaTeX2&epsilon date du premier juin 2001.

### 0.1.3 3. Principe de fonctionnement :

LaTeX n'est en aucun cas un *traitement de texte* classique (WYSIWYG) : le document source est écrit à l'aide d'une entête et de balises avec un simple éditeur (de vi au plus élaboré comme kile). Ce document source portera l'extension **.tex** et sera compilé par la suite afin d'obtenir le document final. Vous pouvez dès ici comprendre que pour utiliser LaTeX une phase d'apprentissage est nécessaire, phase que chacun effectue à son rythme et selon ses besoins. Par contre, une fois habitué à lui, vous aurez du mal à vous en passer. L'utilisation des balises va induire beaucoup d'avantages en plus de la qualité légendaire des documents créés sous LaTeX, et pour en citer quelques uns : l'auteur n'a pas à se soucier dans l'ensemble de la mise en page, ce qui lui permet de s'attacher au fond et non à la forme, la structure sous LaTeX étant très importante, imposera à l'auteur rigueur et cohérence dans son écrit, la modification de documents de très grande taille est tout aussi aisée que celle de documents de petite taille, le portage de LaTeX sur un grand nombre de plateforme dont microsoft windows © ainsi que la stricte utilisation du code ascii permet d'obtenir des sources indépendantes du système d'exploitation. Pour prendre un exemple : il suffit de modifier quelques lignes sur la source du trustbook pour passer tous les titres de rouge à noir ou vert ou peu importe, chose qui pourrait prendre un temps démesuré avec un traitement de texte WYSIWYG.

### 0.1.4 3.1 Compiler un document :

Nous partons d'un fichier source que nous appellerons document.tex pour les besoins de la cause. Dans celui-ci se trouve du "code" LaTeX" ce qui ne le rend pas trop lisible aux yeux des novices. Il va donc falloir compiler celui-ci Pour obtenir le document final. La première question à se poser est le choix du format : postscript ou .pdf. C'est un choix et cela ne regarde que vous. Cependant, je vous avouerai tout de même que le postscript est tout de même de meilleure qualité ;-). Sur le schéma suivant, il vous est indiqué les différents compilateurs ainsi que les fichiers obtenus après compilation. Comme vous pouvez le remarquer

---

<sup>8</sup>[http://www-cs-faculty.stanford.edu/~\sim\\$knuth/](http://www-cs-faculty.stanford.edu/~\sim$knuth/)

<sup>9</sup><http://research.microsoft.com/users/lamport/>

<sup>10</sup><http://www.latex-project.org/>

l'obtention du .pdf nécessite qu'une compilation par pdflatex, alors qu'il y a besoin d'un intermédiaire pour arriver au format postscript final, intermédiaire dont l'extension est **.dvi** (DeVice Independent). Je ne développerai pas davantage ici les différents compilateurs, et me contenterai de ceux que j'utilise. Sachez tout de même que le .pdf obtenu avec pdflatex est beaucoup plus léger que ceux obtenus avec dvipdf et dvipdfm. De plus les liens hypertextes sont actifs avec pdflatex, contrairement à si nous passions par l'intermédiaire dvi. Autant de raisons pour utiliser pdflatex. Passons à la compilation proprement dite, toujours avec mon charmant fichier document.tex. Pour obtenir au final un postscript, il me faudra dans un terminal saisir les commandes suivantes :

```
latex document.tex
dvips document.dvi -o
```

L'option -o ajoutée à la commande dvips permet de dire que la sortie sera écrite dans le fichier document.ps. Sans cette option, elle sera redirigée vers l'imprimante. Il existe beaucoup d'autres options possibles pour la commande dvips. Mon objectif ici n'est pas de vous les expliquer car premièrement je ne les utilise pas (à tort ?), et deuxièmement, d'une distribution à une autre, les options indispensables changent (par exemple, sous Mandriva, on doit pouvoir se passer de "-o"). Revenons maintenant à la commande latex. Dans le cas cité juste au-dessus, il apparaît que nous devons la saisir **une** unique fois. Cela n'est pas toujours vraie (rarement en fait) : dans le cas de création d'une table de matière, d'index, de bibliographie et autres réjouissances, il faudra compiler la source document.tex deux fois, voire trois fois avec cette commande latex. Effectivement si LaTeX sait compter, il n'en demeure pas moins qu'il doit faire plusieurs "passages" pour référencer les choses correctement. Toujours en ce qui concerne cette commande LaTeX : il n'est pas rare que le document source comporte toujours une erreur de code, comme une balise mal fermée. Sachez que plutôt que de vous répondre "ERROR" sans vous en dire plus, il vous délivre toujours un message avec une indication sur la source de l'erreur, voire la ligne où elle se trouve. Les erreurs les plus ennuyeuses à corriger restent les balises non fermées : le numéro de ligne indiqué par LaTeX est alors bien souvent après. En cas d'erreur, LaTeX vous donne la main et attend vos instructions. Elles peuvent être les suivantes :

- **x** : permet de quitter la compilation
- **r** : permet de terminer la compilation, les messages d'erreurs étant ignorés
- **q** : apparemment permet de continuer la compilation, mais j'avoue ne pas avoir tout à fait compris.
- **1 à 9** : permet d'ignorer les x prochains caractères, x étant compris entre 1 et 9
- **i** : permet d'insérer un caractère
- **e** : permet d'éditer le fichier avec vi
- **h** : permet d'avoir des informations supplémentaires sur l'erreur

Vous savez tout maintenant. Passons maintenant à une compilation en vue d'obtenir un format .pdf. Il suffira de saisir simplement la ligne :

```
pdflatex document.tex
```

Et normalement c'est terminé. Cependant plusieurs compilations avec pdflatex peuvent se révéler nécessaires. pdflatex bénéficie en cas d'erreur des mêmes possibilités que la commande latex. Le fichier source doit être compilé pour obtenir un document imprimable. Le format final du document peut alors être **.ps** (postscript : le standard pour l'impression)

ou bien **.pdf** (format propriétaire d'acrobat). L'obtention de ceux-ci nécessitera un intermédiaire : le format **.dvi** (DeVice Independent), visualisable avec `xdvi`. Je vous livre un exemple que vous pouvez télécharger<sup>11</sup> afin de pouvoir le compiler en lisant ces lignes. Pour visionner le fichier `document.ps`, vous pouvez utiliser `gv` ou bien encore `kghostview`, et pour le pdf, `xpdf` sera l'utilitaire de prédilection.

### 0.1.5 3.2 Écrire son document :

Écrire un document source nécessite l'utilisation de plusieurs logiciels :

- **Un éditeur** : ici vous avez pleinement le choix. Ma préférence pour écrire un `.tex` allait pour `kate`, mais depuis peu, l'utilisation de `kile` est loin de me déplaire. Si vous souhaitez utiliser ce dernier, attention à utiliser la touche F2 pour la compilation à la place de l'icône. En effet cette dernière permet de ne pas ignorer les erreurs de code, contrairement à l'icône. C'est tout de même plus satisfaisant un code propre ;-).
- **Un terminal** : pour effectuer la compilation du document `.tex` en `.dvi`, voir en `.ps` ou `.pdf` (à moins d'utiliser `kile`).
- **xdvi** : il vous permettra de visionner le `.dvi` créé à la compilation avec la commande `latex`. Le point important est que `xdvi` interagit avec la création du fichier : si le `.dvi` est modifié, `xdvi` recharge les modifications automatiquement. En conséquence, il ne doit être lancé qu'une fois lors de la première compilation. À noter pour les utilisateurs de KDE qu'ils peuvent aussi utiliser `kdvi`.

Un autre éditeur couramment utilisé est `emacs` avec un module propre à LaTeX, permettant des raccourcis claviers et autres joies. Cela dit, ne l'ayant jamais utilisé, je ne développerai pas plus ici cette possibilité. De même il existe LyX permettant en quelque sorte un WYSIWYG, et encore une fois ne l'utilisant pas... Le conseil que je vous donnerai pour la rédaction de vos premiers documents est d'éviter dans un premier temps des éditeurs tel que `kile` au profit d'un éditeur plus basique tel que `kate`. Le but est simplement de vous forcer à tout écrire afin de mieux assimiler la syntaxe, car je crois que c'est véritablement ainsi que l'on peut assimiler un langage de balise.

---

<sup>11</sup><http://www.truostonme.net/didactels/downloads/document.tar.bz2>