

0.1 Tables des matières (et autres), Index, Bibliographie

0.1.1 1. Table des matières :

0.1.2 1.1 Méthode classique :

Comme cela a déjà été dit dans les bases concernant LaTeX, la table des matières est simplement réalisable en incluant la commande :

```
\tableofcontents
```

La table des matières sera alors introduite à l'emplacement de la commande. Plus intéressant : introduire un élément dans la table des matières sans qu'il soit référencé (numéroté si vous préférez). L'idée est simple : on introduit une commande de titre (`\chapter` par exemple) étoilée (`\chapter*`) afin qu'elle n'influe pas la numérotation, puis on introduit artificiellement la donnée à mettre dans la table des matières. Le code correspondant sera :

```
\addcontentsline{toc}{section}{Ma Section}
\chapter*{Ma Section}
```

Évidemment, vous pouvez remplacer l'option **section** par :

- part
- chapter
- section
- subsection
- subsubsection

cette option définira simplement la mise en forme du titre dans la table des matières. À noter qu'il vous faudra compiler deux fois votre document par LaTeX pour la mettre en place.

0.1.3 1.2 Pour aller plus loin :

Chaque titre (part, chapter, section, etc...) a ce qu'on appelle un niveau de profondeur représenté par un nombre. LaTeX regarde ce niveau de profondeur et le compare à sa référence, le compteur `tocdepth`. le niveau de profondeur du titre est strictement inférieur à `tocdepth`, alors le titre sera affiché dans la table des matières. contraire, s'il est supérieur ou égal, le titre n'apparaît pas dans cette même table des matières. un tableau récapitulant

Titre	Niveau de profondeur
part	-2
chapter	-1
section	0
subsection	1
subsubsection	2
paragraph	3
subparagraph	4

les niveaux de profondeur : À noter que chapter

n'a pas de sens pour la classe article. Comme vous vous en doutez maintenant, il devient possible de décider du contenu de la table des matières. Il suffit simplement pour cela de redéfinir le compteur `tocdepth`, par défaut à 2 pour la classe book ce qui signifie l'affichage des subsections et au-dessus. Ceci se fait par la ligne suivante à mettre avant la commande `\tableofcontents`

```
\setcounter{tocdepth}{valeur}
```

Par exemple, pour le trustbook, le code utilisé est :

```
\setcounter{tocdepth}{1} \tableofcontents
```

Il est aussi possible d'insérer des sommaires localement devant chaque nouvelle partie ou chapitre pour les classes book et report, et devant chaque nouvelle partie ou section pour la classe article. cela, il faut charger le package minitoc dans le préambule du document avec :

```
\usepackage[french]{minitoc}
```

Afin de ne pas avoir de problème, vérifiez que le minitoc.sty que vous avez est une version supérieure ou égale à la version 31 en l'ouvrant, le répertoire étant /usr/share/texmf/tex/latex/minitoc/. tel n'est pas le cas, vous pouvez remplacer votre fichier par celui du site du CTAN : [ici](http://ftp.fdn.fr/pub/CTAN/macros/latex/contrib/supported/minitoc/)¹.

Pour réaliser les tables des matières locales, il faut pour commencer mettre la commande \dominitoc avant la commande \tableofcontents, ainsi que définir la valeur du paramètre minitocdepth (dans le préambule ou dans le document) qui joue le même rôle que le paramètre tocdepth. Par exemple :

```
\usepackage[french]{minitoc}
.....
\begin{document}
\dominitoc
\tableofcontents
.....
\end{document}
```

Il faut ensuite indiquer la commande \minitoc après chaque chapitre (ou partie ou section) où un sommaire est souhaité. , on ne peut pas demander des sommaires pour les parties et chapitres dans un même document. 'est une restriction volontaire afin d'éviter tout document trop lourd. même, pour les documents dont la commande chapter est définie (à savoir les classes book et report), il n'est pas possible de demander un sommaire pour une section, restriction pour ne pas faire encore une fois un document illisible. Pour l'exemple :

```
\chapter{Bureautique}\minitoc
```

\chapter pouvant être remplacé par \part pour les classes book et report, et \part ou \section pour la classe article. Reste la compilation, \tableofcontents en nécessite deux, et l'usage de minitoc une autre. dit il vous faudra faire trois compilations successives pour aboutir à votre document avec des tables de matières correctes (valable aussi avec pdflatex pour compilation). plus je vous conseillerai en cas de changements majeures dans votre document (ajout, suppression, déplacement, etc...) de toujours supprimer le fichier .toc créé lors de compilation précédente, cela évitera à LaTeX de s'emmêler. Un problème connu : seules les versions supérieures ou égale à la version #31 sont compatibles avec le package hyperref. version antérieure conduit à des erreurs de compilation. pouvez vérifier le numéro de version en éditant le fichier minitoc.sty présent dans le répertoire /usr/share/texmf/tex/latex/minitoc/ (première ligne normalement).

0.1.4 2. Table des figures :

Celle-ci se crée facilement avec la commande :

```
\listoffigures
```

¹<http://ftp.fdn.fr/pub/CTAN/macros/latex/contrib/supported/minitoc/>

À noter que lorsque vous utilisez la commande `\caption` lorsque vous introduisez la figure, vous pouvez donner un titre différent à celui qui apparaîtra dans la liste :

```
\caption[titre dans la liste]{titre dans le document}
```

0.1.5 3. Table des tableaux :

Celle-ci se crée facilement avec la commande :

```
\listoftables
```

À noter que lorsque vous utilisez la commande `\caption` lorsque vous introduisez le tableau, vous pouvez donner un titre différent à celui qui apparaîtra dans la liste :

```
\caption[titre dans la liste]{titre dans le document}
```

0.1.6 4. Index :

0.1.7 4.1 Index de base :

Réaliser un index se produit en deux temps :

- Référencer les mots à mettre dans l'index.
- Créer l'index proprement dit.

La première chose à faire est de charger le paquetage permettant la création d'un index. Ceci se fait en plaçant dans l'entête la commande :

```
\usepackage{makeidx}
\makeindex
```

Ensuite, pour référencer un mot dans l'index, il suffit d'introduire la commande `\index{référence}` comme dans l'exemple suivant :

```
je tape mon texte \index{mon_texte}\dots
```

mon_texte apparaîtra alors dans l'index avec le numéro de page correspondante. petit problème existe néanmoins avec les caractères accentués : ces derniers sont considérés comme premier caractère de l'alphabet. Pour remédier à ce problème, il est possible de créer une entrée formatée :

```
je tape mon texte \index{edition@édition}\dots
```

Il est aussi possible de rentrer une sous-entrée à un mot de l'index. Par exemple :

```
Léo Ferré a chanté \og&simles chemises rouges&sim\fg \index{Aragon !Ferré}.
```

Le mot Ferré apparaîtra dans l'index comme une sous-entrée au mot Aragon. Il existe d'autres possibilités pour formater le texte apparaissant dans l'index. Je vous laisse ici le soin de vous reporter à une courte introduction à LaTeX. Arrive maintenant le temps de faire apparaître cette index. La première chose à faire est d'indiquer la commande `\makeindex` dans le document source à l'endroit où vous voulez voir apparaître l'index. Ensuite vient la compilation. Elle se déroule comme suit :

```
latex trustbook.tex
makeindex trustbook
latex trustbook.tex
latex trustbook.tex
dvips trustbook.dvi -o
```

0.1.8 4.2 Index personnalisé :

Vous l'aurez sans doute remarqué, l'index de base n'est pas forcément du plus bel effet : s'il est bien rangé par ordre alphabétique, ce même alphabet n'apparaît pas... plus, nous pourrions souhaiter du gras pour quelques occurrences. LaTeX, fidèle à lui-même, offre comme toujours des possibilités de personnalisation. vous donner un exemple et mieux concrétiser cela, vous pouvez regarder la page donnant les différentes contributions du site sur le TrustBook : c'est ni plus ni moins qu'un index qui y est utilisé :-). La première chose à savoir est qu'il est possible de demander à makeindex d'utiliser un fichier de style. Cela se fait simplement par la commande :

```
makeindex -s mon_index trustbook
```

Vous remarquerez que l'on n'indique pas les extensions dans cette commande. Le fichier `mon_index` doit avoir pour extension `.ist` (autrement dit son vrai nom est `mon_index.ist`), `trustbook` indiquant ni plus ni moins que la source `trustbook.tex`. Nous allons maintenant voir comment écrire ce fichier de style. La première chose à faire pour se lancer dans l'aventure de la personnalisation est, une fois n'est pas coutume, de chercher quelques informations. Pour cela, saisissez dans un terminal la commande suivante :

```
man makeindex
```

La lecture vous informera beaucoup sur les variables. ne rentrerez pas trop dans les détails, et me baserait essentiellement sur ce qui a été fait dans le TrustBook en guise d'exemple explicatif.

L'index est constitué d'entrées. Un mot qui constitue une entrée peut posséder plusieurs sous-entrées et ainsi de suite. exemple, le mot chien peut-être la première entrée, et associé à ce mot, on trouvera en sous-entrées `setter-anglais`, `setter-irlandais`, etc... y a donc une structure hiérarchique entre ces différentes entrées, on dit alors que `chien` va être de niveau 0, `setter-anglais` de niveau 1 et ainsi de suite. Dans notre fichier `mon_index.ist`, nous allons avoir la possibilité de dire à `makeindex` d'ajouter du code entre deux entrées de même niveau (0 par exemple), entre un niveau 0 et un niveau 1 (équivalent à entre un niveau 1 et un niveau 0) et ainsi de suite. Là c'est un peu compliqué je vous l'accorde. premiers essais risquent d'être infructueux, mais il ne faut pas vous démotiver pour autant et pour mieux comprendre le problème lisez le fichier de sortie `.ind` de la commande `makeindex` (dans ma ligne de commande précédente, ce sera `trustbook.ind`). Prenons un exemple pour mieux concrétiser. Dans le `trustbook`, le nom d'auteur joue le rôle d'entrée principale, c'est à dire qu'elle possède un niveau 0. souhaiterai que les auteurs apparaissent en gras. regardant la man de `makeindex`, je vois que `item_0` permet d'ajouter du code entre deux niveaux 0 (à ce moment là, niveaux 1, 2, etc..., ne sont plus considérés puisque nous nous sommes arrêtés au niveau 0) ; me précipite donc dans mon éditeur favoris (kile) pour ouvrir un nouveau fichier que j'appellerai `mon_index.ist`. je vais écrire cette première ligne :

```
item_0 "\n \\item {\bf "
```

où `\n` permet d'effectuer un saut de ligne dans le fichier de sortie (`.ind`). De plus la commande `\item` permet de "hiérarchiser" (on comprendra mieux par la suite), tandis que la commande `\bf` permet les caractères gras. vous l'avez remarqué, il n'y a pas un `"\"` mais deux : `"\"`. En effet les `"\"` sont ignorés, à moins d'être protégés par un autre `"\"`. les commandes s'écrivent donc avec deux anti-slash `"\"`. passage, la commande `\bf` est une commande TeX et non LaTeX, c'est pour cela qu'elle est précédée d'une accolade ouvrante { afin de créer "un groupe" et que les caractères gras ne s'appliquent qu'à ce groupe. 'oubliez

pas non plus que les commandes doivent être entourées de guillemets ". Mais poursuivons... AYE ! J'ai ouvert des balises, mais celles-ci ne sont pas fermées ! Remarquez bien que la commande `makeindex` ne nous insulte pas, elle en laisse le soin à LaTeX :-).

Je continue donc la lecture du man, et Oh Joie !, `item_x1` permet de rajouter du code entre un niveau 0 et un niveau 1 (et réciproquement). En somme je vais donc rajouter la ligne suivante :

```
item_x1 "} \n \subitem"
```

Ici nous avons mis `\subitem`, autrement dit sous-entrée d'un `\item`. C'est plus clair ? :-)
Enfin nous avons notre gras pour les auteurs. Mais misère : si c'est bien trié, l'absence d'un alphabet donne de la peine à s'y retrouver...

Toujours en lisant ce man, nous prenons connaissance de la variable `headings_flag`... érésant... valeur par défaut est 0 (pas d'alphabet), une valeur positive (1) donnera la création d'un alphabet majuscule (sans oublier les symboles et nombres si besoin est), tandis qu'une valeur négative utilisera un alphabet miniscule. allons donc rajouter :

```
headings_flag 1
```

Bien, nous avons déjà beaucoup avancé... manque juste un peu de caractère gras, un petit décalage pour mettre les lettres en évidence. avez compris la méthode, lecture du man et on applique :

```
heading_prefix "\hspace*{-1ex}{\bf\Large "  
heading_suffix "}\n "
```

`hspace*{-1ex}` permet de décaler la lettre de la largeur d'un "x" dans la marge, tandis que `\Large` augmente un peu la taille de la police. Pour résumer la chose, le code utilisé pour le TrustBook est le suivant :

```
headings_flag 1  
heading_prefix "\hspace*{-1ex}{\color{trust}\bf\Large "  
heading_suffix "}\n "  
item_0 "\n \item {\bf "  
item_x1 "} \n \subitem"  
delim_0 ""  
delim_1 "\dotfill"
```

Pour passer à un index avec une unique colonne, je n'ai pas trouvé de solution "propre". Il a fallu modifier la classe (lors de la définition de la commande `index`), en modifiant un `\twocolumn` en `\onecolumn`. De toute façon, à moins d'un cas spécifique, il y a assez peu d'intérêts.

0.1.9 5. Bibliographie :

Ici, je parlerai de la réalisation d'une bibliographie à l'aide de BiBTeX. Ce dernier est spécialement étudié pour cette tâche. Dans un premier temps, il faut écrire la bibliographie dans un fichier portant l'extension **.bib** qui lui est spécialement dédié. contenu va ressembler à cela :

```
@BOOK{motif1,  
author = "Auteur",  
title = "Le titre",
```

```

publisher = "Maison d'\`edition",
year = "Ann\`ee de parution",
}
@ARTICLE{motif2,
author = "Les auteurs",
title = "Le titre",
journal = "Nom du journal",
year = "Ann\`ee de parution",
}
@BOOK{motif3,
author = "Boris Vian",
title = "L'\`ecume des jours",
publisher = "Folio",
year = "1947",
}

```

Nous pouvons appeler ce fichier **test.bib** pour les besoins de la cause.

Comme vous vous en doutez, il existe beaucoup de types d'entrée possibles. en aurez une meilleure idée en consultant les fichiers **/usr/share/texmf/bibtex/bib/base/test.bib** et **/usr/share/texmf/bibtex/** par exemple.

La bibliographie créée, il nous reste à l'introduire dans le document source LaTeX. Cela se fait aisément.

Juste après la commande **\begin{document}** il faut introduire la commande :

```
\bibliographystyle{STYLE}
```

En remplaçant **STYLE** par le style désiré. Les plus utilisés sont en général **plain** et **alpha**. ma part, ma préférence va à **unsrt**, identique à plain dans la forme, il trie en plus par ordre d'apparition les éléments de la bibliographie. existe d'autres styles présents dans **/usr/share/texmf/bibtex/bst/**.

Ensuite, il vous faut dire où introduire la bibliographie à LaTeX. Cela se fait en insérant la commande :

```
\bibliography{test}
```

Attention : test est à remplacer par le nom de votre fichier **sans l'extension**.

Reste à citer la bibliographie dans votre document source. Pour ce, il faut utiliser la commande **\cite{motif}** où motif est la référence dans la bibliographie.

Il peut arriver de vouloir faire apparaître une référence dans la bibliographie, sans pour autant la citer dans le texte. faut alors ajouter la référence bibliographique dans le fichier .bib, puis juste avant la commande **\bibliography{test}**, ajouter :

```
\nocite{motif}
```

où motif est la clé à la référence non-citée. Pour faire apparaître toutes les références non-citées, la commande **\nocite{*}** peut être plus adaptée. Pour terminer, un exemple de fichier source, appelé test.tex :

```

\documentclass[a4paper,11pt]{article}
\usepackage[français]{babel}
\usepackage[latin1]{inputenc}

```

```
\usepackage[T1]{fontenc}
\usepackage{marvosym}
\begin{document}
\bibliographystyle{unsrt}
Nous avons vu blablabla \cite{motif2} et encore blablabla
  \cite{motif1} et blabla.
\nocite{motif3}
\bibliography{test}
\end{document}
```

Reste la dernière phase : la compilation. Elle se déroule ainsi (en suivant le nom des fichiers cités) :

```
latex test.tex
bibtex test
latex test.tex
latex test.tex
dvips test.dvi -o
```

Attention à ne pas citer l'extension du fichier lors de la compilation par BiBTeX ! Une documentation très complète sur BiBTeX du CTAN : [ici](http://ftp.linux.cz/pub/tex/CTAN/documentation/htmlhelp/bibtex-help-html.zip)².

²<http://ftp.linux.cz/pub/tex/CTAN/documentation/htmlhelp/bibtex-help-html.zip>