

Maïeul Rouquette  
avec la participation de Brendan Chabannes et Enimie Rouquette

# (X<sub>E</sub>)L<sup>A</sup>T<sub>E</sub>X

APPLIQUÉ AUX SCIENCES HUMAINES



**LE SEUL LIVRE SUR LATEX SANS UNE SEULE ÉQUATION !**

Version papier : <http://www.atramenta.net/books/latex-sciences-humaines/79>

Mise en page : Maïeul Rouquette

Couverture : Laura Pigeon

Éditeur : Atramenta

Publié sous licence Creative Commons France 3.0 - Paternité - Partage à l'Identique

# Avant-propos

« Du  $\LaTeX$  sed  $\TeX$  »

Proverbe Latin



## Remerciement

Nombreuses sont les personnes à remercier pour leur participation à ce projet. Tout d'abord Brendan Chabannes et ma sœur Enimie qui ont accepté d'écrire plusieurs passages, ainsi que Laura Pigeon pour avoir mis en page la première et la quatrième de couverture.

Christophe Masutti m'a incité à écrire ce livre et je ne me serais pas lancé dans cette rédaction sans ses encouragements. Ma reconnaissance va également à mon directeur de mémoire, Rémi Gounelle, qui a aimablement accepté de prendre sur son temps décanal pour discuter de ce projet.

Deux amis astronomes sont à mentionner : Benjamin pour m'avoir fait découvrir  $\LaTeX$  et Yannick pour m'avoir convaincu, sans le savoir, de l'utiliser pour mon mémoire, au cours de l'une de nos nombreuses conversations électroniques.

Il ne faudrait pas que j'oublie Thomas Boitel, mon correspondant chez Atramenta : il m'a fourni ses conseils avisés d'éditeur dans les derniers mois de préparation de cet ouvrage.

Je tiens également à remercier l'ensemble de la communauté  $\LaTeX$  sans qui ce livre n'aurait pu exister, faute d'objet.

## Au sujet de ce livre

Il va de soi que ce livre a été composé avec X<sub>Y</sub>-L<sup>A</sup>T<sub>E</sub>X. Outre les packages dont il traite, j’ai utilisé le package *minted* pour les citations de code ; les packages *mdframed* et *framed* pour les boîtes colorées.

Ce livre est diffusé sous licence *Creatives Commons - Paternité - Partage des Conditions Initiales à l’Identique 3.0 France*. Sommairement<sup>1</sup>, cela signifie que vous pouvez le diffuser, le dupliquer, le publier et même le modifier si vous respectez deux conditions :

1. que vous citiez mon nom<sup>2</sup> ;
2. que vous offriez les mêmes droits aux destinataires de vos diffusions<sup>3</sup>.

Bien sûr, si vous souhaitez me soutenir, vous pouvez acheter cet ouvrage en version papier, ou simplement m’envoyer un petit mot — vous trouverez aisément comment me contacter sur Internet.

Si vous souhaitez améliorer cette œuvre, soyez le bienvenu. Le code est mis à disposition sur GitHub<sup>4</sup>, un service fonctionnant à l’aide de l’outil de travail collaboratif Git (↗ p. 242, F.1) mais disposant d’une interface d’édition en ligne.

N’hésitez pas à me demander un accès à l’édition du projet !

---

<sup>1</sup>Pour les détails, je renvoie au texte intégral de la licence : <http://creativecommons.org/licenses/by-sa/3.0/fr/legalcode>.

<sup>2</sup>Et que vous ne portiez pas atteinte à mes droits moraux.

<sup>3</sup>Les images de pas et d’éclair servant à indiquer les encarts sont tirées du domaine public et (légèrement) modifiées par mes soins. Elles ne sont donc pas affectées par ces règles. Voir <http://www.openclipart.org/detail/154855/green-steps-by-netalloy> et <http://thenounproject.com/noun/high-voltage/>. L’image de couverture est de Duane Bibby, avec une légère modification. Voir <http://www.ctan.org/lion.html>.

<sup>4</sup>À l’adresse <https://github.com/maieul/latexhumain>.

# Chapitre 1

## Introduction : intérêt de $\LaTeX$ en sciences humaines

### 1.1 Un manque important

Ce livre est le fruit d'un an et demi de travail et d'utilisation quotidienne de  $\LaTeX$  pour la rédaction de notre mémoire de master. Il vient, à nos yeux, combler un vide. En effet, si les ouvrages sur  $\LaTeX$  sont nombreux, rares sont ceux destinés spécifiquement aux sciences humaines.

Si le mot  $\LaTeX$  a déjà été entendu par des oreilles humanistes, il évoque — sauf rares exceptions — au mieux un outil pour les sciences dites dures, au pire la sève d'un arbre ou un plastique aux nombreuses applications.

Certaines raisons pourraient expliquer ce quasi-*vide* :

- Une tendance des chercheurs en sciences humaines à mal connaître ou méconnaître l'outil informatique ;
- $\LaTeX$  paraît *au premier abord* peu convivial ;
- Pendant longtemps  $\LaTeX$  n'a pas disposé d'outils pour gérer convenablement et simplement une bibliographie selon les normes propres aux sciences humaines : notes de bas de page, distinction entre sources primaires et secondaires, etc. ;
- Il fut un temps où la gestion des caractères non latins n'était pas des plus aisées en  $\LaTeX$  ;
- Les éditeurs de sciences humaines acceptent rarement des textes formatés en  $\LaTeX$  parce que les auteurs les rédigent rarement en  $\LaTeX$  parce que les éditeurs les acceptent rarement en  $\LaTeX$  ...

Alors que les chercheurs en sciences humaines sont des spécialistes de l'écrit, durant un temps seuls des logiciels de traitement de texte comme LibreOffice.org ou Microsoft Word ont été utilisés pour la rédaction des travaux universitaires.

Et c'est là un paradoxe : en effet, comme nous allons le voir, ces traitements de texte souffrent de défauts majeurs qui devraient inciter les écrivains à en changer.

## 1.2 Pourquoi L<sup>A</sup>T<sub>E</sub>X ?

### 1.2.1 Inconvénients des traitements de texte

Quand vous rédigez dans votre traitement de texte, comme par exemple Microsoft Word ou LibreOffice.org, celui-ci exécute deux actions simultanées :

- D'une part il stocke dans un fichier la structure logique de votre travail : titres, paragraphes, notes de bas de page etc.
- D'autre part il vous affiche à l'écran le rendu « physique » de votre texte (justification, gras, italiques, etc.), tel qu'un lecteur en disposera.

Pour cette raison on appelle ce type de logiciel WYSIWYG, ce qui en anglais est l'acronyme de *What You See Is What You Get*<sup>1</sup>.

Cette combinaison de deux fonctions différentes dans les traitements de texte entraîne trois conséquences :

- La nécessité d'afficher en temps réel le rendu physique du texte tout en conservant une vitesse élevée du logiciel a pour conséquence une baisse de la qualité typographique. Par exemple :
  - Pour avoir un texte justifié à gauche et à droite, les traitements de texte font varier la taille des espaces entre les mots. Les variations sont parfois considérables, ce qui peut diminuer le confort de lecture. Afin d'éviter ce type d'ennui, les livres classiques coupent les mots en fin de ligne, ce qu'on appelle une césure<sup>2</sup>.
  - Les blancs situés avant certains signes de ponctuation, comme par exemple les points d'exclamation, sont de la même taille que les blancs séparant les mots, alors que les règles typographiques classiques prévoient des blancs plus petits.

---

<sup>1</sup>« Ce que vous voyez est ce que vous produisez ».

<sup>2</sup>La césure ne se fait toutefois pas n'importe où : elle doit respecter des règles propres à chaque langue.

- Le fait qu’un même logiciel s’occupe *à la fois* de l’affichage et de la structure du texte incite à confondre les deux<sup>3</sup> :
  - Une telle pratique pousse à se concentrer sur la forme plutôt que sur le fond et la structure<sup>4</sup>.
  - Les rédacteurs n’utilisent pas toujours la possibilité de séparer sens et forme grâce aux styles. Dans ce cas, lorsque l’on désire changer la forme d’un élément logique, comme le titre d’un chapitre, on doit changer l’ensemble des endroits où cet élément logique est utilisé<sup>5</sup>.
- Les lourds calculs informatiques nécessaires à la mise en forme en temps réel rendent les traitements de textes particulièrement lents comparativement à d’autres logiciels. Cette lenteur est souvent source d’énerverment et de perte de concentration. En outre, ces logiciels exigent bien souvent du matériel récent.

Par ailleurs si les traitements de textes récents disposent d’outils de gestion de bibliographie, ceux-ci manquent en général de souplesse ; c’est pourquoi ils sont rarement utilisés<sup>6</sup>. Ainsi, nombreux sont les rédacteurs à écrire leur bibliographie « à la main » en tapant directement : Nom de l’Auteur, *Titre*, etc. En cas d’erreur, il faut donc corriger l’ensemble des endroits où l’œuvre est citée.

## 1.2.2 Avantages de $\LaTeX$

$\LaTeX$  permet de résoudre l’ensemble des problèmes des traitements de texte. En effet, il sépare deux étapes :

- L’étape de rédaction, qui se passe dans un éditeur de texte. L’auteur frappe son texte et indique par un certain nombre de commandes sa structure (titres, paragraphes, notes de bas de page).
- L’étape de calcul du rendu final se fait seulement ensuite : l’auteur passe son fichier par un compilateur<sup>7</sup>, parfois aussi appelé compositeur<sup>8</sup>. Ce der-

<sup>3</sup>L’auteur de ces lignes est moins sévère envers les traitements de texte que d’autres LaTeXiens : cf. Alain COTTRELL, *Traitements de Texte : Stupides et Inefficaces*, URL : <http://ricardo.ecn.wfu.edu/~cottrell/wp/wp-fr.pdf>.

<sup>4</sup>Toutefois en théorie la formation universitaire en sciences humaines incite à penser *structure et sens d’abord*. Voir un débat sur le blog de l’auteur : URL : <http://geekographie.maiel.net/De-traitement-de-texte-a-Latex#forum20>.

<sup>5</sup>Par exemple, pour une personne qui n’aurait pas utilisé les styles pour désigner ses titres de chapitre, il faudra sélectionner *l’ensemble des titres de chapitres* puis aller dans les menus de mise en forme, etc.

<sup>6</sup>Il est toutefois possible de passer par des outils externes, tels que EndNote ou Zotero.

<sup>7</sup>Qu’on peut sommairement décrire comme un ensemble de scripts informatiques destinés à produire un objet informatique à partir d’un langage plus facile à lire pour les humains.

<sup>8</sup>En réalité, le terme « compositeur » est plus correcte, du point de vue du vocabulaire informatique, que le terme « compilateur ». Toutefois ce dernier est plus souvent utilisé dans la langue courante.

nier programme va lire l'ensemble des commandes du fichier pour produire un nouveau fichier au format PDF<sup>9</sup>.

Cette séparation permet :

- Une qualité typographique supérieure : le compilateur n'ayant pas la contrainte d'un affichage en temps réel, il peut faire des calculs plus lourds : ainsi par exemple  $\LaTeX$  produit des césures typographiques et non pas des blancs à géométrie variable, et équilibre bien mieux la composition du texte.
- Une meilleure séparation du sens et de la forme puisque l'auteur donne uniquement des indications de sens.

En outre  $\LaTeX$  possède un système de gestion de la bibliographie extrêmement puissant qui permet à l'auteur de séparer le contenu de sa bibliographie<sup>10</sup> de son affichage<sup>11</sup>.

Cette séparation entre contenu de la bibliographie et affichage est utile non seulement aux auteurs mais aussi aux éditeurs. En effet, si l'auteur structure correctement sa base de données bibliographique (voir p. 79, 10), l'éditeur peut adapter l'affichage de la bibliographie à ses propres règles : il lui suffit de créer des fichiers de styles bibliographiques suivant une syntaxe simple.

La gestion d'une bibliographie est à la fois l'un des travaux les plus importants en sciences humaines et l'un des plus pénibles, avec de nombreuses sources possibles d'erreurs. Cette simple raison suffit aux yeux de l'auteur à préférer  $\LaTeX$  à un logiciel de traitement de texte<sup>12</sup>.

Un autre avantage de  $\LaTeX$  est la production directe d'un document au format PDF. Lorsque l'on reçoit des documents sous forme numérique, il est bien fréquent qu'à l'ouverture du fichier la mise en page soit perdue, que l'impression soit de mauvaise qualité, ou même qu'il soit tout bonnement impossible d'ouvrir le fichier.

Le *Portable Document Format* permet de pallier ces inconvénients. Il s'agit d'un format ouvert, c'est-à-dire que son créateur, la société Adobe, a publié toutes les spécifications nécessaires à la création de logiciels pouvant lire ce format. Par conséquent, il existe de nombreux lecteurs de PDF utilisables sur un grand nombre de systèmes d'exploitation<sup>13</sup>.

<sup>9</sup>Historiquement  $\LaTeX$  produisait un autre format de fichier : DVI. Mais pour notre propos, cela n'a pas d'importance : dans ce livre nous n'utiliserons que la production de PDF.

<sup>10</sup>Titres, auteurs etc.

<sup>11</sup>Faut-il mettre des *op. cit.*, et si oui où ? faut-il mettre uniquement les initiales des prénoms ou les prénoms en entier ? etc.

<sup>12</sup>L'auteur ainsi que sa sœur se sont décidés à utiliser  $\LaTeX$  dans le cadre de leurs travaux universitaires, et l'élément déclencheur du choix a été la facilité et la souplesse de la gestion bibliographique.

<sup>13</sup>Pour un aperçu des lecteurs libres et gratuits disponibles pour votre système d'exploitation, si vous n'en disposez pas déjà d'un, rendez-vous à la page <http://pdfreaders.org/index.fr.html> (en français).

Le format PDF est conçu pour être lisible de manière universelle. Il embarque en effet dans un même fichier non seulement le texte et les éventuelles images, mais aussi les indications de mise en page et les polices de caractère. Être aussi complet garantit que celui qui visionnera votre document ou bien le lira sous sa forme imprimée verra exactement ce que vous aviez en tête lorsque vous l'avez composé, ce qui constitue un avantage indéniable sur les formats de fichiers tels que celui de Microsoft Word, par exemple. En outre, vous avez la certitude que le PDF que vous conservez est pérenne. Dans la mesure où chacun est libre de concevoir un logiciel permettant la lecture des PDF et que les spécifications sont librement accessibles, vous avez la garantie de ne pas voir le sort de vos publications dépendre de la bonne volonté d'un vendeur de logiciels au cours des années à venir, ce qui fait du PDF un format de choix pour archiver des publications.

La médaille a hélas son revers : format de fichier lisible universellement, le PDF est fort difficile à modifier confortablement : il n'est pas prévu pour cet usage. C'est pourquoi, si vous souhaitez travailler en collaboration sur un ouvrage en utilisant  $\LaTeX$ , il vous faudra travailler différemment (voir p. 241, F).

### 1.2.3 Qu'est-ce qu'un éditeur de texte ?

Nous avons parlé dans les paragraphes précédents de deux types de logiciel, qu'il ne faut pas confondre :

1. Les traitements de texte.
2. Les éditeurs de texte.

Nous avons vu ce qu'étaient les premiers : des logiciels qui s'occupent d'insérer dans un fichier la structure logique d'un texte et de montrer son rendu physique. Les seconds sont simplement des logiciels qui permettent à une personne d'écrire dans un fichier texte et de placer lui même ses commandes de structuration. Toutefois les bons éditeurs de texte font plus que cela, ils aident à la rédaction par différents outils :

- Souvent ils colorient à l'écran les commandes, afin de permettre de mieux les visualiser : c'est ce que l'on appelle la coloration syntaxique.
- Ils proposent des aides pour frapper les commandes les plus fréquentes : raccourcis clavier, boutons, etc.
- Ils offrent parfois un affichage du plan du travail.

Certains de ces éditeurs de texte sont généralistes et adaptés à plusieurs langages informatiques<sup>14</sup>. D'autres sont spécialisés dans tel ou tel langage : ils

---

<sup>14</sup>Par exemple le  $\LaTeX$  et le HTML, ce dernier étant utilisé pour les sites internet.

proposent dans ce cas des outils supplémentaires propres au langage de spécialisation.

Ainsi les éditeurs spécialisés en  $\LaTeX$  proposent des boutons spécifiques afin de lancer le compilateur  $\LaTeX$ .

Pour commencer en  $\LaTeX$  il vous faut donc abandonner votre ancien traitement de texte et choisir un éditeur de texte spécialisé en  $\LaTeX$  : nous en listons plusieurs en annexe (☞ p. 229, B.1).

### 1.3 $\TeX$ , $\LaTeX$ , $X_{\text{T}}\TeX$ et $X_{\text{T}}\LaTeX$ : points communs et différences

Dans ce chapitre nous avons parlé de  $\LaTeX$ . Le titre de ce livre parle pourtant de  $X_{\text{T}}\LaTeX$ . Quelle est la différence ? Voici une brève explication historique, très simplifiée<sup>15</sup>.

1. En 1977 Donal Knuth invente  $\TeX$  qui était un simple compositeur de texte, capable de transformer un texte structuré par des commandes en un texte mis en forme. Avec  $\TeX$  on pouvait également inventer ses propres commandes.
2. L'utilisation de  $\TeX$  était relativement complexe. Leslie Lamport a créé un ensemble de commandes  $\TeX$  pour en simplifier l'usage. Cet ensemble de commandes a permis de former le langage  $\LaTeX$  et le compilateur associé.
3. Par la suite un compilateur dérivé de  $\TeX$  a été créé :  $X_{\text{T}}\TeX$ . Il permet deux choses :
  - Une gestion de l'ensemble des écritures mondiales, par le biais du jeu de caractères Unicode (☞ p. 35, 4.2).
  - Une gestion des nouvelles polices de caractères au format OpenType<sup>16</sup>, apparues au début des années 2000.
4. Pour pouvoir utiliser les commandes de  $\LaTeX$  avec  $\TeX$  on a créé  $X_{\text{T}}\LaTeX$ .

On peut résumer les liens entre  $\TeX$ ,  $\LaTeX$ ,  $X_{\text{T}}\TeX$  et  $X_{\text{T}}\LaTeX$  par le schéma 1.1 (p. 13).

Dans ce livre nous travaillerons sur  $X_{\text{T}}\LaTeX$ . Toutefois comme la plupart de nos propos peuvent s'appliquer indifféremment à  $\LaTeX$  et à  $X_{\text{T}}\LaTeX$ , nous emploierons le terme «  $\LaTeX$  », sauf lorsque nous signalerons une spécificité de  $X_{\text{T}}\LaTeX$ .

<sup>15</sup>Le lecteur curieux trouvera aisément de la documentation plus détaillée sur le sujet, sur Internet notamment.

<sup>16</sup>Ce type de fonte permet, par exemple, une gestion poussée des ligatures entre les caractères.

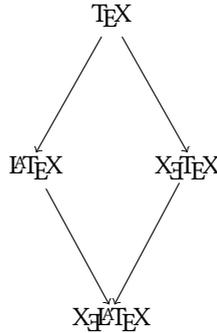


FIG. 1.1 : Les relations entre  $\text{T}_{\text{E}}\text{X}$ ,  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ ,  $\text{X}_{\text{E}}\text{T}_{\text{E}}\text{X}$  et  $\text{X}_{\text{E}}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$

☞ Bien que le sujet soit controversé, il semblerait qu'il faille prononcer le « X » de  $\text{T}_{\text{E}}\text{X}$  comme un « χ » grec, car le nom  $\text{T}_{\text{E}}\text{X}$  viendrait du mot grec « τέχνη » : « art, science ». Ainsi prononcerait-on « latek ».

## 1.4 Publics visés par cet ouvrage

Cet ouvrage vise trois publics distincts.

Tout d'abord, les étudiants et chercheurs en sciences humaines qui ne sont pas rebutés à l'idée d'apprendre un nouvel outil informatique, qui, s'il leur semblera leur faire perdre du temps au début, leur en fera gagner un précieux à l'usage.

Ensuite, les éditeurs de revues et de livres en sciences humaines, pour les inciter à prendre en compte  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  dans leurs choix de format de fichier, et pour leur montrer l'avantage de ce format par rapport aux autres. Nous espérons montrer que  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  permet de résoudre nombre de problèmes d'édition, notamment en ce qui concerne les normes bibliographiques, puisqu'il distingue aisément le sens et la forme.

Enfin les utilisateurs de  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  venant des sciences dites « dures », pour leur montrer les spécificités éditoriales des sciences humaines et la nécessité d'extensions — appelées « package » en  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  — adaptées.

## 1.5 Comment lire ce livre

Cet ouvrage n'est pas un manuel sur  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ . Il se veut plutôt une introduction et ne vise donc à présenter que les bases de  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ . Une fois celles-ci posées le lecteur-rédacteur devrait être en mesure de comprendre la logique de  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  et par la suite être à même de trouver aisément les informations utiles à son projet.

Évidemment, ces bases ne sont pas les mêmes que celles proposées dans d'autres livres d'introduction à  $\LaTeX$ , généralement orientés vers les sciences dites dures. C'est pourquoi vous n'y trouverez pas la manière d'insérer une équation. En revanche nous y détaillons divers éléments abordés souvent trop rapidement dans les autres introductions : ainsi les diverses façons de faire une citation, d'indiquer un changement de langue, la manière d'écrire dans un ou plusieurs alphabets non latins, etc.

Une première partie sera donc consacrée à présenter les principes de fonctionnement de  $\LaTeX$ . Le premier chapitre, sous forme de tutoriel, vise à exposer les concepts essentiels. Les autres chapitres, plus formels, décrivent les outils de base et nécessitent d'avoir compris les concepts. En revanche leur ordre de lecture dépend essentiellement des besoins rédactionnels du lecteur.

Celui-ci pourrait d'ailleurs fort bien commencer par les premiers chapitres de la seconde partie, qui présente les outils de gestion bibliographique en  $\LaTeX$ , en commençant par la constitution de la base de donnée bibliographique.

Une troisième partie cherche à introduire l'ensemble des outils nécessaires pour faciliter la navigation dans le travail final : sommaire, renvois internes, index.

Une quatrième introduit des outils  $\LaTeX$  dont l'auteur a jugé qu'ils pouvaient être particulièrement utiles en sciences humaines.

La séparation entre forme et sens étant au cœur de la logique de  $\LaTeX$ , nous avons jugé important de n'aborder, sommairement, les questions de mise en forme qu'en dernière partie<sup>17</sup>.

Le lecteur nous pardonnera de n'avoir pas mis de conclusion, la nature de ce livre ne s'y prêtant pas. Cependant il trouvera un certain nombre d'annexes, comprenant, outre les indispensables index et bibliographies, diverses informations utiles : comment installer et mettre à jour  $\LaTeX$ , comment trouver de l'aide, une présentation de quelques logiciels autour de  $\LaTeX$ , un glossaire, une présentation d'outils utiles pour le travail à plusieurs sur un même projet.

Chaque chapitre de ce livre commence par une courte introduction mentionnant son objet ainsi que les pré-requis à sa compréhension. Deux types d'encart se situent dans le cours du texte : des encarts « attention », marqués par des éclairs et dont il est inutile de détailler la signification, et des encarts « pour aller plus loin », marqués par des traces de pas et visant à satisfaire la curiosité du lecteur et le plaisir de la digression de l'auteur. Des renvois vers d'autres sections du livre sont indiqués sous la forme ( page, **chapitre.section.sous-section**).

Les épigrammes en tête de chaque partie n'ayant pas de prétentions scientifiques, le savant toléra le léger flou quant à leurs provenance, leurs origines n'ayant pas toujours pu être déterminées avec certitude.

On voudra bien enfin nous excuser de ne pas préciser les numéros de page

---

<sup>17</sup>Exception faite d'un chapitre particulier de la première partie, mais qui aborde d'abord les questions de mise en sens.

lorsque nous renvoyons à un manuel d'un package de  $\LaTeX$  : étant donnée la fréquence avec laquelle certains d'entre eux sont mis à jour, cette information n'aurait guère été pertinente.



## Première partie

# Premiers pas avec L<sup>A</sup>T<sub>E</sub>X

« Les Sabins [...] commencerent la bataille, qui fut aspre  
& dura longuement »

PLUTARQUE, *Romulus*, XXVII, trad. par Jacques AMYOT



# Chapitre 2

## Commencer avec X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X

Nous supposons que vous avez installé  $\text{\TeX}$  (☞ p. 221, A) et un éditeur de texte (☞ p. 229, B) spécialisé en  $\text{\TeX}$ . Voyez en annexe.

La première chose à faire est de vérifier que ce logiciel de traitement de texte enregistre bien en UTF-8<sup>1</sup>. Nous reviendrons plus loin (☞ p. 35, 4.2) sur l'intérêt d'un tel encodage, sachez simplement qu'il permet d'utiliser des signes non latins<sup>2</sup>.

### 2.1 Un premier document

Dans votre éditeur de texte, frappez le code suivant<sup>3</sup> puis cliquez sur le bouton de compilation avec X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X<sup>4</sup> :

↖ Si vous êtes sous Mac, les caractères nécessaires à l'utilisation de  $\text{\TeX}$  ne sont pas visibles directement sur votre clavier. Nous avons listé en annexe la manière de les frapper (☞ p. 222, A.2.1).

```
1 \documentclass[12pt,a4paper]{book}
2 \usepackage{fontspec}
3 \usepackage{xunicode}
4 \usepackage{polyglossia}
5 \setmainlanguage{french}
6
```

---

<sup>1</sup>Le réglage se trouve en général dans les préférences du logiciel, dans une rubrique *enregistrement* ou *encodage* : consultez le manuel de votre logiciel le cas échéant.

<sup>2</sup>Cyrilliques, grecs, sanskrits, hébraïques, etc. Et même extra-terrestres.

<sup>3</sup>Comme nous l'avons expliqué en introduction (p. 11), la coloration que vous voyez ici, si vous lisez la version informatique de ce livre, a un sens syntaxique : ne vous préoccupez pas si elle est différente dans votre éditeur, et ne pensez pas que votre texte apparaîtra ainsi une fois compilé.

<sup>4</sup>Sa position dépend de votre éditeur de texte. Pour le moment vous pouvez vous contenter de ce bouton, mais un jour vous devrez apprendre à faire quelques lignes de commandes : ne vous inquiétez pas, tout sera expliqué.

```

7  \begin{document}
8  \title{Un titre d'ouvrage}
9  \author{Le nom de son auteur}
10
11 \maketitle
12
13 Lorem ipsum dolor sit amet, consectetur adipiscing elit?
14 Morbi commodo ipsum; sed pharetra gravida!
15 Nullam sit amet enim. Suspendisse id : velit vitae ligula.
16 Aliquam erat volutpat.
17 Sed quis velit. Nulla facilisi. Nulla libero.
18
19 Quisque facilisis erat a dui.
20 Nam malesuada ornare dolor.
21 Cras gravida, diam sit amet rhoncus ornare,
22 erat elit consectetur erat, id egestas pede nibh eget odio.
23 % La fin du document
24 \end{document}

```

Regardez le PDF obtenu : afin de comprendre les principes de base de  $\LaTeX$  nous allons lire le code que vous avez copié et le commenter ligne par ligne.

## 2.2 Structure d'un document $\LaTeX$

### 2.2.1 La classe du document

La première ligne `\documentclass[12pt]{book}` déclare la classe du document, ici **book**. Une classe correspond à un choix éditorial — mise en page et organisation générale du document. Le choix de la classe influence entre autres :

- Le nombre de niveaux de titre disponibles.
- Les marges appliquées.
- Les en-têtes et pieds de page.

Il existe en standard plusieurs classes de documents : citons **book**, pour rédiger un livre ; **article** pour un article (si, si !); **beamer** pour une présentation sous forme de diapositives à projeter. Dans cet ouvrage, nous aborderons essentiellement les deux premières, nous ferons également une brève présentation de **beamer** (☞ p. 181, 21).

Tout document  $\LaTeX$  doit commencer par une déclaration de classe, avant toutes autres lignes. La syntaxe est : `\documentclass[⟨options⟩]{⟨classe⟩}`.

⚡ Dans la suite de ce livre, tout texte situé entre crochets (*ainsi*), doit être remplacé dans votre fichier `.tex` par une valeur textuelle.

Les options viennent spécifier certaines propriétés de la classe. Dans notre exemple, nous précisons que la taille de la police du texte courant doit être de 12 pt et que le format du papier est A4. Vous pouvez indiquer plusieurs options, en les séparant par des virgules.

Voici quelques options disponibles et utiles en sciences humaines :

<b>10pt</b>	pour une police de base en 10 pt.
<b>11pt</b>	pour une police de base en 11 pt.
<b>12pt</b>	pour une police de base en 12 pt.
<b>onecolumn</b>	pour un texte sur une seule colonne. C'est le cas par défaut sur les classes citées.
<b>twocolumn</b>	pour un texte sur deux colonnes.
<b>oneside</b>	pour une impression en recto seulement.
<b>twoside</b>	pour une impression en recto-verso <sup>5</sup> .

Nous en préciserons d'autres au fur et à mesure de l'ouvrage, lorsque les notions requises auront été abordées.

## 2.2.2 L'appel aux packages

Voyons les trois lignes suivantes :

```

2 | \usepackage{fontspec}
3 | \usepackage{xunicode}
4 | \usepackage{polyglossia}

```

Il s'agit, comme vous auriez pu le deviner, d'appel à des packages<sup>6</sup>. Un package est un ensemble de fichiers qui ajoutent des fonctionnalités à L<sup>A</sup>T<sub>E</sub>X, c'est l'équivalent d'un plugin sous Firefox.

Le premier package est *fontspec*. Il est utile à X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X pour une typographie avancée, notamment pour disposer d'accents dans le document PDF produit.

Le second package est *xunicode*. Il permet de gérer l'unicode, autrement appelé UTF-8<sup>7</sup>. Il nous permet d'utiliser des caractères non latins (☞ p. 35, 4.2) dans le document `.tex`.

<sup>5</sup>Cette option et la précédente servent essentiellement pour les questions de reliure. En effet, l'option `twoside` produit des marges gauches et droites de tailles différentes, selon que la page est recto ou verso. En outre la numérotation des pages est inscrite à gauche et à droite en alternance.

<sup>6</sup>Nous avons fait volontairement le choix de ne pas traduire ce terme, pour éviter des confusions.

<sup>7</sup>En réalité UTF-8 n'est pas tout à fait Unicode, mais une implémentation de ce dernier. Toutefois, pour simplifier, nous assimilons les deux.

☞ On trouve sur internet de la documentation expliquant qu'il faut, pour insérer un « é », frapper `\'e`, et qui fournit une liste de commandes de caractères accentués.

Ceci était vrai fut un temps. Il est depuis longtemps inutile d'apprendre ces commandes : vous pouvez tranquillement frapper vos lettres accentuées de manière « normale ».

Le troisième, *polyglossia*, permet de gérer facilement un document multilingue (☞ p. 33, 4) et les changements typographiques qu'un tel document implique.

Ces trois packages sont propres à Xe<sub>La</sub>TeX : ils ne fonctionnent pas (☞ p. 12, 1.3) avec E<sub>La</sub>TeX.

Certains packages peuvent recevoir des options qui modifieront leur comportement standard. La syntaxe est alors `\usepackage[options]{package}`.

Tout au long de cet ouvrage, nous aborderons divers packages.

↪ Dans la suite de ce livre, chaque fois que nous décrivons les fonctionnalités d'un package, nous suppose que le package a été chargé au préalable dans le préambule par la commande `\usepackage{package}`.

☞ Lorsque nous parlons d'un package, nous renvoyons souvent à son manuel. Il existe une façon très simple de trouver le manuel d'un package en passant par le terminal : nous expliquons en annexe comment faire (☞ p. 251, I.1).

## 2.2.3 Le français, langue par défaut

Tout de suite après, la ligne `\setmainlanguage{french}` indique que nous utilisons comme langue principale du document le français (☞ p. 33, 4), et que le compositeur de texte devra donc prendre en compte la typographie française. Cette ligne n'est compréhensible par le compilateur que parce que nous avons chargé *polyglossia* au préalable.

☞ Vous entendrez peut-être parler du package *babel*. Il est très souvent utilisé à la place de *polyglossia*, notamment parce qu'il est plus ancien. Toutefois, nous avons choisi pour notre part de nous limiter à *polyglossia*, puisque c'est lui que nous avons utilisé pour nos travaux et qu'il possède plus de fonctionnalités, par exemple pour les langues notées dans un alphabet non latin.

Vous trouverez aisément des informations à propos de *babel* sur Internet.

## 2.2.4 Le corps du document

Tout ce que nous avons vu jusqu'à présent, avant `\begin{document}`, fait partie de ce que l'on appelle le préambule du document. Ce sont des informations

qui n'apparaissent pas dans le document final, mais qui sont utiles à sa composition, autrement dit des méta-données. Tout les packages que vous souhaitez utiliser sont à appeler dans le préambule.

Tout ce qui se trouve entre la ligne `\begin{document}` et `\end{document}` constitue le corps du document, le contenu proprement dit de votre travail.

Enfin, rien de ce qui se trouve après `\end{document}` n'est analysé par le compilateur. Vous pouvez donc y mettre ce que vous voulez, cependant nous ne vous le conseillons pas.

## 2.2.5 Titre, auteur et date : la notion de commande

```

8 | \title{Un titre d'ouvrage}
9 | \author{Le nom de son auteur}
10 | \date{Une date}
11 | \maketitle

```

Les trois premières de ces lignes définissent respectivement le titre (`\title`), l'auteur (`\author`) et la date (`\date`) du travail. En ce qui concerne cette dernière, ne pas l'indiquer revient à indiquer celle du jour de la compilation, et il faut indiquer `\date{}` pour qu'aucune date n'apparaisse.

La dernière ligne affiche ces informations. Si votre document est de classe **book**, alors le compilateur les dispose sur une page à part. S'il est de classe **article**, il les affiche sans provoquer de saut de page.

On peut déroger à cette règle en passant une option à l'appel de classe (☞ p. 21, 2.2.1).

**notitlepage**      pour ne pas avoir de page de titre spécifique.

**titlepage**        pour avoir une page de titre spécifique.

Nous pouvons maintenant définir la notion de commande. Une commande est un bout de code qui est interprété par le compilateur pour effectuer une suite d'opérations, c'est un raccourci d'écriture. Ici la commande `\maketitle` affiche les informations tel que le titre, la date et l'auteur du travail, informations que le compilateur a apprises grâce aux commandes utilisées au préalable.

Une commande peut prendre des arguments, certains facultatifs, d'autres obligatoires. Ces arguments modifient son comportement. Une commande s'appelle avec la syntaxe : `\nom[⟨opt1⟩][⟨...⟩][⟨optn⟩]{⟨obl1⟩}{⟨...⟩}{⟨obln⟩}`.

Entre crochets sont indiqués les arguments optionnels, entre accolades les arguments obligatoires. Ces arguments peuvent eux même contenir des commandes.

L'ordre des arguments dépend de chaque commande, et les arguments optionnels ne sont pas systématiquement avant les arguments obligatoires : ils peuvent être après ou s'intercaler entre. Notez que certaines commandes ne prennent pas d'argument : c'est le cas ici de `\maketitle`.

⚡ À chaque crochet ou accolade ouvert doit correspondre un crochet ou accolade fermé, faute de quoi vous risquez de provoquer une erreur de compilation.

La grande force de  $\LaTeX$  est justement l'utilisation de commandes afin d'éviter la répétition des tâches fréquentes. C'est pourquoi nous apprendrons à définir nos propres commandes (☞ p. 53, 7).

## 2.2.6 Le corps du texte : la manière de rédiger

### Analyse de notre exemple

Regardez maintenant les lignes suivantes et leur résultat à la compilation.

```

13 | Lorem ipsum dolor sit amet, consectetur adipiscing elit?
14 | Morbi commodo; ipsum sed pharetra gravida!
15 | Nullam sit amet enim. Suspendisse id : velit vitae ligula.
16 | Aliquam erat volutpat.
17 | Sed quis velit. Nulla facilisi. Nulla libero.
18 |
19 | Quisque facilisis erat a dui.
20 | Nam malesuada ornare dolor.
21 | Cras gravida, diam sit amet rhoncus ornare,
22 | erat      elit consectetur erat, id egestas pede nibh eget odio.
```

Nous pouvons constater plusieurs choses.

- Une ligne vide produit un changement de paragraphe. Plusieurs lignes vides produisent un seul changement de paragraphe.
- Un retour à la ligne en revanche se comporte comme une espace<sup>8</sup>. C'est une grande différence avec les logiciels WYSIWYG, qui traduisent automatiquement un retour à la ligne par un saut de paragraphe.
- Plusieurs espaces à la suite produisent une seule espace.

Vous connaissez donc les règles de bases de la rédaction d'un texte en  $\LaTeX$ .

### Allons plus loin

Nous l'avons dit,  $\LaTeX$  produit une mise en page et une typographie plus correctes qu'un logiciel de type WYSIWYG. Il est cependant nécessaire de lui fournir un code correct, afin qu'il puisse déterminer comment typographier.

$\LaTeX$  produit automatiquement une espace fine devant les signes de ponctuation double, ! ; ? principalement, comme il se doit en bonne typographie

---

<sup>8</sup>En matière de typographie, ce terme est féminin.

française<sup>9</sup>. Toutefois, nous recommandons d'insérer des espaces dans le fichier `.tex` avant ces signes de ponctuation double, pour le confort de lecture.

⚡ Les espaces avant les signes de ponctuation double sont une spécificité de la typographie française. Il ne sont généralement pas présent dans les autres langues. C'est pourquoi, si vous écrivez dans une autre langue que celle de Molière, il ne faut pas mettre ces espaces. À vous donc de choisir si vous les mettez ou non dans votre fichier source, sachant que  $\LaTeX$  les insérera pour vous le cas échéant, mais ne les supprimera pas dans les langues autres que le français.

En revanche *il est obligatoire de mettre une espace après chaque signe de ponctuation*. Pour ce qui est des points de suspension, il est mieux de ne pas frapper trois points à la suite, mais d'utiliser la commande `\ldots` qui espacera correctement les points<sup>10</sup>.

En ce qui concerne les guillemets, une partie sera consacrée plus tard à l'art et la manière de faire des citations (p. 45, 6.1) en  $\LaTeX$ . Nous n'en parlons donc pas maintenant.

Prêtons attention à certaines lettres ligaturées comme `œ` et `æ`. À la différence de la plupart des traitements de texte,  $\LaTeX$  ne remplace pas automatiquement les suites `oe` et `ae` par `œ` ou `æ`. Il faut donc frapper soi-même ces caractères, ou configurer son éditeur pour qu'il effectue ce remplacement.

Signalons également trois types de tirets :

1. - qui produit un tiret simple (-), utilisé pour les mots composés ;
2. -- qui produit un tiret demi-cadratin (–), en théorie à utiliser pour séparer une page de nombres ;
3. --- qui produit un tiret cadratin (—), pour des incises<sup>11</sup>.

Enfin, il est parfois utile d'insérer une espace insécable, pour éviter que deux mots se trouvent séparés par un retour à la ligne, par exemple entre un nom de souverain et son numéro de règne : « Jean xxxiii ». L'espace insécable est produit par le caractère `~`.

Par ailleurs, comme vous avez pu le constater,  $\LaTeX$  interprète de manière spécifique un certain nombre de caractères : `\{ } ~`, à quoi nous ajoutons `%` & `$`<sup>12</sup>.

Comment faire si nous désirons afficher un de ces caractères ? Il faut les faire précéder du caractère `\`. Ainsi pour insérer le caractère `%`, il faut écrire `\%`.

Trois exceptions toutefois :

<sup>9</sup>Une espace fine est une espace plus petite qu'une espace normale.

<sup>10</sup>Il est tout à fait possible de configurer l'éditeur de texte pour qu'il remplace automatiquement trois points à la suite par cette commande.

<sup>11</sup>Certains éditeurs préfèrent utiliser des tirets demi-cadratin.

<sup>12</sup>Nous ne verrons pas l'utilité  $\LaTeX$  de tout ces caractères, certains servant essentiellement à rédiger des formules mathématiques.

\	qui s'insère avec la commande <code>\textbackslash</code> ;
~	qui s'insère avec la commande <code>\textasciitilde</code> ;
^	qui s'insère avec la commande <code>\textasciicircum</code> .

## 2.2.7 Un commentaire

La ligne suivante est :

```
23 | %La fin du document
```

Il existe en  $\text{\LaTeX}$  une règle simple : tout ce qui se trouve à droite d'un signe % est un commentaire. C'est-à-dire qu'il n'est pas interprété par le compilateur et n'apparaît donc pas dans le document final.

Nous conseillons de se servir des commentaires pour indiquer les grandes structures du document et pour commenter les commandes que vous créez vous-mêmes (☞ p. 53, 7).

Vous pouvez aussi vous en servir, par exemple, pour faire un commentaire à usage personnel ligne à ligne d'un texte que vous traduisez.

En revanche, nous vous déconseillons de l'utiliser pour des notes personnelles lors de la rédaction. Nous vous indiquerons plus loin comment définir une commande personnalisée afin de générer un fichier qui les affiche, pour une relecture, et une autre qui les masque, pour le document final (☞ p. 55, 7.1).

## 2.2.8 La notion d'environnement

Nous avons vu jusqu'à maintenant les notions de package, préambule, commande. Il nous reste à en définir une dernière : celle d'environnement .

Un environnement est une portion de document ayant une signification spécifique et qui par conséquent subit un traitement spécifique. Par exemple, pour indiquer une citation, une liste, etc. Nous découvrirons au fur et à mesure des environnements.

On marque le début d'un environnement  $\langle nom \rangle$  par `\begin{⟨nom⟩}` et on le termine `\end{⟨nom⟩}`.

Dans la classe **article** il existe un environnement utile : **abstract**. On place dans cet environnement un résumé de l'article :

```
1 | \begin{abstract}
2 | Écrivons ici un résumé de l'article.
3 | \end{abstract}
```

Il est possible d'imbriquer des environnements :

```
1  \begin{1}  
2  blabla blab  
3  \begin{2}  
4  blabl blab  
5  \end{2}  
6  blabl  
7  \end{1}
```

En revanche il n'est pas possible de superposer des environnements : ainsi le code suivant ne fonctionne pas et produit une erreur lors de la compilation.

```
1  \begin{1}  
2  blabla blab  
3  \begin{2}  
4  blabl blab  
5  \end{1}  
6  blabl  
7  \end{2}
```

## 2.2.9 Conclusion

Vous avez appris ici les principales notions de  $\LaTeX$ . Pour l'instant, cela doit sans doute paraître très flou : mais au fur et à mesure de votre lecture, vous comprendrez mieux<sup>13</sup>...

---

<sup>13</sup>Enfin, nous l'espérons !



# Chapitre 3

## Structurer son travail

*Après avoir découvert les bases de  $\LaTeX$ , apprenons la manière de structurer son travail.*

### 3.1 Différents niveaux de titres

$\LaTeX$  propose par défaut six ou sept niveaux de titres, selon la classe choisie. Pour introduire un titre dans  $\LaTeX$  – en dehors du titre du travail – il suffit d'utiliser une commande de titre qui possède la syntaxe suivante : `\langle titre \rangle [ \langle titre court \rangle ] { \langle titre long \rangle }`.

Le titre court est facultatif, comme l'indique le fait qu'il soit entre crochets (☞ p. 23, 2.2.5). Il sert pour la table des matières (☞ p. 137, 17) et, éventuellement, pour les en-têtes des pages (☞ p. 191, 22).

Évidemment `\langle titre \rangle` doit être remplacé par le type de titre. Voici les niveaux de titre disponibles, du plus général au plus détaillé. Plus un titre se trouve haut dans la hiérarchie, plus son numéro de niveau est faible.

Commande	Sens	Numéro de niveau
<code>\part</code>	Titre de partie	-1
<code>\chapter</code>	Titre de chapitre	0
<code>\section</code>	Titre de section	1
<code>\subsection</code>	Titre de sous-section	2
<code>\subsubsection</code>	Titre de sous-sous-section	3
<code>\paragraph</code>	Titre de paragraphe	4
<code>\subparagraph</code>	Titre de sous-paragraphe	5

Quelques remarques importantes :

- Le niveau `\chapter` n'existe que dans la classe **book** ;

- Chaque niveau de titre se voit attribuer un numéro. Ce numéro sert lors de l’affichage de la table des matières pour définir sa profondeur. (☞ p. 138, 17.2)
- Les niveaux dont les numéros sont inférieurs à 1 provoquent un changement de page.
- Les niveaux dont les numéros sont supérieurs à 3 ne provoquent pas de changement de paragraphe. Les titres sont positionnés en « lettrine ».

### 3.1.1 Des titres non numérotés

Par défaut, tous les titres sont automatiquement numérotés<sup>1</sup>. Il est possible d’obtenir un titre non numéroté, en faisant suivre le nom de la commande d’un astérisque : `\chapter*{<Un chapitre non numéroté>}`.

Toutefois un titre non numéroté ne sera pas ajouté à la table des matières (☞ p. 137, 17).

Pour contourner ce problème, il faut utiliser la commande : `\addcontentsline{toc}{<1>}{<2>}`, où :

`<1>` est le type de titre ;

`<2>` est le texte du titre ;

Un exemple sera plus parlant :

```
1 | \addcontentsline{toc}{chapter}{Introduction}
2 | \chapter*{Introduction}
```

 Le lecteur alerte se demandera sans doute pourquoi il est nécessaire de mettre `toc` comme premier argument. Cela correspond à l’extension du fichier qui contiendra la table des matières : nous renvoyons au chapitre dédié à ce sujet (☞ p. 137, 17).

## 3.2 Structurer ses fichiers

Jusqu’à maintenant, vous aviez tout mis dans un seul fichier. Une fonctionnalité intéressante de  $\text{\LaTeX}$  est la possibilité d’appeler dans un fichier d’autres fichiers, pour ainsi séparer son travail en plusieurs fichiers, chacun contenant une partie seulement du document final.

Par exemple, il est possible de faire un fichier par chapitre d’un mémoire, ou encore par texte cité dans un exemplier. Seul un fichier « père » est compilé, ce document appelle des fichiers « fils ».

Pourquoi procéder ainsi ?

---

<sup>1</sup>Nous verrons plus loin comment changer la numérotation (☞ p. 201, 23.3.1).

- Pour pouvoir changer plus aisément l’ordre des parties.
- Pour pouvoir « recycler » plus facilement certaines parties.
- Pour pouvoir compiler seulement certaines parties.

Concrètement, comment fait-on ?

1. Le fichier « père » doit systématiquement commencer par un appel de classe, et contenir `\begin{document}` et `\end{document}`.
2. Les fichiers « fils » ne doivent contenir aucun appel de classe, ni les commandes `\begin{document}` et `\end{document}`.
3. Ils sont inclus dans le fichier « père » par l’une des commandes suivantes :
  - `\include{chemin-du-fichier}`, qui entraîne systématiquement un saut de page.
  - `\input{chemin-du-fichier}`, qui n’entraîne pas de saut de page.

La commande `\input`, contrairement à `\include`, peut aussi être appelée dans un fichier « fils », voire dans un fichier « petit-fils » etc.

Nous conseillons de mettre l’ensemble des appels à des packages dans un fichier à part. Ainsi, vous pouvez disposer d’un jeu de packages pour tout vos documents : il suffit d’appeler à chaque fois ce fichier.

### 3.2.1 Comment indiquer le chemin du fichier

La notion de chemin de fichier en informatique renvoie à l’arborescence des dossiers sur un ordinateur.

En  $\text{\LaTeX}$ , le chemin du fichier se compte à partir du fichier « père », celui qui est compilé, y compris lorsqu’on procède à une inclusion dans un fichier « fils ».

On indique le chemin du fichier en séparant chaque dossiers par  $/^2$ . Ainsi, si nous souhaitons inclure le fichier nommé `c.tex` situé dans le dossier `b`, lui même situé dans le dossier `a`, qui se trouve à côté du fichier « père », il faut que nous mettions dans notre fichier « père » : `\input{a/b/c}` ou bien `\include{a/b/c}`.

⚡ Il est déconseillé d’avoir des caractères spéciaux dans le nom des dossiers et des fichiers.

Nous conseillons de mettre les fichiers « fils » dans un ou plusieurs sous-dossier.

---

<sup>2</sup>Cette norme s’applique même sous Windows, qui sépare traditionnellement les répertoires par des `\` dans les chemins.

### 3.3 La classe **book** : structuration globale du document

La classe **book** propose, en plus des niveaux de titres, une manière de structurer en quatre parties son travail : préambules (avant-propos, sommaire, introductions etc.); corps du travail ; appendices ; outils de navigation (index, glossaires, bibliographie, tables des matières, etc.).

Chacune de ces parties est indiquée par une commande initiale, respectivement : `\frontmatter` ; `\mainmatter` ; `\appendix`<sup>3</sup> ; `\backmatter`.

Cette structuration en parties globales a un impact sur la présentation des numéros de page (romains ou arabes) et sur la numérotation des titres.

Ainsi, par défaut :

<code>\frontmatter</code>	donne des titres non numérotés mais présents dans la table des matières. En outre les numéros de pages sont en chiffres romains minuscules.
<code>\mainmatter</code>	donne des titres numérotés. La numérotation des pages est ré-initialisée et est en chiffre arabe.
<code>\appendix</code>	affiche les numéros de chapitres sous forme de lettres majuscules. Le texte « chapitre » est remplacé par « appendice ».
<code>\backmatter</code>	supprime les numéros de chapitres tout en présentant les chapitres dans la table des matières.

---

<sup>3</sup>Cette commande existe aussi dans la classe **article**.

# Chapitre 4

## Gérer les langues avec *Xunicode* et *Polyglossia*

*Dans ce chapitre nous verrons la manière de signaler à L<sup>A</sup>T<sub>E</sub>X les changements de langue, ainsi que les méthodes pour écrire en caractères non latins.*

### 4.1 Indiquer les changements de langue

#### 4.1.1 Pourquoi indiquer les changements de langue ?

Voici les principales raisons :

1. Chaque langue a ses propres règles typographiques (espacement avant et/ou après les signes de ponctuation, par exemple). Indiquer la langue courante permet donc à L<sup>A</sup>T<sub>E</sub>X d'adapter sa typographie.
2. Chaque langue a ses propres règles de césure des mots : indiquer la langue permet d'avoir une césure correcte.

#### 4.1.2 Commandes et environnements de changement de langue

##### Langue principale et langues secondaires

Nous avons vu que dans le préambule on indiquait le français comme langue principale par la commande : `\setmainlanguage{french}` (☞ p. 22, 2.2.3).

Évidemment on pourrait indiquer une autre langue : par exemple si vous écrivez votre travail en grec : `\setmainlanguage{greek}`.

On pourrait même préciser qu'il s'agit du grec ancien<sup>1</sup> :

---

<sup>1</sup>L'auteur de ces lignes aime retarder de quelques mondialisations, c'est pourquoi il préfère le grec ancien à l'anglais.

```
1 | \setmainlanguage[variant=ancien]{greek}
```

On trouve une liste des langues disponibles et de leurs variantes dans la documentation de *polyglossia*<sup>2</sup>. Dans le cas où votre langue n'est pas disponible, trois solutions s'offrent à vous :

- regardez si le package *babel* ne peut rien faire pour vous ;
- écrivez à l'auteur<sup>3</sup> en lui présentant le nom de la langue et ses règles typographiques et de césure et demandez lui gentiment d'ajouter une langue à *polyglossia* ;
- ne respectez pas les bonnes raisons d'indiquer les changements de langue.

Pour pouvoir indiquer des changements de langue, il faut déclarer dans le préambule les langues secondaires : `\setotherlanguage[<options>]{<codelang>}`, où *<codelang>* est remplacé par le code de la langue, par exemple *greek*.

Les options sont principalement les variantes, mais il peut également s'agir d'options d'affichage des nombres ou des dates : voir le manuel de *polyglossia*<sup>4</sup>.

### 4.1.3 Indiquer un changement de langue

Il existe deux manières d'indiquer un changement de langue.

#### Par une commande

On peut le faire par une commande `\text{<codelang>}[<options>]{<texte dans une autre langue>}` par exemple<sup>5</sup> :

```
1 | \textgreek[variant=ancien]{Ἐν ἀρχῇ ἦν ὁ λόγος}
```

#### Par un environnement

Pour des textes plus longs, il peut être intéressant d'utiliser plutôt un environnement *<codelang>* :

```
1 | \begin[variant=ancien]{greek}
2 | Ἐν ἀρχῇ ἦν ὁ λόγος,
3 | καὶ ὁ λόγος ἦν πρὸς τὸν θεόν,
4 | καὶ θεὸς ἦν ὁ λόγος.
5 | \end{greek}
```

<sup>2</sup>François CHARETTE, *Polyglossia : A Babel Replacement for XeLaTeX*.

<sup>3</sup>Du package, pas de ces lignes.

<sup>4</sup>François CHARETTE, « Language-specific options and commands », dans : *Polyglossia : A Babel Replacement for XeLaTeX*.

<sup>5</sup>Jean 1, 1.

### 4.1.4 Le problème du latin

Dans le package *polyglossia* le latin suit la typographie anglaise. Ainsi le francophone aura quelques soucis s'il désire des espaces avant les signes typographiques doubles dans un environnement `latin` ou dans une commande `\textlatin`. Pourtant il pourrait vouloir utiliser cet environnement ou cette commande, afin d'avoir un respect des césures latines.

Pour ce faire, il devra redéfinir dans le préambule l'environnement `latin` par le code suivant :

```
1 | \renewenvironment{latin}{\begin{hyphenrules}{latin}}%
2 | {\end{hyphenrules}}
```

 La commande `\renewenvironment` redéfinit un environnement, dans le cas présent `latin`. Le deuxième argument de la commande indique ce qui se passe lorsque l'on ouvre l'environnement, le troisième argument ce qui se produit lorsqu'on l'on ferme l'environnement. L'environnement `hyphenrules` indique un changement des règles de césure.

## 4.2 Saisir des textes en caractères non latins

Ce que nous allons expliquer maintenant n'a en réalité pas grand chose à voir avec  $\LaTeX$ . Il s'agit en fait d'un problème plus général à l'informatique : comment écrire dans des caractères non latins ? Nous allons ici expliquer la mauvaise méthode, puis la bonne méthode.

Nous commencerons par un peu d'explications techniques très simplifiées : que les puristes nous pardonnent.

### 4.2.1 Les jeux de caractères : ou comment se servir de nombres pour autre chose que des mathématiques

Au départ, un ordinateur ne manipule que des nombres. Mais les ordinateurs servant aux humains, ceux-ci leurs ont appris à « stocker » des caractères, en associant des lettres à des nombres.

Cependant les premiers ordinateurs ayant été développés par des Anglo-Saxons, on n'a attribué des nombres qu'à 127 caractères, ce qui suffisait largement pour écrire en anglais et ajouter des caractères spécifiques, comme les accolades informatiques<sup>6</sup>. Le jeu de caractères connu sous le doux nom d'ASCII<sup>7</sup> a ainsi vu le jour.

<sup>6</sup>Celles dont vous vous servez pour les commandes  $\LaTeX$ .

<sup>7</sup>*American Standard Code for Information Interchange*.

Un jour d'autres peuples que les Anglo-Saxons ont voulu écrire avec un ordinateur et ont souhaité frapper leurs propres caractères. Par exemple les Européens occidentaux ont voulu taper des accents, des cédilles, des trémas et autres joyeusetés. On a donc créé un nouveau système de codage pour représenter les caractères latins occidentaux, en attribuant des nombres à d'autres caractères. On a ainsi formé le jeu de caractères ISO-8859-1.

D'autres ont voulu pouvoir frapper dans leur alphabet, et c'est ainsi que furent inventés des jeux de caractères comme ISO-8859-5 pour le cyrillique. En outre certaines entreprises inventèrent leurs propres manières de stocker des caractères : ainsi Apple inventa MacRoman et Microsoft Windows-1252<sup>8</sup>.

Mais certaines personnes souhaitaient mélanger des caractères de divers alphabets : par exemple écrire tantôt en grec, tantôt en cyrillique, tantôt en caractères latins. Comment faire ? Pendant longtemps, la technique utilisée<sup>9</sup> consistait à écrire dans un jeu de caractères donné, typiquement ISO-8859-1, mais en utilisant une police qui affichait le texte dans un autre alphabet.

Par exemple, pour écrire le caractère grec «  $\alpha$  » on écrivait le caractère latin « a » et on le faisait afficher dans la police SPIonic. Cette méthode posait — et pose encore — de nombreux problèmes :

- elle nécessitait que la police soit présente sur toutes les machines de travail ;
- ne stockant pas l'information exacte sur le caractère, puisqu'elle utilise un code pour désigner autre chose que ce qu'il devait désigner, elle ne permettait pas de faire aisément des recherches ;
- avec  $\LaTeX$ , étant donné que nous ne sommes pas dans un système WYSIWIG, elle rendait la rédaction et la relecture extrêmement pénible ;
- elle était un non-sens informatique et logique.

Une métaphore simple explique le problème : supposons que vous vouliez une maison en brique rouge. Que diriez-vous si votre entreprise de maçonnerie vous posait du parpaing, puis le peignait en rouge pour faire croire que c'est de la brique ? Voilà le problème fondamental de cette méthode : elle fait prendre du parpaing (la lettre « a ») pour de la brique (la lettre «  $\alpha$  ») en se servant d'une peinture (la police de caractères).

Heureusement petit à petit une solution a émergé : elle a consisté à inventer un jeu de caractère qui puisse stocker tous les caractères présents sur la terre, y compris dans le passé, tout en laissant de la place pour les caractères des civilisations extra-terrestres qu'un jour, éventuellement, nous rencontrerions. Ce jeu de caractère s'appelle *Unicode*.

<sup>8</sup>D'où le fait que pendant longtemps les accents « sautaient » régulièrement lorsqu'on envoyait un email d'un ordinateur Apple vers un PC sous Windows ou *vice-versa*.

<sup>9</sup>Qui malheureusement est encore pratiquée, voir apprise, par des personnes peu au courant des évolutions informatiques.

Avec ce jeu de caractères, il est donc possible de mélanger allègrement de l'arabe, du vietnamien, de l'hébreu et du cyrillique dans un même fichier. Toutefois, histoire de compliquer les choses, plusieurs implémentations de ces jeux de caractères ont été inventées, chacune présentant des avantages et des inconvénient divers<sup>10</sup>. La plus courante de ces implémentations est UTF-8.

C'est celle que vous utilisez depuis que vous lisez ce livre, si du moins vous avez lu le chapitre 2.

En un mot : Unicode dans sa variante UTF-8 est aujourd'hui la meilleure méthode pour écrire des fichiers mêlant plusieurs familles de caractères<sup>11</sup>.

☛ Le lecteur narquois fera remarquer que le même problème se pose qu'avec la méthode de la police : à savoir que chaque ordinateur de travail devrait implanter Unicode et UTF-8 chez lui.

L'auteur fera remarquer qu'aujourd'hui tous les ordinateurs possèdent en natif ces possibilités, et qu'il est possible très facilement de l'installer sur des ordinateurs un peu anciens. En outre, avec Unicode on stocke du sens, et non pas de la forme, ce qui permet une plus grande souplesse.

## 4.2.2 Concrètement

Fort bien, fort bien, stockons en UTF-8. Mais comment écrit-on en UTF-8 ? Avec les claviers des ordinateurs vendus en Europe occidentale<sup>12</sup>, nous n'avons pas les caractères grecs à portée de main.

Il faut ici distinguer le support physique : le clavier avec ses touches bien concrètes, et le support logique : le fait que telle touche appuyée donne tel ou tel caractère.

Pour reprendre notre cas, il suffit de dire à notre ordinateur que la touche A correspond au caractère  $\alpha$ . Les ordinateurs récents proposent plusieurs pilotes de clavier en standard<sup>13</sup>. Toutefois ces pilotes de clavier sont généralement destinés aux langues contemporaines, et rarement adaptés aux langues anciennes – par exemple pour les accents sur le grec<sup>14</sup>. Heureusement on trouve aisément sur Internet des pilotes de clavier pour d'autres langues<sup>15</sup>.

---

<sup>10</sup>Par exemple sur le volume des fichiers et les temps de recherche.

<sup>11</sup>L'auteur pense même que, étant donné la baisse des coûts de stockages et de transferts, on ne devrait plus utiliser que ce jeu de caractères, ce qui aurait pour mérite de permettre bien plus facilement à tous les peuples de s'exprimer dans leur langue. Malheureusement son expérience personnelle lui prouve que cela n'est pas encore toujours le cas.

<sup>12</sup>Le lecteur militant voudra bien pardonner cet ethnocentrisme.

<sup>13</sup>Sous Macintosh, cela se règle dans les Préférences Systèmes, panneau « International », sous Windows cela se règle dans les panneaux de configuration, panneau « options régionales et linguistiques » ; sous Linux, on trouve le réglage dans les Paramètres Systèmes, panneau « Pays et langue », onglet « Agencements ».

<sup>14</sup>À l'exception notable de GNU/Linux, qui propose des dispositions de clavier pour le grec polytonique et l'hébreu biblique.

<sup>15</sup>Pour ce qui concerne le grec ancien, le syriaque, l'hébreu ancien, on pourra utiliser les claviers

### 4.2.3 Et les changements de sens d'écriture ?

Certaines langues s'écrivent de droite à gauche, d'autres de gauche à droite. On souhaiterait que les alignements de paragraphes, les positions des titres et d'autres éléments correspondent au sens de la langue. Comment signaler cela à  $\text{\LaTeX}$  ? L'indication des changements de langue (p. 34, 4.1.3) suffit.

 Pour ce qui concerne l'écriture en boustrophédon, on utilisera le package *bidi*. Celui-ci possède des commandes qui permettent d'indiquer des changements de sens.

---

proposés par Michael Langlois : Michael LANGLOIS, *Archives pour la catégorie « Informatique »*, URL : <http://michaellanglois.org/category/it/lang/fr> ; ou encore, pour le grec, les pilotes de l'École Normale Supérieure Département des sciences de l'antiquité ÉCOLE NORMALE SUPÉRIEURE, *Pilotes de clavier Unicode*, URL : <http://www.antiquite.ens.fr/ressources/outils-logiciels/article/pilotes-de-clavier-unicode>.

## Chapitre 5

# Mettre en sens son document (1) : premiers pas

*Nous allons maintenant voir comment mettre en sens notre document, c'est-à-dire comment poser des balises, des repères, pour marquer le « relief » sémantique du texte.*

### 5.1 Mettre en forme n'est pas mettre en sens

Lorsque nous lisons un livre, tous les éléments ne sont pas présentés de la même manière : certains sont en gras, d'autres en italique, en souligné, en couleur, etc.

Tout ceci constitue la *mise en forme* du texte. Si notre livre est bien conçu, ces changements de forme renvoient à des changements de signification : l'italique peut indiquer un titre d'ouvrage ou bien une citation ou une simple insistance, le gras peut indiquer une notion ou une définition ou toute autre signification.

On le voit, la *mise en forme* diffère de la *mise en sens*. Cette dernière est idéalement faite par l'auteur du travail, tandis que l'éditeur s'occupe normalement de la mise en forme et de la mise en page.

C'est d'ailleurs ce qui se passait auparavant quand les auteurs proposaient encore des textes manuscrits à leurs éditeurs : ils indiquaient les éléments à mettre en sens par des signes, mise en sens que l'éditeur transformait en mise en forme<sup>1</sup>.

Dans  $\text{\LaTeX}$ , le principe est le même : il existe des commandes de mise en sens qui sont ensuite transformées en commandes de mise en forme. Mieux : on peut définir ses propres commandes de mise en sens. L'intérêt est évident : pouvoir

---

<sup>1</sup>Dans un traitement de texte de type WYSIWYG, cette distinction se fait normalement à l'aide des styles. Bien souvent malheureusement les utilisateurs ne savent pas s'en servir.

changer rapidement de mise en forme pour un ensemble de données mises en sens.

Un exemple sera plus parlant. Supposons que nous écrivions un livre d'introduction à l'histoire du christianisme antique. Ce livre cite divers auteurs. Nous souhaitons mettre en valeur ces auteurs, et pour ce faire décidons de les mettre en petites capitales.

Donc, à chaque fois que nous citons un auteur, nous indiquons que nous souhaitons avoir son nom en petite capitale. Vient le moment où nous imprimons notre livre, et nous nous rendons compte que le choix des petites capitales n'est pas le plus pertinent, mais qu'il vaudrait mieux mettre du gras. Il ne nous reste alors plus qu'à repérer toutes les petites capitales dans notre texte, à vérifier qu'il s'agit bien de petites capitales indiquant un nom d'auteur et à les remplacer par du gras – travail fastidieux !

En revanche, si au lieu de signaler à chaque occurrence qu'il faut des petites capitales, nous signalons simplement qu'il s'agit d'un nom d'auteur – par exemple en écrivant : `\auteur{Tertullien}` – nous n'aurons qu'une seule ligne à changer pour indiquer que nous souhaitons avoir les noms d'auteur en gras. Mieux : nous pourrions créer très simplement un index des auteurs (☞ p. 144, 18.1.1).

LaTeX propose quelques commandes simples de mise en sens : par exemple celles que nous avons vues plus haut pour indiquer les niveaux de titres (☞ p. 29, 3.1).

Nous allons ici présenter quelques autres commandes et environnements de mise en sens. Dans le chapitre suivant, nous en indiquerons des spécifiques aux citations (☞ p. 45, 6). Dans un troisième chapitre, nous expliquerons comment créer ses propres commandes (☞ p. 53, 7), et nous présenterons alors la manière de mettre en forme.

## 5.2 Commandes de mise en sens

### 5.2.1 Mise en valeur d'un texte

On peut ponctuellement vouloir mettre en valeur un morceau de son écrit. Pour ce faire il existe la commande `\emph{⟨texte en emphase⟩}`. Exemple :

```

1 | On peut se demander si des textes apocryphes
2 | ont été non seulement \emph{utilisés} mais aussi \emph{lus}
3 | dans la liturgie africaine.
```

Concrètement cela se traduit par un italique.

On peut se demander si des textes apocryphes ont été non seulement *utilisés* mais aussi *lus* dans la liturgie africaine.

Toutefois, à la différence d'une commande qui indiquerait directement de mettre le texte en italique, cette commande pourrait, si on voulait, donner un résultat différent, par exemple mettre en couleur.

Une autre propriété intéressante est la gestion des imbrications : par défaut une commande `\emph` à l'intérieur d'une autre commande `\emph` produit un texte en caractères droits.

## 5.2.2 Le paratexte : notes de bas de page et de marge

TeX propose deux commandes pour indiquer des paratextes<sup>2</sup> : pour des notes de bas de page et des notes de marge (la position de ces dernières changeant, quand on est en recto-verso, selon que la page est paire ou impaire). Ces commandes sont, respectivement, `\footnote` et `\marginpar`.

```
1 | Lorem\footnote{Une note de bas de page.} ipsum dolor amat.
2 | Aliquam sagittis\marginpar{Annotation marginale} magna.
```

⚡ On serait tenté d'utiliser cette commande pour citer en note de bas de page une référence bibliographique. Il existe en fait une commande spécifique, que nous étudierons en temps voulu (☞ p. 95, 11.3).

⚡ Certaines mauvaises langues diront qu'il s'agit ici d'une mise en forme et non pas d'une mise en sens. Ils ont partiellement raison, dans la mesure où parfois distinguer la mise en forme de la mise en sens n'est pas évident.

Les personnes vraiment perfectionnistes pourront définir leurs propres commandes pour différencier les différents sens d'une note de marge ou de bas de page.

👉 Certains préfèrent mettre des notes de fin de texte. Bien que nous n'approuvons guère ce choix, nous signalons qu'il est possible d'en produire à l'aide du package *endnotes*.

## 5.2.3 Listes

TeX propose trois types de listes : les listes numérotées, les listes non-numérotées et les listes de description.

### Les listes numérotées

Une liste numérotée est un environnement `enumerate`. Chaque élément de la liste est marqué par la commande `\item`.

---

<sup>2</sup>La question des apparats critiques mise à part, question que nous traiterons plus loin (☞ p. 165, 20).

```

1  \begin{enumerate}
2      \item Premier élément
3      \item Deuxième élément
4      \item Troisième élément
5  \end{enumerate}

```

1. Premier élément
2. Deuxième élément
3. Troisième élément

 Il existe un package *etaremune* proposant l'environnement **etaremune** pour obtenir une liste numérotée à l'envers, avec le plus grand numéro en début de liste.

### Les listes non-numérotées

Une liste non-numérotée est un environnement **itemize**. Chaque élément de la liste est marqué par la commande `\item`.

```

1  \begin{itemize}
2      \item Un élément
3      \item Un autre
4      \item Encore un autre
5  \end{itemize}

```

- Un élément
- Un autre
- Encore un autre

### Les listes de descriptions

Une liste de descriptions fait correspondre une à une des valeurs. Une telle liste peut être utile pour des lexiques, des glossaires, des chronologies, etc. Pour chaque couple, la première valeur est passée comme argument à la commande `\item`. Les listes de définitions sont des environnements **description**.

```

1  \begin{description}
2      \item[325] Concile de Nicée.
3      \item[381] Concile de Constantinople.
4      \item[431] Concile d'Éphèse.
5  \end{description}

```

325	Concile de Nicée.
381	Concile de Constantinople.
431	Concile d'Éphèse.

### 5.2.4 Imbrication des listes

Il est possible d'imbriquer des listes, quels que soient leurs types. On ne peut, par défaut, avoir plus de quatre niveaux d'imbrication.

```

1 \begin{itemize}
2   \item Un élément de premier niveau
3   \begin{enumerate}
4     \item Premier sous élément
5     \item Second sous élément
6   \end{enumerate}
7   \item Un autre élément de premier niveau
8 \end{itemize}

```

- Un élément de premier niveau
  1. Premier sous élément
  2. Second sous élément
- Un autre élément de premier niveau

 On a parfois besoin de personnaliser l'aspect des listes, ou bien encore d'arrêter une numérotation de liste pour la reprendre plus loin. Bertrand Masson a écrit un excellent tutoriel sur la manière d'utiliser le package *enumitem* pour arriver à ces fins<sup>3</sup>.

<sup>3</sup>Bertrand MASSON, *Les fiches de Bébert : dompter ses puces*, URL : <http://bertrandmasson.free.fr/index.php?article5/dompter-ses-puces>.



# Chapitre 6

## Mettre en sens (2) : l'art de citer en LaTeX

*Nous évoquerons dans ce chapitre les citations explicites et textuelles, c'est-à-dire celles où l'auteur du travail ne se contente pas de renvoyer à une source ou à une étude, mais cite des extraits de cette source.*

*Une citation peut se faire de deux manières : dans le corps du propos, elle est alors normalement entourée de guillemets, ou bien dans un paragraphe spécifique. Elle est alors généralement présentée avec des marqueurs typographiques particuliers : changement de la taille de police, de la marge etc.*

*Nous avons vu plus haut qu'il fallait séparer sens et forme (☞ p. 39, 5.1). Nous allons donc présenter ici les commandes servant à marquer des citations.*

↪ Toute citation se doit d'être accompagnée d'une référence, généralement en note de bas de page. Toutefois toute référence n'accompagne pas nécessairement une citation textuelle. C'est pourquoi nous renvoyons pour la gestion des références bibliographiques à la partie qui lui est consacrée.

### 6.1 Citation dans le corps du texte

Les citations dans le corps d'un texte sont normalement entourées de guillemets français : «». Lorsque qu'on cite un texte qui cite un texte, la citation dans la citation s'entoure de guillemets courbes "".

Comme le dit très justement xxx : « Lorsque yyy déclare "zzz" il ne déclare rien du tout ».

Les claviers disponibles sur nos ordinateurs ne disposent généralement en accès direct que des guillemets anglais<sup>1</sup>("). La plupart des logiciels WYSIWYG

---

<sup>1</sup>Une exception notable est la disposition de clavier Bépo.

convertissent automatiquement ces guillemets en guillemets français. Rares sont les éditeurs de texte qui le proposent<sup>2</sup>.

En outre, en vertu du principe de séparation du sens et de la forme, évoqué plus haut (☞ p. 39, 5.1), il est plus pertinent d'utiliser une commande spécifique pour indiquer une citation dans le corps du texte.

Nous allons donc utiliser le package *csquotes* qui propose des commandes pour les citations.

Le package contient une première commande utile : `\enquote{<citation>}`, qui sert pour les citations dans le corps du texte.

```
1 | Comme le dit très justement xxx :
2 | \enquote{Lorsque yyy déclare \enquote{zzz}}
3 | il ne déclare rien du tout}.
```

Comme le dit très justement xxx : « Lorsque yyy déclare  
“zzz” il ne déclare rien du tout ».

Nous constatons que *csquotes* s'occupe automatiquement de choisir les bons guillemets. Par défaut on ne peut imbriquer que deux niveaux de citation. Toutefois une option du package permet d'avoir plus de niveaux de citation. Par exemple pour en avoir trois :

```
1 | \usepackage[maxlevel=3]{csquotes}
```

Le package propose d'autres options et commandes : consultez le manuel<sup>3</sup>.

## 6.2 Citation dans un bloc séparé

LaTeX propose en standard trois environnements pour citer dans un bloc séparé.

### 6.2.1 L'environnement quote

Il est prévu pour des courtes citations d'un paragraphe<sup>4</sup>.

```
1 | \begin{quote}
2 | Le corps de Pierre gît à Rome, disent les hommes,
```

<sup>2</sup>C'est d'ailleurs à nos yeux une des raisons qui fait qu'un site internet d'un quotidien national dit « de référence » n'utilise pas de guillemets français sur sa page d'accueil (à la date du 10 avril 2011), les rédacteurs ne prenant pas le temps de taper les combinaisons complexes de touches nécessaires à la frappe de guillemets français.

<sup>3</sup>Philipp LEHMAN, *The csquotes Package : Context Sensitive Quotation Facilities*.

<sup>4</sup>AUGUSTIN, *Sermon 296*, § 5-6, *Patrologia Latina* 38, col. 6.

```

3 | le corps de Paul gît à Rome, le corps de Laurent aussi,
4 | les corps d'autres martyrs y gisent,
5 | mais Rome est misérable,
6 | elle est dévastée, affligée, saccagée, incendiée.
7 | \end{quote}

```

Ce code produit le résultat suivant :

Le corps de Pierre gît à Rome, disent les hommes, le corps de Paul gît à Rome, le corps de Laurent aussi, les corps d'autres martyrs y gisent, mais Rome est misérable, elle est dévastée, affligée, saccagée, incendiée.

## 6.2.2 L'environnement quotation

Il est prévu pour des citations plus longues<sup>5</sup>.

```

1 | \begin{quotation}
2 | Que rien exceptées les écritures canoniques ne soit lu en église
3 | sous le nom d'écritures divines.
4 |
5 | Les écritures canoniques sont : Genèse, Exode, Lévitique,
6 | Nombres, Deutéronome,
7 | Josué de Noun, Juges, Ruth, 4~livres des règnes,
8 | 2~livres des paralipomènes,
9 | Job, psautier, 5~livres de Salomon,
10 | 12 livres des prophètes mineurs,
11 | de même Isaïe, Jérémie, Ézechiel, Daniel,
12 | Tobie, Judith, Esther,
13 | 2~livres d'Esdras, 2~livres des Maccabées.
14 |
15 | Du nouveau testament sont :
16 | 4~évangiles, un livre des actes des apôtres,
17 | 14~lettres de l'apôtre Paul, 2~de Pierre,
18 | 3~de Jean, 1~de Jude, 1~de Jacques,
19 | l'apocalypse de Jean.
20 |
21 | Que l'Église d'outre-mer soit consultée
22 | pour la confirmation de ce canon.
23 |
24 | De plus, qu'il soit permis de lire les passions des martyrs,

```

<sup>5</sup>*Breviarum Hipponenses*, 36, dans : *Concilia Africae*, éd. Charles MUNIER, Corpus Christianorum Series Latina 149, p. 43.

25 | lorsqu'on célèbre leurs anniversaires.  
 26 | `\end{quotation}`

Que rien exceptées les écritures canoniques ne soit lu en église sous le nom d'écritures divines.

Les écritures canoniques sont : Genèse, Exode, Lévitique, Nombres, Deutéronome, Josué de Noun, Juges, Ruth, 4 livres des règnes, 2 livres des paralipomènes, Job, psautier, 5 livres de Salomon, 12 livres des prophètes mineurs, de même Isaïe, Jérémie, Ézechiel, Daniel, Tobie, Judith, Esther, 2 livres d'Esdras, 2 livres des Maccabées.

Du nouveau testament sont : 4 évangiles, un livre des actes des apôtres, 14 lettres de l'apôtre Paul, 2 de Pierre, 3 de Jean, 1 de Jude, 1 de Jacques, l'apocalypse de Jean.

Que l'Église d'outre-mer soit consultée pour la confirmation de ce canon.

De plus, qu'il soit permis de lire les passions des martyrs, lorsqu'on célèbre leurs anniversaires.

⚡ Le fond gris que vous constatez ici est propre au livre que vous avez sous les yeux. L'environnement `quotation` standard n'a pas de fond.

### 6.2.3 L'environnement `verse` et le package `verse`

L'environnement `verse` permet de citer de façon rapide et simple des poèmes, en gérant notamment le rejet en cas de vers trop long. Chaque vers, à l'exception du dernier, doit se terminer par `\\`. Si le vers est composé de plusieurs strophes, il suffit de sauter une ligne entre chaque strophe. Il ne faut pas mettre `\\` à la fin du dernier vers de chaque strophe.

Cet environnement est très limité : il ne permet pas de numérotter ni d'indenter les vers, ni encore de rajouter un titre. Si l'on est amené à citer fréquemment des vers, il vaut mieux appeler dans le préambule le package `verse`<sup>6</sup>. La citation du poème se fait de la même manière.

Avec ce package il faut indiquer `\poemlines{<n>}` pour numérotter les vers cités : `<n>` définit la fréquence à laquelle les vers sont numérotés.

Citons pour exemple un poème entier, dont les vers sont numérotés, de façon assez traditionnelle, une fois sur cinq. En frappant ceci<sup>7</sup> :

```
1 | \begin{verse}
2 | \poemlines{5}
3 | Demain, dès l'aube, à l'heure où blanchit la campagne,\\
```

<sup>6</sup>Peter WILSON, Herries PRESS et Will ROBERTSON, *Typesetting simple verse with LaTeX*.

<sup>7</sup>Victor HUGO, *Demain dès l'aube ...*

```

4 | Je partirai. Vois-tu, je sais que tu m'attends.\\
5 | J'irai par la forêt, j'irai par la montagne.\\
6 | Je ne puis demeurer loin de toi plus longtemps.
7 |
8 | Je marcherai les yeux fixés sur mes pensées,\\
9 | Sans rien voir au dehors, sans entendre aucun bruit,\\
10 | Seul, inconnu, le dos courbé, les mains croisées,\\
11 | Triste, et le jour pour moi sera comme la nuit.
12 |
13 | Je ne regarderai ni l'or du soir qui tombe,\\
14 | Ni les voiles au loin descendant vers Harfleur,\\
15 | Et, quand j'arriverai, je mettrai sur ta tombe\\
16 | Un bouquet de houx vert et de bruyère en fleur.
17 |
18 | \end{verse}

```

on obtient cela :

Demain, dès l'aube, à l'heure où blanchit la campagne,  
 Je partirai. Vois-tu, je sais que tu m'attends.  
 J'irai par la forêt, j'irai par la montagne.  
 Je ne puis demeurer loin de toi plus longtemps.

Je marcherai les yeux fixés sur mes pensées,  
 Sans rien voir au dehors, sans entendre aucun bruit,  
 Seul, inconnu, le dos courbé, les mains croisées,  
 Triste, et le jour pour moi sera comme la nuit.

Je ne regarderai ni l'or du soir qui tombe,  
 Ni les voiles au loin descendant vers Harfleur,  
 Et, quand j'arriverai, je mettrai sur ta tombe  
 Un bouquet de houx vert et de bruyère en fleur.

5

Si l'on ne cite pas un poème en entier, mais simplement un passage, il faut indiquer à quel vers commence le passage cité, afin que la numérotation des vers que l'on cite soit la bonne.

On utilise la commande `\setverselinenums{⟨premier vers⟩}{⟨premier vers numéroté⟩}` : `⟨premier vers⟩` indique le numéro du premier vers que l'on cite, `⟨premier vers numéroté⟩` où doit commencer la numérotation.

Ainsi, `\setverselinenums{⟨12⟩}{⟨15⟩}` indique que l'extrait cité commence par le douzième vers du poème, et que le premier vers numéroté sera le quinzième. Avec `\setverselinenums{⟨12⟩}{⟨12⟩}`, le premier vers cité sera le premier numéroté.

Ne citant que la troisième strophe, par exemple, il faut frapper :

```

1  \begin{verse}
2  \poemlines{5}
3  \setverselinenums{9}{10}
4
5
6  Je ne regarderai ni l'or du soir qui tombe,\!\!
7  Ni les voiles au loin descendant vers Harfleur,\!\!
8  Et, quand j'arriverai, je mettrai sur ta tombe\!\!
9  Un bouquet de houx vert et de bruyère en fleur.
10
11 \end{verse}

```

pour obtenir, correctement numéroté :

Je ne regarderai ni l'or du soir qui tombe,  
Ni les voiles au loin descendant vers Harfleur,  
Et, quand j'arriverai, je mettrai sur ta tombe  
Un bouquet de houx vert et de bruyère en fleur.

10

 Dans la typographie française, il est d'usage de mettre un crochet droit [ au début d'un rejet. Ni l'environnement ni le package *verse* ne suivent cette règle. Pour obtenir le crochet droit, il faut charger le package *gmverse* en lui passant l'option **squarebr** :

```

1  | \usepackage[squarebr]{gmverse}

```

Puis insérer, une fois commencé l'environnement **verse**, la commande `\versehangrightsquare`.

Le package *verse* permet aussi d'indenter de façon très souple une strophe. On indique avec la commande `\indentpattern{⟨ $n_1 n_2 n_x$ ⟩}` l'indentation de chaque vers contenu dans la strophe encadrée à l'intérieur d'un environnement **patverse** :  $\langle n_1 \rangle$  correspond au premier vers,  $\langle n_2 \rangle$  au deuxième et ainsi de suite. Le premier vers n'est jamais indenté, mais il suffit de le faire précéder de `\vin`. Avec le code suivant :

```

1  \begin{verse}
2  \poemlines{1}
3  \indentpattern{010110}
4  \begin{patverse}
5
6  \vin There was a young lady of Ryde \!\!
7  Who ate some apples and died. \!\!
8  The apples fermented \!\!
9  Inside the lamented \!\!

```

```

10 | And made cider inside her inside. \\
11 | \end{patverse}
12 | \end{verse}

```

on obtient donc<sup>8</sup> :

There was a young lady of Ryde	1
Who ate some apples and died.	2
The apples fermented	3
Inside the lamented	4
And made cider inside her inside.	5

Si l'on a besoin de répéter une indentation tout le long d'un poème, on remplace `patverse` par `patverse*`. Donc pour citer un long extrait en hexamètres dactyliques, plutôt que d'indiquer l'indentation pour chaque vers, il suffit d'indiquer `\indentpattern{01}` et d'insérer le poème, en entier, dans l'environnement `patverse*` pour avoir un vers sur deux indenté.

Le package `verse` offre bien d'autres possibilités qui dépassent la simple citation de poésie dans le cours d'un texte. À l'instar d'un autre package, `poemscol`, c'est un véritable outil pour éditer de la poésie<sup>9</sup>. Ni l'un ni l'autre ne permettent en revanche de faire des éditions bilingues : pour cela, il faut utiliser les packages `ledmac` et `ledpar` (☞ p. 165, 20).

## 6.3 Citations tronquées et modifiées

Le package `csquotes` propose deux commandes spécifiques pour signaler une citation tronquée ou modifiée.

### 6.3.1 Citation tronquée

La commande `\textelp<texte>` signale un texte tronqué. L'argument `<texte>` est inséré après la troncature, entre crochets (par défaut) ; pour ne rien insérer, le laisser vide.

```

1 | \begin{quotation}
2 | Que rien exceptées les écritures canoniques ne soit lu
3 | en église sous le nom d'écritures divines.
4 | \textelp{Suit la liste des écritures canoniques.}
5 |
6 | Que l'Église d'outre-mer soit consultée pour la confirmation

```

<sup>8</sup>Edward LEAR, *There was a Young Lady of Ryde*.

<sup>9</sup>WILSON, PRESS et ROBERTSON, *Typesetting simple verse with LaTeX, op. cit.* ; John BURT, *Typesetting Poetry Collections with poemscol*, Nous vous renvoyons ici aux manuels de ces packages.

```

7 | de ce canon.
8 |
9 | De plus, qu'il soit permis de lire les passions des martyrs,
10 | lorsqu'on célèbre leurs anniversaires.
11 | \end{quotation}

```

Que rien exceptées les écritures canoniques ne soit lu en église sous le nom d'écritures divines. [...] [Suit la liste des écritures canoniques.]

Que l'Église d'outre-mer soit consultée pour la confirmation de ce canon.

De plus, qu'il soit permis de lire les passions des martyrs, lorsqu'on célèbre leurs anniversaires.

 On peut décider de la manière dont la troncature et l'ajout sont signalés : consulter le manuel<sup>10</sup>.

### 6.3.2 Citation modifiée

Pour signaler une modification dans une citation, on utilise `\textins{texte modifié}`.

```

1 | \begin{quotation}
2 | Comme le disait très justement xxx :
3 | \enquote{Lorsque yyy \textins{a déclaré}
4 | \enquote{zzz} il \textins{n'a rien déclaré} du tout}.
5 | \end{quotation}

```

Comme le disait très justement xxx : « Lorsque yyy [a déclaré] “zzz” il [n'a rien déclaré] du tout ».

La commande `\textins*` est une variante, servant pour déclarer les changements mineurs nécessaires au nouveau contexte d'énonciation : mise en majuscules, changement de personne etc.

---

<sup>10</sup>Philipp LEHMAN, « Configuring Ellipses », dans : *The csquotes Package : Context Sensitive Quotation Facilities*.

# Chapitre 7

## Mettre en sens (3) : créer ses propres commandes

*Nous avons parlé longuement de l'intérêt de séparer mise en sens et mise en forme (☞ p. 39, 5.1). Nous avons indiqué que la meilleure manière pour ce faire était de créer des commandes de mise en sens, qui elles-même appelleraient des commandes de mise en forme.*

*Voyons maintenant comment créer ces commandes personnalisées.*

### 7.1 Création d'une commande personnalisée

Nous souhaitons créer une commande personnalisée servant à indiquer que nous parlons d'un auteur : `\auteur{⟨nom⟩}`.

Notre commande se décompose en plusieurs parties : son nom, ici `auteur`, et ses arguments, ici un seul : `⟨nom⟩`.

Une commande LaTeX peut prendre jusqu'à neuf arguments, ce qui est en général bien suffisant. Si nous notons  $\langle N \rangle$  le nombre d'arguments, la syntaxe de la déclaration d'une nouvelle commande est la suivante : `\newcommand{\⟨nom de la commande⟩}[⟨N⟩]{⟨code⟩}`.

↪ Les noms de commandes ne doivent contenir que des caractères latins non accentués.

Les noms sont sensibles à la casse : `\a` est différent de `\A`.

À l'intérieur de la partie `⟨code⟩`, on peut :

- mettre du texte ;
- utiliser des commandes de mise en forme ou de mise en sens ;

- appeler les arguments passés en utilisant la syntaxe #x, où x représente le rang de l'argument.

Prenons toujours notre cas d'une commande pour indiquer les noms. Pour le début, nous souhaitons simplement que les noms d'auteur soient suivis d'un astérisque (\*).

```
1 | \newcommand{\auteur}[1]{#1*}
2 | Il est bien connu que \auteur{Tertullien} n'était pas montaniste
3 | mais tertullianiste.
```

Ce qui nous donne :

Il est bien connu que Tertullien\* n'était pas montaniste  
mais tertullianiste.

Tertullien a été passé en premier argument. Il apparaît donc à la place du #1 de la déclaration de commande.

On pourrait souhaiter passer en second argument des informations complémentaires, comme les dates de vie.

```
1 | \newcommand{\auteur}[2]{#1* (#2)}
2 | Il est bien connu que \auteur{Tertullien}{150? - 220?}
3 | n'était pas montaniste mais tertullianiste.
```

Il est bien connu que Tertullien\* (150? - 220?) n'était pas  
montaniste mais tertullianiste.

 Le lecteur attentif se rappellera qu'il existe des commandes avec des arguments optionnels (☞ p. 23, 2.2.5). Il souhaitera sans doute en créer lui-même.

Si  $\text{\TeX}$  prévoit bien en standard un mécanisme pour permettre cela, celui-ci est peu souple.

Nous conseillons donc de se tourner vers les packages *xargs* et *ifthenelse*<sup>1</sup>.

Mais tout ceci dépasserait le cadre de cet ouvrage.

Fort bien, mais comment faire pour mettre en forme ? Il faut utiliser une commande de mise en forme à l'intérieur de la déclaration de commande de mise en sens. Par exemple, pour mettre en petites capitales la commande est `\textsc`.

```
1 | \newcommand{\auteur}[2]{\textsc{#1}* (#2)}
```

<sup>1</sup>Manuel PÉGOURIÉ-GONNARD, *L'extension xargs*; David CARLISLE, *The ifthen package*.

Il est bien connu que TERTULLIEN\* (150 ? - 220 ?) n'était pas montaniste mais tertullianiste.

Nous décrivons plus loin un ensemble de commandes de mise en forme (☞ p. 57, 7.2).

⚡ Si la possibilité de créer ses propres commandes est une grande force de  $\LaTeX$ , cela ne fait pas toujours la joie des éditeurs.

Il est donc recommandé de discuter au préalable avec son éditeur, et surtout de lui indiquer qu'un balisage sous forme de commandes doit, normalement, lui faciliter la tâche.

Nous recommandons par ailleurs de regrouper toutes ses déclarations de commandes dans un fichier unique, appelé via une commande `\input` (☞ p. 31, 3).

Il est également possible de créer son propre package, pour des commandes utilisées fréquemment dans plusieurs projets<sup>2</sup>.

👉 On peut ainsi très facilement créer une commande `\commentaire` servant pour indiquer des commentaires personnels au cours de la rédaction, afin de les avoir sous les yeux lors d'une relecture papier. Par exemple :

```
1 | \newcommand{\commentaire}[1]{\marginpar{#1}}
```

Lors de l'impression finale, il suffit de changer cette commande par :

```
1 | \newcommand{\commentaire}[1]{} 
```

### 7.1.1 Code d'une commande sur plusieurs lignes

Jusqu'ici nos commandes étaient relativement simples. Mais il peut arriver que leur code se complexifie. On souhaiterait alors l'avoir sur plusieurs lignes.

C'est tout à fait possible, mais il faut simplement mettre un signe pourcentage (%) à la fin de chaque ligne, sous peine d'avoir des espaces intempestives. Exemple sans % :

```
1 | \newcommand{\auteur}[2]{
2 |   \textsc{#1}* (#2)
3 | }
```

Il est bien connu que `_TERTULLIEN*` (150 ? - 220 ?) n'était pas montaniste mais tertullianiste.

<sup>2</sup>Ceci dépasserait le cadre de cet ouvrage : je renvoie à d'autres documents. Par exemples *LaTeX : Créer une extension ou une classe*, URL : [http://fr.wikibooks.org/wiki/LaTeX/Créer\\_une\\_extension\\_ou\\_une\\_classe](http://fr.wikibooks.org/wiki/LaTeX/Créer_une_extension_ou_une_classe).

On constate une espace en trop avant et une après la mention de l’auteur, que nous avons soulignées pour les besoins de l’explication. En revanche, si on met des % :

```
1 | \newcommand{\auteur}[2]{%
2 |   \textsc{#1}* (#2)%
3 | }
```

Il est bien connu que TERTULLIEN\* (150 ? - 220 ?) n’était pas montaniste mais tertullianiste.

Il n’y a plus d’espaces intempestives.

### 7.1.2 Étoiler `\newcommand`

La création des commandes sous  $\LaTeX$  est une fonctionnalité très puissante, mais potentiellement dangereuse. Prenons la commande suivante :

```
1 | \newcommand{\auteur}[1]{\textsc{#1}*}
```

Imaginons que nous l’appelions en oubliant de fermer notre accolade :

```
1 | Il est bien connu que \auteur{Tertullien n’était pas montaniste
2 |   mais tertullianiste.
```

Que va-t-il se passer ?  $\LaTeX$  comprend que l’ensemble de la fin du texte est un argument de la commande, ce qui empêche le compilateur de bien fonctionner et nous n’obtiendrons rien, ou alors un résultat incorrect.

Pour limiter ces soucis,  $\LaTeX$  propose de définir des commandes courtes, dont les arguments ne dépassent pas la longueur d’un paragraphe. En cas d’oubli de fermeture de parenthèses,  $\LaTeX$  arrête automatiquement l’argument à la fin du paragraphe.

Pour définir ces commandes courtes, il suffit d’utiliser `\newcommand*`.

```
1 | \newcommand*{\auteur}[1]{\textsc{#1}*}
```

### 7.1.3 Espace après les commandes sans argument

Il est possible de définir des commandes sans aucun argument. Prenons une commande personnalisée `\cf`, qui permet d’obtenir rapidement la forme « *cf.* »

```
1 | \newcommand{\cf}[0]{\emph{cf.}}
```

Comme notre commande ne prend pas d’argument, il est nécessaire, lorsque nous l’appelons, de la séparer du texte qui suit par une espace, afin que  $\LaTeX$  sache où s’arrête la commande :

1 | Ceci est très intéressant `\cf` cela.

Mais l'espace disparaît lors de la compilation :

Ceci est très intéressant *cf.cela*.

La solution consiste à utiliser dans la définition de commande la commande `\xspace` du package éponyme. Elle permet de gérer automatiquement les espaces qui suivent l'appel à une commande, en tenant compte de la typographie.

1 | `\newcommand{\cf}[0]{\emph{cf.}\xspace}`

Ceci est très intéressant *cf.* cela.

## 7.2 Commandes de mise en forme

Nous listons ici les principales commandes de mise en forme. Pour des besoins avancés, on trouvera aisément de la documentation sur internet. En ce qui concerne les questions d'espacement, un chapitre spécifique leur est dédié (☞ p. 209, 24).

### 7.2.1 Taille des caractères

Il est possible de définir de manière absolue la taille des caractères en indiquant le nombre de points. Toutefois la meilleure méthode est de le définir d'après la taille de base, et d'utiliser les commandes ci-dessous. La taille obtenue dépend de la taille de base définie lors de l'appel au préambule (☞ p. 22, 2.2.4).

Commande	Effet
<code>\tiny</code>	abcdefghijklmnopqrst
<code>\scriptsize</code>	abcdefghijklmnopqrst
<code>\footnotesize</code>	abcdefghijklmnopqrst
<code>\small</code>	abcdefghijklmnopqrst
<code>\normalsize</code>	abcdefghijklmnopqrst
<code>\large</code>	abcdefghijklmnopqrst
<code>\Large</code>	abcdefghijklmnopqrst
<code>\LARGE</code>	abcdefghijklmnopqrst
<code>\huge</code>	abcdefghijklmnopqrst
<code>\Huge</code>	abcdefghijklmnopqrst

Ces commandes ont toutefois un comportement particulier : ce sont des commandes dites « à bascule » (Uwe p. 199, 23.2.1). À la différence de la plupart des commandes, elles ne prennent pas d'argument. Elles doivent être utilisées entre accolades, le texte dont la taille change étant inséré après la commande, dans les accolades.

Ainsi :

```
1 | {\large Texte plus gros}
```

Et non pas :

```
1 | \large{Texte plus gros}
```

## 7.2.2 Style de caractères

Voici une liste non exhaustive de commandes utiles pour personnaliser les styles de caractère.

Commande	Effet
<code>\textit</code>	<i>Italique</i>
<code>\emph</code>	<i>Texte en emphase</i>
<code>\textbf</code>	<b>Gras</b>
<code>\textsc</code>	PETITES CAPITALES
<code>\underline</code>	Souligné (à éviter)
<code>\textsuperscript</code>	Exposant
<code>\textsubscript</code>	Indice (nécessite le package <i>subscript</i> )

## 7.2.3 Couleurs

Les couleurs ne sont pas gérées nativement dans L<sup>A</sup>T<sub>E</sub>X. Il faut passer par le package *color* ou le package *xcolor* qui proposent tous les deux une commande `\textcolor{<nom de la couleur>}{<texte en couleur>}`.

La différence tient principalement au nombre de couleurs disponibles par défaut et à la facilité d'en définir des nouvelles, mais aussi à la possibilité d'appliquer des couleurs à d'autres éléments que le texte. Nous ne présentons ici que le package *xcolor*, de manière succincte. Nous renvoyons à la documentation pour des usages avancés<sup>3</sup>.

### Couleurs standards

Le package *xcolor* propose les couleurs de bases suivantes :

<sup>3</sup>Uwe KERN, *Extending L<sup>A</sup>T<sub>E</sub>X's color facilities : the xcolor package*.

Nom de la couleur	Couleur
black	
blue	
brown	
cyan	
darkgray	
gray	
green	
lightgray	
lime	
magenta	
olive	
orange	
pink	
purple	
red	
teal	
violet	
white	
yellow	

### Couleurs supplémentaires

On peut passer les options `dvipsnames`, `svgnames` ou `x11names` lors de l'appel au package `xcolor`, chacune fournissant un jeu de couleurs. Ne pouvant lister toutes ces couleurs dans cet ouvrage, nous renvoyons à la documentation de `xcolor`<sup>4</sup>.

### Définir ses propres couleurs

Il est possible de définir ses propres couleurs. Il faut utiliser `\definecolor`, selon la syntaxe suivante : `\definecolor{<nom>}{<methode>}{<definition>}`.

Les couleurs peuvent être définies de plusieurs manières<sup>5</sup>. Les deux principales sont<sup>6</sup> :

<sup>4</sup>Uwe KERN, *Colors by Name : the xcolor package*.

<sup>5</sup>On consultera n'importe quel livre scientifique sur les couleurs pour plus de détails. On peut également consulter la documentation du package `xcolor`.

<sup>6</sup>Il existe aussi la méthode de définition par la longueur d'onde pour les couleurs de l'arc en ciel, par pourcentage de gris pour les différents niveaux de gris, ainsi que par teinte, saturation et luminosité, mais ceci nécessiterait un cours de physique lumineuse. Pour les personnes intéressées par les détails, consulter Uwe KERN, *Color models : the xcolor package*.

- La méthode additive en additionnant du rouge, du vert et du bleu (comme sur un écran d'ordinateur).
- La méthode soustractive en superposant du cyan, du magenta, du jaune et du noir (comme sur une imprimante couleur).

Dans les deux cas, il faut donner une valeur comprise entre 0 (inclus) et 1 (inclus) à chacun des composants<sup>7</sup>.

Prenons par exemple la définition de la couleur **rouge bourgogne**<sup>8</sup>. On peut la définir de la manière suivante en méthode additive :

```
1 | \definecolor{rougebourgogne}{rgb}{0.41,0.05,0.05}
```

Ou bien en méthode soustractive<sup>9</sup> de la manière suivante :

```
1 | \definecolor{rougebourgogne}{cmyk}{0,0.88,0.88,0.58}
```

Enfin signalons la méthode HTML qui offre l'avantage d'avoir de nombreux sites internet indiquant les codes couleur HTML<sup>10</sup>.

La même couleur se définit ainsi :

```
1 | \definecolor{rougebourgogne}{HTML}{6B0D0D}
```

---

<sup>7</sup>On pourra trouver une série de code couleur sur le site *Code Couleur*, URL : <http://www.code-couleur.com/>.

<sup>8</sup>Avec Anne Sylvestre les auteurs proclament fièrement « Que Bordeaux me pardonne, j'appartiens au Bourgogne » (Anne SYLVESTRE, *La romanée conti*).

<sup>9</sup>Idéalement le choix de la méthode devrait relever du support de destination. Mais le package *xcolors* propose des systèmes de conversion d'une méthode à l'autre.

<sup>10</sup>En réalité la méthode HTML est une méthode additive avec des valeurs comprises entre 0 et 255, notées en hexadécimal.

# Chapitre 8

## Insérer des éléments non textuels

*Dans ce chapitre nous allons examiner comment insérer des éléments qui ne font pas partie du flux du texte : images, graphismes, tableaux de données.*

### 8.1 Insérer des images

Les images insérables avec X<sub>Y</sub>TEX sont au format JPEG (extension **.jpg**), PNG (extension **.png**) ou PDF (extension **.pdf**). Il est nécessaire d'utiliser le package *graphicx*.

L'insertion d'une image se fait ainsi : `\includegraphics[<options>]{<chemin de l'image>}`.

Le chemin de l'image s'indique de la même manière que le chemin des fichiers inclus (☞ p. 31, 3.2.1). Les principales options, séparables par des virgules, sont les suivantes<sup>1</sup> :

Paramètres	Signification	Exemple
angle	Angle exprimé en degrés. Un nombre positif indique une rotation anti-horaire, un nombre négatif une rotation horaire.	angle=90
height	Hauteur	height=10cm
scale	Redimension proportionnelle	scale=0.5
width	Largeur	width=10cm

---

<sup>1</sup>Pour les autres, on consultera David CARLISLE, « Including Graphics Files », dans : *Packages in the 'graphics' bundle*.

✎ Le centimètre n'est pas la seule unité de mesure disponible en  $\LaTeX$ . Nous en parlons en annexe de ce livre (☞ p. 237, E).

✎ Le package *graphicx* propose de nombreux outils pour manipuler graphiquement du contenu, textuel ou non. Consultez le manuel<sup>2</sup>.

⚡ Évidemment, il est plus correct d'insérer une légende sous une image. Nous expliquons comment faire plus loin (☞ p. 70, 8.4).

## 8.2 Insérer des graphismes et des schémas

On peut vouloir insérer divers types de graphismes ou de schémas : graphiques statistiques, arbres généalogiques, *stemma codicum* etc. Deux solutions s'offrent à nous :

1. Utiliser un logiciel externe qui exporterait le graphisme dans une image, qu'on insérerait comme n'importe quelle image.
2. Utiliser les possibilités de  $\LaTeX$  et du package *TikZ*.

La première solution paraît de prime abord plus simple, puisqu'elle ne nécessite pas d'apprendre de nouveaux éléments en  $\LaTeX$ . Toutefois il faut avec elle gérer plus de fichiers<sup>3</sup>. En outre, les images exportées par ces logiciels étant bien souvent de type bitmap et non pas vectoriel<sup>4</sup>, on perd la possibilité de sélectionner dans le PDF généré par  $\LaTeX$  les textes, et on risque d'avoir des problèmes de redimensionnement.

Nous allons donc ici vous présenter une des possibilités du package *TikZ* : la représentation d'une famille de manuscrits.

Nous avons choisi cet exemple car il est assez simple, tout en étant fréquent en sciences humaines. Mais *TikZ* permet de faire bien plus, comme produire des diagrammes statistiques : nous renvoyons à d'autres ouvrages pour en savoir plus<sup>5</sup>. Ce package possède une syntaxe spécifique.

⚡ Bien que le package s'appelle *TikZ*, il faut dans le préambule l'appeler *sans majuscules* : `\usepackage{tikz}`.

<sup>2</sup>David CARLISLE, *Packages in the 'graphics' bundle*.

<sup>3</sup>Prenons un graphisme généré par un tableur : il faudra conserver la feuille du tableur plus l'image du graphisme généré par celui-ci.

<sup>4</sup>Il existe deux manières en informatique de coder une image : en codant point par point (image bitmap) ou bien en codant simplement les formes, par exemple des cercles ou des segments (image vectorielle). La première méthode est adaptée à des images complexes, comme les photographies, la seconde méthode est conseillée pour les images « simples » telles que les graphismes. La méthode vectorielle possède l'avantage d'offrir une image plus facilement redimensionnable sans perte de qualité et de permettre aisément la sélection de texte.

<sup>5</sup>Outre le manuel Till TANTAU, *Tikz & PGF : Manual for Version 2.0*, on pourra lire Gérard TISSEAU et Jacques DUMA, *TikZ pour l' impatient*, URL : <http://math.et.info.free.fr/TikZ/> ; on pourra également consulter le site *TeXample.net : TikZ and PGF*, URL : <http://www.texample.net/tikz/>.

Nous disposons de six familles de manuscrits d'un même texte. La famille A a engendré les familles B, C, D. Les familles E et F pour leur part sont issues de la famille D.

Le code est le suivant :

```

1  \begin{tikzpicture}
2      \node {A}
3          child { node {B}}
4          child { node {C}}
5          child { node {D}}
6              child {
7                  node{E}
8                  }
9              child {
10                 node{F}
11                 }
12             }
13 ;
14 \end{tikzpicture}

```

Ce qui donne le résultat de la figure 8.1. Analysons le code :

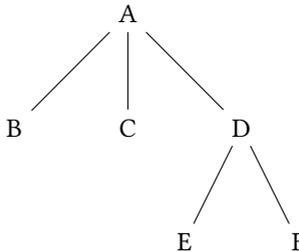


FIG. 8.1 : Exemple de *stemma*

**Ligne 1** L'environnement `tikzpicture` est l'environnement utilisé pour insérer une figure TikZ. Il peut recevoir de nombreux arguments pour modifier certains aspects, tels que l'épaisseur par défaut des lignes, la taille du texte, l'orientation du schéma. Nous renvoyons à la documentation.

**Ligne 2** Un nœud TikZ, matérialisé par la commande `\node`, est un bloc de texte. On pourrait passer des options à cette commande, afin de modifier certains aspects comme la taille du texte.

- Lignes 3 et 4** `child` désigne ici une opération : associer un fils au nœud précédent. Ce fils est lui même un nœud (node).
- Lignes 5 à 12** Même chose que précédemment, sauf que le nœud D se voit attribuer des fils (E et F). Notez le nombre d’accolades et leurs imbrications : c’est ce qui permet à *TikZ* de construire correctement le *stemma*.
- Ligne 13** Le point-virgule est obligatoire après chaque commande *TikZ* (ici `\node`).
- Ligne 14** Fin du graphisme *TikZ*.

Évidemment un tel code est relativement complexe. C’est pourquoi nous conseillons de le commenter attentivement<sup>6</sup> et de prêter extrêmement attention aux accolades. Par ailleurs nous recommandons de mettre chaque graphisme *TikZ* dans un fichier séparé.

Notre *stemma* est relativement simple : il n’y a pas de « consanguinité ». Si cela avait été le cas la construction aurait été plus complexe : on n’aurait pas pu utiliser les opérations de types `child` et il aurait fallu positionner précisément les éléments en utilisant un système de coordonnées<sup>7</sup>.

## 8.3 Création de tableaux

⚡ La création de tableaux en  $\text{\LaTeX}$  nécessite une extrême rigueur. C’est à nos yeux l’un des éléments les plus complexes de  $\text{\LaTeX}$ . C’est pourquoi nous ne proposerons ici qu’un bref aperçu : on trouvera aisément des exemples plus détaillés dans d’autres ouvrages ou sur Internet<sup>8</sup>.

### 8.3.1 Syntaxe de base

La création d’un tableau simple se fait grâce à l’environnement `tabular`, d’après la syntaxe suivante :

```

1 | \begin{tabular}{<type de colonnes>}
2 |   Cellule & Cellule & Cellule \\
3 |   Cellule & Cellule & Cellule \\
4 | \end{tabular}

```

<sup>6</sup>Pour des raisons de place nous ne l’avons pas fait ici.

<sup>7</sup>Nous avons publié sur notre site Internet une série d’articles sur les *stemma* avec *TikZ* : Maïeul ROUQUETTE, *Stemma avec TikZ*, URL : <http://geekographie.maieul.net/88>.

<sup>8</sup>On pourra consulter des fichiers assez didactiques : Bertrand MASSON, *Les fiches de Bébert : Tableau*, URL : <http://bertrandmasson.free.fr/index.php?categorie4/latex-tableau>.

Le caractère & marque un changement de cellule. Le \\ marque un changement de ligne.

L'argument *<type de colonnes>* de l'environnement **tabular** sert à préciser :

1. Le nombre de cellules.
2. Le formatage du texte à l'intérieur des cellules.
3. Éventuellement les séparateurs verticaux des cellules.

Il existe trois types possibles de colonnes, qui précisent l'alignement des textes :

**l**                   Alignement à gauche.

**c**                   Centré.

**r**                   Alignement à droite.

Supposons que nous ayons un tableau à trois colonnes. Nous souhaitons que les cellules de la première colonne soient alignées à gauche, celles de la seconde centrées et celles de la dernière alignées à droite<sup>9</sup>. Il nous faut écrire :

```

1 | \begin{tabular}{|l|c|r|}
2 | Première cellule à gauche
3 | & Première cellule au centre
4 | & Première cellule à droite \\
5 | Seconde cellule à gauche
6 | & Seconde cellule au centre
7 | & Seconde cellule à droite \\
8 | \end{tabular}

```

Première cellule à gauche	Première cellule au centre	Première cellule à droite
Seconde cellule à gauche	Seconde cellule au centre	Seconde cellule à droite

Par défaut, le tableau va prendre la largeur minimale nécessaire à son affichage.

On peut utiliser l'environnement **tabular\*** qui propose un argument supplémentaire, correspondant à la largeur totale du tableau :

```

1 | \begin{tabular*}{<largeur du tableau>}{<type de colonnes>}
2 |    Cellule & Cellule & Cellule \\
3 |    Cellule & Cellule & Cellule \\
4 | \end{tabular*}

```

---

<sup>9</sup>Il s'agit ici d'une dérogation au principe de séparation de fond et de forme. Toutefois le lecteur pourra aisément, lorsqu'il l'aura assimilée, contourner cette dérogation et revenir dans le droit chemin.

Toutefois il peut être pratique d'indiquer des tailles de colonnes : en particulier lorsque nos cellules contiennent des paragraphes.

Pour ce faire trois types supplémentaires de colonnes nous sont proposés par le package *array*<sup>10</sup>. Ces types, à la différence des précédents, déterminent une largeur de colonne. Il faut pour les utiliser connaître les unités de mesure (☞ p. 237, E) de L<sup>A</sup>T<sub>E</sub>X.

Ces trois types de colonnes sont :

**p**{*largeur*}    texte justifié. La plus haute ligne du texte se trouve sur la ligne de base.

**m**{*largeur*}    texte justifié. La ligne du milieu du texte se trouve sur la ligne de base.

**b**{*largeur*}    texte justifié. La plus basse ligne du texte se trouve sur la ligne de base.

La ligne de base est une ligne spécifique, correspondant *grosso-modo* au milieu d'une cellule. Exemple :

```

1  \begin{tabular}{p{0.2\textwidth}|m{0.2\textwidth}|b{0.2\textwidth}|c}
2  Long texte dont la plus haute ligne
3  se trouve sur la ligne de base.
4  &
5  Long texte dont la ligne du milieu
6  se trouve sur la ligne de base.
7  &
8  Long texte dont la plus basse ligne
9  se trouve sur la ligne de base.
10 &
11 Ligne de base
12 \end{tabular}

```

Long texte dont la plus haute ligne se trouve sur la ligne de base.	Long texte dont la ligne du milieu se trouve sur la ligne de base.	Long texte dont la plus basse ligne se trouve sur la ligne de base.	Ligne de base
---	--	---	---------------

<sup>10</sup>En réalité le premier type ne nécessite pas que l'on charge le package.

✎ Il est possible de créer ses propres types de colonnes, afin par exemple d'en avoir avec une typographie particulière. Consultez le manuel du package *array*<sup>11</sup>.

✎ On peut utiliser le package *diagbox* afin d'avoir une cellule divisée en deux dans un sens diagonal, pour les tableaux à doubles entrées.

✎ Le package *tabularx* propose un environnement *tabularx* permettant de calculer automatiquement la taille des colonnes.

✎ Il existe également un package *tabbing* permettant de gérer des alignements de texte sans utiliser un tableau<sup>12</sup>.

✎ On peut simuler certaines fonctionnalités d'un tableur à l'aide du package *spreadtab*<sup>13</sup>.

## Séparation des cellules

On peut tracer des filets de séparation des cellules : pour un filet horizontal, il suffit d'utiliser la commande `\hline`. Pour un filet vertical, il faut mettre le caractère `|`<sup>14</sup> entre les types de cellules.

On trace des filets horizontaux plus petits grâce à `\cline{<col1>-<col2>}`, où `<col1>` et `<col2>` doivent être remplacés par les numéros de colonnes.

```

1  \begin{tabular}{l|c|r}
2  \hline
3  A & B & C \\
4  \hline
5  A1 & B1 & C1 \\
6  \cline{2-3}
7  A2 & B2 & C2 \\
8
9
10 \end{tabular}

```

A	B	C
A1	B1	C1
A2	B2	C2

<sup>11</sup>Frank MITTELBACH et David CARLISLE, « Defining your own column specifiers », dans : *A new implementation of LATEX's tabular and array environment*.

<sup>12</sup>Jean-Pierre DRUCBERT, *The Tabbing package*.

<sup>13</sup>Christian TELLECHEA, *spreadtab*.

<sup>14</sup>Ce caractère, appelé « pipe » ne doit pas être confondu avec un L minuscule.

## Fusion de cellules

Il est possible de fusionner des cellules. Pour une fusion de colonnes, il faut utiliser la commande suivante : `\multicolumn{< nombcol >}{< type >}{< contenu >}`.

`< nombcol >` est le nombre de colonnes à fusionner.

`< type >` est le type de colonne qui en résulte, selon la syntaxe vue plus haut.

`< contenu >` est le contenu de la ligne fusionnée.

Pour une fusion de lignes, il faut charger le package *multirow*. Celui-ci propose une commande `\multirow` dont voici la syntaxe simplifiée<sup>15</sup> :

`\multirow{< lignes >}{< largeur >}{< contenu >}`.

`< lignes >` est le nombre de lignes concernées par la fusion.

`< largeur >` est la largeur du texte. On peut mettre `*` pour laisser L<sup>A</sup>T<sub>E</sub>X s'en charger.

`< contenu >` est le contenu de la cellule issue de la fusion.

```

1  \begin{tabular}{|c|c|c|}
2  \hline
3      \multicolumn{3}{|c|}{Colonnes fusionnées = ABC0} \\
4  \hline
5      A1 & B1 & C1 \\
6  \hline
7      \multirow{2}{*}{Lignes fusionnés = A2-A3}
8      & B2 & C2 \\
9      \cline{2-3}
10     & B3 & C3 \\
11 \hline
12 \end{tabular}

```

Colonnes fusionnées = ABC0		
A1	B1	C1
Lignes fusionnés = A2-A3	B2	C2
	B3	C3

On remarquera les cellules vides correspondant aux anciennes cellules A2 et A3. La commande `\multirow` ne modifie pas la structure du tableau : elle indique simplement que le texte se situe sur plusieurs colonnes.

<sup>15</sup>Simplifiée car nous ne présentons pas les arguments optionnels : nous renvoyons à la documentation : Piet van OOSTRUM, Øystein BACHE et Jerry LEICHTER, *The multirow, bigstrut and bigdelim packages*.

### 8.3.2 Long tableau

Si vous fabriquez ainsi un long tableau, vous constaterez que les cellules dépassent dans la marge inférieure, jusqu'à sortir de la page, au lieu, comme il semblerait logique, de continuer sur les pages suivantes.

Pour éviter cela, il faut utiliser l'environnement **longtable** du package éponyme.

```

1  \usepackage{longtable}
2  ...
3  \begin{longtable}{types colonnes}
4  ...
5  \end{longtable}

```

La syntaxe d'un environnement **longtable** est identique à celle d'un environnement **tabular**. Toutefois le package propose deux commandes supplémentaires pour préciser les en-têtes et pieds de tableau, qui sont répétés sur chaque page. Ces commandes sont, respectivement, `\endhead` et `\endfoot`. Il existe aussi une commande `\endfirsthead` pour préciser l'en-tête spécifique à la première page du tableau et une commande `\endlastfoot` pour préciser le pied spécifique à la dernière page du tableau.

Ces quatre commandes sont optionnelles, elles doivent s'utiliser en début de tableau : il faut placer les éléments correspondants *avant* chacune des commandes, de la manière suivante :

```

1  \begin{longtable}{types colonnes}
2  Cellules d'en-tête.
3  \endhead
4  Cellules d'en-tête pour la première page.
5  \endfirsthead
6  Cellules de pied.
7  \endfoot
8  Cellules de pied pour la dernière page.
9  \endlastfoot
10 Cellules de corps de tableau.
11 \end{longtable}

```

↗ Avec **longtable**, il faut compiler deux fois le fichier `.tex`. En effet,  $\text{X}_{\text{E}}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  — comme d'ailleurs  $\text{E}_{\text{T}}\text{E}_{\text{X}}$  — n'arrive pas à déterminer la taille des colonnes à la première compilation : il procède à un calcul page par page. Cependant il stocke la taille maximale des colonnes dans un fichier temporaire, portant l'extension `.aux`. Lors de la seconde compilation, il se sert de ce fichier temporaire pour indiquer la taille des colonnes sur chacune des pages.

✎ Pour afficher un graphique statistique, nous recommandons d'utiliser le package *TikZ* (☞ p. 62, 2). Toutefois la maîtrise de cet outil peut se révéler relativement complexe. C'est pourquoi nous listons ici quelques packages utiles :

- Le package *csvpie* propose une fonction pour tracer un diagramme circulaire à partir d'un fichier `.csv`<sup>16</sup>.
- Le package *datatool* propose un certain nombre de graphiques statistiques. Il faut enregistrer les données selon une forme liée au package. Nous renvoyons à sa documentation<sup>17</sup>.

## 8.4 Où insérer les éléments non textuels ? la notion de flottants

Nous avons vu comment insérer des éléments non textuels. Cependant vous constaterez rapidement que la mise en forme n'est pas toujours des meilleures, l'élément s'insérant dans le texte à l'endroit précis où il a été appelé, ce qui peut entraîner des espaces blancs disgracieux. En outre, ces éléments non textuels disposent habituellement d'une légende, que nous ne savons pas encore mettre.

Pour résoudre ces deux problèmes — positionnement esthétique et légende —  $\text{\LaTeX}$  utilise la notion de flottant. Un flottant est donc un élément non textuel que  $\text{\LaTeX}$  essaie d'insérer au meilleur endroit du point de vue de l'esthétique et qui dispose (éventuellement) d'une légende.

Il existe deux types principaux de flottant :

- Les figures, générées par l'environnement **figure**.
- Les tableaux, générés par l'environnement **table**.

La syntaxe pour insérer le premier type est la suivante :

```

1 \begin{figure}[<paramètre de placement>]
2     Insertion de la figure ou de l'image,
3     suivant les codes montrés plus haut.
4     \caption{<Légende>}
5 \end{figure}

```

Celle pour insérer le second est la suivante :

```

1 \begin{table}[<paramètre de placement>]
2     Insertion d'un tableau suivant les codes montrés plus haut.
3     \caption{<Légende>}
4 \end{table}

```

<sup>16</sup>Sa documentation se trouve dans celle du package *csvtools* : Nicola TALBOT, « Pie Charts (*csv-sty*) », dans : *csvtools v1.24 : A LATEX 2 $\epsilon$  Package Providing Access to Data Saved in a CSV File*.

<sup>17</sup>Nicola TALBOT, *datatool v 2.03 : Databases and data manipulation*.

Si la commande `\caption` est insérée au-dessus de la figure, de l'image ou du tableau, la légende sera située au-dessus de l'élément et non en-dessous. Il est possible, mais déconseillé, de ne pas mettre de légende.

Nous avons un exemple de flottant dans ce chapitre : la figure 8.1.

✦ On pourrait vouloir changer les intitulés comme *figure* et *tableau*. Tout ceci est possible : nous en parlons plus loin, dans le passage consacré aux chaînes de langue de *polyglossia* (☞ p. 138, 17.3).

De même on pourrait souhaiter renvoyer automatiquement au numéro et la page d'une figure : nous en parlons dans le chapitre consacré à la navigation à l'intérieur d'un document (☞ p. 133, 16).

À noter que, comme les titres de partie, la commande `\caption` peut recevoir un argument optionnel qui compte pour titre dans la table des figures ou la table des tableaux (☞ p. 139, 17.4) : `\caption[<Légende pour la table>]{<Légende>}`.

✦ Il est parfois utile de centrer le contenu du flottant. Il s'agit ici d'une exception au principe de séparation du fond de la forme. Pour procéder à ce centrage, il faut utiliser la commande `\centering`.

```

1 | \begin{figure}[<paramètre de placement>]
2 |   \centering
3 |   Insertion de la figure ou de l'image,
4 |   suivant les codes montrés plus haut.
5 |   \caption{<Légende>}
6 | \end{figure}

```

Pour personnaliser les styles des légendes, la méthode la plus simple reste encore d'utiliser le package *caption*. Nous renvoyons à sa documentation<sup>18</sup>. Pour utiliser une légende en dehors d'un flottant, on peut utiliser le package *ccaption*<sup>19</sup>, qui permet d'autres choses encore.

Si l'on souhaite avoir deux éléments flottants en vis-à-vis, sur la page gauche et sur la page droite, on peut se servir du package *dpfloat*<sup>20</sup>.

Le package *endfloat*<sup>21</sup> permet, quant à lui, de placer systématiquement les flottants à la fin du travail.

Pour avoir des flottants orientés dans un autre sens que celui de la page, on utilisera le package *rotfloat*<sup>22</sup>.

Il est possible d'insérer au sein d'une figure des sous-figures, chacune d'entre elles disposant d'une légende, en plus de la légende principale ; il faut utiliser le *subcaption*<sup>23</sup>.

<sup>18</sup>Axel SOMMERFELDT, *Customizing captions of floating environments using the caption package*.

<sup>19</sup>Peter WILSON, Herries PRESS et Will ROBERTSON, *The ccaption package*.

<sup>20</sup>Jim FOX, *The dpfloat package : Formatting double-page floats in LATEX*.

<sup>21</sup>James Darrell MCCAULEY et Jeff GOLDBERG, *The endfloat package*.

<sup>22</sup>Axel SOMMERFELDT, *The rotfloat package*.

<sup>23</sup>Axel SOMMERFELDT, *The subcaption package*.

### 8.4.1 Choix de l'emplacement du flottant

Le  $\langle$ paramètre de placement $\rangle$  indique à  $\text{\LaTeX}$  comment placer idéalement les flottants. Il s'agit d'un paramètre indicatif que le compilateur essaie autant que possible de suivre, sans pour autant être contraint. Les valeurs sont les suivantes :

Valeur	Signification
h	À l'emplacement de l'appel au flottant
t	En haut d'une page
b	En bas d'une page
p	Sur une page dédiée aux flottants

Si le système des flottants permet généralement de conserver une mise en page correcte tout en n'éloignant pas trop l'élément de son emplacement, il arrive parfois que l'éloignement soit trop important.

La commande `\FloatBarrier` du package *placeins* permet de remédier à ce problème<sup>24</sup>. Tous les flottants appelés avant la commande sont placés avant celle-ci.

---

<sup>24</sup>Donald ARSENEAU, *The placeins package*.

# Deuxième partie

## Gérer sa bibliographie avec L<sup>A</sup>T<sub>E</sub>X

« Empoisonnée de dogmes et de mythes, notre opinion, même la moins ennemie des lumières, a perdu jusqu'au goût du contrôle. Le jour où, ayant pris soin d'abord de ne pas la rebuter par un oiseux pédantisme, nous aurons réussi à la persuader de mesurer la valeur d'une connaissance sur son empressement à tendre le cou d'avance à la réfutation, les forces de la raison remporteront une de leurs plus éclatantes victoires. C'est à la préparer que travaillent nos humbles notes, nos petites références taitillonnes que moquent aujourd'hui, sans les comprendre, tant de beaux esprits. »

Marc BLOCH, *Apologie pour l'histoire*, III, 1



# Chapitre 9

## Introduction à la gestion bibliographique avec $\LaTeX$

*Nous abordons maintenant un des atouts majeurs de  $\LaTeX$  pour les sciences humaines : la possibilité de gérer très simplement une bibliographie complexe, tout en conservant une certaine souplesse dans son affichage.*

### 9.1 Principe général

Il existe un module de gestion de la bibliographie dans la plupart des logiciels WYSIWYG, mais ceux-ci manquent en général de souplesse.

Conséquence, la grande partie des utilisateurs de ces logiciels gèrent leur bibliographie « à la main » — les références bibliographiques sont écrites textuellement, directement dans le fichier, ce qui implique beaucoup de contraintes :

- si on veut changer l'ordre des informations, ou en supprimer, il faut reprendre soi-même l'ensemble des références ;
- la gestion des versions abrégées des références est compliquée : il faut s'assurer soi-même que la référence a déjà été citée, mais pas trop loin, etc.
- Il faut copier-coller soi-même à la fin l'ensemble des références bibliographiques.

La logique de la gestion bibliographique en  $\LaTeX$  est simple. Un fichier `.bib` — éventuellement plusieurs — contient l'ensemble des références bibliographiques. Pour chaque référence, appelée « entrée », sont indiqués les éléments utiles : type de référence (livre, article, actes de colloque, etc.), titre, auteur, pagination,

sous-titre, éditeur, etc.<sup>1</sup> Chaque référence possède une clef unique permettant de la distinguer d'une autre.

Dans le document  $\LaTeX$ , on précise dans le préambule le chemin du fichier `.bib`. À chaque fois que l'on souhaite citer une référence, on utilise une commande en lui passant la clef en argument. Le package *biblatex* se charge alors d'afficher la référence selon un style de citation — c'est-à-dire une présentation — déterminé. Ainsi, si on veut changer l'ordre de présentation, par exemple intervertir l'éditeur et la ville de publication, il suffit de modifier le style, ce qui se fait aisément, puisque les styles sont une série de commandes  $\LaTeX$ . Par ailleurs *biblatex* gère automatiquement les références déjà citées et introduit tout seul, en fonction du style choisi, les abréviations universitaires.

☛ On pourrait se passer de *biblatex* pour gérer une bibliographie avec  $\LaTeX$ . Cependant, les fonctionnalités standards de  $\LaTeX$  pour la gestion bibliographique sont limitées et inadaptées aux sciences humaines.

Vous trouverez peut-être sur internet des fichiers `.bst`. Ces fichiers sont des styles bibliographiques mais pour les fonctionnalités standards de  $\LaTeX$ . Ils ne peuvent servir pour *biblatex*. De plus, il aisé d'avoir ses propres styles *biblatex*, tandis que la syntaxe des fichiers `.bst` est complexe<sup>2</sup>.

Enfin, à la fin du document, ou à tout autre endroit jugé utile, une (ou plusieurs) commande permet d'afficher la bibliographie, qui reprend les références citées<sup>3</sup> dans le document, en les classant et en les affichant selon un style de bibliographie, lui aussi personnalisable.

## 9.2 Une triple compilation

Jusqu'à maintenant, vous aviez compilé une seule fois votre fichier `.tex` avec  $X_{\text{g}}\LaTeX$ . La gestion bibliographique étant quelque chose d'assez complexe, il va falloir procéder à une triple compilation :

1. le fichier `.tex` avec  $X_{\text{g}}\LaTeX$ . Outre le fichier `.pdf`, cette compilation produit un fichier auxiliaire `.bbl` ;
2. ce fichier doit être lui même compilé avec le logiciel Biber ;
3. puis il faut re-compiler le fichier `.tex` avec  $X_{\text{g}}\LaTeX$ .

⚡ Ce manuel a été rédigé en prenant pour référence la version 2.0 du package *biblatex*. Il vous faut donc vérifier que vous disposez de cette version, ou d'une version ultérieure. Le cas échéant, mettez à jour *biblatex* ainsi que Biber ( voir p. 225, A.5) — en général, une version précise de *biblatex* ne fonctionne qu'avec une version précise de Biber.

<sup>1</sup>Par ailleurs ce fichier pourra contenir des commentaires utiles pour la préparation du travail : résumé, notes de lecture, cotation dans une bibliothèque.

<sup>2</sup>Elle est écrite en notation polonaise inversée.

<sup>3</sup>Il est aussi possible, si on le souhaite, d'ajouter une ou plusieurs références non citées.

⚡ La plupart des éditeurs spécialisés en  $\text{\LaTeX}$  ne permettent pas de compiler avec Biber, mais seulement avec BibTeX, qui est en quelque sort son ancêtre. Nous expliquons sur notre blog comment configurer certains de ces éditeurs pour utiliser Biber à la place de BibTeX<sup>4</sup>. Néanmoins, la méthode la plus simple reste encore d'utiliser le logiciel Latexmk ( p. 235, D), qui se chargera de procéder aux compilations.

---

<sup>4</sup>Maïeul ROUQUETTE, *Biber dans des éditeurs LaTeX*, URL : <http://geekographie.maieul.net/51>.



# Chapitre 10

## Remplir sa base de données bibliographique

*La première étape pour gérer une bibliographie avec  $\text{\LaTeX}$  est la constitution d'une base de données bibliographique contenant l'ensemble des références. Celle-ci est un simple fichier texte qui porte une extension **.bib** et possède une structure particulière.*

### 10.1 Structure de base d'un fichier **.bib**

Chaque élément d'une bibliographie est appelé « entrée ». Une entrée bibliographique se caractérise par :

<b>Un type</b>	s'agit-il d'un article, d'un livre, d'actes de colloque ?
<b>Une clef unique</b>	qui permet de distinguer une entrée bibliographique d'une autre. C'est elle qui est mentionnée dans vos fichiers <b>.tex</b> lorsque vous désirez insérer une référence bibliographique.
<b>Des champs</b>	contenant des indications sur l'ouvrage, tels que : auteur, titre, éditeur, etc.

L'avantage de ce système se comprend aisément. En effet, une correction dans la base de donnée bibliographique est automatiquement reportée dans le travail à chaque fois que l'entrée est mentionnée, moyennant une nouvelle compilation avec Biber.

Un fichier **.bib** est un fichier texte contenant une suite d'instructions, de la forme suivante :

```

1 | @type{clef,
2 |     Champ1 = {Valeur 1},
3 |     Champ... = {Valeur ...},
4 |     Champn = {Valeur n}
5 | }

```

Par exemple :

```

1 | @book{Urner1952,
2 |     Author = {Urner, Hans},
3 |     Title = {Die ausserbiblische Lesung
4 |     in Christlichen Gottesdienst},
5 |     Location = {Gottingen},
6 |     Year = {1952}
7 | }

```

Par ces lignes nous déclarons un livre (@book) auquel nous attribuons la clef Urner1952. Ce livre est une œuvre de Urner, Hans de son prénom ; il s'intitule *Die ausserbiblische Lesung in Christlichen Gottesdienst* et a été publié en 1952 à Göttingen.

Il peut être affiché dans le fichier .pdf sous la forme :

Hans URNER, *Die ausserbiblische Lesung in Christlichen Gottesdienst*, Göttingen, 1952

Un fichier .bib n'est rien d'autre qu'une série d'entrées de ce type. Cependant, comme il est assez aisé de « s'emmêler les pinceaux », notamment dans les ouvertures et fermetures des accolades, il vaut la peine d'utiliser un logiciel de gestion de bibliographies (☞ p. 230, B.2), permettant de créer facilement des fichiers au format .bib. En outre ces logiciels permettent en général des recherches croisées dans la base de données. Nous en présentons deux en annexe.

La documentation de *biblatex* contient l'ensemble des champs et des types possibles. Pour éviter de répéter ce manuel, nous vous présenterons les champs en les classant par catégorie. Mais auparavant nous allons donner quelques conseils sur le choix de la clef.

## 10.2 Le choix de la clef

Une entrée bibliographique doit avoir une clef unique. Cette clef doit comporter uniquement des caractères alphanumériques non accentués, éventuellement avec des tirets. Quelques conseils pour le choix de la clef :

### 10.2.1 Pour les œuvres contemporaines

Prendre le nom de l'auteur suivi de la date. Par exemple Urner1952.

### 10.2.2 Pour les œuvres anciennes

Prendre le nom de l'œuvre, éventuellement abrégée. Ainsi ContraFelix pour parler du *Contra Felicem* d'Augustin<sup>1</sup>. Si plusieurs œuvres d'auteurs différents portent le même nom, faire précéder du nom de l'auteur. On peut aussi pour les œuvres sans titre et sans auteur indiquer un numéro dans un catalogue de référence.

## 10.3 Les différents types d'entrées

Voici les différents types d'entrées disponibles<sup>2</sup>.

<b>@article</b>	Article d'un périodique. Pour les contributions de colloque, voir @inproceedings.
<b>@book</b>	Livre. Pour les actes de colloque, voir @proceedings.
<b>@booklet</b>	Livret. Texte sous la forme d'un livre mais sans éditeur commercial officiel.
<b>@bookinbook</b>	Livre dans un livre. Par exemple lorsque un auteur voit ses écrits rassemblés en un seul volume physique.
<b>@collection</b>	Ouvrage collectif mais avec des parties distinctes par auteur.
<b>@inbook</b>	Partie d'un livre.
<b>@incollection</b>	Partie individuelle (avec un auteur propre) dans un ouvrage collectif.
<b>@inproceedings</b>	Contribution à un colloque.
<b>@inreference</b>	Article de dictionnaire, d'encyclopédie, etc.
<b>@manual</b>	Manuel, pas nécessairement sous forme imprimée.
<b>@mvbook</b>	Livre en plusieurs volumes.
<b>@mvcollection</b>	Ouvrages collectifs en plusieurs volumes, avec des parties distinctes par auteurs.

<sup>1</sup>AUGUSTIN, *Contra Felicem*, dans : *Six traités anti-manichéens*, trad. par Régis JOLIVET et Maurice JOURION, Bibliothèque Augustinienne 17.

<sup>2</sup>À vrai dire il est possible de rajouter les siennes, mais nous vous le déconseillons.

<b>@mvproceedings</b>	Actes de colloque en plusieurs volumes.
<b>@mvreference</b>	Dictionnaire, encyclopédie, etc. en plusieurs volumes.
<b>@misc</b>	Entrée générique, pour tout type d'entrée non catégorisable. Par exemple : tableau, manuscrit, <i>ostracon</i> .
<b>@online</b>	Ressource internet. Si une œuvre existe aussi sous une autre forme, choisir le type d'entrée qui y correspond et utiliser le champ <code>url</code> .
<b>@patent</b>	Brevet industriel.
<b>@periodical</b>	Numéro précis d'un périodique.
<b>@report</b>	Rapport technique ou de recherche.
<b>@proceedings</b>	Acte de colloque.
<b>@suppbook</b>	Partie annexe d'un livre, comme par exemple la préface ou les appendices.
<b>@superperiodical</b>	Supplément à un numéro de périodique.
<b>@reference</b>	Dictionnaire, encyclopédie, etc.
<b>@thesis</b>	Thèse de doctorat, mémoire de maîtrise ou tout travail rédigé en vue de l'obtention d'un titre scolaire ou universitaire.
<b>@unpublished</b>	Ouvrage non publié.

Pour les éditions contemporaines d'œuvres anciennes, le choix d'une entrée de type `@book` peut se justifier, en dépit du caractère non livresque de certaines d'entre elles, comme c'est le cas par exemple des lettres d'Augustin. En revanche, pour désigner une œuvre non éditée, ou un manuscrit, nous conseillons le type `@misc`.

## 10.4 Les différents champs possibles

Nous ne listons pas ici tous les champs, mais seulement ceux qui peuvent rentrer dans les catégories les plus utiles.

### 10.4.1 Les champs de personnes

Ces champs servent à désigner des personnes qui ont participé au processus de production de l'œuvre : auteur, annotateur, éditeur (scientifique), etc. Il n'est évidemment pas nécessaire de remplir tous ces champs.

<b>afterword</b>	Auteur(s) de la postface.
<b>annotator</b>	Auteur(s) des annotations.
<b>author</b>	Auteur(s) de l'œuvre.
<b>bookauthor</b>	Auteur(s) du livre dans lequel l'œuvre est insérée.
<b>commentator</b>	Auteur(s) des commentaires.
<b>editor</b>	Éditeur(s) scientifique(s). On peut en préciser le rôle grâce au champ <b>editortype</b> .
<b>editora</b>	Éditeur(s) scientifique(s) ayant un autre rôle. On peut en préciser le rôle grâce au champ <b>editoratype</b> .
<b>editorb</b>	Éditeur(s) scientifique(s) ayant un autre rôle. On peut en préciser le rôle grâce au champ <b>editorbtype</b> .
<b>editorc</b>	Éditeur(s) scientifique(s) ayant un autre rôle. On peut en préciser le rôle grâce au champ <b>editorctype</b> <sup>3</sup> .
<b>foreword</b>	Auteur(s) de la préface.
<b>holder</b>	Titulaire d'un brevet industriel.
<b>introduction</b>	Auteur(s) de l'introduction.
<b>translator</b>	Traducteur(s).

Lorsque des champs possèdent des valeurs égales, par exemple les champs **editor** et **translator**, *biblatex* fusionne ces champs lors de l'affichage. Prenons ainsi l'entrée suivante :

```

1 | @book{DoctrineChretienne,
2 |   Author = {Augustin},
3 |   Editor = {Madeleine Moreau},
4 |   Entrysubtype = {traite},
5 |   Series = {Bibliothèque Augustinienne},
6 |   Title = {De doctrina christiana},
7 |   Translator = {Madeleine Moreau}}
```

<sup>3</sup>Pour ces quatre champs, se reporter à Philipp LEHMAN, « Editorial Roles », dans : *The biblatex Package*.

Elle est affichée ainsi :

AUGUSTIN, *De doctrina christiana*, éd. établie et trad. par Madeleine MOREAU, Bibliothèque Augustinienne

### Comment entrer un nom de personne

Ces différents champs prennent comme valeur un ou plusieurs noms de personne. S'il y a plusieurs noms, il suffit de les séparer par le mot-clef « and ». Par exemple pour les auteurs de l'ouvrage que vous avez entre les mains :

```
1 | author = {Maïeul Rouquette and Enimie Rouquette and Brendan Chabannes}
```

Un nom contient les éléments suivant :

**Prénom(s)** L'initiale doit être en majuscule, le reste en minuscules. Ainsi : « Albert ». Il vaut mieux mettre le prénom complet : on peut confier le travail de limitation à une minuscule à *biblatex*<sup>4</sup>.

**Nom** L'initiale doit être en majuscule, le reste en minuscules. Ainsi : « Londres ». *Biblatex* se charge le cas échéant de mettre en petites capitales.

**Particule (option)** Elle doit être entièrement en minuscules. Ainsi : « de ».

**Suffixe (option)** L'initiale doit être en majuscule. Ce type de donnée est plutôt anglo-saxonne. Exemple : « Junior ».

En ce qui concerne l'ordre des éléments, il peut être :

- « Prénoms (particule) Nom » ;
- « particule Nom, (suffixe) Prénoms ».

Ainsi les entrées Victor Marie Hugo et Hugo, Victor Marie sont équivalentes. Dans le premier cas, Biber considère que le dernier mot commençant par une majuscule est le nom de famille. Dans le second cas, il considère l'ensemble situé avant la virgule comme le nom de famille, ce qui est utile pour les noms composés. Ainsi pour parler de Charles De Gaulle : De Gaulle, Charles.

Si nous parlons de Simone de Beauvoir, le « de » étant une particule nous pouvons utiliser la première syntaxe : dans ce cas Biber considère tout ce qui suit la particule comme constituant le nom : Simone de Beauvoir. Mais la seconde syntaxe fonctionne également : de Beauvoir, Simone.

<sup>4</sup>Il faut pour cela passer l'option `firstnits=true` lors du chargement du package.

Pour distinguer Alexandre Dumas père d'Alexandre Dumas fils, on peut utiliser le suffixe : Dumas, Fils, Alexandre.

Le cas des auteurs anciens, où l'on écrit souvent le prénom suivi de la ville d'exercice ou de naissance, comme par exemple pour Grégoire de Tours, est problématique. Si j'écris : Grégoire de Tours Biber va comprendre qu'il s'agit d'une personne prénommée « Grégoire », dont le nom est « Tours » et la particule « de ». Par conséquent il va l'afficher sous la forme « Tours, Grégoire, de ». Pour éviter ce problème, il suffit d'utiliser des accolades : {Grégoire de Tours}.

Cette méthode peut servir aussi pour les institutions auteurs d'ouvrages. Par exemple : {Centre National de la Recherche Scientifique}.

Pour résumer, voici un exemple de quelques entrées correctes<sup>5</sup>.

```

1  @book{HugoMiserable,
2      title      = {Les misérables},
3      author     = {Victor Marie Hugo}}
4  @book{HugoLegende,
5      title      = {La légende des siècles},
6      author     = {Hugo, Victor Marie}}
7  @book{DeGaulle,
8      title      = {Mémoires},
9      author     = {De Gaulle, Charles}}
10 @book{BeauvoirSexe,
11     title      = {Le Deuxième Sexe},
12     author     = {Simone de Beauvoir}}
13 @book{BeauvoirMemoires,
14     title      = {Mémoires d'une jeune fille rangée},
15     author     = {de Beauvoir, Simone}}
16 @book{Dumas,
17     title      = {La dame aux camélias},
18     author     = {Dumas, Fils, Alexandre}}
19 @book{Gregoire,
20     title      = {Histoire des Francs},
21     author     = {{Grégoire de Tours}}}
```

Victor Marie HUGO, *Les misérables*  
 Victor Marie HUGO, *La légende des siècles*  
 Charles DE GAULLE, *Mémoires*  
 Simone de BEAUVOIR, *Le Deuxième Sexe*  
 Simone de BEAUVOIR, *Mémoires d'une jeune fille rangée*  
 Alexandre DUMAS Fils, *La dame aux camélias*  
 GRÉGOIRE DE TOURS, *Histoire des Francs*

<sup>5</sup>Le lecteur exigeant pardonnera aux auteurs de ne mettre ici que les noms d'auteurs et titres d'œuvres.

👉 En ce qui concerne les œuvres anonymes, la solution est évidemment de ne rien mettre dans le champ **author**. Cependant l’affichage par défaut de ces œuvres anonymes n’est pas nécessairement satisfaisant. Il est possible de le modifier : nous expliquons comment faire sur notre blog<sup>6</sup>, mais pour comprendre notre article il vous faut lire les prochains chapitres de cette partie.

## 10.4.2 Champs de titre

<b>booksubtitle</b>	Sous-titre du livre dans lequel l’entrée se situe.
<b>booktitle</b>	Titre du livre dans lequel l’entrée se situe.
<b>booktitleaddon</b>	Ajout au titre du livre dans lequel l’entrée se situe.
<b>chapter</b>	Chapitre d’un livre. Pour les entrées de type <code>@inbook</code> .
<b>eventitle</b>	Titre du colloque, pour les entrées de type <code>@proceedings</code> et <code>@inproceedings</code> .
<b>issuesubtitle</b>	Sous-titre d’un numéro spécifique d’un périodique. Pour les entrées de type <code>@periodical</code> , le sous-titre du périodique doit aller dans le champ <b>subtitle</b> , celui du numéro dans le champ <b>issuesubtitle</b>
<b>issuetitle</b>	Titre d’un numéro spécifique d’un périodique. Pour les entrées de type <code>@periodical</code> , le titre du périodique doit aller dans le champ <b>title</b> , celui titre du numéro dans le champ <b>is-suetitle</b> .
<b>journalsubtitle</b>	Sous-titre d’un périodique.
<b>journaltitle</b>	Titre d’un périodique. Le champ <b>journal</b> est un alias de ce champ <sup>7</sup> .
<b>mainsubtitle</b>	Sous-titre d’une œuvre en plusieurs volumes.
<b>maintitle</b>	Titre d’une œuvre en plusieurs volumes. Le titre du volume spécifique à une entrée correspond au champ <b>title</b> .
<b>maintitleaddon</b>	Ajout au titre d’une œuvre en plusieurs volumes.
<b>origtitle</b>	Titre original de l’œuvre, si traduction. N’est pas affiché en standard.

<sup>6</sup>Maïeul ROUQUETTE, *Gestions des sources anonymes avec biblatex*, URL : <http://geekographie.maieul.net/66>.

<sup>7</sup>Ce qui signifie que remplir le champ **journal** revient à remplir ce champ.

<b>subtitle</b>	Sous-titre de l'œuvre.
<b>title</b>	Titre de l'œuvre.
<b>titleaddon</b>	Ajout au titre de l'œuvre. Dans ce manuel, nous conseillons de l'utiliser pour les divisions de source (☞ p. 112, 13.4).

Voici quelques exemples afin de comprendre comment utiliser ces champs :

### Un livre avec un sous-titre

```

1 | @book{Saxer1980,
2 |     Author = {Victor Saxer},
3 |     Entrysubtype = {etude},
4 |     Location = {Paris},
5 |     Publisher = {Beauchesne},
6 |     Subtitle = {Les témoignages de Tertullien, Cyprien
7 |     et Augustin à la lumière de
8 |     l'archéologie africaine},
9 |     Title = {Morts, martyrs, reliques en Afrique chrétienne
10 |     aux premiers siècles},
11 |     Year = {1980}}
```

Victor SAXER, *Morts, martyrs, reliques en Afrique chrétienne aux premiers siècles : Les témoignages de Tertullien, Cyprien et Augustin à la lumière de l'archéologie africaine*, Paris : Beauchesne, 1980

### Un livre situé dans un recueil

```

1 | @bookinbook{ContraFelix,
2 |     Author = {Augustin},
3 |     Booktitle = {Six traités anti-manichéens},
4 |     Number = {17},
5 |     Series = {Bibliothèque Augustinienne},
6 |     Title = {Contra Felicem},
7 |     Translator = {Régis Jolivet and Jourion, Maurice}}
```

AUGUSTIN, *Contra Felicem*, dans : *Six traités anti-manichéens*, trad. par Régis JOLIVET et Maurice JOURION, Bibliothèque Augustinienne 17

## Un article dans une revue

```

1 | @article{Junod1992,
2 |     Author = {Eric Junod},
3 |     Entrysubtype = {etude},
4 |     Journaltitle = {Apocrypha},
5 |     Pages = {17-46},
6 |     Title = {\enquote{Apocryphes du Nouveau Testament}
7 |       : une appellation erronée
8 |       et une collection artificielle},
9 |     Volume = {3},
10 |    Year = {1992}}

```

Eric JUNOD, « “Apocryphes du Nouveau Testament” : une appellation erronée et une collection artificielle », dans : *Apocrypha* 3 (1992), p. 17–46

### 10.4.3 Champs d’informations sur la publication

<b>address</b>	Lieu de publication. Alias du champ <b>location</b> .
<b>date</b>	Date de publication, sous la forme AAAA-MM-JJ/AAAA-MM-JJ. La première date indiquée correspond à la date de début, la seconde à celle de fin. Pour n’indiquer qu’une date de début, mettre AAAA-MM-JJ/. On peut ne pas indiquer le mois ou le jour. On peut également utiliser les champs <b>year</b> et <b>month</b> à la place.
<b>edition</b>	Numéro d’édition si plusieurs éditions existent. Doit être un entier ou bien une chaîne de caractères.
<b>eventdate</b>	Date du colloque pour les entrées de type <b>@proceedings</b> et <b>@inproceedings</b> .
<b>howpublished</b>	Pour les entrées de type <b>@misc</b> , précise le mode de publication.
<b>institution</b>	Institution dans laquelle l’œuvre a été produite. Typiquement pour les entrées de type <b>@thesis</b> .
<b>issue</b>	Détail d’un numéro spécifique d’un périodique (par exemple « numéro d’été »). On préférera les champs <b>year</b> , <b>month</b> .
<b>language</b>	Langue de l’œuvre. Le nom de la langue doit, idéalement, être mis comme indiqué dans la documentation de <i>polyglossia</i> <sup>8</sup> .

<sup>8</sup>CHARETTE, *Polyglossia*, *op. cit.*

<b>location</b>	Lieu d'impression.
<b>month</b>	Mois de publication. Doit être un entier compris entre 1 et 12.
<b>number</b>	Numéro d'un périodique ou numéro au sein d'une collection.
<b>organization</b>	Organisation à l'origine d'un manuel ou d'une page internet.
<b>origdate</b>	Date de l'édition originale.
<b>origlanguage</b>	Langue originelle. Le nom de la langue doit, idéalement, être mis comme indiqué dans la documentation de <i>polyglossia</i> <sup>9</sup> .
<b>origlocation</b>	Lieu d'impression de l'édition originale.
<b>origpublisher</b>	Éditeur (commercial) de l'édition originale.
<b>pages</b>	Pages de l'article ou de la partie du livre étudiée.
<b>pagetotal</b>	Nombre total de pages.
<b>part</b>	Pour les livres en plusieurs volumes <i>physiques</i> , indique le numéro du volume physique. Le numéro du volume <i>logique</i> est à indiquer dans le champ <b>volume</b> .
<b>publisher</b>	Éditeur commercial.
<b>pubstate</b>	Pour les œuvres qui ne sont pas encore imprimées, indique le statut :
	<b>inpress</b> œuvre sous presse.
	<b>inpreparation</b> œuvre en préparation.
	<b>submitted</b> œuvre soumise à évaluation.
<b>type</b>	Pour les entrées de type @thesis, @patent et @report, précise le type de travail.  On peut y mettre une valeur personnelle, ou bien prendre l'une des valeurs prédéfinies. Dans ce cas la valeur est automatiquement traduite dans la langue du document.  Pour les entrées de type @thesis, deux possibilités :
	<b>mathesis</b> mémoire de master.
	<b>phdthesis</b> thèse de doctorat.

---

<sup>9</sup>*Ibid.*

Pour les entrées de type @patent, plusieurs possibilités :

<b>patentde</b>	Brevet allemand.
<b>patenteu</b>	Brevet européen.
<b>patentfr</b>	Brevet français.
<b>patentuk</b>	Brevet britannique.
<b>patentus</b>	Brevet états-unien.
<b>etc.</b>	

Pour les entrées de type @report :

<b>techreport</b>	rapport technique.
<b>resreport</b>	rapport de recherche.

<b>url</b>	Url (adresse électronique) d'une publication en ligne.
<b>urldate</b>	Date à laquelle une publication électronique a été consultée.
<b>venue</b>	Lieu du colloque pour les entrées du type @proceedings et @inproceedings.
<b>version</b>	Numéro de révision d'un manuel, d'un logiciel.
<b>volume</b>	Volume dans une œuvre en plusieurs volumes.
<b>volumes</b>	Nombre de volumes dans une œuvres en plusieurs volumes.
<b>year</b>	Année de publication.

Évidemment il n'est pas nécessaire de remplir systématiquement tous ces champs : le lecteur sait mieux que nous lesquels remplir en fonction de ses besoins. Certains champs peuvent contenir plusieurs valeurs, qu'il suffit de séparer par le mot-clef « and ». Prenons un livre copublié par les éditions « Labor et Fides » et « Cerf » : pour indiquer les deux éditeurs, il faut mettre : `publisher ={Labor et Fides and Cerf}`.

#### 10.4.4 Les champs d'identification légale

Il peut être utile d'indiquer des informations comme l'ISBN, etc. Voici les champs possibles<sup>10</sup>. Le lecteur curieux trouvera aisément des informations sur leurs significations.

<sup>10</sup>Par défaut, *bibtex* imprime ces champs s'ils sont remplis. Il est toutefois possible de ne pas les afficher en passant l'option `isbn=false` au chargement du package, voir : Philipp LEHMAN, « Style specific », dans : *The biblatex Package*.

<b>eid</b>	Identifiant électronique d'une entrée de type @article.
<b>isan</b>	<i>International Standard Audiovisual Number</i> , pour les entrées de type audiovisuel.
<b>isbn</b>	<i>International Standard Book Number</i> , pour les livres.
<b>ismn</b>	<i>International Standard Music Number</i> , pour les musiques imprimées, comme par exemple les partitions.
<b>isrn</b>	<i>International Standard Technical Report Number</i> , pour les rapports techniques.
<b>issn</b>	<i>International Standard Serial Number</i> , pour les numéros de revues.
<b>iswc</b>	<i>International Standard Work Code</i> pour les œuvres musicales.

### 10.4.5 Champs d'annotations

Par défaut ces champs ne sont pas imprimés.

<b>abstract</b>	Résumé de l'œuvre.
<b>annotation</b>	Annotation sur l'œuvre.
<b>file</b>	Adresse d'une version informatique locale du travail.
<b>library</b>	Annotation sur la disponibilité en bibliothèque, par exemple le nom de la bibliothèque et la cotation.

 Il peut être intéressant de produire ainsi une bibliographie commentée. L'auteur de ces lignes a publié sur son site une méthode pour cela<sup>11</sup>. Nous conseillons toutefois, avant de mettre en œuvre cette méthode, de lire les quelques chapitres qui vont suivre.

Il existe d'autres champs : si le cœur vous en dit, vous pouvez toujours consulter le manuel de *biblatex*<sup>12</sup>.

<sup>11</sup>Maïeul ROUQUETTE, *Une bibliographie commentée*, URL : <http://geekographie.maieul.net/61>.

<sup>12</sup>Philipp LEHMAN, « Entry Fields », dans : *The biblatex Package*.



# Chapitre 11

## Indiquer des références bibliographiques

Nous avons vu plus haut les différentes manières de citer un texte (☞ p. 45, 6) dans le corps du travail. Nous avons également vu comment remplir une base de données bibliographique (☞ p. 79, 10).

Il ne nous reste plus qu'à mettre en relation les textes cités avec la base de données constituée pour avoir un travail correct, en indiquant les références des citations. C'est l'objet de ce chapitre.

### 11.1 Appel du package *biblatex*

La gestion bibliographique s'effectue avec le package *biblatex*. Dans le préambule, on appelle le package dans sa forme la plus simple par la commande : `\usepackage{biblatex}`.

Cependant, le package dispose de nombreuses options<sup>1</sup>. Celle qui nous intéresse pour le moment est `citestyle`, qui permet de définir la manière dont les références bibliographiques sont affichées, notamment lorsqu'elles sont citées à plusieurs reprises.

Il existe un nombre important de styles de citation livrés en standard. Nous mentionnons ici les principaux<sup>2</sup> :

**numeric**                    chaque entrée se voit attribuer un numéro, qui est appelé lorsque l'on renvoie à cette entrée. La bibliographie finale indique les correspondances.

---

<sup>1</sup>Philipp LEHMAN, « Package Options », dans : *The biblatex Package*.

<sup>2</sup>Se reporter pour plus de détails à Philipp LEHMAN, « Citation Styles », dans : *The biblatex Package*.

<b>authortitle</b>	sont indiqués seulement l'auteur et le titre de l'œuvre.
<b>verbose</b>	la description complète de l'entrée est donnée la première fois, une version abrégée est affichée ensuite.
<b>verbose-ibid</b>	la description complète de l'entrée est donnée, mais si un passage est cité plusieurs fois de suite, l'abréviation <i>ibidem</i> est utilisée.
<b>verbose-note</b>	la description complète de l'entrée est donnée à sa première mention, puis une version abrégée est utilisée.
<b>verbose-trad1 ; verbose-trad2 ; verbose-trad3</b>	la description complète de l'entrée est utilisée, puis les abréviations universitaires de type <i>op. cit.</i> , <i>ibidem</i> . etc. sont utilisées ; <i>biblatex</i> calcule automatiquement l'opportunité d'utiliser une abréviation universitaire, selon l'endroit où l'ouvrage aura été cité précédemment. Voir le manuel pour la description complète des différences entre ces trois styles.

On comprend ici l'un des grands intérêts de  $\LaTeX$  : plus de perte de temps à se demander : « Faut-il que j'utilise ici une version abrégée de la référence ? », *biblatex* le fait pour vous.

Voici donc comment doit se faire l'appel au package si nous choisissons le style de citation `verbose-trad-2` :

```
1 | \usepackage[citestyle=verbose-trad2]{biblatex}
```

Pour les sciences humaines, nous recommandons les styles de la famille `verbose`.

## 11.2 Appel de la base de donnée bibliographique

Pour que  $\LaTeX$  sache où chercher les références bibliographiques, il faut lui signaler quel est le fichier `.bib` à utiliser, pour cela il suffit d'utiliser dans le préambule la commande : `\bibliography{<chemin du fichier>}`.

Le chemin du fichier s'indique de la même manière que n'importe quel chemin de fichier (☞ p. 31, 3.2.1).

⚡ Il est possible d'appeler plusieurs fichiers bibliographiques. Nous le déconseillons, dans la mesure où cela contraint à vérifier qu'il n'y ait pas plusieurs entrées avec la même clef.

## 11.3 Citation d'un élément bibliographique

L'ensemble des commandes de citation ont la syntaxe :

`\langle prefix \rangle cite[\langle prenote \rangle][\langle postnote \rangle]{\langle clef \rangle}`, `\langle prefix \rangle` indique où et comment doit apparaître la référence : directement dans le texte, entre parenthèses, en note de bas de page, etc.

Par exemple, nous souhaitons citer un livre de Victor Saxer que nous avons entré de cette manière dans la base de données :

```

1  @book{Saxer1980,
2      Author = {Victor Saxer},
3      Entrysubtype = {etude},
4      Location = {Paris},
5      Publisher = {Beauchesne},
6      Subtitle = {Les témoignages de Tertullien, Cyprien
7      et Augustin à la lumière de
8      l'archéologie africaine},
9      Title = {Morts, martyrs, reliques en Afrique chrétienne
10     aux premiers siècles},
11     Year = {1980}}
```

Nous écrivons `\cite{Saxer1980}`. Après la troisième compilation (☞ p. 76, 9.2), la référence apparaît selon le style choisi lors de l'appel à *biblatex*. Ainsi, pour un style de la famille *verbose* :

Victor SAXER, *Morts, martyrs, reliques en Afrique chrétienne aux premiers siècles : Les témoignages de Tertullien, Cyprien et Augustin à la lumière de l'archéologie africaine*, Paris : Beauchesne, 1980

L'usage en sciences humaines est de citer en note de bas de page. C'est pourquoi il vaut mieux utiliser : `\footcite{Saxer1980}`, qui met la référence en note de bas de page, en ajoutant automatiquement le point de fin de note. On peut également décider d'utiliser la commande `\parencite`, pour citer entre parenthèses.

Mais on peut aussi choisir la commande `\autocite`. Le mode de citation (note de bas page, citation directe, citation entre parenthèse, etc.) dépend alors du style de citation choisi : pour les styles de la famille *verbose*, la référence est mise en note de bas de page.

 Par défaut les champs bibliographiques sont séparés par des points. Si vous êtes pressé d'avoir des virgules à la place, rendez-vous au chapitre 14.

### 11.3.1 Les arguments *⟨prenote⟩* et *⟨postnote⟩*

Supposons que nous souhaitons afficher un texte avant notre référence. Par exemple : « Voir également ». On utilise l'argument optionnel *⟨prenote⟩*.

```
1 | Blabla \autocite[Voir également][]{Saxer1980} blablabla.
```

Blabla Voir également Victor SAXER, *Morts, martyrs, reliques en Afrique chrétienne aux premiers siècles : Les témoignages de Tertullien, Cyprien et Augustin à la lumière de l'archéologie africaine*, Paris : Beauchesne, 1980 blablabla.

On peut également vouloir afficher quelque chose après la référence. On utilise dans ce cas l'argument *⟨postnote⟩*.

```
1 | Blabla \autocite[Voir également][qui porte sur un
2 | sujet similaire.]{Saxer1980} blabla
```

Blabla Voir également Victor SAXER, *Morts, martyrs, reliques en Afrique chrétienne aux premiers siècles : Les témoignages de Tertullien, Cyprien et Augustin à la lumière de l'archéologie africaine*, Paris : Beauchesne, 1980, qui porte sur un sujet similaire. blabla

⚡ Si on utilise l'argument *⟨prenote⟩*, il est obligatoire d'indiquer un argument *⟨postnote⟩*, fût-il vide. En l'absence de cet argument, *biblatex* interprète ce que vous pensiez être *⟨prenote⟩* comme *⟨postnote⟩*.

### 11.3.2 L'argument *⟨postnote⟩* et la numérotation des pages

L'argument *⟨postnote⟩* peut servir à indiquer les pages précises du passage cité<sup>3</sup>. Il suffit pour cela qu'il contienne une valeur de type numérique, en chiffres arabes ou romains, ou bien une séquence de valeurs numériques.

Par exemple, pour citer la page 25 :

```
1 | \autocite[25]{Saxer1980}
```

Pour citer les pages 25 à 35 :

```
1 | \autocite[25-35]{Saxer1980}
```

Ou encore les pages 25 et 35 :

---

<sup>3</sup>On consultera pour plus de détails : Philipp LEHMAN, « Page Numbers in Citations », dans : *The biblatex Package*.

```
1 | \autocite[25 \& 35]{Saxer1980}
```

Pour non seulement citer la page précise mais aussi ajouter autre chose dans l'argument  $\langle postnote \rangle$ , utilisons la commande `\pno`, afin que *biblatex* insère lui même le « p. » :

```
1 | \autocite[\pno-22, avec lequel nous sommes en désaccord.]{Saxer1980}
```

Victor SAXER, *Morts, martyrs, reliques en Afrique chrétienne aux premiers siècles : Les témoignages de Tertullien, Cyprien et Augustin à la lumière de l'archéologie africaine*, Paris : Beauchesne, 1980, p. 22, avec lequel nous sommes en désaccord.

On pourra également utiliser les commandes `\nopp` pour ne pas afficher de préfixe de pagination, `\psq` pour indiquer qu'il faut également prendre la page suivante et `\psqq` pour indiquer qu'il faut prendre les pages suivantes.

### Problème avec le champ pages

Un problème se pose lorsqu'il y a déjà un champ `pages` de rempli, pour un article par exemple. On se retrouve en effet avec deux numérotations de pages : celle du champ `pages` et celle du passage précis que l'on cite, indiquée par l'option  $\langle postnote \rangle$ . Prenons l'entrée suivante :

```
1 | @article{Junod1992,
2 |     Author = {Eric Junod},
3 |     Entrysubtype = {etude},
4 |     Journaltitle = {Apocrypha},
5 |     Pages = {17-46},
6 |     Title = {\enquote{Apocryphes du Nouveau Testament}
7 |         : une appellation erronée
8 |         et une collection artificielle},
9 |     Volume = {3},
10 |     Year = {1992}}
```

Et appelons-la avec le code suivant :

```
1 | \cite[24]{Junod1992}
```

On obtient ce résultat :

Eric JUNOD, « “Apocryphes du Nouveau Testament” : une appellation erronée et une collection artificielle », dans : *Apocrypha* 3 (1992), p. 17–46p. 24

Pour éviter cet inconvénient, il faut, si on utilise un style de type verbose, passer une option lors de l'appel au package `citepages=omit`

```
1 | \usepackage[citestyle=verbose,citepages=omit]{biblatex}
```

Désormais écrire `\cite[24]{Junod1992}` affiche correctement :

Eric JUNOD, « Apocryphes du Nouveau Testament » : une appellation erronée et une collection artificielle », dans : *Apocrypha* 3 (1992), p. 24

En revanche le problème demeure si l'on souhaite insérer du texte dans l'argument `<postnote>` après le numéro de page.

```
1 | \cite[\pno-24, passage fort intéressant.]{Junod1992}
```

Eric JUNOD, « Apocryphes du Nouveau Testament » : une appellation erronée et une collection artificielle », dans : *Apocrypha* 3 (1992), p. 17–46, p. 24, passage fort intéressant.

Comme vous pouvez le constater, le champ `pages` continue à être affiché. Il s'agit d'une limite de *biblatex*. Nous ne trouvons pas d'autre solution que de le contourner en indiquant le texte qui suit la référence en dehors de l'argument `<postnote>`.

Dans le cas d'une citation en note de bas de page, il faut donc écrire :

```
1 | \footnote{\cite[24]{Junod1992}, passage fort intéressant.}
```

✈ Pour les sources anciennes, on aimerait pouvoir indiquer non seulement la pagination dans l'édition contemporaine mais aussi la division de source (livre, chapitre, paragraphe, etc.). Malheureusement on ne peut avec *biblatex* indiquer qu'un seul élément variable lors de l'utilisation d'une commande de citation.

On peut contourner ce problème en créant des sous-entrées bibliographiques (☞ p. 112, 13.4), dont nous parlons dans un autre chapitre.

👉 Pour les œuvres en plusieurs volumes, on peut utiliser les commandes `\volcite`, `\pvolcite`, `\fvolcite`, `\avolcite` correspondant aux commandes respectives `\cite`, `\parencite`, `\footcite`, `\autocite`.

La syntaxe est : `\{<prefix>cite[<prenote>]{<volume>}[<page>]{<clef>}`.

## Type de pagination

On peut préciser dans sa base de données le type de pagination d'une entrée : page-t-on en pages, en colonnes, etc ? On utilise pour cela le champ **pagination**, en lui donnant l'une des valeurs suivantes :

<b>page</b>	la pagination est sous forme de pages. C'est la valeur standard.
<b>column</b>	la pagination est sous forme de colonnes.
<b>line</b>	la pagination est sous forme de lignes.
<b>verse</b>	la pagination est sous forme de versets / de vers.
<b>section</b>	la pagination est sous forme de sections.
<b>paragraph</b>	la pagination est sous forme de paragraphes.
<b>none</b>	la pagination est libre.

Le champ **pagination** sert lorsque l'on met une indication numérique dans l'argument  $\langle postnote \rangle$ . En revanche si on n'utilise pas d'argument  $\langle postnote \rangle$  et qu'on laisse le contenu du champ **pages**, il est sans effet. Prenons l'entrée suivante :

```

1 | @book{desnos,
2 |   Author = {Robert Desnos},
3 |   Pagination = {verse},
4 |   Pages = {1-3},
5 |   Title = {La fourmi}}
```

Appelons-la en utilisant l'argument  $\langle postnote \rangle$ , puis sans l'utiliser.

```

1 | \autocite[2]{desnos}
2 |
3 | \autocite{desnos}
```

Robert DESNOS, *La fourmi*, v. 2  
 Robert DESNOS, *La fourmi*, p. 1–3

On constate que dans le deuxième cas, **pagination** n'est pas pris en compte. Il faut dans ce cas mettre le type de pagination dans le champ **bookpagination** :

```

1 | @book{desnos,
2 |   Author = {Robert Desnos},
3 |   Pagination = {verse},
4 |   Bookpagination = {verse},
5 |   Pages = {1-3},
6 |   Title = {La fourmi}}
```

Robert DESNOS, *La fourmi*, v. 2  
 Robert DESNOS, *La fourmi*, v. 1–3

☞ On peut éviter de dupliquer l’information dans le champ **bookpagina-tion** en insérant la ligne suivante en début de fichier **.tex** :

```
\DeclareFieldFormat{pages}{\mkpageprefix[pagination]{#1}}
```

La commande `\DeclareFieldFormat` indique la manière d’afficher le champ **pages** : en passant son contenu (**#1**) à une fonction `\mkpageprefix`, à qui on indique de se baser sur le champ **pagination** pour afficher le préfixe de page<sup>4</sup>.

☞ Il est possible de définir ses propres types de pagination en déclarant des nouvelles chaînes de langues (☞ p. 118, 14.4). Nous détaillons sur notre blog la méthode<sup>5</sup>.

## 11.4 Citation de plusieurs œuvres

Il est possible de citer plusieurs œuvres d’une traite, en utilisant une commande dont la syntaxe de base est :

```
\<prefix>cites[<prenote>][<postnote>]{<clef>}[<prenote>][<postnote>]{<clef>}...
```

On peut citer à la suite autant d’œuvres que souhaité. Chaque élément cité a droit à son argument `<prenote>` et `<postnote>`, qui s’utilise de la même manière que pour les citations simples. Voici un exemple d’utilisation :

```
1 | \autocites{Saxer1980}{Junod1992}
```

Victor SAXER, *Morts, martyrs, reliques en Afrique chrétienne aux premiers siècles : Les témoignages de Tertullien, Cyprien et Augustin à la lumière de l’archéologie africaine*, Paris : Beauchesne, 1980 ; Eric JUNOD, « Apocryphes du Nouveau Testament » : une appellation erronée et une collection artificielle », dans : *Apocrypha* 3 (1992), p. 17–46

Il est possible de préciser un texte à afficher avant la liste des références, ainsi qu’un texte à afficher après.

```
1 | \<prefix>cites(Texte avant)(Texte après){Saxer1980}{Junod1992}
```

<sup>4</sup>Voir Philipp LEHMAN, « Formatting Directives », dans : *The biblatex Package*.

<sup>5</sup>Maïeul ROUQUETTE, *Un nouveau type de pagination*, URL : <http://geekographie.maieul.net/69>.

Les séparateurs de citations sont par défaut des points-virgules. Il est possible de les modifier globalement, nous en parlons plus loin (☞ p. 117, 14.2).

Toutefois, si l'argument `<postnote>` d'un élément de la commande de citation multiple finit par un signe de ponctuation, alors le point-virgule n'apparaît pas entre cet élément et le suivant :

```
1 | \autocites[on consultera également :]{Saxer1980}{Urner1952}
```

Donne :

Victor SAXER, *Morts, martyrs, reliques en Afrique chrétienne aux premiers siècles : Les témoignages de Tertullien, Cyprien et Augustin à la lumière de l'archéologie africaine*, Paris : Beauchesne, 1980, on consultera également : Hans URNER, *Die ausserbiblische Lesung in Christlichen Gottesdienst*, Gottingen, 1952

## 11.5 Choix de la forme abrégée

Avec le style `verbose`, on peut faire apparaître les références sous une forme abrégée. Il existe pour cela plusieurs champs, que nous n'avons pas encore mentionnés.

<b>shortauthor</b>	Nom abrégé de l'auteur.
<b>shorteditor</b>	Nom abrégé de l'éditeur.
<b>shorthand</b>	Forme abrégée de la référence.
<b>shorthandintro</b>	Lorsqu'une entrée est citée pour la première fois, et si le champ <b>shorthand</b> est utilisé, le champ <b>shorthandintro</b> sert à introduire la forme abrégée. Par exemple, « cité plus tard : ».
<b>shorttitle</b>	Forme abrégée du titre.

La commande `\printshorthands` permet d'imprimer la liste des abréviations.



# Chapitre 12

## Afficher la bibliographie

*Dans ce chapitre nous allons voir comment afficher et trier une bibliographie.*

### 12.1 Affichage de la bibliographie

La commande pour imprimer une bibliographie est `\printbibliography`. Si vous l'essayez dans votre projet, vous obtenez là où vous l'avez placée une bibliographie reprenant l'ensemble des œuvres citées au cours de votre travail. Ainsi, vous êtes certain de ne rien oublier.

Toutefois une telle bibliographie est peu utile pour les longs travaux : elle mélange allègrement sources primaires et études secondaires, ne fait pas de distinctions entre les chapitres, etc. Nous allons donc voir comment sectionner une bibliographie. Mais auparavant il nous faut mentionner une spécificité propre aux sciences humaines, ainsi que la manière d'ajouter des références non citées.

#### 12.1.1 Choix du style de bibliographie

De même qu'il est possible de choisir, lors de l'appel à *biblatex*, les styles des références bibliographiques, il est possible de choisir un style bibliographique grâce à l'argument `\bibstyle`. Pour les sciences humaines, il vaut mieux utiliser `verbose` qui nous assure d'avoir des références complètes, sans pour autant qu'un numéro soit attribué à chaque entrée comme c'est le cas avec `numeric`, le style par défaut. Ainsi, lors de l'appel au package *biblatex*, il suffit d'écrire :

1 | `\usepackage[bibstyle=verbose,...]{biblatex}`

### 12.1.2 Ajout de références non citées

On peut vouloir ajouter dans une bibliographie des références qui n'ont pas été explicitement citées dans le texte. Il faut utiliser : `\nocite{⟨clef1⟩,⟨clef2⟩,...}`.

Pour ajouter l'ensemble des références d'un fichier `.bib` il suffit d'écrire : `\nocite{*}`.

## 12.2 Diviser une bibliographie

### 12.2.1 Par catégorie de document

Une première manière de trier une bibliographie est de le faire selon le type des documents. Les entrées bibliographiques disposent d'un champ `entrysubtype` qui permet de préciser le type d'entrée : par exemple indiquer si une entrée est une lettre, un traité, une prédication, etc. L'option `subtype` passée à la commande `\printbibliography` permet de n'imprimer que les entrées dont le champ `entrysubtype` correspond. Supposons que nous ne souhaitons afficher que les entrées dont ce champ a pour valeur « lettre ». Il suffit d'écrire :

```
1 | \printbibliography[subtype=lettre]
```

On peut passer une option `title` afin de préciser le titre de notre bibliographie :

```
1 | \printbibliography[subtype=lettre,title=Lettres]
```

Une telle méthode permet aisément de séparer une bibliographie en plusieurs parties selon les catégories de documents.

```
1 | \printbibliography[subtype=concile,title=Actes de concile]
2 | \printbibliography[subtype=traite,title=Traités]
3 | \printbibliography[subtype=lettre,title=Lettres]
```

### Changer le niveau de titre d'une bibliographie

Par défaut, les titres de bibliographie sont de niveaux `\chapter*` (☞ p. 29, 3.1). Mais il est possible de redéfinir ces titres, grâce à `\defbibheading`.

Si nous souhaitons par exemple que le titre de la bibliographie soit de niveau `\subsection`, il nous suffit d'ajouter les lignes suivantes au début de notre fichier `.tex`, si possible dans le préambule :

```
1 | \defbibheading{bibliography}{\bibname}{\subsection{#1}}
```

Analysons ce code :

1. Le premier argument correspond au nom de l'en-tête de bibliographie (du titre de bibliographie). L'en-tête nommé « bibliography » est celui utilisé par défaut. Mais on pourrait très bien définir un en-tête « toto », il suffirait alors de passer une option **heading** à la commande `\printbibliography`.

```
1 | \defbibheading{toto}{\bibname}{\subsection{#1}}
2 | \printbibliography[heading=toto]
```

2. Le second argument, ici `\bibname`, correspond au titre par défaut. Dans notre cas, la commande `\bibname` renvoie simplement à la valeur « Bibliographie ».
3. Le troisième argument contient le code qui est appelé pour créer l'en-tête d'une bibliographie. Ici on déclare qu'on passe le titre (le contenu de `#1`) à une commande `\subsection`.

Avec une telle méthode, il est aisé de trier une bibliographie en sources primaires et sources secondaires. Il suffit d'attribuer une valeur au champ **entry-subtype** des entrées bibliographiques, en la faisant varier selon qu'il s'agisse d'une source primaire ou d'une source secondaire. Par exemple :

```
1 | \defbibheading{bibliography}{\bibname}{\subsection{#1}}
2 | \chapter{Bibliographie}
3 | \section{Sources}
4 | \printbibliography[subtype=traite,title=Traités]
5 | \printbibliography[subtype=lettre,title=Lettres]
6 | \section{Littérature secondaire}
7 | \printbibliography[subtype=outil,title=Outils]
8 | \printbibliography[subtype=etude,title=Études]
```

### 12.2.2 Par sujet

Une autre manière de sectionner une bibliographie est de faire des parties thématiques. On peut pour cela indiquer dans le champ **keywords** des mots clés. Les mots clés doivent être séparés par des virgules.

On peut alors passer l'option **keyword** à la commande `\printbibliography`, et ainsi sélectionner toutes les entrées ayant la valeur « xxx » dans leurs champs **keywords** :

```
1 | \printbibliography[keyword=xxx]
```

*A contrario*, on peut afficher toutes les entrées n'ayant pas la valeur « xxx » dans leurs champs **keywords**.

```
1 | \printbibliography[notkeyword=xxx]
```

Il est enfin possible de combiner plusieurs mots clés. Dans ce cas, les entrées possédant au moins un de ces mots apparaissent :

```
1 | \printbibliography[keyword=positivistes, keyword=naturalistes]
```

 Un certain nombre de personnes conseillent d'utiliser le champ `keyword` pour séparer sources primaires et sources secondaires. L'auteur de ces lignes considère qu'il s'agit là d'une déformation du sens de ce champ, qui ne devrait pas porter sur le statut épistémologique d'une référence mais sur son sujet.

### 12.2.3 Par section du document

*Biblatex* permet aussi d'afficher une bibliographie correspondant à une partie du document, par exemple un chapitre, grâce à la définition de segments bibliographiques — un segment bibliographique étant une partie de document qui a sa bibliographie propre<sup>1</sup>.

La méthode la plus simple est de passer une option `refsegment` lors de l'appel au package *biblatex*. La valeur de cet argument peut-être « part », « chapter », « section », « subsection », qui indique à quel niveau de titre commencer une nouvelle section bibliographique<sup>2</sup>.

Chaque segment bibliographique possède un numéro compris entre 0 et N. On peut connaître celui du segment courant grâce à `\therefsegment`.

On peut ainsi afficher à la fin de chaque chapitre une bibliographie du chapitre :

```
1 | \documentclass{...}
2 | ...
3 | \usepackage[refsegment=chapter]{biblatex}
4 | \defbibheading{bibliography}{\bibname}{\section{#1}}
5 | ...
6 | \begin{document}
7 | \chapter{...}
8 | ...
9 | \printbibliography[refsegment=\therefsegment]
```

<sup>1</sup>En réalité *biblatex* propose deux choses différentes : « segment » et « section » bibliographiques. Seules les sections bibliographiques ont réellement une bibliographie propre, dans la mesure où, si on utilise un style numéroté de bibliographie, la numérotation recommence à chaque changement de section, mais pas à chaque changement de segment. Toutefois il est rare en sciences humaines d'utiliser un style bibliographique numéroté. C'est pourquoi nous ne parlons ici que des segments bibliographiques et non pas des sections bibliographiques. Voir : Philipp LEHMAN, « Bibliography Sections », dans : *The biblatex Package*, ainsi que : Philipp LEHMAN, « Bibliography Segments », dans : *The biblatex Package*.

<sup>2</sup>Il est toutefois possible de créer des sections bibliographiques autrement que par niveaux de titre, voir : *idem*, « Bibliography Segments », *op. cit.*

```

10 | \chapter{...}
11 | ...
12 | \printbibliography[refsegment=\therefsegment]
13 | \end{document}

```

### 12.2.4 Autres manières

*Biblatex* propose de nombreuses autres manières de diviser une bibliographie : ainsi il est possible de définir des filtres personnalisés pour ne sélectionner que les entrées répondant à certains critères, d'assigner des « catégories » à certaines entrées, etc. Nous n'avons présenté ici qu'un aperçu des usages les plus courants : nous renvoyons à la documentation de *biblatex* pour plus de détails<sup>3</sup>.

## 12.3 Tri à l'intérieur d'une bibliographie

On constate que la bibliographie est triée par nom d'auteur, puis par titre, puis par année. Il est possible de passer une option `sorting` lors de l'appel au package *biblatex*. Cette option prend pour valeur l'un des schémas de tri proposés par *biblatex* ou bien définis par vos soins<sup>4</sup>.

Par exemple pour trier par nom d'auteur (indiqué par la lettre `n`), année (`y`), titre (`t`) :

```

1 | \usepackage[sorting=nyt]{biblatex}

```

Parfois la valeur utilisée pour le tri ne doit pas être la valeur ensuite affichée dans la bibliographie. Prenons deux entrées dont les titres sont « Lettre 237 » et « Lettre 64 ». Dans la bibliographie finale, « Lettre 64 » est affiché *après* « Lettre 237 ».

En effet, ce n'est pas le nombre « 237 » qui est reconnu, mais seulement le caractère « 2 », puis le caractère « 3 », etc. C'est pourquoi « Lettre 237 » est situé avant « Lettre 64 », 2 se trouvant avant 6.

Pour contourner ce problème, on peut utiliser le champ `sorttitle`. Celui-ci n'est pas affiché mais a la priorité sur le champ `title` en ce qui concerne le tri dans la bibliographie finale. Voici ce que donnent nos deux entrées :

```

1 | @book{AugustinEpi64,
2 |   Author = {Augustin},
3 |   Sorttitle = {Lettre 064},
4 |   Title = {Lettre 64}}

```

<sup>3</sup>Philipp LEHMAN, « Bibliography Commands », dans : *The biblatex Package*.

<sup>4</sup>Pour les schémas de tri standards voir Philipp LEHMAN, « Default Sorting Schemes », dans : *The biblatex Package*; pour les schémas personnalisés, voir Philipp LEHMAN, « Sorting », dans : *The biblatex Package*.

```
5 |
6 | @book{AugustinEpi237,
7 |     Author = {Augustin},
8 |     Sorttitle = {Lettre 237},
9 |     Title = {Lettre 237},
```

Comme « 0 » se situe avant « 2 », la lettre 64 est bien située avant la lettre 237. Dans la bibliographie, l'entrée aura cependant bien le titre « Lettre 64 », et non pas « Lettre 064 ».

⚡ Un problème semblable peut se poser pour les œuvres anonymes : en l'absence d'auteur, *bibtex* se sert du titre comme premier critère de tri, ce qui a pour conséquence de disperser toutes les œuvres anonymes au sein de la bibliographie, plutôt que de les mettre à un seul endroit, par exemple tout au début.

On peut certes utiliser le champ `sortname` pour résoudre ce problème, mais le plus sain est de définir sa propre règle de tri : nous expliquons comment sur notre blog<sup>5</sup>.

Il existe également un champ `sortyear` pour préciser une autre année de tri.

---

<sup>5</sup>Maïeul ROUQUETTE, *Tri des œuvres anonymes*, URL : <http://geekographie.maieul.net/92>.

# Chapitre 13

## Entrées hiérarchisées

*Dans ce chapitre, nous allons voir comment créer des entrées hiérarchisées, pour organiser mieux organiser sa base de données et pour gérer les divisions de source.*

### 13.1 Principe des sous-entrées bibliographiques

Supposons que vous citiez diverses contributions d'un même colloque. Classiquement vous créez plusieurs entrées bibliographiques sous la forme<sup>1</sup> :

```
1 @inproceedings{clef1,
2     Author = {Anne Onyme},
3     Booktitle = {La sexualité d'Adam à Homo Technologicus},
4     Editor = {Gérard Menvu},
5     Title = {L'influence de LaTeX sur la reproduction des êtres
6     humains},
7     Year = {2011}}
8
9 @inproceedings{clef2,
10     Author = {Jean Séniste},
11     Booktitle = {La sexualité d'Adam à Homo Technologicus},
12     Editor = {Gérard Menvu},
13     Title = {L'arbre du paradis était-il un pêcher?},
14     Year = {2011}}
```

Vous dupliquez ainsi les données sur le colloque (titre, éditeur, etc.). Si par hasard vous vous êtes trompés dans une information, vous devrez la modifier partout. Pour éviter cela Biber prévoit la possibilité de créer des filiations d'entrées : des entrées secondaires « héritant » d'une entrée principale. L'héritage

---

<sup>1</sup>Toute ressemblance avec un colloque s'étant déjà tenu ne saurait être que pure coïncidence.

étant pris en charge lors de l'interprétation du fichier. Pour ce faire, il suffit d'indiquer dans le champ `crossref` des entrées secondaires la clef de l'entrée principale.

```

1  @proceedings{principale,
2      Editor = {Gérard Menvu},
3      Title = {La sexualité d'Adam à Homo Technologicus},
4      Year = {2011}
5  }
6
7  @inproceedings{clef1,
8      Author = {Anne Onyme},
9      Title = {L'influence de LaTeX sur la reproduction des êtres
10     humains},
11     crossref={principale}
12     }
13
14 @inproceedings{clef2,
15     Author = {Jean Séniste},
16     Title = {L'arbre du paradis était-il un pêcher?},
17     crossref={principale}
18     }

```

Après compilation, nous voyons que nos sous-entrées ont correctement hérité des champs de l'entrée principale :

Anne ONYME, « L'influence de LaTeX sur la reproduction des êtres humains », dans : *La sexualité d'Adam à Homo Technologicus*, dir. Gérard MENVU, 2011

Jean SÉNISTE, « L'arbre du paradis était-il un pêcher ? », dans : *La sexualité d'Adam à Homo Technologicus*, dir. Gérard MENVU, 2011

Vous constaterez que Biber procède à un héritage « intelligent » : le champ `date` de l'entrée mère est repris comme champ `date` de l'entrée fille, cependant que le champ `title` devient le champ `booktitle`.

 Il est possible de créer autant de niveaux d'entrée bibliographique que souhaités. Attention cependant à ne pas produire des références circulaires (A héritant de B, héritant de C, héritant de A). Par ailleurs, certains logiciels de gestion bibliographique vous interdisent de créer des sous-sous-entrées bibliographiques : il vous faut alors modifier vos entrées avec votre éditeur de texte.

## 13.2 Sous-entrées dans la bibliographie finale

Lorsqu'une entrée fille est citée, Biber rajoute systématiquement l'entrée mère dans la bibliographie finale. On peut toutefois indiquer un nombre minimal de citations d'entrées filles pour que se fasse l'ajout de l'entrée mère dans la bibliographie. Si nous souhaitons qu'une entrée mère ne soit ajoutée à la bibliographie que si les entrées filles sont citées au moins 10 fois (toutes entrées confondues) :

```
1 | \usepackage[mincrossrefs=10,...]{biblatex}
```

## 13.3 Préciser les héritages de champs

Biber possède un certain nombre de réglages d'héritage de champs<sup>2</sup>. On peut cependant modifier ces héritages, ou en ajouter d'autres. Les champs **entrysubtype**, par exemple, ne sont par défaut pas hérités. Pour modifier les réglages standards, il suffit d'utiliser, dans le préambule : `\DeclareDataInheritance{⟨entrée mère⟩}{⟨entrée fille⟩}{⟨règles⟩}`.

⟨*entrée mère*⟩ désigne le type d'entrée mère : @book, @proceedings, etc. Le symbole \* désigne n'importe quel type d'entrée.

⟨*entrée fille*⟩ désigne le type d'entrée fille. Là aussi le symbole \* désigne n'importe quel type d'entrée.

⟨*règles*⟩ désigne un certain nombre de règles d'héritages, sous forme de commandes.

Les types d'entrées à préciser dans les arguments ⟨*entrée fille*⟩ et ⟨*entrée mère*⟩ ne doivent pas être précédés du signe @.

La première commande d'héritage est la suivante `\noinherit{⟨champ⟩}`. Elle empêche l'héritage d'un champ par une entrée fille.

La seconde est `\inherit[⟨option⟩]{⟨champ source⟩}{⟨champ cible⟩}`. Il n'y qu'une seule option possible : `override=true`. Si cette option est passée, alors la valeur du champ de l'entrée fille est écrasée par celles du champ de l'entrée mère. Sinon, la valeur du champ de l'entrée fille a priorité.

Ainsi, pour permettre l'héritage du champ **entrysubtype** il suffit d'écrire :

```
1 | \DeclareDataInheritance{*}{*}{
2 |     \inherit{entrysubtype}{entrysubtype}
3 | }
```

<sup>2</sup>Philipp LEHMAN, « Default crossref Setup », dans : *The biblatex Package*.

## 13.4 Divisions de source

Les sous-entrées peuvent être utilisées pour contourner une des limites de la gestion bibliographique de L<sup>A</sup>T<sub>E</sub>X : l'impossibilité de gérer à la fois la division d'une source et la pagination. Les textes anciens disposent en général d'une division propre qui permet – en théorie – de repérer un passage dans n'importe quelle édition. Par exemple :

AUGUSTIN, *De doctrina christiana*, II, VIII, 12-13

signifie que nous renvoyons au *De Doctrina Christiana* d'Augustin, livre II, chapitre VIII, paragraphes 12-13. En théorie on peut retrouver le passage concerné dans n'importe quelle édition du traité. Ceci est appelé généralement « division de source ». En revanche, supposons que nous renvoyions à une édition particulière, par exemple celle de la Bibliothèque Augustinienne : on aimerait indiquer les pages précises.

AUGUSTIN, *De doctrina christiana*, II, VIII, 12-13, éd. établie et trad. par Madeleine MOREAU, Bibliothèque Augustinienne, p. 150–155

S'il est aisé lorsqu'on utilise une commande de citation d'indiquer une page précise (☞ p. 96, 11.3.2), il n'est pas possible en revanche d'indiquer une division de source précise. La solution est simple : il suffit de créer une entrée principale correspondante à l'ensemble de l'œuvre, et des entrées secondaires correspondantes à des parties de l'œuvre. Ces entrées secondaires héritent des valeurs de l'entrée principale, sauf pour le champ **pages** et le champ **titleaddon** dans lequel on précise la division de source.

☞ Il n'existe pas de champ spécifiquement prévu pour indiquer la division de source. Pourquoi choisir ce champ **titleaddon** ? Deux raisons poussent l'auteur de ces lignes à adopter cette solution :

1. D'un point de vue sémantique, la division de source peut être considérée comme un ajout au titre.
2. D'un point de vue pratique, ce champ a pour avantage d'être, par défaut, situé par *biblatex* entre le titre et les informations d'édition, sans pour autant être en italique.

☞ Le choix d'une entrée de type **@book** pour la clef `DeDoctChrIIviii18-20` à la place d'une entrée **@inbook** se justifie par des questions d'héritage. On ne donne pas en effet à cette entrée de titre propre. Dès lors, si elle était de type **@inbook**, Biber récupérerait le champ **title** de l'entrée mère et en ferait un **booktitle**, ce qui n'est pas ce que nous souhaitons. En revanche,

en faisant de notre division de source une entrée de type `@book`, on lui permet d'hériter dans son champ `title` du champ `title` de l'entrée mère.

Évidemment le choix de ce champ devrait, idéalement, être discuté en accord avec l'éditeur. De toute façon, il est facile, si le choix ne convient pas, de changer en série les entrées.

Il est possible de vérifier automatiquement que les entrées soient du même type : nous en parlons sur notre blog<sup>3</sup>.

Ainsi pour notre cas, il suffit de créer les deux entrées suivantes :

```

1  @book{DeDoctChr,
2      Author = {Augustin},
3      Editor = {Madeleine Moreau},
4      Entrysubtype = {traite},
5      Series = {Bibliothèque Augustinienne},
6      Title = {De doctrina christiana},
7      Translator = {Madeleine Moreau}}
8
9
10 @book{DeDoctChrIviii18-20,
11     Crossref = {DeDoctChr},
12     Pages = {150-155},
13     Titleaddon = {II, \textsc{viii}, 12-13},
14     }

```

Pour citer notre passage précis, il faut écrire :

```

1  \autocite{DeDoctChrIviii18-20}

```

👉 L'inconvénient d'une telle méthode est qu'elle multiplie les entrées dans la bibliographie finale, puisque celle-ci contiendra à la fois l'œuvre entière et chacune des divisions que nous ajoutons dans notre fichier `.bib`. Nous expliquons sur notre blog comment empêcher cela<sup>4</sup>.

De même cette méthode pose quelques problèmes pour la gestion des *op. cit.* Nous expliquons sur notre blog comment les résoudre<sup>5</sup>.

Pour comprendre ces deux explications il est nécessaire d'avoir lu les chapitres qui vont suivre.

<sup>3</sup>Maïeul ROUQUETTE, *Vérification des filiations avec biblatex (1) : vérifier les types*, URL : <http://geekographie.maieul.net/74>.

<sup>4</sup>Maïeul ROUQUETTE, *Ne pas afficher les sous-entrées bibliographiques*, URL : <http://geekographie.maieul.net/60>.

<sup>5</sup>Maïeul ROUQUETTE, *Division de sources et abréviations*, URL : <http://geekographie.maieul.net/>



# Chapitre 14

## Modifier les styles bibliographiques (1)

*Vous savez désormais utiliser les commandes de citation. Vous pourriez maintenant souhaiter modifier la manière dont les références s'affichent : changer l'ordre des champs, les séparateurs, les textes du type « dans ».*

*Ces deux chapitres vous indiqueront les principes généraux pour faire ces modifications. Ils n'ont pas pour vocation de tout présenter sur le sujet, mais simplement les principes de base. On se reportera au manuel de biblatex pour aller plus loin.*

*Le premier de ces chapitres vous présentera les techniques les plus simples, celles qui ne nécessitent pas d'aller fouiller dans les styles bibliographiques standards.*

### 14.1 Séparateurs d'unité de sens

Prenons un problème courant. Vous avez constaté que, par défaut, les différents champs d'une référence bibliographique<sup>1</sup> sont séparés par des points. On obtient par exemple :

Hans URNER. *Die ausserbiblische Lesung in Christlichen Gottesdienst*. Gottingen, 1952

alors que le règles françaises voudraient que l'on ait :

Hans URNER, *Die ausserbiblische Lesung in Christlichen Gottesdienst*, Gottingen, 1952

---

<sup>1</sup>À l'exception de la date et de quelques autres champs.

À l'intérieur d'une référence bibliographique, *bibtex* distingue des unités de sens<sup>2</sup>. Ici : l'auteur, le titre, l'adresse, l'éditeur, l'année.

Ces unités sont séparées par des signes de ponctuation. Les séparateurs sont définis par la commande `\newunitpunct`.

Il suffit donc de redéfinir la commande en mettant dans le préambule<sup>3</sup> :

```
1 | \renewcommand{\newunitpunct}[0]{, }
```



Nous n'avons pas parlé jusqu'à présent de la commande `\renewcommand`. Celle-ci est identique à la commande `\newcommand` à la différence qu'elle permet de redéfinir une commande déjà existante.

Cette solution semble fonctionner. Mais elle pose un problème : les styles définissent des unités de sens sans se poser la question de savoir si ces unités sont vides ou pas. Prenons l'entrée suivante :

```
1 | @bookinbook{BrefHippone,  
2 |   Booktitle = {Concilia Africae},  
3 |   Entrysubtype = {concile},  
4 |   Number = {149},  
5 |   Pages = {36},  
6 |   Pagination = {paragraph},  
7 |   Series = {Corpus Christianorum Series Latina},  
8 |   Title = {Breviarum Hipponense}}
```

On obtient le résultat suivant :

*Breviarum Hipponense*, dans : , *Concilia Africae*, Corpus Christianorum Series Latina 149, par. 36

La virgule entre les deux points et *Concilia Africae* s'explique parce qu'une unité a été créée, unité qui doit contenir normalement la valeur du champ `bookauthor`.

Pour éviter ce problème, il faut ne pas mettre directement dans la définition de la commande `\newunitpunct` les signes de ponctuation mais utiliser à la place les commandes de ponctuation<sup>4</sup>. Ces commandes affichent en effet la ponctuation uniquement si l'unité qui précède a bien un contenu<sup>5</sup>. Ainsi il faut mettre dans notre cas :

```
1 | \renewcommand{\newunitpunct}[0]{\addcomma\addspace}
```

<sup>2</sup>À vrai dire il existe des unités plus vastes, appelés blocs. Mais ces unités servent pour des réglages avancés de l'affichage de la bibliographie finale.

<sup>3</sup>Cela n'est pas strictement nécessaire, mais c'est une bonne méthode que de mettre l'ensemble des commandes de personnalisation de style dans le préambule. Nous conseillons de les regrouper en un seul fichier, appelé dans le préambule.

<sup>4</sup>Listées dans Philipp LEHMAN, « Punctuation and Spacing », dans : *The biblatex Package*.

<sup>5</sup>Par ailleurs, elles suppriment les espaces multiples qui précèdent.

### 14.1.1 Un séparateur spécifique à deux unités

On peut vouloir qu’entre deux unités spécifiques, par exemple entre le titre et le sous-titre, le séparateur ne soit pas le même qu’entre les autres unités. Vous avez sans doute constaté que c’est d’ailleurs le cas pour le style « verbose » : l’adresse et l’éditeur sont séparées par un deux-points.

Deux cas existent :

1. Le package *biblatex* a prévu une commande spécifique pour notre type d’unité. C’est le cas pour le séparateur avant le sous-titre<sup>6</sup>. Il suffit dans ce cas de redéfinir la commande. Nous souhaitons par exemple avoir un deux-points entre le titre et le sous-titre. Il nous suffit d’indiquer :

```
1 | \renewcommand{\subtitlepunct}[0]{\addspace\addcolon\addspace}
```

2. Le package n’a pas prévu de commande spécifique pour notre type d’unité. Dans ce cas, le problème est plus complexe et nécessite des notions que nous n’avons pas encore abordées. C’est pourquoi nous renvoyons au chapitre suivant (☞ p. 126, 15.2).

## 14.2 Séparateur de références multiples

Lorsque nous utilisons une commande de citation multiple (☞ p. 100, 11.4), les séparateurs par défaut sont des points-virgules :

```
1 | \autocites{Saxer1980}{Junod1992}
```

Victor SAXER, *Morts, martyrs, reliques en Afrique chrétienne aux premiers siècles : Les témoignages de Tertullien, Cyprien et Augustin à la lumière de l’archéologie africaine*, Paris : Beauchesne, 1980 ; Eric JUNOD, « “Apocryphes du Nouveau Testament” : une appellation erronée et une collection artificielle », dans : *Apocrypha* 3 (1992), p. 17–46

On peut là aussi modifier le séparateur en redéfinissant `\multicitedelim`. Par exemple pour avoir une virgule :

```
1 | \renewcommand{\multicitedelim}[0]{\addcomma\addspace}
```

<sup>6</sup>Ces commandes sont peu nombreuses : on les trouvera dans Philipp LEHMAN, « Generic Commands and Hooks », dans : *The biblatex Package*.

## 14.3 Styles de certains blocs

La ponctuation n'est pas le seul cas pour lequel *biblatex* propose des outils pour faire des modifications spécifiques. Certaines parties des références bibliographiques ont également la possibilité d'être configurées grâce à des commandes particulières<sup>7</sup>. Prenons par exemple la particule d'un auteur.

Simone de BEAUVOIR, *Le Deuxième Sexe*

Il est parfois d'usage de mettre la particule entre parenthèses. Il faut pour cela redéfinir la commande `\mkbibnameprefix`. Cette commande prend un argument, qui est la particule. Pour la mettre automatiquement entre parenthèses, il suffit ainsi d'écrire.

```
1 | \renewcommand{\mkbibnameprefix}[1]{(#1)}
```

Voire pour être encore plus correct :

```
1 | \renewcommand{\mkbibnameprefix}[1]{\parentext{#1}}
```

La commande `\parentext` met entre parenthèses un texte passé en argument. On obtient le résultat souhaité :

Simone (de) BEAUVOIR, *Le Deuxième Sexe*

## 14.4 Chaînes de langue

Les chaînes de langue sont ces morceaux de texte figés mais qui dépendent de la langue utilisée. Prenons une entrée de type `@article`. Lorsque nous la citons, nous obtenons :

Eric JUNOD, « "Apocryphes du Nouveau Testament" : une appellation erronée et une collection artificielle », dans : *Apo-crypha* 3 (1992), p. 17–46

La chaîne « dans » sépare le titre de l'article et le nom de la revue. Si nous rédigeons en anglais, nous aurions « in » à la place.

Il peut nous arriver de vouloir changer ces chaînes de langues, par exemple justement pour avoir la forme « in », terme latin souvent utilisé dans les bibliographies. Une chaîne de langue associée à une clef invariable a une valeur qui change selon la langue.

<sup>7</sup>LEHMAN, « Generic Commands and Hooks », *op. cit.*

Les styles bibliographiques possèdent des chaînes de langue par défaut, qu'il est possible de modifier au cas par cas. Pour cela, il nous faut avant tout connaître les clefs, en ouvrant le fichier de langue french.lbx. Pour repérer ce fichier, voir notre explication en annexe (☞ p. 233, C.4).

Une fois ce fichier ouvert, repérons la ligne qui commence par :

```
\DeclareBibliographyStrings
```

Cette ligne est suivie d'autres, qui ont la forme :

```
1 | clef = {{chaîne longue}{chaîne courte}}
```



Chaque chaîne de langue est disponible en version longue ou courte. On peut passer une option lors de l'appel au package *biblatex* pour décider d'utiliser les versions longues.

```
1 | \usepackage[abbreviate=false]{biblatex}
```

Toutefois quand on redéfinit une chaîne de langue, on ne lui attribue qu'une seule version.

Il nous faut donc repérer la clef correspondante à notre chaîne « dans ». Très rapidement nous découvrons qu'il s'agit de la chaîne *in*.

Une fois notre clef repérée, il est facile de lui attribuer une nouvelle valeur, grâce à la commande `\DefineBibliographyStrings{⟨langue⟩}{⟨chaîne⟩}`, qu'il faut placer dans le préambule du fichier.

⚡ Il ne faut surtout pas modifier les fichiers standards : on perdrait les modifications à chaque mise à jour.

```
1 | \DefineBibliographyStrings{french}{%
2 |   in = {\emph{in}}%
3 | }
```

On peut utiliser dans une chaîne de langue des commandes  $\LaTeX$ . Si notre chaîne contient des signes de ponctuation, il faut utiliser les commandes de ponctuation mentionnées plus haut. Les % à la fin de chaque ligne sont là pour éviter les espaces intempestifs (☞ p. 55, 7.1.1).

Si nous souhaitons redéfinir plusieurs chaînes de langue, il nous faut séparer chaque définition par une virgule :

```
1 | \DefineBibliographyStrings{french}{%
2 |   clef1 = {valeur1},%
3 |   clef2 = {valeur2},%
4 |   clefN = {valeurN}%
5 | }
```

☛ On pourrait redéfinir les chaînes de langue consacrées aux abréviations universitaires de type *op. cit.* pour les avoir en italique.

Il y a une solution plus simple. En effet, *biblatex* passe ces chaînes de langue dans une commande `\mkibidem{⟨chaîne⟩}` avant de les afficher.

Il nous suffit donc de redéfinir cette commande :

```
1 | \renewcommand{\mkibid}[1]{\emph{#1}}
```

### 14.4.1 Introduction aux tests conditionnels

Nous allons maintenant prendre un peu d'anticipation sur le prochain chapitre. Vous savez que les styles de la famille « verbose-trad » utilisent des abréviations latines comme *op. cit.* Ces abréviations sont les abréviations universitaires courantes. Toutefois *biblatex* utilise indistinctement *op. cit.* pour toutes les sortes d'entrées, y compris pour les entrées de type @article, alors qu'on pourrait s'attendre à une abréviation comme *art. cit.*

Pour résoudre ce problème il suffit de redéfinir la chaîne de langue qui définit *op. cit.*, en lui attribuant une valeur différente selon le type d'entrée. On va donc utiliser l'un des nombreux tests conditionnels disponibles dans *biblatex*. Il s'agit de la commande : `\iffieldequalstr{⟨champ⟩}{⟨texte⟩}{⟨sioui⟩}{⟨sinon⟩}`.

Cette commande ne peut s'utiliser que dans certaines parties d'un fichier  $\TeX$  : dans les commandes de définition de styles bibliographiques et dans celles de définition d'un chaîne de langue. Elle vérifie que le champ `⟨champ⟩` de l'entrée bibliographique citée correspond à l'argument `⟨texte⟩` : si c'est le cas, elle retourne l'argument `⟨sioui⟩`, et sinon elle retourne l'argument `⟨sinon⟩`.

Ici nous testons le pseudo-champ<sup>8</sup> `entrytype`, qui correspond au type d'entrée sans le @. Nous vérifions s'il s'agit d'une entrée de type article : si c'est le cas nous donnons à la chaîne de langue la valeur « art. cit. », dans le cas contraire nous donnons la valeur « op. cit. ». Au final cela donne :

```
1 | \DefineBibliographyStrings{french}{%
2 |     opcit = \iffieldequalstr{entrytype}{article}%
3 |     {art\adddotspace cit\adddot}%
4 |     {op\adddotspace cit\adddot}%
5 | }
```

On remarquera l'utilisation des commandes de ponctuation (☞ p. 116, 14.1), comme signalé plus haut.

<sup>8</sup> « Pseudo » car il ne se situe pas réellement comme un champ dans l'entrée bibliographique.

# Chapitre 15

## Modifier les styles bibliographiques (2)

*Nous avons vu que BibLaTeX propose un certain nombre de commandes pour personnaliser rapidement les styles bibliographiques. Toutefois ces commandes ne suffisent pas pour des personnalisation avancées. La possibilité de choisir l'ordre d'affichage des champs, par exemple, nécessite ainsi d'aller plus loin dans la compréhension des styles bibliographiques de biblatex.*

### 15.1 Que se passe-t-il lorsqu'on utilise une commande `\langle prefix \rangle cite` ?

Pour comprendre comment personnaliser l'affichage bibliographique, il est nécessaire de connaître sommairement ce qui se passe lorsqu'on utilise une commande de citation.

Lorsqu'on appelle une commande de citation, du type `\langle prefix \rangle cite`, celle-ci va appeler :

- des macros bibliographiques, chargées d'afficher l'argument `\langle prenote \rangle` ou `\langle postnote \rangle`, ou encore de gérer les citations répétées. Les macros bibliographiques sont des types particuliers de commandes, propres au package *biblatex* ;
- un driver<sup>1</sup> bibliographique. Un driver correspond à un type d'entrée (`@article`, `@book`, etc.), et se charge d'afficher les champs de l'entrée dans le bon ordre. Pour cela, il appelle :

---

<sup>1</sup>Bien que le terme « driver » ne soit pas français, nous l'utilisons car on le trouve dans les commandes internes de BibLaTeX

- des commandes de séparation d'unités bibliographiques (☞ p. 115, 14.1) , que nous avons vues plus haut ;
- des macros bibliographiques. Ces macros bibliographiques appellent elles-mêmes :
  - \* des commandes d'impression de champs bibliographiques ;
  - \* éventuellement d'autres macros bibliographiques ;
  - \* des chaînes de langues.

Ceci peut se résumer par le schéma 15.1 (p. 123).

## 15.2 Redéfinir une macro bibliographique : exemple des champs auteur et éditeur

L'ensemble de ces éléments sont entièrement redéfinissables. Nous allons prendre un exemple concret de problématique existante.

Preons l'entrée suivante :

```

1 | @book{Saxer1980,
2 |     Author = {Victor Saxer},
3 |     Entrysubtype = {etude},
4 |     Location = {Paris},
5 |     Publisher = {Beauchesne},
6 |     Subtitle = {Les témoignages de Tertullien, Cyprien
7 |     et Augustin à la lumière de
8 |     l'archéologie africaine},
9 |     Title = {Morts, martyrs, reliques en Afrique chrétienne
10 |     aux premiers siècles},
11 |     Year = {1980}}
```

Elle s'affiche ainsi, avec l'éditeur commercial après l'adresse :

VICTOR SAXER, *Morts, martyrs, reliques en Afrique chrétienne aux premiers siècles : Les témoignages de Tertullien, Cyprien et Augustin à la lumière de l'archéologie africaine*, Paris : Beauchesne, 1980

On pourrait souhaiter avoir l'éditeur avant l'adresse, comme ceci :

VICTOR SAXER, *Morts, martyrs, reliques en Afrique chrétienne aux premiers siècles : Les témoignages de Tertullien, Cyprien et Augustin à la lumière de l'archéologie africaine*, Beauchesne, Paris, 1980

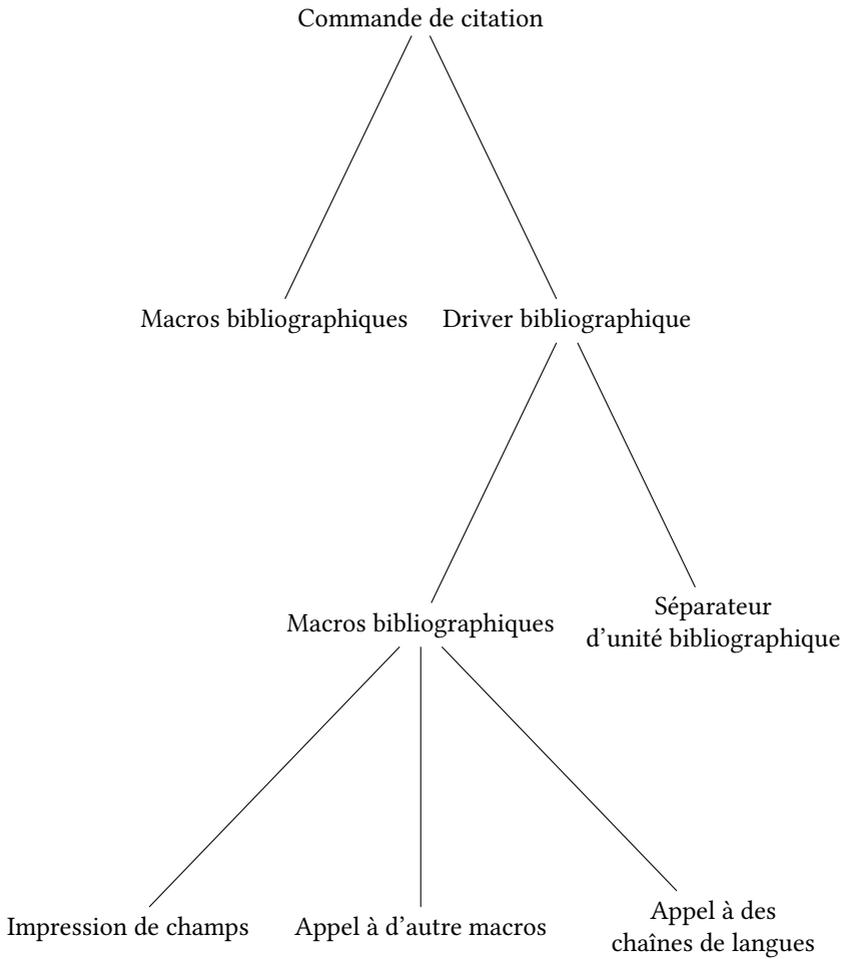


FIG. 15.1 : Le fonctionnement des styles bibliographiques

La première chose à faire va donc être de repérer quelle macro bibliographique modifier. Pour cela, il faut trouver les fichiers (☞ p. 233, C.4) de définition des styles bibliographiques. Il en existe plusieurs :

- un fichier `.def` qui définit les styles invariants, quel que soit le style de bibliographie ou de citation choisi ;
- des fichiers `.cbx` qui définissent les styles utilisés lors de l'utilisation des commandes `\(prefix)cite` ;
- des fichiers `.bbx` qui définissent les styles utilisés lors de l'appel à la commande `\printbibliography`.

Certains fichiers s'appellent mutuellement : par exemple les fichiers `.bbx` contiennent les drivers bibliographiques. Ils sont donc appelés par les fichiers `.cbx`. Ces appels mutuels entre fichiers permettent de garantir une uniformité entre les styles bibliographiques lors de l'utilisation de `\(prefix)cite` et lors de l'utilisation de `\printbibliography`.

Nous supposons que vous utilisez les styles de la famille « verbose ». En ouvrant les fichiers standards (☞ p. 233, C.4), vous pouvez aisément remonter au fichier `standard.bbx`, qui contient les drivers bibliographiques de cette famille. Vous pouvez repérer dedans les lignes suivantes<sup>2</sup> :

```

1  \DeclareBibliographyDriver{book}{%
2  \usebibmacro{bibindex}%
3  \usebibmacro{begentry}%
4  \usebibmacro{author/editor+others/translator+others}%
5  \setunit{\labelnamepunct}\newblock
6  \usebibmacro{maintitle+title}%
7  \newunit
8  \printlist{language}%
9  \newunit\newblock
10 \usebibmacro{byauthor}%
11 \newunit\newblock
12 \usebibmacro{byeditor+others}%
13 \newunit\newblock
14 \printfield{edition}%
15 \newunit
16 \iffieldundef{maintitle}
17   {\printfield{volume}%
18    \printfield{part}}
19   {}%
20 \newunit

```

---

<sup>2</sup>L. 58 au 9 octobre 2011.

```

21  \printfield{volumes}%
22  \newunit\newblock
23  \usebibmacro{series+number}%
24  \newunit\newblock
25  \printfield{note}%
26  \newunit\newblock
27  \usebibmacro{publisher+location+date}%
28  \newunit\newblock
29  \usebibmacro{chapter+pages}%
30  \newunit
31  \printfield{pagetotal}%
32  \newunit\newblock
33  \iftoggle{bbx :isbn}
34    {\printfield{isbn}}
35    {}%
36  \newunit\newblock
37  \usebibmacro{doi+eprint+url}%
38  \newunit\newblock
39  \usebibmacro{addendum+pubstate}%
40  \setunit{\bibpagerefpunct}\newblock
41  \usebibmacro{pageref}%
42  \usebibmacro{finentry}

```

Il s'agit d'un driver bibliographique expliquant comment afficher les entrées de type @book. Il fait appel à des macros bibliographiques *via* les commandes `\usebibmacro`. Ces macros sont communes à plusieurs drivers, ce qui permet d'avoir une certaine uniformité de style, afin par exemple que les noms d'auteurs s'affichent systématiquement de la même façon.

Dans le lot des macros appelées, il en existe un qui nous intéresse en particulier, l'appel à la macro `publisher+location+date` via :

```

27  \usebibmacro{publisher+location+date}

```

En fouillant un peu le même fichier, on repère l'endroit où la macro est définie :

```

1  \newbibmacro*{publisher+location+date}{%
2  \printlist{location}%
3  \iflistundef{publisher}
4    {\setunit*{\addcomma\space}}
5    {\setunit*{\addcolon\space}}%
6  \printlist{publisher}%
7  \setunit*{\addcomma\space}%

```

```

8 | \usebibmacro{date}%
9 | \newunit}

```

Nous allons commenter succinctement ces lignes, avant d'expliquer comment faire pour inverser l'ordre des deux champs.

- Ligne 1** La commande `\newbibmacro*` indique que l'on déclare une nouvelle macro bibliographique, ici **publisher+location+date**. Pour indiquer qu'on redéfinit une macro déjà existante, il faut utiliser dans le préambule<sup>3</sup> la commande `\renewbibmacro*`. La définition de la macro se trouve dans les accolades qui suivent.
- Ligne 2** La commande `\printlist` indique que l'on affiche un champ qui pourrait se présenter sous forme de liste, c'est à dire où le mot clef « and » a un sens. Ici il s'agit du champ **location**.
- Ligne 3** La commande `\iflistundef` teste un champ qui pourrait être une liste, ici le champ **publisher**. Si ce champ est vide, il exécute le contenu de la première accolade (ligne 4), sinon celui de la seconde (ligne 5).
- Ligne 4** Si donc le champ **publisher** est vide, on crée une nouvelle unité bibliographique (☞ p. 115, 14.1), via `\setunit*`, séparée de la précédente par une virgule à laquelle s'ajoute une espace (`\addcomma\space` ☞ p. 116, 14.1).
- Ligne 5** Si le champ **publisher** n'est pas vide, alors on crée une nouvelle unité bibliographique, séparée de la suivante par un deux-points suivi d'une espace (`\addcolon\space`).
- Ligne 6** On imprime le champ **publisher**.
- Ligne 7** On crée une nouvelle unité bibliographique, séparée de la précédente par une virgule suivie d'une espace. À noter que le champ **publisher** étant vide, on n'a qu'une seule virgule : nous renvoyons à nos explication antérieure sur les commandes de ponctuation (☞ p. 116, 14.1).
- Ligne 8** On appelle une macro qui se charge de l'affichage de la date.
- Ligne 9** On crée une nouvelle unité bibliographique. Le signe séparateur est défini par la commande `\newunitpunct` (☞ p. 116, 14.1), vue plus haut.

---

<sup>3</sup>Où ailleurs dans le fichier `.tex` mais en tout cas pas dans les fichiers standards.

Pour inverser l'ordre de nos champs, il suffit donc de redéfinir la macro en inversant l'ordre d'impression des champs. Au passage, on ne veut plus des deux points comme séparateurs, ce qui nous permet de supprimer un test conditionnel.

```

1  \renewbibmacro*{publisher+location+date}{%
2  \printlist{publisher}%
3  \setunit*{\addcomma\space}%
4  \printlist{location}%
5  \setunit*{\addcomma\space}%
6  \usebibmacro{date}%
7  \newunit}

```

Prêtez bien attention aux % de fin de lignes : les oublier signifie risquer d'avoir des espaces indésirables dans ses références bibliographiques.

☞ Il existe d'autres commandes que `\printlist` pour afficher des champs : `\printname` pour imprimer un champ contenant des noms de personne, et `\printfield` pour imprimer un champ ne nécessitant pas de mise en forme particulière.

Pour mieux comprendre quand utiliser l'une ou l'autre de ces commandes, le mieux est de regarder les fichiers standards.

☞ Si vous utilisez le champ **address** à la place du champ **location**, sachez que le premier est considéré comme un alias du second : autrement dit, utiliser **address** revient à utiliser **location**.

On peut déclarer des nouveaux alias de champ via la commande :

```
\DeclareFieldAlias{alias}{original}.
```

## 15.3 Autres exemples : des véritables *op. cit.*

Un des éléments gênants des styles bibliographiques standards de la famille *verbose* est leur manière de gérer les abréviations universitaires de type *op. cit.* En effet, les styles indiquent les *op. cit.* après avoir affiché l'auteur et le titre. Par exemple :

Hans URNER, *Die ausserbiblische Lesung in Christlichen Gottesdienst*, Gottingen, 1952

Victor SAXER, *Morts, martyrs, reliques en Afrique chrétienne aux premiers siècles : Les témoignages de Tertullien, Cyprien et Augustin à la lumière de l'archéologie africaine*, Paris : Beauchesne, 1980

URNER, *Die ausserbiblische Lesung in Christlichen Gottesdienst*, *op. cit.*

SAXER, *Morts, martyrs, reliques en Afrique chrétienne aux premiers siècles*, *op. cit.*

Si nous n'avons qu'une seule entrée dont Victor Saxer est l'auteur, cela est assez inutile. On pourrait avoir une version abrégée sous la forme :

Hans URNER, *Die ausserbiblische Lesung in Christlichen Gottesdienst*, Gottingen, 1952

Victor SAXER, *Morts, martyrs, reliques en Afrique chrétienne aux premiers siècles : Les témoignages de Tertullien, Cyprien et Augustin à la lumière de l'archéologie africaine*, Paris : Beauchesne, 1980

URNER, *op. cit.*

SAXER, *op. cit.*

Nous allons pour cela modifier les styles de *biblatex*, en utilisant la commande : `\ifsingletitle{⟨sioui⟩}{⟨sinon⟩}`.

Cette commande vérifie si une entrée est la seule attribuée à son auteur, et renvoie `⟨sioui⟩` si c'est le cas, `⟨sinon⟩` dans le cas contraire.

Pour la faire fonctionner, il faut passer l'option `singletitle=true` au chargement de *biblatex*.

```
1 | \usepackage[singletitle=true,...]{biblatex}
```

Une fois ceci fait, il est nécessaire de savoir où appliquer cette commande. Commençons par fouiller le fichier `.cbx`, puisqu'il s'agit d'un style pour une commande `\⟨prefix⟩cite`. Recherchons l'expression « `opcit` » qui correspond à la chaîne de langue (☞ p. 120, 14.4.1) qui renvoie « `op. cit.` ».

On la trouve rapidement dans une macro qui s'appelle `cite:title`.

```
1 | \newbibmacro*{cite:title}{%
2 |   \printtext[bibhyperlink]{%
3 |     \printfield[citetitle]{labeltitle}%
4 |     \setunit{\nametitledelim}%
5 |     \bibstring[\mkibid]{opcit}}}
```

Procédons à l'analyse :

- Ligne 1** Le nom de la macro est `cite:title`.
- Ligne 2** La commande `\printtext` sert à deux chose : à mettre directement un texte en s'assurant que *biblatex* gère la ponctuation ou bien à assembler plusieurs champs dans un seul bloc typographique. Ici, nous avons affaire au second usage : `bibhyperlink` signifie que *biblatex* va s'occuper de mettre un lien hypertexte à l'intérieur du document PDF.
- Ligne 3** La commande `\printfield` imprime un champ. Ici le pseudo-champ `labeltitle` : celui-ci renvoie la valeur du champ `shorttitle` s'il est défini, sinon celle de `title` (☞ p. 101, 11.5). Il l'affiche selon le format `citetitle`<sup>4</sup>.
- Ligne 4** Nouvelle unité bibliographique, dont le séparateur est défini par la commande `\nametitledelim`.
- Ligne 5** La commande `\bibstring` sert à appeler une chaîne de langue, ici `opcit`. Le premier argument, dont la valeur est ici `\mkibid`, indique que la chaîne de langue est passée à `\mkibid` avant d'être affichée. Cette commande qu'on peut redéfinir se charge de la mise en forme : nous avons parlé plus haut de la manière de s'en servir pour avoir les abréviations latines (☞ p. 120, 14.4) en italiques.

La ligne qui nous intéresse est donc la ligne 3, puisque nous voulons conditionner l'affichage du champ titre : s'il n'y a qu'une seule œuvre pour l'auteur courant, on peut ne pas l'afficher. Il suffit de redéclarer la macro, en insérant le test conditionnel :

```

1 | \renewbibmacro*{cite:title}{%
2 |   \printtext[bibhyperlink]{%
3 |     \ifsingletitle{}{\printfield[citetitle]{labeltitle}}%
4 |     \setunit{\nametitledelim}%
5 |     \bibstring[\mkibid]{opcit}}

```

☞ Les éditeurs et gérants de revues peuvent très bien définir leurs propres fichiers `.cbx` et `.bbx` pour obtenir un ensemble cohérent de styles. Ces fichiers, qui contiennent drivers et macros bibliographiques, doivent commencer par certaines commandes : nous renvoyons à la documentation de *biblatex*<sup>5</sup>.

<sup>4</sup>On pourrait si on voulait définir une autre manière d'afficher ce champ grâce à la commande `\DeclareFieldFormat`. Voir LEHMAN, « Formatting Directives », *op. cit.*

<sup>5</sup>Philipp LEHMAN, « Author Guide », dans : *The biblatex Package*.



# Troisième partie

## Faciliter la navigation

« Quand on ne sait pas où on va, il faut y aller... et le plus vite possible ! »

Devise shadok



# Chapitre 16

## Renvois internes

*Dans ce court chapitre nous allons nous intéresser à la manière de faire des renvois à l'intérieur d'un document. Il s'agit de permettre d'afficher des textes du type : « Nous renvoyons au chapitre N, page P. »*

### 16.1 Étiqueter des emplacements : `\label`

Pour pouvoir faire des renvois internes, il est nécessaire de placer des « étiquettes » aux endroits vers lesquels on souhaite renvoyer. Cet étiquetage se fait avec la commande `\label{\langle étiquette \rangle}`.

☞ Lors de la relecture, il peut être intéressant d'avoir les étiquettes affichées en marge. On utilisera pour ce faire le package `showlabels`.

### 16.2 Se servir des étiquettes

Après avoir avoir placé des étiquettes, on peut y renvoyer. Il suffit d'insérer une commande de renvoi à l'endroit souhaité. Il est toutefois nécessaire de compiler deux fois avec `XYTEX`. À la première compilation, `XYTEX` note les emplacements des étiquettes dans un fichier. À la seconde compilation, il lit ce fichier afin de procéder aux renvois.

⚡ Si vous modifiez votre texte, il faudra de nouveau compiler deux fois. En effet, les numéros de pages, de titres, de légendes, etc. peuvent avoir changé. Il faut donc que `XYTEX` les apprenne à nouveau. En résumé, pour avoir un document correct il faut :

1. Compiler avec `XYTEX` ;
2. Compiler avec Biber afin d'ajouter les références bibliographiques ;

3. Compiler avec  $X_{\text{L}}\text{TEX}$  ;
4. Re-compiler avec  $X_{\text{L}}\text{TEX}$ , car l'ajout des références bibliographiques a pu modifier le positionnement des étiquettes.

Tout ceci peut paraître bien compliqué et risque d'entraîner des oublis. C'est pourquoi il est conseillé d'utiliser le programme `latexmk` dont nous parlons en annexe (☞ p. 235, D).

### 16.2.1 Renvoyer à une page

Pour renvoyer au numéro de page correspondant à l'étiquette `<etiquette>`, on utilise la commande `\pageref{<etiquette>}`.

```

1 | blabla \label{etiquette}
2 | ...
3 | Nous renvoyons à la page~\pageref{etiquette}.
```

### 16.2.2 Renvoyer à un numéro de section

Pour renvoyer à un numéro de section, on utilise la commande `\ref`.

```

1 | \section{Section} \label{etiquette}
2 | ...
3 | Nous renvoyons à la section~\ref{etiquette}.
```

### 16.2.3 Renvoyer à un titre de section

Pour renvoyer à un titre de section, on utilise la commande `\nameref`.

```

1 | \section{Section} \label{etiquette}
2 | ...
3 | Nous renvoyons à la section \enquote{\nameref{etiquette}}.
```

Toutefois cette commande n'est pas disponible en standard : elle est proposée par le package `hyperref`. Il faut donc charger ce package dans le préambule. Nous documentons plus loin quelques fonctionnalités de ce package (☞ p. 141, 17.6).

## 16.3 Où placer la commande `\label` ?

Pour le moment, nous n'avons vu que des renvois vers des sections, mais le système de renvois est beaucoup plus souple.

Une étiquette permet de renvoyer à tout élément numéroté, comme un titre, une note de bas de page, une légende de flottant. Elle peut aussi renvoyer à un endroit précis en indiquant la page.

- Si l'étiquette `\label` est placée *dans* une commande d'élément numéroté, elle renvoie à cet élément. Par exemple, pour renvoyer à une figure (☞ p. 70, 8.4) :

```

1 | \begin{figure}[paramètre de placement]
2 |     Insertion de la figure
3 |     \caption{Légende\label{figure}}
4 | \end{figure}
5 | ...
6 |
7 | Nous renvoyons à la figure~\ref{figure} située p.~\pageref{figure}.

```

- Si la commande est placée ailleurs, elle renvoie à la page courante et à la section courante<sup>1</sup>.

## 16.4 Comment nommer ses étiquettes ?

Vous êtes bien sûr libre de trouver votre propre système de dénomination des étiquettes. Toutefois il est conseillé d'avoir quelque chose de la forme :  $\langle forme \rangle : \langle nom \rangle$ , où  $\langle forme \rangle$  désigne le type d'élément vers lequel on renvoie.

Exemples :

```

1 | \footnote{Blabla \label{note:nom}}
2 | \section{Titre \label{section:nom}}

```

---

<sup>1</sup>En réalité il est possible de la placer immédiatement après un élément numéroté pour y renvoyer, mais cela ne s'applique pas aux notes de bas de pages.



# Chapitre 17

## Sommaire et table des matières

*Dans ce chapitre nous verrons comment générer un sommaire ou une table des matières. Nous en profiterons pour étudier la manière de modifier les chaînes de langue de polyglossia, ainsi que pour aborder la notion de compteur en  $\text{\LaTeX}$ . Nous indiquerons également les bases de la configuration du package `hyperref`, qui permet, entre autres, de proposer des signets dans le PDF généré.*

### 17.1 Une table simple

La manière la plus simple de faire une table des matières est d'utiliser la commande `\tableofcontents`. Pour que  $\text{\LaTeX}$  puisse afficher cette table des matières, il est nécessaire de compiler deux fois : lors de la première compilation,  $\text{\LaTeX}$  stocke le contenu de la table des matières dans un fichier `.toc` ; lors de la seconde compilation, il se sert de ce fichier pour afficher la table des matières. En règle générale, cette double compilation n'est guère gênante, puisqu'elle est déjà nécessaire lors de l'utilisation d'une bibliographie (US<sup>3</sup> p. 76, 9.2).

⚡ La table des matières prend elle-même un certain volume dans le corps du texte. Par conséquent si votre table des matières se trouve en début d'ouvrage — il serait alors plus juste de l'appeler « sommaire » — elle décale la pagination lors de son affichage. C'est pourquoi il faut parfois compiler trois fois :

1. La première fois,  $\text{\LaTeX}$  stocke les informations dans le fichier `.toc`.
2. La seconde fois, la table des matières s'affiche, décalant la numérotation des pages.  $\text{\LaTeX}$  stocke les nouveaux numéros de pages dans le fichier `.toc`.

3. La troisième fois, la table des matières avec les bons numéros de page s’affiche. Comme sa longueur n’est pas différente de la précédente, il n’y a cette fois pas de décalage des numéros.



Pour personnaliser l’apparence de la table des matières, on peut utiliser le package *titletoc*.

## 17.2 Choisir la profondeur de la table des matières : la notion de compteur

Par défaut la commande `\tableofcontents` affiche tous les niveaux de titre, depuis `\part` jusqu’au `\subparagraph` (☞ p. 29, 3.1). Il est possible de restreindre la profondeur, ou au contraire de l’étendre ; il suffit de redéfinir un compteur  $\LaTeX$  : **tocdepth**.

Un compteur  $\LaTeX$  est un nombre entier stocké dans la mémoire vive de l’ordinateur. Un compteur peut être appelé à certains endroits et peut également être modifié. Typiquement, un compteur est associé à un élément numéroté. Il existe par exemple un compteur **page** correspondant au numéro de la page, incrémenté à chaque changement de page et utilisé pour afficher le numéro de page.

Dans notre cas, le compteur **tocdepth** n’est pas automatiquement modifié. En revanche il sert lors de l’affichage de la table des matières. Chaque niveau de titre possède un numéro (☞ p. 30, 3.1). La commande `\tableofcontents` affiche les niveaux de titre dont le numéro est inférieur ou égal à la valeur du compteur **tocdepth**.

Le niveau `\subsubsection` a comme numéro 3. Pour afficher dans la table des matières tous les niveaux de titres jusqu’au niveau `\subsubsection` inclus, il faut donc affecter la valeur 3 à **tocdepth**. On le fait en utilisant la commande `\setcounter` :

```
1 | \setcounter{tocdepth}{3}
2 | \tableofcontents
```

## 17.3 Table des matières ou sommaire ? Les chaînes de langues de polyglossia

Selon l’usage français, une table des matières se situe en fin d’ouvrage, un sommaire en début d’ouvrage. Par défaut la commande `\tableofcontents` indique que nous avons affaire à une « table des matières » et non à un « sommaire ». Comment avoir le terme « sommaire » ? Il suffit de redéfinir la chaîne de langue.

Nous avons déjà vu la notion de chaîne de langue (☞ p. 118, 14.4) pour la bibliographie. Il s’agit ici d’une idée similaire, mais pour le package *polyglossia*. La syntaxe est toutefois différente. Comme pour les chaînes de langue de la bibliographie, il faut commencer par découvrir quel est l’élément à modifier, en cherchant cette fois dans le fichier  `french.ldf` (☞ p. 233, C.4). Vous pouvez repérer une ligne contenant `\def\captionsfrench{%`. Cette ligne est suivie d’autres lignes sous la forme : `\def\langle chaîne\rangle{\langle Valeur\rangle}`, qui s’enchaînent jusqu’à une accolade fermante.

Pour notre cas, on repère la ligne contenant :

```
\def\contentsname{Table des matières}%
```

Notre chaîne de langue est donc `\contentsname`.

Pour redéfinir une chaîne de langue, il suffit d’écrire, de préférence dans le préambule :

```
1 | \gappto\captionsfrench{\renewcommand{\langle chaîne\rangle}{Valeur}}
```

Dans le cas présent, cela donne :

```
1 | \gappto\captionsfrench{\renewcommand{\contentsname}{Sommaire}}
```



La commande `\def` permet de créer de nouvelles commandes. Toutefois à la différence de la commande `\newcommand`, les commandes ainsi créées appartiennent à  $\TeX$  et non pas à  $\LaTeX$  (☞ p. 12, 1.3).

Définir des commandes en  $\TeX$  et non en  $\LaTeX$  permet plus de souplesse. Mais cela demande plus de maîtrise technique pour éviter divers problèmes, comme par exemple des erreurs de compilation. C’est pourquoi nous n’en parlons pas dans ce livre, sauf ponctuellement.

Les commandes  $\TeX$  peuvent contenir des « sous-commandes ». Ici par exemple, la commande `\captionsfrench` contient la sous-commande `\contentsname`. Nous utilisons la commande `\gappto` pour injecter une sous-commande dans une commande déjà définie.

La commande `\gappto` n’est une commande standard ni de  $\TeX$  ni de  $\LaTeX$ . Elle est définie par le package *etoolbox*, qui est lui même appelé par *polyglossia*.

## 17.4 Table des figures et table des tableaux

En plus de la table des matières,  $\LaTeX$  permet d’afficher une table des tableaux et une table des figures. Il est nécessaire, pour que ces tables soient constituées, que les tableaux et figures aient une légende (☞ p. 70, 8.4).

Pour la table des tableaux, il faut utiliser la commande `\listoftables` et pour celle des figures `\listoffigures`.

## 17.5 Plusieurs tables des matières

La commande `\tableofcontents` ne fonctionne qu’une seule fois dans un document  $\text{\TeX}$ . Comment faire si on souhaite avoir un sommaire et une table des matières, ou bien une table des matières à la fin de chaque partie ?

### 17.5.1 Un sommaire en plus d’une table des matières

Bien que cette pratique soit un peu désuète et rarement utile, on peut souhaiter avoir une table des matières en fin de travail et un sommaire en début. On utilise alors le package *shorttoc*. Celui-ci définit une commande `\shorttoc`  $\langle \text{titre} \rangle \{ \langle \text{niveau} \rangle \}$ .

Pour un sommaire qui ne reprendrait que les niveaux de section et de chapitre, il suffit donc d’écrire :

```
1 | \shorttoc{Sommaire}{0}
```

### 17.5.2 Sommaires partiels

Si l’on désire mettre un sommaire au début de chaque chapitre, il faut utiliser le package *minitoc*<sup>1</sup> en lui passant l’option `french`. On doit alors activer les mini-sommaires grâce à la commande `\dominitoc` :

```
1 | \usepackage[french]{minitoc}
2 | \dominitoc
```

Pour afficher les mini-sommaires il faut placer la commande `\minitoc` aux endroits voulus, par exemple après chaque titre de chapitre.

```
1 | \chapter{A}
2 | \minitoc
3 | ...
4 | \chapter{B}
5 | \minitoc
```

⚡ Le package ne peut calculer correctement les sommaires que si la commande `\tableofcontents` est utilisée. Si vous souhaitez n’afficher que des mini-sommaires mais pas de table générale des matières — ce qui est une idée assez étrange — vous pouvez utiliser `\faketableofcontents` à la place de `\tableofcontents`.

Pour afficher des mini-sommaires par parties, il faut utiliser la commande `\parttoc`. Par exemple :

---

<sup>1</sup>Jean-Pierre DRUCBERT, *The minitoc package*.

```

1  \part{A}
2  \parttoc
3  \chapter{1}
4  \chapter{2}
5  ...
6  \part{B}
7  \parttoc
8  ...

```

Le package *minitoc* est très complexe et contient de nombreuses subtilités ; il permet par exemple de choisir facilement les styles du sommaire. Nous renvoyons au manuel pour plus de détails<sup>2</sup>.

## 17.6 Des signets dans le PDF : le package *hyperref*

Certains PDFs proposent des signets qui permettent d'accéder rapidement à un endroit précis du document<sup>3</sup>. Il est possible avec  $\LaTeX$  d'obtenir un PDF contenant des signets correspondant aux différents niveaux de titre. Pour ce faire il faut utiliser le package *hyperref*.

Ce package propose de nombreuses fonctionnalités : nous avons vu qu'il permettait notamment de créer des renvois vers les titres de sections (☞ p. 134, 16.2.3). Mais il introduit aussi des liens cliquables en interne, par exemple entre un appel de note de bas de page et la note correspondante. Il permet aussi de configurer un certain nombre de méta-données du document PDF.

### 17.6.1 Signet de navigation

Pour générer des signets correspondant au plan du travail, il suffit simplement d'appeler le package et de compiler deux fois — ce qui est de toute façon nécessaire pour obtenir la table des matières.

Un problème se pose toutefois lorsque l'on utilise un titre non numéroté (☞ p. 30, 3.1.1). En dépit de la commande `\addcontentsline`, ce titre n'est pas ajouté dans la liste des signets. Il faut en fait faire précéder cette commande de `\phantomsection`.

```

1  \phantomsection
2  \addcontentsline{toc}{chapter}{Introduction}
3  \chapter*{Introduction}

```

<sup>2</sup>Notamment Jean-Pierre DRUCBERT, « Typesetting of the mini-tables », dans : *The minitoc package*.

<sup>3</sup>L'utilisation concrète de ces signets dépend du lecteur PDF.

## 17.6.2 Réglage des liens

Par défaut, *hyperref* encadre en rouge les liens hypertextes. Ces cadres n'apparaissent pas à l'impression.

On peut vouloir les faire disparaître aussi du document PDF. Il faut alors indiquer :

1. Que nous voulons que les liens soient colorés et non encadrés.
2. Que nous voulons que la couleur soit le noir. En effet la couleur des liens est conservée à l'impression, contrairement à la couleur des cadres.

Nous allons pour cela utiliser la commande `\hypersetup` dans le préambule. La commande prend comme argument des paramètres du package, séparés par des virgules. Pour notre cas, les arguments nécessaires sont :

```

1 | \hypersetup{colorlinks=true,
2 | citecolor=black,
3 | filecolor=black,
4 | linkcolor=black,
5 | urlcolor=black}

```

`colorlinks=true` indique que nous colorons les liens au lieu de les entourer. Pour *hyperref* il existe plusieurs types de liens dont la couleur est configurable. Ici nous indiquons qu'il faut mettre en noir les types de liens suivants :

- les liens de renvois d'une référence bibliographique abrégée à la référence complète ;
- les liens appelant des fichiers externes au PDF ;
- les liens internes (notes de bas de pages, renvois) ;
- les liens externes.

# Chapitre 18

## Index

Dans ce chapitre nous verrons comment faire un ou plusieurs index<sup>1</sup>.

### 18.1 Faire un index simple avec *imakeidx*

#### 18.1.1 Principe de base

Pour indexer un document, il faut utiliser le package *imakeidx* et placer dans le préambule la commande `\makeindex`<sup>2</sup>.

```
1 | \usepackage{imakeidx}  
2 | \makeindex
```

#### Indexer son document

On indexe son document avec la commande `\index{⟨entrée⟩}`. L'entrée apparaît dans l'index sous la forme indiquée par l'argument `⟨entrée⟩`, suivie du numéro de la page où cette commande est placée dans le texte.

Dans l'exemple qui suit<sup>3</sup>, l'index comporte ainsi cinq entrées : « Charlemagne », « Adrien », « Tassilon », « Formose » et « Damase » :

```
1 | Tandis que Charle\index{Charlemagne} était à Rome, il convint  
2 | avec le pape Adrien\index{Adrien} qu'ils enverraient de concert
```

---

<sup>1</sup>Je remercie ma sœur Enimie pour l'avoir rédigé en grande partie.

<sup>2</sup>Le *imakeidx* remplace à la fois le *makeindex* et le *splitindex*, dont il reprend les commandes. Il est plus simple à utiliser, notamment pour la mise en page de l'index 25, et ne nécessite qu'une compilation, contrairement à ses prédécesseurs. Pour obtenir des informations sur les commandes d'indexation, il faut consulter les manuels de *makeindex* et de *SplitIndex*.

<sup>3</sup>ÉGINHARD, *Annales*, Année 781, dans : *Collection des mémoires relatifs à l'histoire de France*, trad. par François GUIZOT, Paris : Brière, 1824, p. 23.

```

3 | des ambassadeurs à Tassilon, duc de Bavière\index{Tassilon}
4 | \textelp{}
5 | Les hommes choisis et envoyés dans cette ambassade furent, de
6 | la part du pape, les évêques Formose\index{Formose} et
7 | Damase\index{Damase}\textelp{}.
```

Lorsqu'une entrée est référencée deux fois dans la même page, cette page n'est indiquée qu'une seule fois.

⚡ Il vaut mieux accoler la commande `\index{⟨entrée⟩}` directement au mot à indexer, sans laisser d'espace, pour éviter toute ambiguïté en cas de changement de page.

L'indexation d'un texte n'est pas automatique. Il faut placer `\index{⟨entrée⟩}` à chaque endroit que l'on veut référencer. On peut bien sûr créer une commande spécifique pour combiner l'indexation avec d'autres actions.

Ainsi, pour indexer automatiquement tous les noms propres d'un texte, déclarons la commande suivante :

```

1 | \newcommand\auteur[2]{#1~\textsc{#2}\index{#2, #1}\xspace}
```

Il suffit ensuite, au cours de la rédaction de son texte, de frapper, par exemple, `\auteur{Victor}{Hugo}` pour obtenir « Victor HUGO » dans le corps de son texte, indexé sous l'entrée « Hugo, Victor ».

## Générer l'index

Si notre fichier principal s'appelle `exemple.tex`, la commande `\makeindex` indique à  $\text{\LaTeX}$ , lors de la compilation avec  $\text{\XeLaTeX}$ , de créer un fichier `exemple.idx`, contenant la liste de toutes les entrées.  $\text{\LaTeX}$  génère aussi un fichier `exemple.ilg` qui contient les messages de compilation de l'index, et un fichier `exemple.ind` qui contient l'index formaté. L'index apparaît dans le document à l'emplacement que l'on a indiqué par la commande `\printindex`.

### 18.1.2 Allons plus loin

#### Créer des subdivisions

Il est possible, avec `imakeidx`, de créer des subdivisions et des subdivisions pour chaque entrée de l'index. La subdivision se crée de la manière suivante : `\index{⟨entrée⟩!⟨sous-entrée⟩}`. La sous-sous-entrée, logiquement, se crée ainsi : `\index{⟨entrée⟩!⟨sous-entrée⟩!⟨sous-sous-entrée⟩}`. On ne peut cependant avoir que ces trois niveaux d'indexation.

On peut ainsi référencer autrement notre premier exemple, en créant les entrées « Évêques » et « Ducs » que l'on subdivise :

```

1 | Tandis que Charle\index{Charlemagne} était à Rome, il convint
2 | avec le pape Adrien\index{Adrien} qu'ils enverraient de concert
3 | des ambassadeurs à Tassilon, duc de Bavière\index{Ducs!Tassilon}.
4 | \textelp{}
5 | Les hommes choisis et envoyés dans cette ambassade furent, de la
6 | part du pape, les évêques Formose\index{Évêques!Formose}
7 | et Damase\index{Évêques!Damase}\textelp{}.
```

Supposons que ce texte soit à la page 5 ; on obtient ainsi dans l'index :

```

Adrien, 5

Charlemagne, 5

Ducs
  Tassilon, 5

Évêques
  Damase, 5
  Formose, 5
```

Bien entendu, on pourrait encore rajouter une subdivision, en distinguant par exemple « Clercs » et « Laïcs », et dans la première catégorie en distinguant « Évêques » et « Papes ».

### Faire des références croisées

Pour qu'une entrée dans l'index renvoie à une autre entrée, on utilise la commande `\index{<entrée>|see<entrée à laquelle on renvoie>}`.

Ainsi, `\index{Tassilon|see{Ducs}}` donne dans l'index :

```
Tassilon, voir Ducs
```

La traduction de *see* change selon la langue indiquée comme `\setmainlanguage` (☞ p. 22, 2.2.3).

### Créer des entrées sur plusieurs pages

Si l'on veut référencer dans l'index non pas un mot mais un passage, il faut placer la commande `\index{<entrée>|{}` au début du passage à indexer et la commande `\index{<entrée>|}` à la fin. Si le passage commence à la page *x* et se termine à la page *y*, on obtient dans l'index :

```
entrée x-y
```

## Entrées formatées

L'indexation avec  $\text{\LaTeX}$  ne gère pas correctement les accents indiqués dans l'argument de la commande `\index` : il classe les mots commençant par un accent à la fin de l'index. La syntaxe `\index{\langle entrée \rangle @ \langle entrée formatée \rangle}` permet de résoudre ce problème. Ainsi, si l'on veut créer une entrée « écrivains », qui soit triée à « e », il faut insérer `\index{ecrivains@écrivains}`.

La commande `\index{\langle entrée \rangle @ \langle entrée formatée \rangle}` permet donc de classer une entrée où l'on veut dans l'index. Pour faire apparaître, par exemple, les empereurs romains dans l'ordre chronologique et non dans l'ordre alphabétique, on peut utiliser les commandes suivantes :

```
1 | \index{Empereurs !empereur1@Auguste}
2 | \index{Empereurs !empereur2@Tibère}
3 | \index{Empereurs !empereur3@Claude}
4 | et ainsi de suite.
```

On obtient alors :

```
Empereurs
  Auguste, x
  Tibère, y
  Claude, z
```

Cette syntaxe est aussi utile pour mettre en évidence une entrée dans l'index en modifiant son aspect : `\index{\langle entrée \rangle @ \textbf{\langle entrée formatée \rangle}}`, fait apparaître l'entrée en gras dans l'index. Ceci est valable pour toutes les commandes agissant sur la fonte. Modifions ainsi comme suit la commande `\auteur` que l'on a créée précédemment (☞ p. 144, 18.1.1) :

```
1 | \newcommand{\auteur[2]}{%
2 |   #1~\textsc{#2}\index{#2 #1@~\textsc{#2}, #1}\xspace}
```

Désormais, `\auteur{Victor}{Hugo}` produit dans l'index « HUGO, Victor ».

## Formater le numéro des pages

Il peut arriver, lorsqu'une entrée est très souvent représentée dans un texte indexé, que l'on veuille mettre en valeur une de ses occurrences, en faisant apparaître en gras dans l'index le numéro de la page où elle se situe : on utilise alors la commande `\index{\langle entrée \rangle | \textbf{}}`.

⚡ Il s'agit bien de `| \textbf{}`, non de `|\textbf{}`.

De même pour faire apparaître le numéro de la page en italique utilise-t-on la commande `\index{⟨entrée⟩|textit}`.

☞ Si vous utilisez le package *hyperref*, vous constaterez que celui-ci insère des liens hypertextes vers les pages au sein de l’index. Toutefois si une de ces pages est formatée, le lien disparaît. Nous expliquons sur notre site comment éviter ce problème<sup>4</sup>.

### 18.1.3 Quelques options du package *imakeidx*

La commande `\makeindex` peut recevoir des arguments optionnels sous la forme `\makeindex[⟨clef=valeur⟩]` ; s’il y a plusieurs arguments, ils doivent être séparés par une virgule. En voici trois<sup>5</sup> :

**title** Permet de changer le nom de l’index, qui par défaut est « Index ». Par exemple, le code suivant permet d’avoir un index intitulé « Index rerum »

```
1 | \makeindex[title = Index rerum]
```

**columns** Indique le nombre de colonnes. Par défaut, l’index est en deux colonnes. Si l’on veut un index en une seule colonne, il suffit de mettre :

```
1 | \makeindex[title = Index rerum, columns = 1]
```

**intoc** Indique si l’index doit apparaître ou non dans la table des matières. Par défaut, l’index n’apparaît pas dans la table des matières. Pour l’y mettre, il suffit d’ajouter cet argument dans la liste.

```
1 | \makeindex[title = Index rerum, columns = 1, intoc]
```

Il existe une autre commande, la commande `\indexsetup`, qui peut, elle aussi, recevoir plusieurs arguments sous la forme `{⟨clef=valeur⟩}`. Voici ceux qui peuvent être utile lorsque l’on n’a qu’un index (nous en verrons quelques autres à la section suivante) :

**level** Indique le niveau de division (section, chapitre, etc) auquel correspond l’index. Par défaut, il s’agit de `\chapter*`.

<sup>4</sup>Maïeul ROUQUETTE, *Pages en gras avec un lien hypertexte dans un index*, URL : <http://geekographie.maieul.net/59>.

<sup>5</sup>Nous en verrons un quatrième pour les index multiples, et un autre lorsque nous évoquerons le formatage de l’index. Pour une liste exhaustive, voir Claudio BECCARI et Enrico GREGORIO, *Package imakeidx*.

**toclevel** Indique le niveau de division (section, chapitre, etc) auquel correspond l'index dans la table des matières. À l'inverse de l'argument précédent, il n'y a pas de contre-oblique :

```
1 | \indexsetup[level=\section*, toclevel=section]
```

Pour terminer, voici la commande `\indexprologue[⟨espace⟩]{⟨texte⟩}` qui permet de mettre un prologue avant l'index. La commande se place juste avant `\printindex`<sup>6</sup>. L'argument `⟨espace⟩` (par défaut `\bigskip`) indique l'espace entre le prologue de l'index et la première entrée ; il doit contenir une commande d'espacement vertical (☞ p. 209, 24.1). L'argument `{⟨texte⟩}` contient le prologue en lui même. Par exemple, pour un index des notions :

```
1 | \indexprologue{Les numéros en gras renvoient aux définitions de notions.}
2 | \printindex
```

## 18.2 Faire plusieurs index

### 18.2.1 Définir ses index

Le package *imakeidx* permet de faire plusieurs index pour un même document. La première étape consiste à définir ces index. On les déclare dans le préambule en utilisant pour chaque index la commande `\makeindex`, déjà étudiée. Chaque index reçoit un nom abrégé — à ne pas confondre avec son titre — qui permettra ensuite d'indiquer à quel index appartient telle entrée indexée dans votre document. Le nom abrégé de l'index est indiqué comme option à la commande `\makeindex`.

Ainsi, pour faire un index des noms propres et un index général, on peut déclarer :

```
1 | \makeindex[title=Index principal]
2 | \makeindex[name=npr, title=Index des noms propres]
```

On remarque ici qu'aucun nom n'a été donné à l'index principal : dans ce cas, le nom abrégé est automatiquement « `idx` ». À la compilation, seront créés, en plus des fichiers `exemple.idx`, `exemple.ilg` et `exemple.ind`, les fichiers `npr.idx`, `npr.ilg` et `npr.ind`.

### 18.2.2 Indexer son texte

Une fois les index déclarés, il faut passer à l'indexation proprement dite. Le principe est le même que pour un seul index, mais au lieu de `\index{⟨entrée⟩}`, on utilise : `\index[⟨nom abrégé⟩]{⟨entrée⟩}`.

<sup>6</sup>Il peut y avoir autant de prologues que d'index.

☞ Toutes les entrées indiquées par `\index{⟨entrée⟩}` sans l'argument *⟨nom abrégé⟩*, sont automatiquement placées dans l'index général « idx ».

On peut ainsi indexer notre texte d'Éginhard de la façon suivante :

```

1 | \index{Charles et la papauté|()}
2 | Tandis que Charle\index[npr]{Charlemagne} était à Rome, il convint
3 | avec le pape Adrien\index[npr]{Adrien} qu'ils enverraient de concert
4 | des ambassadeurs à Tassilon, duc de Bavière\index[npr]{Tassilon}
5 | \textelp{}
6 | Les hommes choisis et envoyés dans cette ambassade furent, de
7 | la part du pape, les évêques Formose\index[npr]{Formose} et
8 | Damase\index[npr]{Damase}(...)\index{Charles et la papauté|)}.
```

### 18.2.3 Imprimer les index

Pour imprimer un index, il suffit d'utiliser la commande `\printindex` en lui passant le nom abrégé de l'index en option. Ainsi, pour imprimer l'index général suivi de celui des noms propres :

```

1 | \printindex
2 | \printindex[npr]
```

☞ Si l'on veut regrouper tous les index en un seul chapitre dont chaque index est une section, on utilisera la commande `\indexsetup`. On peut lui passer comme option [*⟨noclearpage⟩*], pour éviter que chaque index commence à une nouvelle page. Bien sûr, chaque index peut avoir son propre prologue. Voici un exemple :

```

1 | \indexsetup{level=\section*,toclevel=section,noclearpage}
2 | ...
3 | \chapter*{Indices}
4 | \indexprologue{Les numéros en gras renvoient aux définitions de notions.}
5 | \printindex
6 |
7 | \indexprologue{Les auteurs anciens sont indiqués en italiques.}
8 | \printindex[npr]
```

## 18.3 Indexer ses sources

Nous allons maintenant voir comment utiliser les possibilités de *biblatex* et de *imakeidx* pour établir un index des sources primaires.

Pour comprendre cette section, vous devez vous être familiarisé avec les indications sur les macros bibliographiques (☞ p. 121, 15.1).

### 18.3.1 Premier essai

La documentation de *biblatex*<sup>7</sup> nous informe qu’il existe au chargement du package une option `indexing` qui permet d’indexer automatiquement les références bibliographiques. Comme nous ne souhaitons indexer que les références appelées par les commandes `\langle prefix \rangle cite` — et non celles appelées par la commande `\printbibliography` — nous attribuons la valeur `cite` à cette option.

```
1 | \usepackage[indexing=cite]{biblatex}
```

Étant donné qu’il faut à la fois interpréter le fichier `.bib` et faire un index, nous devons procéder aux compilations dans l’ordre suivant :

1. Compilation avec X<sub>Y</sub>TEX.
2. Compilation avec Biber.
3. Compilation avec X<sub>Y</sub>TEX pour que les données bibliographiques soient intégrées dans l’index.

On constate cependant deux problèmes :

1. La bibliographie se trouve mêlée aux autres entrées de l’index.
2. Plus grave : nous avons des entrées pour les auteurs et des entrées pour les titres, au lieu d’avoir des entrées sous la forme : Auteur!Titre.

En outre nous aimerions :

1. Limiter l’indexation aux sources primaires.
2. Indexer aussi, comme troisième niveau d’index, le champ `titleaddon` qui nous sert pour les divisions de source (☞ p. 112, 13.4).

### 18.3.2 Création d’un index spécifique

Pour créer un index spécifique aux sources, rien de particulier : il suffit d’utiliser `imakeidx` et la commande `\makeindex` :

```
1 | \makeindex[name=sources,title=Sources]
```

### 18.3.3 Modifications des macros de *biblatex*

Nous avons donc notre index spécifique. Mais encore faut-il que nous disions à *biblatex* d’y écrire son index. Pour ce faire nous allons d’abord redéfinir la macro `citeindex` qui est appelée à chaque commande `\langle prefix \rangle cite`.

---

<sup>7</sup>LEHMAN, « Package Options », *op. cit.*

```

1  \renewbibmacro{citeindex}{%
2    \ifciteindex{%
3      \indexnames[sources]{author}%
4      \indexfield[sources]{indextitle}%
5      \indexfield[sources]{titleaddon}%
6      \index[sources]{---}%
7    }%
8  {}

```

**ligne 2** la commande `\ifciteindex` vérifie que l’option **indexing** de *biblatex* est bien égale à `true` ou bien à `cite` : ce qui suit entre accolades est exécuté si tel est le cas.

**ligne 3** nous indexons le champ **author**. Nous utilisons le format d’indexation `sources`.

**ligne 4** nous indexons le champ **indextitle**. Ce champ spécial sert à avoir dans l’index un autre titre que dans le corps du document. Si ce champ est vide *biblatex* utilise à la place le champ **title**. Nous utilisons le format d’indexation `sources`.

**ligne 5** nous indexons le champ **titleaddon**. Nous utilisons le format d’indexation `sources`.

**ligne 6** une des limitations de *biblatex* est qu’il ne peut indexer qu’un seul champ à la fois, et n’est pas capable, pour le moment, de produire des entrées d’index à plusieurs niveaux. Avec cette macro **citeindex**, on obtient une indexation séparée pour chaque champ des entrées indexées. Or nous voudrions obtenir une indexation correspondant à la commande :

```

1  | \index{auteur!titre!titleaddon}

```

C’est pourquoi nous avons conçu un script dans le langage python<sup>8</sup>, qui concatène dans le fichier `.idx` les trois indexations en une seule. Mais avant d’utiliser ce script, il faut indexer une fausse valeur, la valeur `---`, qui empêchera le script d’indexer tous les champs d’une entrée : nous ne voulons pas obtenir une entrée de la forme :

```
\index{auteur!titre!titleaddon!auteur2!titre2!}
```

Toutefois, ce script python doit être exécuté avant que  $\LaTeX$  ne transforme le fichier `.idx` en fichier `.ind`. Cette transformation est faite automatiquement par le package *imakeidx*. Cependant,

<sup>8</sup>Nous expliquons plus loin comment s’en servir (p. 153, 18.3.5).

nous pouvons désactiver cet automatisme pour un index particulier, dans le cas présent pour l’index sources. Pour ce faire, il nous suffit de passer l’option `noautomatic` à la commande `\makeindex` :

```
1 | \makeindex[name=sources,title=Sources,noautomatic]
```

Il nous faudra alors, après l’exécution du script python, compiler le fichier `sources.idx` avec le script `MakeIndex`, afin de produire un fichier `sources.ind` :

```
| makeindex sources.idx
```

Après cela, la compilation avec XeLaTeX affiche correctement l’index.

### 18.3.4 Fomat d’indexation *biblatex*

Nous avons dit que nous utilisons le format d’indexation `sources`. Un format d’indexation *biblatex* est simplement la description de l’opération que *biblatex* effectue lorsqu’il doit indexer un champ. Il nous faut donc définir ce format grâce aux commandes `\DeclareIndexNameFormat` et `\DeclareIndexFieldFormat`.

#### Indexation des noms

```
1 | \DeclareIndexNameFormat{sources}{%
2 |   \usebibmacro{index:name}{\index[sources]}{#1}{#3}{#5}{#7}
3 | }
```

Nous indiquons en première ligne que nous déclarons un format d’indexation `sources` pour les noms propres. Dans la ligne suivante, nous déclarons ce que nous faisons : nous appelons une macro `index:name`.

Cette macro est déjà définie par *biblatex*. Elle reçoit plusieurs arguments. Le premier argument est la commande à exécuter : ici `\index[sources]`, qui permet d’indexer dans l’index `sources` défini plus haut. Les autres arguments sont repris des codes de *biblatex* et désignent les différentes parties du nom à indexer<sup>9</sup>.

#### Indexation des autres champs

```
1 | \DeclareIndexFieldFormat{sources}{%
2 |   \ifcurrentfield{indextitle}{\index[sources]{#1@\emph{#1}}}%
3 |   {\index[sources]{#1}}%
4 | }
```

<sup>9</sup>Nous renvoyons le lecteur à la documentation de *biblatex* : Philipp LEHMAN, « Formatting Directives », dans : *The biblatex Package*.

La commande `\DeclareIndexFieldFormat` sert à déclarer la manière d’indexer les champs qui ne sont ni des listes ni des noms. La valeur #1 correspond à la valeur du champ à indexer. En deuxième ligne, nous vérifions grâce à la commande `\ifcurrentfield`, que le champ est **indextitle** : si c’est le cas, nous l’indexons dans l’index source en mettant l’emphase sur le titre pour l’affichage final. Sinon, nous l’indexons simplement dans l’index source.

### 18.3.5 Compilation et concaténation des index

Après la compilation  $X_{\text{E}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ , nous obtenons un fichier `sources.idx`. Si vous l’ouvrez vous constaterez que nous avons des entrées sous la forme :

```
1 | \indexentry{Author}{page}
2 | \indexentry{Titleindex@\emph {Titleindex}}{page}
3 | \indexentry{Titleaddon}{page}
4 | \indexentry{--}{page}
```

Nous souhaitons remplacer ces entrées par des entrées sous la forme :

```
1 | \indexentry[sources]{Author@Author!Titleindex@
2 | \emph {Titleindex}!Titleaddon@Titleaddon}{page}
```

L’auteur de ces lignes a développé un script permettant d’automatiser cette transformation. Par ailleurs ce script modifie également l’ordre de tri pour tenir compte des accents.

Pour utiliser ce script, il vous faut :

- Avoir le logiciel Python installé sur votre ordinateur. Ce logiciel est installé en standard sous Mac Os X et sur la plupart des distributions Linux, mais pas sous Windows<sup>10</sup>.
- Télécharger le fichier <https://github.com/maieul/indexation-sources/zipball/stable>, le décompresser.
- Mettre les fichiers `index.py` et `roman.py` dans le répertoire du fichier `.idx`.
- Ouvrir le fichier `index.py` et modifier la ligne 8 en remplaçant `xxx.idx` par le nom du fichier à concaténer, en l’occurrence `sources.idx`.
- En ligne de commande (☞ p. 231, C) se rendre dans le répertoire, puis frapper l’entrée : `python index.py`.

Après cette concaténation nous devons compiler l’index « sources » en ligne de commande, via le script `MakeIndex` :

```
| makeindex sources
```

<sup>10</sup>Python for windows, URL : <http://www.python.org/download/windows/>.

### 18.3.6 Raffinement

Nous souhaitons n'indexer que les sources primaires. La solution la plus simple est d'utiliser dans le fichier `.bib` un champ personnalisé `usera`. Le package *bibtex* permet en effet à l'utilisateur d'utiliser librement un certain nombre de champs<sup>11</sup>. Dans ce champ, mettre 1 si l'entrée est une source primaire, 2 si l'entrée est une source secondaire.

Il nous suffit de modifier la macro `citeindex`, en introduisant un test (ligne 3) sur la valeur du champ `usera`, grâce à la commande `\iffieldequalstr`.

```

1  \renewbibmacro{citeindex}{%
2  \ifciteindex{%
3  \iffieldequalstr{usera}{1}{%
4  \indexnames[sources]{author}%
5  \indexfield[sources]{indextitle}%
6  \indexfield[sources]{titleaddon}%
7  \sindex[sources]{--}%
8  }{}}%
9  {}}
```

### 18.3.7 Résumé des diverses compilations

Pour obtenir un index des sources primaires, une fois tous les fichiers mis en place, il nous faut donc procéder dans le terminal aux opérations suivantes :

1. `xelatex xxx.tex`
2. `biber xxx`
3. `xelatex xxx.tex`<sup>12</sup>
4. `python index.py`
5. `makeindex sources`
6. `xelatex xxx`



Évidemment, il peut être fastidieux de se souvenir de l'ensemble de ces opérations, de les faire et refaire...

Il existe un programme nommé `latexmk` qui permet d'automatiser ce genre d'opération : nous en parlons en annexe (☞ p. 235, D).

<sup>11</sup>La liste de ces champs est fournie dans Philipp LEHMAN, « Custom Fields », dans : *The biblatex Package*.

<sup>12</sup>Si un sommaire se situe en début d'ouvrage, il peut être nécessaire de compiler plusieurs fois.

# Quatrième partie

## Autres outils pour les sciences humaines

« Le golf? Un sport où l'on doit tenter de contrôler une balle avec des outils tout à fait inadaptés à cet effet. »

Thomas Woodrow WILSON, Président des États-Unis  
d'Amérique



# Chapitre 19

## Gestion des références bibliques avec *Bibleref*

*On peut évidemment considérer la Bible comme une référence bibliographique comme une autre, et inclure ses livres dans la bibliographie générale. La présentation d'une référence biblique diffère toutefois bien souvent de celle d'une référence bibliographique. C'est pourquoi on peut utiliser le package `bibleref` et sa traduction `bibleref-french`.*

### 19.1 Chargement du package et options

Pour que nos références bibliques soient en français, il suffit simplement d'appeler le package `bibleref-french`. Celui-ci charge alors le package `bibleref`.

```
1 | \usepackage{bibleref-french}
```

↪ Au cours de la rédaction de ce livre, ce package a été amélioré. Nous documentons certaines de ces améliorations.

Veillez donc à avoir la dernière version du package, en procédant, si besoin, à une mise à jour (☞ p. 225, A.5).

#### 19.1.1 Forme catholique ou protestante

Le livre que les protestants appellent « Ésaïe » est appelé « Isaïe » chez les catholiques. Pour choisir la forme protestante, il suffit de passer l'option `protestant` à l'appel du package. Le package uniformise les références.

```
1 | \usepackage[protestant]{bibleref}
```

### 19.1.2 Styles de références

Les éditions de la Bible divergent parfois sur la manière d’afficher une référence biblique. En outre, les références peuvent s’afficher tantôt en version abrégée, tantôt en version complète. C’est pourquoi il existe plusieurs styles de références bibliques.

On choisit son style par la commande `\biblerefstyle{<style>}`, de préférence dans le préambule, ou bien par une option de package :

```
1 | \usepackage[<style>]{bibleref}
```

La liste des styles est présentée dans le tableau ci-dessous :

Style	Exemple
default	2 Corinthiens 12, 1–5 ; 7 ; Qohélet 1, 3 ; 2, 5 ; Isaïe 10
defaultshorter	2 Co 12, 1–5 ; 7 ; Qo 1, 3 ; 2, 5 ; Is 10
jerusalem	2 Co 12 1–5, 7 ; Qo 1 3 ; 2 5 ; Is 10
colombe	2 Co 12.1–5,7 ; Qo 1.3 ; 2.5 ; Es 10
colombefull	2 Corinthiens 12.1–5,7 ; Qohélet 1.3 ; 2.5 ; Ésaïe 10
NBS	2 Co 12.1-5,7 ; Qo 1.3 ; 2.5 ; Es 10
NBSfull	2 Corinthiens 12.1-5,7 ; Qohélet 1.3 ; 2.5 ; Ésaïe 10
TOB	2 Co 12.1-5, 7 ; Qo 1.3 ; 2.5 ; Es 10
TOBfull	2 Corinthiens 12.1-5, 7 ; Qohélet 1.3 ; 2.5 ; Ésaïe 10
text	Deuxième Épître aux Corinthiens, chapitre douze verset un à cinq, sept ; Livre du Qohélet, chapitre un verset trois, chapitre deux verset cinq ; Livre d’Isaïe, chapitre dix

En l’absence de style précisé, c’est le style « default » qui l’emporte. Il est évidemment possible de créer ses propres styles : nous renvoyons au manuel de *biblerref* pour en savoir plus<sup>1</sup>.

<sup>1</sup>Nicola TALBOT, « Defining New Styles », dans : *biblerref.sty v1.14 : a LATEX2e package for typesetting bible references*.

## 19.2 Référencer un passage biblique

La syntaxe pour référencer un passage biblique varie selon que l'on souhaite référencer un livre en entier, un chapitre ou certains versets seulement. Voici l'ensemble des syntaxes disponibles<sup>2</sup> :

Syntaxe	Résultat
<code>\bibleverse{Ex}</code>	Exode
<code>\bibleverse{Ex}(20:)</code>	Exode 20
<code>\bibleverse{Ex}(20:17)</code>	Exode 20, 17
<code>\bibleverse{IICo}(12:21)</code>	2 Corinthiens 12, 21
<code>\bibleverse{IICo}(12:21-32)</code>	2 Corinthiens 12, 21–32
<code>\bibleverse{Ex}(20:17)(21:3)</code>	Exode 20, 17 ; 21, 3
<code>\bibleverse{Ex}(15:)(17:)(20:)</code>	Exode 15 ; 17 ; 20
<code>\bibleverse{Ap}(1:2,5,7-9,11)</code>	Apocalypse 1, 2 ; 5 ; 7–9 ; 11
<code>\bibleverse{ICh}(1:3)-(2:7)</code>	1 Chroniques 1, 3–2, 7
<code>\bibleverse{Lc}(1,4-5,8:)</code>	Luc 1 ; 4–5 ; 8

Voici la liste des abréviations des livres :

Abréviation	Livre
Gn	Genèse
Ex	Exode
Lv	Lévitique
Nb	Nombres
Dt	Deutéronome
Jos	Josué
Jg	Juges
Rt	Ruth
IS	1 Samuel
IIS	2 Samuel
IR	1 Rois
IIR	2 Rois
ICh	1 Chroniques
IICh	2 Chroniques
Esd	Esdras
Ne	Néhémie
Est	Esther
Tb	Tobit
Jdt	Judith
IM	1 Maccabées

<sup>2</sup>Nous ne faisons ici que reprendre le manuel.

<b>Abréviation</b>	<b>Livre</b>
IIM	2 Maccabées
Jb	Job
Ps	Psaumes
Pr	Proverbes
Qo	Qohéleth
Ct	Cantiques des cantiques
Sg	Sagesse
Si	Siracide
Is	Isaïe
Es	Ésaïe
Jr	Jérémie
Lm	Lamentations
Ba	Baruch
Ez	Ezéquier
Dn	Daniel
Os	Osée
Jl	Joël
Am	Amos
Ab	Abdias
Jon	Jonas
Mi	Michée
Na	Nahum
Ha	Habaquq
So	Sophonie
Ag	Aggée
Za	Zacharie
Ml	Malachie
Mt	Matthieu
Mc	Marc
Lc	Luc
Jn	Jean
Ac	Actes
Rm	Romains
ICo	1 Corinthiens
IICo	2 Corinthiens
Ga	Galates
Ep	Éphésiens
Ph	Philippiens
Col	Colossiens
ITh	1 Thessaloniens
IITh	2 Thessaloniens

Abréviation	Livre
ITm	1 Timothée
IITm	2 Timothée
Tt	Tite
Phm	Philémon
He	Hébreux
Jc	Jacques
1P	1 Pierre
2P	2 Pierre
IJn	1 Jean
IIJn	2 Jean
IIIJn	3 Jean
Jude	Jude
Ap	Apocalypse

 Les livres proposés correspondent au canon catholique. On peut ajouter d'autres livres : nous renvoyons à la documentation de *bibleref*<sup>3</sup>.

## 19.3 Indexer une référence

On peut remplacer la commande `\bibleverse` par `\ibibleverse`. Cette commande, en plus d'afficher indexe une référence biblique en plus de l'afficher.

Il existe en outre deux commandes ayant la même syntaxe que `\ibibleverse` : `\ibiblechvs`, qui n'affiche pas le nom du livre dans le texte, et `\ibiblevs` qui n'affiche dans le texte que le numéro de verset.

### 19.3.1 Tri dans l'index<sup>4</sup>

Vous pouvez constater que l'ordre de tri dans l'index n'est par défaut pas très bon. Par exemple, les livres numérotés sont classés à la lettre « i ».

Le package fournit des options pour un meilleur tri<sup>5</sup>. Au chargement du package, vous pouvez choisir l'une de ces options :

**indexalphac**      les livres sont triés par ordre alphabétique, suivant les noms catholiques.

<sup>3</sup>Nicola TALBOT, « `addbiblebook` », dans : *bibleref.sty v1.14 : a LATEX2e package for typesetting bible references*.

<sup>4</sup>Ce passage reprend, pour l'essentiel, la documentation du package, dont nous sommes l'auteur.

<sup>5</sup>Pour l'Ancien Testament, les protestants suivent le canon juif pour son contenu mais utilisent l'ordre du canon catholique : voir Albert (de) PURY, « Le canon de l'Ancien Testament », dans : *Introduction à l'Ancien Testament*, dir. Thomas RÖMER, Jean-Daniel MACCHI et Christophe NIHAN, Genève : Labor et Fides, 2005, p. 18–21.

<b>indexalphap</b>	les livres sont triés par ordre alphabétique, suivant les noms protestants.
<b>indexLXX</b>	les livres sont triés dans l'ordre des bibles chrétiennes classiques, pour l'Ancien Testament selon la Septante. L'Ancien et le Nouveau Testaments ne sont pas séparés.
<b>indextestamentsL</b>	les livres sont triés dans l'ordre des bibles chrétiennes classiques, pour l'Ancien Testament selon la Septante. L'Ancien et le Nouveau Testaments forment deux sections.
<b>indexTOB</b>	les livres sont triés dans l'ordre de la <i>Traduction Œcuménique de la Bible</i> , pour l'Ancien Testament selon la bible hébraïque, suivie des deutérocanoniques. L'Ancien et le Nouveau Testaments ne sont pas séparés.
<b>indextestamentsT</b>	les livres sont triés dans l'ordre de la <i>Traduction Œcuménique de la Bible</i> , pour l'Ancien Testament selon la bible hébraïque, suivie des deutérocanoniques. L'Ancien et le Nouveau Testaments forment deux sections.
<b>indexBHS</b>	les livres de la bible hébraïque sont triés selon les bibles hébraïques classiques. La Torah, les Nebiim et les Ketouvim sont dans la même section. Les deutérocanoniques ainsi que les livres du Nouveau Testament ne sont pas triés.
<b>indexTanak</b>	les livres de la bible hébraïque sont triés selon les bibles hébraïques classiques. La Torah, les Nebiim et les Ketouvim sont dans trois sections distinctes. Les deutérocanoniques ainsi que les livres du Nouveau Testament ne sont pas triés.

Pour les options `indexTOB` et `indextestamentsT`, le livre d'Esther est considéré dans sa version hébraïque, et est donc rangé parmi les écrits.

Ainsi pour trier selon l'ordre de la TOB :

```
1 | \usepackage[indexTOB]{bibleref-french}
```

### 19.3.2 Un index à part

Par défaut, les références sont indexées dans l'index principal. Il est toutefois possible de les indexer séparément.

Il faut pour cela d'abord déclarer un nouvel index avec `imakeidx` (☞ p. 148, 18.2).

```
1 | \newindex[Index des références bibliques]{bible}
```

Il faut ensuite redéfinir la commande `\biblerefindex` de la manière suivante :

```
1 | \renewcommand{\biblerefindex}[0]{\index[bible]}
```

Et à l'endroit où l'on souhaite afficher cet index :

```
1 | \printindex[bible]
```

## 19.4 Espaces insécables

Le package *bibleref-french* prévoit d'insérer des espaces insécables :

1. Entre le nombre et le nom quand le nom d'un livre contient un nombre ;
2. Entre le nom du livre et le numéro de chapitre ;
3. Après un tiret demi-cadratin (–) dans un intervalle de versets ;
4. Après les virgules de séparation.

Si beaucoup de références sont indiquées, cela peut amener à des fins de lignes qui dépassent. Pour éviter cela, on peut autoriser temporairement la coupe pure des espaces entre un livre et un numéro de chapitre : `\BRallowhyphch` indique le début de l'autorisation ; `\BRforbidhyphch` en marque la fin.



# Chapitre 20

## Édition critique et textes parallèles

*Nous allons aborder ici deux packages particulièrement utiles en sciences humaines : ledmac pour faire des éditions critiques et ledpar (qui ne peut être utilisé séparément de ledmac) pour éditer un texte et sa traduction en parallèle.*

*Ces packages contiennent moult fonctionnalités : nous n'en présentons ici que quelques unes et renvoyons aux manuels pour les autres<sup>1</sup>.*

⚡ Ces packages ont été améliorés durant la rédaction de ce livre. Veillez à les mettre à jour (☞ p. 225, A.5). Dans le gestionnaire de packages, vous ne trouverez une entrée que pour *ledmac*, pas pour *ledpar*. En mettant à jour *ledmac* vous mettez toutefois aussi à jour *ledpar*.

⚡ Un certain nombre de fonctionnalités de ces packages nécessitent pour fonctionner correctement une double compilation.

### 20.1 Faire une édition critique avec *ledmac*

#### 20.1.1 Numérotter les lignes d'un texte

La première fonctionnalité utile de *ledmac* est de numérotter les lignes d'un texte. On commence par entourer celui-ci des commandes `\beginnumbering` et `\endnumbering`. Il faut ensuite procéder de deux façons différentes selon que le texte est en prose ou en vers.

---

<sup>1</sup>Peter WILSON, Herries PRESS et Maïeul ROUQUETTE, *ledmac : A presumptuous attempt to port EDMAC, TABMAC and EDSTANZA to LaTeX*; Peter WILSON, Herries PRESS et Maïeul ROUQUETTE, *Parallel typesetting for critical editions : the ledpar package*.

## Texte en prose

La numérotation des lignes commence, entre ces deux commandes, avec la commande `\pstart` et se termine avec la commande `\pend` :

```

1  \beginnumbering
2  \pstart %commencer la numérotation
3  Lorem ipsum dolor sit amet, consectetur adipiscing elit?
4  Morbi commodo; ipsum sed pharetra gravida!
5  Nullam sit amet enim. Suspendisse id : velit vitae ligula.
6  Aliquam erat volutpat.
7
8  Sed quis velit. Nulla facilisi. Nulla libero.
9  Quisque facilisis erat a dui.
10 Nam malesuada ornare dolor.
11 Cras gravida, diam sit amet rhoncus ornare,
12 erat elit consectetur erat, id egestas pede nibh eget odio.
13 \pend %terminer la numérotation
14 \endnumbering

```

Nous obtenons ainsi :

Lorem ipsum dolor sit amet, consectetur adipiscing elit? Morbi  
 commodo ; ipsum sed pharetra gravida ! Nullam sit amet enim. Sus-  
 penderit id : velit vitae ligula. Aliquam erat volutpat.  
 Sed quis velit. Nulla facilisi. Nulla libero. Quisque facilisis erat a dui.  
 5 Nam malesuada ornare dolor. Cras gravida, diam sit amet rhoncus  
 ornare, erat elit consectetur erat, id egestas pede nibh eget odio.

Tout texte qui n'est pas situé entre `\pstart` et `\pend` n'est pas numéroté. Cela peut donc être utile pour interrompre temporairement la numérotation :

```

1  \beginnumbering
2  \pstart %commencer la numérotation
3  Lorem ipsum dolor sit amet, consectetur adipiscing elit?
4  Morbi commodo; ipsum sed pharetra gravida!
5  Nullam sit amet enim. Suspendisse id : velit vitae ligula.
6  Aliquam erat volutpat.
7  \pend
8
9  Texte non numéroté
10
11 \pstart Sed quis velit. Nulla facilisi. Nulla libero.

```

```

12 | Quisque facilisis erat a dui.
13 | Nam malesuada ornare dolor.
14 | Cras gravida, diam sit amet rhoncus ornare,
15 | erat elit consectetur erat, id egestas pede nibh eget odio.
16 | \pend %terminer la numérotation
17 | \endnumbering

```

La numérotation recommence à zéro à chaque `\beginnumbering`. Il est toutefois possible d'indiquer de recommencer automatiquement la numérotation à chaque page via la commande `\lineation{page}`.

↗ Si vos sections numérotées sont très longues, il peut arriver l'erreur de compilation « TeX capacity exceeded ». Il faut alors fractionner vos sections en utilisant de temps en temps la commande `\pausenumbering` suivie de `\resumenumbering` au sein de la section numérotée<sup>2</sup>.

Cette séquence de commandes correspond *grosso-modo* à la séquence `\endnumbering\beginnumbering` mais sans ré-initialisation de la numérotation.

## Texte en vers

La numérotation dans les textes en vers est un peu plus complexe. Les commandes `\pstart` et `\pend` disparaissent. Il faut à la place découper son texte en strophes (« stanza »). Pour commencer la numérotation, nous insérons donc la commande `\stanza`. Chaque vers se termine par un `&`, et le dernier vers de la strophe par `\&`.

Il est nécessaire d'indiquer l'indentation de chaque vers de la strophe — faute de quoi on obtient le message « Missing number » — grâce à la commande `\setstanzaindents{⟨ $n_0, n_1, n_2, n_x$ ⟩}`. L'argument  $\langle n_0 \rangle$  indique l'indentation du rejet si le vers est trop long pour tenir sur une seule ligne ;  $\langle n_1 \rangle$  correspond au premier vers,  $\langle n_2 \rangle$  au deuxième et ainsi de suite.

```

1 | \beginnumbering
2 | \setstanzaindents{0,0,4,0,0,0,2,2,8}
3 | \let\endstanzaextra\bigbreak
4 | \stanza
5 | Berlin setz an.&
6 | Es speien die Geschäfte&
7 | die wackern Knaben und die Mädchen aus.&
8 | Jetzt kommt der Feierabend --- aber defte!&
9 | Wir springen nur noch eben rasch nach Haus.&
10 | Die großen Fraun sind ganz auf neu gemalen.&
11 | Wer wird heut abend wohl den Zimt bezahlen?&

```

<sup>2</sup>On peut aussi régler la configuration de TeX pour lui allouer plus de mémoire.

```

12 | Sie lächeln lieb. Das Auto summt heran.&
13 | Berlin setzt an.\&
14 |
15 | \stanza
16 | Berlin brummt auf.&
17 | Wo ich die Paare anseh :&
18 | Hier wird ein harter Dienst straff absolviert.&
19 | Ein Riesenrummel von Grünau bis Wannsee ---&
20 | und alles tadellos organisiert.&
21 | Um jeden Schnapstich fühlst du es bestätigt :&
22 | Marie stark Geld --- heute wird das Ding getätigt!&
23 | Die Spesen fest. Planmäßig der Verlauf ---&
24 | Berlin braust auf.\&
25 | \endnumbering

```

donne ainsi<sup>3</sup> :

```

      Berlin setzt an.
                Es speien die Geschäfte
die wackern Knaben und die Mädchen aus.
Jetzt kommt der Feierabend — aber defte !
5  Wir springen nur noch eben rasch nach Haus.
      Die großen Fraun sind ganz auf neu gemalen.
      Wer wird heut abend wohl den Zimt bezahlen ?
      Sie lächeln lieb. Das Auto summt heran.
                                Berlin setzt an.

10  Berlin brummt auf.
                Wo ich die Paare anseh :
      Hier wird ein harter Dienst straff absolviert.
      Ein Riesenrummel von Grünau bis Wannsee —
      und alles tadellos organisiert.
15  Um jeden Schnapstich fühlst du es bestätigt :
      Marie stark Geld — heute wird das Ding getätigt !
      Die Spesen fest. Planmäßig der Verlauf —
                                Berlin braust auf.

```

Si l'indentation se répète tous les  $n$  vers, on peut assigner la valeur  $n$  au compteur (☞ p. 138, 17.2) **stanzaindentrepitition** pour ne pas avoir à répéter l'indentation dans la commande `\setstanzaindent`.

<sup>3</sup>Kurt TUCHOLSKY, *Berliner Abend*.

Ainsi :

```
1 | \setstanzaindents{12,1,0}
2 | \setcounter{stanzaindentsrepetition}{2}
```

est équivalent à :

```
1 | \setstanzaindents{12,1,0,1,0,1,0,1,0,1,0...}
```

La commande `\endstanzextra` permet d'ajouter un contenu à la fin de chaque strophe. Dans l'exemple qui précède, l'espace vertical entre chaque strophe est ainsi obtenu par la ligne `\let\endstanzextra\bigbreak`.

La commande `\bigbreak` insère un espace vertical ; `\let\cmd1\cmd2` est une commande T<sub>E</sub>X qui copie la commande `\cmd2` dans `\cmd1`.

 Pour obtenir un crochet indiquant les rejets de vers, selon la typographie française traditionnelle, il faut redéfinir la commande `\hangingsymbol`.

```
1 | \renewcommand{\hangingsymbol}{\,}
```

La commande `\,` sert à insérer une espace fine insécable.

Si vos vers ne possèdent pas d'indentation, mettez `0` partout dans la commande `\setstanzaindents`, sauf pour l'indentation de rejet. S'il y a moins de vers qu'indiqué dans `\setstanzaindents`, cela n'est pas problématique, l'inverse en revanche produit une erreur de compilation.

## 20.1.2 Aller plus loin

### Numéroter les paragraphes

Il est possible d'insérer un nombre à chaque `\pstart` en utilisant la commande `\numberpstarttrue`, ce qui permet ainsi de numéroter les paragraphes. À chaque commande `\beginnumbering`, la numérotation recommence.

1. Lorem ipsum dolor sit amet, consectetur adipiscing elit? Morbi commodo; ipsum sed pharetra gravida! Nullam sit amet enim. Suspendisse id: velit vitae ligula. Aliquam erat volutpat.
2. Sed quis velit. Nulla facilisi. Nulla libero. Quisque facilisis erat
- 5 a dui. Nam malesuada ornare dolor. Cras gravida, diam sit amet rhoncus ornare, erat elit consectetur erat, id egetas pede nibh eget odio.

Pour arrêter la numérotation des `\pstarts`, utilisez `\numberpstartfalse`.

☞ Si chaque `\pstart` correspond à un paragraphe, il est possible d'automatiser l'insertion des `\pstarts` et `\pends` en insérant la commande `\autopar` après `\beginnumbering`.

☞ La numérotation se fait à l'aide d'un compteur `pstart`. Nous expliquons plus loin ce qu'est un compteur, comment en modifier l'apparence (☞ p. 201, 23.3.1) et la valeur (☞ p. 208, 23.5). Avec ces informations vous pourrez aisément changer la numérotation des `\pstarts`.

☞ Il est possible de recommencer la numérotation des lignes à chaque `\pstart` via `\lineation{pstart}`.

## Fréquence de numérotation des lignes

Comme on peut le voir dans les exemples précédents, par défaut *ledmac* numérote une ligne sur cinq. Nous pouvons changer cela très simplement avec les commandes `\firstlinenum{<nbre>}` et `\linenumincrement{<nbre>}`.

L'argument *<nbre>* de la commande `\firstlinenum` indique quelle est la première ligne qui sera numérotée ; celui de la commande `\linenumincrement` définit la fréquence avec laquelle les lignes seront numérotées. Ces commandes se placent après `\beginnumbering`.

Ainsi, si nous souhaitons numéroter toutes les lignes en commençant par la première, il faut mettre :

```
1 | \linenumincrement{1}
2 | \firstlinenum{1}
```

## Changer le numéro de certaines lignes

La commande `\setline{<num>}` permet de changer le numéro d'une ligne. La numérotation des lignes suivantes continue à partir de ce nombre ; la commande `\advanceline{<num>}` permet d'ajouter *<num>* au numéro de la ligne.

La commande `\skipnumbering` permet de ne pas prendre en compte la ligne courante dans la numérotation.

## Sous-numéro

Il est possible d'indiquer des lignes comme « sous-lignes », c'est-à-dire de stopper temporairement la numérotation et de démarrer une sous-numérotation.

La sous-numérotation commence avec `\startsub` et finit avec `\endsub`. Si la première commande se situe au milieu d'une ligne, la sous-numérotation commence à la ligne suivante.

Les commandes `\subfirstlinenum{<nbre>}` et `\sublinenumincrement{<nbre>}` permettent de régler la première sous-ligne numérotée et la fréquence de sous-numérotation.

```

1  \firstlinenum{1}
2  \linenumincrement{1}
3  \firstsublinenum{1}
4  \sublinenumincrement{1}
5  \beginnumbering
6  \pstart
7  Lorem\startsub ipsum dolor sit amet,
8  consectetur adipiscing elit?
9  Morbi commodo; ipsum sed pharetra gravida!
10 Nullam sit amet enim. Suspendisse id : velit vitae ligula.
11 Aliquam erat volutpat.\endsub
12
13
14 Sed quis velit. Nulla facilisi. Nulla libero.
15 Quisque facilisis erat a dui.
16 Nam malesuada ornare dolor.
17 Cras gravida, diam sit amet rhoncus ornare,
18 erat elit consectetur erat, id egestas pede nibh eget odio.
19 \pend
20 \endnumbering

```

```

1  Loremipsum dolor sit amet, consectetur adipiscing elit? Morbi
1.1 commodo; ipsum sed pharetra gravida! Nullam sit amet enim. Sus-
1.2 pendisse id : velit vitae ligula. Aliquam erat volutpat.
2  Sed quis velit. Nulla facilisi. Nulla libero. Quisque facilisis erat a dui.
3  Nam malesuada ornare dolor. Cras gravida, diam sit amet rhoncus
4  ornare, erat elit consectetur erat, id egestas pede nibh eget odio.

```

## Style et position des numéros des lignes

Les numéros des lignes apparaissent par défaut dans la marge de gauche. La commande `\linenummargin{⟨arg⟩}` redéfinit cette position. L'argument peut prendre la valeur :

<b>left</b>	qui est la valeur par défaut ;
<b>right</b>	pour avoir les numéros à droite ;
<b>inner</b>	pour les avoir dans la marge intérieure ;
<b>outer</b>	pour les avoir dans la marge extérieure.

Il existe cinq styles pour numérotter les lignes, sélectionnés par la commande `\linenumberstyle{⟨style⟩}` et par `\sublinenumberstyle{⟨style⟩}` pour la sous-numérotation :

<b>arabic</b>	les numéros des lignes sont en chiffres arabes, c'est le style par défaut ;
<b>Alph</b>	pour des lettres majuscules ;
<b>alph</b>	pour des lettres minuscules ;
<b>Roman</b>	pour des chiffres romains majuscules ;
<b>roman</b>	pour des chiffres romains minuscules.

La distance `\linenumsep` représente la distance (☞ p. 237, E) entre le texte et le numéro de la ligne, par défaut, 1 pica. Elle est modifiable par la commande : `\setlength{\linenumsep}{⟨distance⟩}` (☞ p. 210, 24.2).

De la même façon nous pouvons modifier, via `\renewcommand`, `\numlabfont`, qui indique la taille du numéro et qui est ainsi définie par *ledmac* :

```
1 | \newcommand{\numlabfont}{\normalfont\scriptsize}
```

### 20.1.3 L'apparat critique

Les notes de bas de page se présentent de cette façon : `\Afootnote{⟨texte⟩}`. Il en existe cinq sortes, de `\Afootnote` à `\Efootnote`.

Ce type de notes appelle un « lemme », c'est-à-dire une partie du texte annoté reprise dans la note de bas de page, précédée du numéro de la ou des lignes correspondantes.

La note de bas de page est appelée ainsi : `\edtext{⟨lemme⟩}{⟨commande⟩}`. L'argument `⟨commande⟩` contient l'appel à la note : `\Afootnote`, `\Bfootnote`, etc.

Il est aussi possible d'utiliser des notes de bas de page « normales », appelées par un numéro dans le texte. Il en existe là aussi cinq sortes : de `\footnoteA` à `\footnoteE`.

Dans les notes de type `\Afootnote...`, lorsque le « lemme » est trop grand, nous pouvons l'abrégé, via : `\edtext{⟨le lemme trop long⟩}{\lemma{⟨le lemme raccourci⟩}\Bfootnote{⟨la note⟩}}`.

☞ Il est aussi possible de mettre des notes de fin. Là encore, il y en a cinq sortes : de `\Aendnote` à `\Eendnote`.

☞ La commande `\marginpar` ne fonctionne pas dans les textes numérotés. À la place, le package propose les commandes `\ledleftnote{⟨note⟩}` et `\ledrightnote{⟨note⟩}` pour des notes de marge gauche et droite.

☞ Il existe aussi une commande `\ledsidenote{⟨note⟩}` qui insère par défaut la note dans la marge de droite, mais `\sidenotemargin{left}` permet de dire de l'insérer à gauche.

Dans l'exemple suivant, nous avons utilisé deux types de notes avec lemme, dont l'une où nous avons mis un lemme « raccourci », et un type de note sans lemme<sup>4</sup> :

```

1  \beginnumbering
2  \setstanzaindents{0,0,1,0,1,0,1,0,1}
3  \stanza
4  Quando erit illa dies, cum nostrum intrabis in \edtext{ortum}
5  {\Afootnote{hortum, \textit{F}}},&
6  Atque leges nostras ungue libente rosas?\footnoteA{In exilio}&
7  Et tua magna sitis mage seu \edtext{mage crescere}
8  {\Afootnote{mage se accrescere \textit{G}}}, gliscet,&
9  Dum quod semper amas, carmine plenus eris.&
10 \edtext{Si qua istis fuerint, ut erunt, uitiosa camenis}
11 {\lemma{Si \dots camenis}\Bfootnote{\textit{0v. Trist. IV}},
12 1 Si qua meis fuerint, ut erunt, vitiosa libellis, parce}},&
13 Parce, precor : scriptor non mihi doctus inest.&
14 Quaeso, tuum nobis fidum trans mitte ministrum,&
15 Qui tua grata mihi perferat orsa, uale.\&
16 \endnumbering

```

1 Quando erit illa dies, cum nostrum intrabis in ortum,  
2 Atque leges nostras ungue libente rosas ?<sup>1</sup>  
3 Et tua magna sitis mage seu mage crescere gliscet,  
4 Dum quod semper amas, carmine plenus eris.  
5 Si qua istis fuerint, ut erunt, uitiosa camenis,  
6 Parce, precor : scriptor non mihi doctus inest.  
7 Quaeso, tuum nobis fidum trans mitte ministrum,  
8 Qui tua grata mihi perferat orsa, uale.

---

1 ortum ] hortum, *F*  
3 mage crescere ] mage se accrescere *G*

---

5 Si ...camenis ] *Ov. Trist. IV*, 1 Si qua meis fuerint, ut erunt, vitiosa libellis, parce

---

<sup>1</sup>In exilio

<sup>4</sup>THÉODULF D'ORLÉANS, *Carmen LXXII*, éd. Ernst DÜMMLER, *Monumenta Germaniae Historica, Poetae latini aevi Carolini*, Berlin, 1881, p. 437–581.

⚡ Il faut compiler deux fois pour obtenir le numéro des lignes dans l'apparat critique.

Pour plus de facilité vous pouvez changer le nom des types de notes en indiquant par exemple dans son préambule :

```

1 | \let\variantes\Afootnote
2 | \let\citations\Bfootnote
3 | \let\éclaircissements\footnoteA
4 | Et ainsi de suite...

```

Mais il est aussi conseillé de créer des commandes personnelles pour simplifier — et surtout clarifier — plus encore la rédaction d'un apparat critique ; par exemple :

```

1 | \usepackage{ifthen, xargs}
2 | \newcommand*{\variantes}[3][2]{%
3 |   \edtext{#1}{%
4 |     \ifthenelse{\equal{#2}{}}{\Lemma{#2}}%
5 |   \Afootnote{#3}}%
6 | }

```

Cette commande est un peu plus complexe parce qu'elle fait appel aux packages *xargs* et *ifthen*.

La commande `\newcommandx`, fournie par *xargs*, permet de créer une commande avec des arguments optionnels. Ici, la commande créée appelle trois arguments : le premier argument est le lemme ; le deuxième argument, optionnel — ce qu'indique, dans notre code, l'argument [2] — contient le lemme raccourci ; enfin le dernier argument contient l'annotation.

Ligne 4, la commande `\ifthenelse⟨test⟩⟨sioui⟩⟨sinon⟩` du package *ifthen* permet d'effectuer un test : ici nous regardons si le deuxième argument est vide (`\equal{#2}{}`) : si ce n'est pas le cas, alors nous l'utilisons comme argument d'une commande `\lemma`.

L'utilisation de la commande que nous venons de créer peut donc se résumer ainsi : `\variantes{⟨lemme⟩}[⟨lemme plus court⟩]{⟨annotation⟩}`.

Ce qui est plus simple que : `\edtext{⟨le lemme trop long⟩}{\lemma{⟨le lemme raccourci⟩}\Bfootnote{⟨la note⟩}}`

👉 Les packages *ifthen* et *xargs* sont extrêmement utiles, mais nous ne pouvons détailler ici leurs utilisation. Nous renvoyons à leurs documentations<sup>5</sup>.

<sup>5</sup>CARLISLE, *The ifthen package, op. cit.* ; PÉGOURIÉ-GONNARD, *L'extension xargs, op. cit.*

## 20.2 Mettre deux textes en vis-à-vis : le package *ledpar*

Le package *ledpar*, qui sert à mettre un texte et sa traduction en vis-à-vis, fonctionne avec *ledmac*. Tout ce qui concerne l'apparat critique et la numérotation dans *ledmac* est ainsi valable ici aussi. Nous considérons donc, pour éviter les redites, que vous avez lu ce qui précède.

### 20.2.1 Principes

Le package *ledmac* permet d'obtenir très facilement deux textes en vis-à-vis ; c'est donc un package particulièrement utile lorsque l'on veut faire de l'édition bilingue.

Son principe est simple : des textes numérotés — écrits entre `\beginnumbering` et `\endnumbering` — sont placés dans les environnements **Leftside** pour le texte de gauche, **Rightside** pour le texte de droite.

Le package *ledpar* fait correspondre en vis-à-vis chaque « boîte » d'un texte à la boîte correspondante de l'autre texte : il faut donc qu'il y ait le même nombre de boîtes des deux côtés. Pour la poésie, chaque vers (terminé par `&`), est une boîte ; en prose, les boîtes sont délimitées par `\pstart` et `\pend`.

Il est recommandé, en prose, de mettre chaque paragraphe dans une boîte — pour obtenir une synchronisation la plus fine possible — donc de débiter chaque paragraphe par `\pstart` et de le finir par `\pend`, ou d'utiliser la commande `\autopar` (☞ p. 170, 20.1.2).

Le package *ledpar* peut mettre les textes en vis-à-vis sur deux colonnes ou sur deux pages : bien sûr dans ce dernier cas le texte de gauche est toujours imprimé sur une page paire. Nous utilisons dans le premier cas l'environnement **pairs** ; l'environnement **pages** dans le second.

Les commandes `\Columns` ou `\Pages`, selon qu'il s'agisse de l'environnement **pairs** ou **pages**, indiquent à  $\text{\LaTeX}$  d'imprimer les boîtes jusqu'ici gardées en mémoire.

Résumons !

Pour mettre deux textes en vis-à-vis sur deux pages :

```

1 | \begin{pages} % Des textes en vis-à-vis sur deux pages
2 | \begin{Leftside} % Texte de la page de gauche
3 | \beginnumbering % Début de la numérotation
4 | \pstart % Ici, prose : donc \pstart et \pend
5 |
6 | Le texte dans une langue
7 |
8 | \pend
9 | \endnumbering % Fin de la numérotation

```

```

10 \end{Leftside} % Fin du texte de gauche
11
12 \begin{Rightside} % La même chose à droite
13 \beginnumbering
14 \pstart
15
16 Le texte dans une autre langue
17
18 \pend
19 \endnumbering
20 \end{Rightside}
21 \Pages
22
23 \end{pages} % Fin la composition en parallèle.

```

⚡ Si l'un des textes n'est pas dans la langue principale du document, il faut préciser le changement de langue (☞ p. 33, 4) à l'intérieur de l'environnement **Leftside** ou **Rightside**.

```

1 \begin{Leftside}
2 \begin{<langue>}
3 \end{<langue>}
4 \end{Leftside}

```

⚡ Comme avec *ledmac*, la numérotation (☞ p. 167, 20.1.1) recommence à chaque `\beginnumbering`. On peut toutefois faire recommencer la numérotation à chaque page via les commandes `\lineation` et `\lineationR`, qui ont la même syntaxe, vue plus haut.

Nous obtenons parfois le message d'erreur « Too many `\pstart` without printing ». Pas de panique ! Cela signifie qu'il y a trop de boîtes en mémoire :  $\TeX$  doit en imprimer quelques-unes avant de pouvoir continuer. Il y a deux solutions possibles à ce problème.

La première solution consiste à augmenter le nombre de boîtes que *ledpar* peut garder en mémoire, en utilisant dans le préambule `\maxchunks{<nbre>}`, où *<nbre>* est le nombre de boîtes en mémoire. Par défaut, ce nombre est de dix. Il suffit donc de l'augmenter pour qu'il soit égal ou supérieur au nombre de boîtes de son texte.

Toutefois  $\TeX$  ne dispose pas d'une place illimitée pour les boîtes, c'est pourquoi il est souvent préférable d'utiliser la seconde solution.

Celle-ci, que nous pouvons d'ailleurs combiner avec la première, consiste à couper le texte en plusieurs grands ensembles de boîtes — c'est-à-dire en plusieurs environnements **Leftside** et **Rightside**, se terminant chacun par `\Columns` ou `\Pages`. Cette solution est préférable pour les longs textes. Les divisions internes du texte se reflètent alors dans les environnements **Leftside** et **Rightside**. Cependant, en mettant simplement :

```

1  \begin{pages}
2  \begin{Leftside} \beginnumbering
3  La première partie du texte en langue originale
4  \endnumbering \end{Leftside}
5
6  \begin{Rightside} \beginnumbering
7  La traduction de la première partie
8  \endnumbering \end{Rightside}
9  \Pages
10
11 \begin{Leftside} \beginnumbering
12 La deuxième partie du texte
13 \endnumbering \end{Leftside}
14
15 \begin{Rightside} \beginnumbering
16 La traduction de la deuxième partie
17 \endnumbering \end{Rightside}
18 \Pages
19
20 \end{pages}

```

nous nous retrouvons avec une numérotation qui recommence à zéro pour la deuxième partie du texte.

La commande `\memorydump` permet de résoudre ce problème. Elle équivaut en fait à un `\pausenumbering` immédiatement suivi d'un `\resumenumbering`. Il faut donc mettre :

```

1  \begin{pages}
2  \begin{Leftside} \beginnumbering
3  La première partie du texte en langue originale
4  \end{Leftside} % pas de \endnumbering ici!...
5
6  \begin{Rightside} \beginnumbering
7  La traduction de la première partie
8  \end{Rightside} % et là non plus...
9  \Pages
10
11 \begin{Leftside} \memorydump % et, ici, pas de \beginnumbering..
12 La deuxième partie du texte
13 \endnumbering \end{Leftside}
14
15 \begin{Rightside} \memorydump %...car la numérotation continue!
16 La traduction de la deuxième partie

```

```

17 | \endnumbering \end{Rightside} % Ne pas oublier de mettre
18 | % tout de même \endnumbering quand on arrive à la fin ..
19 |
20 | \Pages
21 |
22 | \end{pages}

```

Ainsi, la numérotation du texte se fait en continue entre la première partie et la deuxième partie.

✎ Il est bien sûr possible de mettre en parallèle un texte en vers et sa traduction en prose.

✎ Si vous utilisez la commande `\numberpstarttrue` (☞ p. 169, 20.1.2) pour numéroter les `\pstarts`, sachez que la numérotation gauche et la numérotation droite fonctionnent avec deux compteurs différents : `pstartL` et `pstartR` (☞ p. 201, 23.3.1 ; ☞ p. 208, 23.5).

## 20.2.2 Affiner la présentation

### Vers de longueurs inégales

Si des vers sur des pages parallèles prennent un nombre de lignes inégales, par exemple si le vers de gauche contient un rejet et non celui de droite, alors `ledpar` introduit un blanc après le vers le droite, pour que le vers suivant soit exactement au même niveau dans la page de droite et dans la page de gauche.

Il est possible de dire à `ledpar` de ne pas introduire ce blanc, mais de produire un léger décalage des vers, décalage rattrapé à chaque changement de pages. Pour ce faire, il faut passer l'option `shiftedverses` au chargement du package.

```

1 | \usepackage[shiftedverses]{ledpar}

```

### Tailles et séparateurs des colonnes

Par défaut, les colonnes font 45% d'une largeur normale de texte. Nous pouvons modifier ce réglage en redéfinissant les longueurs `\Lcolwidth` et `\Rcolwidth` respectivement pour la colonne gauche et droite. Nous utilisons pour cela la commande `\setlength` (☞ p. 210, 24.2 ; ☞ p. 238, E.2). Par exemple pour avoir 47% de la largeur normale d'un texte, il faut mettre :

```

1 | \setlength{\Lcolwidth}{0.47\textwidth}
2 | \setlength{\Rcolwidth}{0.47\textwidth}

```

De même nous pouvons définir un filet de séparation grâce à la longueur `\columnrulewidth` (☞ p. 237, E).

```

1 | \setlength{columnrulewidth}{0.4pt}

```

## Numérotation des lignes de droite

Par défaut, les numéros des lignes de droite sont suivies d'un « R », correspondant au terme « Right ». Pour obtenir à la place un « D », il suffit de redéfinir la commande `\Rlineflag` :

```
1 | \renewcommand{\Rlineflag}{D}
```

Si nous ne souhaitons rien, mettre tout simplement :

```
1 | \renewcommand{\Rlineflag}{}
```



# Chapitre 21

## Présentations avec L<sup>A</sup>T<sub>E</sub>X : Beamer

Si L<sup>A</sup>T<sub>E</sub>X permet d'écrire du texte, il offre également la possibilité de créer facilement des diapositives de présentation pour un exposé ou une conférence<sup>1</sup>. Ceci se fait à l'aide d'une classe spécifique : **beamer**.

Nous présenterons ici les bases de Beamer afin de bien préparer des supports de présentation. Nous renvoyons à d'autres ouvrages pour plus de détails<sup>2</sup>.

Le PDF résultant de la compilation d'un fichier Beamer peut être affiché sous forme de diaporama contrôlable à l'aide des logiciels classiques de lecture de PDF.

### 21.1 Préambule et premières diapositives

Beamer étant une classe L<sup>A</sup>T<sub>E</sub>X, le document doit commencer par :

```
1 | \documentclass{beamer}
```

Le reste du préambule contient le même contenu qu'un fichier .tex classique : appels aux packages, définitions d'éventuelles commandes, etc.

On peut définir la manière dont les diapositives vont s'afficher, en utilisant dans le préambule la commande `\settheme`. Par exemple, pour utiliser le thème « Darmstadt » :

```
1 | \settheme{Darmstadt}
```

---

<sup>1</sup>Pour remplacer Microsoft Powerpoint ou LibreOffice Impress.

<sup>2</sup>On consultera pour une introduction plus approfondie Marie-Claude COLLILIEUX, *Découverte de Beamer*, 2007, URL : <http://mccclnews.free.fr/latex/introbeamer.php> ; pour les usages avancées on se servira du manuel : Till TANTAU, Joseph WRIGHT et Vedran MILETIC, *The beamer class*.

Un certain nombre de thèmes sont décrits dans le manuel de Beamer<sup>3</sup>.

Avec Beamer, l'unité de base est la diapositive, appelée *frame*. Une diapositive se crée avec l'environnement *frame* et peut contenir n'importe quel contenu  $\LaTeX$ . Par exemple :

```

1 \begin{frame}
2   \begin{itemize}
3     \item Élément 1
4     \item Élément 2
5     \item Élément 3
6     \item Etc.
7   \end{itemize}
8 \end{frame}

```

À noter que l'environnement `frame` peut être remplacé par la commande homonyme :

```

1 \frame{
2   \begin{itemize}
3     \item Élément 1
4     \item Élément 2
5     \item Élément 3
6     \item Etc.
7   \end{itemize}
8 }

```

## 21.2 Diapositives de titre

Il est possible d'utiliser dans le document Beamer les commandes `\section` et `\subsection` (☞ p. 29, 3.1)<sup>4</sup> et d'utiliser dans le préambule les commandes de description : `\title`, `\author` et `\date` (☞ p. 23, 2.2.5).

Ces informations servent toutefois uniquement à structurer le document, afin de pouvoir afficher le plan. C'est pourquoi il faut les entrer *en-dehors* des diapositives.

Pour les afficher dans une diapositive, en revanche, il faut utiliser les commandes `\sectionpage` et `\subsectionpage` pour les niveaux de titres ; `\titlepage` pour les informations sur la date, l'auteur et le titre de la conférence.

```

1 \documentclass{beamer}
2 ...

```

<sup>3</sup>Till TANTAU, Joseph WRIGHT et Vedran MILETIC, « Themes », dans : *The beamer class*.

<sup>4</sup>Le manuel de `beamer` déconseille d'utiliser `\subsubsection` parce que « c'est le diable ». Cf. : Till TANTAU, Joseph WRIGHT et Vedran MILETIC, dans : *The beamer class*.

```

3  \title{Titre de l'exposé}
4  \author{Nom de son auteur}
5  \date{Date}
6  \begin{document}
7
8  \frame{\titlepage}
9
10 \section{Titre de section}
11 \frame{\sectionpage}
12
13 \subsection{Titre de sous-section}
14 \frame{\subsectionpage}
15 \end{document}

```

## 21.3 Affichage différé : les couches Beamer

Afficher tous les éléments d'une diapositive en une seule fois présente peu d'intérêt : il vaut mieux les afficher petit à petit afin de les commenter au fur et à mesure. Beamer utilise pour cela la notion de « slides », que nous traduisons par celle de « couches ». Il s'agit de dire à une commande de ne s'exécuter que sur une ou plusieurs couches spécifiques.

La syntaxe de l'affichage couche par couche (en anglais *overlay specification*) est : <a-b> où a désigne la couche de départ et b désigne la couche d'arrivée. On peut également séparer les numéros de couches spécifiques par des virgules. Voici des exemples pour mieux comprendre :

```

1  \begin{frame}
2  \begin{itemize}
3      \item<1>Élément affiché uniquement sur la première couche.
4      \item<2->Élément affiché à partir de la deuxième couche.
5      \item<2-4>Élément affiché sur les couches 2 à 4.
6      \item<-4>Élément affiché jusqu'à la couche 4, incluse.
7      \item<1,3-5>Élément affiché sur les couches 1 et 3 à 5.
8  \end{itemize}
9  \end{frame}

```

Toutes les commandes  $\LaTeX$  ne réagissent pas nécessairement aux spécifications de couche : par exemple la commande `\emph` ne réagit pas : si vous utilisez une spécification de couche avec cette commande, la spécification apparaît sur la diapositive.

On peut vouloir obtenir un PDF où toutes les couches d'une diapositive apparaissent d'un seul coup, afin par exemple d'avoir une version papier : il suffit d'utiliser l'option de package `handout`.

```
1 | \documentclass[handout]{beamer}
```

 Il est possible de fabriquer un PDF destiné à l'impression sur transparents — en cas de défaillance du vidéo-projecteur — en précisant quelles couches doivent être séparées sur les transparents : nous renvoyons au manuel<sup>5</sup>.

Beamer propose des commandes et des environnements pour des usages avancés des spécifications de couche, notamment pour remplacer des éléments par d'autres éléments, tout en accordant à chacun la même place sur la diapositive, pour éviter d'avoir des « effets de déplacement » d'un élément d'une couche à l'autre. Ceci dépassant cette introduction, nous renvoyons au manuel<sup>6</sup>.

## 21.4 Mise en valeur

### 21.4.1 Blocs

Il existe plusieurs manières de mettre en valeur certains éléments d'une diapositive. La première consiste à les placer dans des blocs pour les encadrer. Beamer propose plusieurs blocs standards, qui prennent à la fois un titre et un contenu. Ces blocs sont : bloc standard, bloc d'exemple et bloc d'alerte.

```
1 | \begin{block}{Un bloc normal}
2 |   Un texte à encadrer
3 | \end{block}
4 | \begin{alertblock}{Attention}
5 |   Ce texte est important
6 | \end{alertblock}
7 | \begin{exampleblock}{Exemple}
8 |   Un exemple des propos tenus
9 | \end{exampleblock}
```

 Il est possible de créer ses propres blocs, le manuel explique comment<sup>7</sup>. Il existe aussi des éléments proches des blocs que sont les encadrés, pour mettre en valeur des définitions ou des théorèmes<sup>8</sup>.

<sup>5</sup>Till TANTAU, Joseph WRIGHT et Vedran MILETIC, « Creating Transparencies », dans : *The beamer class*.

<sup>6</sup>Till TANTAU, Joseph WRIGHT et Vedran MILETIC, « Dynamically Changing Text or Images », dans : *The beamer class*.

<sup>7</sup>Till TANTAU, Joseph WRIGHT et Vedran MILETIC, « Framed and Boxed Text », dans : *The beamer class*.

<sup>8</sup>Till TANTAU, Joseph WRIGHT et Vedran MILETIC, « Theorem Environments », dans : *The beamer class*.

Évidemment, il est possible de spécifier un niveau de couche ; la spécification doit alors se trouver après le titre du bloc :

```

1 | \begin{alertblock}{Attention}<2>
2 |   Un message d'alerte sur la couche 2.
3 | \end{alertblock}

```

L'apparence des blocs dépend du thème choisi.

### 21.4.2 La commande alert

La commande `\alert` sert à mettre en valeur une partie d'un texte. Elle peut recevoir une spécification de couche. Cela est très utile pour indiquer à l'auditoire le sujet dont on parle.

```

1 | \begin{frame}
2 | \begin{description}
3 |   \item\alert<1>{Nous parlons d'abord d'une chose}
4 |   \item\alert<2>{Nous parlons ensuite d'une autre}
5 |   \item\alert<3>{Et enfin d'une troisième}
6 | \end{description}
7 | \end{frame}

```



Le package *spot* permet des mises en valeur sous forme de taches lumineuses.

## 21.5 Notes de conférence

Il peut être utile pour le conférencier d'avoir, en plus des diapositives, des notes lui servant d'aide-mémoire au cours de son exposé.

On crée une note avec la commande `\note` dans une diapositive.

```

1 | \begin{frame}
2 | \note{Un élément à se rappeler pour le signaler au public}.
3 |
4 | Le contenu proprement dit de la diapositive.
5 | \end{frame}

```

On peut afficher plusieurs notes sous forme de liste en passant à la commande `\note` un argument optionnel ayant pour valeur `item`.

```

1 | \begin{frame}
2 | \note[item]{Un élément à se rappeler pour le signaler au public}.
3 | \note[item]{Un autre élément}

```

```

4 |
5 | Le contenu proprement dit de la diapositive.
6 | \end{frame}

```

Les notes n'apparaissent pas sur la diapositive, mais sont reportées après, dans une page spéciale du fichier PDF. Cela n'est pas très pratique : on souhaiterait avoir d'une part les diapositives, d'autre part les notes.

On obtient ce résultat en plaçant `\setbeameroption` dans le préambule, avec le bon argument :

```

1 | \setbeameroption{hide notes}

```

signifie que nous produisons un PDF contenant uniquement les diapositives. *A contrario* :

```

1 | \setbeameroption{show only notes}

```

indique que nous désirons un fichier ne contenant que les notes.



Il est également possible de produire un fichier contenant les diapositives et les notes, mais capable, avec le bon logiciel, de les envoyer sur deux écrans différents. Nous renvoyons au manuel pour plus de détails<sup>9</sup>.

## 21.6 Écrire son article dans le même fichier que la présentation

Il peut être judicieux d'écrire l'article correspondant à la présentation dans le même fichier que celui-ci. Il suffit pour cela d'écrire le contenu de l'article entre les environnements (ou les commandes) `frame` — mettre les titres des sections hors des diapositives révèle ici son avantage : ces titres sont aussi ceux de l'article.

```

1 | \section{Introduction}
2 |
3 | Introduction de notre article
4 | \begin{frame}
5 |     Diapositive pour l'introduction de la présentation
6 | \end{frame}
7 |
8 | \section{Première partie}
9 |

```

---

<sup>9</sup>Till TANTAU, Joseph WRIGHT et Vedran MILETIC, « Taking Advantage of Multiple Screens », dans : *The beamer class*.

```

10 | Texte de la première partie
11 | \begin{frame}
12 |     Diapositive pour la première partie
13 | \end{frame}
14 |
15 | etc.
```

 On peut aussi introduire des miniatures des diapositives dans l'article : consultez le manuel<sup>10</sup>.

### 21.6.1 Diapositives ou article ?

Par défaut, Beamer va afficher les textes situés entre les diapositives dans des diapositives spécifiques. Pour éviter cela, on passe l'option `ignorenonframetext` lors de l'appel de la classe.

```
1 | \documentclass[ignorenonframetext]{beamer}
```

Pour obtenir uniquement la version article, il suffit de changer la classe du document et de charger le package *beamerarticle*.

```

1 | \documentclass{article}
2 | \usepackage{beamerarticle}
```

### 21.6.2 Automatiser les réglages

Plutôt que de changer systématiquement les premières lignes du document, autant utiliser les possibilités d'inclusion (☞ p. 30, 3.2) de fichiers offertes par L<sup>A</sup>T<sub>E</sub>X.

Pour cela, créer trois fichiers : `article.tex`, `presentation.tex` et `contenu.tex`. Dans `article.tex` mettre :

```

1 | \documentclass{article}
2 | \usepackage{beamerarticle}
3 | \input{contenu}
```

Dans `presentation.tex` mettre :

```

1 | \documentclass[ignorenonframetext]{beamer}
2 | \input{contenu}
```

Dans `contenu.tex` mettre les appels aux packages communs aux deux versions ainsi que le contenu de la présentation / de l'article.

<sup>10</sup>Till TANTAU, Joseph WRIGHT et Vedran MILETIC, « Including Slides from the Presentation Version in the Article Version », dans : *The beamer class*.



## Cinquième partie

# Quelques éléments de mise en forme

« La typographie est simple, aussi simple que de jouer du violon ! »

Maximilien Vox, typographe



# Chapitre 22

## Personnaliser en-têtes et pieds de pages

*Nous allons voir comment personnaliser les en-têtes et pieds de pages à l'aide du package fancyhdr.*

### 22.1 Utiliser l'un des styles standards

LaTeX propose quatre styles standards de page, qui définissent le contenu des en-têtes et pieds de pages. Pour utiliser l'un de ces styles, on emploie la commande `\pagestyle{<style>}`.

On peut également modifier le style d'une page en particulier grâce à la commande `\thispagestyle{<style>}`.

Ces styles sont :

<b>empty</b>	pas d'en-tête ni de pied de page.
<b>plain</b>	pas d'en-tête mais pieds de pages contenant le numéro de page en centré. C'est le style correspondant aux pages de début de chapitre (☞ p. 205, 23.4).
<b>heading</b>	pas de pied, en-têtes contenant le titre du chapitre, de la section ou de la sous-section et le numéro de la page. C'est le style par défaut.
<b>myheading</b>	similaire au précédent, mais l'en-tête peut être personnalisé <sup>1</sup> .

---

<sup>1</sup>Nous tenons cette information du manuel de Franck MITTELBACH et Michel GOOSSENS, *LaTeX Companion*, 2<sup>e</sup> éd., Paris : Pearson, 2006, URL : <http://www.pearson.fr/livre/?GCOI=27440100048330>, p. 222, qui ne précise pas comment faire.

On voit vite les limites de ces styles. Ainsi, comment avoir à la fois les titres en en-tête et les numéros des pages en pied de page ? Comment indiquer son nom ou la date en pied de page ?

## 22.2 Premiers exemples avec *fancyhdr*

Le package *fancyhdr* propose un autre style, qu'il est aisé de personnaliser via des commandes spécifiques. Pour faire fonctionner *fancyhdr* il faut écrire les lignes suivantes dans le préambule :

```
1 | \usepackage{fancyhdr}
2 | \pagestyle{fancy}
```



Les pages de début de chapitre ont automatiquement le style « plain ». Pour désactiver ce style par défaut, il faut modifier la commande `\chapter`. Nous en parlons dans un autre chapitre (☞ p. 205, 23.4).

On peut ensuite utiliser les six commandes de *fancyhdr* qui servent à définir le contenu des en-têtes et pieds de pages :

<code>\lhead</code>	reçoit comme argument le contenu de la partie gauche de l'en-tête, justifiée à gauche.
<code>\chead</code>	reçoit comme argument la partie centrale de l'en-tête, centrée.
<code>\rhead</code>	reçoit comme argument la partie droite de l'en-tête, justifiée à droite.
<code>\lfoot</code>	reçoit comme argument la partie gauche du pied de page, justifiée à gauche.
<code>\cfoot</code>	reçoit comme argument la partie centrale du pied de page, centrée.
<code>\rfoot</code>	reçoit comme argument la partie droite du pied de page, justifiée à droite.

Supposons que nous souhaitions afficher le numéro des pages en pied de page centré, en indiquant également le nombre total de pages. Nous allons utiliser le package *totpages* qui nous permet, après deux compilations, ou plus si le nombre de pages varie entre les compilations, d'obtenir le nombre total de pages.

```
1 | \usepackage{fancyhdr}
2 | \pagestyle{fancy}
3 | \usepackage{totpages}
4 | \cfoot{{\thepage} / \ref{TotPages}}
```

La commande `\thepage` indique la valeur du compteur `page` (☞ p. 138, 17.2), correspondant au numéro de page.

Vous pouvez constater le résultat sur cette page.

## 22.3 Pages recto verso et alternance gauche-droite

Lorsqu'un travail est imprimé en recto verso, on peut souhaiter que l'en-tête et le pied de page gauches des pages impaires correspondent à l'en-tête et au pied de page droits des pages paires et *vice-versa*.

Le package *fancyhdr* a prévu ce cas. Il propose deux commandes :

- `\fancyhead[⟨position⟩]{⟨texte d'en-tête⟩}`
- `\fancyfoot[⟨position⟩]{⟨texte de pied de page⟩}`

L'argument `⟨position⟩` peut prendre une ou plusieurs des valeurs suivantes :

C	centre
LO	gauche des pages impaires (= intérieur des pages de droite, si on écrit de gauche à droite).
RO	droite des pages impaires (= extérieur des pages de droite, si on écrit de gauche à droite).
LE	gauche des pages paires (= extérieur des pages de gauche, si on écrit de gauche à droite).
RE	droite des pages paires (= intérieur des pages de gauche, si on écrit de gauche à droite).

Ainsi pour mettre le numéro de page à l'extérieur du pied de page, il suffit d'écrire :

```
1 | \fancyfoot[LE,RO]{\thepage}
```

Évidemment, si l'on dit à  $\text{\LaTeX}$  de générer un fichier à destination d'une impression monoface (☞ p. 21, 2.2.1), il considère qu'il n'y a que des pages recto, c'est-à-dire impaires.

## 22.4 Titres dans l'en-tête : le mécanisme des marqueurs

Dans la classe `book`, les en-têtes contiennent par défaut les titres de chapitres sur la page de gauche et de sections sur la page de droite. En utilisant le style

fancy, on insère automatiquement les titres de chapitres du côté interne et ceux de sections dans le côté externe.

Si on utilise *fancyhdr* pour ne personnaliser que le pied de page, on peut vouloir rétablir pour le reste la présentation par défaut. Il est nécessaire pour cela de comprendre le mécanisme des marqueurs de  $\LaTeX$ .

Le principe de base est simple : des commandes de marquage vont stocker en mémoire des marqueurs. Lesdits marqueurs sont appelés par d'autres commandes.

Les deux commandes de marquage sont :

- `\markboth{marqueur gauche}{marqueur droit}`
- `\markright{marqueur droit}`

Les deux commandes d'appel des marqueurs sont :

- `\leftmark` qui retourne l'argument *marqueur gauche* de la dernière commande `\markboth`.
- `\rightmark` qui retourne l'argument *marqueur droit* de la première commande `\markright` ou `\markboth` située sur la page courante. En revanche, si la page en cours ne contient aucune de ces commandes, alors `\rightmark` retourne l'argument *marqueur droit* du dernier `\markright` ou `\markboth` utilisé.

Concrètement, les commandes `\markboth` et `\markright` sont appelées automatiquement dans la classe **book** par les commandes `\chapter` et `\section` via les commandes `\chaptermark` et `\sectionmark`.

## 22.4.1 Appeler les marqueurs dans les styles fancy

Pour rétablir la présentation originelle, il nous faut insérer les commandes `\leftmark` et `\rightmark` dans nos commandes `\fancyhead`.

```

1 | \fancyhead[LE,R0]{}
2 | \fancyhead[RE]{\leftmark}
3 | \fancyhead[L0]{\rightmark}

```

Avec ceci, nous obtenons à nouveau le titre de chapitre à gauche et le titre de section à droite.

⚡ Les en-têtes et pieds de pages sont calculés par  $\LaTeX$  après le reste de la page. C'est pourquoi on obtient dans l'en-tête le contenu de la dernière section de la page.

## 22.4.2 Redéfinir les commandes `\chaptermark` et `\sectionmark`

Supposons désormais que vous ne souhaitez plus voir les titres des sections dans les en-têtes. Le plus simple est alors de redéfinir la commande `\sectionmark`, pour la rendre nulle :

```
1 | \renewcommand{\sectionmark}[1]{}

```

⚡ Notez bien que nous indiquons que la commande `\sectionmark` prend un argument, alors même que nous ne nous en servons pas. Mais lorsque la classe `book` appelle cette commande, elle lui passe bien un argument, qui est le titre de la section.

Comme alors on affiche plus que le titre du chapitre, il est peut-être inutile de préciser qu'il s'agit d'un chapitre. Il faut alors redéfinir la commande `\chaptermark`, par exemple sous la forme suivante :

```
1 | \renewcommand{\chaptermark}[1]{\markboth {\MakeUppercase{%
2 |   \thechapter-#1}}{}}
```

👉 Si vous vous amusez à fouiller le fichier `book.cls`, vous verrez que la commande `\chaptermark` est définie deux fois. En réalité, sa définition est conditionnée par l'option de classe `twoside` ou `oneside` (☞ p. 21, 2.2.1).

👉 La commande `\thechapter` sert à afficher le numéro du chapitre, stocké dans un compteur `chapter`. Nous parlons plus loin des compteurs et des commandes `\the<compteur>` (☞ p. 201, 23.3.1).

## 22.5 Filet d'en-têtes et de pieds de pages

Le package `fancyhdr` inclut un filet entre le corps du texte et l'en-tête. Il est possible de modifier l'épaisseur de ce filet, ainsi que de rajouter un filet avant le pied de page.

On redéfinit pour cela les commandes `\headrulewidth` et `\footrulewidth`, qui doivent renvoyer une longueur (☞ p. 237, E). Par exemple, pour supprimer le filet au-dessous de l'en-tête mais en ajouter un de 0,05 mm au-dessus du pied de page, on écrit :

```
1 | \renewcommand{\headrulewidth}[0]{0pt}
2 | \renewcommand{\footrulewidth}[0]{0.05mm}
```



# Chapitre 23

## Apparence des textes

*Dans ce chapitre nous verrons comment changer l'apparence de nos textes : polices de caractères, interlignes, styles des niveaux de titres.*

### 23.1 Police de caractères

Il existe trois grandes familles de polices de caractères :

<b>À empattements</b>	pour le corps du texte. Ces empattements améliorent le confort de lecture, sur version papier du moins.
<b>Sans empattement</b>	normalement pour les titre sur les couvertures.
<b>À pas fixe</b>	qui se caractérisent par le fait que la largeur des caractères ne varie pas en fonction des lettres, mais reste uniforme d'une lettre à l'autre. Ces polices sont utilisées par les informaticiens pour citer du code informatique. Dans notre domaine, elles peuvent éventuellement servir pour une transcription diplomatique d'un texte ancien. On utilisera pour cela l'environnement <b>verbatim</b> .

Le package *fontspec*<sup>1</sup> propose de définir une police pour chaque famille de caractères. Ainsi pour ce livre avons-nous choisi les polices « Linux Libertine » pour les caractères avec empattements, « Linux Biolinum » pour les caractères sans empattements<sup>2</sup> et « DejaVu Sans Mono » pour les caractères à espacements fixes.

---

<sup>1</sup>Will ROBERTSON et Khaled HOSNY, *The fontspec package*.

<sup>2</sup>Nous n'utilisons cette police que pour la version informatique. Pour la version papier, nous avons confié la création de la couverture à Laura Pigeon, qui a fait son propre choix de police.

Nous avons indiqués dans notre préambule que nous souhaitons les utiliser :

```

1 | \setmainfont[Mapping=tex-text]{Linux Libertine}
2 | \setsansfont[Mapping=tex-text]{Linux Biolinum}
3 | \setmonofont[Scale=0.75]{DejaVu Sans Mono}

```

La commande `\setmainfont` sert à définir la police avec empattements (corps du texte), `\setsansfont` celle sans empattements et `\setmonofont` la police à espacements fixes. L'argument optionnel peut prendre un certain nombre de paramètres pour la police.

Ici nous avons indiqué avec le paramètre `Scale` que nous souhaitons réduire de 25 % la taille de la police « *DejaVu Sans Mono* », car il s'agit d'une police assez grande.

L'option `Mapping=tex-text` des deux premières commandes permet de gérer de manière particulière certaines suites de caractères, par exemple de remplacer trois tirets courts (- - -) par un tiret long (—) (☞ p. 25, 2.2.6).

Il existe d'autres paramètres, par exemple pour gérer finement les ligatures, précisés dans le manuel de *fontspec*<sup>3</sup>.



Le package *fontspec* permet de définir d'autres familles de caractères. Toutefois il n'est en général pas recommandé de varier les polices dans un texte : c'est pourquoi nous ne détaillons pas ici, mais nous nous contentons de renvoyer à la documentation du package<sup>4</sup>.

## 23.2 Interlignes

Le package *setspace* — qui ne possède pas de documentation officielle — permet de changer facilement l'interligne du document en proposant trois interlignages : simple, c'est le réglage standard, double et « un et demi ». En général une interligne de 1,5 est recommandée pour les travaux universitaires, sauf pour les notes de bas de pages, où un interlignage simple est de mise. Cela tombe bien : le package n'applique le changement d'interlignes qu'au corps du texte. Les notes de bas de page sont donc préservées.



Le package *setspace* « se contente » de définir des nouvelles commandes pour exécuter des suites complexes de commandes  $\TeX$  et  $\LaTeX$  nécessaires à la définition de l'interligne en tenant compte de la taille de la police.

Si donc vous souhaitez définir une autre taille d'interligne, vous pouvez essayer de vous inspirer du code du package. Mais ceci nécessite de maîtriser assez bien  $\LaTeX$ .

<sup>3</sup>Will ROBERTSON et Khaled HOSNY, « Font independent options », dans : *The fontspec package*.

<sup>4</sup>Will ROBERTSON et Khaled HOSNY, « New commands to select font families », dans : *The fontspec package*.

Les trois commandes qui nous intéressent sont :

- `\singlespacing` ;
- `\onehalfspacing` ;
- `\doublespacing`.

Elles s'utilisent de la manière suivante :

```

1 | \doublespacing
2 | Ce texte est en interligne double.
3 | \onehalfspacing
4 | Ce texte est en interligne 1,5.
5 | \singlespacing
6 | Ce texte est en interligne simple.
```

Tant qu'aucune commande de changement d'interligne n'est utilisée, le texte reste dans l'interligne courante.

### 23.2.1 « Commande à arguments » et « commande de bascule »

Jusqu'à maintenant, à quelques exceptions près, nous n'avons vu que des commandes prenant un certain nombre d'arguments. Ici nos commandes ne prennent pas d'argument, mais changent des paramètres, jusqu'à nouvel ordre. On appelle de telles commandes *commandes de bascule*. Ainsi, nos commandes de bascule du package *setspace* changent l'interligne de notre texte. L'utilisation des commandes de bascule est nécessaire pour la redéfinition de certaines commandes de L<sup>A</sup>T<sub>E</sub>X, dont les commandes des titres (☞ p. 203, 23.3.2).

Une commande de bascule entourée d'accolades n'a d'effet qu'au sein de la paire d'accolade. Une commande de bascule utilisée dans un environnement n'a d'effet que dans cet environnement. Si une commande de bascule est utilisée dans le préambule, elle a en général un effet dans l'ensemble du document.

Nous avons déjà présenté plus haut des commandes à bascule : celles de modification de la taille des caractères (☞ p. 57, 7.2.1).

Il est donc recommandé de mettre la commande `\onehalfspacing` en tout début de fichier, par exemple dans le préambule, pour agir sur tout le corps du texte.

### 23.2.2 Environnements de changement d'interligne

Le package *setspace* propose également trois environnements qui permettent de limiter la portée d'un changement d'interligne.

```

1 | \begin{singlespace}
2 | Texte avec une interligne simple
```

```

3 | \end{singlespace}
4 | \begin{onehalfspace}
5 | Texte avec une interligne 1,5.
6 | \end{onehalfspace}
7 | \begin{doublespace}
8 | Texte avec une interligne double.
9 | \end{doublespace}

```

### 23.2.3 (Re)définir un environnement : exemple avec l'environnement quotation

En général, les citations ont un interlignage plus réduit que le corps du texte. Aussi la tentation serait-elle grande, pour obtenir cet effet dans un document en interligne 1,5, de frapper ceci :

```

1 | \begin{singlespace}
2 | \begin{quotation}
3 | Une citation assez longue.
4 | \end{quotation}
5 | \end{singlespace}

```

Une telle méthode viole le principe de séparation de fond et de forme. Elle multiplie en outre les lignes de code. Il serait plus sage de redéfinir l'environnement `quotation`. Voici comment faire simplement<sup>5</sup>. Nous avons déjà vu plus haut un exemple de redéfinition d'environnement, pour l'environnement `latin` (☞ p. 35, 4.1.4).

Il s'agit ici non seulement de redéfinir l'environnement, mais aussi de réutiliser les propriétés de l'ancien environnement. On procède en insérant ce code en début de document, dans le préambule :

```

1 | \let\oldquotation\quotation
2 | \let\endoldquotation\endquotation
3 | \renewenvironment{quotation}
4 |     {\begin{oldquotation}\singlespace}
5 |     {\end{oldquotation}}

```

Commentaires :

**ligne 1** la commande `\let` est une commande  $\TeX$  qui copie une commande dans une autre commande. Ici nous copions la com-

---

<sup>5</sup>Nous nous sommes basés sur la classe `bredele` : Christophe MASUTTI, *Bredele : Un modèle pour une thèse en LaTeX*, URL : <http://linuxetleschoses.tuxfamily.org/Logiciels/bredele.php>.

mande `\quotation` dans la commande `\oldquotation`<sup>6</sup>. La commande `\quotation` quant à elle correspond au début de l’environnement **quotation**, c’est-à-dire à ce qui est exécuté par `\begin{quotation}`.

**ligne 2** nous copions la commande `\endquotation`, correspondant à la fin de l’environnement **quotation** — c’est à dire à ce qui est exécuté lors du `\end{quotation}`, à l’intérieur d’une commande `\endoldquotation`.

En créant ces deux commandes `\oldquotation` et `\endoldquotation` nous avons créé un environnement **oldquotation**.

**ligne 3** nous redéfinissons l’environnement **quotation**.

**ligne 4** au début de cet environnement (☞ p. 35, 4.1.4), nous ouvrons l’environnement **oldquotation** puis nous exécutons la commande `\singlinspace`.

**ligne 5** à la fin de l’environnement **quotation**, nous fermons l’environnement **oldquotation**. La commande `\singlinspace` ne s’applique qu’au contenu de l’environnement dans lequel elle est située (☞ p. 199, 23.2.1). C’est pourquoi il n’est pas nécessaire de l’annuler par une commande `\onehalfspace`.

☞ Si vous avez compris nos propos sur les commandes de début et de fin d’environnement, vous pouvez vous rendre compte que nous aurions pu obtenir le même résultat avec :

```
1 | \let\endoldquotation\endquotation
2 | \renewcommand{\quotation}{\oldquotation\singlinspace}
```

## 23.3 Personnaliser les titres

### 23.3.1 Redéfinir la numérotation

Nous avons abordé plus haut la notion de compteur. À chaque niveau de titre,  $\text{\LaTeX}$  associe un compteur **niveau**. Ainsi le compteur associé au niveau de titre `\chapter` est **chapter** (☞ p. 138, 17.2).

La valeur d’un compteur peut s’afficher via la commande `\the<compteur>`. Ainsi, si vous écrivez :

```
1 | \thechapter : \thesection
```

<sup>6</sup>Bien que « old » soit un terme anglais, nous l’utilisons car il est de convention de préfixer ainsi les commandes copiées depuis une autre commande.

Vous constatez qu'à la compilation apparaît le numéro du chapitre, tel qu'il apparaît dans l'en-tête, et le numéro de section, tel qu'il apparaît dans le titre. Ici par exemple nous obtenons :

```
23 : 23.3
```

Puisque les compteurs s'affichent à l'aide de commandes, il est aisée de redéfinir celles-ci. Si vous fouillez le fichier (☞ p. 233, C.4) `book.cls`, vous pouvez constater<sup>7</sup>, les deux (re)définitions de commande suivantes :

```
1 | \renewcommand \thechapter {\@arabic\c@chapter}
2 | \renewcommand \thesection {\thechapter.\@arabic\c@section}
```

**Ligne 1**      `\thechapter` affiche la valeur du compteur `chapter`, retournée par `\c@chapter`. Cette valeur ne peut pas être affichée directement, elle doit être au préalable formatée par la commande `\@arabic`, qui l'affiche sous forme de chiffres arabes.

**Ligne 2**      `\thesection` retourne `\thechapter` suivi d'un point, suivi de la valeur du compteur `section` affichée sous forme de chiffres arabes.

Supposons maintenant que nous souhaitons :

1. Afficher en chiffres romains majuscules le numéro de chapitre.
2. Afficher une parenthèse fermante après le numéro de section.

Il nous suffit de redéfinir ainsi ces commandes :

```
1 | \renewcommand \thechapter {\@Roman\c@chapter}
2 | \renewcommand \thesection {\thechapter.\@arabic\c@section}}
```

Si vous écrivez ces lignes dans votre préambule, vous n'arriverez pas à compiler. Cela tient à une raison simple : les commandes contenant le caractère `@` ne peuvent pas être utilisées dans un fichier `.tex`, mais uniquement dans les fichiers de définition de classe (`.cls`) ou de package (`.sty`).

Pour pouvoir les utiliser dans un fichier `.tex`, il faut les entourer des commandes `\makeatletter` et `\makeatother`.

Ce qui nous donne :

```
1 | \makeatletter
2 | \renewcommand \thechapter {\@Roman\c@chapter}
3 | \renewcommand \thesection {\thechapter.\@arabic\c@section}}
4 | \makeatother
```

<sup>7</sup>Lignes 287-288, à la date où nous écrivons.

On obtient donc un affichage de la forme :

XXIII Chapitre  
XXIII.3) Section

 Nous parlons plus loin de la manière de manipuler les compteurs (☞ p. 208, 23.5).

### 23.3.2 Définir l'apparence : sections et niveaux inférieurs

Pour personnaliser l'apparence des titres, le mieux est de regarder ce que nous proposent les classes standards. Prenons le cas de la classe **book**, définie dans le fichier (☞ p. 233, C.4) `book.cls`. En cherchant un peu, nous trouvons<sup>8</sup> :

```

1 | \newcommand\section{\@startsection {section}{1}{\z@}%
2 |     {-3.5ex \@plus -1ex \@minus -.2ex}%
3 |     {2.3ex \@plus .2ex}%
4 |     {\normalfont\Large\bfseries}}
```

Commentons ces quelques lignes :

- ligne 1** La commande `\section` fait appel à `\@startsection`. Cette dernière commande ajoute un niveau de titre à la table des matières. Malgré son nom, elle peut servir pour n'importe quel niveau de titre (par exemple, pour les `\subsections`). Elle prend plusieurs arguments. Le premier est le niveau : ici `section`. Le second est le niveau de profondeur : `1`. Le troisième est l'indentation préalable : ici `\z@`, c'est-à-dire une longueur nulle.
- ligne 2** quatrième argument de la commande `\@startsection`, qui indique l'espace vertical *avant* le titre. Cet espace est idéalement de 3,5 ex (☞ p. 237, E), mais peut être compris entre 1,5 ex ( $3,5 - 2$ ) et 4,5 ex ( $3,5 + 1$ ) : cette marge de manœuvre est indiquée par les commandes `\@minus` et `\@plus` (☞ p. 238, E.3).
- ligne 3** cinquième argument de `\@startsection`, qui indique l'espace vertical après le titre. Notre longueur étant positive, le texte qui suit débute un nouveau paragraphe. Une longueur négative (comme c'est le cas pour la définition de la commande `\paragraph`) indique qu'il n'y a pas de changement de paragraphe. Ici donc, l'espace après le titre est de 2,3 ex, mais peut atteindre 2,5 ex ( $2,3 + 0,2$ ).

<sup>8</sup>Ligne 414 à la date du 19 septembre 2011.

**ligne 4** sixième et dernier argument de `\@startsection`, qui indique l'apparence proprement dite de notre titre. Il s'agit d'un texte en police normale — c'est-à-dire celle définie par la commande `\setmainfont` —, en taille `\Large`, mise en gras (`\bfseries`). Ici toutes nos commandes sont prises comme des commandes de bascule, la bascule finissant avec l'accolade fermante. La commande `\bfseries` est l'équivalent à bascule de la commande `\textbf`. Il est recommandé de n'utiliser dans les styles de titre que des commandes à bascule, pour éviter des espaces indésirables.

Supposons maintenant que nous souhaitons avoir notre titre en italique et en police sans empattements. Il nous suffit de redéfinir la commande `\section`, en entourant notre définition de `\makeatletter` et `\makeatother` (☞ p. 202, 23.3.1).

```

1 \makeatletter
2 \renewcommand\section{\@startsection {section}{1}{\z@}%
3     {-3.5ex \@plus -1ex \@minus -.2ex}%
4     {2.3ex \@plus .2ex}%
5     {\sffamily\Large\itshape}}
6 \makeatother

```

La commande `\sffamily` produit une bascule vers la police sans empattement définie par `\setsansfont`, tandis que `\itshape` produit une bascule vers un texte en italique.

## 23.4 Définir l'apparence : chapitres et niveaux supérieurs

Si l'on cherche dans le fichier `book.cls` la commande `\chapter`, on trouve<sup>9</sup> :

```

1 \newcommand\chapter{\if@openright\cleardoublepage\else\clearpage\fi
2     \thispagestyle{plain}%
3     \global\@topnum\z@
4     \@afterindentfalse
5     \secdef\@chapter\@schapter}

```

Deux éléments nous intéressent ici.

Le premier est la ligne 2 : `\thispagestyle{plain}`. On comprend ainsi la raison pour laquelle les pages de début de chapitre ne sont pas numérotées au même endroit que les autres : le style de cette page est « plain » et non pas

<sup>9</sup>Lignes 360 et suivantes, à la date du 6 août 2011.

« heading », comme pour les autres pages (☞ p. 191, 22.1). Par conséquent, si l'on souhaite que les pages de titres aient les mêmes en-têtes et pieds de pages que les pages standards, il faut redéfinir la commande `\chapter` en supprimant `\thispagestyle{plain}`.

```

1  \makeatletter
2  \renewcommand\chapter{\if@openright\cleardoublepage %
3      \else\clearpage\fi
4      \global\@topnum\z@
5      \@afterindentfalse
6      \secdef\chapter\@schapter}
7  \makeatother

```

Le second élément intéressant est la dernière ligne. La commande `\secdef` renvoie vers `\@chapter` si nous utilisons `\chapter` et vers `\@schapter` si nous utilisons `\chapter*`.

En fouillant les codes des commandes `\@chapter` et `\@schapter` nous trouvons qu'elles appellent `\@makechapterhead` et `\@makeschapterhead`, respectivement.



`\@chapter`, `\@schapter`, `\@makechapterhead` et `\@makeschapterhead` sont définies *via* la commande `\def` et non pas `\newcommand`.

En effet la définition est faite en  $\TeX$  et non pas en  $\LaTeX$ . Pour nos propos, cela n'a pas grande influence.

Intéressons-nous à la commande `\@makechapterhead`<sup>10</sup>.

```

1  \def\@makechapterhead#1{%
2  \vspace*{50\p@}%
3  {\parindent \z@ \raggedright \normalfont
4   \ifnum \c@secnumdepth >\m@ne
5     \if@mainmatter
6       \huge\bfseries \@chapapp\space \thechapter
7       \par\nobreak
8       \vskip 20\p@
9     \fi
10    \fi
11    \interlinepenalty\@M
12    \Huge \bfseries #1\par\nobreak
13    \vskip 40\p@
14  }}

```

Analysons-là :

<sup>10</sup>Ligne 386, à la date du 19 septembre 2011.

- ligne 2** la commande `\vspace` produit un espace vertical, ici de 50 pt, comme l'indique le `50\p@`. La présence d'une astérisque indique que notre espace vertical continue après un changement de page.
- ligne 3** la commande `\parindent` précise l'indentation du paragraphe à ce point précis : ici une longueur nulle (`\z@`). La commande `\raggedright` signifie que le texte va être aligné à gauche – à l'inverse `\raggedleft` signifierait que le texte serait aligné à droite. Quant à la commande `\normalfont`, inutile de dire qu'elle signifie que la police standard – celle définie par la commande `\setmainfont` – est utilisée.
- lignes 4-5, 9, 10** on conditionne l'affichage d'un numéro de chapitre au positionnement dans la partie principale du document, celle qui suit `\mainmatter` (☞ p. 32, 3.3), et à la valeur du compteur **secnumdepth**, que vous pouvez redéfinir pour empêcher la numérotation de certains niveaux de titre (☞ p. 208, 23.5).
- ligne 6** on affiche en taille `\huge` (☞ p. 57, 7.2.1) et en gras (`\bfseries`) la chaîne de langue de début de chapitre (`\@chapapp`) suivie d'une espace et du numéro de chapitre : le compteur **chapter**, affiché *via* la commande `\thechapter`.
- ligne 7** on insère un paragraphe (`\par`) et on empêche un changement de ligne (`\nobreak`).
- ligne 8** on insère un espace vertical de 20 pt.
- ligne 11** au niveau visé par ce livre la compréhension de cette ligne n'est pas indispensable. Toutefois, pour les curieux, cette ligne sert à prévenir un titre s'étalant sur plusieurs lignes. Sommaire-ment, on peut dire que pour prévenir les ruptures de lignes ou les changements de pages,  $\text{\LaTeX}$  utilise des paramètres appelés « penalty ». Plus une « penalty » est importante, plus la probabilité d'une rupture est faible. Ici la « penalty » d'interlignage – de rupture de ligne – est définie à 1000 (`\@M`).
- ligne 12** on affiche en taille `\Huge` et en gras (`\bfseries`) le titre du chapitre (`#1`). On introduit ensuite un paragraphe (`\par`), en demandant d'éviter les changements de pages (`\nobreak`).
- ligne 13** on insère un espace vertical de 40 pt.

Nous allons maintenant fabriquer un nouveau style<sup>11</sup>, où nous souhaitons avoir :

- Le texte « chapitre » et le titre alignés à droite.
- Le texte « chapitre » en petites capitales, sans gras, et en taille normale.
- Un trait horizontal au-dessous du titre.

Nous allons donc reprendre le code existant et le copier-coller dans notre préambule entre `\makeatletter` et `\makeatother`. Nous allons ensuite effectuer les modifications suivantes :

- Remplacer `\raggedright` par `\raggedleft` pour avoir nos textes à droite.
- Remplacer le premier `\huge\bfseries` par `\scshape`, pour avoir « chapitre » en petites capitales et en taille normale.
- Insérer la commande `\hrulefill`, qui produit un filet horizontal, entre les lignes 12 et 13 (☞ p. 213, 24.5).

Cela nous donne au final :

```

1  \makeatletter
2  \def\@makechapterhead#1{%
3    \vspace*{50\p@}%
4    {\parindent \z@ \raggedleft \normalfont
5     \ifnum \c@secnumdepth >\m@ne
6       \if@mainmatter
7         \scshape \@chapapp\space \thechapter
8         \par\nobreak
9         \vskip 20\p@
10        \fi
11       \fi
12       \interlinepenalty\@M
13       \Huge \bfseries #1\par\nobreak
14       \hrulefill
15       \vskip 40\p@
16     }}
17  \makeatother

```

Évidemment, il faudrait appliquer le même type de modifications sur la commande `\makeschapterhead`. On pourrait aussi s’amuser à modifier ce qui est produit par `\part`. Pour cela il faut, de la même façon, fouiller le fichier `book.cls`.

<sup>11</sup>Nous nous inspirons ici du style de la classe `bredele` MASUTTI, *Bredele*, *op. cit.*

## 23.5 Manipuler les compteurs : le cas des notes de bas de page

Nous avons vu comment modifier l'aspect d'un compteur (☞ p. 201, 23.3.1). Mais comment modifier la valeur d'un compteur ? Il y a pour cela deux commandes : `\setcounter{<compteur>}{<valeur>}` pour affecter la valeur `<valeur>` au compteur `<compteur>`, et `\addtocounter{<compteur>}{<valeur>}` pour additionner la valeur `<valeur>` au compteur `<compteur>`. La valeur ajoutée pouvant être négative, il est ainsi possible de faire reculer un compteur.

En outre, il existe une commande `\refstepcounter` qui permet d'incrémenter d'une unité un compteur. C'est celle qui est utilisée par les commandes de niveau de titre (☞ p. 29, 3.1).

Un nouveau compteur se crée avec `\newcounter{<compteur>}[<compteur2>]`. Si l'argument `<compteur2>` est présent, alors il indique que la valeur du compteur `<compteur>` doit être réinitialisée lorsque la commande `\refstepcounter` est appliquée au compteur **compteur2**.

La commande `\@addtoreset{<compteur>}{<compteur2>}` permet d'appliquer cette règle sur un compteur déjà existant.

Ainsi, dans le fichier `book.cls`, vous remarquerez la ligne suivante<sup>12</sup> :

```
1 | \@addtoreset{footnote}{chapter}
```

Ce qui signifie que le compteur **footnote**, qui correspond à la numérotation des notes de bas de page, est réinitialisé à chaque nouveau chapitre numéroté.

Si nous souhaitons annuler cette commande, pour avoir des notes de bas de page en numérotation continue, il nous faut utiliser le package *remreset* et sa commande `\@removefromreset`, qui annule un réglage de réinitialisation.

```
1 | \usepackage{remreset}
2 | \makeatletter
3 | \@removefromreset{footnote}{chapter}
4 | \makeatother
```

---

<sup>12</sup>Ligne 718, à la date du 24 août 2011

# Chapitre 24

## Éléments de mise en page

*Dans ce chapitre, nous allons voir quelques commandes permettant de modifier la mise en page. Rappelons qu'il ne faut normalement pas utiliser ces commandes telles quelles, mais les imbriquer dans des commandes de mise en sens (☞ p. 39, 5.1).*

*Il ne s'agit ici que d'une très courte introduction : d'autres ouvrages en parlent mieux que nous<sup>1</sup>. En ce qui concerne les unités de longueurs, nous renvoyons à l'annexe E (☞ p. 237, E).*

### 24.1 Espacements

On peut produire un espace horizontal via `\hspace{⟨longueur⟩}`. Si on utilise `\hspace*`, la fin de ligne n'arrête pas l'espace.

Un espace vertical se produit par `\vspace{⟨longueur⟩}`. Si on utilise `\vspace*`, le changement de page n'affecte pas l'espace, qui continue sur la page suivante. Toutefois, nous avons pu constater qu'une telle commande pose parfois des problèmes pour la mise en page du paragraphe qui la précède.

### 24.2 Longueurs de mise en page

Pour redéfinir des paramètres tels que l'espacement vertical entre deux paragraphes ou l'indentation initiale d'un paragraphe, il ne faut pas utiliser les commandes d'espacement, mais tout simplement redéfinir les longueurs de  $\LaTeX$ .

Nous avons vu que pour l'interligne, il valait mieux utiliser le package *setspace* (☞ p. 198, 23.2). Il nous reste donc deux longueurs : l'indentation initiale

---

<sup>1</sup>En particulier Vincent LOZANO, *Tout ce que vous avez toujours voulu savoir sur  $\LaTeX$  sans jamais oser le demander*, In Libro Veritas, 2008, URL : <http://www.framabook.org/latex.html> ; Franck MITTELBACH et Michel GOOSSENS, *The  $\LaTeX$  Graphics Companion*, Boston : Addison Wesley, 2007.

du paragraphe et l'espace entre les paragraphes. Ces longueurs sont `\parindent` et `\parskip`.

Pour les redéfinir, on utilise `\setlength{⟨longueur⟩}{⟨valeur⟩}`. Par exemple, pour redéfinir la longueur de l'indentation à 3 cadrats :

```
1 | \setlength{\parindent}{3ex}
```

Il existe aussi `\addtolength{⟨longueur⟩}{⟨valeur⟩}` qui ajoute `⟨valeur⟩` à la longueur déjà existante.

Ainsi pour ajouter 1 pt à l'espacement entre deux paragraphes :

```
1 | \addtolength{\parskip}{1pt}
```

Ces redéfinitions doivent se pratiquer *après* le préambule, car elles ne s'appliquent qu'au groupe logique où elles surviennent, un groupe logique correspondant soit à un environnement, soit à un espace délimité par une paire d'accolades.



On peut supprimer l'indentation d'un paragraphe précis en le commençant par la commande `\noindent`.

## 24.3 Marges

Bien qu'il ne soit pas conseillé de le faire, on peut redéfinir les marges avec le package *geometry*. Lors du chargement du package, on dispose des options suivantes :

**lmargin** pour la marge gauche sur une impression une face, pour la marge intérieure sur une impression deux faces.

**rmargin** pour la marge droite sur une impression une face, pour la marge extérieure sur une impression deux faces.

**tmargin** pour la marge du haut.

**bmargin** pour la marge du bas.

Ainsi pour une impression avec seulement 1 cm de marge — ce qui n'est pas à conseillé, si ce n'est pour un brouillon — on peut charger ainsi le package :

```
1 | \usepackage[lmargin=1cm,rmargin=1cm,tmargin=1cm,bmargin=1cm]{geometry}
```

On peut aussi modifier temporairement les réglages, pour une ou plusieurs pages, *via* la commande `\newgeometry{⟨réglages⟩}`. On restaure ensuite les réglages initiaux avec la commande `\restoregeometry`.

Par exemple pour avoir une page sans marge, on écrit :

```

1 | \newgeometry{lmargin=0cm,rmargin=0cm,tmargin=0cm,bmargin=0cm}
2 | Une page avec une marge nulle.
3 | \newpage
4 | \restoregeometry

```

Le package *geometry* possède de nombreuses autres fonctionnalités : nous renvoyons à sa documentation<sup>2</sup>.

⚡ Si vous utilisez *geometry* et les notes de marge produites via `\marginpar`, vous pouvez constater que les pages de gauches (paires) n’affichent pas les notes. Le problème vient du fait que *geometry* redéfinit un certain nombre de longueurs. Pour afficher correctement les notes, il vous faut modifier la longueur `\marginparwidth`, en lui affectant la même valeur que `\leftmargin`.

```

1 | \setlength{\marginparwidth}{\leftmargin}

```

## 24.4 Textes en retrait

On peut souhaiter avoir un environnement qui produit du texte en retrait, semblable à l’environnement `quotation`, comme un environnement `exemple` avec un retrait gauche de 3 em et un retrait droit de 1em.

Si nous fouillons dans le fichier (☞ p. 233, C.4) `book.cls`, nous trouvons que l’environnement `quotation` est défini de la manière suivante<sup>3</sup> :

```

1 | \newenvironment{quotation}
2 |     {\list{}{\listparindent 1.5em %
3 |         \itemindent \listparindent
4 |         \rightmargin \leftmargin
5 |         \parsep \z@ \@plus\p@}%
6 |     \item\relax}
7 |     {\endlist}

```

Que nous pouvons analyser ainsi :

**ligne 2 à 5** code exécuté en début d’environnement.

**ligne 7** code exécuté en fin d’environnement. Nous fermons un environnement de type `list` — bien qu’effectivement, il ne s’agisse pas d’une liste !

**ligne 2** la commande `\list{<itemdefault>}{<réglages typographiques>}` ouvre un environnement `list`. Le premier argument correspond

<sup>2</sup>Hideo UMEKI, *The geometry package*.

<sup>3</sup>Ligne 486, au 19 septembre 2011.

à ce qu’affiche la commande `\item` en l’absence d’argument optionnel. Comme dans **quotation** on n’utilise pas `\item`, ce premier argument est nul. Le second argument contient des commandes réglant la mise en forme. La première, `\listparindent`, indique l’indentation des paragraphes à l’intérieur de la liste : ici 1,5 em.

- ligne 3** la seconde commande, `\itemindent`, indique l’indentation du premier élément de chaque `\item` : ici `\itemindent` se voit attribuer la valeur de `\listparindent`.
- ligne 4** la marge de droite de notre liste (`\rightmargin`) se voit attribuer la même valeur que la marge de gauche (`\leftmargin`). Cette valeur a été définie auparavant dans le fichier `book.cls`.
- ligne 5** la longueur séparant deux paragraphes à l’intérieur de la liste (`\parsep`) est fixée à 0 pt (`\z@`), mais peut s’étendre (`\@plus`) jusqu’à 1 pt (`\p@`) (p. 238, E.3). Le second argument de la commande `\list` finit ici.
- ligne 7** une liste ne peut pas fonctionner sans commande `\item`. C’est pourquoi cette commande est appelée ici. Pour éviter des ennuis, on indique que le texte de cet `\item` est nul, grâce à la commande `\relax`, qui est une commande servant à ne rien faire alors même que  $\TeX$  attend un contenu.

Maintenant, nous voulons notre environnement **exemple**, avec :

- Une marge gauche plus forte, de 3 cadratin (3 em).
- Une marge droite de 1 cadratin (1 em).
- Un texte en italique.

Nous allons pour cela insérer dans le deuxième argument de la commande `\list` les lignes suivantes :

```
1 | \leftmargin 3em
2 | \rightmargin 1em
3 | \itshape
```

Ce qui nous donne au final :

```
1 | \makeatletter
2 | \newenvironment{exemple}
3 |     {\list{}{\listparindent 1.5em%
4 |         \itemindent \listparindent
```

```

5          \leftmargin 3em
6      \rightmargin 1em
7      \itshape
8          \parsep          \z@ \@plus\p@}%
9          \item\relax}
10         {\endlist}
11 \makeatother

```

✎ Pour des textes encadrés ou sur fond coloré, on pourra utiliser le package *mdframed* et / ou le package *fancybox*<sup>4</sup>.

## 24.5 Trait horizontal

Un trait horizontal se trace grâce à `\rule[⟨profondeur⟩][⟨longueur⟩]{⟨épaisseur⟩}`, où `⟨profondeur⟩` correspond à l'écart entre le trait et le bas de la ligne de texte.

On peut aussi utiliser les commandes `\hrulefill` et `\dotfill` qui permettent de tracer respectivement des lignes et des points, en « comprimant » le texte qui suit, c'est-à-dire en étirant au maximum la longueur de la ligne ou de la suite de points. On dispose également de la commande `\hfill` qui permet de comprimer le texte qui suit en le faisant précéder d'espaces.

Exemple :

```

1 \hfill Du texte comprimé.
2
3 \hrulefill Du texte comprimé.
4
5 \dotfill Du texte comprimé.

```



Pour des traits plus complexes, par exemple en diagonale ou en pointillés, on se reportera au package *epic*<sup>5</sup> ou, selon le cas, à *TikZ*<sup>6</sup>.

<sup>4</sup>On peut également consulter, si on aime les défis de codage en  $\TeX$  /  $\LaTeX$ , Vincent LOZANO, « Boîtes », dans : *Tout ce que vous avez toujours voulu savoir sur LaTeX sans jamais oser le demander*, In Libro Veritas, 2008, p. 73–82, URL : <http://www.framabook.org/latex.html>.

<sup>5</sup>Conrad KWOK, *EEPIC Extensions to epic and LATEX Picture Environment Version 1.1*.

<sup>6</sup>TANTAU, *Tikz & PGF, op. cit.*



# Chapitre 25

## Formater son index

*La mise en forme par défaut de l'index n'est pas très satisfaisante. On peut heureusement la modifier pour rendre l'index plus lisible.*

### 25.1 Comment faire

Les commandes pour formater l'index sont à placer dans un fichier `monstyle.ist` que vous devez créer vous-même.

⚡ Pensez à placer ce fichier à un endroit où  $\text{\TeX}$  peut le trouver, par exemple à côté du fichier principal.

Sans le package *imakedix*, il fallait compiler en ligne de commande (☞ p. 231, C), selon cette syntaxe :

```
| makeindex -s monstyle.ist monfichier.idx
```

Il faut encore procéder ainsi lorsque vous passez l'option **noautomatic** à la commande `\makeindex` pour un index particulier, comme par exemple pour celui des sources primaires (☞ p. 152, 18.3.3).

L'option `-s` indiquait à Makeindex qu'il devait utiliser le fichier de style `monstyle.ist`.

En revanche, si vous ne passez pas l'option **noautomatic** à `\makeindex`, il vous faut passer l'option **options=-s monstyle.ist** à la commande `\makeindex` :

```
1 | \makeindex[options=-s monstyle.ist]
```

## 25.2 Quelques commandes

Nous allons donner ici quelques exemples de commandes qu'il est possible de mettre dans son fichier `monstyle.ist`.

⚡ Dans le fichier de styles `.ist`, on n'écrit pas en  $\LaTeX$  mais en  $\TeX$ . Les commandes sont ainsi encadrées par des guillemets anglais ". La barre contre-oblique, quant à elle, n'est prise en compte que si elle est précédée d'une autre contre-oblique.

Le manuel de la commande `MakeIndex`<sup>1</sup> nous indique que, pour modifier ce qui est inséré entre le premier niveau d'item (l'entrée) — ou niveau 0 — et le ou les numéros de page, on utilise l'option `delim_0`; de même, pour définir ce qui est inséré entre le deuxième niveau d'item (la sous-entrée) — ou niveau 1, et les numéros, on utilise l'option `delim_1`, et l'option `delim_2` pour les sous-sous-entrées — ou niveau 2 d'itemisation. Le choix par défaut est une virgule suivie d'un blanc : ", "

Ainsi, si nous désirons que le numéro de la page soit justifié à droite, et non simplement séparé de l'entrée par une virgule et un blanc, nous devons utiliser la commande `\hfill` (☞ p. 213, 24.5) que nous indiquons dans notre fichier de style par les lignes suivantes :

```
1 | delim_0 "\\hfill "
2 | delim_1 "\\hfill "
3 | delim_2 "\\hfill "
```

Nous obtenons donc :

Adrien	5
Charlemagne	5
Ducs	
Tassilon	5
Évêques	
Damase	5
Formose	5

Si nous préférons une ligne de points, il faut remplacer `"\\hfill "` par `"\\dotfill "`.

<sup>1</sup>Que l'on peut lire en frappant dans le terminal (☞ p. 231, C) `man makeindex` et que l'on peut quitter en frappant la lettre `q`, ou, si l'on utilise Windows, que l'on peut consulter à cette adresse <http://www.fiveanddime.net/man-pages/makeindex.1.html>.

↙ Les espaces situées entre les commandes et les deuxièmes guillemets fermants sont indispensables. En effet, en l'absence de ces espaces, MakeIndex accole, dans le fichier `.ind` généré, les commandes aux numéros de page, ce qui bloque alors la compilation, puisque  $\LaTeX$  se retrouve alors face à des commandes inconnues, du style `\hfill`.

Si l'on continue la lecture du manuel, on apprend aussi comment faire apparaître « sq. », ou ce que l'on veut, quand une même entrée est indexée dans trois pages ou plus à la suite. Il suffit d'indiquer `suffix_3p "~sq."`

Voyons maintenant comment l'on peut insérer les lettres de l'alphabet entre les groupes d'entrées. Cette manipulation permet de rendre plus lisible un index un peu long.

Le manuel indique que l'option par défaut `headings_flag 0` ne met pas de séparateurs entre les groupes, que l'option `headings_flag 1` permet d'obtenir des lettres majuscules comme séparateur, et l'option `headings_flag -1` des lettres minuscules.

Définissons ensuite la manière dont vont apparaître ces lettres. On utilise la commande `heading_prefix`. Supposons que nous voulions faire apparaître ces lettres séparatrices en gras. N'oublions pas que l'on code en  $\TeX$ ... Nous écrivons donc :

```
1 | heading_prefix "{\bfseries "
```

Si nous voulons qu'elles apparaissent en italique, nous écrivons `"\itshape"` (on utilise `\bfseries` et `\itshape`, qui sont des commandes  $\TeX$ , et non `\textbf` `\textit`).

Nous pouvons de même changer la taille des lettres séparatrices en ajoutant `\large` ou `\Large`, et ainsi de suite. (☞ p. 57, 7.2.1)

Mais nous avons ouvert une accolade... il nous faut la refermer :

```
1 | heading_suffix " }\nopagebreak\n "
```

`\nopagebreak` évite qu'une lettre-séparatrice se retrouve seule en fin de page. Le `\n` est une petite coquetterie pour permettre d'avoir, dans notre fichier `.ind`, un retour à la ligne après `\nopagebreak`.

Ce n'est pas fini. Nos lettres sont actuellement justifiées à gauche, comme les entrées de l'index. Nous voudrions qu'elles soient centrées :

```
1 | headings_flag 1
2 | heading_prefix " {\bfseries\large\hfill "
3 | heading_suffix " \hfill}\nopagebreak\n "
```

Les deux `\hfill` étirent, pour le premier, l'espace avant la lettre vers la droite, et pour le second l'espace suivant la lettre vers la gauche – voilà nos lettres centrées.

Adrien		5
	<b>C</b>	
Charlemagne		5
	<b>D</b>	
Ducs		
Tassilon		5
	<b>E</b>	
Évêques		
Damase		5
Formose		5

Si nous préférons qu'elles soient simplement légèrement décalées vers la droite, nous pouvons utiliser, à la place de `\hfill`, `\hspace*{1em}` qui rajoute à gauche un espace d'un cadratin (☞ p. 237, E; ☞ p. 209, 24.1).

Quand on a compris ce système, il est alors assez simple de formater à sa guise l'index. C'est pourquoi, pour les autres options, nous renvoyons ici à la lecture du manuel<sup>2</sup>.

Voici tout de même, à titre indicatif, le fichier final :

```

1 | headings_flag 1
2 | suffix_3p "~sq."
3 | heading_prefix " {\bfseries\large\hfill "
4 | heading_suffix " \hfill}\nopagebreak\n "
5 | delim_0 "\\hfill "
6 | delim_1 "\\hfill "
7 | delim_2 "\\hfill "
```

---

<sup>2</sup>On pourra aussi consulter Vincent LOZANO, « Allure de l'index », dans : *Tout ce que vous avez toujours voulu savoir sur LaTeX sans jamais oser le demander*, In Libro Veritas, 2008, p. 168–170, URL : <http://www.framabook.org/latex.html>.

## Sixième partie

# Annexes

« Avec un escalier prévu pour la montée, on réussit souvent à monter plus bas qu'on serait descendu avec un escalier prévu pour la descente. »

Dicton shadok



# Annexe A

## Installer L<sup>A</sup>T<sub>E</sub>X

*Avant de pouvoir se servir de E<sub>T</sub>E<sub>X</sub> (X<sub>E</sub>L<sup>A</sup>T<sub>E</sub>X), il faut l'installer sur son ordinateur. Voici comment faire pour les systèmes d'exploitation courants<sup>1</sup>.*

### A.1 La notion de distribution

En général, on n'installe pas L<sup>A</sup>T<sub>E</sub>X tout seul, on installe une distribution L<sup>A</sup>T<sub>E</sub>X. Une distribution est un ensemble de fichiers comprenant :

- les logiciels T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X, et généralement X<sub>E</sub>L<sup>A</sup>T<sub>E</sub>X, mais aussi d'autres logiciels qui appartiennent à la « famille T<sub>E</sub>X<sup>2</sup> », que nous verrons en temps voulu ;
- des fichiers permettant d'étendre les possibilités de L<sup>A</sup>T<sub>E</sub>X : les packages et les classes ;
- de la documentation sur ces fichiers.

Il existe deux distributions courantes : TeX Live, qui se veut portable et est très adaptée pour MacOS X et Linux, et MikTeX pour Windows uniquement. L'objet de ce chapitre est donc d'expliquer comment installer ces distributions.

### A.2 Installer TeX Live sous Mac Os X

La méthode la plus simple consiste à installer MacTeX<sup>3</sup>. Celle-ci comprend non seulement une version de TeX Live pour Mac, mais aussi des logiciels pour

---

<sup>1</sup>Merci à Brendan Chabannes pour la rédaction de la majeure partie de ce chapitre.

<sup>2</sup>Pour prendre une analogie avec les logiciels connus, quand vous installez Microsoft Office, vous installez non seulement Word, Excel et Powerpoint, mais aussi des logiciels utiles à ces derniers, par exemple pour tracer des organigrammes.

<sup>3</sup><http://www.tug.org/mactex/>

faciliter la rédaction avec  $\LaTeX$ .

Pour l'installer, il suffit de télécharger le fichier d'installation sur le site, en page d'accueil. Il s'agit d'un fichier `.zip`, qu'il faut le cas échéant décompresser. Dedans se trouve un fichier `.mpkg` : un double clic dessus suffit à lancer le logiciel d'installation. Il vaut mieux conserver les réglages standards d'installation.

Une fois l'installation effectuée, vous pouvez vous rendre dans le dossier « TeX » du dossier « Applications » de votre Mac.

Ce dossier comprend plusieurs applications :

- BibDesk** qui est un logiciel pour gérer les fichiers bibliographiques au format `.bib` (☞ p. 79, 10).
- Excalibur** qui est un logiciel de correction orthographique prévu pour reconnaître les commandes  $\LaTeX$ . Nous ignorons sa qualité, ne l'ayant pas essayé.
- LaTeXit** qui est un logiciel pour rédiger encore plus facilement des équations en  $\LaTeX$  et les exporter vers divers logiciels. Comme ce livre est l'un des rares sur  $\LaTeX$  à ne pas expliquer comment faire des équations avec, nous ne nous étendrons pas.
- TeXLive Utility** qui sert à mettre à jour les différents modules de  $\LaTeX$  : classes, packages, etc. (☞ p. 226, A.5.1)
- TeXworks et TeXShop** qui sont deux éditeurs de textes prévus pour  $\LaTeX$ . Choisissez l'un d'entre eux pour commencer en  $\LaTeX$  (☞ p. 19, 2).

En outre, un panneau de configuration, « Distribution TeX » est ajouté dans les Préférences Systèmes. Dans la plupart des cas, vous n'aurez pas à vous en servir. Il permet en effet de choisir entre plusieurs versions de la distribution (par exemple entre la version 2011 et la version 2012), ce qui est rarement nécessaires.

### A.2.1 Les caractères $\LaTeX$ sous Mac Os X

Les claviers Apple n'indiquent pas directement les caractères spéciaux disponibles. Voici comment frapper la plupart des caractères nécessaires à l'utilisation de  $\LaTeX$ .

Caractère	Frappe
\	⌘ + ⌥ + /
{	⌘ + (
}	⌘ + )
[	⌘ + ⌥ + (

Caractère	Frappe
]	␣ + ↑ + )
	␣ + ↑ + L
~	␣ + N

Et pour les claviers de Suisse romande, en utilisant la barre supérieure de chiffres :

Caractère	Frappe
\	␣ + ↑ + 7
{	␣ + 8
}	␣ + 9
[	␣ + ↑ + 5
]	␣ + ↑ + 6
	␣ + ↑ + 7
~	␣ + N

## A.3 Installer TeX Live sous GNU/Linux

La plupart du temps, les paquets TeX Live sont disponibles dans les dépôts de votre distribution. Néanmoins, la majeure partie des distributions GNU/Linux ne gèrent pas certains packages récents tels que Biber<sup>4</sup>. C'est pourquoi nous privilégierons l'installation à partir du programme que vous pouvez trouver sur le site de TeX Live.

Une installation complète une fois effectuée pèse environs trois giga-octets. Le temps passant et les mises à jour s'accumulant, il est recommandé d'avoir quatre giga-octets d'espace libre sur le disque dur.

### A.3.1 Au préalable

Assurez-vous avant toute chose d'avoir le paquet `perl-tk` installé sur votre système. Si vous ne le possédez pas, vous devrez lancer l'installation en ligne de commande uniquement.

À titre d'exemple, le paquet se nomme « `perl-tk` » sous Debian et Ubuntu, « `perl-Tk-804` » sous Fedora et ses dérivés.

Sous Ubuntu, par exemple, il existe un moyen très simple d'installer ce paquet : tapez `apt://perl-tk` dans la barre d'adresse de votre navigateur web Firefox.

---

<sup>4</sup>Les choses sont toutefois en train d'évoluer : les prochaines versions de systèmes tels que Debian ou Ubuntu (respectivement nommées *Wheezy* et *Quantal Quetzal*) disposeront de TeX Live 2012 et de Biber.

Rendez-vous alors sur <http://tug.org/texlive/acquire-netinstall.html>. Choisissez le fichier *install-tl-unx.tar.gz*. Une fois téléchargé, décompressez-le puis rendez-vous dans le dossier nouvellement créé à l'aide d'un terminal, par exemple en tapant `cd ~/Téléchargements/install-tl-*/`. Vous pouvez alors lancer l'installation en mode administrateur, par exemple ainsi :

```
| sudo perl install-tl -gui wizard
```

Cette commande lancera le programme d'installation de la distribution complète qui reconnaîtra automatiquement l'architecture de votre plate-forme.

### A.3.2 Installation et configuration

Après avoir donné votre mot de passe, il vous suffira de suivre les étapes suivantes :

1. Cliquez simplement sur « Suivant ».
2. Attendez que le téléchargement soit fini. Il vous faut choisir un dossier d'installation : choisissez de préférence un répertoire hors du système de base, comme `/$HOME/texlive20125`.
3. Conservez A4 pour la taille de papier.
4. Cliquez sur Installer.
5. À la fin de l'installation, un message vous informera qu'il vous faut ajuster votre « \$PATH ». Ouvrez le fichier `~/ .bashrc` (créez-le s'il n'existe pas), et ajoutez-y les lignes suivantes :<sup>6</sup>

```
PATH="/$HOME/texlive2012/bin/x86_64-linux :$PATH"
```

L'installation est terminée. Vous pouvez fermer votre terminal pour que les modifications appliquées à votre « \$PATH » soient prises en compte.

## A.4 MiKTeX sous Windows

Les systèmes Windows diffèrent considérablement des systèmes GNU/Linux et Mac OS X. Par conséquent, s'il est possible de faire une installation manuelle de la distribution TeX Live sous Windows, la procédure est fastidieuse et complexe.

<sup>5</sup>« \$HOME » est une *variable d'environnement*. Vous pourriez tout aussi bien taper « /home/justin/ » si votre nom d'utilisateur est « justin ».

<sup>6</sup>« \$PATH » est une variable d'environnement qui permet au système de savoir où chercher les exécutables. TeX Live n'étant pas installé dans les dossiers par défaut, puisqu'il est étranger au système de base, c'est à vous de l'intégrer. Ajustez le champ entre guillemets selon le message qui s'affiche à l'écran, en prenant soin de finir par « :\$PATH ».

C'est pourquoi nous nous tournerons vers une distribution consacrée exclusivement à Windows, qui automatise les tâches d'installation de la distribution  $\LaTeX$  et permet de gérer son installation en se conformant aux pratiques qui ont cours sur ce système. Cette distribution a pour nom MiKTeX. Son installateur est très complet : en plus d'une distribution  $\LaTeX$ , il installe aussi un logiciel graphique de mise à jour des paquets et un éditeur de texte.

### A.4.1 Installation

Il faut se rendre sur la page de téléchargement de la distribution : <http://miktex.org/2.9/setup>. Choisissez le fichier *Net Installer*, et non le *Basic Installer*.

⚡ Vous remarquerez que des installateurs sont fournis pour les versions dites respectivement 32 bits et 64 bits de Windows. Si vous ne connaissez pas ces termes, veuillez à choisir la version *32 bits*. Cette dernière peut en effet s'exécuter sur les deux types de plate-formes, tandis que la réciproque n'est pas vraie. Si vous savez que vous disposez d'un système en 64 bits, vous pouvez choisir sans crainte l'installateur qui y correspond.

Une fois l'installateur téléchargé sur votre ordinateur, un double-clic lance la procédure. Il faut d'abord accepter la licence du logiciel. Puis choisissez « Download MiKTeX », et, à l'écran suivant, « Basic MiKTeX ».

Il vous est demandé de choisir une source de téléchargement. Préférez une source proche de votre domicile, donc pour un utilisateur habitant dans le Nord de la France, les serveurs français, anglais ou allemands font amplement l'affaire.

À l'étape suivante il faut choisir un répertoire de téléchargement. Choisissez de préférence un dossier qui se trouve dans « Mes Documents », comme par exemple `C:\Documents and Settings\Votre utilisateur\Mes documents\miktex`. Lorsque vous avez achevé cette étape, l'installateur télécharge tous les composants dont il a besoin, vous avertit qu'il a terminé son travail, puis s'arrête.

Lancez-le alors une seconde fois. Mais au lieu de choisir « Download MiKTeX », choisissez à présent « Install MiKTeX », puis de nouveau « Basic MiKTeX ». Préférez alors une installation en tant qu'administrateur, pour tous les utilisateurs de l'ordinateur. Puis sélectionnez le dossier qui contient les composants ; dans notre exemple précédent, il s'agissait de `C:\Documents and Settings\Votre utilisateur\Mes documents\miktex`.

Conservez les choix par défaut sur les deux écrans suivants : ils sont corrects. Vous pouvez achever le processus.

## A.5 Mettre à jour les packages

La plupart des packages sont listés sur <http://www.ctan.org/tex-archive/> qui sert de dépôt pour tous les projets tournant autour de  $\TeX$  et  $\LaTeX$ . C'est en

général sur les serveurs du CTAN (*Comprehensive TeX Archive Network*) que les logiciels de mise à jour des packages vont chercher ceux-ci.

✉ Pour être tenu au courant des nouveaux packages et des mis à jours des anciens packages, vous pouvez vous abonner à la liste <https://lists.dante.de/mailman/listinfo/ctan-ann>.

⚡ Les mises à jour de la distribution TeX Live ne sont fournies que pour une période d'un an environ après sa sortie (généralement entre mai et juillet). Passé ce délai, lorsque vous tenterez de mettre à jour votre distribution TeX Live, vous trouverez un message de ce type :

```
/home/justin/texlive2011/bin/x86_64-linux/tlmgr: The
TeX Live versions supported by the repository
(2012–2012)
do not include the version of the local installation
(2011). Goodbye.
```

ou bien

```
TeX Live 2011 is frozen forever and will no longer be
updated.
```

Dans un tel cas, il vous faut soit installer une nouvelle version de la distribution TeX Live, en recommençant la procédure d'installation, soit continuer à utiliser l'ancienne qui ne reçoit plus de mises à jour. TeX Live est d'une grande stabilité, mais ce n'est pas nécessairement le cas de tous les packages que l'on peut être amené à utiliser. Si l'un des packages que vous utilisez est affecté d'un bug qui vous gêne, vous n'aurez d'autre choix que de mettre à niveau la distribution TeX Live. Dans le cas contraire, vous pouvez parfaitement vous contenter de l'ancienne version.

### A.5.1 Sous Mac OS X

Pour mettre à jour les packages, le plus simple est d'utiliser le logiciel « TeX-Live Utility », livré avec MacTeX. Une fois le logiciel ouvert, il suffit de choisir l'onglet « Manage Updates », puis de sélectionner le package à mettre à jour — s'il ne s'affiche pas, c'est qu'il est à jour — avant de cliquer sur « Update ». À noter qu'on peut aussi utiliser l'onglet « Manage Packages » pour gérer l'installation de nouveaux packages.

### A.5.2 Sous Linux

Il y a deux moyens de mettre à jour votre installation  $\LaTeX$  sous GNU/Linux.

1. Le premier moyen de mettre à jour simplement est d'utiliser `tlmgr` dans un terminal. La commande permettant de tout mettre à jour est la suivante : `tlmgr update --self --all`. S'il s'avère nécessaire d'installer un nouveau paquet, il faut utiliser la commande `tlmgr install paquet`.
2. Vous pouvez lancer l'interface graphique de gestion des paquets en tapant la commande `tlmgr gui`.

### A.5.3 Sous Windows

Dans le menu Démarrer, vous trouvez une section nommée fort justement MiKTeX, suivie d'un numéro de version. Dans cette section, il y a notamment une sous-section consacrée à la « Maintenance » — répondant au doux nom de « Maintenance (Admin) » —, dans laquelle vous avez accès à deux outils utiles : le gestionnaire de mises à jour de packages (« Updates »), et le gestionnaire d'installation de nouveaux packages (« Package Manager »).

Remarquez que la plupart du temps, vous n'aurez pas besoin d'installer explicitement un nouveau package. En effet, au moment de la compilation, MiKTeX détecte que vous sollicitez un composant qui n'est pas présent sur votre système et vous propose de l'installer.



## Annexe B

# Quelques logiciels pour travailler avec $\LaTeX$

*Ce chapitre présente quelques logiciels pour travailler avec  $\LaTeX$ . Cette liste est non exhaustive. Ne sont proposés que des logiciels gratuits et sous licence libre.*

### B.1 Éditeurs de texte spécialisés en $\LaTeX$

#### B.1.1 TeXMaker (Multi-plateformes)

Des logiciels « grand public » il est sans doute le mieux adapté aux longs travaux de rédaction : possibilités d'afficher le plan, d'ouvrir automatiquement des fichiers inclus, disposition en onglets, réglages avancés des raccourcis de frappe, etc. On peut lui reprocher un temps de démarrage relativement long.

Pour demander l'enregistrement en UTF-8, il faut se rendre dans les préférences, onglet « Editeur ».

Malheureusement ce logiciel ne propose pas en standard de bouton pour compiler avec  $\XeTeX$ . Il vous faudra donc avant toutes choses vous rendre dans les préférences, onglet « Commandes » et remplacer tous les « latex » ou « pdflatex » par « xelatex », puis valider.

Pour compiler avec  $\XeTeX$ , il faudra utiliser l'outil de compilation avec  $\LaTeX$ .

#### B.1.2 TeXWorks (Multi-plateformes)

Sans doute le premier logiciel à utiliser pour commencer avec  $\LaTeX$ . Les options sont relativement limitées, ce qui favorise une prise en main rapide. En

outre le système de visualisation des PDFs est extrêmement pratique, permettant de voir en parallèle la version PDF et la version  $\LaTeX$ . Toutefois on trouvera vite ce logiciel limité en terme de fonctionnalités.

Pour demander l'enregistrement en UTF-8, il faut se rendre dans les préférences, onglet « Editeur ».

### B.1.3 TeXShop (Mac)

Ce logiciel est livré avec MacTeX. Il propose des boutons de composition au dessus de chaque fenêtre. Relativement léger à utiliser, il y manque cependant les fonctionnalités d'affichage du plan et d'ouverture automatique des fichiers inclus. Il possède peu de réglages avancés de rédaction. Sa force reste, à nos yeux, sa rapidité de lancement et sa fluidité.

Pour demander l'enregistrement en UTF-8, il faut se rendre dans les préférences, onglet « Document ».

## B.2 Logiciels de gestion bibliographique au format .bib

Il existe deux principaux logiciels de gestion de fichier `.bib` : JabRef<sup>1</sup> et BibDesk<sup>2</sup>. Tous les deux proposent de nombreuses fonctionnalités : recherches des doublons, tri selon plusieurs critères, import / export dans d'autres formats que `.bib`. Le choix entre les deux est donc délicat. Les utilisateurs Mac préféreront sans doute BibDesk, en raison de la proximité de son organisation avec certains logiciels Apple.

Pour BibDesk, le réglage en UTF-8 se fait dans les préférences, bouton « fichier ». Pour Jabref, il se fait dans les préférences, onglet « general ».

---

<sup>1</sup><http://jabref.sourceforge.net/>, multi-plateformes.

<sup>2</sup><http://bibdesk.sourceforge.net/>, livré avec MacTeX, disponible uniquement sous Mac.

# Annexe C

## Introduction à la ligne de commande

*La plupart des éditeurs de textes spécialisés en LaTeX proposent un bouton pour mettre en œuvre les principales commandes de compilation : XeLaTeX, MakeIndex, parfois Biber.*

*Il devient toutefois très rapidement nécessaire de pouvoir faire plus : par exemple si l'on utilise le script de gestion des index de sources primaires (☞ p. 153, 18.3.5), il faut pouvoir l'exécuter.*

*On utilise pour cela le terminal du système d'exploitation, qui permet d'exécuter directement ces commandes. Voici une brève introduction à son utilisation dans le cadre d'un usage de XeLaTeX.*

### C.1 La notion de répertoire courant

Ce qu'on appelle « répertoire courant » correspond à l'emplacement où l'on se situe dans l'arborescence des fichiers de l'ordinateur. Quand on veut utiliser les lignes de commande pour se servir de XeLaTeX, la première chose à faire est de changer le répertoire courant pour se rendre dans le dossier dans lequel se situe les fichiers à compiler.

### C.2 Mac OS X et Linux

Sur Mac OS X, le terminal se situe dans le dossier « Utilitaires » du dossier « Applications ».

Sous GNU/Linux, le Terminal se trouve dans le menu « Application » – « Accessoires », ou bien il vous suffit d'appuyer sur la touche « Windows » puis de taper les premières lettres de l'application : `term` devrait suffire. Cliquez sur le symbole qui apparaît.

Le répertoire courant est généralement indiqué à gauche de la ligne, le symbole `~` représentant le dossier de départ. Pour lister son contenu, frapper `ls`. Pour valider une commande, il faut frapper sur la touche Entrée.

Pour vous déplacer dans un répertoire, il suffit de frapper la commande `cd` suivie du dossier où vous souhaitez vous rendre.

Ainsi la commande `cd projet-latex` vous fait pénétrer dans le répertoire `projet-latex`, ce que vous pouvez vérifier avec la commande `dir`. Pour vous déplacer dans un répertoire parent, frappez simplement `cd ..`

Plutôt que de remonter de dossier en dossier avec la commande `cd` jusqu'au répertoire courant, on peut aussi taper `cd` suivi directement du chemin complet (☞ p. 31, 3.2.1) du dossier où l'on veut se rendre. Sous Mac ou sous Linux, si l'on n'est pas sûr du chemin, on peut le vérifier en faisant simplement glisser l'icône du dossier sur la fenêtre du Terminal : le chemin s'affiche alors.

Lorsque vous êtes dans le répertoire où vous souhaitez exécuter une commande, vous pouvez la lancer de la même manière :

```
| xelatex nomdufichieràcompiler.tex
| biber nomdufichieràcompiler
| makeindex nomdufichieràcompiler
| etc.
```

Enfin la commande `ls` vous permet d'afficher le contenu du dossier dans lequel vous vous trouvez.

## C.3 Windows

Sous Windows, le terminal s'appelle « Invite de commandes ». Le langage est différent de celui que l'on trouve dans les systèmes de type Unix comme Linux et Mac OS X. On trouve cependant quelques similarités.

Pour démarrer une invite de commande, pressez simultanément la touche Windows de votre clavier et la touche R. Dans l'invite qui s'ouvre frappez `cmd` puis Entrée. Vous voilà face à une console.

Toute commande, une fois frappée, est validée par une pression sur la touche Entrée.

Le répertoire courant est indiqué à gauche du curseur clignotant. La commande `dir` vous indique ce qui se trouve dans le répertoire courant.

Pour vous déplacer dans un répertoire, il suffit de frapper la commande `cd` suivie du dossier où vous souhaitez vous rendre.

Ainsi la commande `cd projet-latex` vous fait pénétrer dans le répertoire `projet-latex`, ce que vous pouvez vérifier avec la commande `dir`. Pour vous déplacer dans un répertoire parent, frappez simplement `cd ..`

Lorsque vous êtes dans le répertoire où vous souhaitez exécuter une commande, vous pouvez la lancer de la même manière :

```
| xelatex nomdufichieràcompiler.tex
```

La commande `dir` affiche enfin le contenu du répertoire dans lequel vous êtes acuellement.

## C.4 Trouver les fichiers standards

Pour pouvoir personnaliser le comportement standard de  $\text{\LaTeX}$  ou d'un package, par exemple les styles de *biblatex*, il est en général nécessaire de regarder les fichiers standards pour redéfinir telle ou telle commande ou macro bibliographique.

Il faut donc trouver leurs emplacements. Dans le fichier `.log` produit lors de la compilation est listé l'ensemble des fichiers chargés par le compilateur. Il est général aisé de connaître le nom du fichier : il s'agit souvent de celui du package ou de la classe. Une recherche dans le fichier permet donc de trouver le chemin, par exemple :

```
/usr/local/texlive/2012/texmf-dist/tex/latex/biblatex/cbx/verbose-trad2.cbx.
```

En général les fichiers se trouvent dans des répertoires invisibles via l'interface standard du système d'exploitation. Pour les ouvrir, il faut donc utiliser la ligne de commande, et frapper

**Sous Mac**      `open chemin-du-fichier.`

**Sous Windows** `chemin-du-fichier.` Cependant, par défaut, le copier-coller est désactivé dans l'invite de commande. On peut contourner cela en copiant-collant dans la barre d'adresse du navigateur Firefox.

**Sous Linux**    `nom_de_lediteur chemin-du-fichier.`

⚡ Il ne faut jamais modifier les fichiers standards. En effet, en cas de mise à jour de ces fichiers, vous perdriez toutes vos modifications. Il vaut mieux en général redéfinir les commandes, quitte à grouper ces redéfinitions dans un fichier.



# Annexe D

## Faciliter les compilations avec Latexmk

*Il est fréquent avec  $\text{\LaTeX}$  de devoir faire plusieurs compilations, avec différents programmes, dans un ordre précis. Le programme Latexmk permet de gérer automatiquement ces différentes compilations.*

### D.1 Principe

Latexmk est un script installé avec toutes les distributions  $\text{\LaTeX}$ . Il s'utilise via la ligne de commande (☞ p. 231, C). Latexmk consulte les fichiers de compilation (`.log`) pour savoir quels sont les scripts à exécuter, et dans quel ordre. Très simple d'utilisation, il nécessite toutefois quelques adaptations pour être compatible avec  $\text{\XeLaTeX}$ . Nous les présentons ici, mais le programme propose beaucoup plus de réglages : nous renvoyons au manuel<sup>1</sup>.

### D.2 Adaptation pour $\text{\XeLaTeX}$

Pour configurer le script, il vous faut créer un fichier `latexmrc` à côté du fichier à compiler. Dans ce fichier, écrivez les lignes suivantes, en prêtant attention aux points-virgules de fins de ligne :

```
$pdf_mode = "1" ;  
$pdflatex = "xelatex" ;
```

---

<sup>1</sup>John COLLINS, *Latexmk*.

La première ligne signifie que vous demandez à Latexmk de vous produire des fichiers **.pdf** et non pas **.dvi**, qui est le format « historique » produit par  $\LaTeX$ .

La seconde ligne signifie que vous demandez à Latexmk d'utiliser  $X_{\LaTeX}$  et non pas  $\LaTeX$  pour produire ces **.pdf**.

Rendez vous avec le terminal (☞ p. 231, C) dans le dossier du fichier à compiler. Frappez ensuite la commande suivante :

```
| latexmk fichier
```

où `fichier` est le nom de votre fichier principal. Vous voyez alors défiler les différentes compilations. Latexmk se charge de faire autant de compilations que nécessaires, en tenant compte des éventuels déplacements d'étiquettes. À la fin de ces différentes compilations, vous trouverez le fichier **.pdf** désiré.

Toutefois, si vous appelez une entrée bibliographique non définie (☞ p. 79, 10) ou une étiquette de renvoi interne (☞ p. 133, 16) non définie, Latexmk ne procède qu'à cinq séries de compilations<sup>2</sup>. Il vous indique cependant les entrées ou étiquettes problématiques, par des lignes ayant cette forme :

```
| Latexmk : Citation 'XXX' on page YYY undefined
| Latexmk : Reference 'XXX' on page YYY undefined
```

Il vous est donc aisé de retrouver les entrées bibliographiques (=« Citation ») et les étiquettes (=« Reference ») non définies.

## D.3 Adaptation pour l'index des sources primaires

Si vous utilisez le script de gestion de l'index des sources primaires (☞ p. 153, 18.3.5), il vous faudra également ajouter la ligne suivante :

```
| $makeindex="python index.py;makeindex %S" ;
```

Cette ligne indique à latexmk d'exécuter le script python avant la commande `makeindex`, le code `%S` désignant le fichier pour lequel `makeindex` doit être exécuté.

---

<sup>2</sup>Si au bout de cinq compilation le logiciel ne parvient pas à un résultat stable, il s'arrêtera avec une erreur.

# Annexe E

## Unités de mesure en L<sup>A</sup>T<sub>E</sub>X

Nous expliquons ici les principales unités de mesure en L<sup>A</sup>T<sub>E</sub>X, mais aussi les manières d'indiquer des tailles relatives : indiquer, par exemple, que nous souhaitons avoir la moitié de la longueur de la ligne.

### E.1 Unités de mesure

L<sup>A</sup>T<sub>E</sub>X étant d'origine anglophone, le séparateur des décimales doit être un point et non pas une virgule. Ainsi pour préciser une taille de 1,5 cm, il faut écrire : 1.5cm et non pas 1,5cm.

Les unités de mesure peuvent être divisées en deux catégories : unités absolues et unités relatives à la taille de la police courante. Nous ne présentons ici que les unités les plus courantes.

Unité	Nom	Explication
<i>Unités absolues</i>		
pt	Point typographique pica	Unité de base des mesures typographiques dans les systèmes anglo-saxons. C'est celle qui est utilisée dans quasiment tous les logiciels. Correspond approximativement à 0,35 mm.
mm	millimètre	
cm	centimètre	
in	Pouce anglo-saxon	Correspond approximativement à 2,54 cm.
<i>Unités relatives</i>		
em	Cadratin	Correspond à la largeur d'un m dans la police courante.

Unité	Nom	Explication
ex	Œil	Correspond à la hauteur d'un x dans la police courante.

## E.2 Longueurs relatives

Il est également possible d'indiquer des longueurs relatives à certains éléments. Il suffit d'indiquer un nombre décimal devant une commande de longueur. Ainsi pour un trait horizontal d'une moitié de largeur de la ligne :

```
1 | \noindent\rule{0.5\linewidth}{1pt}
```

Voici les principales commandes de longueur :

Commande	Explication
<code>\baselineskip</code>	Distance entre le bas d'une ligne et celui de la ligne suivante.
<code>\linewidth</code>	Largeur d'une ligne. Dans certains environnements (par exemple <b>quotation</b> ) les lignes n'ont pas la même longueur que dans le texte principal.
<code>\textheight</code>	Hauteur maximale du texte dans une page. Correspond <i>grosso-modo</i> à la longueur entre la marge haute et la marge basse.
<code>\textwidth</code>	Largeur maximale de texte dans une page. Correspond <i>grosso-modo</i> à la longueur entre la marge gauche et la marge droite.
<code>\parindent</code>	Longueur de l'indentation du paragraphe.
<code>\parskip</code>	Distance verticale entre deux paragraphes.

## E.3 Élasticité

Il est possible de définir une certaine élasticité pour les longueurs, qui permet de s'adapter à l'espace disponible et aux contraintes de mise en page<sup>1</sup>. On utilise

<sup>1</sup>Par exemple concernant les veuves et orphelines.

pour cela les mot clefs `minus` et `plus`. Ainsi, si nous souhaitons afficher un espace blanc (☞ p. 209, 24.1) vertical d'une longueur comprise entre 1 cm et 3 cm, mais tournant idéalement autour de 2,5 cm, nous écrivons :

```
1 | \vspace{2.5cm minus 1.5cm plus 0.5cm}
```

Cette syntaxe est utilisée dans les commandes de définitions des en-têtes de section (☞ p. 203, 23.3.2), sous une forme légèrement différente : `\@plus` et `\@minus` ne sont que des alias  $\TeX$  des mot clefs  $\TeX$  `plus` et `minus`.

Plus globalement, avec une syntaxe de longueur `a minus b plus c` la longueur finale `m` est telle que<sup>2</sup>  $a - b \leq m \leq a + c$ .

---

<sup>2</sup>Nous n'insérons pas ici une équation mais une inéquation. Le sous-titre de ce livre est donc pleinement justifié.



## Annexe F

# Travail collaboratif : les logiciels de suivi des révisions

*La plupart des traitements de texte proposent des systèmes pour travailler à plusieurs sur un même fichier et garder un historique des révisions, indiquant les modifications, leurs dates et leurs auteurs respectifs.*

*Comment faire en L<sup>A</sup>T<sub>E</sub>X ? La solution la plus simple est d'utiliser un logiciel de suivi des révisions, utilisé par les programmeurs qui travaillent à plusieurs sur un projet.*

### F.1 Principe

Imaginons que Bob et Alice travaillent ensemble sur un projet<sup>1</sup>. Ils souhaitent pouvoir échanger facilement leurs modifications.

Ils déposent pour cela les fichiers initiaux sur un serveur (un ordinateur distant). Puis chacun récupère ces fichiers sur son propre ordinateur, à l'aide de commandes spécifiques au logiciel utilisé.

Bob fait des modifications (1) : il envoie ces modifications sur le serveur, en les accompagnant d'un petit message les résumant. Alice peut alors récupérer ces modifications en local (2). Si Bob n'a modifié qu'un seul fichier, seul ce fichier sera récupéré par Alice : cela permet à Alice de modifier d'autres fichiers pendant ce temps. Une fois ses modifications faites, Alice les envoie sur le serveur, en les accompagnant d'un petit message les résumant (3), et Bob peut ainsi les récupérer. Ces étapes sont représentées dans le schéma F.1 (p. 242).

Il est aussi possible de faire d'autres opérations : renommer, déplacer, dupliquer les fichiers sur le serveur distant. En outre, ces logiciels permettent, si

---

<sup>1</sup>Le nombre de collaborateurs avec de tels systèmes est illimité : ici nous n'en prenons que deux pour simplifier.

deux personnes ont fait des modifications en même temps sur le même fichier, de proposer de ne garder que certaines de ces modifications.<sup>2</sup>

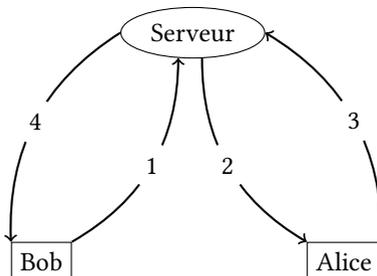


FIG. F.1 : Fonctionnement des logiciels de suivi des révisions

Un tel système présente de nombreux intérêts :

- il est aisé de s'assurer que l'on possède bien la dernière version du projet, contrairement aux échanges classiques par courriels qui peuvent rapidement semer la confusion ;
- on dispose d'un historique des versions permettant de revenir en arrière, le cas échéant<sup>3</sup> ;
- contrairement aux systèmes des logiciels de traitement de texte, l'historique des versions ne se situe pas à l'intérieur du fichier de travail, ce qui évite une prise de poids et un certain nombre de problèmes (notamment ralentissement et corruption du fichier).

## F.2 Mise en pratique

Il existe de nombreux logiciels de suivi des révisions : citons SVN, Git et Mercurial. Le premier est sans doute plus simple en première approche, mais les deux autres possèdent des fonctionnalités plus souples, notamment parce qu'il n'y a pas un serveur unique, comme sur notre schéma, mais plusieurs serveurs qui peuvent être synchronisés.

Un certain nombre d'hébergeurs proposent des services SVN, Git ou Mercurial<sup>4</sup>.

<sup>2</sup>Le jargon nomme cette étape une *résolution de conflit*.

<sup>3</sup>C'est pourquoi une personne seule peut aussi trouver un intérêt à utiliser un tel système, pour se garantir un suivi du travail.

<sup>4</sup>Toutefois par souci de confidentialité il est conseillé d'utiliser un hébergement SVN / Git / Mercurial interne à l'institution dans laquelle ce projet commun s'inscrit, en demandant, le cas échéant, au service informatique de le fournir.

Git et Mercurial ne fonctionnent pas nécessairement avec un serveur distant. C'est pourquoi on peut facilement s'en servir pour gérer un historique de modification sur un projet mono-rédacteur.

La manipulation de ces outils peut se faire en ligne de commande, mais il existe également des logiciels graphiques.

Nous invitons le lecteur curieux à se renseigner sur internet pour plus d'informations : il y trouvera aisément documentations et tutoriels<sup>5</sup>.

---

<sup>5</sup>Nous recommandons, pour Git, cet excellent livre : Scott CHACON, *Pro Git : professional version control*, URL : <http://progit.org/book/fr/>; pour notre part, nous avons écrit un petit tutoriel sur l'Utilisation de Git avec une seule personne : Maïeul ROUQUETTE, *Garder un historique de son travail grâce à Git*, URL : <http://geekographie.maieul.net/83>.



# Annexe G

## Trouver de l'aide

*Comment faire lorsque l'on est coincé sur un point particulier, qu'on a relu les différents manuels plusieurs fois<sup>1</sup> ? Demander de l'aide à d'autres utilisateurs de  $\text{\LaTeX}$ .*

*Voici quelques endroits où le faire.*

### G.1 Forums internet

Les forums Internet sur  $\text{\LaTeX}$  sont pléthore. En anglais, on peut utiliser celui de « *LaTeX Community* » : <http://www.latex-community.org/forum/>. En français, on peut utiliser celui du site « *Developpez.net* » <http://www.developpez.net/forums/f149/autres-langages/autres-langages/latex/>.

### G.2 Messagerie instantanée

Il est possible de demander de l'aide sur différents salons de messagerie instantanée fonctionnant par IRC<sup>2</sup>. En général on peut y trouver de l'aide assez rapidement.

Pour se connecter à un salon de discussion IRC, on peut utiliser le plugin Chatzilla du logiciel libre Firefox.

En français, l'adresse est `irc://irc.rezosup.org/latex`; en anglais `irc://irc.freenode.net/latex`.

---

<sup>1</sup>Signalons au passage la possibilité de télécharger une aide sur l'ensemble des erreurs de compilation avec  $\text{\LaTeX}$  : Franck MITTELBACH et Michel GOOSSENS, « Détecter et résoudre les problèmes », dans : *LaTeX Companion*, 2<sup>e</sup> éd., Paris : Pearson, 2006, URL : <http://www.pearson.fr/livre/?GCOI=27440100048330>, l'éditeur met gracieusement ce chapitre à disposition.

<sup>2</sup>Lointain ancêtre de Skype, MSN et autres GoogleTalk.

## G.3 Liste de discussion

En français, celles de l'association Gutenberg : <http://www.gutenberg.eu.org/?Listes-de-diffusion-gerees-par>; en anglais, la liste suivante : <http://groups.google.com/group/comp.text.tex/topics>, dont il est possible de se servir via un logiciel de gestion de « Newsgroups », comme par exemple Mozilla Thunderbird.

# Annexe H

## Glossaire

<b>Argument</b>	Paramètre passé à une commande et qui détermine en tout ou partie son résultat.
<b>Champ bibliographique</b>	Élément d'une référence bibliographique, tel que nom de l'auteur, titre, éditeur...
<b>Classe</b>	<p>Indication du type éditorial du document à produire. Son choix se fait au tout début du document <math>\LaTeX</math> via <code>\documentclass[<i>\langle options \rangle</i>]{<i>\langle classe \rangle</i>}</code>.</p> <p>La classe influence notamment le rendu final et les commandes disponibles.</p>
<b>Clef bibliographique</b>	Identifiant unique attribué à une référence bibliographique. Il est conseillé de n'y mettre que des caractères alphanumériques non accentués.
<b>Clef d'index</b>	Code indiquant la position d'une entrée dans un index. Il est nécessaire de définir une clef d'index pour les entrées comportant des caractères accentués. Les clefs d'index peuvent également servir pour des index non alphabétiques (par exemple des index par date de règne.). La syntaxe est : <code>\index{<i>\langle clef \rangle</i>@<i>\langle entrée \rangle</i>}</code> .
<b>Commande</b>	Morceau de code interprété par le compilateur pour effectuer une suite d'opérations. Une commande est un raccourci d'écriture. $\LaTeX$ propose des commandes, les packages en rajoutent et il est possible de définir ses propres commandes.

<b>Commentaire</b>	Texte qui n'est pas interprété par le compilateur. Tout ce qui est situé entre le signe % et la fin de ligne est un commentaire.
<b>Compilateur</b>	« Logiciel » chargé d'interpréter le fichier .tex pour produire un fichier .pdf. Dans ce livre nous utilisons le compilateur X <sub>Y</sub> LaTeX, plus récent que LaTeX.
<b>Compteur</b>	Nombre entier stocké dans la mémoire de l'ordinateur et manipulable par diverses commandes. Par exemple, les éléments numérotés se voient associer un compteur.
<b>Environnement</b>	Portion de document ayant une signification spécifique et qui, par conséquent, reçoit un traitement spécifique. On utilise <code>\begin{&lt;nom de l'environnement&gt;}</code> pour indiquer le début d'un environnement ; on utilise <code>\end{&lt;nom de l'environnement&gt;}</code> pour en indiquer la fin. LaTeX définit des environnements, les packages en rajoutent et il est possible de créer ses propres environnements.
<b>Flottant</b>	Élément non textuel (image ou tableau, par exemple) dont l'insertion dans le flux du texte est automatiquement calculée par LaTeX, en fonction de recommandations fournies par le rédacteur.
<b>Macro bibliographique</b>	Sous-élément d'un style bibliographique, en général chargé de gérer l'affichage d'un ou plusieurs champs.
<b>Package</b>	Chargés dans le préambule, les packages sont des fichiers qui permettent d'étendre les fonctionnalités de LaTeX.
<b>Préambule</b>	Partie du code .tex située avant <code>\begin{document}</code> . Son contenu n'est pas affiché dans le document final, mais sert à divers usages, notamment le chargement de packages.
<b>Style bibliographique</b>	Manière d'afficher une entrée bibliographique, indépendamment de son contenu. Le package <i>biblatex</i> propose un certain nombre de styles bibliographiques, qui sont personnalisables.
<b>WYSIWYG</b>	Abréviation de <i>What You See Is What You Get</i> : « ce que vous voyez est ce que vous produisez ». Se dit d'un

logiciel qui affiche à l'écran le résultat final. Les traitements de texte comme LibreOffice ou Microsoft Word sont des logiciels WYSIWYG.



# Annexe I

## Bibliographie succincte

### I.1 Manuels des packages

Nous ne listons pas ici l'ensemble des manuels des divers package abordés. Toutefois une question peut se poser : où trouver ces manuels ?

La solution la plus simple est d'ouvrir le terminal :

- Sous Mac, ouvrir l'application « Terminal » dans le dossier « Utilitaires ».
- Sous Windows, ouvrir « l'invite des commandes » dans « Démarrer » : « Tous les programmes » : « Accessoires ».
- Sous GNU/Linux, appuyer sur la touche « Windows » et taper `terminal`.

Puis de frapper dans la fenêtre qui apparaît :

```
| texdoc nomdupackage
```

Par exemple pour le package *biblatex* :

```
| texdoc biblatex
```

Frapper un retour à la ligne : le manuel du package devrait s'ouvrir avec le logiciel adéquat (généralement le manuel est au format PDF).

### I.2 Livres généralistes

Les livres sur  $\TeX$  sont pléthores. Cependant rares sont ceux qui abordent  $X_{\text{E}}\TeX$ , *polyglossia* et *biblatex*. C'est pourquoi nous conseillons de les utiliser essentiellement pour les besoins les plus avancés de mise en page.

- CHEVALIER, Céline dir., *ET<sub>X</sub> pour l'impatient*, H&K, 2009, URL : [http://www.h-k.fr/liens/tp/latex\\_pour\\_l\\_impatient.html](http://www.h-k.fr/liens/tp/latex_pour_l_impatient.html).
- LOZANO, Vincent, *Tout ce que vous avez toujours voulu savoir sur LaTeX sans jamais oser le demander*, In Libro Veritas, 2008, URL : <http://www.framabook.org/latex.html>.
- MITTELBACH, Franck et Michel GOOSSENS, *LaTeX Companion*, 2<sup>e</sup> éd., Paris : Pearson, 2006, URL : <http://www.pearson.fr/livre/?GCOI=27440100048330>.

### I.3 Livres et textes sur des points spécifiques

- COLLILIEUX, Marie-Claude, *Découverte de Beamer*, 2007, URL : <http://mcclinux.free.fr/latex/introbeamer.php>.
- MITTELBACH, Franck et Michel GOOSSENS, « Détecter et résoudre les problèmes », dans : *LaTeX Companion*, 2<sup>e</sup> éd., Paris : Pearson, 2006, URL : <http://www.pearson.fr/livre/?GCOI=27440100048330>, l'éditeur met gracieusement ce chapitre à disposition.
- *The LaTeX Graphics Companion*, Boston : Addison Wesley, 2007.
- TISSEAU, Gérard et Jacques DUMA, *TikZ pour l'impatient*, URL : <http://math.et.info.free.fr/TikZ/>.

### I.4 Sites internet

Comme pour les livres généralistes, l'intérêt en terme de contenu, par rapport à la problématique de ce livre, peut être très variable.

- MASSON, Bertrand, *Les fiches de Bébert : pour une véritable Publication Assistée par Ordinateur*, URL : <http://bertrandmasson.free.fr/>.
- ROUQUETTE, Maïeul, *Geekographie Maïeulesque : LaTeX*, URL : <http://geekographie.maieul.net/-LaTeX->.
- TeXample.net*, URL : <http://www.texample.net/>.
- TeXample.net : TikZ and PGF*, URL : <http://www.texample.net/tikz/>.

# Annexe J

## Index

### J.1 Commandes

Ne sont pas référencées les commandes situées dans les exemples colorés.  
En règle générale ces commandes sont expliquées avant ou après ces exemples.  
Les entrées en gras renvoient à la définition de la commande.

<code>\,</code> .....	169	<code>\addcomma</code> .....	126
<code>\@M</code> .....	206	<code>\addcontentsline</code> .....	30, 141
<code>\@addtoreset</code> .....	208	<code>\addtocounter</code> .....	208
<code>\@arabic</code> .....	202	<code>\addtolength</code> .....	210
<code>\@chapapp</code> .....	206	<code>\advanceline</code> .....	170
<code>\@chapter</code> .....	205, 205	<code>\Aendnote</code> .....	172
<code>\@makechapterhead</code> .....	205, 205	<code>\Afootnote</code> .....	172, 172
<code>\@makeschapterhead</code> .....	205, 205	<code>\alert</code> .....	185
<code>\@minus</code> .....	203, 239	<code>\appendix</code> .....	32, 32
<code>\@plus</code> .....	203, 212, 239	<code>\auteur</code> .....	40, 53, 144, 146
<code>\@removefromreset</code> .....	208	<code>\author</code> .....	23, 182
<code>\@schapter</code> .....	205, 205	<code>\autocite</code> .....	95, 98
<code>\@startsection</code> .....	203, 203, 204	<code>\autopar</code> .....	170, 175
<code>\&amp;</code> .....	167	<code>\avolcite</code> .....	98
<code>\langle chaîne\rangle</code> .....	139	<code>\backmatter</code> .....	32, 32
<code>\langle prefix\ranglecite</code> 95, 98, 121, 124, 128, 150		<code>\baselineskip</code> .....	238
<code>\langle prefix\ranglecites</code> .....	100	<code>\begin</code> .....	22, 23, 26, 31, 201, 248
<code>\langle titre\rangle</code> .....	29	<code>\beginnumbering</code> 165, 167, 169, 170, 175, 176	
<code>\addcolon</code> .....	126	<code>\Bfootnote</code> .....	172, 172, 174

- `\bfseries`... 204, 204, 206, 207, 217  
`\biblerefindex` ..... 163  
`\biblerefstyle` ..... 158  
`\bibleverse` ..... 161  
`\bibliography` ..... 94  
`\bibname`..... 105  
`\bibstring` ..... 129  
`\bigbreak`..... 169, 169  
`\bigskip`..... 148  
`\BRallowhyphbch` ..... 163  
`\BRforbidhyphbch` ..... 163  
  
`\c@chapter` ..... 202  
`\caption` ..... 71, 71  
`\captionsfrench` ..... 139, 139  
`\centering`..... 71  
`\cf`..... 56  
`\cfoot`..... 192  
`\chapter` . 29, 29, 192, 194, 201, 204, 205  
`\chapter*`..... 30, 104, 205  
`\chaptermark` ..... 194, 195  
`\thead`..... 192  
`\cite`..... 95, 98  
`\cline`..... 67  
`\cmd1`..... 169  
`\cmd2`..... 169  
`\columnrulewidth` ..... 178  
`\Columns` ..... 175, 176  
`\commentaire`..... 55  
`\contentsname` ..... 139, 139  
  
`\date`..... 23, 23, 182  
`\DeclareBibliographyStrings` . 119  
`\DeclareDataInheritance` .... 111  
`\DeclareFieldAlias` ..... 127  
`\DeclareFieldFormat` .... 100, 129  
`\DeclareIndexFieldFormat` 152, 153  
`\DeclareIndexNameFormat` .... 152  
`\def` ..... 139, 139, 205  
`\defbibheading` ..... 104  
`\DefineBibliographyStrings` . 119  
`\definecolor` ..... 59, 59  
  
`\documentclass` ..... 20, 20, 247  
`\dominitoc` ..... 140  
`\dotfill`..... 213, 216  
`\doublespacing` ..... 199  
  
`\edtext`..... 172, 172, 174  
`\Eendnote` ..... 172  
`\Efootnote` ..... 172  
`\emph` ..... 40, 41, 58, 183  
`\end`..... 23, 26, 31, 201, 248  
`\endfirsthead` ..... 69  
`\endfoot`..... 69  
`\endhead`..... 69  
`\endlastfoot`..... 69  
`\endnumbering` ..... 165, 167, 175  
`\endoldquotation` ..... 201  
`\endquotation` ..... 201  
`\endsanzaextra` ..... 169, 169  
`\endsub`..... 170  
`\enquote`..... 46  
`\equal`..... 174  
  
`\faketableofcontents` ..... 140  
`\fancyfoot` ..... 193  
`\fancyhead` ..... 193, 194  
`\firstlinenum` ..... 170, 170  
`\FloatBarrier` ..... 72  
`\footcite` ..... 95, 98  
`\footnote`..... 41  
`\footnoteA` ..... 172  
`\footnoteE` ..... 172  
`\footnotesize` ..... 57  
`\footrulewidth` ..... 195  
`\frame`..... 182  
`\frontmatter` ..... 32, 32  
`\fvolcite`..... 98  
  
`\gappto` ..... 139, 139  
  
`\hangingsymbol` ..... 169  
`\headrulewidth` ..... 195  
`\hfill` ..... 213, 216, 216, 217, 218  
`\hfilll`..... 217

<code>\hline</code> .....	67	<code>\tfoot</code> .....	192
<code>\hrulefill</code> .....	207, 213	<code>\thead</code> .....	192
<code>\hspace</code> .....	209	<code>\lineation</code> .....	167, 170, 176
<code>\hspace*</code> .....	209, 218	<code>\lineationR</code> .....	176
<code>\Huge</code> .....	57, 206	<code>\linenumberstyle</code> .....	172
<code>\huge</code> .....	57, 206, 207	<code>\linenumincrement</code> .....	170
<code>\hypersetup</code> .....	142	<code>\linenummargin</code> .....	171
<code>\bibblechvs</code> .....	161	<code>\linenumsep</code> .....	172, 172
<code>\bibbleverse</code> .....	161	<code>\linewidth</code> .....	238
<code>\bibblevs</code> .....	161	<code>\list</code> .....	211, 212
<code>\ifciteindex</code> .....	151	<code>\listoffigures</code> .....	139
<code>\ifcurrentfield</code> .....	153	<code>\listoftables</code> .....	139
<code>\iffieldequalstr</code> .....	120, 154	<code>\listparindent</code> .....	212
<code>\iflistundef</code> .....	126	<code>\mainmatter</code> .....	32, 32, 206
<code>\ifsingletitle</code> .....	128	<code>\makeatletter</code> .....	202, 204, 207
<code>\ifthenelse</code> .....	174	<code>\makeatother</code> .....	202, 204, 207
<code>\include</code> .....	31, 31	<code>\makeindex</code> .....	143, 144, 147, 148, 150, 152, 215
<code>\includegraphics</code> .....	61	<code>\makeschapterhead</code> .....	207
<code>\indentpattern</code> .....	50, 51	<code>\maketitle</code> .....	23, 23
<code>\index</code> .....	143, 144–147, 148, 148, 149, 152, 247	<code>\makindex</code> .....	215
<code>\indexprologue</code> .....	148	<code>\marginpar</code> .....	41, 172, 211
<code>\indexsetup</code> .....	147, 149	<code>\marginparwidth</code> .....	211
<code>\inherit</code> .....	111	<code>\markboth</code> .....	194, 194
<code>\input</code> .....	31, 31, 55	<code>\markright</code> .....	194, 194
<code>\item</code> .....	41, 42, 212	<code>\maxchunks</code> .....	176
<code>\itemindent</code> .....	212	<code>\memorydump</code> .....	177
<code>\itshape</code> .....	204, 217, 217	<code>\minitoc</code> .....	140
<code>\label</code> .....	133, 133–135, 264	<code>\mkbibnameprefix</code> .....	118
<code>\LARGE</code> .....	57	<code>\mkibid</code> .....	129, 129
<code>\Large</code> .....	57, 204, 217	<code>\mkibidem</code> .....	120
<code>\large</code> .....	57, 217	<code>\mkpageprefix</code> .....	100
<code>\Lcolwidth</code> .....	178	<code>\multicitedelim</code> .....	117
<code>\ldots</code> .....	25	<code>\multicolumn</code> .....	68
<code>\ledleftnote</code> .....	172	<code>\multirow</code> .....	68, 68
<code>\ledrightnote</code> .....	172	<code>\nameref</code> .....	134
<code>\ledsidenote</code> .....	173	<code>\nametitledelim</code> .....	129
<code>\leftmargin</code> .....	211, 212	<code>\newbibmacro*</code> .....	126
<code>\leftmark</code> .....	194, 194	<code>\newcommand</code> .....	53, 116, 139, 205
<code>\lemma</code> .....	172, 174	<code>\newcommand*</code> .....	56
<code>\let</code> .....	169, 169, 200	<code>\newcommandx</code> .....	174

<code>\newcounter</code> .....	208	<code>\printindex</code> .....	144, 148, 149
<code>\newgeometry</code> .....	210	<code>\printlist</code> .....	126, 127
<code>\newunitpunct</code> .....	116, 116, 126	<code>\printname</code> .....	127
<code>\nobreak</code> .....	206	<code>\printshorthands</code> .....	101
<code>\nocite</code> .....	104	<code>\printtext</code> .....	129
<code>\node</code> .....	63, 64	<code>\psq</code> .....	97
<code>\noindent</code> .....	210	<code>\psqq</code> .....	97
<code>\noinherit</code> .....	111	<code>\pstart</code> 166, 166, 167, 169, 170, 175, 176, 178	
<code>\nom</code> .....	23	<code>\pvolcite</code> .....	98
<code>\nopagebreak</code> .....	217	<code>\quotation</code> .....	201
<code>\nopp</code> .....	97	<code>\raggedleft</code> .....	206, 207
<code>\normalfont</code> .....	206	<code>\raggedright</code> .....	206, 207
<code>\normalsize</code> .....	57	<code>\Rcolwidth</code> .....	178
<code>\note</code> .....	185	<code>\ref</code> .....	134
<code>\numberpstartfalse</code> .....	169	<code>\refstepcounter</code> .....	208
<code>\numberpstarttrue</code> .....	169, 178	<code>\relax</code> .....	212
<code>\numlabfont</code> .....	172	<code>\renewbibmacro*</code> .....	126
<code>\oldquotation</code> .....	201	<code>\renewcommand</code> .....	116, 172
<code>\onehalfspace</code> .....	201	<code>\renewenvironment</code> .....	35
<code>\onehalfspacing</code> .....	199, 199	<code>\restoregeometry</code> .....	210
<code>\p@</code> .....	206, 212	<code>\resumenumbering</code> .....	167, 177
<code>\pageref</code> .....	134	<code>\rfoot</code> .....	192
<code>\Pages</code> .....	175, 176	<code>\rhead</code> .....	192
<code>\pagestyle</code> .....	191	<code>\rightmargin</code> .....	212
<code>\par</code> .....	206	<code>\rightmark</code> .....	194, 194
<code>\paragraph</code> .....	29, 203	<code>\Rlineflag</code> .....	179
<code>\parencite</code> .....	95, 98	<code>\rule</code> .....	213
<code>\parentext</code> .....	118	<code>\scriptsize</code> .....	57
<code>\parindent</code> .....	206, 210, 238	<code>\scshape</code> .....	207
<code>\parsep</code> .....	212	<code>\secdef</code> .....	205
<code>\parskip</code> .....	210, 238	<code>\section</code> ....	29, 182, 194, 203, 204
<code>\part</code> .....	29, 138, 207	<code>\sectionmark</code> .....	194, 195
<code>\parttoc</code> .....	140	<code>\sectionpage</code> .....	182
<code>\pausenumbering</code> .....	167, 177	<code>\setbeameroption</code> .....	186
<code>\pend</code> .....	166, 166, 167, 170, 175	<code>\setcounter</code> .....	138, 208
<code>\phantomsection</code> .....	141	<code>\setlength</code> .....	172, 178, 210
<code>\pno</code> .....	97	<code>\setline</code> .....	170
<code>\poemlines</code> .....	48	<code>\setmainfont</code> .....	198, 204, 206
<code>\printbibliography</code> .	103, 104, 105, 124, 150	<code>\setmainlanguage</code> .....	22, 33, 145
<code>\printfield</code> .....	127, 129		

<code>\setmonofont</code> .....	198	<code>\textins</code> .....	52
<code>\setotherlanguage</code> .....	34	<code>\textins*</code> .....	52
<code>\setsansfont</code> .....	198, 204	<code>\textit</code> .....	58, 217
<code>\setstanzaindents</code> ...	167, 168, 169	<code>\textlatin</code> .....	35
<code>\setunit*</code> .....	126	<code>\textsc</code> .....	54, 58
<code>\setverselinenums</code> .....	49, 49	<code>\textsubscript</code> .....	58
<code>\shorttoc</code> .....	140	<code>\textsuperscript</code> .....	58
<code>\sidenotemargin</code> .....	173	<code>\textwidth</code> .....	238
<code>\singlespace</code> .....	201	<code>\the&lt;compteur&gt;</code> .....	195, 201
<code>\singlespacing</code> .....	199	<code>\thechapter</code> ....	195, 202, 202, 206
<code>\skipnumbering</code> .....	170	<code>\thepage</code> .....	193
<code>\small</code> .....	57	<code>\therefsegment</code> .....	106
<code>\space</code> .....	126	<code>\thesection</code> .....	202
<code>\ssfamilly</code> .....	204	<code>\thispagestyle</code> .....	191, 204
<code>\stanza</code> .....	167	<code>\tiny</code> .....	57
<code>\startsub</code> .....	170	<code>\title</code> .....	23, 182
<code>\subfirstlinenum</code> .....	170	<code>\titlepage</code> .....	182
<code>\sublinenumberstyle</code> .....	172	<code>\underline</code> .....	58
<code>\sublinenumincrement</code> .....	170	<code>\usebibmacro</code> .....	125
<code>\subparagraph</code> .....	29, 138	<code>\usepackage</code> .....	22, 62, 93
<code>\subsection</code> ..	29, 104, 105, 182, 203	<code>\usetheme</code> .....	181
<code>\subsectionpage</code> .....	182	<code>\variantes</code> .....	174
<code>\subsubsection</code> .....	29, 138, 182	<code>\versehangrightsquare</code> .....	50
<code>\tableofcontents</code> ....	137, 138, 140	<code>\vin</code> .....	50
<code>\text&lt;code&gt;lang&gt;</code> .....	34	<code>\volvcite</code> .....	98
<code>\textasciicircum</code> .....	26	<code>\vspace</code> .....	206, 209
<code>\textasciitilde</code> .....	26	<code>\vspace*</code> .....	209
<code>\textbackslash</code> .....	26	<code>\xspace</code> .....	57
<code>\textbf</code> .....	58, 204, 217	<code>\z@</code> .....	203, 206, 212
<code>\textcolor</code> .....	58		
<code>\textelp</code> .....	51		
<code>\textheight</code> .....	238		

## J.2 Classes

article..... 20, 23, 26, 32  
 beamer .....

book... 20, 23, 29, 32, 193–195, 203  
 bredele..... 200, 207

## J.3 Champs bibliographiques

address . . . . .	127	labeltitle . . . . .	129
author . . . . .	86, 151	location . . . . .	88, 126, 127
bookauthor . . . . .	116	month . . . . .	88
bookpagination . . . . .	99, 100	pages . . . . .	97–100, 112
booktitle . . . . .	110, 112	pagination . . . . .	99, 100
crossref . . . . .	110	publisher . . . . .	126
date . . . . .	110	shorthand . . . . .	101
editor . . . . .	83	shorthandintro . . . . .	101
editoratype . . . . .	83	shorttitle . . . . .	129
editorbtype . . . . .	83	sortname . . . . .	108
editorctype . . . . .	83	sorttitle . . . . .	107
editortype . . . . .	83	sortyear . . . . .	108
entrysubtype . . . . .	104, 105, 111	subtitle . . . . .	86
entrytype . . . . .	120	title . 86, 107, 110, 112, 113, 129, 151	
indextitle . . . . .	151, 153	titleaddon . . . . .	112, 150, 151
issuesubtitle . . . . .	86	translator . . . . .	83
issuetitle . . . . .	86	url . . . . .	82
journal . . . . .	86	usera . . . . .	154
keyword . . . . .	106	volume . . . . .	89
keywords . . . . .	105	year . . . . .	88

## J.4 Environnements

<i>&lt;codelang&gt;</i> . . . . .	34	itemize . . . . .	42
abstract . . . . .	26	latin . . . . .	35, 200
description . . . . .	42	Leftside . . . . .	175, 176
enumerate . . . . .	41	list . . . . .	211
etaremun . . . . .	42	longtable . . . . .	69
exemple . . . . .	211, 212	oldquotation . . . . .	201
figure . . . . .	70	pages . . . . .	175
frame . . . . .	182, 186	pairs . . . . .	175
hyphenrules . . . . .	35	patverse . . . . .	50, 51
		patverse* . . . . .	51

quotation 47, 48, 200, 201, 211, 212, 238  
 quote ..... 46  
 Rightside ..... 175, 176  
 table ..... 70

tabular ..... 64, 65, 69  
 tabular\* ..... 65  
 tikzpicture ..... 63  
  
 verbatim ..... 197  
 verse ..... 48, 50

## J.5 Packages

array ..... 66, 67  
  
 babel ..... 22, 34  
 beamerarticle ..... 187  
 biblatex 76, 80, 83, 84, 90, 91, 93–98, 103, 106, 107, 112, 115–121, 128, 129, 149–152, 154, 233, 248, 251  
 bibleref ..... 157, 158, 161  
 bibleref-french ..... 157, 163  
 bidi ..... 38  
  
 caption ..... 71  
 ccaption ..... 71  
 color ..... 58  
 csquotes ..... 46, 51  
 csvpie ..... 70  
 csvtools ..... 70  
  
 datatool ..... 70  
 diagbox ..... 67  
 dpfloat ..... 71  
  
 eepic ..... 213  
 endfloat ..... 71  
 endnotes ..... 41  
 enumitem ..... 43  
 etaremune ..... 42  
 etoolbox ..... 139  
  
 fancybox ..... 213  
 fancyhdr ..... 191–195  
 fontspec ..... 21, 197, 198

framed ..... 6  
  
 geometry ..... 210, 211  
 gmverse ..... 50  
 graphicx ..... 61, 62  
 hyperref ... 134, 137, 141, 142, 147  
  
 ifthen ..... 174  
 ifthenelse ..... 54  
 imakedix ..... 215  
 imakeidx ... 143, 144, 147–151, 162  
  
 ledmac .. 51, 165, 170, 172, 175, 176  
 ledpar ..... 51, 165, 175, 176, 178  
 longtable ..... 69  
  
 makeindex ..... 143  
 mdframed ..... 6, 213  
 minitoc ..... 140, 141  
 minted ..... 6  
 multirow ..... 68  
  
 placeins ..... 72  
 poemscol ..... 51  
 polyglossia ... 22, 33–35, 71, 88, 89, 137, 139, 251  
  
 reset ..... 208  
 rotfloat ..... 71  
  
 setspace ..... 198, 199, 209  
 shorttoc ..... 140  
 showlabels ..... 133  
 SplitIndex ..... 143  
 splitindex ..... 143

---

spot . . . . .	185	totpages . . . . .	192
spreadtab . . . . .	67	verse . . . . .	48, 50, 51
subcaption . . . . .	71	xargs . . . . .	54, 174
subscript . . . . .	58	xcolor . . . . .	58, 59
tabbing . . . . .	67	xcolors . . . . .	60
tabularx . . . . .	67	xspace . . . . .	57
TikZ . . . . .	62, 64, 70, 213	xunicode . . . . .	21, 33
titleoc . . . . .	138		

# Table des matières

<b>Avant-propos</b>	<b>3</b>
Remerciement . . . . .	5
Au sujet de ce livre . . . . .	6
<b>1 Introduction</b>	<b>7</b>
1.1 Un manque important . . . . .	7
1.2 Pourquoi $\LaTeX$ ? . . . . .	8
1.2.1 Inconvénients des traitements de texte . . . . .	8
1.2.2 Avantages de $\LaTeX$ . . . . .	9
1.2.3 Qu'est-ce qu'un éditeur de texte ? . . . . .	11
1.3 TeX, LaTeX, XeTeX, XeLaTeX : points communs et différences . . . . .	12
1.4 Publics visés par cet ouvrage . . . . .	13
1.5 Comment lire ce livre . . . . .	13
<b>Premiers pas avec <math>\LaTeX</math></b>	<b>17</b>
<b>2 Commencer avec XeLaTeX</b>	<b>19</b>
2.1 Un premier document . . . . .	19
2.2 Structure d'un document $\LaTeX$ . . . . .	20
2.2.1 La classe du document . . . . .	20
2.2.2 L'appel aux packages . . . . .	21
2.2.3 Le français, langue par défaut . . . . .	22
2.2.4 Le corps du document . . . . .	22
2.2.5 Titre, auteur et date : la notion de commande . . . . .	23
2.2.6 Le corps du texte : la manière de rédiger . . . . .	24
2.2.7 Un commentaire . . . . .	26
2.2.8 La notion d'environnement . . . . .	26
2.2.9 Conclusion . . . . .	27

<b>3</b>	<b>Structurer son travail</b>	<b>29</b>
3.1	Différents niveaux de titres . . . . .	29
3.1.1	Des titres non numérotés . . . . .	30
3.2	Structurer ses fichiers . . . . .	30
3.2.1	Comment indiquer le chemin du fichier . . . . .	31
3.3	La classe <b>book</b> : structuration globale du document . . . . .	32
<b>4</b>	<b>Gérer les langues avec <i>Xunicode</i> et <i>Polyglossia</i></b>	<b>33</b>
4.1	Indiquer les changements de langue . . . . .	33
4.1.1	Pourquoi indiquer les changements de langue ? . . . . .	33
4.1.2	Commandes et environnements de changement de langue . . . . .	33
4.1.3	Indiquer un changement de langue . . . . .	34
4.1.4	Le problème du latin . . . . .	35
4.2	Saisir des textes en caractères non latins . . . . .	35
4.2.1	Les jeux de caractères : ou comment se servir de nombres pour autre chose que des mathématiques . . . . .	35
4.2.2	Concrètement . . . . .	37
4.2.3	Et les changements de sens d'écriture ? . . . . .	38
<b>5</b>	<b>Mettre en sens son document (1) : premiers pas</b>	<b>39</b>
5.1	Mettre en forme n'est pas mettre en sens . . . . .	39
5.2	Commandes de mise en sens . . . . .	40
5.2.1	Mise en valeur d'un texte . . . . .	40
5.2.2	Le paratexte : notes de bas de page et de marge . . . . .	41
5.2.3	Listes . . . . .	41
5.2.4	Imbrication des listes . . . . .	43
<b>6</b>	<b>Mettre en sens (2) : l'art de citer en LaTeX</b>	<b>45</b>
6.1	Citation dans le corps du texte . . . . .	45
6.2	Citation dans un bloc séparé . . . . .	46
6.2.1	L'environnement <b>quote</b> . . . . .	46
6.2.2	L'environnement <b>quotation</b> . . . . .	47
6.2.3	L'environnement <b>verse</b> et le package <i>verse</i> . . . . .	48
6.3	Citations tronquées et modifiées . . . . .	51
6.3.1	Citation tronquée . . . . .	51
6.3.2	Citation modifiée . . . . .	52
<b>7</b>	<b>Mettre en sens (3) : créer ses propres commandes</b>	<b>53</b>
7.1	Création d'une commande personnalisée . . . . .	53
7.1.1	Code d'une commande sur plusieurs lignes . . . . .	55
7.1.2	Étoiler <code>\newcommand</code> . . . . .	56
7.1.3	Espace après les commandes sans argument . . . . .	56
7.2	Commandes de mise en forme . . . . .	57

7.2.1	Taille des caractères . . . . .	57
7.2.2	Style de caractères . . . . .	58
7.2.3	Couleurs . . . . .	58
<b>8</b>	<b>Insérer des éléments non textuels</b>	<b>61</b>
8.1	Insérer des images . . . . .	61
8.2	Insérer des graphismes et des schémas . . . . .	62
8.3	Création de tableaux . . . . .	64
8.3.1	Syntaxe de base . . . . .	64
8.3.2	Long tableau . . . . .	69
8.4	La notion de flottants . . . . .	70
8.4.1	Choix de l'emplacement du flottant . . . . .	72
	<b>Gérer sa bibliographie avec L<sup>A</sup>T<sub>E</sub>X</b>	<b>73</b>
<b>9</b>	<b>Introduction</b>	<b>75</b>
9.1	Principe général . . . . .	75
9.2	Une triple compilation . . . . .	76
<b>10</b>	<b>Remplir sa base de données bibliographique</b>	<b>79</b>
10.1	Structure de base d'un fichier <b>.bib</b> . . . . .	79
10.2	Le choix de la clef . . . . .	80
10.2.1	Pour les œuvres contemporaines . . . . .	81
10.2.2	Pour les œuvres anciennes . . . . .	81
10.3	Les différentes types d'entrées . . . . .	81
10.4	Les différents champs possibles . . . . .	82
10.4.1	Les champs de personnes . . . . .	83
10.4.2	Champs de titre . . . . .	86
10.4.3	Champs d'informations sur la publication . . . . .	88
10.4.4	Les champs d'identification légale . . . . .	90
10.4.5	Champs d'annotations . . . . .	91
<b>11</b>	<b>Indiquer des références bibliographiques</b>	<b>93</b>
11.1	Appel du package . . . . .	93
11.2	Appel de la base de donnée bibliographique . . . . .	94
11.3	Citation d'un élément bibliographique . . . . .	95
11.3.1	Les arguments <i>⟨prenote⟩</i> et <i>⟨postnote⟩</i> . . . . .	96
11.3.2	L'argument <i>⟨postnote⟩</i> et la numérotation des pages . . . . .	96
11.4	Citation de plusieurs œuvres . . . . .	100
11.5	Choix de la forme abrégée . . . . .	101

<b>12</b>	<b>Afficher la bibliographie</b>	<b>103</b>
12.1	Affichage de la bibliographie . . . . .	103
12.1.1	Choix du style de bibliographie . . . . .	103
12.1.2	Ajout de références non citées . . . . .	104
12.2	Diviser une bibliographie . . . . .	104
12.2.1	Par catégorie de document . . . . .	104
12.2.2	Par sujet . . . . .	105
12.2.3	Par section du document . . . . .	106
12.2.4	Autres manières . . . . .	107
12.3	Tri à l'intérieur d'une bibliographie . . . . .	107
<b>13</b>	<b>Entrées hiérarchisées</b>	<b>109</b>
13.1	Principe des sous-entrées bibliographiques . . . . .	109
13.2	Sous-entrées dans la bibliographie finale . . . . .	111
13.3	Préciser les héritages de champs . . . . .	111
13.4	Divisions de source . . . . .	112
<b>14</b>	<b>Modifier les styles bibliographiques (1)</b>	<b>115</b>
14.1	Séparateurs d'unité de sens . . . . .	115
14.1.1	Un séparateur spécifique à deux unités . . . . .	117
14.2	Séparateur de références multiples . . . . .	117
14.3	Styles de certains blocs . . . . .	118
14.4	Chaînes de langue . . . . .	118
14.4.1	Introduction aux tests conditionnels . . . . .	120
<b>15</b>	<b>Modifier les styles bibliographiques (2)</b>	<b>121</b>
15.1	Que se passe-t-il lorsqu'on utilise <code>\langle prefix \rangle cite</code> . . . . .	121
15.2	Redéfinir une macro bibliographique . . . . .	122
15.3	Autres exemples : des véritables <i>op. cit.</i> . . . . .	127
	<b>Faciliter la navigation</b>	<b>131</b>
<b>16</b>	<b>Renvois internes</b>	<b>133</b>
16.1	Étiqueter des emplacements : <code>\label</code> . . . . .	133
16.2	Se servir des étiquettes . . . . .	133
16.2.1	Renvoyer à une page . . . . .	134
16.2.2	Renvoyer à un numéro de section . . . . .	134
16.2.3	Renvoyer à un titre de section . . . . .	134
16.3	Où placer la commande <code>\label</code> ? . . . . .	134
16.4	Comment nommer ses étiquettes ? . . . . .	135

<b>17</b>	<b>Sommaire et table des matières</b>	<b>137</b>
17.1	Une table simple . . . . .	137
17.2	Choix de la profondeur . . . . .	138
17.3	Table des matières ou sommaire ? . . . . .	138
17.4	Table des figures et table des tableaux . . . . .	139
17.5	Plusieurs tables des matières . . . . .	140
17.5.1	Un sommaire en plus d'une table des matières . . . . .	140
17.5.2	Sommaires partiels . . . . .	140
17.6	Des signets dans le PDF : le package <i>hyperref</i> . . . . .	141
17.6.1	Signet de navigation . . . . .	141
17.6.2	Réglage des liens . . . . .	142
<b>18</b>	<b>Index</b>	<b>143</b>
18.1	Faire un index simple avec <i>imakeidx</i> . . . . .	143
18.1.1	Principe de base . . . . .	143
18.1.2	Allons plus loin . . . . .	144
18.1.3	Quelques options du package <i>imakeidx</i> . . . . .	147
18.2	Faire plusieurs index . . . . .	148
18.2.1	Définir ses index . . . . .	148
18.2.2	Indexer son texte . . . . .	148
18.2.3	Imprimer les index . . . . .	149
18.3	Indexer ses sources . . . . .	149
18.3.1	Premier essai . . . . .	150
18.3.2	Création d'un index spécifique . . . . .	150
18.3.3	Modifications des macros de <i>biblatex</i> . . . . .	150
18.3.4	Format d'indexation <i>biblatex</i> . . . . .	152
18.3.5	Compilation et concaténation des index . . . . .	153
18.3.6	Raffinement . . . . .	154
18.3.7	Résumé des diverses compilations . . . . .	154
	<b>Autres outils pour les sciences humaines</b>	<b>155</b>
<b>19</b>	<b>Gestion des références bibliques avec <i>Bibleref</i></b>	<b>157</b>
19.1	Chargement du package et options . . . . .	157
19.1.1	Forme catholique ou protestante . . . . .	157
19.1.2	Styles de références . . . . .	158
19.2	Référencer un passage biblique . . . . .	159
19.3	Indexer une référence . . . . .	161
19.3.1	Tri dans l'index . . . . .	161
19.3.2	Un index à part . . . . .	162
19.4	Espaces insécables . . . . .	163

<b>20</b>	<b>Édition critique et textes parallèles</b>	<b>165</b>
20.1	Faire une édition critique avec <i>ledmac</i> . . . . .	165
20.1.1	Numéroter les lignes d'un texte . . . . .	165
20.1.2	Aller plus loin . . . . .	169
20.1.3	L'apparat critique . . . . .	172
20.2	Mettre deux textes en vis-à-vis : le package <i>ledpar</i> . . . . .	175
20.2.1	Principes . . . . .	175
20.2.2	Affiner la présentation . . . . .	178
<b>21</b>	<b>Présentations avec <math>\LaTeX</math> : Beamer</b>	<b>181</b>
21.1	Préambule et premières diapositives . . . . .	181
21.2	Diapositives de titre . . . . .	182
21.3	Affichage différé : les couches Beamer . . . . .	183
21.4	Mise en valeur . . . . .	184
21.4.1	Blocs . . . . .	184
21.4.2	La commande <code>alert</code> . . . . .	185
21.5	Notes de conférence . . . . .	185
21.6	Écrire son article dans le même fichier que la présentation . . . . .	186
21.6.1	Diapositives ou article ? . . . . .	187
21.6.2	Automatiser les réglages . . . . .	187
	<b>Quelques éléments de mise en forme</b>	<b>189</b>
<b>22</b>	<b>Personnaliser en-têtes et pieds de pages</b>	<b>191</b>
22.1	Utiliser l'un des styles standards . . . . .	191
22.2	Premiers exemples avec <i>fancyhdr</i> . . . . .	192
22.3	Pages recto verso et alternance gauche-droite . . . . .	193
22.4	Titres dans l'en-tête : le mécanisme des marqueurs . . . . .	193
22.4.1	Appeler les marqueurs dans les styles fancy . . . . .	194
22.4.2	Redéfinir <code>\chaptermark</code> et <code>\sectionmark</code> . . . . .	195
22.5	Filet d'en-têtes et de pieds de pages . . . . .	195
<b>23</b>	<b>Apparence des textes</b>	<b>197</b>
23.1	Police de caractères . . . . .	197
23.2	Interlignes . . . . .	198
23.2.1	« Commande à arguments » et « commande de bascule » . . . . .	199
23.2.2	Environnements de changement d'interligne . . . . .	199
23.2.3	Redéfinir un environnement : <code>quotation</code> . . . . .	200
23.3	Personnaliser les titres . . . . .	201
23.3.1	Redéfinir la numérotation . . . . .	201
23.3.2	Définir l'apparence : sections et niveaux inférieurs . . . . .	203
23.4	Définir l'apparence : chapitres et niveaux supérieurs . . . . .	204

23.5	Manipuler les compteurs . . . . .	208
<b>24</b>	<b>Éléments de mise en page</b>	<b>209</b>
24.1	Espaces . . . . .	209
24.2	Longueurs de mise en page . . . . .	209
24.3	Marges . . . . .	210
24.4	Textes en retrait . . . . .	211
24.5	Trait horizontal . . . . .	213
<b>25</b>	<b>Formater son index</b>	<b>215</b>
25.1	Comment faire . . . . .	215
25.2	Quelques commandes . . . . .	216
	 <b>Annexes</b>	 <b>219</b>
<b>A</b>	<b>Installer <math>\LaTeX</math></b>	<b>221</b>
A.1	La notion de distribution . . . . .	221
A.2	Installer TeX Live sous Mac Os X . . . . .	221
A.2.1	Les caractères $\LaTeX$ sous Mac Os X . . . . .	222
A.3	Installer TeX Live sous GNU/Linux . . . . .	223
A.3.1	Au préalable . . . . .	223
A.3.2	Installation et configuration . . . . .	224
A.4	MiKTeX sous Windows . . . . .	224
A.4.1	Installation . . . . .	225
A.5	Mettre à jour les packages . . . . .	225
A.5.1	Sous Mac OS X . . . . .	226
A.5.2	Sous Linux . . . . .	226
A.5.3	Sous Windows . . . . .	227
<b>B</b>	<b>Quelques logiciels pour travailler avec <math>\LaTeX</math></b>	<b>229</b>
B.1	Éditeurs de texte spécialisés en $\LaTeX$ . . . . .	229
B.1.1	TeXMaker (Multi-plateformes) . . . . .	229
B.1.2	TeXWorks (Multi-plateformes) . . . . .	229
B.1.3	TeXShop (Mac) . . . . .	230
B.2	Logiciels de gestion bibliographique au format <b>.bib</b> . . . . .	230
<b>C</b>	<b>Introduction à la ligne de commande</b>	<b>231</b>
C.1	La notion de répertoire courant . . . . .	231
C.2	Mac OS X et Linux . . . . .	231
C.3	Windows . . . . .	232
C.4	Trouver les fichiers standards . . . . .	233

<b>D</b>	<b>Faciliter les compilations avec Latexmk</b>	<b>235</b>
D.1	Principe . . . . .	235
D.2	Adaptation pour XeLaTeX . . . . .	235
D.3	Adaptation pour l'index des sources primaires . . . . .	236
<b>E</b>	<b>Unités de mesure en <math>\LaTeX</math></b>	<b>237</b>
E.1	Unités de mesure . . . . .	237
E.2	Longueurs relatives . . . . .	238
E.3	Élasticité . . . . .	238
<b>F</b>	<b>Travail collaboratif : les logiciels de suivi des révisions</b>	<b>241</b>
F.1	Principe . . . . .	241
F.2	Mise en pratique . . . . .	242
<b>G</b>	<b>Trouver de l'aide</b>	<b>245</b>
G.1	Forums internet . . . . .	245
G.2	Messagerie instantanée . . . . .	245
G.3	Liste de discussion . . . . .	246
<b>H</b>	<b>Glossaire</b>	<b>247</b>
<b>I</b>	<b>Bibliographie succincte</b>	<b>251</b>
I.1	Manuels des packages . . . . .	251
I.2	Livres généralistes . . . . .	251
I.3	Livres et textes sur des points spécifiques . . . . .	252
I.4	Sites internet . . . . .	252
<b>J</b>	<b>Index</b>	<b>253</b>
J.1	Commandes . . . . .	253
J.2	Classes . . . . .	257
J.3	Champs bibliographiques . . . . .	257
J.4	Environnements . . . . .	258
J.5	Packages . . . . .	259