Chapitre troisième
Utilisation avancée

6 Tables

Pas les tableaux, hein, les tables. Ici, on parlera des tables des matières, des sections, et de comment tout cela se met en forme. Après un rapide rappel du mode de fonctionnement de LaTeX sur ce thème, on étudiera un certain nombre de packages qui permettent de modifier tout ça.

6.1 LaTeX et les tables

Ce que LaTeX propose pour personnaliser la mise en forme des sections laisse franchement à désirer. Disons qu'on peut faire beaucoup de chose (même pas tout, en tous cas pas sans changer radicalement ce que fait LaTeX), mais qu'il faut absolument savoir programmer en LaTeX, c'est-à-dire qu'il faut avoir lu et compris le chapitre neuvième page 571. Si tu n'as pas envie de te lancer dans d'aussi grands travaux, passe directement aux sections suivantes qui montrent comment, en utilisant des packages bien choisis, on peut faire largement aussi bien que ce que propose LaTeX en standard.

6.1.1 Les titres

LaTeX fournit une commande magique qui est utilisée par toutes les classes de document standard pour définir les commandes de sectionnement. Cette commande magique est `\@startsection`[153].

`\@startsection`{⟨*nom*⟩}{⟨*niveau*⟩}{⟨*indentation*⟩}{⟨*avant*⟩}{⟨*après*⟩}{⟨*fonte*⟩}

Le premier argument est le nom de ce type de section. Le nom sera utilisé par la suite pour faire un certain nombre de calculs dans ton dos. En particulier, le {⟨*nom*⟩} est utlisé comme nom du compteur à incrémenter et à afficher (la commande pour afficher sera `\the`⟨*nom*⟩). De même il sera utilisé pour déterminer la commande, `\`⟨*nom*⟩`mark`, qui se chargera de placer les marques. De même, l'entrée en table des matières portera le {⟨*nom*⟩} comme type.

Le deuxième argument est le niveau de ce type de section. C'est un entier. C'est ce qui permet de déterminer si ce type de section doit être numéroté (les sections jusqu'au niveau `secnumdepth` sont numérotées).

Le troisième argument est l'indentation à utiliser. C'est une longueur. Si elle est négative, le titre dépassera dans la marge.

Le quatrième argument est la taille du blanc à laisser avant le titre. Plus exactement, si cette taille est négative, alors le premier paragraphe après le titre ne sera pas indenté. C'est la valeur absolue de ce paramètre qui est prise comme taille du blanc vertical avant le titre. Comme la commande `\@startsection` commence un nouveau paragraphe (et termine l'éventuel précédent), il y a également `\parskip` d'ajouté au blanc vertical avant le titre (en général, `\parskip`, qui est le blanc à mettre au début de chaque paragraphe, est de taille nulle).

[153] Comme son nom comporte un `@`, normalement, elle ne doit être utilisée que pour écrire un package ou une nouvelle classe de documents. Tu ne dois normalement pas t'en servir dans ton document.

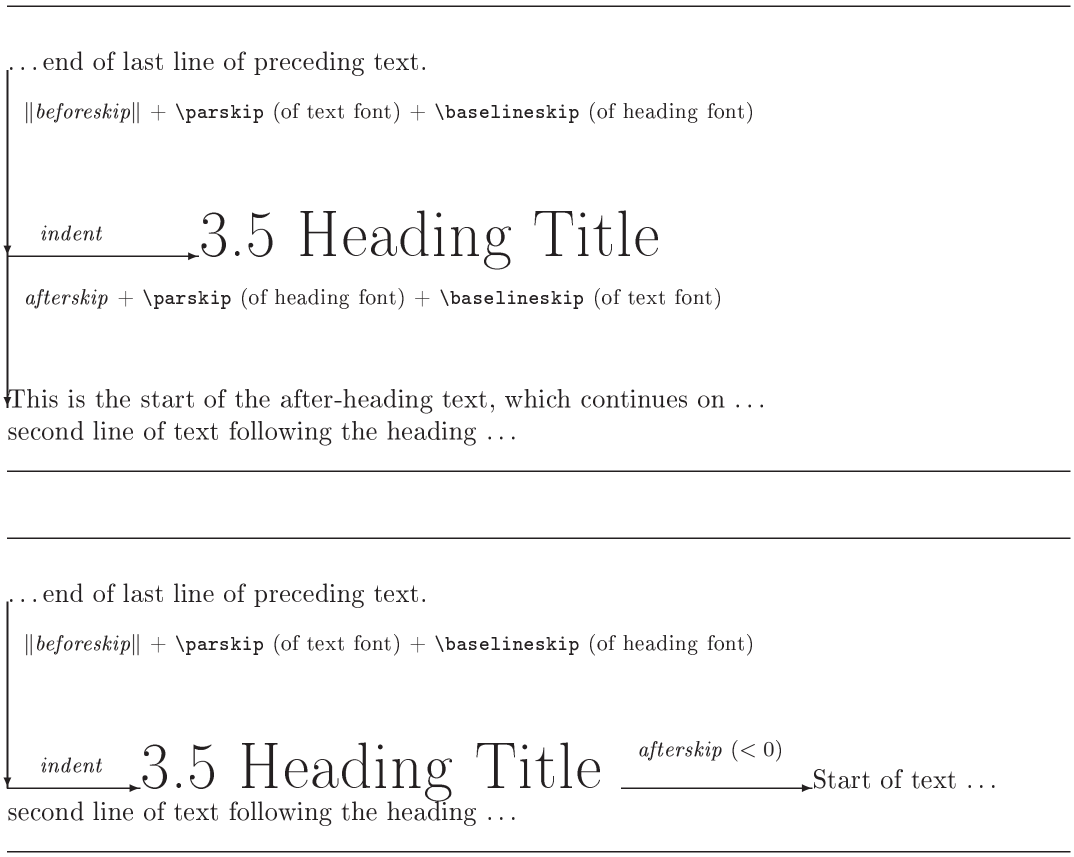


FIG. 6.1: Forme générale d'un titre de section produit par `\@startsection`

Le cinquième argument est la taille du blanc à laisser après le titre. Si sa valeur est positive, le titre est *hors ligne*, c'est-à-dire qu'il est seul sur sa ligne, et ce blanc est alors un blanc vertical. Si sa valeur est négative, le titre est *en ligne*, c'est-à-dire que le texte commencera directement après le titre, sur la même ligne, et il s'agit alors d'un blanc horizontal. Là encore, quand le titre est hors ligne, la distance `\parskip` est ajoutée. Un effet tout à fait pénible de ces deux arguments dont le signe est important est qu'on ne peut pas laisser moins que `\parskip` avant et après un titre. En effet, si tu voulais réduire en mettant une valeur négative, que ce soit pour {⟨*après*⟩} ou pour {⟨*avant*⟩}, alors tu changerais simplement la mise en forme du titre, mais pas son espacement.

Le sixième argument indique quelles commandes sont utilisées pour définir la mise en forme du titre. Tu peux y mettre par exemple, des commandes de changement de fonte et/ou de taille.

La figure 6.1 présente les différents paramètres utilisés par la commande `\@startsection`.

Quand cette commande est utilisée, par exemple dans les classes standard, ça ressemble à ça (extrait du source de la classe **article**) :

```
\newcommand\section{\@startsection {section}{1}{\z@}%
                                   {-3.5ex \@plus -1ex \@minus -.2ex}%
                                   {2.3ex \@plus.2ex}%
```

```
\renewcommand{\thesection}
  {\Roman{section}}
\section{Section un}
\subsection{Sous-section}
\label{ss-sect}

Ce texte sert \‘a remplir.
Ici une r\’ef\’erence: \ref{ss-sect}.
```

I Section un

I.1 Sous-section

Ce texte sert à remplir. Ici une référence : I.1.

FIG. 6.2: Personnalisation d'un compteur de section

```
{\normalfont\Large\bfseries}}
```

$\boxed{\texttt{\textbackslash section[}\langle\textit{titre court}\rangle\texttt{]\{}\langle\textit{titre long}\rangle\texttt{\}}}$

$\boxed{\texttt{\textbackslash section*\{}\langle\textit{titre long}\rangle\texttt{\}}}$

La commande `\@startsection` consommera donc des arguments supplémentaires, après les six qu'on vient d'exposer. Si le sixième argument est suivi d'une étoile, alors elle produit une section non numérotée qui n'est pas envoyée en table des matières ; sinon la section sera numérotée (si son niveau est moins que `secnumdepth`) et envoyée en table des matières. Le version sans étoile prend un argument optionnel qui est envoyé en table des matières et utilisé comme titre courant à la place du titre de la section. En général on fait ça pour avoir un titre plus court, pour éviter qu'il ne déborde de l'en-tête qui contient le titre courant.

6.1.2 Profondeur de numérotation

La commande `\@startsection`, comme tu viens de le voir, prend en deuxième paramètre un numéro qui donne la profondeur du titre en question. Tous les titres ont un numéro, quelle que soit leur profondeur ; mais pour certains titres, ce numéro est affiché et pour d'autres pas. En fait, LaTeX affiche le numéro des titres dont la profondeur est inférieure ou égale à la valeur du compteur `secnumdepth`. Si ce compteur vaut 1, tous les titres jusqu'à `\section` inclus auront un numéro affiché, les titres en dessous, à partir de `\subsection` n'auront pas de numéro affiché.

Si tu veux changer la valeur de ce compteur, par exemple pour numéroter jusqu'à `\paragraph` inclus, tu peux faire :

```
\setcounter{secnnumdepth}{4}
```

En théorie, d'après les modes d'emploi officiels, tu n'es supposé faire ça que dans le préambule de ton document (avant le `\begin{document}`) ; dans la pratique si tu fais ça en cours de document, par exemple pour changer la profondeur de numérotation d'un chapitre à l'autre, ça se passe bien.

6.1.3 Personnaliser les titres

Maintenant que tu connais un peu mieux la commande qui se trouve derrière quasiment tous les titres de section de LaTeX[154], tu peux les personnaliser assez facilement, soit en te

[154] Les chapitres, par exemple, sont produit de manière très différente. Dans la classe **report**, par exemple, ils sont le niveau de sectionnement 0, et ils n'utilisent pas la commande `\@startsection`.

```
\makeatletter
\renewcommand{\@seccntformat}[1]{%
  \fbox{%
    \textcolor{red}{%
      \csname the#1\endcsname
    }%
  }\quad
}
\renewcommand{\thesection}
  {\Roman{section}}
\section{Section un}
\subsection{Sous-section}
\label{ss-sect}

Ce texte sert \`a remplir.
Ici une r\'ef\'erence: \ref{ss-sect}.
```

$\boxed{\text{I}}$ **Section un**

$\boxed{\text{I.1}}$ **Sous-section**

Ce texte sert à remplir. Ici une référence : I.1.

FIG. 6.3: Personnalisation de la mise en forme d'un compteur de section

---

faisant ta propre classe de documents, soit simplement dans un package à toi, soit encore dans le préambule de ton document.

Pour comprendre comment utiliser cette commande, il vaut mieux avoir lu au moins le début du chapitre neuvième page 571 qui traite de la programmation. Tu y apprendras entre autre comment utiliser une commande qui, comme celle-là, a un `@` dans son nom, et comment créer un package et/ou une classe.

Une autre personnalisation classique, et qui se fait très différemment, est de changer la façon dont les titres de section sont numérotés. Une première approche, simple, est de redéfinir les commande `\the`⟨*compteur*⟩, comme c'est expliqué au chapitre neuvième. Ainsi, pour numéroter les sections en chiffres romains, on peut faire :

```
\renewcommand{\thesection}{\Roman{section}}
```

Ce qui te numérote les sections en chiffres romains, et adapte automatiquement la numérotation des sous-sections. Cette mise en forme du compteur sera également utilisée pour les références, comme te le montre l'exemple de la figure 6.2 page précédente. Par contre, ça montre aussi la limite du système : si je veux changer la mise en forme du compteur, par exemple pour l'encadrer ou l'écrire en rouge, je ne peux pas utiliser cette méthode, le résultat ne serait pas bon : si le numéro de section est $\boxed{1}$[155], alors le numéro de sous-section sera $\boxed{1}$.1, ainsi que les références. Ce qui n'est pas, a priori, ce qu'on cherche.

`\@seccntformat{`⟨*nom*⟩`}`

Ce qu'il faut faire, pour ce type de modification, c'est re-définir la commande `\@seccntformat`. Cette commande est appelée par `\@startsection` pour afficher le numéro de section, et reçoit en premier argument le *nom* du compteur à afficher. Sa définition par défaut est à peu près celle-ci[156] :

```
\newcommand\@seccntformat[1]{\csname the#1\endcsname\quad}
```

---

[155] Pour les gens qui n'ont pas la couleur, c'est encadré *et* rouge.

[156] Pour comprendre la syntaxe de `\csname`, il vaut mieux te reporter au chapitre neuvième. En gros, quand on dit :

```
\csname toto\endcsname
```

C'est ce que j'ai utilisé dans l'exemple de la figure 6.3 page ci-contre.

On peut aussi redéfinir cette commande pour que les numéros de section dépassent dans la marge. La définition serait alors quelque chose comme ça :

```
\renewcommand\@seccntformat[1]{%
  \protect\makebox[0pt][r]{%
    \csname the#1\endcsname\quad}
}
```

C'est aussi une bonne solution si on veut mettre un séparateur entre le numéro de la section et le titre, par exemple mettre un point après le numéro de section.

6.1.4 La table des matières

La table des matières est gérée, en gros, par quelques commandes. La principale, mais qu'on n'utilise jamais directement, est `\contentsline`, c'est cette commande qui est utilisée dans le fichier `.toc`.

`\contentsline{`⟨*nom*⟩`}{texte}{`⟨*page*⟩`}`

À chaque fois que tu commences une section, ou que tu poses un `\caption` dans un flottant, une ligne de ce type là est ajoutée dans le fichier de table des matières correspondant (le fichier `.toc`[157] pour la table des matières, `.lof`[158] pour la liste des figures, etc.

Si la section concernée est numérotée, alors le `{texte}` commence par un appel à la commande `\numberline` qui est chargée de mettre en forme le numéro de la section dans la table des matières.

Ainsi, on trouvera classiquement dans un fichier de table des matières des choses comme :

```
\contentsline {section}{\numberline {1}Ma section}{2}
```

L'argument {⟨*nom*⟩} est le même que celui qui est passé en premier à `\@startsection`. La mise en forme d'une ligne de la table des matières est assurée par la commande `\l@`⟨*nom*⟩, qui doit être définie en conséquence, et doit prendre deux arguments : le premier est le `{texte}`, et le second est la {⟨*page*⟩}.

`\addcontentsline{`⟨*extension*⟩`}{`⟨*nom*⟩`}{`⟨*texte*⟩`}`

La commande `\addcontentsline` sert à ajouter une entrée dans la table des matières (un appel à `\contentsline`). Le numéro de page[159] sera ajouté automatiquement.

`\@dottedtocline{`⟨*niveau*⟩`}{`⟨*indentation*⟩`}{`⟨*larg. numéro*⟩`}{texte}{`⟨*page*⟩`}`

Les classes standard de LaTeX se basent sur la commande `\@dottedtocline` pour mettre en forme leur table des matières. La figure 6.4 page suivante présente le rôle des différents arguments de cette commande, ainsi que le rôle de quelques paramètres globaux.

Le ⟨*niveau*⟩ est comparé avec `tocdepth` (dans un table des matières, `lofdepth` pour une liste des figures, etc) pour savoir si oui ou non ce titre doit apparaître dans la table.

ça revient à appeler la commande `\toto`. L'appel que nous avons ici :
`\csname the#1\endcsname`
si l'argument est `section` revient à appeler `\thesection`.

[157] Table of contents, littéralement.

[158] List of figures.

[159] Le bon, ce qui n'est pas simple. Fort heureusement TeX prévoit un mécanisme qui permet que les écritures dans les fichiers puissent être retardées jusqu'à la fin de la page en cours. C'est le même mécanisme qui est utilisé pour les références, ou pour les marques.

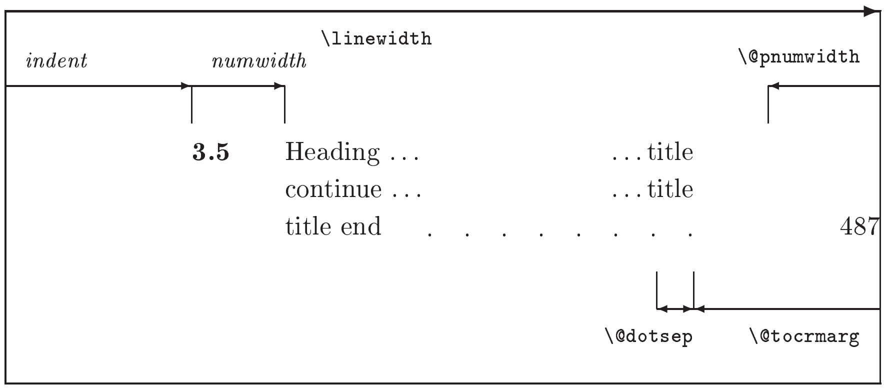


FIG. 6.4: Forme générale d'une entrée de la table des matières

---

Curieusement, ce n'est pas forcément le même que pour l'appel à `\@startsection`. En effet on peut imaginer que tu définisses ta commande `\section` avec un niveau de 1, et que dans `\l@section` tu donnes un niveau de 3. Je n'en vois pas l'intérêt, à vrai dire. Il est donc plus prudent et plus raisonnable de reprendre le *même* niveau dans les deux commandes.

Voilà comment est déclarée la commande `\l@subsection` dans la classe **article** :

```
\newcommand*\l@subsection{\@dottedtocline{2}{1.5em}{2.3em}}
```

Si tu décides, par exemple, de numéroter tes sections en chiffres romains, il te faudra probablement redéfinir l'ensemble de ces commandes, simplement parce qu'elles prévoient une largeur de numéro de section qui n'est pas suffisante pour des chiffres romains. En général, elles prévoient jusqu'à deux chiffres, ce qui est bien pour la plupart des documents (99 sections, ça fait déjà un document volumineux et fouilli). Mais en chiffres romains, rien que 18, ça prend d'avantage de place (XVIII).

6.1.5 Profondeur de la table

LaTeX envoie vers la table des matières, c'est-à-dire vers le fichier `.toc`, *tous* les niveaux de titre, sauf ceux qui sont produits par les versions étoilées des commandes (`\section*` par exemple). Ces titres seront numérotés, dans le document comme dans la table des matières, en fonction du compteur `secnumdepth`. Par contre, la commande `\tableofcontents`, ou plus exactement la commande `\@dottedtocline`, comme nous venons de le voir, n'affichera dans la table des matières que les niveaux de titres qui sont de niveau inférieur ou égal à la valeur du compteur `tocdepth` (ou `lofdepth` pour la liste des figures, ou `\lotdepth` pour la liste des tableaux). Ces compteurs indiquant la profondeur ne peuvent être positionnés qu'avant la table des matières. Par exemple, pour indiquer que tu ne veux voir apparaître que les chapitres dans la table :

```
\setcounter{tocdepth}{0}
\tableofcontents
```

6.2 Le package `indentfirst`

Un petit

```
\usepackage{indentfirst}
```

te permettra de résoudre ce difficile problème avec LaTeX qui est qu'il n'indente pas le premier paragraphe après un titre alors qu'on ne voudrait pas toujours qu'il en soit ainsi. C'est très bourrin, mais ça marche impeccable, y'a qu'à voir ce document.

Ce package nous est offert par David Carlisle [29].

Ce que fait ce package est fait par défaut par l'option `frenchb` de `babel`. Il n'est normalement pas utile de l'appeler, puisque tu as appelé `babel` pour que ton document soit en français.

6.3 Citations et titre : le package `epigraph`

Ce package, que nous devons à Peter Wilson [222], permet de mettre en forme des citations autour des titres de chapitre ou de section, comme ça se voit parfois dans la littérature.

Le principe est relativement simple. Le package fournit une commande `\epigraph` qui prend deux arguments, et un environnement `epigraphs` qui ressemble à une liste et qui contient des `\qitem` qui prennent deux arguments. L'environnement mettra en forme les `\qitem` comme s'il s'agissait d'autant d'`\epigraph`.

`\epigraph{`⟨*texte*⟩`}{`⟨*auteur*⟩`}`

Spontanément, la commande `\epigraph` met en forme le ⟨*texte*⟩ dans une `minipage` de largeur `\epigraphwidth`. Elle utilise l'environnement dont le nom est dans `\textflush` pour choisir la justification du texte de la citation. Par défaut cette commande est définie pour utiliser l'environnement `flushleft` pour que le texte de la citation ne soit pas justifié (ce genre de citation est souvent mise en forme sur une largeur faible, si on justifie, ça va faire des grands blancs). Le nom de l'auteur est mis en forme dans un environnement dont le nom est dans `\sourceflush`, par défaut `flushright`.

Voyons un petit exemple avant d'aller plus loin :

```
\epigraph{On n'oublie rien de rien,\\
          on s'habitue, c'est tout.}
         {\textit{On n'oublie rien}\\
          Jacques Brel}
```

produira :

On n'oublie rien de rien,
on s'habitue, c'est tout.

On n'oublie rien
Jacques Brel

La `minipage` contenant la citation est elle-même dans un environnement dont le nom est dans `\epigraphflush`, dont la valeur par défaut est `flushright`. On note bien, sur le résultat de l'exemple : la `minipage` est à droite de la page (`flushright`), le texte à l'intérieur est au fer à gauche (`flushleft`), et le nom de l'auteur est au fer à droite (`flushright`). La fonte utilisée est fixée par `\epigraphsize`, et l'épaisseur du trait entre la citation et le nom de l'auteur par `\epigraphrule`. Par défaut, les valeurs sont celles-là :

```
\newcommand{\epigraphsize}{\small}
\setlength{\epigraphrule}{.4pt}
```

Les deux longueurs `\beforeepigraphskip` et `\afterepigraphskip` sont les tailles des blancs laissés respectivement avant et après l'ensemble de la `minipage`.

```
% Classe report, option a5paper, geometry avec une marge
% de 1.5cm
\usepackage{epigraph}
\chapter{Le titre du chapitre}
\setlength\epigraphwidth{.6\linewidth}
\epigraphhead{%
  \epigraph{%
  Rien n'est jamais acquis \`a l'homme Ni sa force\\
  Ni sa faiblesse ni son c\oe{}ur Et quand il croit\\
  ouvrir ses bras son ombre est celle d'une croix\\
  Et quand il veut serrer son bonheur il le broie\\
  Sa vie est un \'etrange et douloureux divorce\\
  Il n'y a pas d'amour heureux
  }{\textit{Il n'y a pas d'amour heureux}\\
  Georges Brassens}
}
```

Rien n'est jamais acquis à l'homme Ni sa force
Ni sa faiblesse ni son cœur Et quand il croit
ouvrir ses bras son ombre est celle d'une croix
Et quand il veut serrer son bonheur il le broie
Sa vie est un étrange et douloureux divorce
Il n'y a pas d'amour heureux

Il n'y a pas d'amour heureux
Georges Brassens

Chapitre 1

Le titre du chapitre

FIG. 6.5: Exemple d'utilisation du package `epigraph`

Un des problèmes classiques pour ce type de citation est de pouvoir les placer au tout début des chapitres, *avant* le titre du chapitre. Ce n'est pas simple, parce que la commande `\chapter` change de page toute seule. Placer cette citation *après* le titre du chapitre, c'est enfantin, mais *avant*, ce n'est pas simple.

`\epigraphhead[`⟨*offset*⟩`]{`⟨*texte*⟩`}`

Le package contourne cette difficulté en s'attaquant au style de page. La commande `\epigraphhead` rattache son argument ⟨*texte*⟩ à l'en-tête de la page en cours en le décalant vers le bas de ⟨*offset*⟩. Il fait ça en définissant un nouveau style de page, epigraph, qu'il applique à la page sans trop te demander ton avis. Si tes premières pages de chapitre sont comme celles prévues par défaut par LaTeX c'est-à-dire avec le numéro de page en pied de page au centre, alors tout va bien. Mais si ce n'est pas le cas, par exemple si tu as utilisé le package `rplain`, alors ça ne marche pas : l'effet de `rplain` est annulé, puisqu'il agit sur les pages en style plain et que ta page de chapitre n'est plus dans ce style-là.

L'exemple de la figure 6.5 montre comment utiliser `\expigraphhead` et le résultat que ça peut produire.

Quand on veut placer une citation à cet endroit-là, il peut se poser quelques problèmes. Par exemple que la citation soit trop longue, et vienne s'écrire sur le titre du chapitre. Un début de solution peut être de faire comme dans l'exemple, d'élargir un peu la place réservée à la citation. Une autre solution peut être de décaler vers le bas le titre du chapitre, avec la commande `\dropchapter` qui est fournie par `epigraph`. Un appel à `\dropchapter` décalera vers le bas les titres de tous les chapitres jusqu'à l'appel de `\undodrop`. Tu peux donc faire comme ça :

```
\dropchapter{3cm}
\chapter{Le titre}
\epigraphhead{Tr\`es long...}
\undodrop
```

pour ne décaler que *ce* titre de chapitre, sans déplacer les autres.

Enfin, une dernière commande intéressante fournie par ce package, mais qui n'a pas grand chose à y faire, est la commande `\cleartoevenpage`. Elle fait en gros comme `\cleardoublepage`, mais en s'assurant d'arriver sur une page de *gauche*. Elle prend un argument optionnel, qui sera le contenu de la page vide, si elle est amenée à produire une page vide.

Ainsi, si tu veux que tes têtes de chapitre soient sur des pages de droite, comme c'est le cas par défaut pour la classe de document **book** en recto-verso, mais que tu veux utiliser la page de gauche pour mettre une photo, ou une citation, ou que sais-je encore qui soit en rapport avec le nouveau chapitre, tu peux procéder comme ça :

```
\cleartoevenpage[\thispagestyle{empty}]
% Ma citation, je suis sur la page de gauche
% avant la tete de chapitre
\thispagestyle{empty}
\epigraph{...}{...}
\chapter{Mon chapitre}
```

Tu as tout intérêt, pour la page de gauche, à choisir un style de page qui n'affiche pas le titre du chapitre en cours dans l'en-tête, puisque, pour LaTeX, tu es toujours dans le chapitre précédent.

6.4 Changer les têtes de chapitre : le package `fncychap`

Ce package, que nous devons à Ulf A. Lindgren [125], définit quelques mises en forme de têtes de chapitre différentes de ce qu'on peut trouver dans les classes de document standard de LaTeX.

Sur le principe, c'est assez simple. Le package prévoit six options qui correspondent aux six styles de chapitres pré-définis. Ces options sont les suivantes : Sonny (exemple figure 6.6 page suivante), Lenny (figure 6.7), Conny (figure 6.8), Glenn (figure 6.9), Rejne (figure 6.10), et Bjarne (figure 6.11). Si un de ces style correspond à ce que tu cherches, il n'y a pas à aller chercher plus loin : tu utilises l'option correspondante, comme montré dans les exemples.

Le package prévoit cependant quelques commandes permettant de modifier ces styles pour les personnaliser. Il considère qu'il y a trois morceaux de texte dans une tête de chapitre : le nom du chapitre (ou « chapter name », en français c'est le mot chapitre, en anglais c'est le mot chapter, en allemand kapitel, etc), le numéro du chapitre, et le titre du chapitre.

Le package définit des commandes pour modifier des détails de mise en forme. Toutes ne sont pas véritablement intéressantes à utiliser cependant.

La première série de commandes permet de dire si le nom du chapitre (le mot chapitre) doit être en capitales ou pas. Par défaut, `babel`, qui se charge de le franciser, le définit avec une majuscule. La commande `\ChNameUpperCase` indique de l'écrire tout en capitales (comme dans le stype Conny par exemple) ; la commande `\ChNameLowerCase` indique de l'écrire tout en minuscules ; et la commande `\ChNameAsIs` indique de ne pas y toucher. La

- ```
  \usepackage[Sonny]{fncychap}
  \chapter{Le titre}
  Ce texte sert \'a remplir...
  ```

CHAPITRE 1

Le titre

Ce texte sert à remplir...

FIG. 6.6: Exemple du style Sonny pour `fncychap`

- ```
  \usepackage[Lenny]{fncychap}
  \chapter{Le titre}
  Ce texte sert \'a remplir...
  ```

Chapitre 1

Le titre

Ce texte sert à remplir...

FIG. 6.7: Exemple du style Lenny pour `fncychap`

- `\usepackage[Conny]{fncychap}`
  `\chapter{Le titre}`
  `Ce texte sert \'a remplir...`

CHAPITRE 1

Le titre

Ce texte sert à remplir...

FIG. 6.8: Exemple du style Conny pour `fncychap`

- `\usepackage[Glenn]{fncychap}`
  `\chapter{Le titre}`
  `Ce texte sert \'a remplir...`

CHAPITRE 1

Le titre

Ce texte sert à remplir...

FIG. 6.9: Exemple du style Glenn pour `fncychap`

```
\usepackage[Rejne]{fncychap}
\chapter{Le titre}
Ce texte sert \`a remplir...
```

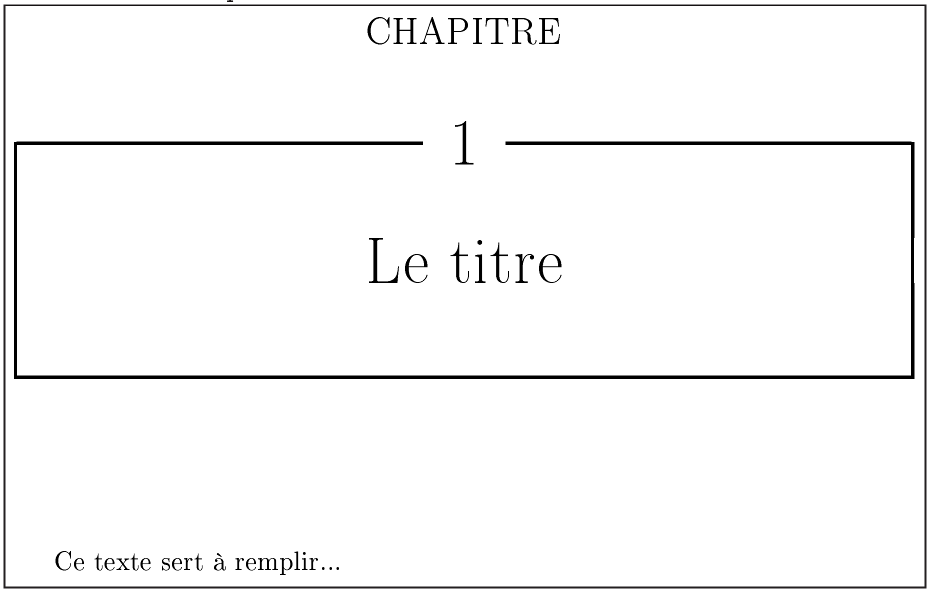


FIG. 6.10: Exemple du style Rejne pour `fncychap`

```
\usepackage[Bjarne]{fncychap}
\chapter{Le titre}
Ce texte sert \`a remplir...
```

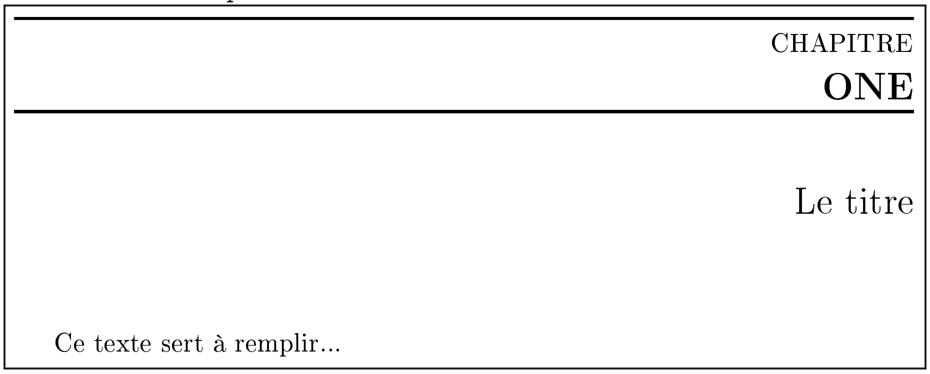


FIG. 6.11: Exemple du style Bjarne pour `fncychap`

```
\usepackage[Bjarne]{fix-fncychap}
\chapter{Le titre}
Ce texte sert \`a remplir...
```

CHAPITRE

UN

Le titre

Ce texte sert à remplir...

FIG. 6.12: Exemple du style Bjarne pour `fix-fncychap`

---

deuxième série de commande permet de faire les même types de modifications sur le *titre* du chapitre. `\ChTitleUpperCase` indique de le mettre en capitales ; `\ChTitleLowerCase` de le mettre en minuscules ; et `\ChTitleAsIs` de ne pas y toucher.

La troisième série de commandes définit les fontes (et en règle générale les choix typographiques) pour chacun des trois éléments. Ces commandes prennent en argument des commandes de changement de taille, de fonte, ou d'alignement, par exemple. `\ChNameVar` définit comment on écrit le nom du chapitre ; `\ChNumVar` définit comment on écrit le numéro du chapitre ; et `\ChTitleVar` comment on écrit le titre du chapitre.

Le package prévoit également quelques commandes qui permettraient à quelqu'un sachant coder des macros en LaTeX de re-définir lui-même ses en-têtes de chapitre. Pour cet aspect-là de l'utilisation du package, il vaut mieux que tu te reportes à la documentation d'origine (en anglais malheureusement), mais, de toutes façons, ce n'est pas très intéressant.

Le dernier point concernant ce package concerne le style Bjarne. Comme tu as pu le voir sur l'exemple de la figure 6.11 page ci-contre, il essaye d'écrire le numéro du chapitre en toutes lettres, mais il l'écrit en anglais. Le package `fix-fncychap` prend les mêmes options que `fncychap`, et fait exactement la même chose. Sauf pour le style Bjarne. Pour ce style-là, la numérotation se fera en fonction de la langue utilisée par `babel`. Comme je ne parle pas tellement d'autre langue que le français et l'anglais, je n'ai pas prévu dans ce package d'autre langue que ces deux-là, mais il peut facilement être étendu. Ce package est disponible en :
`http://jmpl.fr.eu.org/JMPL/fix-fncychap.sty`
La figure 6.12 présente un exemple d'utilisation de ce package.

6.5 Citations en tête de chapitre : le package `fancychapters`

Ce tout petit package, nous le devons à Jörg HEIKÖTTER [84].

Ce package ne fait pas grand chose. Il crée une commande `\Chapter` qui fonctionne grosso-modo comme `\chapter` sauf qu'elle permet de faire apparaître une citation *avant* le titre du chapitre. Bin oui, la faire apparaître *après*, c'est fastoche, il suffit de dire :

```
\chapter{Le chapitre}
La citation.
```

```
* \usepackage{epigraph}
* \usepackage{fancychapters}
\Chapter{%
  \epigraph{Il fait bôôôôôôôôôôôôôôô}
           {\textsc{P. Vincent}}}
  {Le chapitre du monsieur}
Texte pour remplir...
```

Il fait bôôôôôôôôôôôôôôô

P. VINCENT

Chapitre 1

Le chapitre du monsieur

Texte pour remplir...

FIG. 6.13: Exemple d'utilisation de `fancychapters`

en ajoutant éventuellement quelques commandes de formatage de la citation (changement de fonte, mise au fer à droite, etc).

`\Chapter{⟨citation⟩}[⟨titre court⟩]{⟨titre long⟩}`

`\Chapter{⟨citation⟩}*{⟨titre long⟩}`

La syntaxe de la commande `\Chapter` est simple : `[⟨titre court⟩]` et `{⟨titre long⟩}` sont les deux arguments classiques de `\chapter`. Si tu avais envie d'utiliser `\chapter*`, alors il faut légèrement déplacer l'étoile et la mettre à la place du titre court. Bien entendu, je te rappelle que `\chapter*` ne peut pas avoir de titre court, puisqu'un tel chapitre n'est pas recensé en table des matières.

La figure 6.13 page ci-contre te présente un exemple d'utilisation des deux package `epigraph` et `fancychapters` ensemble.

6.6 Citations en tête de chapitre : le package `quotchap`

Ce tout petit package, que nous devons à Karsten Tinnefeld [205], change l'apparence des têtes de chapitre et permet de placer des citations au-dessus du titre.

Il propose un environnement (`savequote`), qui prend un argument optionnel permettant de choisir la largeur de la citation (par défaut dix centimètres). Dans cet environnement, la commande `\qauthor` sert à indiquer l'origine de la citation.

Pour ce qui est de re-définir la forme des têtes de chapitre, le package est assez peu souple. Il permet juste de choisir la fonte du numéro de chapitre et s'il doit être indiqué en gris ou pas. Par défaut, le niveau de gris est de 60% (option grey prise par défaut). Tu peux indiquer l'option nogrey pour demander que ce soit en noir. Les autres options, charter, palatino, avantgarde, bookman (par défaut), courier, helvetica, newcentury, times et utopia permettent de changer la fonte utilisée pour le numéro du chapitre.

Quand on prend l'option grey (prise par défaut), le package `color` est utilisé pour définir la couleur chaptergrey qui sera utilisée par la suite. Cette couleur est un gris à 60% en RGB. Tu peux la redéfinir comme il te plaira. Voir à ce sujet le paragraphe ?? page ?? qui parle du package `color`.

Le contenu de l'environnement `savequote` est consommé par la commande `\chapter` qui le suit. Si tu veux utiliser deux fois la même citation, il te faut donc la recopier.

La figure 6.14 page suivante te montre un exemple d'utilisation de ce package.

6.7 Numéroter en chiffres romains : le package `romannum`

Ce package, que nous devons à Peter Wilson [223], sert à remplacer la numérotation classique (en chiffres arabes) de LaTeX par une numérotation en chiffres romains, majuscules ou minuscules.

Le principe d'utilisation du package est simple : tu lui donnes une ou plusieurs des options du tableau 6.1 page 335 pour lui indiquer quels numéros doivent être passés en chiffres romains. Quelle que soit l'option que tu choisis, les numéros de page seront en chiffres romains.

Ce package se base intensivement sur les commandes `\roman` et `\Roman` de LaTeX, qui sont des commandes qui ne peuvent mettre en forme que des compteurs LaTeX. Il propose également les deux commandes `\romannum` et `\Romannum` pour mettre en forme n'importe quoi qui se ramène à un entier sans forcément être un compteur (par exemple dans la date).

```
\usepackage{quotchap}
\begin{savequote}[5cm]
{\centering\textsc{Cyrano}\par}
Duel qu'en l'h\^otel Bouguiguon \\
Monsieur de Bergerac eut avec un belitre.\par
{\centering\textsc{Valvert}\par}
Qu'est-ce encore que \c{c}a monsieur?\par
{\centering\textsc{Cyrano}\par}
C'est le titre.\par
\qauthor{Edmond Rostand, Cyrano de Bergerac}
\end{savequote}
\chapter{C'est le titre}
Texte pour remplir
```

CYRANO
Duel qu'en l'hôtel Bouguiguon
Monsieur de Bergerac eut avec un belitre.
VALVERT
Qu'est-ce encore que ça monsieur ?
CYRANO
C'est le titre.

Edmond Rostand, Cyrano de Bergerac

1

C'est le titre

Texte pour remplir

FIG. 6.14: Exemple d'utilisation de `quotchap`

Option	**Description**
Section, section	Numéros de chapitres, sections, etc, en chiffres romains. Majuscules avec Section, minuscules avec section.
Equation, equation	Numéros des équations en chiffres romains majuscules ou minuscules.
Caption, caption	Numéros des flottants.
Footnote, footnote	Numéros des notes de pied de page.
Enumerate, enumerate	Numéros des listes `enumerate` de premier niveau en chiffres romains majuscules, et de troisième niveau en chiffres romains minuscules. enumerate inverse majuscules et minuscules.
Year	Écrit l'année en chiffre romains majuscules dans la date produite par `\today`.
Day, day	L'année et le jour tels que produits par `\today` sont en chiffres romains majuscules. Le jour est en minuscules pour day.
Most	Équivalent de Section, Equation, Caption, Footnote et Enumerate.
most	Équivalent de section, equation, caption, footnote et enumerate.

À noter que les options Year, Day et day, c'est-à-dire celles qui portent sur les chiffres romains dans la date, sont implémentées en réécrivant la commande `\today`. Dans un document en anglais, le résultat devrait être satisfaisant, mais dans un document en français, selon le mode de francisation (et la version des divers packages), au mieux, ça ne fera rien (la date sera toujours en français, mais sans chiffres romains), au pire, la date sera bien en chiffre romains, mais en anglais.

Ces trois options sont donc proscrites pour les utilisateurs francophones.

Tab. 6.1: Options du package `romannum`

```
* \documentclass
*     [twoside]{expl}
* \usepackage[Section]{romannum}
* \usepackage{hotplug}
* \HotReplace\datefrench
*       \number\romannum
* \HotReplace\datefrench
*       \number\Romannum
* % \texte comme d'habitude
  Date: \today.

  \section{Section}
  \texte \texte \par
  \subsection{Essai}
  \texte \texte \par
  \subsection{Encore}
  \texte \texte \par
```

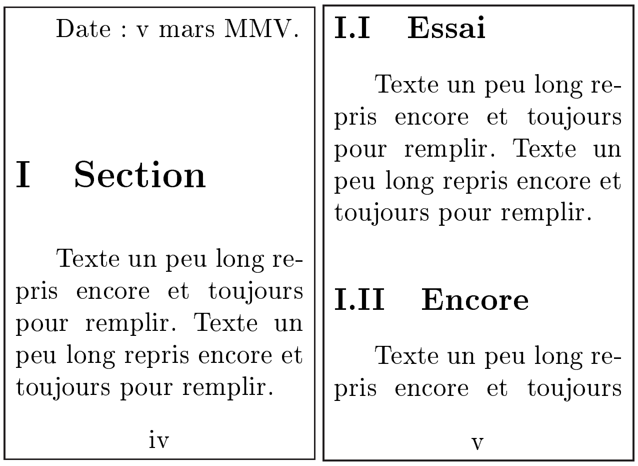


Fig. 6.15: Exemple d'utilisation de `romannum`

```
* \documentclass{expl}
* \usepackage{sectsty,ulem}
* \sectionfont{%
*    \mdseries\scshape
*    \centering\nohang}
* \subsectionfont{%
*    \centering\nohang
*    \underline}
* \subsubsectionfont{%
*    \mdseries\scshape
*    \raggedright}
* % \texte comme d'habitude
  \section{Section}
  \texte \texte \par
  \subsection{Essai un petit
    peu long}
  \texte \par
  \subsection{Encore}
  \texte \par
  \subsubsection{Titre avec
    des mots courts qui est
    long}
```

1 SECTION

Texte un peu long repris encore et toujours pour remplir. Texte un peu long repris encore et toujours pour remplir.

1.1 Essai un petit peu long

Texte un peu long repris encore et toujours pour remplir.

4

1.2 Encore

Texte un peu long repris encore et toujours pour remplir.

1.2.1 TITRE AVEC DES MOTS COURTS QUI EST LONG

Texte un peu long repris encore et toujours pour remplir. Texte un peu long repris encore et toujours pour remplir.

5

FIG. 6.16: Exemple d'utilisation de `sectsty`

L'exemple de la figure 6.15 page précédente montre une utilisation du package. Il montre aussi comment on peut obtenir la date en français en n'utilisant pas les options du packages (les options en question ne fonctionnent pas pour le français).

Note que les chiffres romains prennent en général plus de place que les chiffres arabes (compare XXXVIII et 38 pour t'en convaincre), ce qui fait que la table des matières risque de devenir ignoblement laide. L'auteur du package suggère d'utiliser le package `tocloft` (qui est également de lui, d'ailleurs) pour corriger ça.

6.8 Mise en forme des titres : le package `sectsty`

Ce package, que nous devons à Rowland MCDONNELL [133], permet, de manière assez simple, de modifier la mise en forme des titres de section.

Sur le principe, c'est facile : tu indiques, soit pour chaque niveau de sectionnement, soit pour tous, quelles modifications tu veux apporter par rapport à la fonte par défaut de la classe de documents que tu utilises. Ce que tu indiques, ce sont en fait des commandes de changement de fonte. Ces changements sont appliqués par rapport à la fonte par défaut. Ainsi, si tu veux des titres centrés en petite capitales, sachant que le règlage par défaut est en gras dans la fonte par défaut de ton document (a priori Computer Modern), tu indiqueras `\mdseries` pour annuler le gras, `\scshape` pour passer en petites capitales, et `\centering` pour dire de centrer le titre.

`\allsectionsfont{`⟨*fonte*⟩`}`

La commande la plus rapidement utile est la commande `\allsectionsfont` qui permet de changer les titres de tous les niveaux de sectionnement d'un seul coup, ainsi, pour avoir tous les titres en petites capitales non grasses, il suffit de faire appel à :

```
\allsectionsfont{\mdseries\scshape}
```

Pour à peu près tous les niveaux de sectionnement (il y a des cas particulier pour les chapitres et les parties) tu peux également changer l'alignement. Dans le cas général, en temps normal, les titres de section sont justifiés (ce qui est un choix curieux en fait) ; en ajoutant une commande de changement d'alignement à la commande de changement de fonte (`\raggedright` pour mettre le texte au fer à gauche, `\raggedleft` pour mettre au fer à droite, ou `\centering` pour centrer). En temps normal, non seulement le titre est justifié, mais également la deuxième ligne (et les suivantes) ont une indentation qui fait démarer le texte à côté du numéro de section et non pas en-dessous. Pour un alignement au fer à gauche, c'est parfait. Mais pour un titre centré, ca fait que la deuxième ligne (et les suivantes) ne sera pas centrée ni par rapport à la page, ni par rapport à la première, ce qui est très laid. Pour dire au package de supprimer cette indentation quand on n'en veut pas, il suffit d'ajouter `\nohang` aux directives d'alignement.

`\sectionfont{`⟨*fonte*⟩`}`

Plutot que de modifier tous les niveaux de sectionnement, tu peux décider de n'en modifier que certains, en utilisant pour chacun la commande qui lui est dédiée. Ainsi `\chapterfont` changera la fonte des titres de chapitre, `\sectionfont` les titres de section, `\subsectionfont` les titres de sous-section, et ainsi de suite pour `\subsubsectionfont`, pour `\paragraphfont`, pour `\subparagraphfont` et pour `\partfont`.

La figure 6.16 page précédente présente un exemple d'utilisation de ces commandes.

À noter cependant quelques cas particuliers : les parties ont un sens et une mise en forme très différents dans les différentes classes (compare l'effet de `\part` entre **article** et **report** pour t'en convaincre). Pour ces deux niveaux de sectionnement, il n'est pas très intéressant de jouer avec les commandes du package pour changer la mise en forme.

De même, par défaut, les commandes `\paragraph` et `\subparagraph` créent des titres qui sont en fait le début de ton paragraphe de texte. Si tu joues avec l'alignement (par exemple pour centrer) avec ce package, le résultat sera, au mieux, curieux.

Pour certains titres le numéro et le titre sont clairement des entités isolées : c'est le cas pour les chapitres et pour les parties, dans les document ayant des chapitres (classes **report** et **book**). Pour ces deux cas-là, on peut paramètrer chacun indémendament de l'autre. Ça se fait avec les commandes `\partnumberfont` et `\parttitlefont` pour les titres de parties (la commande `\partfont` étant un raccourcis pour passer la même fonte aux deux) ; et avec les commandes `\chapternumberfont` et `\chaptertitlefont` pour les chapitres (la même remarque s'applique pour `\chapterfont`).

6.8.1 Le soulignement

D'entrée de jeu, il faut rappeler : souligner, ça ne se fait pas. C'est une mauvaise transposition pour les machines à écrire et pour l'écriture manuscrite de ce qui devrait être du gras. Typographiquement parlant, souligner, c'est aberrant.

Toutefois, beaucoup de gens veulent faire des choses aberrantes. Trop de gens. Mais si vraiment on te force, et que tu n'as pas le choix, on *peut* demander à `sectsty` de souligner les titres, même si on *ne doit pas*.

Pour comprendre comment ça marche, c'est assez simple. Supposons que le style par défaut du titre de section qui t'intéresse soit `\normal`[160], et que le style que tu ajoutes à

[160]Le style par défaut effectif dépend du niveau (pas la même taille pour tous les niveaux de sectionnement,

celui-là soit `\PourMoi`, par exemple en faisant appel à :

```
\sectionfont{\PourMoi}
```

En fait, les commandes invoquées par

```
\section{Le titre}
```

vont ressembler à ça :

```
{\normal\PourMoi{Le titre}}
```

Ce qui est intéressant à noter, ce sont les accolades : la première et la dernière font que si `\normal` et `\PourMoi` sont des commandes qui changent de fonte sans prendre de paramètre (par exemple `\bfseries` et `\itshape`), le changement de fonte induit par ces deux commandes s'arrêtera à la dernière accolade. La deuxième paire d'accolades, celle autour du titre, fait que si ta commande de formatage à toi prend un paramètre, ce paramètre, c'est le titre. Ainsi, tu peux avoir cet effet là :

```
\sectionfont{\itshape\underline}
\section{Le titre}
donne:
{\normal\itshape\underline{Le titre}}
```

Du coup, tu peux utiliser n'importe quelle commande qui prenne un argument du moment qu'elle est utilisée en dernier. Et en particulier, tu peux utiliser ça pour souligner ton titre en utilisant la commande native de LaTeX, `\underline`.

Comme cette demande, bien qu'aberrante (l'ais-je assez dit ?), est relativement fréquente, elle subit un traitement de faveur de la part de `sectsty`. En effet, la commande `\underline` de LaTeX souffre d'un grave défaut : son contenu est définitivement monobloc et ne peut pas changer de ligne (en plus du fait que le trait de soulignement soit franchement mal placé). Un package existe pour corriger ce grave défaut[161] : `ulem` (voir à son sujet le paragraphe 2.44 page 148).

Si le package `sectsty` détecte que le package `ulem` est chargé (cette détection à lieu durant le `\begin{document}`, donc l'ordre de chargement n'est pas important dans ce cas là), la commande `\underline` est redéfinie, *uniquement pendant que LaTeX est en train de mettre en forme un titre de section*, pour appeler les commandes du package `ulem`. Ainsi, si tu utilises `\underline` pour souligner un titre (ce qui est *mal*), et que tu as chargé le package `ulem`, le résultat sera moins pire.

$\boxed{\texttt{\textbackslash ulemheading}\{\langle \textit{cmde}\rangle\}\{\langle \textit{texte}\rangle\}}$

Le package `ulem` propose plusieurs commandes pour traiter le soulignement. Celle prise par défaut est `\uline` qui fait un souligné simple. Si tu veux en utiliser une autre, par exemple `\uuline` pour un souligné double, tu peux le faire en utilisant la commande `\ulemheading`, par exemple comme ceci :

```
\sectionfont{\ulemheading{\uuline}}
```

par exemple) et des packages chargés : les packages de changement global de fonte changent aussi les titres. Du coup, je me ramène à une hypothèse commune. Si tu veux, considères que le style par défaut est `\bfseries` et que celui que tu ajoutes est `\itshape`.

[161] Ce n'est plus désormais le seul package qui traite ce problème, mais, celui-là, il est reconnu par `sectsty`.

```
\usepackage{sectsty,ulem}
\newcommand\boite[1]{%
  \begin{center}
  \fbox{\begin{minipage}
              {.9\linewidth}
  \centering #1
  \end{minipage}}
  \end{center}
}
\sectionfont{%
   \nohang\boite}
\subsectionfont{%
   \sectionrule{3ex}{3pt}
               {-1ex}{1pt}}
% \texte comme d'habitude
\section{Section}
\texte \texte \par
\subsection{Essai un petit
  peu long}
\texte \par
\subsection{Encore}
\texte \par
\subsubsection{Titre avec
  des mots courts qui est
  long}
```

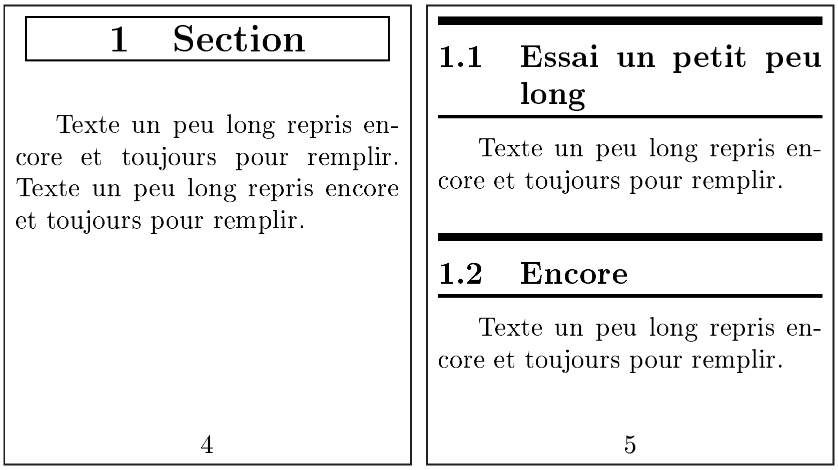


FIG. 6.17: Exemple d'utilisation avancée de `sectsty`

6.8.2 Traits et cadres

Dans la série des choses qu'il faut probablement éviter, sauf si c'est bien fait, il y a également le fait d'encadrer un titre, ou de le faire précéder et/ou suivre d'un trait horizontal sur la largeur de la page. Pour les cadres, c'est assez simple, il suffit de combiner ce qui est expliqué au chapitre neuvième page 571 avec ce que nous avons vu au titre 5 page 299 pour créer une commande qui produise le type d'encadrement qu'on veut.

`\sectionrule{`⟨*décal. haut*⟩`}{`⟨*épaiss. haut*⟩`}{`⟨*décal. bas*⟩`}{`⟨*épaiss. bas*⟩`}{`⟨*titre*⟩`}`

Pour les filets horizontaux avant et après le titre, une commande spéciale, `\sectionrule`, est prévue par le package. Elle prend cinq paramètres. Le premier indique de combien il faut rehausser le filet du haut, sa position par défaut étant au pied de la première ligne du titre ; le deuxième indique l'épaisseur de ce filet ; le troisième indique de combien il faut rehausser le filet du bas, sa position par défaut étant au pied de la dernière ligne (il veut mieux le descendre, donc donner une valeur négative) ; et le quatrième paramètre indiquant l'épaisseur du filet du bas. Enfin, le dernier argument est le titre de la section, tu peux donc utiliser cette commande un peu comme une commande de soulignement.

Ces deux possibilités sont illustrées par l'exemple de la figure 6.17 page précédente.

6.8.3 Autres infos et usages

Un usage amusant de `\chapterfont` peut être de s'en servir pour appeler `\thispagestyle`. C'est un moyen simple de faire en sorte que les pages de tête de chapitre ne soient plus en plain, mais dans le style que tu voudras, et que tu auras pu définir, par exemple avec le package `fancyhdr` (voir à ce sujet le paragraphe 4.3 page 200).

Le package `sectsty` redéfinit toutes les commandes de gestion des sections, et va donc poser des problèmes de compatibilité avec les packages qui en font de même. Le premier cas d'incompatibilité est avec `fncychap`, qui redéfinit la commande `\chapter`. Si tu veux utiliser une des configuration de `fncychap`, et disposer des possibilités de `sectsty` pour les autres section, il faut charger les deux packages dans cet ordre :

```
\usepackage{sectsty,fncychap}
```

En effet, dans ce cas là, la définition de `\chapter` par `sectsty` sera écrasée par la définition fournie par `fncychap`. Cependant, tu ne pourras pas modifier la fonte du titre de chapitre, ce qui n'est pas très grave : le package `fncychap` offre déjà, pour ça, des possibilités intéressantes.

Un autre package qui cuisine un peu autour des commandes de sectionnement est le package `minitoc`. Cependant, ce dernier est un peu moins brutal : il *modifie* les commandes, alors que `sectsty` les *écrase*. Du coup, il suffit de charger `minitoc` en deuxième pour que tout se passe bien.

6.9 Titres sans sérifications : le package `sfheaders`

Ce microscopique package, que nous devons à Maurizio Loreti, change la fonte utilisée pour tous les titres de section pour une variante sans sérifications (celle de `\textsf`).

Il suffit de l'utiliser pour qu'il prenne effet immédiatement. Il n'est compatible qu'avec les classes standard de LaTeX (**article**, **report** et **book**), avec d'autres classes il peut créer des problèmes. Il fonctionne en redéfinissant les commandes de sectionnement. Fonctionnellement, le package `sectsty` permet plus de choses, et l'utilisation de `sfheaders` est équivalente à :

```
\usepackage{sectsty}
\allsectionsfont{\sffamily}
```

6.10 Mise en forme des titres : le package varsects

Ce petit package, que nous devons à Daniel TAUPIN, offre des possibilités similaires à celles de sectsty, mais sur une toute autre approche. L'idée de base du package est relativement simple : il redéfinit les commandes de sectionnement des classes standard de LaTeX, mais, au lieu d'y coder en dur le règlage de fonte et les tailles de blanc à utiliser, il externalise ces informations dans des macros que l'on peut alors redéfinir pour changer l'apparence des sections.

Cette approche, plus basique que celle du package sectsty est un peu plus délicate à utiliser pour un débutant, mais elle offre l'avantage de pouvoir modifier la taille des blanc autour des titres sans devoir tout redéfinir, ce qui est très pratique quand on veut se faire rapidement une classe de document personnalisée.

Voilà quelques exemples d'utlisation du package :

```
\usepackage{varsects}
% Titre de chapitre en \textsf:
\renewcommand\ChapterFont{\sffamily\bfseries}
% Un grand blanc en tete de chapitre pour une citation:
\renewcommand\ChapterTopSpace{4cm}
% Un blanc plus fort avant les titres de section:
\renewcommand\SectionSpaceBefore{5.5}
```

Toutes les commandes pouvant être redéfinies ont recensées dans le tableau 6.2 page suivante.

Si tu te contentes de charger le package, sans redéfinir aucune des commandes qu'il propose, tu retrouveras la mise en forme des classes de document de LaTeX telle qu'elle était définie en 1997 (c'est de là que viennent les valeurs par défaut). Autant que je sache cette mise en forme n'a pas changée depuis. Donc, normalement, si tu te contentes de charger le package, ça ne change rien à ton document.

6.11 Mise en forme des titres : le package titlesec

Ce package, que nous devons à Javier BEZOS [17], offre une interface puissante pour redéfinir assez facilement la mise en forme des titres de ton document. Il modifie très profondément la mise en forme des titres telle qu'elle est définie par LaTeX par défaut.

Le package titlesec a deux niveaux d'utilisation. Le premier niveau, facile, permet de faire quelques modifications rapides, en passant des options au package. On ne peut pas changer en profondeur la mise en forme des titres, mais on peut y faire des ajustements intéressants. Le deuxième niveau est très puissant, mais franchement hostile. Il permet de redéfinir complètement ce qu'on veut comme rendu pour les titres, mais il faut vraiment du temps pour comprendre comment ça marche, et c'est presque un investissement à refaire à chaque fois : ce n'est pas tous les jours qu'on crée de toutes pièces une mise en page, du coup il va se passer du temps avant que tu ais de nouveau besoin du package, et comme il est complexe tu auras sans doute tout oublié.

6.11.1 Utilisation simple

L'utilisation simple se borne, essentiellement, à utiliser les options du tableau 6.3 page 343 pour changer la fonte ou l'alignement des titres de section. Ces simples règlages permettent cependant d'arriver à des résultats intéressants et différents. Par exemple, un appel comme :

```
\usepackage[sc,center]{titlesec}
```

Macro	**Signification**
\ChapterSize	Taille du titre du chapitre.
\ChapterNameSize	Taille du mot chapitre et du numéro.
\ChapterFont	Fonte du titre *et* du numéro.
\ChapterTopSpace	Taille du blanc vertical avant le mot chapitre. Valeur par défaut `50pt`.
\ChapterModSpace	Taille du blanc vertical entre le mot chapitre et le titre. Valeur par défaut `20pt`.
\ChapterBotSpace	Taille du blanc vertical après le titre du chapitre et avant le texte. Valeur par défaut `40pt`.
\SectionFont	Fonte du titre de `\section`.
\SectionSize	Taille du titre de `\section`.
\SectionSpaceBefore	Blanc vertical avant le titre (voir macro `\@startsection` page 319), une valeur, exprimée dans l'unité `ex`, qui correspond à la partie rigide du blanc. Valeur par défaut : `3.5`.
\SectionSpaceAfter	Idem, mais après le titre. Valeur par défaut `2.3`.
\SubSectionFont	Fonte de `\subsection`.
\SubSectionSize	Taille de `\subsection`.
\SubSectionSpaceBefore	Blanc vertical avant. Valeur par défaut `3.25`.
\SubSectionSpaceAfter	Blanc vertical après. Valeur par défaut `1.5`.
\SubSubSectionFont	Fonte de `\subsubsection`.
\SubSubSectionSize	Taille de `\subsubsection`.
\SubSubSectionSpaceBefore	Blanc vertical avant. Valeur par défaut `3.25`.
\SubSubSectionSpaceAfter	Blanc vertical après. Valeur par défaut `1.5`.
\ParagraphFont	Fonte de `\paragraph`.
\ParagraphSize	Taille de `\paragraph`.
\ParagraphSpaceBefore	Blanc vertical avant. Valeur par défaut `3.25`. Le blanc après est horizontal et ne peut pas être paramétré.
\SubParagraphFont	Fonte de `\subparagraph`.
\SubParagraphSize	Taille de `\subparagraph`.
\SubParagraphSpaceBefore	Blanc vertical avant. Valeur par défaut `3.25`. Le blanc après est horizontal et ne peut pas être paramétré.

TAB. 6.2: Macros de paramètrage de `varsects`

Options	Rôle
rm, sf, tt	Options permettant de fixer la famille de fonte utilisée pour les titres.
md, bf	Idem pour la série.
up, it, sl, sc	Idem pour la forme.
big, medium, small, tiny	Options fixant la taille utilisée pour les titres. L'option big correspond au règlage par défaut de LaTeX ; l'option tiny indique que tous les titres sont de la même taille que le texte ; les deux autres sont des intermédiaires.
raggedleft, raggedright, center	Option indiquant l'alignement des titres.
compact	Réduit l'espacement autour des titres.

TAB. 6.3: Options simples du package `titlesec`

---

produit un résultat qui rappelle les classes de document de l'$\mathcal{AMS}$. On peut produire un résultat comparable à celui du package `sfheaders` en appelant :

```
\usepackage[sf]{titlesec}
```

`\titlelabel{`⟨*format numéro*⟩`}`

L'utilisation simple prévoit également deux commandes. La première est `\titlelabel` qui permet de changer la mise en forme des numéros des sections. Elle prend comme argument la mise en forme du numéro. Dans son argument on place `\thetitle` à la place du numéro de section. Cette commande sera remplacée par `\thesection` pour les sections, `\thesubsection` pour les sous-section, etc.

Le règle par défaut est :

```
\titlelabel{\thetitle\quad}
```

Tu peux par exemple ajouter un point après le numéro, ou l'écrire en rouge encadré en faisant comme ça :

```
% Avec un point:
\titlelabel{\thetitle.\quad}
% Encadr\'e en rouge:
\titlelabel{\fbox{\color{red}\thetitle}\quad}
```

`\titleformat*{`⟨*commande*⟩`}{`⟨*fonte*⟩`}`

La deuxième commande prévue par l'utilisation simple est `\titleformat*`. Elle prend deux paramètres. Le premier est la commande de sectionnement que tu veux modifier, et le second est la fonte que tu veux utiliser pour ce type de titres. La fonte que tu précises ici, pour *ce type de titre* vient à la place de ce que tu as pu préciser dans les options de chargement du package.

L'exemple de la figure 6.18 page suivante montre l'utilisation de certaines de ces options pour obtenir des titres sans sérifications et gras, ainsi que l'utilisation de la commande `\titlelabel` pour avoir des numéros encadrés et rouge (c'est juste pour l'exemple, hein, esthétiquement le résultat est, heu, discutable), et de la commande `\titleformat*` pour que les sous-sections soient dans une autre fonte (là encore, ce n'est pas un choix forcément judicieux).

```
* \usepackage{color}
* \usepackage[sf,bf]{titlesec}
* \titlelabel{\fbox{%
*     \color{red}%
*     \thetitle}\quad}
* \titleformat*{\subsection}
*     {\sffamily}
* % \texte comme d'habitude
  \section{Section}
  \texte \texte \par
  \subsection{Essai un petit
    peu long}
  \texte \par
  \subsection{Encore}
  \texte \par
  \subsubsection{Titre avec
    des mots courts qui est
    long}
```

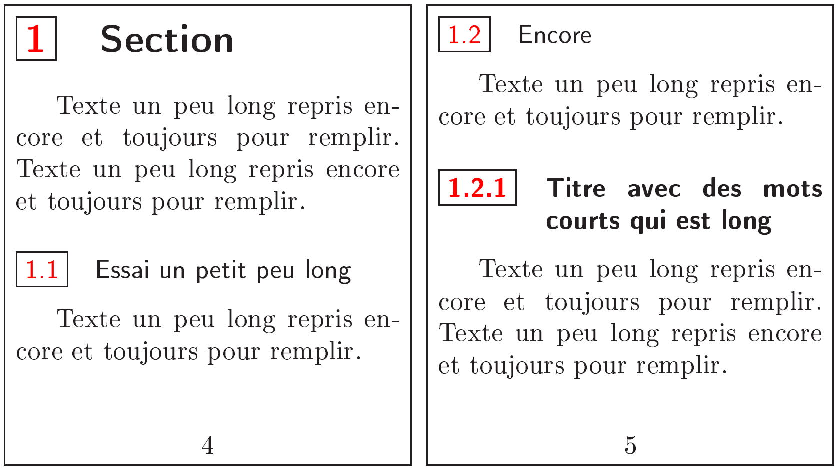


FIG. 6.18: Exemple d'utilisation simple du package `titlesec`

6.11.2 Principe de l'utilisation complexe

Au premier abord, le principe est relativement simple. Pour définir la mise en forme d'un titre, il faut en premier lieu choisir la classe d'un titre (pleine page, en haut de page, dans le texte), puis un format de base parmis une liste de formats connus (indenté, encadré, etc). À partir de ce format basique, tu pourras préciser des paramètres (valeur de telle ou telle dimension, position de tel ou tel élément, etc), et tu pourras ajouter des commandes LaTeX à certains moments clefs (avant le titre, après le numéro, après le titre, etc).

L'ensemble forme une infinité de possibilités de paramètrages, à tel point qu'il est délicat de s'y retrouver.

Et ce d'autant plus que le package ne se comporte pas toujours de la même façon vis-à-vis des différents types de titres[162]. En effet, pour la majorité des titres, il suppose qu'ils sont définis peu ou prou comme dans les classes de base de LaTeX, et il va éplucher les définitions pour aller chercher les paramètres clefs (les différents paramètres de `\@startsection`) et en faire son règlage par défaut. Mais pour `\chapter` et pour `\part`, qui sont faits de manière très différente, il serait trop compliqué d'aller à la pêche aux infos. Du coup, spontanément, il va avoir tendance à les ignorer.

6.11.3 Les classes

Le package `titlesec` prévoit trois classes de titres, celles du tableau 6.4 page ci-contre.

La notion de classe est, pour `titlesec`, intimement liée à la notion de hiérarchie des titres. La hiérarchie des titres indique que les chapitres sont dans les parties, que les sections sont dans les chapitres, etc. Il modélise ça, comme LaTeX, avec des valeurs entières.

[162]Il a cependant de bonnes excuses pour faire comme ça, mais la conséquence est que c'est parfois compliqué de comprendre ce qui se passe.

Classe	**Description**
page	Les titres de cette classe-là prennent toute une page, c'est le cas par exemple des `\part` dans les classes **report** et **book**.
top	Les titres de cette classe-là sont toujours en haut de page, et sont donc toujours précédés d'un changement de page (et d'une purge des flottants en cours, d'ailleurs). C'est le cas par exemple de `\chapter`.
straight	Tous les autres titres sont de la classe straight, les titres de cette classe-là sont dans le texte, en ce sens qu'il n'y a pas de changement de page imposé autour.

TAB. 6.4: Classes de titre pour `titlesec`

`\titleclass{`⟨*commande*⟩`}[`⟨*origine*⟩`]{`⟨*classe*⟩`}`

La commande `\titleclass` a deux syntaxes. La première permet d'indiquer quel est le niveau de sectionnement le plus élevé de ton document. Le premier argument est la commande de sectionnement correspondante, le second est son numéro hiérarchique, et le troisième est sa classe. Le principe veut que le numéro hiérarchique que tu passeras ici servira de base au décompte de ceux des autres niveaux, sa valeur a donc très peu d'importance.

Une utilisation typique :

```
\titleclass{\part}[-1]{page}
```

Cette forme-là de la commande est par définition très rare : si tu utilises une classe de documents classique, tu ne t'en serviras pas (le package détectera tout seul les niveaux existants, et donc le niveau le plus haut), et si tu ne te bases pas sur l'auto-détection, tu t'en serviras au plus une fois (logique, il ne peut pas y avoir deux types de niveau le plus élevé). A priori, cet usage aura lieu dans la définition d'une classe que tu serais en train de te fabriquer.

`\titleclass{`⟨*commande*⟩`}{`⟨*classe*⟩`}[`⟨*niveau précédent*⟩`]`

Cette seconde forme de la commande, où le niveau hiérarchique, optionnel, est placé en dernier, est d'un usage plus classique. Tu es succeptible de t'en servir pour chaque niveau de sectionnement de ton document. Soit pour en changer la classe, soit pour changer son niveau hiérarchique. Ici, le troisième paramètre n'est optionnel que si tu *changes* la classe d'un titre, pas si tu *définis* la classe d'un nouveau type de titre. De plus, le niveau que tu donnes est la commande de niveau suppérieur, par exemple, si tu veux créer un `\subchapter` en classe `top`, tu peux dire :

```
\titleclass{\subchapter}{top}[\chapter]
```

Autant, changer la classe, on peut voir rapidement à quoi ça sert : par exemple, dans ce livre, les `\section` sont de classe `top` (du moins quand il n'est pas mis en forme en deux colonnes) ; autant le changement du niveau hiérarchique n'a d'intérêt que dans des cas plus rares, le plus probable étant le fait que tu veuilles ajouter un niveau intermédiaire dans ton document, par exemple un `\subchapter`. Si tu veux créer ton propre niveau de sectionnement, le package ne fera pas le travail à ta place, ou du moins pas tout : il ne s'occupera que de la mise en forme et de l'incrémentation des compteurs, il restera à ta charge de définir le compteur, de dire sa mise en forme (*i.e.* `\thesection`), les marques, etc.

```
* \usepackage[loadonly]{titlesec}
* \newcounter{un}
* \renewcommand\theun{\Roman{un}}
* \newcommand\unmark[1]{%
*     \markboth{\theun\quad#1}{}}
* \titleclass{\un}[1]{top}
* \titleformat{\un}[frame]{}
*     {\filright
*      \quad\textsc{sect.}
*      \theun\quad}{6pt}
*     {\Large\bfseries\centering}
* \titlespacing*{\un}
*     {-15pt}{2ex plus 1ex}
*     {1ex plus .5ex}[-15pt]
* \newcounter{deux}[un]
* \renewcommand\thedeux{%
*   \theun.\arabic{deux}}
* \titleclass{\deux}
*            {straight}[\un]
* % \deux de type runin
* % \texte comme d'habitude
  \un{Titre un}
  \texte\par
  \deux{Plouf}
  \texte \texte\par
  \deux{Coin}
  \texte \par
  \un{Titre deux}
  \texte\par
  \deux{Pouet}
```

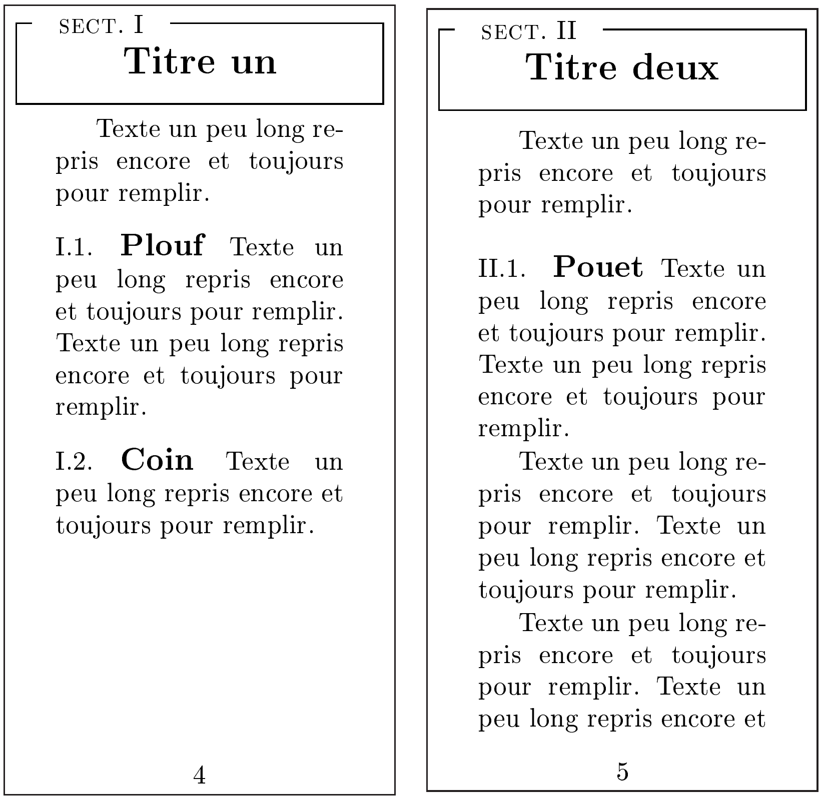


FIG. 6.19: Exemple de création d'un niveau de sectionnement

La figure 6.19 présente un exemple complet d'utilisation du package `titlesec` pour créer un nouveau niveau de sectionnement, que j'ai appelé `\un`, pour qu'il ne puisse pas être confondu avec le `\section` qui existe de base. L'option loadonly du package lui indique de *ne pas* chercher à analyser l'existant, puisqu'on ne va pas en tenir compte.

À noter une curiosité du package : comme il prévoit les classes top et page, qui supposent des changements de page, il s'arrange pour honorer les options classiques openany et openright des classes de document habituelles. Or il se trouve que certaines classes (par exemple **article**) ne définissent pas ces options. Du coup, pour éviter une erreur, le package se comporte dans ce cas là *comme si l'option openright était active*. Jusque-là, rien de choquant. Ce qui est embêtant c'est qu'il n'y a pas moyen de lui dire de faire autrement : pas moyen de lui dire que par défaut, il devrait considérer que c'est l'option openany qui est activée.

6.11.4 Mise en forme des titres

Tu as déjà entr'apperçu les deux commandes utiles pour définir la mise en forme des titres (je m'en sers dans l'exemple de la figure 6.19) : `\titleformat` et `\titlespacing`. Ces deux commandes sont supposées être indépendantes, la première règlant la mise en forme

Forme	Description
hang	C'est la mise en forme des titres classiques de LaTeX, le titre forme un paragraphe, et le numéro est en exergue par rapport au titre.
block	Dans cette mise en forme le titre forme simplement un paragraphe de texte. Très pratique pour les mises en forme véritablement complexes : tu peux utiliser tous les artifices que tu veux pour faire la mise en forme, y compris les environnements de dessin.
display	Le numéro forme un premier paragraphe, et le titre un second ; un peu à la manière des en-têtes de chapitre classiques de LaTeX (à condition de faire précéder le numéro du mot chapitre, de choisir les bonnes tailles, et de mettre les bons espacements).
runin	Le titre forme le début d'un paragraphe de texte : à moins que tu ne changes toi-même de paragraphe dans le corps de ton texte juste après le titre, le texte viendra à la suite, sur la même ligne.
leftmargin	Le titre sera écrit dans une (très) faible largeur dans la marge de gauche. C'est dangereux pour les titres longs, sauf à avoir des marges gigantesques. Le texte de la section correspondante commence au même niveau que le haut du titre.
rightmargin	Le titre sera écrit dans la marge de droite.
drop	Le texte viendra s'écrire autour d'un rectangle réservé pour le titre. Dans certains cas (par exemple si le premier paragraphe de texte est trop court), il y a un risque que du texte vienne s'écrire par-dessus le titre.
wrap	Comme drop, sauf que la largeur du rectangle réservé est ici celle de la plus longue des lignes du titre.
frame	Le titre sera placé dans un cadre, le numéro apparaissant dans le bord haut du cadre.

TAB. 6.5: Formats de titres prédéfinis pour `titlesec`

du titre, et la seconde règlant l'espacement autour. Dans la pratique, ce n'est pas très vrai : la façon dont les paramètres d'espacement sont pris en compte dépend de la mise en forme choisie.

`\titleformat`{⟨*commande*⟩}[⟨*format*⟩]{⟨*formatage*⟩}{⟨*numéro*⟩}
{⟨*séparation*⟩}{⟨*avant*⟩}[⟨*après*⟩]

Cette commande est un des piliers du package : elle re-définit la commande de sectionnement qui lui est donnée en premier paramètre, {⟨*commande*⟩}, pour en changer la mise en forme. Le deuxième paramètre, [⟨*format*⟩], permet de spécifier le format à utiliser pour ce niveau de titre, à choisir dans la liste du tableau 6.5. Les autres paramètres peuvent changer de sens selon le format retenu : chaque paramètre est supposé avoir un rôle précis, mais sera interprété par le format, chaque format n'ayant pas forcément les mêmes capacités de mise en forme que les autres.

Le premier paramètre est, en fait, plus complexe que simplement le nom de la commande. Il peut prendre la forme d'une liste de clefs/valeurs. Les clefs possibles sont name, numberless, et page. Ça permet d'appliquer des mises en forme radicalement différentes au titre selon qu'il sera sur une page recto ou verso, et selon qu'il sera numéroté ou pas. À l'utilisation, ca donnera, par exemple ça :

```
% Recto ou verso, usage classique:
```

```
\titleformat{\section}[...]...
% Recto seulement:
\titleformat{name=\section,page=odd}[...]...
% Verso seulement:
\titleformat{name=\section,page=even}[...]...
% Verso seulement, non numerote:
\titleformat{name=\section,page=even,numberless}[...]...
```

Cette fonctionnalité n'est que rarement utile.

`\titlespacing{⟨commande⟩}{⟨gauche⟩}{⟨blanc haut⟩}{⟨blanc bas⟩}[⟨droite⟩]`

Cette commande définit l'espacement autour du titre, que ce soit en horizontal pour que le titre soit renfoncé par rapport aux marges du texte, ou pour qu'au contraire il dépasse dans la marge ; ou vertical, pour le séparer plus ou moins du texte alentour. La version avec une étoile (`\titlespacing*`) indique de supprimer l'indentation du premier paragraphe suivant le titre (c'est l'usage dans la typographie anglosaxone). Plus exactement, la version étoilée respectera l'usage LaTeX, c'est-à-dire que les titres obéiront à l'usage du package `indentfirst` et aux règlages de `babel`, alors que la version sans étoile fait fi des usages LaTeX et indente le paragraphe d'après sans se poser de question. Du coup, en règle générale, il vaut mieux utiliser la version avec une étoile : au moins, on ne sera pas surpris, et ça fonctionnera comme d'habitude, les autres packages y retrouveront leurs marques.

Le premier paramètre, {⟨*commande*⟩}, indique la commande de sectionnement (par exemple `\subsection`) pour laquelle on redéfinit l'espacement. Le deuxième paramètre, {⟨*gauche*⟩}, indique le blanc sur le côté gauche (en général le renfoncement par rapport à la marge). Le troisième paramètre, {⟨*blanc haut*⟩}, indique (en général) le blanc à laisser avant le titre. Le quatrième paramètre, {⟨*blanc bas*⟩}, indique (en général) le blanc vertical à laisser après le titre. Enfin, le dernier paramètre, optionnel, [⟨*droite*⟩] indique le renfoncement à respecter par rapport à la marge de droite.

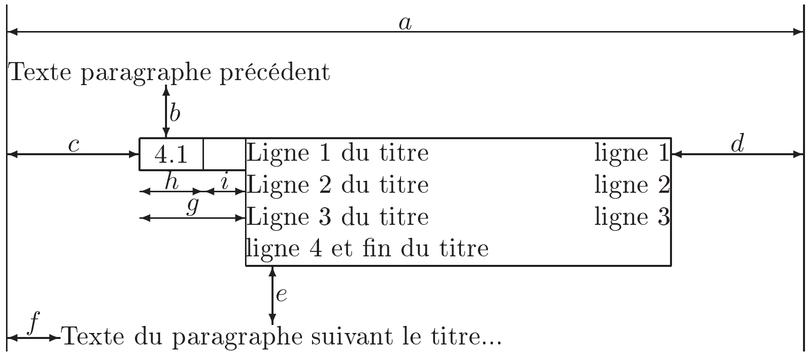
Le sens exact de chacun des paramètres dépendra du ⟨*format*⟩ qui aura été choisi par la commande `\titleformat` pour ce niveau de sectionnement là.

6.11.5 Le format hang

La figure 6.20 page suivante présente schématiquement la mise en forme d'un titre dont le format est `hang`. On repère clairement l'usage des quatre paramètres de la commande `\titlespacing` pour faire les quatre blancs autour du titre. À noter que les paramètres {⟨*gauche*⟩} et/ou [⟨*droite*⟩] sont pris en compte. Si tu ne précises pas la valeur de [⟨*droite*⟩], le titre ira jusqu'à la marge de droite.

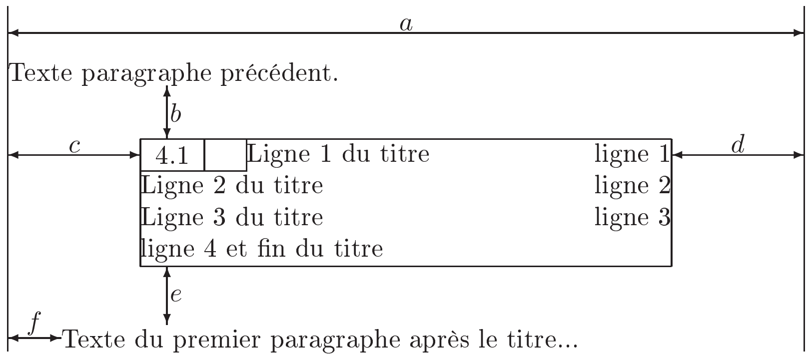
Le paramètre {⟨*formatage*⟩} de la commande `\titleformat` contient des commandes de mise en forme qui sont éxécutées juste après que le blanc vertical du haut ait été mis et juste avant que le paragraphe du titre soit mis en forme. Si tu veux que le titre ne soit pas justifié (pour peu que ton texte ne soit pas très large, et que le titre soit écrit gros, la justification ne produit pas un beau résultat) tu peux le faire à ce niveau là en indiquant `\raggedright` par exemple[163], ou en utilisant une des commandes d'espacement fournies par le package. La commande `\filright` indique de remplir de blanc à droite (et donc de composer le titre en drapeau, au fer à gauche) ; la commande `\filleft` fait le contraire, venant coller le titre à la marge droite ; et la commande `\filcenter` mettra autant de blanc à gauche qu'à droite

[163]Il vaut mieux ne pas utiliser les commandes classiques de LaTeX, et préférer celles du package, en particulier, celles de LaTeX font sauter les règlages des marges ⟨*droite*⟩ et ⟨*gauche*⟩ ; alors que celles du package les conservent.



- a largeur de la page (`\linewidth`) ;
- b blanc vertical avant le titre (paramètre {⟨*blanc haut*⟩} de `\titlespacing`) ;
- c marge supplémentaire à gauche (paramètre {⟨*gauche*⟩} de `\titlespacing`) ;
- d marge supplémentaire à droite (paramètre `[`⟨*droite*⟩`]` de `\titlespacing`) ;
- e blanc vertical après le titre (paramètre {⟨*blanc bas*⟩} de `\titlespacing`) ;
- f indentation du paragraphe suivant le titre, supprimée si tu utilises `\titlespacing*` et conservée si tu utilises `\titlespacing` ;
- g l'indentation du titre, qui fait la longueur du texte produit par le paramètre {⟨*numéro*⟩} de `\titleformat` plus la séparation indiquée par le paramètre {⟨*séparation*⟩} de `\titleformat`, dans un cas classique, c'est la taille du numéro (h sur le dessin) plus le blanc de séparation (i sur le dessin, `\quad` par défaut pour LaTeX) ;
- h largeur naturelle du numéro de la section tel que produit par {⟨*numéro*⟩} ;
- i taille du blanc de séparation entre le numéro et le titre ({⟨*séparation*⟩}), en fait, il s'ajoute à ce que tu choisis comme commandes pour écrire le numéro.

FIG. 6.20: Format hang du package `titlesec`



Dans le format block le numéro est pris dans le paragraphe du titre, il ne dépasse pas. Sinon, les réglages sont similaires à ceux de hang. Ce format est plus adapté aux titres qui seront centrés sur leur largeur, par exemple.

a largeur de la page (`\linewidth`) ;

b blanc vertical avant le titre (paramètre {⟨*blanc haut*⟩} de `\titlespacing`) ;

c marge supplémentaire à gauche (paramètre {⟨*gauche*⟩} de `\titlespacing`) ;

d marge supplémentaire à droite (paramètre [⟨*droite*⟩] de `\titlespacing`) ;

e blanc vertical après le titre (paramètre {⟨*blanc bas*⟩} de `\titlespacing`) ;

f indentation du paragraphe suivant le titre, supprimée si tu utilises `\titlespacing*` et conservée si tu utilises `\titlespacing`.

Le numéro de la section est produit par le paramètre {⟨*numéro*⟩} de `\titleformat`. Le blanc de séparation entre le numéro et le titre est définit par la longueur passée dans le paramètre {⟨*séparation*⟩} de `\titleformat`.

FIG. 6.21: Format block du package `titlesec`

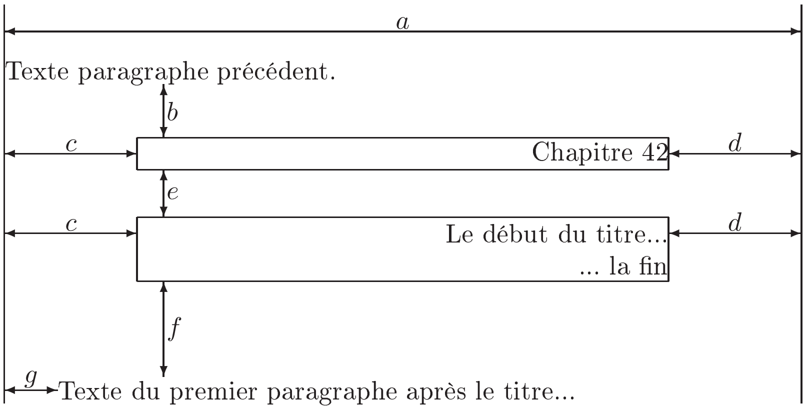
---

produisant un titre presque[164] centré. La commande `\fillast` justifie toutes les lignes, sauf la dernière, qui est centrée. Enfin les commandes `\filinner` et `\fillouter` agissent comme `\filright` et `\filleft` mais en tenant compte du fait que la page soit recto ou verso : `\filinner` ajoute du blanc dans la marge intérieure et `\filouter` ajoute du blanc dans la marge extérieure.

Le paramètre [⟨*numéro*⟩] de `\titleformat` est clairement exposé sur la figure : c'est le contenu de ce paramètre qui produira le numéro tel qu'il dépasse par rapport au reste du titre, y compris un éventuel blanc entre le numéro et le titre. Le paramètre {⟨*séparation*⟩} est une longueur, et indique la taille du blanc de séparation entre le numéro et le titre. Le paramètre {⟨*avant*⟩} contient du code de mise en forme, qui servira à mettre en forme le titre (et qui ne s'appliquera pas au numéro), si la dernière commande de ce paramètre prend un paramètre, ce paramètre est le titre lui même (pour souligner le titre, par exemple[165]). Enfin le paramètre optionnel [⟨*après*⟩] peut contenir du code de mise en forme qui sera placé après le titre : si tu veux ajouter un point final après tes titres (ce qui ne se fait en général pas, sauf eventuellement pour les titres de type runin) tu peux ajouter le point en question dans ce paramètre optionnel.

[164]Pour le format hang, ce ne sera pas bien centré, à cause de la première ligne qui est plus longue et qui dépasse sur le côté gauche (le numéro de la section ne sera pas pris en compte dans le calcul du centrage). Pour obtenir un titre centré, il vaut mieux utiliser le format block.

[165]Mais il me semble que j'ai déjà dit qu'il ne fallait jamais souligner.



- *a* largeur de la page (`\linewidth`) ;
- *b* blanc vertical avant le titre (paramètre {⟨*blanc haut*⟩} de `\titlespacing`) ;
- *c* marge supplémentaire à gauche, paramètre {⟨*gauche*⟩} de `\titlespacing`, la même marge supplémentaire est ajoutée pour les deux parties du titre ;
- *d* marge supplémentaire à droite, paramètre [⟨*droite*⟩] de `\titlespacing`, la même pour les deux parties ;
- *e* blanc vertical entre les deux parties du titre, paramètre {⟨*séparation*⟩} de `\titleformat` ;
- *f* blanc vertical après le titre, paramètre {⟨*blanc bas*⟩} de `\titlespacing` ;
- *g* indentation du paragraphe suivant le titre, supprimée si tu utilises `\titlespacing*` et conservée si tu utilises `\titlespacing`.

FIG. 6.22: Format display du package `titlesec`

6.11.6 Le format block

La figure 6.21 page ci-contre présente schématiquement la mise en forme d'un titre de format block. Les paramètres de mise en forme sont interprétés de la même façon que pour hang, sauf que le numéro ne dépasse pas sur le côté gauche du titre. L'intérêt principal est de pouvoir faire facilement un titre centré.

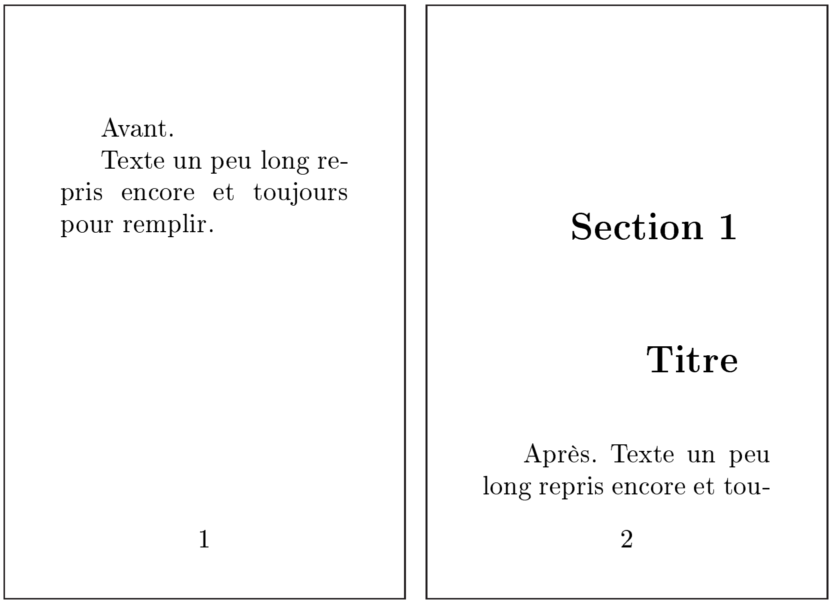
6.11.7 Le format display

La figure 6.22 présente schématiquement un titre au format display. Dans ce format là, le titre est décomposé en deux parties. La première reprend le {⟨*numéro*⟩} et la deuxième reprend le titre lui-même. C'est par exemple ce que fait LaTeX tout seul pour les chapitres. Ca fait partie des formats pour lesquels ils est classique d'utiliser une classe top pour que le titre soit toujours en haut d'une nouvelle page.

Dans ce format, les marges supplémentaires à gauche et à droite s'appliquent aux deux parties, et sont définies comme d'habitude par les paramètres {⟨*gauche*⟩} et [⟨*droite*⟩] de `\titlespacing`. Le blanc vertical entre les deux parties est définit par le paramètre {⟨*séparation*⟩} de `\titleformat`.

Le contenu de la première partie est le paramètre {⟨*numéro*⟩} de `\titleformat`. Si tu veux re-définir la commande `\chapter` de LaTeX, il te faut tenir compte de certaines contraintes. La première est que le mot chapitre doit dépendre de la langue utilisée. Pour ça, c'est facile, il suffit d'utiliser la commande `\chaptername` : le package `babel` se charge de la redéfinir quand on change de langue. Mais cette précaution n'est pas suffisante : dans l'annexe, les chapitres sont des annexes. Il vaut donc mieux utiliser la commande `\chaptertitlename`, fournie par le package, qui renverra sur `\chaptername` quand on est dans la partie princi-

```
* \usepackage[frenchb]{babel}
* \usepackage{titlesec}
* \titleclass{\section}{top}
* \titleformat{\section}[display]
*   {\filleft\Large\bfseries}
*   {Section \thesection}
*   {2em}{\bfseries}
* \titlespacing*{\section}
*   {0pt}{2em}
*   {2em}[12pt]
* % \texte comme d'habitude
  Avant.\par
  \texte\par
  \section{Titre}
  Apr\`es. \texte \par
  Plus loin. \texte \par
```



Cet exemple montre comment on peut créer un titre au format display. L'appel à `\titleclass` indique qu'il faut changer de page avant le titre, sans cet appel le titre pourrait commencer sur la première page (s'il y a assez de place).

La commande `\filleft` est utilisée pour coller le titre sur la droite, comme la commande `\chapter` de LaTeX. Bien entendu ou pourrait préférer d'autres alignements.

Enfin, sur cet exemple une marge supplémentaire à droite de 12 points est utilisée, pour que le titre ne soit pas aligné sur la marge. On pourrait, à l'opposé, utiliser une valeur négative pour que le titre dépasse dans la marge.

FIG. 6.23: Exemple d'utilisation du format display du package `titlesec`

---

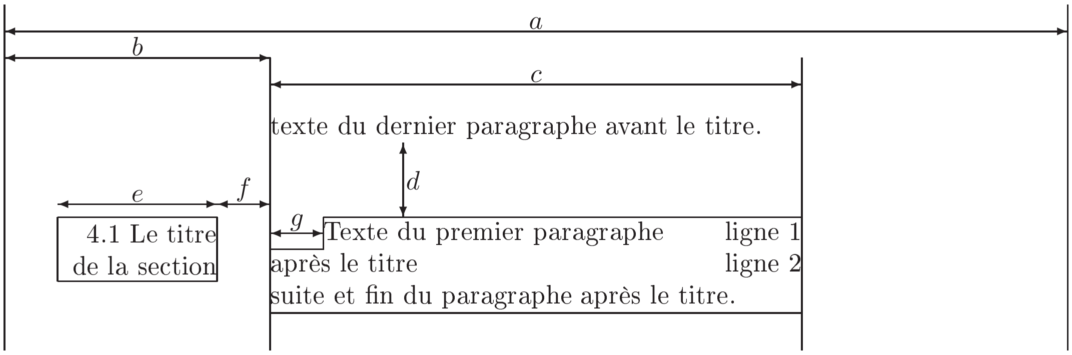
pale du document et qui renverra sur `\appendixname` quand on est dans les annexes. Et ça tombe rudement bien : `babel` s'occupe aussi de redéfinir `\appendixname` quand on change de langue.

Là où ça devient plus compliqué, c'est si tu veux définir un nouveau mot, autre que chapitre. Par exemple « titre », comme on en croise parfois dans les textes de loi. Le plus simple sera alors de compter sur `babel` pour s'en occuper. Ça pourra donner par exemple :

```
\addto\captionsfrench{%
  \renewcommand\titlename{Titre}}
\addto\captionsenglish{%
  \renewcommand\titlename{Title}}
```

Pour ce type d'usage, il vaut mieux cependant être relativement familier avec le package `babel`, et savoir l'étendre proprement. En effet, ce que je viens de t'indiquer là créera des problèmes si la langue anglaise n'est pas chargée dans `babel`. Pour en apprendre plus sur la façon d'apprendre à `babel` à produire le bon texte dans la bonne langue, reportes toi au passage sur ce package (titre 5 page 584) au chapitre neuvième page 571 qui traite de la programmation.

La figure 6.23 donne un exemple complet, mais assez minimal, d'utilisation du format display.



- *a* largeur du papier (`\paperwidth`) ;
- *b* marge de gauche du document (`\oddsidemargin` ou `\evensidemargin` selon qu'on soit sur une page recto ou verso) ;
- *c* largeur de la zone de texte (`\textwidth`) ;
- *d* blanc vertical avant le titre, paramètre {⟨*blanc haut*⟩} de `\titlespacing` ;
- *e* largeur du titre, définie par le paramètre {⟨*gauche*⟩} de `\titlespacing` (oui, c'est curieux, mais c'est comme ça) ;
- *f* séparation entre le titre et la marge, définie par le paramètre {⟨*blanc bas*⟩} de `\titlespacing` ;
- *g* indentation du paragraphe après le titre, conservée ou non, selon que tu utilises `\titlespacing` ou `\titlespacing*`.

FIG. 6.24: Format leftmargin du package `titlesec`

6.11.8 Les formats **leftmargin** et **rightmargin**

La figure 6.24 présente schématiquement le format leftmargin. Le format rightmargin fonctionne exactement de la même manière, mais avec le titre dans la marge de droite.

Les paramètres d'espacement pour ce format sont utilisés de manière inhabituelle, par rapport aux formats que tu as déjà vu. En effet, la largeur du titre n'est plus basée sur la largeur du texte, puisque le titre sera dans la marge. De même, la notion de marge droite (ou gauche) supplémentaire n'a pas de sens, puisque le titre n'est plus dans le texte. Le package `titlesec` ne se soucie pas de la taille réelle de la marge dans laquelle il mettra le titre, ça, c'est le boulot de packages comme `geometry` ou `vmargin`. La largeur du titre est fixée par le paramètre {⟨*gauche*⟩} de `\titlespacing`, et la distance entre le titre et la marge est fixée par le paramètre {⟨*blanc bas*⟩} de `\titlespacing` (techniquement, c'est bien le blanc, horizontal, qui est entre le titre et le paragraphe suivant, simplement, dans ce format, le titre et le paragraphe suivant sont au même niveau).

Le paramètre {⟨*blanc haut*⟩} de `\titlespacing` joue son rôle habituel : il représente le blanc vertical avant le titre, qui sera *de facto* le blanc entre le paragraphe avant le titre et le paragraphe après le titre.

C'est probablement le format le plus complexe à utiliser, pour plusieurs raisons. D'abord, c'est un format dans lequel le titre se trouve, a priori, dans une largeur très restreinte (la marge), du coup si tu l'appliques sur un document qui n'est pas fait pour, tu vas avoir des titres ignoblement laids et sur douze lignes. Ensuite, le titre ne peut pas être coupé entre deux pages, du coup, le paragraphe non plus, ou plus exactement il faut que le ou les paragraphes après le titre soient en un seul bloc, or LaTeX n'offre pas facilement ce type de garantie. Le package essaye de prendre des précautions dans ce cas là, mais c'est un problème convenablement complexe pour qu'il puisse ne pas y arriver. Une solution simple pour se prémunir de ce problème est de n'utiliser ce format que pour des titres de la classe top.

```
* \usepackage[frenchb]{babel}
* \usepackage{titlesec}
* \titleformat
*    {name=\section,page=even}
*    [leftmargin]
*    {\filleft\scshape}
*    {\thesection}{.5em}{}
* \titleformat
*    {name=\section,page=odd}
*    [rightmargin]
*    {\filright\itshape}
*    {\thesection}{.5em}{}
* \titlespacing{\section}
*             {.7in}{1em}{1em}
* % \texte comme d'habitude
* % \geometry pour les marges
  \section{Section}
  \texte \par
  \subsection{Essai}
  Rien. \texte \par
  \subsection{Encore}
  Rien non plus ici. \par
  \section{Section}
```

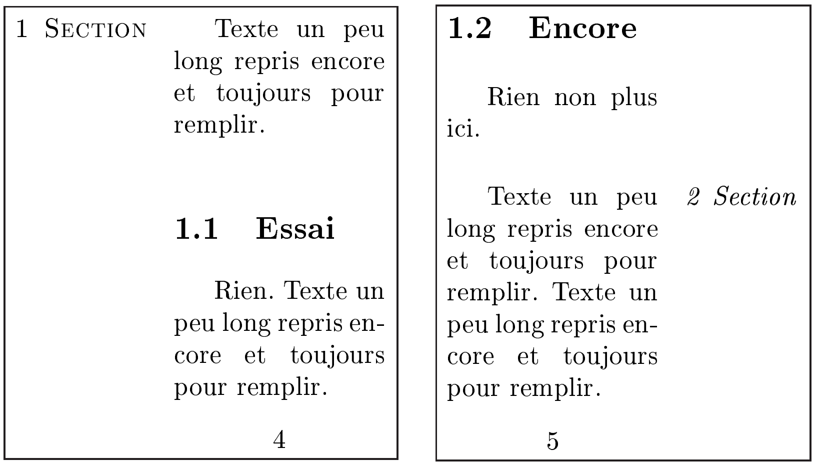


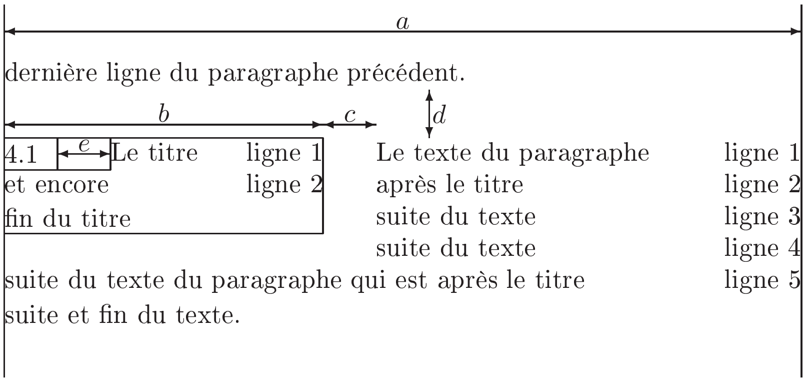
FIG. 6.25: Exemple d'utilisation du format leftmargin du package `titlesec`

Une autre difficulté est que, généralement, si une marge est assez large pour contenir un titre (mettons, pour l'exemple, que ce soit la marge extérieure), ce n'est pas forcément le cas des *deux* marges. Du coup, dire que le titre sera toujours dans la marge de droite risque de poser de gros problème si le titre tombe sur une page où la marge de droite n'est pas celle qui est assez grande (la marge extérieure pour notre exemple)[166], par exemple si le titre tombe sur une page verso. Pour règler ce problème, on peut utiliser les options avancées du premier paramètre de la commande `\titleformat` de manière à dire que le titre soit toujours dans la marge extérieure. Ça se dira comme ça :

```
% Page recto, marge de droite
\titleformat{name=\section,page=odd}
            [rightmargin]...
% Page verso, marge de gauche
\titleformat{name=\section,page=even}
            [leftmargin]...
```

La figure 6.25 présente un exemple d'utlisation de cette mécanique. Pour bien visualiser le fait que ce sont bien deux mises en forme différentes qui sont utilisées (avec également des marges et des alignements différents, si tu as envie), j'ai indiqué dans l'exemple que les titres au recto soient en italique et que les titres au verso soient en petites capitables. Bien entendu, c'est un choix qui, esthétiquement, n'est pas véritablement judicieux. Les marges de l'exemple ont été modifiées, par rapport aux exemples habituels, pour faire de la place

[166]Bien entendu, ce problème n'a de sens que pour les documents recto-verso où les marges gauches et droites sont alternées d'une page sur l'autre.



a largeur du texte (`\linewidth`) ;

b largeur du bloc de titre, paramètre {⟨*gauche*⟩} de `\titlespacing` ;

c largeur de la séparation horizontale entre le bloc de titre et le paragraphe de texte, paramètre {⟨*blanc bas*⟩} de `\titlespacing` (techniquement, c'est bien le blanc qui se trouve après le titre, il n'est juste pas en bas) ;

d blanc vertical avant le titre, paramètre {⟨*blanc haut*⟩} de `\titlespacing` ;

e blanc de séparation entre le numéro et le titre, paramètre {⟨*séparation*⟩} de `\titleformat`.

FIG. 6.26: Format drop du package `titlesec`

aux titres, le code correspondant n'est pas montré.

6.11.9 Les formats drop et wrap

La figure 6.26 présente schématiquement le format drop. C'est un format qui est relativement complexe à manipuler : le titre se retrouve sur une largeur relativement étroite (quand même moins que pour rightmargin) et du coup il peut donner des résultats curieux. Comme le texte vient se couler autour du titre, les mêmes problèmes de changement de page que pour rightmargin peuvent se poser.

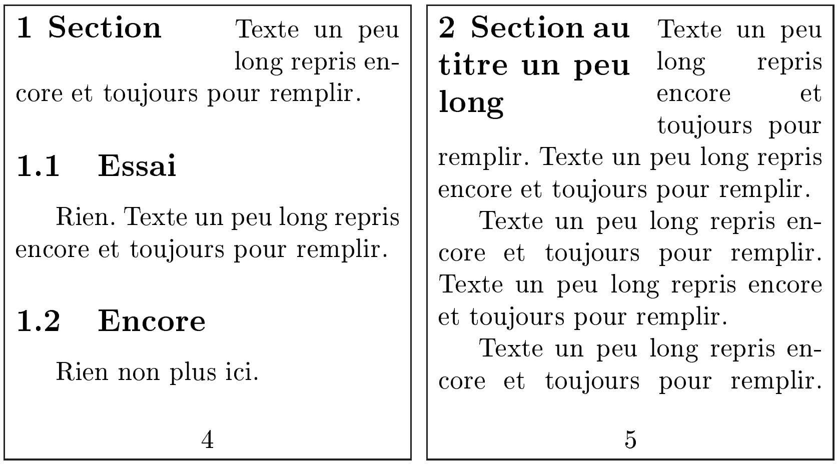
Comme d'habitude, le paramètre {⟨*numéro*⟩} de `\titleformat` précise la mise en forme du numéro, et {⟨*séparation*⟩} donne la taille du blanc entre le numéro et le titre. Le paramètre {⟨*avant*⟩} est du code de mise en forme qui sert pour le titre (et n'est pas appliqué au numéro), ainsi que le paramètre optionnel [⟨*après*⟩].

La vraie originalité est le traitement des paramètres de `\titlespacing`. Le paramètre {⟨*gauche*⟩} donne la largeur du bloc de titre, le paramètre {⟨*blanc haut*⟩} donne la taille du blanc vertical qui se trouve avant le titre, et {⟨*blanc bas*⟩} donne la taille du blanc horizontal entre le bloc de titre et le texte. Le paramètre optionnel [⟨*droite*⟩] n'a pas de sens pour ce format.

La différence entre drop et wrap est relativement simple : dans le second format, le titre n'est pas justifié par défaut, et est composé dans la plus petite largeur possible. Non seulement on supprime la justification, mais en plus on retient comme largeur du bloc de titre la plus longue des lignes obtenues[167]. Le résultat est donc que le texte du paragraphe après le titre se retrouve à une distance plus constante du titre : quand le titre est court, il n'occupe pas toute la largeur possible.

[167] Quand on compose un paragraphe sans le justifier, il est fréquent qu'aucune ligne n'atteigne spontanément la largeur à remplir.

```
* \usepackage[frenchb]{babel}
* \usepackage{titlesec}
* \titleformat{\section}
*     [drop]{\large\bfseries}
*     {\thesection}{1ex}{}[]
* \titlespacing*{\section}
*     {1in}
*     {1em plus 1ex}
*     {1em}
* % \texte comme d'habitude
* % \geometry pour les marges
  \section{Section}
  \texte \par
  \subsection{Essai}
  Rien. \texte \par
  \subsection{Encore}
  Rien non plus ici. \par
  \section{Section au titre
  un peu long}
```


1 Section Texte un peu long repris encore et toujours pour remplir.

1.1 Essai

Rien. Texte un peu long repris encore et toujours pour remplir.

1.2 Encore

Rien non plus ici.

4

2 Section au titre un peu long Texte un peu long repris encore et toujours pour remplir. Texte un peu long repris encore et toujours pour remplir.

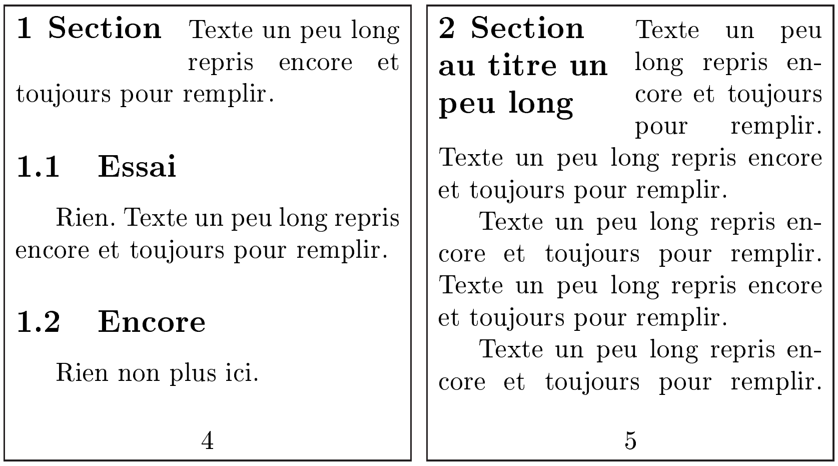
Texte un peu long repris encore et toujours pour remplir. Texte un peu long repris encore et toujours pour remplir.

Texte un peu long repris encore et toujours pour remplir.

5

FIG. 6.27: Exemple d'utilisation du format `drop` du package `titlesec`

```
* \usepackage[frenchb]{babel}
* \usepackage{titlesec}
* \titleformat{\section}
*     [wrap]{\large\bfseries}
*     {\thesection}{1ex}{}[]
* \titlespacing*{\section}
*     {1in}
*     {1em plus 1ex}
*     {1em}
* % \texte comme d'habitude
* % \geometry pour les marges
  \section{Section}
  \texte \par
  \subsection{Essai}
  Rien. \texte \par
  \subsection{Encore}
  Rien non plus ici. \par
  \section{Section au titre
  un peu long}
```


1 Section Texte un peu long repris encore et toujours pour remplir.

1.1 Essai

Rien. Texte un peu long repris encore et toujours pour remplir.

1.2 Encore

Rien non plus ici.

4

2 Section au titre un peu long Texte un peu long repris encore et toujours pour remplir. Texte un peu long repris encore et toujours pour remplir.

Texte un peu long repris encore et toujours pour remplir. Texte un peu long repris encore et toujours pour remplir.

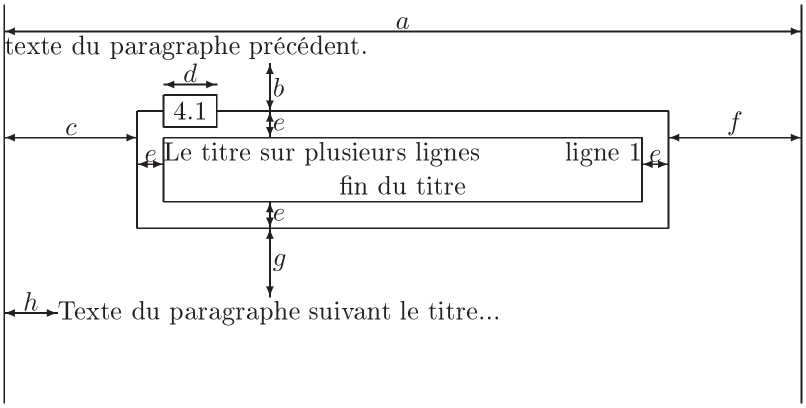
Texte un peu long repris encore et toujours pour remplir.

5

FIG. 6.28: Exemple d'utilisation du format `wrap` du package `titlesec`

Autrement dit, dans le cas de `wrap`, le paramètre {⟨*gauche*⟩} de `\titlespacing` indique la largeur *maximale* du bloc de titre.

Les deux figures 6.27 et 6.28 de la présente page présentent le même document. La première utilise le format `drop` pour les `\section`. La seconde utilise exactement les mêmes paramètres, mais avec le format `wrap`.



a largeur du texte (`\linewidth`) ;

b taille du blanc vertical avant le titre, paramètre {⟨*blanc haut*⟩} de `\titlespacing` ;

c marge supplémentaire à gauche, paramètre {⟨*gauche*⟩} de `\titlespacing` ;

d largeur du numéro tel que produit par le paramètre {⟨*numéro*⟩} de `\titleformat` ;

e séparation entre le cadre et le titre, paramètre {⟨*séparation*⟩} de `\titleformat` ;

f marge supplémentaire à droite, paramètre [⟨*droite*⟩] de `\titlespacing` ;

g taille du blanc vertical après le titre, paramètre {⟨*blanc bas*⟩} de `\titlespacing` ;

h indentation du premier paragraphe après le titre, maintenue si tu utilises `\titlespacing` et supprimée si tu utilises `\titlespacing*`.

FIG. 6.29: Le format frame du package `titlesec`

6.11.10 Le format frame

La figure 6.29 présente schématiquement le format frame. Dans ce format, le titre est décomposé en deux parties : le numéro, qui est placé dans la ligne du haut du cadre, et le titre, qui est encadré. Il semble raisonnable, mais pas obligatoire, de choisir une mise en forme centrée pour le titre. La figure 6.19 page 346 que nous avons déja vue propose un bon exemple d'utilisation de ce format, avec des titres qui dépassent dans les marges.

Les paramètres de `\titlespacing` jouent leurs rôles habituels : les blanc verticaux avant et après le titre, ainsi que les éventuelles marges supplémentaires à gauche ou à droite.

Pour ce qui concerne `\titleformat`, le paramètre {⟨*séparation*⟩} indique la distance entre le cadre et le texte qu'il contient. À noter également la position du résultat de {⟨*numéro*⟩} : si tu fais appel aux commandes d'alignement fournies par le package (`\filright` et compagnie), ça changera la position horizontale du numéro qui pourra être à gauche, au centre, ou à droite.

6.11.11 Options avancées du package

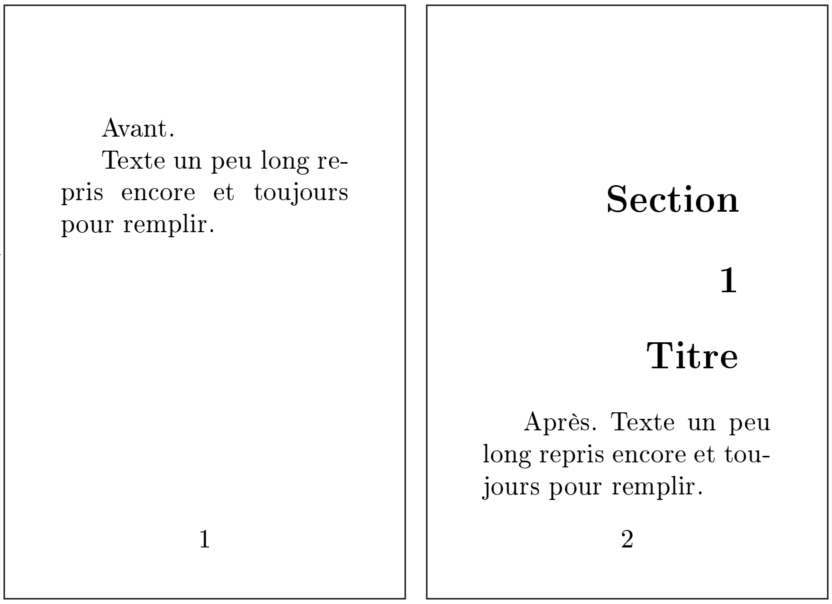
Dans son usage avancé, ce package prévoit un grand nombre d'options. Bien entendu, elles n'ont pas (ou peu) de sens dans l'usage simple. Ces options forment un enssemble hétéroclite, et font l'objet du tableau 6.6 page suivante. Certaines de ces options sont associées à des commandes.

L'option nobottomtitles s'assure qu'il reste au moins `\bottomtitlespace` en bas de la page avant d'y placer un titre. Cette commande est une longueur. Si l'option (par défaut) bottomtitles est utilisée, la commande vaut une longueur légèrement négative (-1 point) permettant à un titre d'être tout en bas de la page. Si tu utilises l'option nobottomtitles, la commande est définie pour valoir 20% de la hauteur de la page. Tu peux changer cette

Option	**Description**
indentafter	Force à ce que les paragraphes après les titres soient indentés (comme en français).
noindentafter	Force à ce que les paragraphes après les titres ne soient *pas* indentés (comme en anglais). L'interaction avec les autres packages, et avec les habitudes LaTeX ne sont pas claires.
rigidchapters	Force la hauteur des titres au format display à être constante, c'est à dire que {⟨*blanc haut*⟩} plus {⟨*blanc bas*⟩} est la *hauteur totale* entre la marge du haut et le début du texte quoi qu'il advienne.
rubberchapters	La hauteur des titres display dépend de la place que prend le titre. C'est le mode de fonctionnement par défaut de LaTeX et du package.
nobottomtitles	Cette option fait en sorte que si un titre se trouve en bas de page (trop près de la marge du bas), la page soit raccourcie (sans essayer de l'étirer sur toute la hauteur).
nobottomtitles*	Fait la même chose en utilisant un algorithme de clacul plus précis, mais cet algorithme peut poser problème avec les formats leftmargin, rightmargin, drop et wrap.
bottomtitles	Le contraire des deux options précédentes, la simple règle qui dit qu'il faut au moins une ligne de texte après le titre est appliquée.
aftersep	Quand deux titres se suivent, seul le {⟨*blanc bas*⟩} du premier titre est utilisé. Option par défaut.
largestsep	Avec cette option, le plus grand des deux blancs, entre {⟨*blanc bas*⟩} du premier titre et {⟨*blanc haut*⟩} du second, est utlisé.
calcwidth	Demande au package de calculer la longueur de la plus longue ligne du titre pour display, block et hang comme il le fait spontanément pour wrap. La longueur en question est disponible dans `\titlewidth`.
newparttoc	Indique au package de faire en sorte que les entrées en table des matières des `\part` soient comme les autres (ce n'est pas le cas par défaut avec LaTeX). Ça permet aux packages qui changent la table des matières, comme `titletoc` de manipuler les entrées de `\part` normalement.
oldparttoc	Indique de rester sur le comportement de LaTeX.
clearempty	Modifie `\cleardoublepage` pour que la page éventuellement blanche générée soit *entièrement* blanche (avec le style de page empty).
loadonly	Indique au package de ne pas chercher à récupérer la mise en forme des titres lors de son chargement. Utile si tu veux redéfinir tous les titres, ou si tu veux créer les tiens pour ta propre classe de documents.

TAB. 6.6: Options avancées du package `titlesec`

```
\usepackage[frenchb]{babel}
\usepackage{titlesec}
\titleclass{\section}{top}
\titleformat{\section}[display]
    {\Large\bfseries\filleft}
    {Section}
    {1em}
    {\titleline[r]{\thesection}
     \vspace{1em}}
\titlespacing*{\section}
  {0pt}{1em}
  {1em}[12pt]
% \texte comme d'habitude
Avant.\par
\texte\par
\section{Titre}
Apr\`es. \texte \par
Plus loin. \texte \par
```


Avant.
Texte un peu long repris encore et toujours pour remplir.

1

Section

1

Titre

Après. Texte un peu long repris encore et toujours pour remplir.

2

FIG. 6.30: Exemple d'utilisation de la commande `\titleline` du package `titlesec`

---

valeur à tout moment (y compris si tu n'as pas utilisé l'option, puisque l'option se borne à redéfinir la commande), par exemple en faisant :

`\renewcommand{\bottomtitlespace}{5cm}`

Attention cependant : si tu mets une valeur trop grande, le résultat pourrait être catastrophique. Il ne faut pas en abuser.

L'option calcwidth demande au package, pour les formats hang, block et display de calculer la longueur de la ligne la plus longue du titre. Bien entendu ce calcul n'a pas forcément toujours un sens, en particulier, si le titre fait plusieurs lignes et est justifié, le résultat est prévisible : la largeur de la justification. Le calcul qui est effectué est relativement juste, mais il donne une largeur vue par LaTeX, par une largeur à l'œil. Par exemple, pour le format hang, le numéro et le blanc de séparation sont indiqués à LaTeX comme ayant une largeur nulle, du coup, il n'en tient pas compte pour mesurer la longueur des lignes.

Une fois que le calcul est effectué, tu récupères la longueur de la plus longue des lignes dans `\titlewidth`, la longueur de la première ligne dans `\titlewidthfirst` et la longueur de la dernière ligne dans `\titlewidthlast`. C'est très pratique pour certaines manipulations, par exemple si tu choisis comme alignement pour le titre `\fillast` qui justifie l'ensemble sauf la dernière ligne qui est centrée. Connaissant la longueur de cette ligne (avant le centrage) tu peux placer en dessous un trait qui soit pile de la même longueur ; chose qui est infernale à faire sans cette aide.

6.11.12 Commandes aditionnelles

`\titleline[`⟨*alignement*⟩`]{`⟨*texte*⟩`}`

Permet d'inclure une ligne de texte, de la même largeur que le titre, à des endroits où le package aurait probablement refusé que tu en mettes, par exemple dans le paramètre

{⟨*avant*⟩} de certains formats[168]. Par défaut, la commande produit une ligne en l'étirant sur la largeur demandée (comme `\makebox` avec l'option `s`). Soit tu ajoutes le blanc aux bons endroits, soit tu changes l'alignement avec l'option (`l` pour aligner à gauche, `c` pour centrer, et `r` pour aligner à droite).

Par exemple, pour que le titre du format display soit sur trois lignes, au lieu de deux par défaut, tu peux faire quelque chose comme ça :

```
\titleformat{\section}[display]
    {\Large\bfseries\filleft}
    {Section}
    {2em}
    {\titleline[r]{\thesection}}
```

La figure 6.30 page précédente présente ce type de construction curieuse où {⟨*numéro*⟩} ne contient pas le numéro : le numéro est affiché par l'appel à `\titleline` dans {⟨*avant*⟩}.

`\titleline*`[⟨*alignement*⟩]{⟨*texte*⟩}

Cette variante fonctionne de manière légèrement différente. Dans un premier temps, le texte est mis dans une boîte de largeur `\titlewidth` (ça ne marche donc pas si tu n'as pas utilisé l'option calcwidth) qui elle même est placée dans la largeur du texte avec l'alignement indiqué en option. Ça permet, par exemple, en choisissant judicieusement l'alignement, de placer un trait qui soit de la largeur du titre.

`\titlerule`[⟨*épaisseur*⟩]

Permet de placer un filet horizontal, sur toute la largeur, l'épaisseur par défaut est de 0,4 point (comme d'habitude). Ça peut s'utiliser dans à peu près tous les paramètres, et si nécessaire le package placera tout seul la commande dans un appel à `\titleline`.

`\titlerule*`[⟨*largeur*⟩]{⟨*texte*⟩}

Crée une ligne en répétant {⟨*texte*⟩} indéfiniment. Si tu précises [⟨*largeur*⟩] il placera un exemple de {⟨*texte*⟩} tous les [⟨*largeur*⟩]. Par exemple, pour faire un pointillé (je trouve ça laid, mais il y a des gens que ça amuse) :

```
\titlerule*[5mm]{.}
```

placera, en gros, un point tous les 5 millimètres.

La figure 6.31 page suivante présente un exemple d'utilisation de toutes ces commandes. La commande `\titlerule` est utilisée dans {⟨*formatage*⟩} pour placer un filet épais sur toute la largeur du titre (pas seulement `\titlewidth`) avant le titre (juste après le blanc vertical du haut[169]). La commande `\titleline` est utilisée dans {⟨*avant*⟩} pour écrire le numéro, comme dans l'exemple précédent. La commande `\titlerule*` est utilisée dans [⟨*après*⟩] pour dessiner un pointillé sous le titre. Pour que ce pointillé soit de la largeur naturelle du titre (et pas toute la place prévue entre les marges) on place l'appel à `\titlerule*` dans un appel à `\titleline*`. À noter la façon dont le paramètre [⟨*après*⟩] est passé : sa valeur est mise entre accolades pour que LaTeX ne se trompe pas dans les crochets.

[168] Pour certains formats, le paramètre {⟨*avant*⟩} (ou [⟨*après*⟩], d'ailleurs) ne peut contenir que du matériel vertical. Un texte n'est pas du matériel vertical, et si tu en mets, ça fera râler le package.
Les formats hang, block et display n'acceptent que du matériel vertical dans [⟨*après*⟩].
La majorité des formats n'acceptent que du matériel vertical dans {⟨*avant*⟩}.
Dans ces cas-là, tu peux utiliser cette commande pour ajouter du texte, ou un trait, par exemple.

[169] Si on voulait placer le filet *avant* le blanc, il faudrait indique à `\titlespacing` un {⟨*blanc haut*⟩} qui soit nul, et ajouter une appel à `\vspace` dans {⟨*formatage*⟩}.

```
\usepackage[frenchb]{babel}
\usepackage[calcwidth]
                      {titlesec}
\titleclass{\section}{top}
\titleformat{\section}[display]
    {\titlerule[.8pt]
     \Large\bfseries\filleft}
    {Section}
    {1em}
    {\titleline[r]{\thesection}
     \vspace{1em}}
    [{\titleline*[r]{%
      \titlerule*[5mm]{.}}}]
\titlespacing*{\section}
  {0pt}{1em}
  {1em}[12pt]
% \texte comme d'habitude
Avant.\par
\texte\par
\section{Titre}
Apr\`es. \texte \par
Plus loin. \texte \par
```

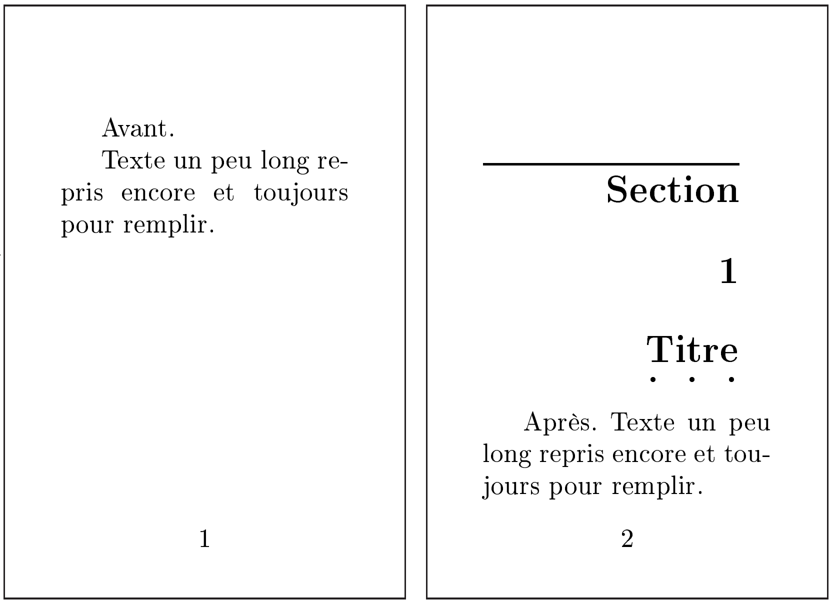


FIG. 6.31: Exemple d'utilisation des commandes `\titlerule`, `\titlerule*`, `\titleline` et `\titleline*` du package `titlesec`

Si on avait écrit ça :

```
\titleformat{\section}[display]
    ....
    [\titleline*[r]{\titlerule*[5mm]{.}}]
```

ce qui semble plus naturel, alors LaTeX aurait pris comme valeur de `[`⟨*après*⟩`]` `\titleline*[r`. Hé oui : il s'arrête au premier crochet fermant qu'il croise (LaTeX sait compter les accolades pour en vérifier l'imbrication, mais il ne sait pas compter les crochets). En ajoutant des accolades autour du contenu de `[`⟨*après*⟩`]` LaTeX ne *voit pas* les crochets qui sont cachés dans les accolades, et trouve donc le bon crochet fermant.

En règle générale, on considère que, juste avant un titre, c'est pas mal comme changement de page. Le package respecte cette habitude. Il te permet même de dire à quel point c'est bien de changer de page avant un titre. LaTeX choisit ses changements de page en utilisant un système de pénalités. Plus la pénalité est élevée, plus c'est grave de changer de page. Si on atteind 10.000, c'est infiniment grave, donc interdit. Si on est à 0, c'est normal (genre entre deux lignes de texte). Si on est à -10.000, c'est infiniment bien, donc obligatoire.

Si tu as une commande de sectionnement `\MaSection` (donc pas forcément une de celles de LaTeX), et que tu définit la commande `\MaSectionbreak`, le package s'en sert pour déterminer à quel point c'est bien de changer de page avant une section de ce type là. Si tu n'as pas définit `\MaSectionbreak`, il applique la règle habituelle de LaTeX, à savoir une pénalité de -300, ce qui est déjà bien incitatif. Quelques exemples de définition :

```
% Bon point de coupure, conserver le blanc avant le titre
% si on change de page
```

Court	Long	Description
a ou n	arabic	Chiffres arabes (6).
R	Roman	Chiffres romains majuscules (VI).
r	roman	Chiffres romains minuscules (vi).
L	Letter	Lettres majuscules (F).
l	letter	Lettres minuscules (f).
g	greek	Lettres grecques minuscules (ζ)
d	doubleletter	Doubles lettres minuscules (ff).
b	doublegreek	Doubles lettres grecques minuscules ($\zeta\zeta$)

TAB. 6.7: Numérotations connues par le package `alnumsec`

```
\newcommand\MaSectionbreak{%
  \addpenalty{-500}
  \vspace*{0cm}
}
% Changer de page tout le temps (comme la classe top)
\newcommand\MaSectionbreak{%
  \clearpage
}
% Ouvrir sur une page de gauche (comme les chapitres du
% TeXbook), \cleartoevenpage vient de epigraph
\usepackage{epigraph}
\newcommand\MaSectionbreak{%
  \cleartoevenpage
}
```

6.11.13 Autres fonctionnalités

Le package `titlesec` offre également des fonctionnalités en dehors de la mise en forme des titres, en particulier il propose une approche intéressante pour définir les styles de page et pour utiliser les marques. Ces fonctionnalités sont abordées au paragraphe 4.4 page 203.

6.12 Changer la numérotation : le package `alnumsec`

Ce package, que nous devons à Frank KÜSTER [113], permet de changer complètement l'approche de la numérotation des titres de LaTeX. Il permet de revenir à un modèle qu'on connaissait autrefois (même s'il n'a jamais été intensivement utilisé dans des livres), et surtout un modèle que les profs aiment utiliser quand ils font leurs cours au tableau.

Le package permet, en quelques commandes simples, de choisir des numérotations différenciées selon le niveau (certains en chiffres romains, d'autres en chiffres arabes, avec des lettres grecques, etc). Spontanément, il ne reprend pas le numéro de la section précédente. Par exemple, si les chapitres sont numérotés en chiffres romains majuscules, et si les sections sont numérotées en chiffres arabes, la deuxième section du deuxième chapitre sera simplement numérotée « 2 ». On peut lui dire, en utilisant l'option usehighlevels lors du chargement du package, de garder le numéro de la section d'au-dessus. En pareil cas, notre section sera numérotée « II.2 ».

Le package fournit une commande principale, qui te permet de décrire la numérotation

```
\usepackage{alnumsec}
\alnumsecstyle{RLalg}
\surroundletter{}{()}
\surroundgreek{}{[]}
\section{Section}
\label{sect}
\subsection{Ss-section}
\label{ssect}
\subsubsection{Ss-ss-section}
\paragraph{Paragraphe}
\subparagraph{Ss-paragraphe}
La section est \ref{sect}, la
sous-section est \ref{ssect}.
```

I. Section

A. Ss-section

1. Ss-ss-section

a) Paragraphe

α**] Ss-paragraphe** La section est I., la sous-section est A..

1

FIG. 6.32: Exemple d'utilisation du package `alnumsec`

---

que tu veux, et des commandes annexes pour règler les détails. La commande centrale est `\alnumsecstyle` qui prend en paramètre la liste des types de numérotation, le premier style sera utilisé pour le premier niveau de sectionnement (`\chapter` ou `\section`, selon ta classe de document), le deuxième au niveau suivant, et ainsi de suite. Le style est représenté par une lettre : celle de la colonne de gauche du tableau 6.7 page précédente.

Par défaut, pour tous les styles de numérotation (sauf les lettres minuscules et les lettres grecques minuscules, où le package utilise une parenthèse fermante), le package utilise un point après le numéro. Ce point fait partie, pour lui, du numéro de la section : si tu fais une référence vers la section concernée, le point est repris dans la référence, au risque d'obtenir un double point en fin de ligne (ce qui est plutôt ennuyeux) : tu trouves un exemple de ce résultat malencontreux dans l'exemple de la figure 6.32.

Ce comportement par défaut peut être modifié. Le package te permet de modifier ce qu'il place à gauche et à droite des numéros. En fait, ce qu'il va modifier, c'est la définition qu'il associe au type de numérotation. Ainsi, quand tu précises `r`, par défaut, ça veut dire « *chiffre romain minuscule suivi d'un point* » ; et tu peux expliquer au package que tu préfèrerais que ce soit, par exemple « *chiffre romain minuscule entre parenthèses* ». Cet exemple précis se dit avec la commande `\surroundroman` :

```
\surroundroman{(}{)}
```

Après cet appel, à chaque fois que tu feras référence à `r` dans `\alnumsecstyle`, le package produira des chiffres romains minuscules entre parenthèses pour le niveau de sectionnement correspondant. Dans l'esprit qui préside à l'utilisation de ce package, chaque niveau de sectionnement est supposé avoir son propre mode de numérotation, du coup le fait de pouvoir personnaliser chacun des modes de numérotation devrait suffir à pouvoir personnaliser tous les niveaux de sectionnement.

`\surroundroman[`⟨*gauche*⟩`][`⟨*droite*⟩`]{`⟨*gauche*⟩`}{`⟨*droite*⟩`}`

Toutefois, le package prévoit que tu puisse utiliser *deux fois* le même type de numérotation pour deux niveaux de titre différents. Dans ce cas-là, les arguments {⟨*gauche*⟩} et {⟨*droite*⟩} seront utilisés spontanément pour ce type de numérotation. Bien entendu les commandes équivalentes existent pour les autres numérotations, leur nom est systématiquement `\surround`⟨*nom*⟩ où ⟨*nom*⟩ est le nom long du mode de numérotation (deuxième colonne du tableau 6.7 page 362). Les informations définies par `[`⟨*gauche*⟩`]` et `[`⟨*droite*⟩`]` seront utilisées pour les niveaux de sectionnement que tu indiqueras comme étant alternatifs : tous les numéros de type `r` apparaîtront par défaut en utilisant {⟨*gauche*⟩} et {⟨*droite*⟩}, sauf ceux que tu auras spécifié comme utilisant la forme alternative (`[`⟨*gauche*⟩`]` et `[`⟨*droite*⟩`]`).

Il y a un niveau de sectionnement n tel que tous les niveaux de sectionnement strictement avant utilisent la forme normale et tous les niveaux suivant (y inclus n) utilisent la forme alternative. Ce niveau se précise en utilisant la commande `\otherseparators`. Par exemple, si tu utilises des chiffre romains minuscules pour le niveau 3 (sous-sous-section) et pour le niveau 5 (sous-paragraphe), tu peux faire ça pour que le niveau 5 ait son numéro entre parenthèses[170] :

```
\alsecnumstyle{Rnrlr}
\surroundroman[(][)]{}{.}
\otherseparators{5}
```

`\alnumsectionlevels`{⟨*départ*⟩}{⟨*liste des niveaux*⟩}

Enfin, la dernière commande fournie par ce package, qui n'est que rarement utile, te permet d'utiliser ce package sur une classe qu'il ne connaît pas. En effet, pour choisir les niveaux de sectionnement qu'il saura traiter, et pour en déterminer l'ordre, le package se base sur le nom de la classe de document que tu utilises. Si tu utilises une classe de document dans laquelle la commande `\chapter` est définie, il va faire cet appel là, considérant que le niveau le plus élevé est le chapitre, et qu'il porte le numéro 0 (c'est le choix fait dans les classes **report** et **book** de LaTeX) :

```
\alnumsectionlevels{0}{chapter,section,subsection...}
```

par contre si `\chapter` n'est pas définie, il fera cet appel-là, considérant que le niveau le plus élevé est la section et qu'elle porte le numéro 1 :

```
\alnumsectionlevels{1}{section,subsection,...}
```

Si tu veux te faire ta propre classe de document, avec tes propres niveaux de sectionnement, tu peux faire appel à cette commande pour que le package s'adapte à ta classe.

Si tu regardes attentivement l'exemple de la figure 6.32 page précédente, et en particulier le α, tu verras qu'il n'est pas en gras. C'est le seul des numéros qui ne soit pas en gras. En fait, les numéros sont mis en gras avec `\textbf` ou avec `\bfseries`, or ces commandes s'appliquent à du *texte*. Il se trouve que le α utilisé par le package `alnumsec` est en *maths*, et donc que les changements de style du texte ne le concernent pas. Tu connais déjà la commande qui permet de dire que les maths doivent être en gras (`\mathversion` avec le paramètre `bold`). Il te suffit de l'utiliser dans `\surroundgreek`, dans la partie {⟨*gauche*⟩}, et du coup ton numéro sera bien en gras, comme te le prouve l'exemple de la figure 6.33 page ci-contre. Tu peux également constater que ça n'a pas d'influence sur les maths du document (le changement est local). Et, bien entendu, tu rencontreras le même problème avec la numérotation en doubles lettres grecques.

[170] Je trouve que ce mode de configuration est spécialement tordu : par moment on raisonne en terme de type de numérotation, et à d'autres moments on raisonne en terme de profondeur des niveaux de sectionnement concernés. Bizarre. Mais enfin, ça marche, du moins pour la majorité des cas raisonnables.

```
\usepackage{alnumsec}
\alnumsecstyle{RLalg}
\surroundletter{}{)}
\surroundgreek
  {\mathversion{bold}}{]}
\section{Section}
\label{sect}
\subsection{Ss-section}
\label{ssect}
\subsubsection{Ss-ss-section}
\paragraph{Paragraphe}
\subparagraph{Ss-paragraphe}
La section est \ref{sect}, la
sous-section est \ref{ssect}.

$\alpha$ est normal dans le texte.
```

I. Section

A. Ss-section

1. Ss-ss-section

a) Paragraphe

$\boldsymbol{\alpha}$] Ss-paragraphe La section est I., la sous-section est A..

α est normal dans le texte.

1

FIG. 6.33: Exemple d'utilisation du package `alnumsec`

6.13 Numérotation et table des matières : le package `tocvsec2`

Ce package, que nous devons à Peter WILSON [227], permet de modifier la profondeur de numérotation des sections, ainsi que la profondeur de la table des matières, non pas globalement, comme le prévoit LaTeX, mais localement : chapitre par chapitre ou section par section, par exemple.

Ce package sert, essentiellement, à modifier la valeur de `tocdepth` (profondeur de la table des matières) et/ou de `secnumdepth` (profondeur de numérotation des titres) avec une interface plus simple et plus riche que celle de LaTeX. Le premier grand changement est que tu peux désigner les niveaux de sectionnement par leur nom, plutot que par la valeur numérique de leur profondeur : tu peux lui dire de numéroter jusqu'aux sous-sous-sections, sans avoir à te souvenir que la profondeur correspondante est 2.

`\maxtocdepth{`⟨*titre*⟩`}`

`\settocdepth{`⟨*titre*⟩`}`

La première commande, `\maxtocdepth`, dont le nom est particulièrement mal choisi, est prévue pour changer globalement la profondeur de la table des matières dans un document. Cette commande doit être utilisée avant la table des matières, et indique la profondeur maximale en vigueur au début de la table des matières. Ensuite, dans le cours de ton document, tu peux à tout moment changer cette profondeur en appelant `\settocdepth`. Pour tout ce qui se trouve après l'appel, la table des matières aura la pronfondeur que tu as demandé. Ainsi, si tu souhaites que dans le premier chapitre la table des matières aille jusqu'aux sections seulement, et que dans le deuxième chapitre elle aille jusqu'aux sous-sous-section, ça se demande comme ça :

```
\chapter{Premier}
\settocdepth{section}
...
\chapter{Deuxi\`eme}
```

```
\settocdepth{subsubsection}
...
```

\maxsecnumdepth{⟨*titre*⟩}

\setsecnumdepth{⟨*titre*⟩}

Ces deux commandes fonctionnent comme les deux précédentes : `\maxsecnumdepth` est prévu pour être utilisé dans le préambule de ton document, pour indiquer la valeur de départ, et la commande `\setsecnumdepth` peut être utilisée à tout moment pour modifier cette valeur. Comme les deux commandes précédentes, elles prennent en paramètre le *nom* d'un niveau de titre.

6.14 Faire un sommaire : le package `shorttoc`

Ce petit package, que nous devons à Jean-Pierre F. Drucbert [55], permet de résoudre un problème classique quand on veut respecter les usages typographiques français avec LaTeX : faire un sommaire.

En effet, l'usage pour un livre français est d'avoir au début du livre un sommaire, qui fasse une page au plus, et qui reprenne les titres principaux (typiquement les titres de chapitre ou de partie), et éventuellement en fin d'ouvrage une table des matières qui, elle, peut être très longue.

La première idée qui vient, pour faire ça avec LaTeX, serait d'utiliser deux appels à `\tableofcontents` en changeant la valeur de `tocdepth` entre les deux, quelque chose comme ça :

```
\setcounter{tocdepth}{0}
\tableofcontents %Sommaire
...
\setcounter{tocdepth}{3}
\tableofcontents %La totale
```

Malheureusement, cette solution-là ne marche pas. En effet, la commande `\tableofcontents` fait des choses dans ton dos, pour être précis elle fait tout ça :

- si le fichier `.toc` existe, l'inclure pour faire la table des matières ;
- allouer un nouveau fichier de sortie ;
- ouvrir ce nouveau fichier en écrasant l'ancien `.toc`.

Du coup, lors du premier appel à `\tableofcontents`, on obtient un sommaire[171], mais ce premier appel écrase le fichier `.toc`, et du coup lors du deuxième appel, le fichier est vide ; du coup le deuxième appel à `\tableofcontents` produit une table des matières vide (il n'y a que le titre).

\shorttableofcontents{⟨*titre*⟩}{⟨*profondeur*⟩}

La commande `\shorttableofcontents` permet d'inclure une table des matières ayant un autre titre (le premier paramètre) et une autre profondeur (le second) que la table des matières principale qu'on obtient avec `\tableofcontents`. Attention : la table des matières principale doit apparaître en dernier, puisqu'elle écrase le fichier `.toc`, mais surtout *elle doit appraître.* En effet, c'est elle qui déclenchera le fait que le fichier `.toc` existe, si tu ne

[171] Il reste cependant le fait qu'il ne s'appelle pas sommaire, mais ce problème-là serait assez simple à résoudre.

l'appelles pas, tu n'auras pas de table des matières du tout. La commande `\shorttoc` est la même, mais avec un nom d'une longueur plus raisonnable.

Ainsi, pour obtenir un sommaire en début de volume, tu peux faire ça :

```
\shorttoc{Sommaire}{0}
...
\tableofcontents
```

Si tu veux obtenir la construction inverse, c'est-à-dire un sommaire en fin de volume, il suffit de tricher en faisant produire à `\shorttoc` la table des matières et en faisant produire le sommaire à `\tableofcontents`, par exemple en faisant ça[172] :

```
\shorttoc{Table des mati\`eres}{3}
...
\setcounter{tocdepth}{0}
\renewcommand\contentsname{Sommaire}
\tableofcontents
```

Le package `shorttoc` peut être chargé avec deux options. La première, prise par défaut, loose, indique de respecter l'espacement vertical de la table des matières que prévoit LaTeX. La seconde, tight, indique de resserer légèrement. En effet, quand on ne garde que les chapitres, par exemple pour faire un sommaire, il y a trop de blanc entre les titres de chapitres.

`\anothertableofcontents[`⟨*suffix*⟩`]{`⟨*document*⟩`}{`⟨*titre*⟩`}{`⟨*profondeur*⟩`}`

La commande `\anothertableofcontents` permet de reprendre dans ton document la table des matières d'un autre document (en supposant, bien entendu, que le fichier `.toc` de cet autre document soit accessible). Le premier paramètre, optionnel, te permet de récupérer un autre fichier que le `.toc`, par exemple pour reprendre la liste des tableaux tu indiqueras `lot`. Le deuxième paramètre indique le nom de l'autre document (le nom du fichier principal, sans l'extension) ; le troisième indique le titre que tu veux donner à cette table, et enfin le dernier paramètre indique la profondeur souhaitée.

La commande `\anothertoc` est la même, sous un nom plus court.

Enfin, il faut malheureusement noter que ce package est assez brutal dans son mode de fonctionnement. Il est donc parfaitement incompatible avec la majorité des packages qui interviennent avec les titres et/ou avec la table des matières. Si tu utilises le package `tocvsec2` (voir à ce sujet le paragraphe 6.13 page 365) le sommaire sera faux, puisqu'il respectera les directives de `tocvsec2` pour la profondeur. Si tu utilises des packages qui modifient les marques associées aux chapitres et sections, ces modifications ne seront pas prises en compte pour le chapitre (ou la section, selon la classe de documents) du sommaire. Cependant, dans un document simple qui a besoin d'un sommaire, ça reste une très bonne solution.

6.15 Une table par chapitre : le package `minitoc`

Ce package, que nous devons à Jean-Pierre F. Drucbert [56, 57], a un but simple à expliquer mais complexe à atteindre avec LaTeX : obtenir une table des matières par chapitre.

[172] La façon dont je re-définis la commande `\contentsname` dans cet exemple est un peu cavalière, mais elle ne devrait pas poser de problème : cette commande sera re-définie par `babel` à chaque changement de langue, mais la langue ne devrait pas changer avant que le titre de la table des matières ait été produit ; par contre elle peut changer pendant la table des matières, par exemple si un titre de chapitre est dans un passage en espagnol. Si tu veux changer le titre de la table des matières, la méthode officielle est celle-là :
`\addto\captionsfrench{\def\contentsname{Sommaire}}`

Commande	Description
`\doparttoc`	Prépare des tables des matières (`toc` pour « table of contents ») au niveau des parties (`\part`).
`\doparllof`	Prépare des listes de figures (`lof` pour « list of figures ») au niveau des parties.
`\dopartlot`	Prépare des listes de tableaux (`lot` pour « list of tables ») au niveau des parties.
`\dominitoc` `\dominilof` `\dominilot`	Idem au niveau chapitre.
`\dosecttoc` `\dosectlof` `\dosectlot`	Idem au niveau section.

TAB. 6.8: Commandes de `minitoc` pour préparer les mini-tables

Le but est en fait subtilement plus complexe : obtenir une table des matières, et/ou une liste des figures, et/ou une liste des tableaux, par chapitre, et/ou par partie, ou par section[173]. Pour la table des matières d'un chapitre, on parle de « minitoc », pour la liste des figures d'un chapitre, de « minilof » et pour la liste des tableaux de « minilot » ; pour les même choses mais au niveau section on parlera de « secttoc », de « sectlof » et de « sectlot » ; et au niveau des parties on parlera de « parttoc », de « partlof » et de « partlot ». On regroupera les neuf notions sur le terme générique « mini-table ».

Bien entendu, ce package n'a de sens que s'il est utilisé avec une classe de documents qui a la notion de titre, de sous-titre, etc. Ça n'a par exemple pas de sens de vouloir l'utiliser avec la classe **letter** ou avec **lettre**.

Le package fonctionne essentiellement en trois temps : d'abord il prépare ses mini-tables en lisant les fichiers de tables des matières globaux (les `.toc`, `.lot` et `.lof`) et en écrivant autant de petits fichiers qu'il y a de mini-tables ; ensuite[174] il fait confiance à `\tableofcontents`[175] pour faire son travail et collecter toutes les informations utiles ; enfin dans les chapitres (ou sections ou parties) où tu voudras des mini-tables, il te suffira de les demander.

Le tableau 6.8 reprend les neuf commandes utiles pour la première phase de l'utilisation de `minitoc`. Tu ne devrais en toute logique pas utiliser toutes ces commandes dans ton document. Ces commandes sont à utiliser *après* le préambule (donc après `\begin{document}`), mais *avant* `\tableofcontents` et ses associés.

Le tableau 6.9 page suivante reprend, lui, les neuf commandes correspondantes que tu utiliseras tout au long de ton document pour faire apparaître tes mini-tables là où tu les veux.

Le package donne des titres aux mini-tables qu'il crée. Par défaut, ces titres sont en

[173]Le package prévoit les tables par partie si la classe de documents possède un niveau `\part`. Le package ne prévoit les tables par section que si la classe de document ne possède pas de niveau `\chapter`.

[174]Il est très important que cette étape soit *après* la préparation, je te rappelle que la commande `\tableofcontents` écrase le fichier `.toc`. Si tu essayais d'inverser les deux premières étapes, la préparation des mini-tables ne serait plus possible, le fichier `.toc` étant devenu vide. Bien entendu la même remarque vaut pour les listes des figures et des tableaux.

[175]Si tu ne veux pas mettre de table des matières globale dans ton document, jugeant que les tables partielles suffiront, il faut quand même que cette étape ait lieu. Pour ce cas précis, le package prévoit trois commandes qui vont créer les tables sans les imprimer : `\faketableofcontents` pour la table des matières, `\takelistoftables` pour la liste des tableaux et `\fakelistoffigures` pour la liste des figures.

Commandes	Description
`\parttoc`, `\partlof`, `\partlot`	Fait apparaître la table des matières, la liste des figure, ou la liste des tableaux, de la partie concernée.
`\minitoc`, `\minilof`, `\minilot`	Idem pour un chapitre.
`\secttoc`, `\sectlof`, `\sectlot`	Idem pour une section.

TAB. 6.9: Commandes de `minitoc` pour placer les mini-tables

```
\usepackage{minitoc}
\dosecttoc
\mtcselectlanguage{french}
\setcounter{tocdepth}{1}
\tableofcontents
\section{Une section}
\secttoc
\subsection{Ss-section}
\subsubsection{Ss-ss-section}
\subsection{Ss-section}
\subsection{Ss-section}
\texte
\section{Autre}
```

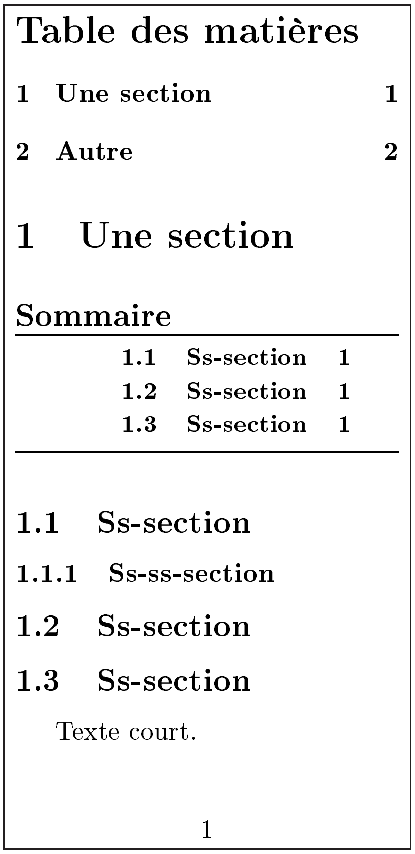


Table des matières

1 Une section **1**

2 Autre **2**

1 Une section

Sommaire

1.1 Ss-section 1
1.2 Ss-section 1
1.3 Ss-section 1

1.1 Ss-section

1.1.1 Ss-ss-section

1.2 Ss-section

1.3 Ss-section

Texte court.

1

FIG. 6.34: Exemple d'utilisation du package `minitoc`

1. afrikaan (afrikaans)
2. arab (arabic)
3. armenian
4. bahasa
5. bangla
6. basque
7. bicig
8. brazil
9. breton
10. bulgarian
11. buryat
12. catalan
13. chinese1
14. chinese2
15. croatian
16. czech
17. danish
18. dutch
19. english (american, canadian)
20. esperant (esperanto)
21. estonian
22. ethiopia (ethiopian)
23. finnish
24. french (frenchb, frenchle, frenchpro, francais, acadien, canadien)
25. galician
26. german (austrian)
27. germanb
28. greek
29. greek-mono
30. greek-polydemo
31. greek-polykatha
32. hangul1
33. hangul2
34. hangul3
35. hangul4
36. hanja1
37. hanja2
38. hebrew
39. icelandic
40. interlingua
41. irish
42. italian
43. japanese
44. japanese2
45. japanese3
46. japanese4
47. japanese5
48. latin
49. latvian (letton)
50. lithuanian
51. lsorbian
52. magyar (hungarian)
53. mongol
54. ngermanb (ngerman, naustrian)
55. norsk
56. nynorsk
57. polish
58. polish
59. portuges
60. romanian
61. russian
62. russianb
63. russianc
64. russian2m
65. russian2o
66. samin
67. scottish
68. serbian
69. serbianc
70. slovak
71. slovene
72. spanish (castillan)
73. swedish
74. thai
75. turkish
76. ukraineb
77. usorbian
78. vietnam (vietnamese)
79. welsh

TAB. 6.10: Langues prévues par le package `minitoc`

anglais. Pour changer la langue des titres, il prévoit une commande `\mtcselectlanguage` qui fonctionne comme le `\selectlanguage` de `babel`. La liste des langues disponibles fait l'objet du tableau 6.10.

La figure 6.34 page précédente présente un exemple simple d'utilisation du package `minitoc` : un document de type **article**[176], avec deux sections, une table des matières globale qui ne reprend que les titres de section (`tocdepth` est positionné à 1) et une table des matières au début de la première section qui reprend les titres des sous-sections.

6.15.1 Changer la profondeur

Par défaut, les mini-tables ne contiennent que les titres jusqu'aux sous-sections. Si tu veux plus de détails dans les mini-tables, tu peux modifier le compteur `minitocdepth` pour changer les mini-tables des chapitres, le compteur `secttocdepth` pour celles des sections ou `parttocdepth` pour celles des parties.

[176]Pas vraiment la classe **article** de LaTeX, mais la classe habituelle des exemples dans ce livre, qui est basée dessus. La principale différence étant la taille du papier par défaut.

Pour les listes de figures ou de tableaux, la notion de profondeur n'a pas de sens : on y recense les figures et les tableaux, voilà tout.

6.15.2 Perturber la numérotation

Le package `minitoc` a besoin de dresser une liste complète, et sans se tromper, de toutes les mini-tables qu'il doit créer. Pour ce faire il suppose que tu n'agis pas de manière trop brutale sur la numérotation naturelle de LaTeX, et il se charge d'intercepter les appels importants (début d'une chapitre, par exemple). Malheureusement, il y a un grand nombre d'occasions dans lesquelles, pas forcément de manière consciente, tu modifies cette numérotation.

La première, c'est quand tu crées, par exemple, un chapitre non-numéroté pour l'introduction de ton rapport, et que tu envoies ce chapitre en table des matières (on a évoqué ce point au paragraphe 3.1.3 page 13). Le compteur des chapitres n'est en effet pas incrémenté, et pourtant `minitoc` trouve la trace d'un chapitre dans le fichier `.toc`. Ça va très gravement le perturber. Pour palier à ce problème, le package prévoit que tu utilises la commande `\addstarredchapter` pour ajouter ton chapitre non-numéroté à la table des matières. De même il prévoit les commandes `\addstarredpart` pour les parties et `\addstarredsection` pour les sections. Ces commandes sont à utiliser à la place de `\addcontentsline` pour les niveaux de titre (section, chapitre et/ou parties) qui contiennent une mini-table. Pour les sections de niveaux inférieurs, on fait comme d'habitude, avec `\addcontentsline`. Ainsi, pour un chapitre d'introduction non numéroté qui contiendrait une mini-table, on pourrait procéder comme ça :

```
\chapter*{Introduction}
\addstarredchapter{Introduction}
\minitoc
...
\section*{Premi\`ere section}
\addcontentsline{toc}{section}{Premi\`ere section}
...
```

De la même manière, si quelque part dans ton document tu utilises un chapitre non numéroté, `minitoc` va s'embrouiller non plus dans le décompte de ses mini-tables, mais dans le décompte des chapitres. La solution est d'appeler la commande `\adjustmtc` dans le chapitre en question pour qu'il retrouve ses petits (`mtc` veut dire `MiniToC`). Le même problème existe pour les parties non-numérotées, on le résoud avec `\adjustptc`, et bien entendu, pour les section : `\adjuststc`.

`\firstchapteris`{⟨*numéro*⟩}

Dans certains documents, le premier chapitre n'est pas le chapitre 1 : soit on a un chapitre 0, soit on commence par un autre chapitre quelconque (genre le chapitre 4, parce que c'est le tome 2 d'un livre). Le package `minitoc` a besoin de compter les chapitres comme il faut, et pour ça, si le premier chapitre n'est pas le chapitre 1, il faut que tu lui dises où commencer avec la commande `\firstchapteris`. Bien entendu, les commandes analogues pour le numéro de la première partie (`\firstpartis`) et pour le numéro de la première section (`\firstsectionis`) sont définies. La commande sur les chapitres n'a de sens que dans les documents avec des chapitres (classes **report** et **book**) et la commande sur les sections n'a de sens que dans les classes sans chapitres (**article**).

6.15.3 Mise en forme des mini-tables

\mtcsetfont{⟨*table*⟩}{⟨*niveau*⟩}{⟨*fonte*⟩}

La commande `\mtcsetfont` permet de modifier la fonte utilisée pour composer les titres dans les mini-tables. Le premier paramètre indique de quelle table on souhaite modifier la mise en forme. Sa valeur peut être le nom de n'importe quel type de table (minitoc, partlof, etc). Le deuxième paramètre indique de quel niveau de sectionnement on souhaite changer la mise en forme, par exemple section ou subsection. Ce deuxième paramètre peut également être une étoile (*) auquel cas le changement sera appliqué à tous les niveaux. Enfin le troisième paramètre indique la fonte à utiliser.

Les seuls changements qui puissent être faits au niveau global (en passant une étoile en deuxième paramètre) sont des changements de taille, ou des redéfinitions des fontes. Cette option n'est donc utile que pour changer globalement la taille de la table (à condition de ne pas préciser la taille pour chaque type d'entrée) ou pour composer la table dans une fonte tout à fait différente, par exemple en Times.

Ainsi, le code suivant

```
\mtcsetfont{secttoc}{*}{\renewcommand\rmdefault{txr}\small}
\mtcsetfont{secttoc}{subsection}{\bfseries}
\mtcsetfont{secttoc}{subsubsection}{\itshape}
```

dit que les tables des matières au niveau des sections seront composées en Times et en petit, que les entrées pour les sous-sections seront en gras, et que les entrées pour les sous-sous-sections seront en italique.

L'exemple de la figure 6.35 page suivante présente une utilisation de ce code. Pour bien voir la différence, la table au niveau section est présentée deux fois, une première fois en Times, et une seconde en Computer Modern. Ça illustre également le fait qu'on peut changer la mise en forme des tables en cours de document, même si c'est en règle générale plutôt une mauvaise idée, a priori.

\mtcsettitlefont{⟨*table*⟩}{⟨*fonte*⟩}

La commande `\mtcsettitlefont` permet de changer la fonte du titre de la table, selon la même logique que la commande `\mtcsetfont` que tu vient de voir. Le titre de la table lui-même est défini par l'option de langue que tu as utilisée. Si pour une raison ou pour une autre tu veux changer ce titre, c'est assez simple.

\mtcsettitle{⟨*table*⟩}{⟨*titre*⟩}

La commande `\mtcsettitle` permet en effet de redéfinir le titre d'un type de table. Cependant, le titre en question sera écrasé à chaque changement de langue. Deux solutions à ce problème : soit redéfinir les titres après chaque changement de langue (en espèrant qu'il n'y en ait pas trop), et en particulier après le `\begin{document}` ; soit faire en sorte que `babel` se charge de cette redéfinition à chaque changement de langue. En particulier, pour le français, ça ressemblerait à ça :

```
\addto\extrasfrench{%
  \mtcsettitle{secttoc}{Contenu}%
}
```

L'exemple de la figure 6.36 page 374 présente une utilisation de ces deux commandes.

```
* \usepackage{minitoc}
* \setlength{\stcindent}{0pt}
* \mtcsetfont{secttoc}{*}{%
*     \renewcommand\rmdefault{txr}%
*     \small
* }
* \mtcsetfont{secttoc}
*     {subsection}{\bfseries}
* \mtcsetfont{secttoc}
*     {subsubsection}{\itshape}
* \setcounter{secttocdepth}{3}
  \dosecttoc
  \mtcselectlanguage{french}
  \setcounter{tocdepth}{1}
  \faketableofcontents
  \section{Une section}
  \secttoc
  \mtcsetfont{secttoc}{*}{\small}
  \secttoc
  \subsection{Ss-section}
  \subsubsection{Ss-ss-section}
  \subsection{Ss-section}
  \subsection{Ss-section}
  \texte
  \section{Autre}
```

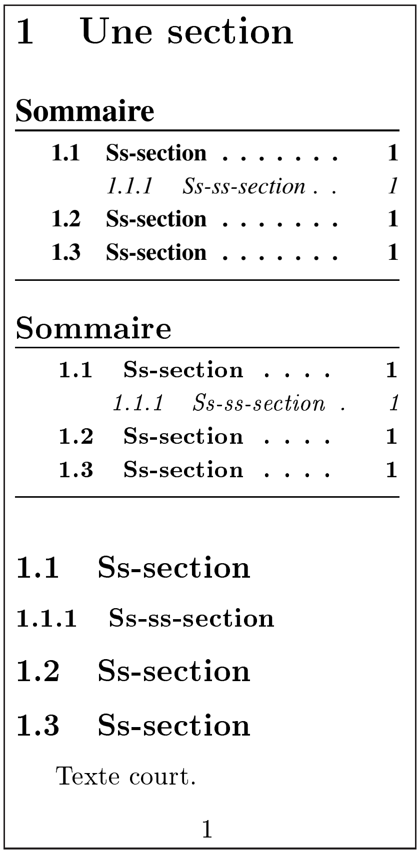


FIG. 6.35: Exemple d'utilisation de la commande `\mtcsetfont` du package `minitoc`

`\mtcsetformat`{⟨*table*⟩}{⟨*paramètre*⟩}{⟨*valeur*⟩}

La commande `\mtcsetformat` permet de modifier la mise en forme des entrées dans une minitable. Le premier paramètre indique quelle table est modifiée. Le deuxième paramètre indique quel paramètre de la mise en forme on modifie, et le dernier indique quelle valeur il faut prendre.

Quand le deuxième paramètre est dotinterval, la valeur est exprimée en *math units* (`mu`[177]) et est la distance entre les points de conduite pour ce type de table. Si tu mets une valeur suffisament grande (plusieurs milliers), alors la distance à laisser entre deux points est tellement grande que LaTeX n'aura même pas la place de mettre le premier de ces points : c'est un moyen bizarre (mais le seul officiel) de supprimer les points de conduite. Pour cette option, on ne précise pas l'unité de la valeur, ainsi on indique `12.5` pour signifier `12.5mu`.

Quand le deuxième paramètre est pagenumwidth, la valeur indique la largeur du numéro de page. Le règlage par défaut de LaTeX prévoit des numéros de page sur trois chiffres, ce qui est généralement suffisant, sauf bien entendu pour un très gros livre, ou pour des numéros de page en chiffre romains.

Enfin le deuxième paramètre peut également être tocrightmargin auquel cas il correspond

[177] Pour mémoire : il y a 18 `mu` dans un `em`, et `1em` représente le corps de fonte en cours, c'est-à-dire à peu près la largeur de la lettre m minuscule.

```
* \usepackage{minitoc}
* \setlength{\stcindent}{0pt}
* \setcounter{secttocdepth}{3}
* \mtcsettitlefont
*     {secttoc}{\sffamily\bfseries}
* \mtcselectlanguage{french}
* \addto\extrasfrench{%
*   \mtcsettitle{secttoc}{Contenu}%
* }
  \dosecttoc
  \setcounter{tocdepth}{1}
  \faketableofcontents
  \section{Une section}
  \secttoc
```

1 Une section

Contenu

1.1	**Ss-section**	**1**
1.1.1	Ss-ss-section . .	1
1.2	**Ss-section**	**1**
1.3	**Ss-section**	**1**

1.1 Ss-section

1.1.1 Ss-ss-section

1.2 Ss-section

1.3 Ss-section

Texte court.

2 Autre

1

FIG. 6.36: Exemple d'utilisation de la commande `\mtcsettitle` du package `minitoc`

à la marge réservée à droite de la table. C'est dans cette marge qu'est le numéro de page : tocrightmargin devrait donc être plus grand que pagenumwidth.

La figure 6.37 page suivante présente un exemple d'utilisation de la commande `\mtcsetformat` qui réduit la marge de droite (l'exemple ayant très peu de pages, une place réduite pour le numéro de page conviendra tout à fait) et ressère les points de conduite.

Les mini-tables sont mises en forme dans un environnement qui augmente les marges à gauche et à droite. Cette marge est modifiable est changeant la valeur d'une longueur. Pour les tables au niveau partie, c'est la longueur `\ptcindent` ; pour les tables au niveau chapitre c'est `\mtcindent` ; et pour les tables au niveau section c'est `\stcindent`.

Tu peux modifier ces valeur, par exemple en faisant appel à :

```
\setlength{\stcindent}{0pt}
```

J'ai utilisé ce réglage dans tous mes exemples, la très faible largeur de ces exemples ne laissant pas assez de place pour une telle indentation.

Le seul point de mise en forme que le package `minitoc` ne permet pas de modifier est la place réservée au numéro de section.

```
* \usepackage{minitoc}
* \setcounter{secttocdepth}{3}
* \mtcselectlanguage{french}
* \setlength{\stcindent}{0em}
* \mtcsetformat{secttoc}
*         {pagenumwidth}{1em}
* \mtcsetformat{secttoc}
*         {tocrightmargin}{1em}
* \mtcsetformat{secttoc}
*         {dotinterval}{1}
  \dosecttoc
  \setcounter{tocdepth}{1}
  \faketableofcontents
  \section{Une section}
  \secttoc
```

1 Une section

Sommaire

1.1 Ss-section

1.1.1 Ss-ss-section

1.2 Ss-section

1.3 Ss-section

Texte court.

2 Autre

1

FIG. 6.37: Exemple d'utilisation de la commande `\mtcsetformat` du package `minitoc`

`\minitoc[⟨alignement⟩]`

`\dominitoc[⟨alignement⟩]`

Par défaut, le titre des mini-tables est à gauche, comme tu peux le voir sur l'exemple de la figure 6.37. Tu peux changer cet alignement soit pour toutes les mini-tables d'un même type, soit pour une mini-table choisie, en passant un argument optionnel à la commande de préparation (changement pour toutes les tables) soit à la commande de la mini-table en question (changement pour une seule table). Cet argument optionnel peut valoir `l` pour que le titre soit à gauche (choix par défaut), ou `c` pour qu'il soit centré, ou `r` pour qu'il soit à droite, ou encore `n` (ou `e` qui est équivalent) pour qu'il n'y ait pas de titre.

Ainsi, si tu veux que tous les titres des `\minilof` soient à droite, sauf celle du premier chapitre que tu veux avec un titre centré, tu peux faire comme ça :

```
\dominilof[r]
...
\chapter{...}
\minilof[c]
...
\chapter{...}
```

Mini-table	Avant Valeur par défaut	Après Valeur par défaut	Style de page Valeur par défaut
`\parttoc`	`\beforeparttoc`	`\afterparttoc`	`\thispageparttocstyle`
`\partlof`	`\beforepartlof`	`\afterpartlof`	`\thispagepartlofstyle`
`\partlot`	`\beforepartlot`	`\afterpartlot`	`\thispagepartlotstyle`
	`\cleardoublepage`	`\cleardoublepage`	`\thispagestyle{empty}`

TAB. 6.11: Mise en forme des tables de partie par `minitoc`

```
\minilof
...
```

Les neuf commandes préparation, et les neuf commandes d'exécution, des mini-tables peuvent prendre cet argument optionnel.

`\mtcsetrules`{⟨*table*⟩}{⟨*valeur*⟩}

Par défaut, la majorité des tables est encadrée de deux filets horizontaux (un avant et un après) : en fait toutes les mini-tables sauf les tables de parties (`\parttoc`, `\partlof` et `\partlot`) dans les documents ayant un niveau chapitre (classes **report** et **book**). Tu peux indiquer sur quelles tables tu veux ou te ne veux pas, de filets en utilisant la commande `\mtcsetrules`. Elle prend en premier paramètre le nom du type de table que tu veux modifier, et en second paramètre une valeur qui indique si tu veux ou pas des filets avant et après cette table. Le second paramètre peut valoir yes, YES, on, ON, y, Y, true, TRUE, t, T ou 1 pour indiquer que tu veux des filets ; ou no, NO, off, OFF, n, N, false, FALSE, f, F ou 0 pour indiquer que tu ne veux pas de filets.

`\mtcsetpagenumbers`{⟨*table*⟩}{⟨*valeur*⟩}

En temps normal, les mini-tables indiquent les titres et les numéros de page, comme pour les tables normales. Certains préfèrent avoir simplement un plan qui indique de quoi parlera le chapitre (ou la section), pour ça il vaut mieux supprimer les numéros de page. La commande `\mtcsetpagenumbers` te permet d'indiquer, pour chaque type de table, si tu veux ou pas avoir des numéros de page. Le premier paramètre indique quel type de table tu veux modifier, et le second si tu veux les numéros de page. Le second paramètre peut valoir yes (ou l'une des valeurs alternatives décrites pour `\mtcsetrules`) pour dire que tu veux les numéros de page, ou no (ou une des valeurs alternatives) pour indiquer que tu ne veux pas de numéros de page. Si tu supprimes les numéros de page, il vaut mieux supprimer également les points de conduite avec la commande `\mtcsetformat`.

Ainsi, si tu veux supprimer les numéros de page pour les secttoc, et supprimer les filets, tu procéderas comme ça :

```
\mtcsetrules{secttoc}{no}
\mtcsetpagenumbers{secttoc}{no}
\mtcsetformat{secttoc}{dotinterval}{1000}
```

L'exemple de la figure 6.38 page suivante présente le résultat de ces commandes.

Les mini-tables au niveau des parties (`\parttoc`, etc.), dans les classes avec la commande `\chapter`, sont normalement sur des pages à part. Si le document est recto-verso, ce sont des pages recto. Ces pages ne sont normalement pas numérotées, et ne portent ni en-tête ni pied de page. Ce comportement vient de la définition par défaut des commandes du

```
* \usepackage{minitoc}
* \setcounter{secttocdepth}{3}
* \mtcselectlanguage{french}
* \setlength{\stcindent}{0em}
* \mtcsetformat{secttoc}
*         {dotinterval}{1000}
* \mtcsetpagenumbers{secttoc}{no}
* \mtcsetrules{secttoc}{no}
  \dosecttoc
  \setcounter{tocdepth}{1}
  \faketableofcontents
  \section{Une section}
  \secttoc
```

1 Une section

Sommaire

1.1 Ss-section

1.1.1 Ss-ss-section

1.2 Ss-section

1.3 Ss-section

Texte court qui se répète encore et toujours pour remplir. Texte court qui se répète encore et toujours pour remplir. Texte court qui se répète encore et toujours pour remplir.

1

FIG. 6.38: Exemple d'utilisation des commandes `\mtcsetrules` et `\mtcsetpagenumbers` du package `minitoc`

---

tableau 6.11 page ci-contre. Si ce n'est pas le comportement que tu souhaites, par exemple pour avoir la table au dos de la page de la partie, tu peux modifier ces définitions. Notes que, par défaut, seule la *première* page de ces tables est dans le style empty. Tu peux bien entendu fixer le style de la première page en utilisant `\thispagestyle` et le style des pages suivantes avec `\pagestyle`, mais il faudra penser à remettre le style normal après la mini-table. Voila ce que ça pourrait donner :

```
\renewcommand\thispageparttocstyle{%
  \thispagestyle{premiere-parttoc}%
  \pagestyle{autre-parttoc}%
}
...
\part{...}
\parttoc
\pagestyle{normal}
```

Pour ce qui est de (re-)définir des styles de page, reportes toi à la section correspondante (au titre 4 page 196).

6.15.4 Le package `mtcoff`

Si tu as composé tout ton document pour qu'il y ait des mini-tables un peu partout, et que tu souhaites, pour une raison ou une autre, obtenir un document *sans* ces mini-tables, plutot que d'aller mettre toutes les commandes concernées en commentaire, il te suffit de charger le package `mtcoff`, sans options, à la place de `minitoc`.

6.16 Table des matières : le package `titletoc`

Ce package, que nous devons à Javier Bezos [17], permet de modifier la mise en forme de la table des matières de manière assez poussée. Il permet également de faire des tables partielles de manière assez intéressante. Il est livré et documenté avec le package `titlesec`, du même auteur, certains détails fonctionnent mieux quand on utilise les deux en même temps.

Le package `titletoc` fonctionne sur un modèle assez similaire à celui de `titlesec` (pas étonnant) : des commandes plus ou moins complexe, et prennant un grand nombre de paramètres, dont certains optionnels, permettent de définir la mise en forme de la table des matières.

Le package considère qu'il y a deux grandes manières de présenter les titres dans une table des matières : la méthode « hors-ligne » qui est celle que tu connais, et qui consiste à avoir un paragraphe[178] pour chaque entrée ; et la méthode « en-ligne » qui consiste à regrouper toutes les entrées d'un même niveau (ou de plusieurs niveaux d'ailleurs) en un paragraphe, en séparant les différentes entrées par un signe de ponctuation.

6.16.1 Les titres « hors-ligne »

La forme la plus courante de ces titres est celle qu'on croise dans la présentation par défaut de la table des matières de LaTeX par défaut. La commande `\dottedcontents` permet d'obtenir un résultat similaire à ce qu'on obtient avec la commande `\@dottedtocline` de LaTeX, mais selon un procédé très différent : quand on utilise la commande de LaTeX, on doit redéfinir une commande interne[179] cachée derrière le rendu de la table des matières (la commande `\l@section` si on traite des titres de `\section`). Dans l'approche du package, cette manipulation est considérée comme une étape de configuration : la commande `\dottedcontents` se chargera de redéfinir ce dont elle a besoin sans que tu ais à te préoccuper de la mécanique interne.

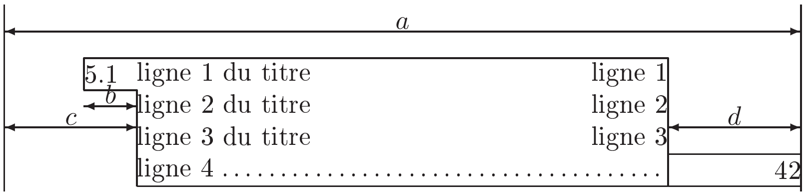
```
\dottedcontents{⟨section⟩}[⟨gauche⟩]{⟨avant⟩}{⟨largeur numéro⟩}
                {⟨leurgeur point⟩}
```

Le premier paramètre, {⟨*section*⟩} est le *nom* du niveau de titre dont tu définis la mise en forme (le nom, pas la commande, donc `section`, pas `\section`). Le deuxième paramètre, [⟨*gauche*⟩][180] indique une marge supplémentaire à appliquer sur le côté gauche ; ce paramètre est une longueur. C'est le paramètre qui faira que les titres de sous-section sont plus à droite

[178] En regardant trop rapidement, tu pourrais croire que c'est une ligne par entrée de la table des matières, mais ce n'est pas vrai, puisque certaines entrées occupent plusieurs lignes. C'est bien un paragraphe par entrée, et avec des règles d'alignement qui ne sont pas simples.

[179] Cette approche pénible et compliquée vient du fait que quand LaTeX a été mis au point, au début des années 80, il n'était pas prévu de pouvoir personnaliser la mise en forme à ce point-là sans avoir à tout refaire. L'approche qui consiste à rendre tout ça très paramètrable est beaucoup plus récente. Un document utilisant de nombreux packages comme `titletoc`, sur un ordinateur du début des années 80, donnerait quelque chose de très long à compiler, et demanderait probablement d'avoir une version sur mesure de TeX pour qu'il ait assez de mémoire pour tout traiter. Bien entendu, ces limitations ne sont plus tellement d'actualité.

[180] Bien que ce paramètre soit entre crochets, il n'est absolument pas optionnel. Il faut le donner.



- *a* largeur du texte (à l'intérieur des marges habituelles), `\linewidth` ;
- *b* largeur du numéro de la section, paramètre {⟨*largeur numéro*⟩} de la commande `\dottedcontents` ;
- *c* marge gauche supplémentaire, paramètre `[`⟨*gauche*⟩`]` de la commande `\dottedcontents` ;
- *d* marge droite supplémentaire, paramètre {⟨*droite*⟩} de la commande `\contentsmargin`, ce paramètre est global à la table des matières et non spécifique à un type d'entrée, cette marge sert pour mettre le numéro de page.

FIG. 6.39: Forme générale du résultat de `\dottedcontents`

---

que les titres de section. Le troisième paramètre, {⟨*avant*⟩}, peut contenir n'importe quel code LaTeX et sera exécuté juste avant le paragraphe, tu peux y placer de quoi ajouter du blanc vertical, ou des commandes de changement de fonte. Le quatrième paramètre, {⟨*largeur numéro*⟩}, est la largeur du numéro de la section qui dépassera sur le côté gauche du titre. Enfin, le cinquième et dernier paramètre indique l'espacement qu'il doit y avoir entre les points de conduite qui sont entre le titre et le numéro de page, le bon goût typographique indique qu'il vaut mieux que ce soit le même pour tous les types de titre.

La figure 6.39 présente graphiquement les différents paramètres de la commande `\dottedcontents`.

`\contentsmargin`{⟨*droite*⟩}

La commande `\contentsmargin` sert à préciser la marge supplémentaire à ajouter à droite. Cette marge est supposée être la même, a priori, pour tous les niveaux de titre de manière à ce que les numéros de page soient alignés. Tu peux modifier cette marge pour un niveau de titre particulier en placant un appel à `\contentsmargin` dans le paramètre {⟨*avant*⟩} de la commande `\dottedcontents`.

Le paramètrage par défaut des classes de LaTeX ressemble à ça :

```
\contentsmargin{2.55em}
\dottedcontents{section}[1.5em]{}{2.3em}{1pc}
\dottedcontents{subsection}[3.8em]{}{3.2em}{1pc}
```

Quelques éléments à relever : la place réservée au numéro de section (`2.3em` dans un cas et `3.2em` dans l'autre) est plus grand pour les sous-sections parce que le numéro de sous-section est normalement plus long que le numéro de section (forcément, il commence par le numéro de section) ; l'espacement entre les points de conduite est d'un pica (soit 12 points) et est le même pour tous les titres ; enfin la marge gauche des sous-sections est très exactement la marge gauche des sections plus la largeur du numéro de section, ainsi le numéro de la sous-section commencera au même niveau que le texte du titre de la section.

La figure 6.40 page suivante te présente un exemple d'utilisation de `\dottedcontents`, ainsi que l'utilisation de `\contentsmargin` dans le paramètre {⟨*avant*⟩} pour réduire la largeur des titres de niveau inférieur. Cet exemple montre certaines limitations de la commande `\dottedcontents` : si tu veux une fonte différente pour le numéro de section, pour le titre, et pour le numéro de page, ce n'est pas possible ; si tu veux que le numéro de page soit décalé

```
* \usepackage{titletoc}
* \contentsmargin{1em}
* \dottedcontents{section}%
*   [1.5em]{%
*     \addvspace{3pt}}
*   {1.5em}{1pc}
* \dottedcontents{subsection}%
*   [3.5em]{%
*     \contentsmargin{3em}%
*     \itshape}
*   {2em}{1pc}
  \tableofcontents
  % contenu du document non montre.
```

Table des matières

1 Un section dont le titre est très long pour voir ce que ça fait 1

1.1 Premier truc : une sous-section avec un titre très long pour voir l'alignement des marges 1

1.2 Deuxième truc . 2

1.3 Troisième . . . 3

1.4 Encore un. . . 3

2 Section suivante 4

2.1 Sous-section . . 4

2.2 Une autre . . . 4

3 Encore une section . . . 4

3.1 Encore 4

3.2 Ici aussi . . . 4

1

Quelques remarques sur les choix faits dans cet exemple :
- une marge supplémentaire à gauche pour les sections de `[1.5em]` ;
- dans cette marge, le numéro de section prend précisément `{1.5em}` : le numéro est donc le long de la marge normale, le texte du titre étant renfoncé ;
- une marge supplémentaire à gauche pour les sous-sections de `[3.5em]` ;
- dans cette marge, le numéro de section prend `{2em}`, le numéro commence donc à `1.5em` de la marge de gauche, soit juste sous le titre de la section ;
- la marge droite est augmentée de `{1em}` en temps normal, sauf pour les sous-sections où elle est renfoncée de `{3em}`, ça ne change pas la position du numéro de page (il reste collé tout à droite) mais ça réduit la place disponible pour le titre et pour les points de conduite ;
- les titres de sous-section sont en italique, *tout* le titre est en italique, y compris le numéro de section et le numéro de page.

FIG. 6.40: Exemple d'utilisation de la commande `\dottedcontents` du package `titletoc`

---

pour les sous-sections, ce n'est pas possible, etc.

`\titlecontents`{⟨*section*⟩}`[`⟨*gauche*⟩`]`{⟨*avant*⟩}{⟨*avant numéroté*⟩} {⟨*avant non-numéroté*⟩}{⟨*remplir et page*⟩}`[`⟨*après*⟩`]`

La commande `\titlecontents` offre une interface plus complexe, et du coup plus complète que celle de `\dottedcontents` ; elle permet par exemple de contrôler précisément les fontes et le placement de chacun des éléments. Bien entendu, si la mise en forme générique de la commande simple te suffit, il vaut mieux utiliser la commande simple.

Le premier paramètre, {⟨*section*⟩}, est le nom du type de titre concerné. Le deuxième,

[⟨*gauche*⟩] n'est toujours pas optionnel et indique la marge supplémentaire à gauche ; le troisième, {⟨*avant*⟩}, indique un formatage qui a lieu avant le début du titre, il peut contenir par exemple un blanc vertical ou une modification de la marge de droite ; le quatrième paramètre, {⟨*avant numéroté*⟩}, est utilisé en mode horizontal, juste devant le titre s'il est numéroté, si la dernière commande de ce paramètre prend un argument, ce sera le titre de la section ; le cinquième paramètre, {⟨*avant non-numéroté*⟩}, joue exactement le même rôle pour les sections qui ne sont pas numérotées ; le sixième paramètre, {⟨*remplir et page*⟩}, est utilisé juste après le titre pour produire les eventuels points de conduite et le numéro de page ; enfin le dernier paramètre est ajouté après l'entrée (en mode vertical) par exemple pour ajouter du blanc vertical après.

Le quatrième paramètre, {⟨*avant numéroté*⟩} fera normalement appel à `\thecontentlabel` pour afficher le numéro de section. De la même manière le sixième paramètre, {⟨*remplir et page*⟩} fera probablement appel à une commande comme `\titlerule`[181] pour remplir la dernière ligne du titre, et à `\thecontentspage` pour afficher le numéro de page.

La figure 6.41 page suivante présente un exemple d'utilisation de la commande `\titlecontents` pour produire un résultat très similaire à l'exemple précédent, mais où cette fois-ci seul le titre est mis en italique (pas les numéros de sous-section ni les numéros de page), et où le numéro de page des sous-section est renfoncé par rapport à celui des sections. L'usage de la commande `\makebox` est expliqué dans le passage sur les boîtes (au titre 5 page 299), le fonctionnement des commandes `\rlap` et `\llap`, lui, est abordé dans le chapitre sur la programmation (chapitre neuvième page 571).

La construction pour le numéro de section est tellement classique que le package prévoit une commande exprès pour ça, la commande `\contentslabel`.

`\contentslabel`[⟨*numéro*⟩]{⟨*largeur*⟩}

Le premier paramètre, optionnel, vaut par défaut `\thecontentslabel`[182] ; le second paramètre, obligatoire, indique la largeur à utiliser pour composer le numéro de section (équivalent du paramètre {⟨*largeur numéro*⟩} de la commande `\dottedcontents`, qui est illustré à la figure 6.39 page 379). La commande utilise `\hfil` pour « pousser » le numéro de section d'un côté ou de l'autre de la boîte (fonction des options de chargement du package, j'y reviendrai). Si tu veux pousser le numéro en question de l'autre côté, il suffit de mettre un `\hfill` au bon endroit.

Par exemple, par défaut le package produit quelque chose comme ça : 4.2, si tu préfères ça : 4.2 –; il faut placer la numéro sur la droite (il est à gauche par défaut) et le faire suivre par un tiret (ce qui n'est pas prévu par le package). Si tu veux (drôle d'idée, à ne pas appliquer dans la vraie vie) que cette mise en forme ne soit utilisée que pour *ce* niveau de titre, une solution est d'utiliser dans l'argument {⟨*avant numéroté*⟩} de `\titlecontents` cette contruction-là :

```
\contentslabel[\hfill\thecontentslabel\enspace--]{3em}
```

La figure 6.42 page 383 reprend cette mise en forme.

De même, la construction pour le numéro de page est tellement classique qu'elle est reprise dans une commande spéciale : `\contentspage`.

---

[181] Cette commande est décrite dans le passage sur `titlesec` au paragraphe 6.11 page 341.

[182] Bon, en vrai, il vaut quelque chose de plus compliqué. Déjà, le numéro de la section est suivi de la commande `\ttl@idot` qui produira un point si tu as utilisé l'option `dotinlabel` lors du chargement du package et qui ne produira rien sinon : c'est comme ça que l'option produit le point après les numéros de section dans la table des matières. Ensuite, si tu as utilisé l'option `rightlabels` lors du chargement, il y a la commande `\enspace` qui sert à produire un petit blanc : en effet l'option en question place le numéro de section sur la *droite* de sa boîte ce qui l'amènerait à toucher le titre de la section ; le blanc évite cette collision.

```
\usepackage{titletoc}
\contentsmargin{1em}
\dottedcontents{section}%
  [1.5em]{%
    \addvspace{3pt}}
  {1.5em}{1pc}
\titlecontents{subsection}%
  [3.5em]{%
    \contentsmargin{3em}}
  {\llap{\makebox[2em][l]{%
    \thecontentslabel}}%
   \textit}
  {\textit}
  {\titlerule*[1pc]{.}%
   \rlap{\makebox[1em][r]{%
     \thecontentspage}}}
\tableofcontents
% contenu du document non montre.
```

Table des matières

1

Quelques remarques sur les choix faits dans cet exemple :
- les sections sont présentées comme dans l'exemple précédent ;
- les sous-sections sont présentées de manière différente, la marge de droite en renfoncée de `2em` de plus que celle des sections, soit `{3em}` en tout ;
- le numéro de la sous-section (`\thecontentslabel`) est mis à gauche (`[l]`) dans une boîte de `[2em]` de large, et cette boîte dépasse à gauche dans la marge (`\llap`) ;
- le titre est mis en italique ;
- les points de conduite sont également espacés d'un pica, ils se retrouvent alignés avec les points de conduite des sections ;
- le numéro de page est à droite (`[r]`) dans une boîte de `[1em]` de large, cette boîte dépasse à droite dans la marge (`\rlap`).

Le résultat est que le numéro de page des sous-sections est renfoncé par rapport à celui des sections, et que *seul le titre* est en italique : le numéro de sous-section et le numéro de page ne le sont pas.

FIG. 6.41: Exemple d'utilisation de la commande `\titlecontents` du package `titletoc`

`\contentspage[⟨page⟩]`

Cette commande ne prend qu'un seule paramètre, optionnel, qui est par défaut `\thecontentspage`. Elle place son argument optionnel (donc, par défaut, le numéro de page) calé à droite dans une boîte qui occupe la marge supplémentaire à droite. Il n'y a pas de paramètre prévu pour lui dire qu'elle fasse une autre largeur que celle de la marge supplémentaire. Par contre, on peut tricher en mettant du blanc dans le contenu. Ainsi, pour reproduire l'effet de l'exemple précédent, on peut remplacer la construction complexe à base de `\rlap` et de `\makebox` par

```
* \usepackage{titletoc}
* \contentsmargin{1em}
* \dottedcontents{section}%
*   [1.5em]{%
*     \addvspace{3pt}}
*   {1.5em}{1pc}
* \titlecontents{subsection}%
*   [4.5em]{%
*     \contentsmargin{3em}}
*   {\contentslabel[%
*     \hfill\thecontentslabel
*     ~--]{3em}\textit}
*   {\textit}
*   {\titlerule*[1pc]{.}%
*    \rlap{\makebox[1em][r]{%
*      \thecontentspage}}}
  \tableofcontents
  % contenu du document non montre.
```

Table des matières

1

Quelques remarques sur les choix faits dans cet exemple :
- la marge de gauche des sous-sections a été renforcée pour prévoir de la place pour la forme bizarre des numéros de sous-section ;
- la marge de droite ne bouge pas, il y a donc moins de place pour le titre, ça devient vraiment très laid.

FIG. 6.42: Exemple d'utilisation de la commande `\contentslabel` du package `titletoc`

---

un appel comme :

`\contentspage[\thecontentspage\hspace{2em}]`

Pratique également pour modifier la fonte du numéro de page (pour utiliser des nombres elzéviriens, par exemple).

6.16.2 Les titres « en-ligne »

Cette forme est très appréciée dans certains ouvrages. Elle offre l'avantage de fournir une table des matières plus compacte dans certains cas. L'idée est de regrouper les entrées d'un même niveau (et des niveaux en-dessous) en un même paragraphe, par exemple de regrouper les titres des sous-sections en un paragraphe sous le titre de la section qui les contient. Le contraire marche (mettre tous les titres d'un niveau en un paragraphe, qui s'interromp pour mettre les titres du niveau d'en-dessous en hors-ligne), mais le résultat est très surprenant et ne semble pas intéressant.

```
\titlecontents*{⟨section⟩}[⟨gauche⟩]{⟨avant⟩}{⟨avant numéroté⟩}
               {⟨avant non-numéroté⟩}{⟨remplir et page⟩}
               [⟨début⟩] [⟨séparateur⟩][⟨fin⟩]
```

La commande `\titlecontents*` fonctionne comme `\titlecontents` pour la majorité des paramètres, sauf que, a priori, elle ne fait pas un paragraphe par entrée de la table des matières.

Pour les premiers paramètres, on n'est pas dépaysé : {⟨*section*⟩} est le nom du niveau de titre, [⟨*gauche*⟩] est la marge supplémentaire à gauche (celle qui s'appliquera pour tout le paragraphe), {⟨*avant*⟩} sera placé avant le paragraphe, {⟨*avant numéroté*⟩} sera mis (en mode horizontal) juste avant le titre s'il est numéroté, et {⟨*avant non-numéroté*⟩} s'il ne l'est pas ; {⟨*remplir et page*⟩} sera utilisé pour placer le numéro de page.

Ensuite viennent les trois paramètres optionnels. Le premier, [⟨*début*⟩] est mis au début du paragraphe, après {⟨*avant*⟩}, mais avant le {⟨*avant (non-)numéroté*⟩} du titre en question ; il n'est utilisé que si le titre fait suite à un titre de niveau plus élevé ; en temps normal ce paramètre n'est pas utilisé. Le deuxième, [⟨*séparateur*⟩], est inséré entre deux titres de ce niveau là. Le troisième est inséré à la fin du paragraphe.

Ainsi, si notre paragraphe contient 3 titres, il contiendra, dans cet ordre :

1. {⟨*avant*⟩}
2. {⟨*avant (non-)numéroté sect 1*⟩}
3. titre section 1
4. {⟨*remplir et page sect 1*⟩}
5. [⟨*séparateur*⟩]
6. {⟨*avant (non-)numéroté sect 2*⟩}
7. titre section 2
8. {⟨*remplir et page sect 2*⟩}
9. [⟨*séparateur*⟩]
10. {⟨*avant (non-)numéroté sect 3*⟩}
11. titre section 3
12. {⟨*remplir et page sect 3*⟩}
13. [⟨*fin*⟩]

Notes bien que le paramètre optionnel [⟨*début*⟩] *n'est pas utilisé*. Ce paramètre n'est utilisé que dans un cas très précis : quand il y a imbrication de deux niveaux de titre.

Imaginons que nous ayons 3 titres, et que le deuxième contienne 2 sous-titres, voilà la liste des éléments auxquels on arrive :

1. {⟨*avant*⟩}
2. {⟨*avant (non-)numéroté sect 1*⟩}
3. titre section 1
4. {⟨*remplir et page sect 1*⟩}
5. [⟨*séparateur*⟩]
6. {⟨*avant (non-)numéroté sect 2*⟩}
7. titre section 2
8. {⟨*remplir et page sect 2*⟩}
   (a) {⟨*avant ss-titre*⟩}
   (b) [⟨*début ss-titre*⟩]
   (c) {⟨*avant (non-)numéroté ss-sect 2.1*⟩}
   (d) titre sous-section 2.1
   (e) {⟨*remplir et page ss-sect 2.1*⟩}
   (f) [⟨*séparateur ss-titre*⟩]
   (g) {⟨*avant (non-)numéroté ss-sect 2.2*⟩}
   (h) titre sous-section 2.2

Option	Description
leftlabels	Modifie la commande `\contentslabel` pour que le numéro de section soit calé à gauche (choix par défaut).
rightlabels	Cale le numéro de section à droite.
dotinlabels	Ajoute un point après le numéro de section.
nodotinlabels	N'ajoute pas de point. Choix par défaut.

TAB. 6.12: Options du package `titletoc` modifiant le fonctionnement de la commande `\contentslabel`

(i) {⟨*remplir et page ss-sect 2.2*⟩}
(j) [⟨*fin ss-titre*⟩]

9. [⟨*séparateur*⟩]
10. {⟨*avant (non-)numéroté sect 3*⟩}
11. titre section 3
12. {⟨*remplir et page sect 3*⟩}
13. [⟨*fin*⟩]

Voyons tout d'abord un exemple d'utilisation simple : mettre les sous-sections comme un paragraphe sous le titre de la section qui les contient. On mettra les titres de sous-section en italique, les numéros de sous-section en gras (pour que ca fasse un point d'arrêt clair entre deux titres), un point-virgule pour séparer les titres les uns des autres, et un point pour terminer le paragraphe. Ça se demande comme ça[183] :

```
\titlecontents*{subsection}[1.5em]{}
  {\textbf{\thecontentslabel}~\textit}
  {\textit}{, \thecontentspage}
  [; ][.]
```

Cette mise en forme est reprise dans l'exemple de la figure 6.43 page suivante.

Sur la même base, on peut ajouter les titres de sous-sous-sections dans le paragraphe, par exemple en les mettant entre parenthèses. Ça pourrait donner quelque chose comme ça :

```
\titlecontents*{subsubsection}[1.5em]{}
  {\thecontentslabel~}{}{}
  [ (][; ][)]
```

Cettre mise en forme est celle de l'exemple de la figure 6.44 page suivante.

6.16.3 Options du package

Le package `titletoc` ne prend que très peu d'options, et elles ne modifient que la commande `\contentslabel`. Ces options sont présentées dans le tableau 6.12.

`\contentsuse`{⟨*nom du flottant*⟩}{⟨*extension du fichier*⟩}

Le package `titletoc` traite, spontanément, la table des matières, la liste des figures, et la liste des tableaux. Comme on peut créer d'autre types de flottants, et donc d'autres tables,

[183]Le commentaire de la figure 6.43 page suivante explique en quoi il faut ruser un peu plus que cette simple construction pour obtenir le bon résultat. Cette construction trop simple produira un espacement faux pour le point-virgule.

```
\usepackage{titletoc}
\contentsmargin{1em}
\dottedcontents{section}%
  [1.5em]{%
    \addvspace{3pt}\bfseries}
  {1.5em}{1pc}
\titlecontents*{subsection}%
  [1.5em]{}
  {\textbf{\thecontentslabel}%
   ~\textit}
  {\textit}
  {, \thecontentspage}
  [\pv][.]
\setcounter{tocdepth}{2}
\newcommand\pv{; }
\tableofcontents
% contenu du document non montre.
```

Table des matières

1 Un section dont le titre est très long pour voir ce que ça fait 1
1.1 *Premier truc, un titre très long*, 2 ; **1.2** *Deuxième truc*, 3 ; **1.3** *Troisième*, 5 ; **1.4** *Encore un*, 5.

2 Section suivante . . . 7
2.1 *Sous-section*, 7 ;

1

À noter sur cet exemple : la construction bizarre, qui consiste à utiliser une commande, `\pv` à la place du point-virgule directement dans la commande permet de résoudre un subtil problème d'espacement pour le français.

En français, il doit y avoir un petit blanc devant le point-virgule, comme devant toutes les ponctuations doubles. Le package `babel` se charge d'ajouter tout seul cet espacement. Mais il ne s'en charge qu'à partir du `\begin{document}`. Ainsi, si j'avais directement mis un point-virgule dans l'appel à `\titlecontents*`, il aurait été négligé par `babel`. Comme la commande `\pv` est définie *après* le `\begin{document}`, le point-virgule qu'elle contient sera convenablement traité.

FIG. 6.43: Exemple d'utilisation de la commande `\titlecontents*` du package `titletoc`

```
\usepackage{titletoc}
\contentsmargin{1em}
% section et subsection comme
% l'exemple precedent
\titlecontents*{subsubsection}%
  [1.5em]{}
  {\thecontentslabel~}{}{}
  [ (][\pv][)]
\setcounter{tocdepth}{3}
\newcommand\pv{; }
\tableofcontents
% contenu du document non montre.
```

Table des matières

1 Un section dont le titre est très long pour voir ce que ça fait 1
1.1 *Premier truc, un titre très long*, 2 ; **1.2** *Deuxième truc*, 3 (1.2.1 Un détail ; 1.2.2 Un autre détail) ; **1.3** *Troisième*, 5 ; **1.4** *Encore un*, 5.

1

FIG. 6.44: Exemple d'utilisation de la commande `\titlecontents*` du package `titletoc`

---

il prévoit qu'on puisse en déclarer. Par défaut il ne déclare que les deux types de flottants classiques, mais tu peux en ajouter avec la commande `\contentsuse`. En fait, la commande indique qu'on peut trouver des entrées de type {⟨*nom du flottant*⟩} dans les fichiers dont l'extension est {⟨*extension du fichier*⟩}. À partir de ce moment-là, le package te permettra de personnaliser les entrées de type {⟨*nom du flottant*⟩} avec `\titlecontents` et autres.

6.16.4 Tables partielles

Le package `titletoc` est incompatible avec le package `minitoc`, vraiment incompatible. Son auteur a même renoncé à le rendre compatible, et a jugé plus simple d'implémenter directement les tables partielles. Il l'implémente bien pour la table des matières, mais pas pour la liste des tableaux ou la liste des figures.

Le mode de fonctionnement est nettement plus rustique que celui de `mintoc`, et demande de ta part une intervention nettement plus explicite. En contrepartie, l'avantage est qu'il y a moins de commandes à comprendre et à maitriser. Le premier point important à comprendre est que le package ne va pas chercher à comprendre tout seul où commencent et finissent les différentes tables partielles : c'est toi qui le lui dit,

`\startcontents[`⟨*type*⟩`]`

`\stopcontents[`⟨*type*⟩`]`

`\resumecontents[`⟨*type*⟩`]`

Ces trois commandes prennent en paramètre le type de table partielle qui t'intéresse, la valeur par défaut étant `default`. C'est par ce biais que tu peux gérer des tables partielles à plusieur niveaux. Si, par exemple, tu veux une table par partie qui reprenne les chapitres, et une table par chapitre qui reprenne les sections et sous-section, tu peux utiliser deux `[`⟨*type*⟩`]` différents. La première commande, `\startcontents`, commence une table partielle d'un type donné et termine la table précédente du même type s'il y en avait une. Cette commande est donc généralement suffisante : elle commencera et terminera les tables d'une manière qui semble raisonnable.

La deuxième commande, `\stopcontents` te permet de terminer ou de suspendre une table partielle, par exemple pour exclure certaines sections de la table partielle, ou pour la terminer prématurément. Son usage est relativement rare. La troisième commande, `\resumecontents`, permet de reprendre une table partielle qui aurait été suspendue *à condition qu'aucune autre table du même type n'ait été commencée.* Son usage est bien entendu encore plus rare, puisque cette commande ne peut avoir d'intérêt qu'après un appel à `\stopcontents`.

`\printcontents[`⟨*type*⟩`]{`⟨*préfixe*⟩`}{`⟨*niveau*⟩`}{`⟨*début*⟩`}`

Cette commande sert à affiche la table des matières partielle. Le premier paramètre, optionnel, indique le type de la table (utile si tu veux en avoir de plusieurs types). Le deuxième paramètre permet d'avoir une mise en forme différente entre différents type de tables (j'y reviendrai). Le troisième paramètre indique le niveau des titres les plus élevés qui vont t'intéresser dans la table en question : `-1` pour les parties dans **report** ou **book**, `0` pour les chapitres, `0` pour les parties dans un **article**, `1` pour les sections, `2` pour les sous-sections, etc. Le quatrième et dernier paramètre est un bout de code LaTeX qui sera placé en tête de la table. Tu peux l'utiliser pour fixer la profondeur de la table (en modifiant le compteur `tocdepth` qui donne le niveau des sections les plus petites que tu veux voir dans la table), pour changer les marges externes de la table (avec `\contentsmargin`, par exemple), pour placer un titre, etc.

Les entrées pour un niveau donné, disons `subsection` par exemple, dans la table des matières sont mises en forme d'apès ce que tu as indique à `\titlecontents` et consors en utilisant le type `subsection`. Si tu veux une mise en forme différente pour les tables partielles, il suffit d'utiliser le deuxième paramètre de `\printcontents` : {⟨*préfixe*⟩}. Si tu utilises par exemple `abc` comme valeur, alors ce seront les types `abcsubsection` qui seront utilisés s'ils sont définis. Par exemple :

```
% Table des matieres principale:
\titlecontents{section}...
\titlecontents{subsection}...
% Table partielle en debut de partie:
\titlecontents{p-section}...
\titlecontents{p-subsection}...
% Table partielle en debut de chapitre:
\titlecontents{c-section}...
\titlecontents{c-subsection}...
...
\part{...}
\startcontents[part]
\printcontents[part]{p-}{0}{...}
\chapter{...}
\startcontents[chapter]
\printcontents[chapter]{c-}{1}{...}
...
```

Si tu utilises le package `titlesec` pour mettre en forme les titres de ton document, tu peux alors utiliser les commandes de ce package pour inclure automatiquement les appels à `\startcontents`, voire les appels à `\printcontents` dans le code qui est exécuté après le titre. Très pratique si tu es en train d'écrire ta propre classe de documents.

La figure 6.45 page ci-contre présente un exemple de table partielle. Cet exemple montre quelques-unes des limitations de `titletoc` par rapport à `minitoc` : la mise en forme de la table partielle n'est pas faite automatiquement, c'est toi qui doit la faire à la main. Dans l'exemple c'est ce que je fais en utilisant `\subsection*` pour faire le titre de la table partielle et en ajoutant `\bigskip` en dessous pour laisser un peu de blanc. Dans un usage plus sérieux, tu aurais tout intérêt à te définir une commande simple qui fasse tout ça, par exemple, pour simuler le vocabulaire de `\minitoc`, tu pourrais faire quelque chose comme ça :

```
\newcommand\secttoc{%
  \subsection*{Sommaire}
  \hrule
  \medskip
  \printcontents[sect]{s-}{2}{}
  \medskip
  \hrule
  \bigskip
}
```

Les fonctionnalités de `titletoc` sont en effet intéressantes comme base pour construire quelque chose de puissant, mais en lui-même le package me parait trop rustique pour faire des tables des matières partielles. Et, bien entendu, il reste la limitation (complexe à contourner) sur le fait qu'il manque la liste partielle des figures ou des tableaux.

6.17 Bibliograhie et index en table des matières : le package `tocbibind`

Ce package, pas si simple que ça, que nous devons à Peter WILSON [225], permet d'ajouter automatiquement la table des matières, la liste des figures, la liste des tableaux, la biblio-

```
\usepackage{titletoc}
\contentsmargin{2em}
\dottedcontents{section}[2em]
  {\bfseries}{2em}{1pc}
\titlecontents*{subsection}%
  [2em]{}{\thecontentslabel~\textit}
  {}{, \thecontentspage}[; ][.]
\dottedcontents{s-subsection}%
  [2em]{\bfseries}{2em}{1pc}
\tableofcontents
\section{Titre}
\startcontents[sect]
\printcontents[sect]{s-}{2}
  {\subsection*{Sommaire}}
\bigskip
Voila de quoi \c{c}a parle.
\subsection{Le sujet}
Le sujet...
\subsection{Pr\'ecision}
Bla bla...
\subsection{Autre chose}
Bla bla...
\subsection{D\'etail}
Un d\'etail cependant...
\section{La suite}
\startcontents[sect]
\printcontents[sect]{s-}{2}
  {\subsection*{Sommaire}}
\bigskip
On continue...
% ...
```

Table des matières

1 Titre 1
1.1 *Le sujet*, 1; 1.2 *Précision*, 1; 1.3 *Autre chose*, 1; 1.4 *Détail*, 1.
2 La suite 2
2.1 *Encore*, 2; 2.2 *Aussi*, 2.

1 Titre

Sommaire

1.1 Le sujet 1
1.2 Précision 1
1.3 Autre chose. . . . 1
1.4 Détail 1

Voila de quoi ça parle.

1.1 Le sujet

Le sujet...

1

FIG. 6.45: Exemple d'utilisation des tables partielles avec `titletoc`

graphie et l'index dans la table des matières.

L'objectif du package peut sembler simple, au premier abord, mais il ne l'est pas. Si on essayait de le faire à la main, on pourrait vouloir faire ça :

```
\tableofcontents
\addcontentsline{toc}{chapter}{\contentsname}
```

Mais ça ne marcherait pas, et ce pour une raison très simple : l'entrée est ajoutée à la *fin* de `\tableofcontents`, et donc si la table des matières commence en page 1, et qu'elle fait 3 pages de long (ce qui n'est pas invraissemblable du tout), alors c'est en page 3 qu'aura lieu l'appel à `\addcontentsline`. Ce n'est évidemment pas le bon résultat. La seule solution est donc de redéfinir la commande `\tableofcontents` pour qu'elle se charge d'ajouter l'entrée en table des matières. Mais cette redéfinition n'est pas simple non plus : la définition de cette commande n'est pas dans le noyau de LaTeX, mais dans la classe de documents que tu utilises. Pour redéfinir la commande, il faut donc savoir, en premier lieu, comment elle est définie par ta classe de documents, et toutes les classes ne le font pas de la même manière.

Option	**Description**
notbib	N'inclus pas la bibliographie dans la table des matières.
notindex	N'inclus pas l'index dans la table des matières.
nottoc	N'inclue pas la table des matières dans la table des matières.
notlot	N'inclue pas la liste des tableaux dans la table des matières.
notlof	N'inclue pas la liste des figures dans la table des matières.
chapter	Utilise `\chapter*` pour faire les titres si c'est possible.
section	Utilise `\section*` pour faire les titres si c'est possible.
numbib	Numérote la bibliographie.
numindex	Numérote l'index.
other	N'utilise ni `\chapter*` ni `\section*` pour faire les titres.
none	Désactive tout sauf un petit détail du package.

TAB. 6.13: Options de chargement du package `tocbibind`

Spontanément, le package fonctionne bien avec les classes standard de LaTeX (**article**, **report**, **book**, **proc** et **ltxdoc**), et a de bonnes chances de fonctionner avec d'autres classes.

Si tu es dans un cas standard[184], en chargeant le package il fera son office : la table des matières, la liste des figures, la liste des tableaux, la bibliographie, et l'index sont envoyés dans la table des matières. Si tu veux supprimer certains de ces éléments (par exemple pour que la table des matières ne se référence pas elle-même), tu peux utiliser les options du tableau 6.13.

`\tocfile{⟨titre⟩}{⟨extension⟩}`

La commande `\tocfile` est la commande que le package cache derrière la table des matières, la liste des figures, et la liste des tableaux, ainsi, pour `tocbibind`, `\tableofcontents` est équivalent à :

```
\tocfile{\contentsname}{toc}
```

6.17.1 Utilisation avec une autre classe

Si tu utilises le package dans une situation non-standard, ta vie risque de devenir compliquée. La première solution, ce sont les options chapter et section, elles sont à utiliser pour indiquer au package d'utiliser `\chapter*` ou `\section*` pour faire les titres des cinq éléments qu'il doit re-définir ; ça devrait suffir dans l'immense majorité des cas.

Si ta classe de document ne définit ni `\chapter*` ni `\section*`, ou si tu ne veux utiliser aucune de ces deux commandes pour faire le début des cinq éléments concernés, mais que tu veux utiliser `\machin*` par exemple, il faut passer l'option other au package et utiliser la commande `\tocotherhead` comme ça :

```
\tocotherhead{machin}
```

Pour les titres des cinq éléments qu'il re-définit, le package essayes de coller aux usages de LaTeX (ce qui lui permet d'interagir très bien avec `babel`, entre autres), mais il se peut que ta classe de document se fiche de ce mode d'interaction et définisse elle-même ces titres (c'est

[184]Tu utilises une classe standard, et pas trop de packages qui risquent de modifier les comportements que `tocbibind` attends : tu n'ajoutes pas de nouveau type de flottant, la bibliographie reste gérée par LaTeX, l'index est classique, etc.

Commande	Description
`\setindexname`	Change le nom de l'index.
`\settocname`	Change le nom de la table des matières.
`\setlotname`	Change le nom de la liste des tableaux.
`\setlofname`	Change le nom de la liste des figures.
`\settocbibname`	Change le nom de la bibliographie.

TAB. 6.14: Commandes de `tocbibind` pour fixer les titres

par exemple le cas de certaines classes mal faites pour des thèses), en dur. Dans ce cas-là, le package n'arrivera pas à reprendre la bonne valeur. Du coup, il prévoit les commandes du tableau 6.14 pour fixer ces titres. Tel que fonctionne le package, ces commandes ne sont pas compatibles avec `babel` : par exemple si tu modifies le nom de la table des matières avec la commande `\settocname`, au prochain changement de langue la définition de `babel` reprendra le dessus :

```
\usepackage{tocbibind}
\settocname{Programme}
% \contentsname vaut "Programme"
...
\begin{document}
\selectlanguage{francais}
% \contentsname vient d'etre redefinit par babel
\tableofcontents
```

Il faut donc re-définir le titre qui t'intéresse *juste avant* d'inclure l'élément correspondant.

6.17.2 Numéroter les listes

Si tu veux[185] que les listes, tables, etc. soient numérotées, le package prévoit quelques options pour ça. En fait, c'est très simple, il suffit qu'il fasse appel à `\chapter` à la place de `\chapter*`, ou à `\section` au lieu de `\section*`.

Cependant, le package considère que tu ne veux probablement pas voire apparaître le mot « Chapitre » en haut de la table des matières[186], et pour ça il se propose de faire un chapitre simplifié, ce qu'il appelle un `\simplechapter` dont l'en-tête ressemble à celui d'un chapitre, mais sans le mot chapitre.

`\simplechapter[`⟨*texte*⟩`]`

Si tu appelles cette commande, alors la commande `\chapter` ne produira plus tout à fait les mêmes têtes de chapitre, ce qu'elle produira va ressembler à :

[⟨*texte*⟩] 1{⟨*delim*⟩} Table des matières

Où `[`⟨*texte*⟩`]` est le texte que tu as indiqué en paramètre optionnel à `\simplechapter`, et {⟨*delim*⟩} est le contenu de la commande `\simplechapterdelim` (commande qui par défaut est vide, mais que tu peux redéfinir). Du coup, ces appels :

```
\simplechapter[Chap.]
```

[185] Ou si on te force, parce qu'en général, la majorité des gens *ne veulent pas* faire ce genre de conneries, mais ils s'y trouvent contraints, probablement pour de mauvaises raisons.

[186] Je trouve qu'il abuse un peu ce faisant, mais la fonctionnalité qu'il propose est intéressante.

```
\renewcommand\simplechapterdelim{:}
\chapter{Essai}
```

produiront ce résultat-là :

Chap. 1 : Essai

La commande `\restorechapter` permet de rendre à la commande `\chapter` son comportement normal (ce qui est important, pour que tu puisses l'utiliser dans le reste de ton document).

La commande `\tocchapter` fait appel à `\simplechapter` sans lui passer le paramètre optionnel (`[`⟨*texte*⟩`]` est donc vide), et fait en sorte que les tables, listes, index et bibliographie utilisent `\chapter` là où ils auraient utilisé `\chapter*`[187]. Le résultat est que tu obtiens des en-têtes qui sont numérotés mais ne portent pas le mot chapitre, et *ne peuvent pas* utiliser le paramètre optionnel de `\simplechapter`.

Si tu veux *vraiment* que la table des matières soit un chapitre à part entière, voilà comment tu peux procéder :

```
\tocchapter
\restorechapter
\tableofcontents
```

En effet, la commande `\restorechapter` rend à `\chapter` son mode de fonctionnement normal, et comme `\tocchapter` à dit au package d'utiliser `\chapter` au lieu de `\chapter*` pour faire les titres, tout fonctionne. Attention cependant : à partir de ce point-là dans ton source, tout ce qui sera produit par `\tocfile`, la liste des figures, par exemple, utilisera `\chapter`. Tu as donc tout intérêt à placer ces changements dans un groupe[188], et même à placer le tout dans une commande. On l'a vu tout à l'heure, le package `tocbibind` redéfinit `\tableofcontents` en se basant sur `\tocfile`, on va donc faire la même chose. Pour que la table des matières soit un chapitre à part entière, il suffit donc de faire :

```
\renewcommand\tableofcontents{%
  \bgroup
  \tocchapter
  \restorechapter
  \tocfile{\contentsname}{toc}
  \egroup
}
```

Si l'utilisation de `\simplechapter` te convient, mais que tu aurais aimé pouvoir utiliser le paramètre optionel, voilà comment tu peux faire :

```
\renewcommand\tableofcontents{%
  \bgroup
  \tocchapter
  \restorechapter
  \simplechapter[Chap.]
  \tocfile{\contentsname}{toc}
  \egroup
}
```

[187]Ne pas confondre : si la liste des figures est faite avec `\section*` dans ton document, par exemple si tu as utilisé l'option `section`, alors ta liste *ne sera pas modifiée* par la commande `\tocchapter`.

[188]Si tu ne sais pas ce qu'est un groupe, reporte-toi au chapitre neuvième page 571 qui traite de la programmation. Mais ce n'est pas très important pour ce qu'on cherche à faire ici.

Attention : quand on utilise cette astuce (appeler `\simplechapter` après `\tocchapter` pour exploiter le paramètre optionel), l'usage de `\restorechapter` est important : si tu ne l'utilises pas, ça revient à faire deux `\simplechapter` de suite, et du coup la définition originale de `\chapter` est définitivement[189] perdue.

La commande `\tocsection` fonctionne un peu comme `\tocchapter` : là où le package aurait utilisé `\section*` pour les titres, il utilisera `\section`. Il n'y a rien du style de `\simplechapter` pour le cas des sections (l'allure des section est *déjà* simple de toutes façons), la commande `\tocsection` est donc plus simple.

Quand le package utilise un type de titre qu'il n'a pas détecté tout seul, c'est-à-dire quand tu as utilisé l'option `other` et que tu as appelé la commande `\tocotherhead`, quel que soit le type de titre que tu ais utilisé, il considère que c'est une section, et donc fonctionne avec `\tocsection`. Du coup, tu peux user (et abuser) de ce fonctionnement, par exemple pour avoir des niveaux différents pour les différentes tables, ou pour que la table des matières soit un vrai chapitre, par exemple en faisant ça :

```
\usepackage[other]{tocbibind}
\renewcommand\tableofcontents{%
  \bgroup
  \tocotherhead{chapter}
  \tocsection
  \tocfile{\contentsname}{toc}
  \egroup
}
\renewcommand\listoffigures{%
  \bgroup
  \tocotherhead{section}
  \tocsection
  \tocfile{\listfigurename}{lof}
  \egroup
}
\renewcommand\listoftables{%
  \bgroup
  \tocotherhead{section}
  \tocsection
  \tocfile{\listtablename}{lot}
  \egroup
}
```

La figure 6.46 page suivante reprend ces définitions dans un exemple de ce qu'on peut obtenir.

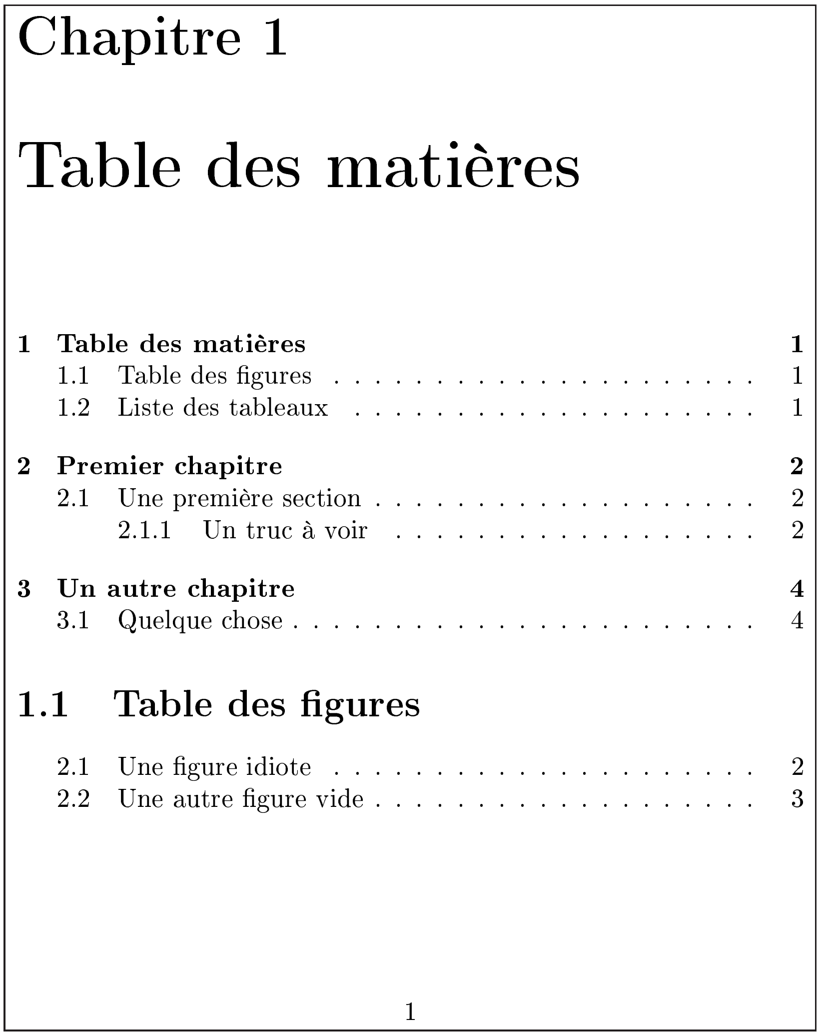
6.17.3 Utilisation des marques

Dans les classes de documents standard de LaTeX, le titre courant est systématiquement en majuscules. Si tu utilises `tocbibind` dans ce contexte-là, tout va bien : il suit cet usage-là.

[189]Enfin, définitivement jusqu'à la fin du groupe, ce qui dans notre cas est la fin de la commande `\tableofcontents`, cette perte ne serait donc pas très grave. Mais si tu reprends ces commandes directement dans ton document, sans utiliser `\bgroup` et `\egroup` alors la définition originale de `\chapter` est perdue jusqu'à la fin du groupe en cours également. Et il y de bonnes chances pour que, en pareil cas, le groupe en cours soit celui qui a été commencé par `\begin{document}`. La définition de `\chapter` serait donc perdue jusqu'à la fin du document.

```
\documentclass[a5paper]{report}
\usepackage[a5paper]{geometry}
\usepackage[other]{tocbibind}
% Definition des commandes comme vu dans le texte
\tableofcontents
\listoffigures
\listoftables

\chapter{Premier chapitre}
Bla bla bla
% etc
```



Chapitre 1

Table des matières

1 Table des matières **1**
1.1 Table des figures 1
1.2 Liste des tableaux 1

2 Premier chapitre **2**
2.1 Une première section 2
2.1.1 Un truc à voir 2

3 Un autre chapitre **4**
3.1 Quelque chose 4

1.1 Table des figures

2.1 Une figure idiote 2
2.2 Une autre figure vide 3

1

Fig. 6.46: Exemple d'utilisation perverse de `tocbibind`

Par contre, si tu personalises tes titres courants, par exemple avec un des packages que nous avons vu au titre 4 page 196, et que tu as personnalisé l'usage des marques, il y a fort à parier que tout ne se passe pas bien.

Le package `tocbibind` utilise la commande `\tocetcmark` pour positionner les marques pour les titres courants de la table des matières et des listes de figures et tableaux. Tu peux redéfinir cette commande pour qu'elle fasse quelque chose qui soit plus en accord avec ce dont tu as besoin. Cette commande prend un paramètre, qui est le titre du chapitre (ou de la section) correspondante.

6.17.4 Listes de flottants

Les packages qui définissent de nouveau flottants (voir à ce sujet au titre 7 page 403 à ce sujet) définissent aussi les commandes permettant d'avoir les listes de ces flottants. Tu peux redéfinir ces commandes comme on l'a vu précédemment pour `\listoffigures`, en utilisant `\tocfile`.

6.18 Table des matières : le package `tocloft`

Ce package, que nous devons à Peter Wilson [226], permet de changer la mise en forme de la table des matières. Il ne remet pas en cause la mécanique interne de LaTeX telle que nous l'avons vue au début (au paragraphe 6.1 page 319), simplement il re-définit les commandes `\l@chapter`, `\l@section`, etc, pour créer des versions qui font appel à des paramètres només (plutot que de les inclure en dur), et du coup tu pourras changer la valeur de chacun de ces paramètres sans avoir à toucher à des macros internes de LaTeX. C'est nettement moins puissant que `titletoc`, mais ça a beaucoup moins de chances de poser problème avec d'autres packages.

Les paramètres que tu peux personnaliser sont très nombreux, mais tu ne pourras cependant pas changer en profondeur la mise en forme (pas moyen, par exemple, de regrouper des entrées dans un paragraphe).

6.18.1 Paramètres de la table

Dans la suite, je noterai ⟨*toc*⟩ pour `toc` quand on parle de la table des matières, `lot` quand on parle de la liste des tableaux, ou `lof` pour la liste des figures. Ainsi, quand je parle des commandes `\cftmark`⟨*toc*⟩, ce sont les commandes `\cftmarktoc`, `\cftmarklot`, `\cftmarklof`, et éventuellement d'autres si tu définis d'autres tables.

Les commandes `\cftmark`⟨*toc*⟩ définissent les titres courrants des tables. La définition par défaut est sensiblement équivalente à ça :

```
\newcommand{\cftmarktoc}{%
  \markboth{\MakeUppercase\contentsname}%
           {\MakeUppercase\contentsname}}
```

Libre à toi de les redéfinir en fonction de ton style de page avec `\renewcommand`.

Les commandes `\cftbefore`⟨*toc*⟩`titleskip` sont des longueurs (dont tu peux changer la valeur par `\setlength`, donc) qui indiquent la taille du blanc vertical à laisser au dessus du titre de la table. Les commandes `\cftafter`⟨*toc*⟩`titleskip` qui sont également des longueurs indiquent la taille du blanc vertical après le titre.

Les commandes `\cft`⟨*toc*⟩`titlefont` indiquent la fonte à utiliser pour le titre de la table (`\Large`, `\bfseries`, etc) ; en fait ce sont juste des commandes qui sont exécutées avant le titre (et dans le même groupe), tu peux y ajouter autre chose qu'un changement de fonte.

```
\usepackage{tocloft}
\renewcommand\cfttoctitlefont{%
  \Large\sffamily\bfseries
}
\renewcommand\cftloftitlefont[1]{%
  \Large\ttfamily
  \raggedleft#1\par\nobreak
}
\tableofcontents
\listoffigures
\section{Premi\`ere section}
\begin{figure}[b]
Rien
\caption{Un truc idiot}
\end{figure}
```

Table des matières

`Table des figures`

1 Première section

1

FIG. 6.47: Exemple de mise en forme du titre de la table des matières avec le package `tocloft`

Les commandes `\cftafter`⟨*toc*⟩`title` sont ajoutées après le titre, mais pas dans le même groupe (en fait, après que le groupe du titre ait été terminé).

Par défaut, il n'y a pas de commande d'alignement mise en œuvre, du coup tu peux utiliser `\centering`, ou autre pour ajuster le centrage du titre. Si tu as besoin d'insérer quelque chose entre le titre de la table et la table elle-même, tu peux le placer dans la commande `\cftafter`⟨*toc*⟩`title`, mais dans ce cas là il faut garder en tête que le blanc vertical de `\cftafter`⟨*toc*⟩`titleskip` sera placé après.

La mise en forme du titre (de table des matières, par exemple) ressemble à ça :

```
{\cfttoctitlefont\contentsname}{\cftaftertoctitle}
\cftmarktoc
\par\nobreak
\vspace{\cftaftertoctitleskip}
```

Si tu veux modifier l'alignement du titre (mettons que tu veuilles le centrer, avec `\centering`, donc), il faut faire attention : les commandes qui changent l'alignement d'un paragraphe sont prises en compte *à la fin* du paragraphe. Si tu fais le changement dans `\cfttoctitlefont`, le changement sera annulé à la fin du groupe, c'est-à-dire à l'accolade fermante après `\contentsname`, et donc bien avant l'appel à `\par` qui termine le paragraphe. De là plusieurs solutions sont possibles. La première est de mettre la commande de centrage dans `\cftmarktoc`, mais elle n'a vraiment rien à y faire. La deuxième est de mettre un appel à `\centering\par` dans la commande `\cftaftertoctitle`, ce qui est mieux, mais pas très orthodoxe. Enfin la troisième solution est de dire que la commande `\cfttoctitlefont` prend un argument, qui dans notre cas sera le titre, et se charge de le mettre en forme.

Cette troisième solution donnerait ça, par exemple :

```
\renewcommand\cfttoctitlefont[1]{%
  \centering#1\par
}
```

La figure 6.47 reprend cette idée pour modifier le centrage de la liste des figures. Bien entendu, les choix typographiques faits dans l'exemple ne sont pas forcément intelligents pour un vrai document.

6.18.2 Mise en forme générale des entrées

Un certain nombre de commandes sont utilisées comme valeur par défaut de paramètres, ou s'appliquent à l'ensemble des entrées de la table.

`\cftdotsep` définit l'espacement par défaut entre les points de conduite. Tu pourras, même si ce n'est pas conseillé du tout, changer cette valeur pour certaines entrées ou pour certaines tables. La macro `\cftnodots` contient une valeur dont on est certain qu'elle est trop grande : quand l'espacement entre deux points est trop grand, LaTeX n'a même pas la place de placer le premier, et du coup il n'y a pas du tout de points de conduite (c'est un peu crade, mais efficace).

La largeur des numéros de page est normalement la même pour toutes les entrées, on la modifie avec `\cftsetpnumwidth`. C'est plutôt une mauvaise idée de la changer pour une entrée spécifique, mais on verra quand même un peu plus loin comment faire. De la même manière, la taille de la marge supplémentaire à droite est fixée par `\cftsetrmarg`, et il vaut mieux ne pas la changer en cours de route. Les numéros de page sont placés dans cette marge supplémentaire, il faut donc que la marge supplémentaire soit au moins aussi grande que la largeur des numéros de page, sinon, certains numéros de page pourraient venir s'écrir sur les points de conduite.

Par défaut, dans LaTeX, il n'y a pas de blanc entre les paragraphes de la table des matières (en dehors des blancs que chaque entrée requiert), si tu veux ajouter un blanc systématiquement à chaque paragraphe, tu peux le faire en changeant la valeur de `\cftparskip` avec `\setlength`.

6.18.3 Mise en forme spécifique des entrées

Dans ce qui suit, je noterai ⟨*sec*⟩ le nom abbrégé d'un type de titre : `part` pour `\part`, `chap` pour `\chapter`, `sec` pour `\section`, `subsec` pour `\subsection`, `subsubsec` pour `\subsubsection`, `para` pour `\paragraph`, `subpara` pour `\subparagraph`, `fig` pour les titres des figures, `subfig` pour les titres des sous-figures (package `subfigure`, voir à ce sujet le paragraphe 7.11 page 413), `tab` pour les titres des tableaux, et `subtab` pour les titres des sous-tableaux (package `subfigure` également). Ainsi, quand je parle des commandes `\cftbefore`⟨*sec*⟩`skip` ce sont les commandes `\cftbeforesecskip`, `\cftbeforechapskip`, etc.

Les commandes `\cftbefore`⟨*sec*⟩`skip` sont des longueurs et indiquent la taille du blanc vertical à laisser avant chaque entrée de type ⟨*sec*⟩.

Les commandes `\cft`⟨*sec*⟩`indent` sont également des longueurs et indiquent l'indentation des entrées de ce type-là par rapport à la marge de gauche. L'indentation, dans la mise en forme par défaut de LaTeX est la blanc qui va de la marge de gauche jusqu'au début du numéro de section. Les commandes `\cft`⟨*sec*⟩`numwidth` sont également des longueurs, et indiquent la largeur du numéro de section.

Un entrée est mise en forme en prennant en compte les paramètres suivant grosso-modo sous cette forme-là :

```
{\cft⟨sec⟩font {\cft⟨sec⟩presnum 4.3.2\cft⟨sec⟩aftersnum \hfil}%
              \cft⟨sec⟩aftersnumb Le titre}%
{\cft⟨sec⟩leader}%
{\cft⟨sec⟩pagefont 42}%
\cft⟨sec⟩afterpnum\par
```

```
\usepackage{tocloft}
\renewcommand\cftsubsecafterpnum{%
  \advance\rightskip by 10pt}
\tableofcontents
\clearpage
```

Table des matières

1

FIG. 6.48: Exemple de mise en forme d'une table des matières avec le package `tocloft`

Les commandes `\cft`⟨*sec*⟩`font` indiquent donc la fonte globale de l'entrée, pour le numéro et le titre. Les commandes `\cft`⟨*sec*⟩`pagefont` indiquent la fonte pour le numéro de page (il vaut mieux que ce soit la même, en principe). Les 3 commandes qui entourent le numéro de la section ne sont utilisées que si la section est numérotée : `\cft`⟨*sec*⟩`presnum` précède le numéro de section (par exemple pour en changer le centrage ou la fonte) ; `\cft`⟨*sec*⟩`aftersnum` suit directement le numéro de section[190] ; et la commande `\cft`⟨*sec*⟩`aftersnumb` précède immédiatement le titre (mais elle reste attachée au numéro de section et n'est pas utilisée pour les sections non-numérotées).

Les commandes `\cft`⟨*sec*⟩`leader` produisent les points de conduite. Il vaut mieux ne pas trop y toucher si tu ne sais pas faire. Les points sont espacés de `\cft`⟨*sec*⟩`dotsep` Enfin, les commandes `\cft`⟨*sec*⟩`afterpnum` viennent clore l'ensemble.

Si tu veux, par exemple, que le numéro de page des sous-sections soit renfoncé sur la droite par rapport aux numéros de page des sections, il suffit de réduire temporairement la marge de droite. Le package ne prévoit rien pour ça, mais il se trouve que cette information n'est utilisée par LaTeX qu'a la toute fin d'un paragraphe. Il se trouve que, juste avant la fin du paragraphe, la commande `\cftsubsecafterpnum` est appelée. Il suffit que cette commande modifie la marge de droite pour qu'elle soit changée pour tout le paragraphe. On peut donc la redéfinir avec :

```
\renewcommand\cftsubsecafterpnum{%
  \advance\rightskip by 10pt}
```

La figure 6.48 présente un exemple d'une telle utilisation.

`\cftsetindents`{⟨*sec*⟩}{⟨*indent*⟩}{⟨*largeur numéro*⟩}

La commande `\cftsetindents` modifie les valeurs de `\cft`⟨*sec*⟩`indent` et `\cft`⟨*sec*⟩`numwidth`.

Si tu veux que les numéros de page soient directement après le titre de la section (et non pas le long de la marge de droite avec des points de conduite), tu peux l'obtenir en

[190] En vrai, le numéro est dans une boîte, pas dans un groupe. La commande `\hfil` fait que le numéro de section est *a priori* à gauche de sa boite. Tu peux changer ça en mettant d'autres commandes d'alignement via `\cft`⟨*sec*⟩⟨*pre ou after*⟩`snum`.

```
\usepackage{tocloft}
\renewcommand
    \cftsubsecleader{,\nobreak}
\renewcommand
    \cftsubsecafterpnum
    {.\cftparfillskip}
\renewcommand
    \cftsubsecfont{\itshape}
\renewcommand
    \cftsubsecpagefont{\itshape}
\tableofcontents
\clearpage
```

Table des matières

1

Cet exemple fonctionne, mais il pose plusieurs problèmes, qui sont des limitations du package `tocloft`, et qui ne sont pas simples à contourner. Le principal est que le numéro de page est mis en forme dans un boîte qui fait `\@pnumwidth`, et que ce n'est pas modifiable. On peut changer la façon dont le numéro est centré dans cette boîte (par défaut il est à droite de la boîte), mais on ne peut pas ne pas avoir cette boîte. La seule solution est de redéfinir la commande `\cft`⟨*sec*⟩`fillnum` qui prend le numéro de page en argument et se charge de produire les points de conduite et le numéro de page.

FIG. 6.49: Exemple de mise en forme d'une table des matières avec le package `tocloft`

---

redéfinissant `\cft`⟨*sec*⟩`leader` pour qu'elle soit vide et `\cft`⟨*sec*⟩`afterpnum` pour qu'elle vienne remplir le paragraphe :

```
\renewcommand{\cftsecleader}{}
\renewcommand{\cftsecafterpnum}{\cftparfillskip}
```

La commande `\cftparfillskip` est fournie par le package pour ce cas très précis : sans cette commande les règlages choisis font que la dernière ligne du paragraphe est étirée jusqu'au bord droit.

La figure 6.49 présente le même document que l'exemple précédent avec une mise en forme différente de la table des matières. Dans cet exemple les points de conduite ont été supprimés pour les sous sections, le numéro de page est placé directement après le titre, séparé par une virgule, et l'entrée est mise en italique.

`\cftpagenumbersoff{`⟨*type*⟩`}`

`\cftpagenumberson{`⟨*type*⟩`}`

Ces commandes suppriment (ou pas) le numéro de page pour les entrées de type {⟨*type*⟩} (où {⟨*type*⟩} est le nom complet un type d'entrée, par exemple `subsection` ou `figure`). Si tu veux supprimer les numéros de page seulement pour certains passages de ton document, tu peux utiliser ça :

```
\addtocontents{toc}{\cftpagenumbersoff{chapter}}
% Passage où les numéros de page des chapitre ne
% seront pas dans la table des matières
\addtocontents{toc}{\cftpagenumberson{chapter}}
```

```
% Ici les numéros de page des chapitre sont de
% nouveau présents.
```

6.18.4 Nouvelles listes et entrées

Le package prévoit quelques commandes pour faciliter la création de nouvelles listes (par exemple pour faire ta liste de théorèmes, ou d'autre chose). Malheureusement, l'interface est assez pauvre, et pas forcément très pratique.

`\newlistof[`⟨*groupe*⟩`]{`⟨*entrée*⟩`}{`⟨*extension*⟩`}{`⟨*titre*⟩`}`

La commande `\newlistof` crée une nouvelle liste (comme la liste des figures), pour d'autres éléments que tu voudrais numéroter et recenser dans une table. Le deuxième paramètre, {⟨*entrée*⟩} est le nom d'un compteur qui sera créé pour compter les éléments de ce type-là. Le premier paramètre, [⟨*groupe*⟩], optionnel, indique de quel autre compteur celui-là dépend. Si tu utilises `[chapter]` le compteur {⟨*entrée*⟩} sera remis à zéro à chaque changement du compteur de chapitre. Le troisième paramètre, {⟨*extension*⟩} indique l'extension du nom de fichier à utiliser pour cette nouvelle liste, et le dernier paramère, {⟨*titre*⟩}, indique le titre de la nouvelle liste. Une nouvelle commande, `\listof`⟨*entrée*⟩ est créée. Un compteur ⟨*entrée*⟩`depth` est également créé qui indique la profondeur souhaitée pour cette liste, comme `tocdepth` pour la table des matières.

Dans la mesure où la création du compteur n'est pas optionnelle, c'est mal pratique à utiliser pour la liste des flottants qu'on aurait pu définir avec certains packages d'extension (par exemple `float`, discuté au paragraphe 7.4 page 407).

`\newlistentry[`⟨*groupe*⟩`]{`⟨*entrée*⟩`}{`⟨*extension*⟩`}{`⟨*niveau - 1*⟩`}`

La commande `\newlistentry` définit une nouvelle entrée pour une liste. Elle en définit le compteur (sale habitude). Les paramètres sont les mêmes que pour `\newlistof`, sauf le dernier, {⟨*niveau - 1*⟩} qui indique la profondeur moins un du nouvel élément dans la liste.

La figure 6.50 page suivante présente un exemple complet de création d'une nouvelle liste avec le package `tocloft`.

6.18.5 Commandes exotiques

Le package prévoit également un certain nombre de commandes plus ou moins expérimentales (qui pourraient donc disparaître ou être modifiées dans les prochaines versions).

Le premier jeu de trois commandes permet d'ajouter une (pas forcément) courte description d'un chapitre après son titre.

`\cftchapterprecis{`⟨*Texte*⟩`}`

`\cftchapterprecistoc{`⟨*Texte*⟩`}`

`\cftchapterprecishere{`⟨*Texte*⟩`}`

La première commande, `\cftchapterprecis`, fait simplement appel aux deux autres avec le même texte en argument. La deuxième, `\cftchapterprecishere`, ajoute le texte juste-là où elle est appelée et seulement là. Le texte est mis dans l'environnement `quote` et est légèrement rapproché du titre du chapitre par rapport à un texte normal. La troisième commande, `\cftchapterprecistoc`, renvoit le texte dans la table des matières, sans le faire apparaître dans le chapitre. Le texte y sera aligné avec le titre du chapitre précédent, et sera

```
* \usepackage{tocloft}
* \newlistof{idee}{loi}{Id\'ees}
* \newenvironment{idee}[1]{%
*   \refstepcounter{idee}%
*   \noindent
*   \textbf{\titreidee} #1\par
*   \addcontentsline{loi}{idee}
*        {\protect
*         \numberline{\theidee}
*         #1}
*   \begin{quote}
* }{%
*   \end{quote}
* }
* % \texte comme d'habitude
  \newcommand\titreidee{%
    Id\'ee \theidee:}
  \listofidee
  \section{Mon document}
  \begin{idee}{Tester Om\'ega}
  \texte \texte
  \end{idee}
```

Idées

1 Mon document

Idée 1 : Tester Oméga

> Texte un peu long qu'on répète encore et toujours pour remplir. Texte un peu

1

Le titre de l'environnement `idee` est fait en deux fois, en passant par la commande `\titreidee` pour que les deux-points qui sont dans ce titre soient correctement traités par le package `babel`.

Si les deux points avaient été dans une commande définie avec le `\begin{document}` il n'auraient pas été pris en compte par `babel` et l'espacement aurait été faux.

FIG. 6.50: Exemple d'une nouvelle liste avec le package `tocloft`

mis en italique (ce choix n'est pas paramètrable, de même que tu ne peux pas facilement ajouter du blanc après ce texte).

`\cftlocalchange`{⟨*extension*⟩}{⟨*largeur num. page*⟩}{⟨*marge droite*⟩}

La commande `\cftlocalchange` permet de changer localement la largeur réservée au numéro de page et la marge supplémentaire à droite dans une table, par exemple pour insérer une entrée spéciale qui aurait un numéro de page épouvantablement long (en chiffres romains, par exemple). Le premier paramètre indique la table concernée (`toc`, `lof`, etc) ; le deuxième indique la nouvelle largeur des numéros de page, et le troisième indique la nouvelle marge supplémentaire à droite (il faut que ce soit au moins autant que la largeur du numéro de page).

Par exemple, pour indiquer le titre du dessin de la couverture :

```
\cftlocalchange{lof}{5cm}{5cm}
\cftaddtitleline{lof}{figure}{Dessin de M.C. Escher}{couverture}
\cftlocalchange{lof}{1.55em}{2.55em}
```

`\cftaddtitleline`{⟨*extension*⟩}{⟨*type*⟩}{⟨*titre*⟩}{⟨*page*⟩}

`\cftaddnumtitleline`{⟨*extension*⟩}{⟨*type*⟩}{⟨*numéro*⟩}{⟨*titre*⟩}{⟨*page*⟩}

Cette commande, `\cftaddtitleline` fonctionne exactement comme `\addcontentsline`, sauf que tu peux préciser le numéro de page (voir page 323 à ce sujet). Pour ajouter un titre numéroté, il faudra utiliser `\aftaddnumtitleline`.