

# Les extensions titlesec, titleps et titletoc \*

Javier Bezos<sup>†</sup>

2011-12-15

## Table des matières

<b>1. Introduction</b>	<b>1</b>
<b>2. Interface simple</b>	<b>2</b>
2.1. Format, 2.—2.2. Espacement, 2.—2.3. Majuscules, 3.—2.4. Outils, 3.	
<b>3. Interface avancée</b>	<b>3</b>
3.1. Format, 3.—3.2. Espacement, 4.—3.3. Outils liés à l’espacement, 5.—3.4. Filets, 6.—3.5. Styles de page, 8.—3.6. Sauts de page, 8.—3.7. Autres options de l’extension, 9.—3.8. Paramétrages étendus, 9.—3.9. Creating new levels and changing the class, 10.	
<b>4. Additional Notes</b>	<b>11</b>
4.1. Fixed Width Labels, 11.—4.2. Starred Versions, 11.—4.3. Variants, 12.—4.4. Putting a Dot after the Section Title, 12.	
<b>5. titleps and Page Styles</b>	<b>13</b>
<b>6. Contents : The titletoc package</b>	<b>13</b>
6.1. A ten-minute guide to titletoc, 13.—6.2. And more, 15.—6.3. Partial TOC’s, 17.—6.4. Partial lists, 18.—6.5. Examples, 18.—6.6. Inserting a figure in the contents, 18.—6.7. Marking entries with asterisks, 19.	
<b>7. La philosophie de titlesec</b>	<b>19</b>
<b>8. Annexes</b>	<b>19</b>
9.1. Un exemple complet, 22.—9.2. Classes standards, 23.—9.3. Exemple de chapitre, 23.	

## 1. Introduction

Cette extension est essentiellement un remplacement — partiel ou total — des macros  $\LaTeX$  liées au sectionnement — principalement les titres, en-têtes et tables des matières. Le but est d’apporter de nouvelles fonctionnalités non disponibles actuellement sous  $\LaTeX$  : si vous souhaitez juste une interface plus accessible que celle d’un  $\LaTeX$  standard mais sans changer la façon dont  $\LaTeX$  travaille, vous pouvez recourir à fancyhdr de Piet van Oostrum, sectsty de Rowland McDonnell et tocloft de Peter Wilson, avec lesquels vous pouvez faire de jolies choses.<sup>1</sup>

Entre autres nouvelles fonctionnalités se trouvent :

- différentes classes et « formes » de titres, avec des outils autorisant une grande variété de formats. Vous pouvez définir des formats différents pour les pages de gauche et de droite, avec des titres numérotés ou pas, mesurer la largeur d’un titre, ajouter un nouveau niveau de sectionnement, utiliser des graphiques et bien plus encore. L’Annexe montre un bon nombre d’exemples : allez-y tout de suite !
- des en-têtes et pieds de pages définis sans utilisation des commandes de type `\...mark`, et contenant éventuellement ✖ **des marques de haut, de première valeur et de double valeur** ✖. Des marques de haut de page correctement synchronisées avec les titres, sans incompatibilités

---

\*L’extension titlesec est actuellement en version 2.10.0. © 1998–2011 Javier Bezos. L’extension titletoc est actuellement en version 1.6. L’extension titleps est actuellement en version 1.1.0 © 1999–2011 Javier Bezos. Javier Bezos. Tous droits réservés.

<sup>†</sup>Pour des rapports d’erreur, commentaires et suggestions, allez à <http://www.tex-tipografia.com/contact.html>. L’anglais n’étant pas mon point fort, contactez-moi lorsque vous trouvez des erreurs dans ce manuel. D’autres extensions du même auteur : gloss (avec José Luis Díaz), enumitem, accents, tensind, esindex, dotlessi, babeltools.

1. Dans la mesure où les commandes de sectionnement sont réécrites, leur comportement peut être quelque peu différent dans certains cas.

avec la mécanique des flottants. Des éléments décoratifs ajoutés aisément, incluant des environnements d'image.

- de jolies tables des matières de forme libre, avec la possibilité de grouper les entrées de différents niveaux en un paragraphe ou de changer les formats des entrées au milieu d'un document.

Titlesec fonctionne avec les classes standards et de nombreuses autres, incluant celles de l'AMS, et elle fonctionne sans aucune difficulté avec `hyperref`.<sup>2</sup> Malheureusement, elle n'est pas compatible avec `memoir`, qui propose ses propres outils avec un sous-ensemble limité des fonctionnalités disponibles dans titlesec.

Comme d'habitude, chargez l'extension de façon classique avec `\usepackage`. Redéfinissez alors les commandes de sectionnement avec les paramétrages simples prédéfinis (voir section « Interface simple ») ou avec les commandes mises à disposition si vous souhaitez des formats plus élaborés (voir section « Interface avancée »). Dans ce dernier cas, vous avez uniquement besoin de redéfinir les commandes que vous utiliserez. Les deux méthodes sont disponibles au même moment mais parce que `\part` est habituellement implémentée de façon non standard, elle reste inchangée par les paramétrages prédéfinis et doit être modifiée avec l'aide de l'« Interface Avancée ».

## 2. Interface simple

La façon la plus simple de changer le format se fait par le moyen d'un ensemble d'options de l'extension et par un couple de commandes. Si les fonctionnalités apportées par cet ensemble d'outils vous donne satisfaction, vous n'avez pas besoin d'aller plus loin dans ce manuel. Lisez uniquement cette section et ignorez les suivantes.

### 2.1. Format

Il y a trois groupes d'options contrôlant les fontes, tailles et alignements. Vous n'avez pas besoin de définir tous ces groupes puisque des valeurs par défaut sont disponibles pour chacun ; cependant, vous devez au moins utiliser une option parmi celles possibles si vous souhaitez utiliser ce « paramétrage simple ».

rm sf tt md bf up it sl sc

Option sélectionnant la famille/série/forme correspondante. La valeur par défaut est `bf`.

big medium small tiny

Option sélectionnant la taille des titres. Elle est fixée par défaut à `big`, ce qui donne la taille retenue dans les classes standards. Avec `tiny`, les sectionnements (chapitres exceptés) sont composées à la même taille que le texte. `Medium` et `small` sont des présentations intermédiaires entre les deux premières.

raggedleft center raggedright

Option contrôlant l'alignement.

### 2.2. Espacement

compact

Cette option est indépendante de celles vues ci-dessus et réduit l'espace au-dessus et au-dessous des titres.

---

2. Cependant, notez bien que les classes AMS réimplémentent les commandes internes d'origine. Ces changements seront perdus ici. La compatibilité avec `hyperref` a été testée avec `dvips`, `dvipdfm` et `pdftex` mais il s'agit d'une fonctionnalité non suivie. Pensez à vérifier que votre version d'`hyperref` est compatible avec titlesec.

## 2.3. Majuscules

```
uppercase
```

2.9 Mise en majuscules des titres. Selon la classe, cela peut ne marcher avec `\chapter` et `\part`.

## 2.4. Outils

```
\titlelabel{<format-de-numérotation>}
```

Changement du format de la numérotation des sections, sous-sections, etc. Une commande `\thetitle` est fournie et correspond respectivement à `\thesection`, `\thesubsection`, etc. La valeur par défaut des classes standards est

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

et vous pouvez simplement ajouter un point après le compteur avec

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

Ceci a été fait dans ce document.

```
\titleformat*{<commande>}{<format>}
```

Cette commande permet de changer le `<format>` d'une commande de sectionnement, comme par exemple :

```
\titleformat*{\section}{\itshape}
```

## 3. Interface avancée

Deux commandes sont mises à disposition pour changer le format des titres. La première est utilisée pour le format « interne », autrement dit la forme, la fonte, la numérotation, etc. La seconde définit le format « externe », autrement dit l'espacement avant et après, l'indentation, etc. Ce principe est pensé pour simplifier les définitions car, dans la plus plupart des cas, vous souhaitez modifier soit l'espacement, soit le format<sup>3</sup>. Ceci redéfinit les commandes de sectionnement existantes mais n'en crée par de *nouvelles*. De nouveaux niveaux de sectionnement peuvent être ajoutés avec `\titleclass`, comme décrit ci-après, et leur format peut être fixé par les commandes décrites ici.

### 3.1. Format

Un ensemble de formes contrôlant la distribution basique des éléments d'un titre est proposé. Les formes disponibles sont :

**hang** est la valeur par défaut, avec une numérotation en retrait (comme avec `\section` des classes standards).

**block** compose le titre complet en un bloc (un paragraphe) sans mise en forme additionnelle. Utile pour les titres centrés<sup>4</sup> et les mises en forme spéciales (incluant des outils graphiques comme `picture`, `pspicture`, etc.).

**display** place la numérotation dans un paragraphe séparé (comme avec `\chapter` dans les classes standards).

**runin** Un titre suivi directement du texte sur la même ligne (comme avec `\paragraph` dans les classes standards).

3. L'information est « extraite » des commandes de sectionnement de la classe, excepté pour les chapitres et parties. Les définitions sont supposées utiliser `@startsection` — si les sections ont été définies sans cette commande, des valeurs arbitraires de format sont proposées et peuvent être modifiées par la suite. Malheureusement, il n'y a pas de manière de récupérer les formats des chapitres ou parties ; un format similaire à celui des classes standards est alors utilisé.

4. La numérotation sera légèrement déplacé vers la gauche si le titre fait plus d'une ligne de long et que la forme `hang` est utilisée, exception faite des cas avec des `\\` explicites.

**leftmargin** place le titre dans la marge de gauche. Les titres en bas de page sont déplacés sur la page suivante et ne *\*stick out\** pas dans la marge du bas, ce qui signifie que de longs titres peuvent mener à des pages trop peu remplies<sup>5</sup>. Dans ce cas, vous pouvez augmenter l'étirabilité des éléments de la page, utiliser `\raggedbottom` ou utiliser l'option d'extension `nobottomtitles` décrite ci-après. Dans la mesure où le mécanisme retenu est indépendant de celui des paragraphes placés dans les marges, ils peuvent se superposer. Un synonyme obsolète à cette forme est `margin`.

**rightmargin** est similaire à `leftmargin` mais pour la marge de droite.

**drop** dispose le texte autour du titre, uniquement si le premier paragraphe est plus que le titre (sinon, ils se superposent). Les commentaires associés à `leftmargin` s'appliquent également ici.

**wrap** est assez proche de `drop`. La seule différence est que, tandis que l'espace réservé dans `drop` pour le titre est fixé, il est automatiquement ajusté avec `wrap` à la ligne la plus longue. Les limitations expliquées ci-après pour `calwidth` s'appliquent également ici.

**frame** Similaire à `display`, mais le titre sera encadré.

Notez bien, cependant, que certaines formes n'ont aucun sens dans le cas des chapitres et des parties.

```
\titleformat{<commande>}[<forme>]{<format>}{<num>}{<sep>}{<code-avant>}{<code-après>}
```

Ici,

- `<commande>` est la commande de sectionnement à redéfinir, autrement dit `\part`, `\chapter`, `\section`, `\subsection`, `\subsubsection`, `\paragraph` ou `\subparagraph`.
- La forme du paragraphe est fixée par `<forme>`, dont les valeurs possibles sont celles décrites ci-dessus.
- `<format>` est le format appliqué à l'ensemble du titre — numérotation et texte. Cet argument peut contenir des éléments verticaux (et horizontaux avec certaines formes) qui sont composés juste après l'espace au-dessus du titre.
- La numérotation est définie dans `<num>`. Vous pouvez le laisser vide s'il n'y a pas de numérotation de sectionnement à ce niveau, mais cela n'est pas recommandé car le nombre n'est pas supprimé dans la table des matières et dans les en-têtes (ou titres courants).
- `<sep>` est la séparation horizontale entre la numérotation et le corps du titre et doit être une longueur (qui ne doit pas être vide). Cet espace est vertical dans la forme `display`; dans `frame`, il s'agit de la distance entre le texte et l'encadrement. `<num>` et `<sep>` sont tous deux ignorés dans les versions étoilées des commandes de sectionnement. Si vous utilisez `picture` et ses semblables, mettez ce paramètre à 0 pt.
- `<code-avant>` est un code précédent le corps du titre. La toute dernière commande peut prendre un argument qui est le texte du titre<sup>6</sup>. Toutefois, avec l'option d'extension `explicit`, le titre doit être donné explicitement avec `#1` (voir ci-dessous).
- `<code-après>` est le code suivant le corps du titre. Le matériel alors composé est en mode vertical avec `hang`, `block` et `display` tandis qu'il est en mode horizontal avec `runin` et `leftmargin` (`[2.7]` avec ce dernier au début du paragraphe). Sinon, le code est ignoré.

```
\chaptertitlename
```

Elle renvoie par défaut à `\chaptername` sauf dans le cas des annexes où elle renvoie vers `\appendixname`. Utilisez-là en lieu et place de `\chaptername` lorsque vous définissez un chapitre.

### 3.2. Espacement

```
\titlespacing*{<commande>}{<gauche>}{<avant-sep>}{<après-sep>}{<droite-sep>}
```

La version étoilée supprime l'indentation du paragraphe suivant le titre, sauf avec `drop`, `wrap` et

5. Cependant, les flottants placés une ou deux lignes après le titre interfèrent avec la découpe de page utilisée ici et le titre peut parfois rester placé en bas de page

6. Rappelez-vous que la taille de la fonte peut être changée de façon sécurisée uniquement entre les paragraphes et que les changements dans le texte doivent être faits localement avec un groupe ; Sinon le ✖ **leading** ✖ pourrait être faux — trop grand ou trop petit.

runin pour lesquels cette possibilité n’a pas de sens.

- `<gauche>` augmente la marge de gauche, sauf pour les formes `...margin` et `drop` où ce paramètre fixe la largeur du titre, pour `wrap`, la largeur maximale, et dans `runin` l’indentation juste avant le titre. Avec une valeur négative, le titre ✖ **overhangs** ✖<sup>7</sup>.
- `<sep-avant>` est l’espace vertical avant le titre.
- `<sep-après>` est la séparation entre le titre et le texte — verticale avec `hang`, `block`, et `display`, horizontale avec `runin`, `drop`, `wrap` and `...margin`. En retenant une valeur négative, vous pouvez définir un espace réel plus petit que `\parskip`<sup>8</sup>.
- Les formes `hang`, `block` et `display` disposent de la possibilité d’augmenter la marge `<sep-droit>` avec cet argument optionnel.

Si vous n’aimez pas saisir les valeurs complètes des espaces, incluant les paramètres plus et moins, une abréviation `*n` est disponible. Dans l’argument `<sep-avant>` c’est équivalent à `n ex` avec une certaine ✖ **extensibilité** ✖ et une infime ✖ **compressabilité** ✖. Dans le `<sep-après>`, une certaine ✖ **extensibilité** ✖ (plus petite) et aucune ✖ **compressabilité** ✖<sup>9</sup>. Ainsi vous pouvez écrire :

```
\titlespacing{\section}{0pt}{*4}{*1.5}
```

Les longueurs `\beforetitleunit` et `\aftertitleunit` sont utilisées comme unités dans les paramètres `*` et vous pouvez les changer si vous n’aimez pas les valeurs prédéfinies (ou pour de légers changements dans la composition, par exemple).

**Notes.** `\titlespacing` ne fonctionne pas avec `\chapter` et `\part` à moins que vous ne changiez leur format de titre aussi bien par l’utilisation de `\titleformat`, les paramétrages simples ou `\titleclass`. Les arguments de `\titlespacing` doivent être des dimensions ; `\stretch` inclut une commande et ainsi génère une erreur.

### 3.3. Outils liés à l’espacement

Ces commandes sont fournies comme outils pour `\titleformat` et `\titlespacing`.

```
\filright \filcenter \filleft \fillast \filinner \filouter
```

Variantes des commandes `\ragged...`, avec de légères différences. En particulier, les commandes `\ragged...` suppriment les espaces à droite et à gauche disposés par `\titlespacing`<sup>10</sup>. `fillast` justifie le paragraphe, à l’exception de sa dernière ligne qui est centrée<sup>11</sup>. Ces commandes fonctionnent également pour la numérotation associée à `frame`.

`\filinner` et `\filouter` correspondent à `\filleft` ou `\filright` selon la page où elles se trouvent. Du fait du découpage asynchrones des pages par  $\TeX$ , ces commandes peuvent être utilisées uniquement dans `\chapter`. Si vous souhaitez un outil général pour définir différents formats dépendant de la page, voyez « ✖ **Paramétrage étendu** ✖ » ci-dessous.

```
\wordsep
```

L’espace ✖ **inter-mot** ✖ pour la fonte courante.

```
indentafter noindentafter (options d’extension)
```

#### Contournement des paramétrages pour toutes les commandes de sectionnement<sup>12</sup>

7. Ce paramètre n’est pas équivalent à `<indent>` de `\@startsection`, qui ne fonctionne pas correctement. Avec une valeur négative dans ce dernier et si `<indent>` est plus grand que la largeur de la numérotation, la première ligne du titre commencera dans la marge extérieure, comme attendu, mais les lignes suivantes ne le feront pas ; pire, les lignes seront raccourcies du côté de la marge droite.

8. Voir Goossens, Mittelbach and Samarin : *LaTeX Companion*, Reading, Addison Wesley, 1993, p. 25.

9. Ils correspondent à `n` fois `1ex` plus `.3ex` minus `.06ex` et `1ex` plus `.1ex` respectivement.

10. Rappelez-vous que l’extension `ragged2e` définit quelques commandes additionnelles pour l’alignement comme `\justifying`.

11. Le nom est un petit peu étrange mais il est court.

12. Anciennement nommées `indentfirst` et `nonindentfirst`, maintenant obsolètes.

<code>rigidchapters rubberchapters</code> (options d'extension)
---

Avec `rigidchapters`, l'espacement pour les titres des chapitres est toujours le même et `sep-après` dans `\titlespacing` ne représente pas l'espace entre le bas du texte du titre et le corps du texte comme décrit plus haut mais depuis le haut du texte du titre, autrement `\sep-avant` + `\sep-après` est maintenant une distance fixe entre le haut du **✖ corps de page ✔** et le texte principal. La valeur par défaut est `rubberchapters` où `\sep-après` désigne la séparation entre le titre et le texte, comme à l'accoutumée. En effet le nom est trompeur car il ne s'applique pas qu'au chapitre par défaut mais à tout titre de **✖ classe haute ✔** (les classes sont développées dans la suite du document).

<code>bottomtitles nobottomtitles nobottomtitles*</code> (options d'extension)
--

Si `nobottomtitles` est sélectionné, les titres proches de la marge du bas sont renvoyés en page suivante et la marge sera **✖ ragged ✔**. L'espace minimal requis pour que la marge du bas ne déplace pas le titre est fixée (approximativement) par

```
\renewcommand{\bottomtitlespace}{\length}
```

dont la valeur par défaut est `.2\textheight`. Une simple marge **✖ ragged ✔** sur la page précédente est obtenue en mettant une valeur de 0 pt. `bottomtitles` est l'option par défaut, qui donne à `\bottomtitlespace` une valeur négative.

L'option `nobottomtitles*` fournit des calculs plus précis mais les titres des formes `margin`, `wrap` ou `drop` peuvent être mal placés. En temps normal, vous devriez utiliser la version étoilée.

<code>aftersep largestsep</code> (options d'extension)
--

Par défaut, lorsqu'il y a deux titres consécutifs, l'espace `\sep-après` du premier titre est utilisé entre les deux titres. Parfois, ce comportement n'est pas souhaité, en particulier quand l'espace `\sep-avant` est plus grand que `\sep-après` (sinon le comportement par défaut semble préférable). Avec `largestsep`, le plus grand espace est utilisé. La valeur par défaut est `aftersep`.

<code>\\ \\*</code> <code>pageatnewline</code> (option d'extension)
--

**[2.6]** Dans les versions 2.6 et suivantes, `\\` ne permet pas de saut de page et devient un équivalent de `\\*`. Comme je présume que personne ne veut un saut de page dans un titre, ceci est le comportement par défaut. Si, pour une raison étrange, vous souhaitez autoriser ces sauts de page dans les titres, utilisez l'option `pageatnewline`, qui est fournie pour des questions de rétrocompatibilité.

### 3.4. Filets

L'extension inclut des outils pour aider à l'ajout de filets et autres éléments au-dessus et au-dessous du titre. Les marges des titres pouvant être modifiées, ces commandes prennent en compte les paramètres locaux pour placer les filets proprement. Elles tiennent également compte de l'espace utilisé dans les titres marginaux.

<code>\titleline[alignement]{matériel horizontal}</code> <code>\titlerule[hauteur]</code> <code>\titlerule*[largeur]{texte}</code>
--

La commande `\titleline` permet l'insertion d'une ligne, qui peut contenir du texte et d'autres matériaux « horizontaux ». Elle est pensée principalement pour des filets et des **✖ leaders ✔** mais elle peut servir également pour d'autres besoins. La ligne a une largeur fixée et doit être remplie, autrement dit `\titleline{CHAPITRE}` produit une boîte **✖ sous-remplie ✔**. Ici le paramètre optionnel `alignement` (l, r ou c) peut aider, de telle sorte que vous puissiez simplement saisir `\titleline[c]{CHAPTER}` <sup>13</sup>.

<sup>13</sup>. La valeur par défaut est le paramètre `s` de la commande `\makebox`.

L'utilisation de `\titleline` à des endroits où un matériel vertical n'est pas attendu peut conduire à des anomalies. En d'autres termes, vous pouvez l'utiliser dans les **✖ arguments ✖** `<format>` (toujours) et `<code-après>`, que ce soit dans les formes `hang`, `display` et `block`. Pour la forme `display`, vous pouvez la placer au tout début des **✖ arguments ✖** `<before-code>` et `<label>`. Mais vous pouvez tester cette commande ailleurs : il y a de grandes chances qu'elle fonctionne à d'autres endroits.

La commande `\titlerule`, qui est incluse automatiquement dans `\titleline` si nécessaire, peut être utilisée pour construire des filets et des **✖ fillers ✖**. La version non étoilée trace des filets épais de 0.4 pt ou de `<height>` si ce dernier est présent. Par exemple,

```
\titlerule[.8pt]%
\vspace{1pt}%
\titlerule
```

traces deux filets de différents épaisseurs avec une séparation de 1 pt.

La version étoilée fait des **✖ leaders ✖** avec le `<texte>` répété dans des boîtes de sa largeur naturelle. Cette largeur de boîte peut être modifiée avec `<largeur>` mais la largeur de la première boîte reste inchangée pour que le `<texte>` soit aligné avec les bords gauche et droit de l'espace à remplir.

Par exemple, avec

```
\titleformat{\section}[leftmargin]
{\titlerule*[1pc]{.}%
\vspace{1ex}%
\bfseries}
{... définitions à la suite
```

des **✖ leaders ✖** s'étendant au-dessus du texte principal et du titre précédent la section.

`calwidth` (option d'extension)

La forme `wrap` a la capacité de mesurer les lignes du titre pour mettre en forme le paragraphe. Cette capacité peut être étendue à trois autres formes — `display`, `block` et `hang` — avec cette option d'extension. La longueur de la plus longue ligne est restituée dans `\titlewidth`<sup>14</sup>.

Du point de vue de  $\TeX$ , toute boîte est considérée comme du matériel composable. Si une boîte a été élargie avec de l'espace ou, inversement, si une boîte avec du texte a été réduite, la valeur de `\titlewidth` peut être faussée (du point de vue d'un humain). La forme `hang`, par exemple, recourt en interne à de telles boîtes mais, dans ce cas, ce comportement est souhaité quand le titre est aligné à droite ; sinon la forme `block` produit de meilleurs résultats. En d'autres termes, l'utilisation de boîtes dont la largeur a été manipulée peut être source d'erreurs<sup>15</sup>. Pour être complet, certaines commandes peuvent embrouiller  $\TeX$  et interrompre l'analyse du titre. Mais, si vous vous en tenez à du texte, `\\` et `\\[...]` (et il y a peu de chance que vous puissiez avoir besoin d'autre chose), vous ne devriez pas rencontrer de problèmes.

Un autre point important est que les paramètres `<code-avant>`, `<num>`, `<sep>`, et `<titre>` (mais pas `<code-après>`) sont évalués deux fois au niveau local ; si vous augmentez un compteur *globalement*, vous l'augmenterez alors deux fois. Dans la plupart des cas, placer une **✖ assignation conflictuelle ✖** dans le `<code-après>` passera mais vous pouvez également passer la commande suivante.

`\iftitlemeasuring{<vrai>}{<faux>}`

**2.9** Quand le titre est mesuré (première passe), l'argument `<vrai>` est traité tandis que, quand le titre est réellement composé (seconde passe), l'argument `<faux>` est traité.

`\titleline*[{<alignement>}]{<matériel horizontal>}`

Une variante de `\titleline` à utiliser uniquement avec `calwidth`. Le texte sera d'abord placé dans une boîte de largeur `\titlewidth` ; cette boîte sera à son tour placée dans la boîte principale avec

14. Il existe deux paramètres plus fins `\titlewidthfirst` and `\titlewidthlast` qui restituent la longueur de la première et de la dernière ligne respectivement. Il n'y a pas de fonctions dédiées pour s'en servir mais vous pouvez assigner leurs valeurs à `\titlewidth` et utiliser alors `\titleline*`.

15. Ce qui inclut les lignes justifiées dont l'espace **✖ inter-mot ✖** a été augmenté.

l'alignement indiqué. Il n'y a pas d'équivalent `\titlerule` et ceci explique pourquoi vous devez le placer explicitement dans un `\titleline*` si vous voulez que `\titlewidth` soit pris en compte :

```
\titleline*[c]{\titlerule[.8pc]}
```

### 3.5. Styles de page

**2.8** Vous pouvez assigner un style de page à des **✖ niveaux de classe ✖** `top` ou `page`, aussi bien qu'au chapitre de base avec la commande suivante <sup>16</sup> :

```
\assignpagestyle{<commande>}{<style de page>}
```

Par exemple, pour supprimer le numéro de page avec les chapitres :

```
\assignpagestyle{\chapter}{empty}
```

### 3.6. Sauts de page

<code>\sectionbreak</code>	<code>\subsectionbreak</code>	<code>\subsubsectionbreak</code>
<code>\paragraphbreak</code>	<code>\subparagraphbreak</code>	<code>\&lt;section&gt;break</code>

En définissant ces commandes avec `\newcommand`, différents saut de page peuvent être appliqués à différents niveaux hiérarchiques. Dans celles non définies, une pénalité avec une valeur interne fournie par la classe est retenue (typiquement  $-300$ ). Par exemple,

```
\newcommand{\sectionbreak}{\clearpage}
```

fait que les sections commencent sur une nouvelle page. Dans certaines mises en page, l'espace au-dessus du titre est préservé même si la section commence sur une nouvelle page ; ceci est accompli avec :

```
\newcommand{\sectionbreak}{%
  \addpenalty{-300}%
  \vspace*{0pt}}
```

**2.6** `\<section>break` est également disponible avec la classe **✖ top ✖**. Les valeurs admises sont `\cleardoublepage` (valeur par défaut avec `openright`) et `\clearpage` (valeur par défaut avec `openany`). Ainsi, vous pouvez annuler l'effet de `openright` en définissant `\chapterbreak` comme `\clearpage`, du moment que sa classe a été changée en `top` (dans cet exemple, les parties conserveront le paramétrage `openright`).

```
\chaptertolists
```

**2.6** Si cette commande est définie, **✖ l'espace blanc écrit d'habitude dans les listes ✖** (que sont la liste des figures et la liste des tables) est remplacé par le code de cette commande. Si vous ne voulez pas de l'espace quand un chapitre commence, définissez-le comme vide, autrement dit,

```
\newcommand{\chaptertolists}{}%
```

Cette commande n'est pas un outil général pour contrôler l'espacement dans les listes et n'est valable que dans les titres de classe **✖ top ✖**. Aussi, elle ne fonctionnera pas avec les chapitres **✖ de base ✖** sauf si vous changez leur classe (d'un autre côté, `\...tolists` peut être utilisé dans tout titre dans la classe est **✖ top ✖**).

<sup>16</sup>. Nommée dans l'éphémère version 2.7 `\titlepagestyle`.



### 3.7. Autres options de l'extension

`explicit` (option d'extension)

**2.7** Avec cette option, le titre n'est pas implicite à la fin de *code-avant* mais doit être indiqué explicitement avec #1 comme dans, par exemple :

```
\titleformat{\section}
{..}
{\thesection}{..}{#1.}
```

(Comparez ceci avec l'exemple de la section 4.4.)

`newparttoc oldparttoc` (options d'extension)

Les parties standards écrivent leur numéro en table des matières de façon non standard. Vous pouvez changer cela avec `newparttoc` de façon à ce que `titletoc` ou une extension similaire puisse manipuler cette entrée. Standard parts write the toc entry number in a non standard way (ceci fonctionne dans le cas où `\part` a été redéfini).

`cleareempty` (option d'extension)

Modifie le comportement de `\cleardoublepage` de sorte que le style de page `empty` soit utilisé dans les pages vides.

`toctitles` (option d'extension)

**2.6** Change le comportement de l'argument optionnel dans les commandes de sectionnement de telle manière qu'elles fixent uniquement le titre en en-tête et non l'entrée en table des matières qui sera donc basée sur le titre complet.

`newlinetospace` (option d'extension)

**2.6** Remplace chaque occurrence de `\\` ou `\\*` dans les titres par un espace dans les en-têtes et les entrées en table des matières. De cette façon, vous n'avez pas à répéter le titre juste pour retrancher une commande de mise en forme.

### 3.8. Paramétrages étendus

✱The first argument of both `\titleformat` and `\titlespacing` has an extended syntax which allows to set different formats depending on the context.<sup>17</sup> This argument can be a list of key/value pairs in the form :

`<key>=<value>, <key>=<value>, <key>, <key>, ...`

Currently, only pages and unnumbered versions are taken care of, besides the sectioning command name. Thus, the available keys are :

- `name`. Allowed values are `\chapter`, `\section`, etc.
- `page`. Allowed values are `odd` or `even`.
- `numberless`. A valueless key, it is not necessary unless you want to set different numbered (without this key) and unnumbered (with `numberless`) variants.

The basic form described above with the name of a sectioning command, say

```
\titleformat{\section} ...
```

is in fact an abbreviation for

<sup>17</sup>. The `keyval` package is required for making use of it.

```
\titleformat{name=\section} ...
```

Let's suppose we'd like a layout with titles in the outer margin. We might set something like

```
\titleformat{name=\section,page=even}[leftmargin]
{\filleft\scshape}{\thesection}{.5em}{}

\titleformat{name=\section,page=odd}[rightmargin]
{\filright\scshape}{\thesection}{.5em}{}


```

Since the page information is written to the aux file, at least two runs are necessary to get the desired result.

The “number” version is usually fine when generating unnumbered variants since removing the label is the only change required in most cases, but if you need some special formatting, there is the `numberless` key which defines an alternative version for sections without numbers (namely those with level below `secnumdepth`, in the front and back matters and, of course, the starred version). For instance

```
\titleformat{name=\section}{...% The normal definition follows
\titleformat{name=\section,numberless}{...% The unnumbered
% definition follows


```

Neither `<label>` nor `<sep>` are ignored in numberless variants.

These keys are available to both `\titleformat` and `\titlespacing`. By using `page` in one (or both) of them, odd and even pages will be formatted differently. Actually, “even” and “odd” are well established L<sup>A</sup>T<sub>E</sub>X terms, but misleading. In one side printing the “odd” pages refer to “even” pages as well (cf. `\oddsidemargin`).

If you intend to create different odd/even *and* different numbered/unnumbered versions, it is recommended defining the four variants.

If you remove the page specifier from a sectioning command you must remove the `.aux` file.

### 3.9. Creating new levels and changing the class

While the shapes and the like modify the behaviour of titles related to the surrounding text, title classes allow to change the generic behaviour of them. With the help of classes you may insert, say, a new subchapter level between chapter and section, or creating a scheme of your own. *Making a consistent scheme and defining all of related stuff like counters, macros, format, spacing and, if there is a TOC, TOC format is left to the responsibility of the user.* There are three classes : `page` is like the book `\part`, in a single page, `top` is like `\chapter`, which begins a page and places the title at the top, and `straight` is intended for titles in the middle of text.<sup>18</sup>

```
\titleclass{<name>}{<class>}
\titleclass{<name>}{<class>}[<super-level-cmd>]
```

If you do not use the optional argument, you just change the `<class>` of `<name>`. For example :

```
\titleclass{\part}{straight}
```

makes `part` of `straight` class.

When the second form is used, the level number is the following of `<super-level-cmd>`. For example :

```
\titleclass{\subchapter}{straight}[\chapter]
\newcounter{subchapter}
\renewcommand{\thesubchapter}{\Alph{subchapter}}
```

creates a level under chapter (some additional code is shown as well, but you must add to it the corresponding `\titleformat` and `\titlespacing` settings).<sup>19</sup> If the chapter level is 0, then the subchapter one is 1 ; the levels below are increased by one (section is 2, subsection is 3, and so on).

There are two sectioning commands which perform some extra actions depending of its name and ignoring the class :

<sup>18</sup>. There is an further class named `part` to emulate the article `\part`, but you should not use it at all. Use the `straight` class instead. Remember some features rely in these classes and `titlesec` does not change by default the definition of `\part` and `\chapter`.

<sup>19</sup>. Regarding counters, the `remreset` package can be useful.

- `\chapter` logs the string defined in `\chaptertitlename` and the matter is taken into account.
- `\part` does not encapsulates the label in the toc entry, except if you use the `newparttoc` option.

`loadonly` (Package option)

Let us suppose you want to create your sectioning commands from scratch. This package option ignores any previous definitions, if any, and hence removes the possibility of using the options described in “Quick Reference.” Then you use the `titlesec` tools, and define the corresponding counters and labels.

`\titleclass{<name>}[<start-level-num>]{<class>}`

Here, the `<name>` title is considered the top level, with number `<start-level-num>` (typically 0 or  $-1$ ). It should be used only when creating sectioning commands from scratch with the help of `loadonly`, and there must be exactly one (no more, no less) declaration of this kind. After it, the rest of levels are added as explained above.

## 4. Additional Notes

This part describes briefly some  $\text{\LaTeX}$  commands, useful when defining sectioning titles.

### 4.1. Fixed Width Labels

The `\makebox` command allows to use fixed width label, which makes the left margin of the actual title (not the label) to lie in the same place. For instance (only the relevant code is provided) :

```
\titleformat{\section}
{..}
{\makebox[2em]{\thesection}}{..}{..}
```

See your  $\text{\LaTeX}$  manual for further reference on boxing commands.

### 4.2. Starred Versions

Using sectioning commands in the starred version is strongly discouraged. Instead, you can use a set of markup oriented commands which are easy to define and modify, if necessary. Thus, you can test different layouts before choosing amongst them.

Firstly remember if you say

```
\setcounter{secnumdepth}{0}
```

sections will be not numbered but they will be included in both toc and headers.

Now, let's suppose you want to include some sections with a special content ; for example, a section (or more) with exercises. We will use an environment named `exercises` whose usage is :

```
\section{A section}
Text of a normal section.

\begin{exercises}
\section{Exercises A}
Some exercises

\section{Exercises B}
Some exercises
\end{exercises}
```

The following definition suppresses numbers but neither toc lines nor headers.

```
\newenvironment{exercises}
{\setcounter{secnumdepth}{0}}
{\setcounter{secnumdepth}{2}}
```

The following one adds a toc line but headers will remain untouched :

```
\newenvironment{exercises}
{\setcounter{secnumdepth}{0}%
 \renewcommand\sectionmark[1]{}
 \setcounter{secnumdepth}{2}}
```

The following one updates the headers but there will be no toc line :

```
\newenvironment{exercises}
{\setcounter{secnumdepth}{0}%
 \addtocontents{toc}{\protect\setcounter{tocdepth}{0}\ignorespaces}}
{\setcounter{secnumdepth}{2}%
 \addtocontents{toc}{\protect\setcounter{tocdepth}{2}\ignorespaces}}
```

(I find the latter a bit odd in this particular example ; the first and second options are more sensible. The `\ignorespaces` is not very important, and you need not it unless there is unwanted space in the toc.)

That works with standard classes, but if you are using `fancyhdr` or `titlesec` to define headers you need further refinement to kill the section number. In `titlesec` that's accomplished with `\ifthesection` (see below).

As you can see, there are no `\addcontentsline`, no `\markboth`, no `\section*`, just logical structure. Of course you may change it as you wish ; for example if you decide these sections should be typeset in small typeface, include `\small`, and if you realize you do not like that, remove it.

While the standard  $\LaTeX$  commands are easier and more direct for simple cases, I think the proposed method above is far preferable in large documents.

### 4.3. Variants

Let's suppose we want to mark some sections as “advanced topics” with an asterisk after the label. The following code does the job :

```
\newcommand{\secmark}{}
\newenvironment{advanced}
{\renewcommand{\secmark}{*}}
{}
\titleformat{\section}
{..}
{\thesection\secmark\quad}{..}{..}
```

To mark the sections write

```
\begin{advanced}
\section{...}
...
\end{advanced}
```

That marks sections but not subsections. If you like being redundant and marking the subsection level as well, you must define it accordingly.

### 4.4. Putting a Dot after the Section Title

Today this styling is not used, but formerly it was fairly common. The basic technique was described above, but here is a reminder :

```
\newcommand{\periodafter}[1]{#1.}
\titleformat{\section}
{..}
{\thesection}{..}{..\periodafter}
```

If you had to combine this dot with some command (perhaps an underlining), you can say :

```
\newcommand{\periodafter}[2]{#1{#2.}}
\titleformat{\section}
{..}
{\thesection}{..}{..\periodafter{\ul}} % \ul from soul package
```

However, you might prefer the package option `explicit`.

## 5. titleps and Page Styles

The titleps package provides tools for one-stage setting of page styles (headlines and footlines). A higher-level interface is used, where the mark mechanism is hidden and there is no need to deal with `\leftmarks` and `\rightmarks`—just use a command or variable registered as a “mark” as the expected value will be returned, i.e., those when the mark was emitted, either by a sectioning command or explicitly with `\chaptermark`, `\sectionmark`, etc. A simple example, whose meaning should be obvious, is :

```
\newpagestyle{main}{
  \sethead[\thepage][\chaptertitle][(\thesection] % even
    {\thesection)}{\sectiontitle}{\thepage}} % odd
\pagestyle{main}
```

Other features are :

- Working top marks, compatible with floats (unlike the standard `\topmark`, which does not work correctly in  $\LaTeX$ ).
- Access to top, first and bot marks in a single headline/footline (e.g., the first and last section numbers).
- Marks for more than 2 sectioning levels.
- Simple (and not so simple) headrules and footrules.
- Headlines and footlines for pages with floats.
- Headlines and footlines for specific floats (a sort of `\thispagestyle` for floats).
- Multiple sets of marks (named here *marksets* and *extra marks*).

It can be used without titlesec, but you will get most of it when used together. To load it as a separate package, use the customary `\usepackage{titleps}`, but with titlesec you have to load it with :

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

Please, read `titleps.pdf` (or `typeset titleps.tex`) for further information.

## 6. Contents : The titletoc package

This package is a companion to the titlesec package and it handles toc entries. However, it is an independent package and you can use it alone. The philosophy is similar to that of titlesec—instead of hooking the commands as defined by standard  $\LaTeX$  and classes, there are new commands which you can format the toc entries with in a generic way. This means you have to learn just two new basic command and a couple of tools, no more, and you have access to new features. Paragraph format and fonts are set with commands like `\l`, `\makebox`, `\large`, `\itshape`, and so on, and entries are not shaped in any fashion because they are pretty free form.

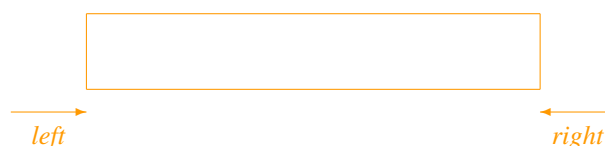
The behaviour of entries defined with titletoc are different at some points to those created with the standard commands. In particular :

- Pages are never broken between entries if the first one is of an higher level than the second one as, for instance, between a section and a subsection. If both of them are of the same level, the break is allowed, and if the first is lower than the second, it is considered a good place for a page break.
- The symbols in the leaders are not centered but flushed right. That is usually more convenient.

I would like to note no attempt to handle tocs can be complete because the standard  $\LaTeX$  commands write directly some formatting commands which cannot be removed. This is particularly important in lists of figures and tables, and in the `\part` command.<sup>20</sup>

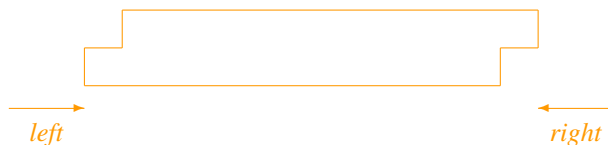
### 6.1. A ten-minute guide to titletoc

Toc entries are treated as rectangular areas where the text and probably a filler will be written. Let’s draw such an area (of course, the lines themselves are not printed) :

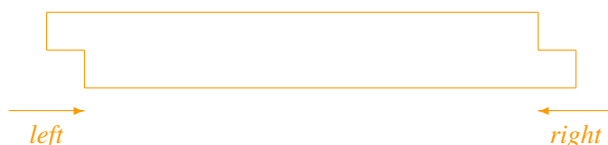


<sup>20</sup>. But some of these issues are fixed by titlesec.

The space between the left page margin and the left edge of the area will be named  $\langle left \rangle$ ; similarly we have  $\langle right \rangle$ . You are allowed to modify the beginning of the first line and the ending of the last line. For example by “taking up” both places with `\hspace*{2pc}` the area becomes :



And by “clearing” space in both places with `\hspace*{-2pc}` the area becomes :



If you have seen tocs, the latter should be familiar to you– the label at the very beginning, the page at the very end :

```
3.2 This is an example showing that toc
    entries fits in that scheme . . . . 4
```

All you need is to put these elements in the right way. If you have reserved the space with `\hspace*{-2pc}`, simply put a box 2 pc width containing the section label or page so that this space will be retrieved ; this layout is used so often that two commands are provided which does that for you :

- `\contentslabel{<length>}` creates the space at the beginning and prints the section number.
- `\contentspage` creates a space at the end of length  $\langle right \rangle$  and prints the page number aligned at the right.

Now, we are about to show the three basic commands :

```
\dottedcontents{<section>}[<left>]{<above-code>}
                    {<label width>}{<leader width>}
```

Here :

- $\langle section \rangle$  is the section name without backslash : part, chapter, section, etc. figure and table are allowed, too. (The backslash is omitted because we are dealing with the concept and not the `\part`, `\section`, etc. macros themselves. Furthermore, figure and table are environments.)
- $\langle above-code \rangle$  is code for the global formatting of the entry. Vertical material is allowed. At this point the value of `\thecontentslabel` (see below) is known which enables you to take decisions depending on its value (with the help of the `ifthen` package). You may use the `titlesec \filleft`, `\filright`, `\filcenter` and `\fillast` commands.
- $\langle left \rangle$  even if bracketed is currently mandatory and it sets the left margin from the left page margin.
- $\langle label width \rangle$  is the width of the space created for the label, as described above.
- $\langle leader width \rangle$  is the width of the box containing the char to be used as filler, as described below.

The definitions for section and subsection entries in the book class are roughly equivalent to :

```
\contentsmargin{2.55em}
\dottedcontents{section}[3.8em]{}{2.3em}{1pc}
\dottedcontents{subsection}[6.1em]{}{3.2em}{1pc}
```

```
\titlecontents{<section>}[<left>]{<above-code>}
                    {<numbered-entry-format>}{<numberless-entry-format>}
                    {<filler-page-format>}[<below-code>]
```

Here  $\langle section \rangle$ ,  $\langle left \rangle$  and  $\langle above-code \rangle$  like above, and

- `<numbered-entry-format>` is in horizontal mode and it will be used just before the entry title. As in `\titleformat`, the last command can take an argument with the title.
- `<numberless-entry-format>` is like the above if there is, well, no label.
- `<filler-page-format>` is self explanatory. Fillers are created with the `\titlerule` command which is shared by that package and titlesec. However, when used in this context its behaviour changes a little to fit the needs of toc leaders.<sup>21</sup> You might prefer a `\hspace` instead.
- And finally `<below-code>` is code following the entry for, say, vertical space.

When defining entries, use `\addvspace` if you want to add vertical space, and `\\*` instead of `\\` for line breaks.

This command can be used in the middle of a document to change the format of toc/lot/lof entries at any point. The new format is written to the toc file and hence two runs are necessary to see the changes.

```
\contentsmargin{<right>}
```

The value set is used in all of sections. If you are wondering why, the answer is quite simple : in most of cases the `<right>` margin will be constant. However, you are allowed to change it locally in the `<before-code>` arguments. Note as well that the default space in standard classes does not leave room to display boldfaced page number above 100 and therefore you might want to set a larger margin with this command.

The book class formats section entries similarly (but not equally) to :

```
\titlecontents{section}
    [3.8em] % ie, 1.5em (chapter) + 2.3em
    {}
    {\contentslabel{2.3em}}
    {\hspace*{-2.3em}}
    {\titlerule*[1pc]{.}\contentspage}
```

Compare this definition with that given above and you will understand how `\dottedcontents` is defined.

Although standard classes use font dependent units (mainly em), it is recommended using absolute ones (pc, pt, etc.) to ensure they entries are aligned correctly.

## 6.2. And more

Strict typographical rules state full text lines shouldn't surpass the last dot of the leaders ; ideally they should be aligned. Surprisingly enough,  $\TeX$  lacks of a tool for doing that automatically—when you fill a box with leading dots, they can be centered in the box with the `\cleaders` primitive , “justified” with `\xleaders` or aligned with the outermost enclosing box with `\leaders`, but there is no way to align them with the “current” margin.

So, the only way to get a fine layout is by hand. To do , you can use the an optional argument in the `\contentsmargin` command whose syntax in full is the following :

```
\contentsmargin[<correction>]{<right>}
```

The `<correction>` length is added to the `<right>` one in all of lines except the last one, where the leaders are placed. For instance, if the text lines are 6 pt longer than the last dot, you should rewrite the `\contentsmargin` command to add a `<correction>` of 6 pt.<sup>22</sup> Unlike the standard  $\LaTeX$  tools, the `\titlerule*` command has been designed so that the `<correction>` will have the minimum value possible.

```
\thecontentslabel \thecontentspage
```

Contains the text with the label and the page with no additional formatting, except written by the class.

21. For  $\TeX$ nicians, the default `\xleaders` becomes `\leaders`.

22. Usefully, many dvi previewers allow to get the coordinates of the pointed location.

```
\contentslabel[⟨format⟩]{⟨space⟩}
\contentspage[⟨format⟩]
```

As described above, but with different *⟨format⟩*s. The defaults are just `\thecontentslabel` and `\thecontentspage`, respectively.

```
\contentspush{⟨text⟩}
```

Prints the *⟨text⟩* and increases *⟨left⟩* by the width of *⟨text⟩*. It is similar to the hang shape of `titlesec`.

```
\titlecontents*{⟨section⟩}[⟨left⟩]{⟨above-code⟩}
    {⟨numbered-entry-format⟩}{⟨numberless-entry-format⟩}
    {⟨filler-page-format⟩}[⟨separator⟩]
    or ...{⟨filler-page-format⟩}[⟨separator⟩][⟨end⟩]
    or ...{⟨filler-page-format⟩}[⟨begin⟩][⟨separator⟩][⟨end⟩]
```

This starred version groups the entries in a single paragraph. The *⟨separator⟩* argument is the separator between entries, and there is a further optional argument with an ending punctuation. For example, this document sets :

```
\titlecontents*{subsection}[1.5em]
    {\small}
    {\thecontentslabel. }
    {}
    {, \thecontentspage}
    [.-.-][.]
```

whose result is showed in the contents at the very beginning of this document. Note the paragraph format must be written in the *⟨above-code⟩* argument.

Let us explain how the optional arguments works. First note the number of them determines their meaning—since there should be a separator between entries this one is always present ; on the other hand, *⟨begin⟩* is rarely used and hence it has the lowest “preference.” The simplest case is when the titles are of the same level ; in this case the *⟨separator⟩* and the *⟨end⟩* parameters (which default to empty) are inserted between consecutive entries and at the end of the block, respectively. *⟨before-code⟩* is executed just once at the very beginning of the block and its declarations are local to the whole set of entries.

Now suppose we want to group entries of two levels ; in this case a nesting principle applies. To fix ideas, we will use section and subsection. When a subsection entry begins after a section one, *⟨before-code⟩* is executed and *⟨begin⟩* of subsection is inserted, which should contain text format only. Then subsections are added inserting separators as explained above. When a section arrives, the ending punctuation of subsection and the separator of section is added (except if the block is finished by a subsection, where the ending of section is added instead). We said “after a section” because a subsection never begins a block.<sup>23</sup> The subsection entries are nested inside the section ones, and declarations are again local.

An example will illustrate that.

```
\titlecontents*{section}[0pt]
    {\small\itshape}{}{}
    {}[ \textbullet\ ]{.}

\titlecontents*{subsection}[0pt]
    {\upshape}{}{}
    {, \thecontentspage}[ (][. ][]]
```

produces something similar to :

*The first section • The second one • The third one* (A subsection in it, 1. Another, 2) • *A fourth section* (A subsection in it, 1. Another, 2).

---

23. In rare cases that could be necessary, yet.



```
\contentsuse{<name>}{<ext>}
```

Makes titletoc aware of the existence of a contents file with *<ext>* extension. Mainly, it makes sure the command `\contentsfinish` is added at the end of the corresponding contents (and which must be added at the end of tocs made by hand). The package performs

```
\contentsuse{figure}{lof}
\contentsuse{table}{lot}
```

```
leftlabels rightlabels (Package options)
```

These package options set how the labels are aligned in `\contentslabel`. Default is `rightlabels`. With `leftlabels` the default *<format>* for `\contentslabel` becomes `\thecontentstlabel\enspace`.

```
dotinlabels (Package option)
```

With this package option, a dot is added after the label in `\contentslabel`.

### 6.3. Partial TOC's

```
\startcontents[<name>]
```

At the point where this command is used, a partial toc begins (note the document doesn't require a `\tableofcontents` for partial tocs to work). The *<name>* argument allows different sets of tocs and it defaults to `default`. These sets may be intermingled, but usually will be nested. For example, you may want two kinds of partial tocs : by part and by chapter (besides the full toc, of course). When a part begins, write `\startcontents[parts]`, and when a chapter `\startcontents[chapters]`. This way a new toc is started at each part and chapter.<sup>24</sup>

```
\stopcontents[<name>]
\resumecontents[<name>]
```

Stops the partial toc of *<name>* kind, which may be resumed. Since partial contents are stopped by `\startcontents` if necessary, those macros will not be used very often.

```
\printcontents[<name>]{<prefix>}{<start-level>}{<toc-code>}
```

Print the current partial toc of *<name>* kind. The format of the main toc entries are used, except if there is a *<prefix>*. In such a case, the format of *<prefix><level>* is used, provided it is defined. For example, if prefix is `l` and the format of `lsection` is defined, then this definition will be used ; otherwise, the format is that of `section`. The *<start-level>* parameter sets the top level of the tocs—for a part toc it would be 0 (chapter), for a chapter toc 1 (section), and so on. Finally, *<toc-code>* is local code for the current toc ; it may be used to change the `tocdepth` value or `\contentsmargin`, for instance.

A simple usage might look like (provided you are using titlesec as well) :

```
\titleformat{\chapter}[display]
{...}{...}{...} % Your definitions come here
[\vspace*{4pc}%
\startcontents
\printcontents{l}{1}{\setcounter{tocdepth}{2}}]

\titlecontents{lsection}[0pt]
{\small\itshape}{}{}
{}[ \textbullet\ ]{.}
```

The included entries are those in levels 1 to 2 inclusive (i.e., 1 and 2).

---

24. All partial tocs are stored in a single file with extension `.ptc`.

## 6.4. Partial lists 2.6

You may want to create partial LOFs and LOTs. The syntax is similar to that of partial TOCs and what was said for them can be applied here. The commands are :

```
\startlist[⟨name⟩]{⟨list⟩}
\stoplist[⟨name⟩]{⟨list⟩}
\resumelist[⟨name⟩]{⟨list⟩}
\printlist[⟨name⟩]{⟨list⟩}{⟨prefix⟩}{⟨toc-code⟩}
```

Here *⟨list⟩* is either lof or lot. Note as well `\printlist` does not have the *⟨start-level⟩* argument, because figures and tables have not levels. Currently, only those two float lists are supported, but in a future release support for more kinds of float lists will be added. Unfortunately, many classes write some formatting commands to these lists (more precisely, `\addvspace` in chapters) ; I'm still not sure how to remove these commands without removing as well others which can be wanted, but for the time being a quick trick to remove these spaces is to redefine `\addvspace` in the *⟨toc-code⟩* with `\renewcommand\addvspace[1]{}.`

## 6.5. Examples

```
\titlecontents{chapter}
[0pt]
{\addvspace{1pc}%
\itshape}%
{\contentsmargin{0pt}%
\bfseries
\makebox[0pt][r]{\huge\thecontentslabel\enspace}%
\large}
{\contentsmargin{0pt}%
\large}
{\quad\thepage}
[\addvspace{.5pc}]
```

The chapter number is out at the edge of the page margin, in a font larger than the font of the title. If the chapter lacks of number (because, say, it is the preface or the bibliography) it is not boldfaced. The page number follows the title without fillers, but after an em-space.

```
\titlecontents{chapter}
[3pc]
{\addvspace{1.5pc}%
\filcenter}
{CHAPTER \thecontentslabel\*\*.2pc}%
\huge}
{\huge}
{} % That is, without page number
[\addvspace{.5pc}]
```

The chapter title is centered with the chapter label on top of it. There is no page number.

## 6.6. Inserting a figure in the contents

The `\addtocontents` command is still available and you may use it to perform special operation, like inserting a figure just before or after of an entry. Sadly, fragile arguments are not allowed and writing complex code could be a mess. The trick is to define a command to perform the required operations which in turn is written with `\protect`.

Let's suppose we want to insert a figure before an entry.

```
\newcommand{\figureintoc}[1]{
\begin{figure}
\includegraphics{#1}%
\end{figure}}
```

makes the dirty work.

In the place where a figure is inserted write :

```
\addtocontents{\protect\figureintoc{myfig}}
```

## 6.7. Marking entries with asterisks

Let's now resume a problem explained in relation with titlesec : marking sections with asterisks to denote an “advanced topic” unless the star should be printed in the toc as well. Here is the code :

```
\newcommand{\secmark}{}
\newcommand{\marktoc}[1]{\renewcommand{\secmark}{#1}}
\newenvironment{advanced}
  {\renewcommand{\secmark}{*}%
   \addtocontents{toc}{\protect\marktoc{*}}}
  {\addtocontents{toc}{\protect\marktoc{}}}
\titleformat{\section}
  {...}
  {\thesection\secmark}{...}{...}
\titlecontents{section}[...]{...}
  {\contentslabel[\thecontentslabel\secmark]{1.5pc}}{...}{...}
```



## 7. La philosophie de titlesec

Une fois que vous avez lu cette documentation, il devrait être clair que cette extension n'est pas destinée à l'utilisateur occasionnel qui aime la présentation classique et souhaite juste y apporter quelques modifications simples. Il s'agit ici d'un outil pour le typographe sérieux qui a une idée claire de la présentation souhaitée et qui n'avait auparavant pas les moyens pour l'obtenir. Rien n'est fait non plus pour améliorer votre goût en matière de mise en forme des sectionnements.

## 8. Annexes

Les exemples suivants servent d'illustrations. Dans cette partie, \parskip est fixé à 0 pt.



**9 Ceci est un exemple de la commande de section définie ci-dessous et, je dirais même plus, ceci est un exemple de la commande de section définie ci-dessous**

```
\titleformat{\section}[block]
  {\normalfont\bfseries\filcenter}{\fbox{\itshape\thesection}}{1em}{}
```



SECTION 10

## Un titre encadré

```
\titleformat{\section}[frame]
  {\normalfont}
  {\filright
   \footnotesize
   \enspace SECTION \thesection\enspace}
  {8pt}
  {\Large\bfseries\filcenter}
```



## 11. Un titre avec filet horizontal

```
\titleformat{\section}
  {\titlerule
   \vspace{.8ex}%
   \normalfont\itshape}
  {\thesection.}{.5em}{}
```



12

### Un autre titre avec filet horizontal

```
\titleformat{\section}[block]
  {\normalfont\sffamily}
  {\thesection}{.5em}{\titlerule\[\.8ex\bfseries}
```



.....

13 La largeur du filet ci-dessus est celle de la plus longue  
ligne de ce titre augmentée de deux picas

.....

14 Celui-ci est plus court

```
\titleformat{\section}[block]
{\filcenter\large
\addtolength{\titlewidth}{2pc}%
\titeline*[c]{\titlerule*{.6pc}{\tiny\textbullet}}}%
\addvspace{6pt}%
\normalfont\sffamily}
{\thesection}{1em}{}
\titlespacing{\section}
{5pc}{*2}{*2}[5pc]
```



## SECTION 15

Ceci est un exemple de la commande de section définie ci-dessous et, je dirais même plus, ceci est un exemple de la commande de section définie ci-dessous. Répétons-le : ceci est un exemple de la commande de section définie ci-dessous et, je dirais même plus, ceci est un exemple de la commande de section définie ci-dessous

```
\titleformat{\section}[display]
{\normalfont\filllast}
{\scshape section \oldstylenums{\thesection}}
{\lex minus .1\ex}
{\small}
\titlespacing{\section}
{3pc}{*3}{*2}{3pc}
```



CETTE PARTIE EST LE TITRE LUI-MÊME et cette partie est le corps du texte associé...

```
\titleformat{\section}[runin]
{\normalfont\scshape}
{}{0pt}{}
\titlespacing{\section}
{\parindent}{*2}{\wordsep}
```



## 16. Un exemple simple d'une forme « wrap » sur une section

**16. Un exemple simple d'une forme « wrap » sur une section**

Suivi d'un peu de texte pour montrer le résultat. Suivi d'un peu de texte pour montrer le résultat. Suivi d'un peu de texte pour montrer le résultat. Suivi d'un peu de texte pour montrer le résultat. Suivi d'un peu de texte pour montrer le résultat.

**17. Et un autre** Notez comment le texte enveloppe (en anglais, « wrap ») le titre et que l’espace qui lui est réservé est automatiquement réajustée. Et il est suivi par un peu de texte pour illustrer le propos. Et il est suivi par un peu de texte pour illustrer le propos.

```
\titleformat{\section}[wrap]
  {\normalfont\fontseries{b}\selectfont\filright}
  {\thesection.}{.5em}{}
\titlespacing{\section}
  {12pc}{1.5ex plus .1ex minus .2ex}{1pc}
```

■

**§ 18. Un titre attaché à l’ancienne.**—Bien sûr, vous pourriez préférer juste un point après le titre. Dans ce cas, l’argument optionnel devrait être [. ] et l’espace attenante devrait avoir une valeur assez forte (1em, par exemple).

```
\titleformat{\section}[runin]
  {\normalfont\bfseries}
  {\S\ \thesection.}{.5em}{}[.---]
\titlespacing{\section}
  {\parindent}{1.5ex plus .1ex minus .2ex}{0pt}
```

■

**Exemple  
de section  
marginale**

Suivi d’un peu de texte pour montrer le résultat. Mais ne vous arrêtez pas de lire car l’exemple qui suit illustre comment tirer parti d’autres extensions. La dernière commande dans le dernier argument peut prendre un argument, à savoir le titre sans autre commande additionnelle incluse. Nous donnons ici juste le code mais vous pouvez l’essayer. Donc, avec l’extension soul, vous pourriez écrire

```
\newcommand{\secformat}[1]{\MakeLowercase{\so{#1}}}%
% \so spaces out letters
\titleformat{\section}[block]
  {\normalfont\scshape\filcenter}
  {\thesection}
  {1em}
  {\secformat}
```

Le titre marginal ci-dessus a été défini avec le code suivant :

```
\titleformat{\section}[leftmargin]
  {\normalfont
    \titlerule*{.6em}{\bfseries.}%
    \vspace{6pt}%
    \sffamily\bfseries\filleft}
  {\thesection}{.5em}{}
\titlespacing{\section}
  {4pc}{1.5ex plus .1ex minus .2ex}{1pc}
```

■

Les exemples suivants sont pensés pour des chapitres. Cependant, cette documentation ne dispose par de la commande \chapter et illustre ce point en utilisant \section avec quelques rares changements.

## CHAPTER 19

# Le titre

```
\titleformat{\chapter}[display]
```

```

{\normalfont\Large\filcenter\sffamily}
{\titlerule[1pt]}%
\vspace{1pt}%
\titlerule
\vspace{1pc}%
\LARGE\MakeUppercase{\chaptertitlename} \thechapter}
{1pc}
{\titlerule
\vspace{1pc}%
\Huge}

```

■

## CHAPTER XX

---

### Le titre

---

```

\renewcommand{\thechapter}{\Roman{chapter}}
\titleformat{\chapter}[display]
{\bfseries\Large}
{\filleft\MakeUppercase{\chaptertitlename} \Huge\thechapter}
{4ex}
{\titlerule
\vspace{2ex}%
\filright}
[\vspace{2ex}%
\titlerule]

```

### 9.1. Un exemple complet

Voici maintenant un exemple complet d'une structure de titres.

```

\documentclass[twoside]{report}
\usepackage[sf,sl,outermarks]{titlesec}

% \chapter, \subsection...: pas de code additionnel

\titleformat{\section}
{\LARGE\sffamily\slshape}
{\thesection}{1em}{}
\titlespacing{\section}
{-6pc}{3.5ex plus .1ex minus .2ex}{1.5ex minus .1ex}

\titleformat{\paragraph}[leftmargin]
{\sffamily\slshape\filright}
{}{}{}
\titlespacing{\paragraph}
{5pc}{1.5ex minus .1 ex}{1pc}

% 5+1=6, autrement dit, la marge négative de gauche dans
% la section

```

```

\widenhead{6pc}{0pc}

\renewpagestyle{plain}{}

\newpagestyle{special}[\small\sffamily]{
  \headrule
  \sethead[\usepage][\textsl{\chaptertitle}][
    {}{\textsl{\chaptertitle}}{\usepage}]

\newpagestyle{main}[\small\sffamily]{
  \headrule
  \sethead[\usepage][\textsl{\thechapter. \chaptertitle}][
    {}{\textsl{\thesection. \sectiontitle}}{\usepage}]

\pagestyle{special}

\begin{document}

---Table des matières

\pagestyle{main}

---Corps du document

\pagestyle{special}

---Index
\end{document}

```

## 9.2. Classes standards

Voici, pour votre information, comment peuvent être définies les commandes des classes standards.

```

\titleformat{\chapter}[display]
  {\normalfont\huge\bfseries}{\chaptertitlename\ \thechapter}{20pt}{\Huge}
\titleformat{\section}
  {\normalfont\Large\bfseries}{\thesection}{1em}{}
\titleformat{\subsection}
  {\normalfont\large\bfseries}{\thesubsection}{1em}{}
\titleformat{\subsubsection}
  {\normalfont\normalsize\bfseries}{\thesubsubsection}{1em}{}
\titleformat{\paragraph}[runin]
  {\normalfont\normalsize\bfseries}{\theparagraph}{1em}{}
\titleformat{\subparagraph}[runin]
  {\normalfont\normalsize\bfseries}{\thesubparagraph}{1em}{}

\titlespacing*{\chapter}      {0pt}{50pt}{40pt}
\titlespacing*{\section}      {0pt}{3.5ex plus 1ex minus .2ex}{2.3ex plus .2ex}
\titlespacing*{\subsection}   {0pt}{3.25ex plus 1ex minus .2ex}{1.5ex plus .2ex}
\titlespacing*{\subsubsection}{0pt}{3.25ex plus 1ex minus .2ex}{1.5ex plus .2ex}
\titlespacing*{\paragraph}    {0pt}{3.25ex plus 1ex minus .2ex}{1em}
\titlespacing*{\subparagraph} {\parindent}{3.25ex plus 1ex minus .2ex}{1em}

```

## 9.3. Exemple de chapitre

Un dernier exemple montre comment tirer parti de l'environnement `picture` pour des formats de sectionnement moins courants. Même avec les outils simples fournis par la version standard de  $\text{\LaTeX}$ , vous pouvez créer des titres marquants. Mais vous pouvez en faire d'encore plus élaborés avec, par exemple, `pspicture` (de l'extension `PSTricks`) ou en incorporant des graphiques créés avec l'aide de programmes externes.

```
\usepackage[dvips]{color}
```

```

\usepackage[rigidchapters,explicit]{titlesec}

\DeclareFixedFont{\chapterfont}{T1}{phv}{bx}{n}{11cm}

\titlespacing{\chapter}{0pt}{0pt}{210pt}
% La plupart des titres ont un peu de profondeur. L'espace total
% est un peu plus large que la boîte contenant l'image.

\titleformat{\chapter}[block]
  {\begin{picture}(330,200)}
  {\put(450,80){%
    \makebox(0,0)[rb]{%
      \chapterfont\textcolor[named]{SkyBlue}{\thechapter}}}}
  \put(0,230){%
    \makebox(0,0)[lb]{%
      \Huge\sffamily\underline{Chapitre \thechapter}}}}
  {0pt}
  {\put(0,190){\parbox[t]{300pt}{%
    \Huge\sffamily\filright#1}}}
  [\end{picture}]

```

(Les valeurs exactes à utiliser dépendent de la zone de texte, de la classe, de `\unitlength`, du format du papier, etc.)