

# Le package **lybook** : taper des cours pour des lycéens

Code source disponible sur <https://github.com/bc-latex/ly-book>.

Version 0.0.0-beta développée et testée sur Mac OS X.

Christophe BAL

2019-09-24

## **Table des matières**



Première partie

Mise en forme



# Chapitre A

## Généralités

### I Pourquoi ?

Le but de ce mini-package est de réunir l'ensemble des outils utilisés par l'auteur pour rédiger ses cours. La mise en forme est celle de ce document. Elle se veut la moins extravagante possible car les cours rédigés devront être imprimés.

### II Avertissement

La mise en forme s'appuie sur l'utilisation de sections `\chapter` au minimum. C'est ce que permet la classe `memoir` utilisée pour rédiger cette documentation.

### III Titre du manuel (obligatoire pour le moment)

Si vous souhaitez indiquer un titre de manuel avec sa table des matières, il suffit d'utiliser la macro `\buildfront` qui admet trois arguments et s'utilise comme suit <sup>[1]</sup>.

#### *Code L<sup>A</sup>T<sub>E</sub>X*

```
\buildfront
  {Titre du manuel}
  {Nom et prénom de l'auteur}
  {Date(s)}
  {Niveau enseigné}
  {Lieu de l'enseignement}
```

L'emploi de cette commande produira le titre, l'ajout de la table des matières ainsi que la gestion d'une numérotation romaine pour les pages du sommaire et arabo-indienne pour le contenu principal, et aussi l'impression de pieds-de-page indiquant le niveau et le lieu du cours.

### IV Numérotation des parties, chapitres et sections

La numérotation retenue des titres est la suivante <sup>[2]</sup>.

1. Pour les parties, on utilise 1, 2, 3, ...

---

[1]. Tous les arguments sont obligatoires pour le moment.

[2]. Ces choix sont permis par l'utilisation du package `titlesec`.

2. Pour les chapitres, on utilise A, B, C, ..., le compteur n'étant pas remis à zéro lors de l'utilisation d'une nouvelle partie.
3. Pour les sections, on utilise I, II, III, ...
4. Pour les sous-sections, on utilise 1, 2, 3, ...
5. Pour les sous-sous-sections, on utilise i, ii, iii, ...

# Chapitre B

## Donner du sens au contenu

### I Écrire un théorème, une remarque, ...

#### 1 Théorèmes et compagnie

Des environnements permettent de rédiger des théorèmes, des propositions, des corollaires, des exemples, des remarques, ... avec un éventuel sous-titre via un argument optionnel<sup>[1]</sup>. Voici la liste complète des environnements<sup>[2]</sup>.

<code>conjecture</code>	<code>exercice</code>	<code>remark</code>
<code>corollary</code>	<code>lemma</code>	<code>solution</code>
<code>definition</code>	<code>methodology</code>	<code>theorem</code>
<code>definitions</code>	<code>notation</code>	<code>vocabulary</code>
<code>demo</code>	<code>notations</code>	<code>warning</code>
<code>example</code>	<code>proposition</code>	<code>warnings</code>

La numérotation est faite relativement aux sections et est commune à tous ces environnements<sup>[3]</sup>.

#### 2 Preuves et solutions

L'environnement `solution` sert à la rédaction de solutions d'exercices avec un éventuel sous-titre via un argument optionnel. On prendra garde que la numérotation sera celle de tout environnement<sup>[4]</sup> qui précède immédiatement la solution et qui a une numérotation commune avec celle des théorèmes comme par exemple les propositions ou les remarques bien que ce ne soit pas pour des exercices.

L'environnement `demo` est similaire sert à l'environnement `solution` mais pour rédiger des preuves<sup>[5]</sup>. Pour les preuves rédigées en classe avec les élèves, il suffit d'utiliser la commande `\demotodo` qui évite d'avoir à taper un environnement `demo` de contenu vide. Cette macro admet elle aussi un éventuel sous-titre via un argument optionnel.

#### 3 Comment ne pas numéroter ?

Les environnements et les macros ont tous une version étoilée qui n'utilise pas de numérotation<sup>[6]</sup>.

---

[1]. On utilise juste ici les possibilités du package `amsmath`.

[2]. Noter que seul `notation` propose une version avec un pluriel.

[3]. Ce choix est le plus pratique pour un lecteur de documents sur papier.

[4]. Y compris un environnement qui n'est pas un exercice !

[5]. L'auteur assume cet anglicisme.

[6]. Ceci permet peut être utile en début de chapitre par exemple.

## 4 Un exemple d'utilisation de la numérotation

Voici un exemple de code, présenté sur deux colonnes<sup>[7]</sup> où `\fakecontent` est une macro définie dans le code de ce document.

<i>Code L<sup>A</sup>T<sub>E</sub>X</i>	
<pre>\begin{theorem}[Important]   \fakecontent \end{theorem}</pre>	<pre>\begin{exercice*}[Mise en pratique]   \fakecontent \end{exercice*}</pre>
<pre>\begin{demo}[Unicité]   \fakecontent \end{demo}</pre>	<pre>\begin{solution*}[Les grandes lignes]   \fakecontent \end{solution*}</pre>
<pre>\begin{demo}[Existence]   \fakecontent \end{demo}</pre>	<pre>\begin{example}   \fakecontent \end{example}</pre>
	<pre>\demotodo</pre>

La mise en forme est la suivante.

<i>Rendu réel</i>
<p><b>Théorème I.1</b> (Important). <i>Bla, bla, bla, bla, bla, bla, bla, bla, bla, bla bla, bla, bla, bla, bla, bla, bla bla, bla, bla, bla, bla, bla, bla, bla, bla la, bla, bla, bla, bla, bla, bla la, bla, bla, bla, bla, bla ...</i></p>
<p><b>Preuve de I.1</b> (Unicité). <i>Bla, bla, bla, bla, bla, bla, bla, bla, bla bla, bla, bla, bla, bla, bla, bla, bla bla, bla, bla, bla, bla, bla, bla, bla, bla la, bla, bla, bla, bla, bla, bla la, bla, bla, bla, bla, bla, bla, bla, bla ...</i></p>
<p><b>Preuve de I.1</b> (Existence). <i>Bla, bla, bla, bla, bla, bla, bla, bla, bla bla, bla, bla, bla, bla, bla, bla, bla bla, bla, bla, bla, bla, bla, bla, bla, bla la, bla, bla, bla, bla, bla, bla la, bla, bla, bla, bla, bla, bla, bla, bla ...</i></p>
<p><b>Exercice</b> (Mise en pratique). <i>Bla, bla, bla, bla, bla, bla, bla, bla, bla bla, bla, bla, bla, bla, bla, bla, bla bla, bla, bla, bla, bla, bla, bla, bla, bla la, bla, bla, bla, bla, bla, bla la, bla, bla, bla, bla, bla, bla, bla, bla ...</i></p>
<p><b>Solution</b> (Les grandes lignes). <i>Bla, bla, bla, bla, bla, bla, bla, bla, bla bla, bla, bla, bla, bla, bla, bla, bla bla, bla, bla, bla, bla, bla, bla, bla, bla la, bla, bla, bla, bla, bla, bla la, bla, bla, bla, bla, bla, bla, bla, bla ...</i></p>
<p><b>Exemple I.2.</b> <i>Bla, bla, bla, bla, bla, bla, bla, bla, bla bla, bla, bla, bla, bla, bla, bla, bla bla, bla, bla, bla, bla, bla, bla la, bla, bla, bla, bla, bla, bla, bla la, bla, bla, bla, bla, bla, bla, bla, bla ...</i></p>
<p><b>Preuve de I.2.</b></p>

[7]. Nous avons utilisé ici l'environnement `multicols-sep` proposé par le package `lybook`.



## II Listes

L'environnement `bullets` permet d'utiliser des listes utilisant des puces comme celle ci-dessous <sup>[8]</sup>.

- On utilise `\begin{bullets} ... \end{bullets}`.
- Dans cet environnement, on utilise `\item` pour chaque nouveau point.

## III Multi-colonnes et traits verticaux

L'environnement `multicols-sep` qui s'utilise comme `multicols` du package `multicol` à la différence près qu'il y a un trait vertical de séparation avec un l'ajout d'un espace entre chacun des traits et les contenus juste à leur droite. Voici un exemple de code.

### *Code $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$*

```
\begin{multicols-sep}{3}
Colonne 1

Que dire ? Quoi écrire ?

Colonne 2

Que dire ? Quoi écrire ?

Colonne 3

Que dire ? Quoi écrire ?
\end{multicols-sep}
```

La mise en forme obtenue est la suivante.

### *Rendu réel*

Colonne 1	Colonne 2	Colonne 3
Que dire ? Quoi écrire ?	Que dire ? Quoi écrire ?	Que dire ? Quoi écrire ?

## IV Entre des guillemets

En tapant `\lyquote{un texte entre des guillemets}`, on obtient « *un texte entre des guillemets* ».

[8]. C'est le package `enumitem` qui fait tout le boulot.

## V Des cadres

Pour bien séparer certains contenus particuliers comme du code par exemple, il est possible d'utiliser des cadres avec ou sans titre<sup>[9]</sup>. Voici un exemple de code où `\fakecontent` est une macro définie dans le code de ce document.

### *Code L<sup>A</sup>T<sub>E</sub>X*

```
\begin{frame-gene}{Un titre personnalisable}
  \fakecontent
\end{frame-gene}

\begin{frame-gene}{} % ATTENTION ! Le titre est un argument obligatoire.
  Pas de titre pour moi. Merci !
\end{frame-gene}
```

Nous mettons directement la mise en forme obtenue ci-dessous.

### *Un titre personnalisable*

Bla, bla, bla, bla, bla, bla, bla, bla, bla bla, bla, bla, bla, bla, bla, bla, bla bla, bla, bla, bla, bla, bla, bla, bla la, bla, bla, bla, bla, bla, bla, bla la, bla, bla, bla, bla, bla, bla, bla, bla ...

Pas de titre pour moi. Merci !

On peut contrôler la largeur relativement à la largeur de la ligne via un argument optionnel. Dans le code précédent, l'utilisation de `\begin{frame-gene}[.5]{...}` au lieu de `\begin{frame-gene}{...}`, on obtient ce qui suit.

### *Un titre personnalisable*

Bla, bla, bla, bla, bla, bla, bla, bla, bla bla, bla, bla, bla, bla, bla, bla bla, bla, bla, bla, bla, bla, bla la, bla, bla, bla, bla, bla, bla la, bla, bla, bla, bla, bla, bla ...

Pas de titre pour moi. Merci !

[9]. Merci au package `tcolorbox`.

## Deuxième partie

### Rédactions plus techniques



# Chapitre C

## Extraire les résultats importants

I Les options `extract` et `short` — TODO

II Les environnements `forshort` et `forlong` — TODO

????

`ifshort`

`iflong`

`elifshort`

`eliflong`

`forshort`

`forlong`

`toyt` via `coomment` ets astuce