

beaulivre

ÉCRIVEZ VOS LIVRES
DE MANIÈRE COLORÉE

Correspondant à : beaulivre 2021/08/08

JINWEN XU

Août 2021, à Pékin

Préface

beaulivre fait partie de la série de classes `colorist`, dont le nom est tiré des mots « beau » et « livre ». L'ensemble de la collection comprend `colorart` et `lebhart` pour la composition d'articles, et `colorbook` et `beaulivre` pour celle des livres. Mon intention initiale en les concevant était d'écrire des brouillons et des notes qui ont l'air coloré mais pas éblouissant.

beaulivre prend en charge plusieurs langues, notamment le chinois (simplifié et traditionnel), l'anglais, le français, l'allemand, l'italien, le japonais, le portugais (européen et brésilien), le russe et l'espagnol. Ces langues peuvent être commutées de manière transparente dans un seul document. En raison de l'utilisation de polices personnalisées, beaulivre demande soit $\text{Xe}_{\text{L}}^{\text{A}}\text{T}_{\text{E}}\text{X}$ soit $\text{Lua}_{\text{A}}^{\text{A}}\text{T}_{\text{E}}\text{X}$ pour la compilation.

Cette documentation est composée à l'aide de beaulivre (avec l'option `allowbf`). Vous pouvez le considérer comme une courte introduction et une démonstration.

ASTUCE

La prise en charge multilingue, les environnements de type théorème, les marques de brouillon et quelques autres fonctionnalités sont fournis par la boîte à outils [ProjLib](#). Ici, nous ne discutons que brièvement de la façon de l'utiliser avec cette classe de document. Pour plus d'informations, veuillez vous référer à la documentation de [ProjLib](#).

Table des matières

I INSTRUCTION

| | |
|--|-----------|
| Avant de commencer | 3 |
| 1 Utilisation et exemples | 5 |
| 1.1 Comment l'ajouter | 5 |
| 1.2 Exemple - Un document complet | 5 |
| 1.2.1 Initialisation | 6 |
| 1.2.2 Choisir la langue | 6 |
| 1.2.3 Marques de brouillon | 6 |
| 1.2.4 Environnements de type théorème | 6 |
| 2 À propos des polices par défaut | 7 |
| 3 Les options | 9 |
| 4 Instructions par sujet | 11 |
| 4.1 Configurer la langue | 11 |
| 4.2 Théorèmes et comment les référencer | 12 |
| 4.3 Définir un nouvel environnement de type théorème | 13 |
| 4.4 Draft mark | 14 |
| 5 Problèmes connus | 15 |

II DÉMONSTRATION

| | |
|---------------------------------------|-----------|
| 6 Titres de niveau 0 (chapter) | 19 |
| 6.1 Titres de niveau 1 (section) | 19 |
| 6.1.1 Titres de niveau 2 (subsection) | 19 |

| | |
|---|----|
| 6.2 Lists | 20 |
| 6.2.1 Example pour une list (itemize) | 20 |
| 6.2.2 Example pour une list (enumerate) | 20 |
| 6.2.3 Example pour une list (description) | 21 |

PARTIE I

INSTRUCTION

Vous pouvez ajouter quelque texte d'introduction ici via `\parttext<text>`.

Avant de commencer

Pour utiliser les classes de documents décrites ici, vous devez :

- installer TeX Live ou MikTeX de la dernière version possible, et vous assurer que `colorist` et `projlib` sont correctement installés dans votre système $\text{T}_{\text{E}}\text{X}$.
- télécharger et installer les polices requises, voir « À propos des polices par défaut ».
- être familiarisé avec l'utilisation de base de $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, et savoir comment compiler vos documents avec `pdf $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$` , `X Y $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$` ou `Lua $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$` .

1

Utilisation et exemples

1.1 Comment l'ajouter

Il suffit simplement de mettre

```
\documentclass{beaulivre}
```

comme première ligne pour utiliser la classe beaulivre.

ATTENTION

Vous devez utiliser le moteur \LaTeX ou \LuaTeX pour compiler.

1.2 Exemple - Un document complet

Regardons d'abord un document complet.

```
1 \documentclass{colorbook}
2 \usepackage{ProjLib}
3
4 \UseLanguage{French}
5
6 \begin{document}
7
8 \frontmatter
9
10 \begin{titlepage}
11     <code for titlepage>
12 \end{titlepage}
13
14 \tableofcontents
15
16 \mainmatter
17
18 \part{<part title>}
19 \parttext{<text after part title>}
20
21 \chapter{<chapter title>}
22
23 \section{<section title>}
24
25 \dnf<Plus de contenu est nécessaire.>
```

```

26
27 \begin{theorem}\label{thm:abc}
28     Ceci est un théorème.
29 \end{theorem}
30 Référence du théorème: \cref{thm:abc}
31
32 \backmatter
33
34 ...
35
36 \end{document}

```

Si vous trouvez cela un peu compliqué, ne vous inquiétez pas. Examinons maintenant cet exemple pièce par pièce.

1.2.1 Initialisation

```

\documentclass{beaulivre}
\usepackage{ProjLib}

```

L’initialisation est simple. La première ligne ajoute la classe de document `beaulivre`, et la deuxième ligne ajoute la boîte à outils `ProjLib` pour obtenir des fonctionnalités supplémentaires.

1.2.2 Choisir la langue

```

\UseLanguage{French}

```

Cette ligne indique que le français sera utilisé dans le document (d’ailleurs, si seul l’anglais apparaît dans votre article, alors il n’est pas nécessaire de choisir la langue). Vous pouvez également changer de langue de la même manière plus tard au milieu du texte. Les langues prises en charge sont les suivantes : chinois simplifié, chinois traditionnel, japonais, anglais, français, allemand, espagnol, portugais, portugais brésilien et russe.

Pour une description détaillée de cette commande et d’autres commandes associées, veuillez vous référer à la section sur le support multilingue.

1.2.3 Marques de brouillon

```

\dntf<(some hint)>

```

Lorsque vous avez des endroits qui ne sont pas encore finis, vous pouvez les marquer avec cette commande, ce qui est particulièrement utile lors de la phase de brouillon.

1.2.4 Environnements de type théorème

```

\begin{theorem}\label{thm:abc}
    Ceci est un théorème.
\end{theorem}
Référence du théorème: \cref{thm:abc}

```

Les environnements de type théorème couramment utilisés ont été prédéfinis. De plus, lors du référencement d’un environnement de type théorème, il est recommandé d’utiliser `\cref{label}` — de cette manière, il ne serait pas nécessaire d’écrire explicitement le nom de l’environnement correspondant à chaque fois.

2

À propos des polices par défaut

Par défaut, cette classe de document utilise Palatino Linotype comme police anglaise principale ; Source Han Serif, Source Han Sans et Source Han Mono comme police chinoise principale, sans empattement et monospace ; et utilise partiellement Neo Euler comme police mathématique. Vous devez télécharger et installer ces polices vous-même. La série de polices Source Han peut être téléchargée sur <https://github.com/adobe-fonts> (il est recommandé de télécharger la version Super-OTC, afin que la taille de téléchargement soit plus petite). Neo Euler peut être téléchargé sur <https://github.com/khaledhosny/euler-otf>. Lorsque la police correspondante n'est pas installée, la police fournie avec TeX Live sera utilisée à la place et l'effet peut être réduit.

De plus, Source Code Pro est utilisé comme police anglaise sans empattement, New Computer Modern Mono comme police anglaise monospace, ainsi que certains symboles dans les polices mathématiques Asana Math, Tex Gyre Pagella Math et Latin Modern Math. Ces polices sont déjà disponibles dans TeX Live ou MikTeX, ce qui signifie que vous n'avez pas besoin de les installer vous-même.

- English main font. English sans serif font. English typewriter font.
- 简体中文主要字体，简体中文无衬线字体，简体中文等宽字体
- 繁體中文主要字體，繁體中文無襯線字體，繁體中文等寬字體
- 日本語のメインフォント、日本語のサンセリフフォント、日本語の等幅フォント
- Démonstration de maths : $\alpha, \beta, \gamma, \delta, 1, 2, 3, 4, a, b, c, d,$

$$\text{li}(x) := \int_2^x \frac{1}{\log t} dt$$

3

Les options

- Les options de langue EN / english / English, FR / french / French, etc.
 - Pour les noms d’options d’une langue spécifique, veuillez vous référer à *<language name>* dans la section suivante. La première langue spécifiée sera considérée comme la langue par défaut.
 - Les options de langue ne sont pas nécessaires, elles servent principalement à augmenter la vitesse de compilation. Sans eux, le résultat serait le même, justement plus lent.
- draft ou fast
 - L’option fast permet un style plus rapide mais légèrement plus rugueux, les principales différences sont :
 - * Utilisez une configuration de police mathématique plus simple ;
 - * N’utilisez pas hyperref ;
 - * Activez le mode rapide de la boîte à outils ProjLib.

ASTUCE

Pendant la phase de brouillon, il est recommandé d’utiliser le fast option pour accélérer la compilation. Quand dans fast mode, il y aura un filigrane “DRAFT” pour indiquer que vous êtes actuellement en mode brouillon.

- a4paper ou b5paper
 - Options de format de papier. Le format de papier par défaut est 8.5 pouces × 11 pouces.
- palatino, times, garamond, noto, biolinum | useosf
 - Options de police. Comme son nom l’indique, la police avec le nom correspondant sera utilisée.
 - L’option useosf est pour activer les chiffres à l’ancienne.
- allowbf
 - Afficher les titres en gras. Lorsque cette option est utilisée, le titre principal, les titres de tous les niveaux et les noms des environnements de type théorème seront en gras.
- runin
 - Utilisez le style « runin » pour \subsubsection
- puretext ou nothms
 - Mode texte pur. Ne pas définir les environnements de type théorème.
- nothmnum, thmnum ou thmnum=<counter>
 - Les environnements de type théorème ne seront pas numérotés / numérotés dans l’ordre 1, 2, 3... / numérotés dans <counter>. Ici, <counter> doit être un compteur intégré (tel que subsection) ou un compteur défini dans le préambule. Si aucune option n’est utilisée, ils seront numérotés dans chapter (livre) ou section (article).

— `regionalref`, `originalref`

- Lors du référencement, si le nom de l’environnement de type théorème change avec la langue actuelle. Par défaut `regionalref` est activé, c’est-à-dire que le nom correspondant à la langue courante est utilisé ; par exemple, lors du référencement d’un environnement de type théorème dans un contexte français, les noms « Théorème, Définition ... » seront utilisés quel que soit le contexte linguistique dans lequel se trouve l’environnement d’origine. Si `originalref` est activé, alors le nom restera toujours le même que l’environnement d’origine ; par exemple, lors du référencement d’un théorème écrit dans le contexte français, même si l’on est actuellement dans le contexte anglais, il sera toujours affiché comme « Théorème ».
- En mode `fast`, l’option `originalref` n’aura aucun effet.

De plus, les options `oneside` et `twoside` couramment utilisées lors de la composition de livres sont également disponibles. La disposition recto-verso est utilisée par défaut.

4

Instructions par sujet

4.1 Configurer la langue

beaulivre prend en charge plusieurs langues, notamment le chinois (simplifié et traditionnel), l'anglais, le français, l'allemand, l'italien, le japonais, le portugais (européen et brésilien), le russe et l'espagnol. La langue peut être sélectionnée par les macros suivantes :

- `\UseLanguage{<language name>}` est utilisé pour spécifier la langue. Le réglage correspondant de la langue sera appliqué après celui-ci. Il peut être utilisé soit dans le préambule ou dans le texte. Lorsqu'aucune langue n'est spécifiée, « English » est sélectionné par défaut.
- `\UseOtherLanguage{<language name>}{<content>}`, qui utilise les paramètres de langue spécifiés pour composer `<content>`. Par rapport à `\UseLanguage`, il ne modifiera pas l'interligne, donc l'interligne restera stable lorsque le texte CJK et occidental sont mélangés.

`<language name>` peut être (il n'est pas sensible à la casse, par exemple, French et french ont le même effet) :

- chinois simplifié : CN, Chinese, SChinese ou SimplifiedChinese
- chinois traditionnel : TC, TChinese ou TraditionalChinese
- anglais : EN ou English
- français : FR ou French
- allemand : DE, German ou ngerman
- italien : IT ou Italian
- portugais : PT ou Portuguese
- portugais (brésilien) : BR ou Brazilian
- espagnol : ES ou Spanish
- japonais : JP ou Japanese
- russe : RU ou Russian

De plus, vous pouvez également ajouter de nouveaux paramètres à la langue sélectionnée :

- `\AddLanguageSetting{<settings>}`
 - Ajoutez `<settings>` à toutes les langues prises en charge.
- `\AddLanguageSetting(<language name>){<settings>}`
 - Ajoutez `<settings>` à la langue `<language name>` sélectionnée.

Par exemple, `\AddLanguageSetting(German){\color{orange}}` peut rendre tout le texte allemand affiché en orange (bien sûr, il faut alors ajouter `\AddLanguageSetting{\color{black}}` afin de corriger la couleur du texte dans d'autres langues).

4.2 Théorèmes et comment les référencer

Des environnements tels que `definition` et `theorem` ont été prédéfinis et peuvent être utilisés directement.

Plus précisément, les environnements prédéfinis incluent : `assumption`, `axiom`, `conjecture`, `convention`, `corollary`, `definition`, `definition-proposition`, `definition-theorem`, `example`, `exercise`, `fact`, `hypothesis`, `lemma`, `notation`, `observation`, `problem`, `property`, `proposition`, `question`, `remark`, `theorem`, et la version non numérotée correspondante avec un astérisque `*` dans le nom. Les titres changeront avec la langue actuelle. Par exemple, `theorem` sera affiché comme « Theorem » en mode anglais et « Théorème » en mode français.

Lors du référencement d'un environnement de type théorème, il est recommandé d'utiliser `\cref{<label>}`. De cette façon, il n'est pas nécessaire d'écrire explicitement le nom de l'environnement correspondant à chaque fois.

EXEMPLE

```
\begin{definition}[Des choses étranges] \label{def: strange} ...
```

will produce

DÉFINITION 4.1 (Des choses étranges) C'est la définition de certains objets étranges. Il y a approximativement un espace d'une ligne avant et après l'environnement de type théorème, et il y aura un symbole pour marquer la fin de l'environnement.

`\cref{def: strange}` s'affichera sous la forme : **DÉFINITION 4.1**.

Après avoir utilisé `\UseLanguage{French}`, un théorème s'affichera sous la forme :

THEOREM 4.2 (Useless) A theorem in English.

Par défaut, lors du référencement, le nom du théorème correspond au contexte courant. Par exemple, le nom de la définition ci-dessus sera en français dans le contexte français courant : **DÉFINITION 4.1** and **THEOREM 4.2**. Si vous voulez que le nom du théorème corresponde toujours à la langue du contexte dans lequel se trouve le théorème, vous pouvez ajouter `originalref` aux options globales.

Voici les principaux styles d'environnements de type théorème :

THÉORÈME 4.3 Style de théorème : `theorem`, `proposition`, `lemma`, `corollary`, ...

Démonstration | Style d'épreuve



Style de remarque



CONJECTURE 4.4 Style de conjecture

EXEMPLE Style d'exemple : `example`, `fact`, ...

PROBLÈME 4.5 Style de problème : `problem`, `question`, ...

Pour l'esthétique, les définitions adjacentes seront reliées entre elles automatiquement :

DÉFINITION 4.6 Première définition.

DÉFINITION 4.7 Deuxième définition.

4.3 Définir un nouvel environnement de type théorème

Si vous avez besoin de définir un nouvel environnement de type théorème, vous devez d'abord définir le nom de l'environnement dans le langage à utiliser :

— `\NameTheorem[⟨language name⟩]{⟨name of environment⟩}{⟨name string⟩}`

Pour `⟨language name⟩`, veuillez vous référer à la section sur la configuration de la langue. Lorsqu'il n'est pas spécifié, le nom sera défini pour toutes les langues prises en charge. De plus, les environnements avec ou sans astérisque partagent le même nom, donc `\NameTheorem{envname*}{...}` a le même effet que `\NameTheorem{envname}{...}`.

Ensuite, créez cet environnement de l'une des cinq manières suivantes :

- `\CreateTheorem*{⟨name of environment⟩}`
 - Définir un environnement non numéroté `⟨name of environment⟩`
- `\CreateTheorem{⟨name of environment⟩}`
 - Définir un environnement numéroté `⟨name of environment⟩`, numéroté dans l'ordre 1, 2, 3, ...
- `\CreateTheorem{⟨name of environment⟩}[⟨numbered like⟩]`
 - Définir un environnement numéroté `⟨name of environment⟩`, qui partage le compteur `⟨numbered like⟩`
- `\CreateTheorem{⟨name of environment⟩}<⟨numbered within⟩>`
 - Définir un environnement numéroté `⟨name of environment⟩`, numéroté dans le compteur `⟨numbered within⟩`
- `\CreateTheorem{⟨name of environment⟩}(⟨existed environment⟩)`
`\CreateTheorem*{⟨name of environment⟩}(⟨existed environment⟩)`
 - Identifiez `⟨name of environment⟩` avec `⟨existed environment⟩` ou `⟨existed environment⟩*`.
 - Cette méthode est généralement utile dans les deux situations suivantes :
 1. Pour utiliser un nom plus concis. Par exemple, avec `\CreateTheorem{thm}(theorem)`, on peut alors utiliser le nom `thm` pour écrire le théorème.
 2. Pour supprimer la numérotation de certains environnements. Par exemple, on peut supprimer la numérotation de l'environnement `remark` avec `\CreateTheorem{remark}(remark*)`.

ASTUCE

Cette macro utilise la fonctionnalité de `amsthm` en interne, donc le traditionnel `theoremstyle` lui est également applicable. Il suffit de déclarer le style avant les définitions pertinentes.

Voici un exemple. Le code suivant :

```
\NameTheorem[FR]{proofidea}{Idée}  
\CreateTheorem*{proofidea*}  
\CreateTheorem{proofidea}<section>
```

définit un environnement non numéroté `proofidea*` et un environnement numéroté `proofidea` (numérotés dans la section) respectivement. Ils peuvent être utilisés dans le contexte français. L'effet est le suivant :

Idée | La environnement `proofidea*`.

Idée 4.3.1 | La environnement `proofidea`.



4.4 Draft mark

Vous pouvez utiliser `\dnf` pour marquer la partie inachevée. Par exemple :

- `\dnf` ou `\dnf<...>`. L'effet est : `Pas encore fini #1` ou `Pas encore fini #2: ...`.

Le texte à l'intérieur changera en fonction de la langue actuelle. Par exemple, il sera affiché sous la forme `To be finished #3` en mode anglais.

De même, il y a aussi `\needgraph` :

- `\needgraph` ou `\needgraph<...>`. L'effet est :

`Il manque une image ici #1`

ou

`Il manque une image ici #2: ...`

Le texte de l'invite change en fonction de la langue actuelle. Par exemple, en mode anglais, il sera affiché sous la forme

`A graph is needed here #3`

5

Problèmes connus

- Les paramètres de police ne sont pas encore parfaits.
- La conception n'est pas encore terminée, y compris la table des matières, les pièces et les chapitres.
- Comme de nombreuses fonctionnalités sont basées sur la boîte à outils ProjLib, colorist (et donc colorart, lebhart et colorbook, beaulivre) hérite de tous ses problèmes. Pour plus de détails, veuillez vous référer à la section « Problèmes connus » de la documentation de ProjLib.
- Le mécanisme de gestion des erreurs est incomplet : pas de messages correspondants lorsque certains problèmes surviennent.
- Il y a encore beaucoup de choses qui peuvent être optimisées dans le code.

PARTIE II

DÉMONSTRATION

6

Titres de niveau 0 (chapter)

Qu'est que c'est?. C'est une phrase français avant le lorem ipsum. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

6.1 Titres de niveau 1 (section)

Qu'est que c'est?. C'est une phrase français avant le lorem ipsum. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

6.1.1 Titres de niveau 2 (subsection)

Qu'est que c'est?. C'est une phrase français avant le lorem ipsum. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Titres de niveau 3 (subsubsection)

Qu'est que c'est?. C'est une phrase français avant le lorem ipsum. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

TITRES DE NIVEAU 4 (PARAGRAPH) Qu'est que c'est ?. C'est une phrase français avant le lorem ipsum. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

6.2 Lists

6.2.1 Example pour une list (itemize)

- Premier point dans une list
- Deuxième point dans une list
- Troisième point dans une list
- Quatrième point dans une list
- Cinquième point dans une list

Example pour une list (4*itemize)

- Premier point dans une list
 - Premier point dans une list
 - * Premier point dans une list
 - Premier point dans une list
 - Deuxième point dans une list
 - * Deuxième point dans une list
 - Deuxième point dans une list
- Deuxième point dans une list

6.2.2 Example pour une list (enumerate)

1. Premier point dans une list
2. Deuxième point dans une list
3. Troisième point dans une list
4. Quatrième point dans une list
5. Cinquième point dans une list

Example pour une list (4*enumerate)

1. Premier point dans une list
 - a) Premier point dans une list
 - (i) Premier point dans une list
 - A. Premier point dans une list
 - B. Deuxième point dans une list
 - (ii) Deuxième point dans une list
 - b) Deuxième point dans une list
2. Deuxième point dans une list

6.2.3 Exemple pour une list (description)

Premier point dans une list

Deuxième point dans une list

Troisième point dans une list

Quatrième point dans une list

Cinquième point dans une list

Exemple pour une list (4*description)

Premier point dans une list

 Premier point dans une list

 Premier point dans une list

 Premier point dans une list

 Deuxième point dans une list

 Deuxième point dans une list

 Deuxième point dans une list

Deuxième point dans une list