

# lebhart

## ÉCRIVEZ VOS ARTICLES DE MANIÈRE COLORÉE

JINWEN XU

ProjLib@outlook.com

Novembre 2021, à Paris

### RÉSUMÉ

lebhart fait partie de la série de classes `colorist`, dont le nom est tiré du mot allemand « `lebhaft` » (animé), combiné avec les trois premières lettres de « `artikel` » (article). 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.

lebhart 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, lebhart demande soit  $X_{\LaTeX}$  soit  $Lua_{\LaTeX}$  pour la compilation.

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

### TABLE DES MATIÈRES

Avant de commencer	1
1 Utilisation et exemples	2
1.1 Comment l'ajouter	2
1.2 Exemple - Un document complet	2
2 À propos des polices par défaut	4
3 Les options	5
4 Instructions par sujet	6
4.1 Configurer la langue	6
4.2 Théorèmes et comment les référencer	7
4.3 Définir un nouvel environnement de type théorème	8
4.4 Draft mark	9
4.5 Titre, résumé et mots-clés	9
5 Problèmes connus	10

### 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  $TeX$ .
- télécharger et installer les polices requises, voir « À propos des polices par défaut ».
- être familiarisé avec l'utilisation de base de  $LaTeX$ , et savoir comment compiler vos documents avec  $pdfLaTeX$ ,  $X_{\LaTeX}$  ou  $Lua_{\LaTeX}$ .

---

Correspondant à : lebhart 2021/11/09

# 1 UTILISATION ET EXEMPLES

## 1.1 COMMENT L'AJOUTER

Il suffit simplement de mettre

---

```
\documentclass{lebhart}
```

---

comme première ligne pour utiliser la classe lebhart. Veuillez noter que vous devez utiliser le moteur  $\text{Xe}_{\text{L}}\text{A}_{\text{T}}\text{E}_{\text{X}}$  ou  $\text{Lua}_{\text{L}}\text{A}_{\text{T}}\text{E}_{\text{X}}$  pour compiler.

## 1.2 EXEMPLE - UN DOCUMENT COMPLET

Regardons d'abord un document complet.

---

```
1 \documentclass{lebhart}
2 \usepackage{ProjLib}
3
4 \UseLanguage{French}
5
6 \begin{document}
7
8 \title{<title>}
9 \author{<author>}
10 \date{\PLdate{2022-04-01}}
11
12 \maketitle
13
14 \begin{abstract}
15     Ceci est un résumé. \dnf<<some hint>>
16 \end{abstract}
17 \begin{keyword}
18     AAA, BBB, CCC, DDD, EEE
19 \end{keyword}
20
21 \section{Un théorème}
22
23 \begin{theorem}\label{thm:abc}
24     Ceci est un théorème.
25 \end{theorem}
26 Référence du théorème: \cref{thm:abc}
27
28 \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{lebhart}
\usepackage{ProjLib}
```

---

L'initialisation est simple. La première ligne ajoute la classe de document lebhart, 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 Titre, informations sur l'auteur, résumé et mots-clés

---

```
\title{<title>}
\author{<author>}
\date{\PLdate{2022-04-01}}
\maketitle
```

```
\begin{abstract}
  <abstract>
\end{abstract}
\begin{keyword}
  <keywords>
\end{keyword}
```

---

Cette partie commence par le titre et le bloc d'informations sur l'auteur. L'exemple montre l'utilisation de base, mais en fait, vous pouvez également écrire comme :

```
\author{<author 1>}
\address{<address 1>}
\email{<email 1>}
\author{<author 2>}
\address{<address 2>}
\email{<email 2>}
...
```

---

De plus, vous pouvez également écrire à la manière  $\mathcal{AMS}$ , c'est-à-dire :

```
\title{<title>}
\author{<author 1>}
\address{<address 1>}
\email{<email 1>}
\author{<author 2>}
\address{<address 2>}
\email{<email 2>}
\date{\PLdate{2022-04-01}}
\subclass{*****}
\keywords{<keywords>}

\begin{abstract}
  <abstract>
\end{abstract}
```

---

```
\maketitle
```

### 1.2.4 Marques de brouillon

```
\dnf<<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.

### 1.2.5 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.

#### ASTUCE

Si vous souhaitez utiliser la classe standard à la place plus tard, remplacez simplement les deux premières lignes par :

```
\documentclass{article}
\usepackage[a4paper,margin=1in]{geometry}
\usepackage[hidelinks]{hyperref}
\usepackage[palatino,amsfashion]{ProjLib}
```

ou utilisez la classe  $\mathcal{AMS}$  :

```
\documentclass{amsart}
\usepackage[a4paper,margin=1in]{geometry}
\usepackage[hidelinks]{hyperref}
\usepackage[palatino]{ProjLib}
```

#### ASTUCE

Si vous aimez la classe de document actuelle, mais que vous souhaitez un style plus « simple », vous pouvez utiliser l'option `classical`, comme ceci :

```
\documentclass[classical]{lebart}
```

## 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.

### 3 LES OPTIONS

lebhart propose les options suivantes :

- 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.
- `delaythms`
  - Reportez la définition des environnements de type théorème à la fin du préambule. Utilisez cette option si vous souhaitez que les environnements soient numérotés dans un compteur personnalisé.
- `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 (avec l'option `delaythms` activée). 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.

## 4 INSTRUCTIONS PAR SUJET

### 4.1 CONFIGURER LA LANGUE

lebart 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 : **DEFINITION 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 *name of environment* non numéroté
- `\CreateTheorem{name of environment}`
  - Définir un environnement *name of environment* numéroté dans l'ordre 1, 2, 3, ...
- `\CreateTheorem{name of environment}[numbered like]`
  - Définir un environnement *name of environment* numéroté, qui partage le compteur *numbered like*
- `\CreateTheorem{name of environment}<numbered within>`
  - Définir un environnement *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}<subsection>
```

définit un environnement non numéroté `proofidea*` et un environnement numéroté `proofidea` (numérotés dans la sous-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`

#### 4.5 TITRE, RÉSUMÉ ET MOTS-CLÉS

lebart possède à la fois les caractéristiques des classes standard et celles des classes  $\mathcal{AMS}$ .

Par conséquent, le titre et les informations sur l'auteur peuvent être soit écrits de la manière habituelle, conformément à la classe standard article :

---

```
\title{<title>}
\author{<author>\thanks{<text>}}
\date{<date>}
\maketitle
\begin{abstract}
  <abstract>
\end{abstract}
\begin{keyword}
  <keywords>
\end{keyword}
```

---

ou écrit à la manière des classes  $\mathcal{AMS}$  :

---

```
\title{<title>}
\author{<author>}
\thanks{<text>}
\address{<address>}
\email{<email>}
\date{<date>}
\keywords{<keywords>}
\subclass{<subclass>}
\begin{abstract}
  <abstract>
\end{abstract}
\maketitle
```

---

Les informations sur l’auteur peuvent contenir plusieurs groupes, écrits comme suit :

---

```
\author{\langle author 1\rangle}
\address{\langle address 1\rangle}
\email{\langle email 1\rangle}
\author{\langle author 2\rangle}
\address{\langle address 2\rangle}
\email{\langle email 2\rangle}
...
```

---

Parmi eux, l’ordre mutuel de `\address`, `\curraddr`, `\email` n’est pas important.

## 5 PROBLÈMES CONNUS

- Les paramètres de police ne sont pas encore parfaits.
- La conception de la table des matières est pas si belle.
- 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.