

---

# colorist, write your articles or books in a colorful way\*

JINWEN

April 2021, Beijing

---

## ABSTRACT

The colorist is a series of styles and classes for you to typeset your articles or books in a colorful manner. My original intention in designing this series is to write drafts and notes that look colorful yet not dazzling.

The entire collection includes `colorist.sty`, which is the main style shared by all the following classes; `colorart.cls` for typesetting articles and `colorbook.cls` for typesetting books. They compile with any major  $\text{\TeX}$  engine, with native support to English and French typesetting via `\UseLanguage` (see the instruction below for detail).

You can also found `lebhart` and `beaulivre` on CTAN. They are the enhanced version of `colorart` and `colorbook` with unicode support. With this, they can access to more beautiful fonts, and also have native support for Chinese typesetting. On the other hand, they need to be compiled with  $\text{\XeLaTeX}$  or  $\text{\LuaLaTeX}$  (not  $\text{\pdfLaTeX}$ ).

This documentation is typeset using `colorart`. You can think of it as a short introduction and demonstration.

Since the main body of `colorist` is modified from the minimalist series, some elements have not been completely redesigned yet, especially the TOC, part and chapter style. These will be added gradually in the future versions.

## CONTENTS

---

1	How to load it	1
1.1	A useful switch: the option <code>fast</code> or <code>draft</code>	2
2	Some instructions	2
2.1	Theorems and how to reference them	2
2.2	Define a new theorem-like environment	2
2.3	Draft mark	3
2.4	Language configuration	3
3	Document templates	4

## 1 HOW TO LOAD IT

---

You can directly use `colorart` or `colorbook` as your document class. In this way, you can directly begin writing your document, without having to worry about the configurations.

```
\documentclass{colorart} or \documentclass{colorbook}
```

And of course, you can also use the default classes `article` or `book`, and load the `colorist` package. This way, only the basic styles are set, and you can thus use your preferred fonts and page layout. All the features mentioned in this article (except for draft marks) are provided.

```
\usepackage{colorist}
```

---

\*Corresponding to: `colorist` 2021/04/05

## 1.1 | A useful switch: the option fast or draft

You can use the option `fast` or `draft` to typeset in a fast but slightly rough manner. The main differences are:

- Do not use `hyperref`;
- Use draft mode for all `tcolorboxes`.

During the writing stage of your document, it is recommended to use the `fast` option to speed up compilation and improve the smoothness of your writing experience. At the end, you can remove the “fast” mark to get the final version.

## 2 SOME INSTRUCTIONS

### 2.1 | Theorems and how to reference them

Environments such as definitions and theorems have been pre-defined and can be used directly, for example:

```
\begin{definition}[Strange things] \label{def: strange} ...
```

will produce

**DEFINITION 2.1** (Strange things) This is the definition of some strange objects.

**DEFINITION 2.2** (Another definition) Different definitions are automatically glued together, and there will be no gaps in between.

When referencing, you can directly use clever reference `\cref{(label name)}`. For example, `\cref{def: strange}` will be displayed as: **DEFINITION 2.1**.

The following are several other styles of theorem-like environments:

**THEOREM 2.3** Theorem style: theorem, proposition, lemma, corollary

**PROOF** | Proof style



*Remark style*



**CONJECTURE 2.1** Conjecture style

**EXAMPLE** Example style: example, fact

**PROBLEM 2.1** Problem style

### 2.2 | Define a new theorem-like environment

First define the name of this environment in the language used: `\(name of environment)(language name)`. Where `(language name)` can be `EN`, `FR`, `CN`, etc., and then define this environment in one of the following four ways:

- `\CreateTheorem*{(name of environment)}`
- `\CreateTheorem{(name of environment)}[(numbered like)]`
- `\CreateTheorem{(name of environment)}<(numbered within)>`
- `\CreateTheorem{(name of environment)}`

For example,

```
\def\proofideanameEN{Idea}  
\CreateTheorem*{proofidea}
```

defines an unnumbered environment `proofidea`, which supports using in the English context, and the effect is as follows:

IDEA | ...

### 2.3 | Draft mark

You can use `\dnf` to mark the unfinished part. For example:

- `\dnf:` `To be finished #1`
- `\dnf<Still need ...>:` `To be finished #2: Still need ...`

Similarly, there is `\needgraph`

- `\needgraph:`

A graph is needed here #1

- `\needgraph<About ...>:`

A graph is needed here #2: About ...

### 2.4 | Language configuration

You can use `\UseLanguage{(name of language)}` at any time to change the language, Language names include Chinese, English, French (the case of the first letter is arbitrary, for example, “chinese” is also acceptable). With this, the effects of various commands and environments will also change accordingly.

For example, after using `\UseLanguage{French}`, the theorem and the draft mark will be displayed as:

THÉORÈME 2.4 (Inutile) Un théorème en français. À terminer ici #3

When referenced, the name of the theorem always matches the language of the region in which the theorem is located, for example, the definition of the beginning is still displayed in English in the current French mode : DEFINITION 2.1 and THÉORÈME 2.4

```
%! TEX program = xelatex
\documentclass{lebhartfast}

\UseLanguage{French}

\begin{document}

\title{Titre}
\author{Nom}
\date{03 / 2021, Lieu}

\maketitle

%% Texte ici

\end{document}
```

(`\UseLanguage` can be placed either in the preamble or in the body part, and can be used repeatedly as needed)