colorist

WRITE YOUR ARTICLES OR BOOKS IN A COLORFUL WAY

Jinwen XU

ProjLib@outlook.com

October 2021, Paris

Abstract

colorist is a series of styles and classes for you to typeset your articles or books in a colorful manner. The original intention in designing this series was to write drafts and notes that look colorful yet not dazzling. With the help of the ProjLib toolkit, also developed by the author, the classes provided here have multi-language support, preset theorem-like environments with clever reference support, and many other functionalities. Notably, using these classes, one can organize the author information in the $\mathcal{AMS}$ fashion, makes it easy to switch to journal classes later for publication.

Finally, this documentation is typeset using the colorart class (with the option `allowbf`). You can think of it as a short introduction and demonstration.

Contents

Before you start

In order to use the package or classes described here, you need to:

- install TeX Live or MikTeX of the latest possible version, and make sure that `colorist` and `projlib` are correctly installed in your TEX system.
- download and install the required fonts if needed.
- be familiar with the basic usage of LATEX, and knows how to compile your document with pdfLATEX, XELATEX or LuaLATEX.

Corresponding to: colorist `2021/10/23`

1 Introduction

colorist is a series of styles and classes for you to typeset your articles or books in a colorful manner. The original intention in designing this series was 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 of the following classes; `colorart.cls` for typesetting articles and `colorbook.cls` for typesetting books. They compile with any major TEX engine, with native support to English, French, German, Italian, Portuguese (European and Brazilian) and Spanish 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 additionally have native support for Chinese, Japanese and Russian typesetting. On the other hand, they need to be compiled with XELATEX or LuaLATEX (not pdfLATEX).

With the help of the ProjLib toolkit, also developed by the author, the classes provided here have multi-language support, preset theorem-like environments with clever reference support, and many other functionalities such as draft marks, enhanced author information block, mathematical symbols and shortcuts, etc. Notably, using these classes, one can organize the author information in the $\mathcal{AMS}$ fashion, makes it easy to switch to journal classes later for publication. For more detailed information, you can refer to the documentation of ProjLib by running `texdoc projlib` in the command line.

2 Usage and examples

2.1 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}
```

Tip

You may wish to use lebhart or beaulivre instead, which should produce better result. All the examples later using colorart or colorbook can be adopted to lebhart and beaulivre respectively, without further modification.

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 are provided.

```
\documentclass{article} or \documentclass{book}
\usepackage{colorist}
```

2.2 Example - colorart

Let's first look at a complete example of colorart (the same works for lebhart).

```
1 \documentclass{colorart}
2 \usepackage{ProjLib}
3
4 \UseLanguage{French}
5
6 \begin{document}
7
8 \title{⟨title⟩}
9 \author{⟨author⟩}
```

```
\date{\PLdate{2022-04-01}}

\maketitle

\begin{abstract}
    Ceci est un résumé. \dnf<⟨some hint⟩>
\end{abstract}
\begin{keyword}
    AAA, BBB, CCC, DDD, EEE
\end{keyword}

\section{Un théorème}

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

\end{document}
```

If you find this example a little complicated, don't worry. Let's now look at this example piece by piece.

2.2.1 Initialization

```
\documentclass{colorart}
\usepackage{ProjLib}
```

Initialization is straightforward. The first line loads the document class colorart, and the second line loads the ProjLib toolkit to obtain some additional functionalities.

2.2.2 Set the language

```
\UseLanguage{French}
```

This line indicates that French will be used in the document (by the way, if only English appears in your article, then there is no need to set the language). You can also switch the language in the same way later in the middle of the text. Supported languages include Simplified Chinese, Traditional Chinese, Japanese, English, French, German, Spanish, Portuguese, Brazilian Portuguese and Russian[1].

For detailed description of this command and more related commands, please refer to the section on the multi-language support.

2.2.3 Title, author information, abstract and keywords

```
\title{⟨title⟩}
\author{⟨author⟩}
\date{\PLdate{2022-04-01}}
\maketitle

\begin{abstract}
    ⟨abstract⟩
\end{abstract}
```

[1]The language Simplified Chinese, Traditional Chinese, Japanese and Russian requires Unicode support, thus the classes lebhart or beaulivre.

```
\begin{keyword}
    ⟨keywords⟩
\end{keyword}
```

This part begins with the title and author information block. The example shows the basic usage, but in fact, you can also write:

```
\author{⟨author 1⟩}
\address{⟨address 1⟩}
\email{⟨email 1⟩}
\author{⟨author 2⟩}
\address{⟨address 2⟩}
\email{⟨email 2⟩}
...
```

In addition, you may also write in the $\mathcal{AMS}$ fashion, i.e.:

```
\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}}
\subjclass{*****}
\keywords{⟨keywords⟩}

\begin{abstract}
    ⟨abstract⟩
\end{abstract}

\maketitle
```

2.2.4 Draft marks

```
\dnf<⟨some hint⟩>
```

When you have some places that have not yet been finished yet, you can mark them with this command, which is especially useful during the draft stage.

2.2.5 Theorem-like environments

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

Commonly used theorem-like environments have been pre-defined. Also, when referencing a theorem-like environment, it is recommended to use `\cref{⟨label⟩}` — in this way, there is no need to explicitly write down the name of the corresponding environment every time.

Tip

If you wish to switch to the standard class later, just replace the first two lines with:

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

or to use the $\mathcal{AMS}$ class:

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

Tip

If you like the current document class, but want a more "plain" style, then you can use the option `classical`, like this:

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

2.3 Example - colorbook

Now let's look at an example of colorbook (the same works for beaulivre).

```
\documentclass{colorbook}
\usepackage{ProjLib}

\UseLanguage{French}

\begin{document}

\frontmatter

\begin{titlepage}
    ⟨code for titlepage⟩
\end{titlepage}

\tableofcontents

\mainmatter

\part{⟨part title⟩}
\parttext{⟨text after part title⟩}

\chapter{⟨chapter title⟩}

\section{⟨section title⟩}
```

```
...

\backmatter

...

\end{document}
```

There is no much differences with colorart, only that the title and author information should be typeset within the `titlepage` environment. Currently no default titlepage style is given, since the design of the title page is a highly personalized thing, and it is difficult to achieve a result that satisfies everyone.

In the next section, we will go through the options available.

3 The options

colorist offers the following options:

- The language options `EN` / `english` / `English`, `FR` / `french` / `French`, etc.
  - For the option names of a specific language, please refer to ⟨*language name*⟩ in the next section. The first specified language will be used as the default language.
  - The language options are optional, mainly for increasing the compilation speed. Without them the result would be the same, only slower.
- `draft` or `fast`
  - The option `fast` enables a faster but slightly rougher style, main differences are:
    - Use simpler math font configuration;
    - Do not use hyperref;
    - Enable the fast mode of ProjLib toolkit.

> **Tip**
>
> During the draft stage, it is recommended to use the `fast` option to speed up compilation. When in `fast` mode, there will be a watermark "DRAFT" to indicate that you are currently in the draft mode.

- `allowbf`
  - Allow boldface. When this option is enabled, the main title, the titles of all levels and the names of theorem-like environments will be bolded.
- `runin`
  - Use the "runin" style for `\subsubsection`
- `puretext` or `nothms`
  - Pure text mode. Do not load theorem-like environments.
- `delaythms`
  - Defer the definition of theorem-like environments to the end of the preamble. Use this option if you want the theorem-like environments to be numbered within a custom counter.
- `nothmnum`, `thmnum` or `thmnum`=⟨*counter*⟩
  - Theorem-like environments will not be numbered / numbered in order 1, 2, 3... / numbered within ⟨*counter*⟩. Here, ⟨*counter*⟩ should be a built-in counter (such as `subsection`) or a custom counter defined in the preamble (with the option `delaythms` enabled). If no option is used, they will be numbered within `chapter` (book) or `section` (article).
- `regionalref`, `originalref`
  - When referencing, whether the name of the theorem-like environment changes with the current language. The default is `regionalref`, *i.e.*, the name corresponding to the current language is used; for example, when referencing a theorem-like environment in English context, the names "Theorem, Definition..." will be used no matter which language context the original environment is in. If `originalref` is enabled, then the name will always remain the same as the original place; for example, when referencing a theorem written in the French context, even if one is currently in the English context, it will still be displayed as "Théorème".
  - In `fast` mode, the option `originalref` will have no effect.

Additionally, colorart and colorbook offers the following options:

- `a4paper` or `b5paper`
  - Optional paper size. The default paper size is 8.5in × 11in.

- `palatino`, `times`, `garamond`, `noto`, `biolinum` | `useosf`
  - Font options. As the name suggest, font with corresponding name will be loaded.
  - The `useosf` option is used to enable the old-style figures.

4 Instructions by topic

4.1 Language configuration

colorart has multi-language support, including English, French, German, Italian, Portuguese (European and Brazilian) and Spanish. The language can be selected by the following macros:

- `\UseLanguage{`⟨*language name*⟩`}` is used to specify the language. The corresponding setting of the language will be applied after it. It can be used either in the preamble or in the main body. When no language is specified, "English" is selected by default.
- `\UseOtherLanguage{`⟨*language name*⟩`}{`⟨*content*⟩`}`, which uses the specified language settings to typeset ⟨*content*⟩. Compared with `\UseLanguage`, it will not modify the line spacing, so line spacing would remain stable when CJK and Western texts are mixed.

⟨*language name*⟩ can be (it is not case sensitive, for example, `French` and `french` have the same effect):

- Simplified Chinese: `CN`, `Chinese`, `SChinese` or `SimplifiedChinese`
- Traditional Chinese: `TC`, `TChinese` or `TraditionalChinese`
- English: `EN` or `English`
- French: `FR` or `French`
- German: `DE`, `German` or `ngerman`
- Italian: `IT` or `Italian`
- Portuguese: `PT` or `Portuguese`
- Portuguese (Brazilian): `BR` or `Brazilian`
- Spanish: `ES` or `Spanish`
- Japanese: `JP` or `Japanese`
- Russian: `RU` or `Russian`

In addition, you can also add new settings to selected language:

- `\AddLanguageSetting{`⟨*settings*⟩`}`
  - Add ⟨*settings*⟩ to all supported languages.
- `\AddLanguageSetting(`⟨*language name*⟩`){`⟨*settings*⟩`}`
  - Add ⟨*settings*⟩ to the selected language ⟨*language name*⟩.

For example, `\AddLanguageSetting(German){\color{orange}}` can make all German text displayed in orange (of course, one then need to add `\AddLanguageSetting{\color{black}}` in order to correct the color of the text in other languages).

4.2 Theorems and how to reference them

Environments such as `definition` and `theorem` have been preset and can be used directly.

More specifically, preset environments include: `assumption`, `axiom`, `conjecture`, `convention`, `corollary`, `definition`, `definition-proposition`, `definition-theorem`, `example`, `exercise`, `fact`, `hypothesis`, `lemma`, `notation`, `observation`, `problem`, `property`, `proposition`, `question`, `remark`, `theorem`, and the corresponding unnumbered version with an asterisk `*` in the name. The titles will change with the current language. For example, `theorem` will be displayed as "Theorem" in English mode and "Théorème" in French mode.

When referencing a theorem-like environment, it is recommended to use `\cref{`⟨*label*⟩`}`. In this way, there is no need to explicitly write down the name of the corresponding environment every time.

Example

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

will produce

Definition 4.1 (Strange things) This is the definition of some strange objects. There is approximately a one-line spacing before and after the theorem environment, and there will be a symbol to mark the end of the environment.

`\cref{def: strange}` will be displayed as: **Definition 4.1**.
After using `\UseLanguage{French}`, a theorem will be displayed as:

Théorème 4.2 (Inutile) Un théorème en français.

By default, when referenced, the name of the theorem matches the current context. For example, the definition above will be displayed in French in the current French context : **Définition 4.1** et **Théorème 4.2**. If you want the name of the theorem to always match the language of the context in which the theorem is located, you can add `originalref` to the global options.

The following are the main styles of theorem-like environments:

Theorem 4.3 Theorem style: theorem, proposition, lemma, corollary, ...

Proof | Proof style

Remark style

Conjecture 4.4 Conjecture style

Example Example style: example, fact, ...

Problem 4.5 Problem style: problem, question, ...

For aesthetics, adjacent definitions will be connected together automatically:

Definition 4.6 First definition.

Definition 4.7 Second definition.

4.3 Define a new theorem-like environment

If you need to define a new theorem-like environment, you must first define the name of the environment in the language to use:

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

For ⟨*language name*⟩, please refer to the section on language configuration. When ⟨*language name*⟩ is not specified, the name will be set for all supported languages. In addition, environments with or without asterisk share the same name, therefore, `\NameTheorem{envname*}{...}` has the same effect as `\NameTheorem{envname}{...}`.

And then define this environment in one of following five ways:

- \CreateTheorem*{⟨*name of environment*⟩}
  - Define an unnumbered environment ⟨*name of environment*⟩
- \CreateTheorem{⟨*name of environment*⟩}
  - Define a numbered environment ⟨*name of environment*⟩, numbered in order 1,2,3,...
- \CreateTheorem{⟨*name of environment*⟩}[⟨*numbered like*⟩]
  - Define a numbered environment ⟨*name of environment*⟩, which shares the counter ⟨*numbered like*⟩
- \CreateTheorem{⟨*name of environment*⟩}<⟨*numbered within*⟩>
  - Define a numbered environment ⟨*name of environment*⟩, numbered within the counter ⟨*numbered within*⟩
- \CreateTheorem{⟨*name of environment*⟩}(⟨*existed environment*⟩)
  \CreateTheorem*{⟨*name of environment*⟩}(⟨*existed environment*⟩)
  - Identify ⟨*name of environment*⟩ with ⟨*existed environment*⟩ or ⟨*existed environment*⟩*.
  - This method is usually useful in the following two situations:
    1) To use a more concise name. For example, with \CreateTheorem{thm}(theorem), one can then use the name thm to write theorem.
    2) To remove the numbering of some environments. For example, one can remove the numbering of the remark environment with \CreateTheorem{remark}(remark*).

Tip

This macro utilizes the feature of amsthm internally, so the traditional theoremstyle is also applicable to it. One only needs declare the style before the relevant definitions.

Here is an example. The following code:

```
\NameTheorem[EN]{proofidea}{Idea}
\CreateTheorem*{proofidea*}
\CreateTheorem{proofidea}<subsection>
```

defines an unnumbered environment proofidea* and a numbered environment proofidea (numbered within subsection) respectively. They can be used in English context. The effect is as follows:

Idea | The proofidea* environment.

Idea 4.3.1 | The proofidea environment.

4.4 Draft mark

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

- \dnf or \dnf<...>. The effect is: To be finished #1 or To be finished #2 : ... .
  The prompt text changes according to the current language. For example, it will be displayed as Pas encore fini #3 in French mode.

Similarly, there is \needgraph :

- \needgraph or \needgraph<...>. The effect is:

A graph is needed here #1

or

A graph is needed here #2 : ...

The prompt text changes according to the current language. For example, in French mode, it will be displayed as

```
Il manque une image ici #3
```

4.5 Title, abstract and keywords

colorart has both the features of standard classes and that of the $\mathcal{AMS}$ classes.

Therefore, the title part can either be written in the usual way, in accordance with the standard class article:

```
\title{⟨title⟩}
\author{⟨author⟩\thanks{⟨text⟩}}
\date{⟨date⟩}
\maketitle
\begin{abstract}
    ⟨abstract⟩
\end{abstract}
\begin{keyword}
    ⟨keywords⟩
\end{keyword}
```

or written in the way of $\mathcal{AMS}$ classes:

```
\title{⟨title⟩}
\author{⟨author⟩}
\thanks{⟨text⟩}
\address{⟨address⟩}
\email{⟨email⟩}
\date{⟨date⟩}
\keywords{⟨keywords⟩}
\subjclass{⟨subjclass⟩}
\begin{abstract}
    ⟨abstract⟩
\end{abstract}
\maketitle
```

The author information can contain multiple groups, written as:

```
\author{⟨author 1⟩}
\address{⟨address 1⟩}
\email{⟨email 1⟩}
\author{⟨author 2⟩}
\address{⟨address 2⟩}
\email{⟨email 2⟩}
...
```

Among them, the mutual order of \address, \curraddr, \email is not important.

5 Known issues

- The font settings are still not perfect.
- The TOC design does not look very nice.
- Since many features are based on the ProjLib toolkit, colorist (and hence colorart, lebhart and colorbook, beaulivre) inherits all its problems. For details, please refer to the "Known Issues" section of the ProjLib documentation.
- The error handling mechanism is incomplete: there is no corresponding error prompt when some problems occur.
- There are still many things that can be optimized in the code.