

cltart 使用说明

cltian¹——cl.tian@qq.com

1 cltart 文档类简介

“cltart”是为 CTEX 的 article 类定制的一个文档类，用于简化常见中文文档书写。使用很简单：

```
\documentclass[arg1, arg2, ..., argN]{cltart}
```

具体的页面配置如下：

- 使用 A4 页面，行间距 1.5 倍，段前段后 0 磅，section 段前 24 磅、段后 6 磅，subsection 段前 12 磅、段后 6 磅，subsection 段前 12 磅、段后 6 磅
- 标题三号黑体加粗居中，信息五号黑体加粗居中，章节黑体小四黑体左顶格²
- 正文中文使用宋体小四，英文使用 Times New Roman 小四。
- 页眉页脚使用五号字
- caption 段前 6 磅，段后 0 磅，使用五号字
- 参考文献提供 GBT-7714 和 IEEE Trans 格式，有中文建议使用 GBT-7714 格式。文档使用五号字，1.5 倍行距
- 对浮动体进行设置，使之排版更紧密

2 预定义命令和环境

CTEX 中默认使用黑体表示加粗，使用楷体表示斜体，这与 Microsoft Word 不同，cltart 使用 xeLaTeX 的 AutoFakeBold 和 AutoFakeSlant 加以设置，使之与 Microsoft Word 显示类似。如要取消此功能，即用黑体表示加粗，使用楷体表示斜体，在导言区使用\nofake 命令³。如：

```
\documentclass[super,list,bibieee,myhdrone,table,math]{cltart}
\nofake
\begin{document}
...
...
\end{document}
```

关于文档开头的命令。cltart 摒弃了 LaTeX 内置的生成标题、作者、日期等的命令。自建命令完成这些功能，这些命令可写可不写，不写则无此信息。其中，\clttitle 命令和\cltinfo 命令请注意先后顺序。

¹cltian 个人主页：<http://github.com/foowaa/documentation>

²请确定宋体、黑体、楷体和 Times New Roman 已安装。中文字体问题请参看《CTeX 宏集手册》。

³详细的字体设置：<https://en.wikibooks.org/wiki/LaTeX/Fonts>和<https://github.com/wklchris/Note-by-LaTeX>

- `\clttitle` 命令，用在开头，生成标题。如果不写则无标题。
- `\cltinfo` 命令，用在开头，生成相关信息（如作者，联系方式，日期等）。如果不写则无信息。
- `\cltheading` 命令，用在开头，如传入 `myhdrone`, `myhdrtwo`，则必须设置，其为页眉信息。请参看节-4。

如本文开头是这样写的：

```
\documentclass[super,list,bibieeee,myhdrone,table,math]{cltart}
\begin{document}
\cltheading{cltart使用说明by\ cltian}
\clttitle{cltart使用说明}
\cltinfo{cltian——tianchunlin123@gmail.com}
...
...
\end{document}
```

便捷的空格、换行、换页命令：

- `\smallblank`，可传入参数 `N`，会生成 `N` 个 `1/3em` 的空格。
- `\bigblank`，可传入参数 `N`，会生成 `N` 个 `1em` 的空格。
- `\nextline`，可传入参数 `N`，会生换行 `N` 次，换行之后文字不缩进。（适用于文字之间）
- `\nextlineindent`，可传入参数 `N`，会生换行 `N` 次，换行之后文字缩进。（适用于文字之间）
- `\blankline`，可传入参数 `N`，会产生 `N` 个空白行。（适用于不同元素之间）
- `\blankpage`，可传入参数 `N`，另起一页，并生成 `N` 个空白页。
- `\nextpage`，无参数，另起一页。

预定义 `\famousquote` 环境，此环境有 2 个参数，其中第一个参数表示名人的姓名，第二个参数表示出处。例如：

归去来兮，田园将芜胡不归。既自以心为形役，奚惆怅而独悲。悟已往之不谏，知来者之可追。识迷途其未远，觉今是而昨非。

—— 陶渊明
《归去来兮辞》

3 参考文献选项

如果要使用参考文献，请**务必**传入参数 `<numbers|super|authoryar>`，如要使用 IEEE Trans 参考文献格式，使用参数 `bibieeee`，否则默认使用 GBT-7714。传递参数方法如下，之后不再赘述。

```
\documentclass[super,bibieeee]{cltart}
```

3.1 引用格式

- numbers: [1]
- super: 上标 [1]
- authoryear: (Jones, 1995)

引用使用\cite,例如 如文献^[1,2]。如要在 super 中使用 numbers,使用命令\citens,例如 如文献 [1, 2]

3.2 参考文献格式

默认使用 GBT-7714 格式,如想使用 IEEE Trans 格式,请传入 bibieee 参数(注意,只有传入 <numbers|super|authoryar> 后,传入 bibieee 才是有效的)。cltart 对 IEEEbib.bst 做了一些更改,使之可以产生和 IEEE Trans 上一样的参考文献格式而不必引入 IEEETrans.cls。注意:IEEE Trans 格式比较适用于参考文献为英文论文,否则不要使用。

4 页眉页脚

采用 fancyhdr 包,我们预定义了 6 种页眉页脚格式,分别为: myhdrone, myhdrtwo, myhdrthree, myhdrfour, myhdrfive 和默认,不设置即为默认格式。

- myhdrone——必须设置\cltheading,页眉左侧为\cltheading 的内容,右侧为页码。
- myhdrtwo——必须设置\cltheading,页眉居中为\cltheading 的内容,页脚居中为页码。
- myhdrthree——页眉左侧为章节号和章节标题,右侧为页码。
- myhdrfour——页眉居中为章节号和章节标题,页脚居中为页码。
- myhdrfive——页眉页脚为空。
- 默认——页眉为空,页脚居中为页码。

5 list 参数

引入 verbatim, listings, tcolorbox, salgpseudocode, algorithm, algorithmicx 包,并进行了一些配置,可以支持 verbatim,源代码和伪代码。此外针对 lstlistings 中没有 Javascript 支持的问题,cltart 已添加支持,名称为“JS”。效果如下:

```

1 typedef struct ImageData {
2     ImageData() {
3         data = nullptr;
4         width = 0;
5         height = 0;
6         num_channels = 0;
7     }
8
9     ImageData(int32_t img_width, int32_t img_height,
10             int32_t img_num_channels = 1) {
11         data = nullptr;

```

```

12     width = img_width;
13     height = img_height;
14     num_channels = img_num_channels;
15 }
16
17 uint8_t* data;
18 int32_t width;
19 int32_t height;
20 int32_t num_channels;
21 } ImageData;
22
23 typedef struct {
24     double x;
25     double y;
26 } FacialLandmark;
27
28 uint8_t* data;
29 int32_t width;
30 int32_t height;
31 int32_t num_channels;
32 } ImageData;
33
34 typedef struct {
35     double x;
36     double y;
37 } FacialLandmark;
38 }

```

在传入 list 参数后，可以继续传入“**precolorbox**”参数（注意：在传入 list 参数下，才能传入此参数）以使用预设的颜色框⁴。预设的框体主要有 4 种：

- 文字框。
- 带标题的可分页句段框。
- 无标题的可分页句段框。
- 代码框。

文字框预设的样式有 2 种，颜色有 5 种，所以共有 10 种框。示例如下：

`\boxonegreen` 这是 `\boxonegreen`, `\boxoneviolet` 这是 `\boxoneviolet`, `\boxonered`
 这是 `\boxonered`, `\boxoneyellow` 这是 `\boxoneyellow`, `\boxoneblue` 这是 `\boxoneblue`,
`\boxtwogreen` (这是 `\boxtwogreen`), `\boxtwoviolet` (这是 `\boxtwoviolet`), `\boxtwored`
 (这是 `\boxtwored`), `\boxtwoyellow` (这是 `\boxtwoyellow`), `\boxtwoblue` (这是 `\boxtwoblue`)。

带标题的可分页句段框有 5 种，分别为：`\parboxbluetitle`, `\parboxredtitle`, `\parboxyellowtitle`, `\parboxgreentitle`, `\parboxvoiletttitle`，调用格式如：

```

\begin{parboxyellowtitle}{标题}
内容
\end{parboxyellowtitle}

```

效果如下：



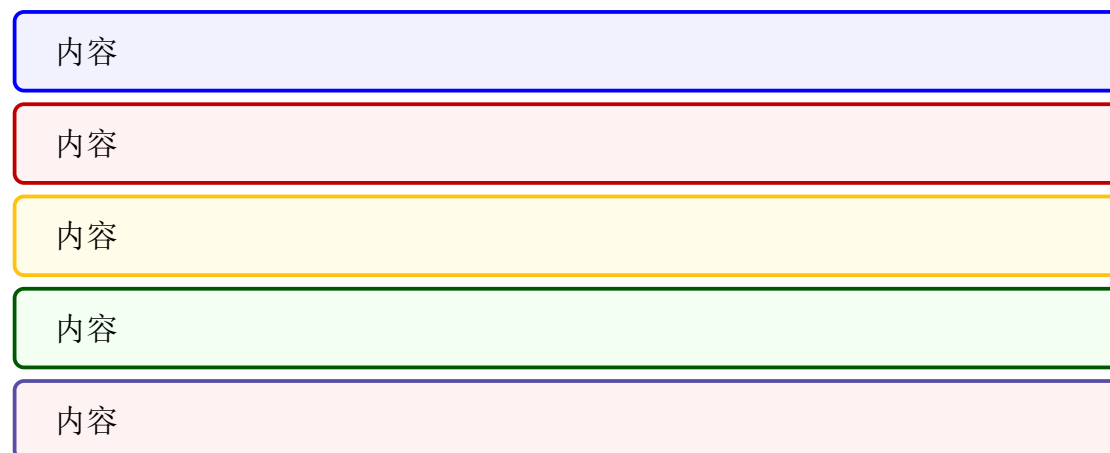
⁴`tcolorbox` 的详细使用见<https://mirror.hmc.edu/ctan/macros/latex/contrib/tcolorbox/tcolorbox.pdf>



无标题的可分页句段框有 5 种，分别为：`\parboxbluenotitle`，`\parboxrednotitle`，`\parboxyellownotitle`，`\parboxgreennotitle`，`\parboxvoiletnotitle`，调用格式如：

```
\begin{parboxbluenotitle}
内容
\end{parboxbluenotitle}
```

效果如下：



在 `cltart` 中定义了各种语言的代码框，以 `tcb{lstlistings_name}` 命名，定义的 `{lstlistings_name}` 有 Ada, Objective (即 objective C), Sharp (即 C#), C, C++, Assembler, Awk, Caml, bash, Delphi, erlang, Fortran, Gnuplot, Haskell, HTML Lingo, Java, Scala, Lua, make, Lisp, Matlab, Mathematica, Octave, Pascal, Perl, PHP, PostScript, Prolog Python, Ruby, SQL, SAS, R, Tex, LaTeX, VBScript, Verilog, VHDL, XML, JS, 使用如下：

```
\begin{tcbC++}
typedef struct ImageData {
    ImageData() {
        data = nullptr;
        width = 0;
        height = 0;
    }
};
```

```

        num_channels = 0;
    }
\end{tcbC++}

```

效果如下：

```

1 typedef struct ImageData {
2     ImageData() {
3         data = nullptr;
4         width = 0;
5         height = 0;
6         num_channels = 0;
7     }
8
9     ImageData(int32_t img_width, int32_t img_height,
10             int32_t img_num_channels = 1) {
11         data = nullptr;
12         width = img_width;
13         height = img_height;
14         num_channels = img_num_channels;
15     }
16
17     uint8_t* data;
18     int32_t width;
19     int32_t height;
20     int32_t num_channels;
21 } ImageData;
22
23 typedef struct {
24     double x;
25     double y;
26 } FacialLandmark;
27
28     uint8_t* data;
29     int32_t width;
30     int32_t height;
31     int32_t num_channels;
32 } ImageData;
33
34 typedef struct {
35     double x;
36     double y;
37 } FacialLandmark;
38 }

```

6 杂项

math。引入了 amsmath, mathtools, amsfons, amssymb, mathdots, mathrsfs, yhmath, esint, extarrows 包方便高级数学公式书写。例如：

Pascal' s rule is

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

color。引入了 xcolor 包⁵。传入 color 参数相当于：

⁵具体使用参见<https://en.wikibooks.org/wiki/LaTeX/Colors>

figure。默认导入 `graphicx` 和 `caption` 包，并进行了图片样式的调整。如要使用子图和双语图等高级功能（引入 `subcaption`, `bicaption`, `float`, `placeins`, `overpic` 包），请传入 `figure` 选项⁹。

一定会导入的包。`graphicx`, `caption`, `geometry`, `indentfirst`, `ulem`（下划线包），`setspace`, `fontspec`

中文复制乱码。如遇中文复制乱码，在`\{documentclass}` 之后，另起一行，写一个全角空格，再另起一行，正常书写¹⁰。

参考文献

- [1] C. Tian, Y. Yuan, and X. Lu, “Deep temporal architecture for audiovisual speech recognition,” in *CCF Chinese Conference on Computer Vision*. Springer, 2017, pp. 650–661.
- [2] Y. Yuan, C. Tian, and X. Lu, “Auxiliary loss multimodal gru model in audio-visual speech recognition,” *IEEE Access*, vol. 6, pp. 5573–5583, 2018.

⁹详情使用见https://en.wikibooks.org/wiki/LaTeX/Importing_Graphics和https://en.wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions

¹⁰[xelatex 编译中文 pdf，复制乱码，求解决？ - 知乎](#)