

# zL<sup>A</sup>T<sub>E</sub>X 用户手册

Eureka

2025 年 1 月 14 日

## CONTENTS NEW

<b>1</b>	<b>基本介绍</b>	<b>3</b>	<b>6.2</b>	<b>indexref . . . . .</b>	<b>13</b>
<b>2</b>	<b>宏包机制</b>	<b>3</b>	<b>6.3</b>	<b>layout . . . . .</b>	<b>15</b>
	2.1 基本宏包 . . . . .	3	<b>6.4</b>	<b>pageinfo . . . . .</b>	<b>16</b>
<b>3</b>	<b>安装使用</b>	<b>4</b>	<b>6.5</b>	<b>theme . . . . .</b>	<b>18</b>
	3.1 在线使用 . . . . .	4	<b>6.6</b>	<b>thm . . . . .</b>	<b>20</b>
	3.2 本地安装 . . . . .	4	<b>6.7</b>	<b>titlesec . . . . .</b>	<b>27</b>
	3.3 最小工作示例 . . . . .	5	<b>6.8</b>	<b>toc . . . . .</b>	<b>28</b>
<b>4</b>	<b>zL<sup>A</sup>T<sub>E</sub>X 配置</b>	<b>6</b>	<b>7</b>	<b>Libraries</b>	<b>29</b>
	4.1 导言区 . . . . .	6	<b>7.1</b>	<b>fancy . . . . .</b>	<b>29</b>
<b>5</b>	<b>杂项</b>	<b>11</b>	<b>7.2</b>	<b>mathalias . . . . .</b>	<b>30</b>
<b>6</b>	<b>Modules</b>	<b>12</b>	<b>7.3</b>	<b>slide . . . . .</b>	<b>35</b>
	6.1 fontcfg . . . . .	12	<b>7.4</b>	<b>theme . . . . .</b>	<b>39</b>
			<b>Index</b>		<b>43</b>

## 1 基本介绍

---

`\zLaTeX`

---

`\zLaTeX`

Updated: 2024-11-05

---

用于输出本宏集对应的 logo，使用示例如下：

Hello `\zLaTeX{}`；你好 `\zLaTeX{}`。

-----  
Hello zL<sup>A</sup>T<sub>E</sub>X；你好 zL<sup>A</sup>T<sub>E</sub>X。

zL<sup>A</sup>T<sub>E</sub>X 文档类默认基于 article 文档类，但是你仍然可以在加载本文档类时选择加载其他的文档类，通过设置选项 `<class>` 的值为 article 或者是 ctexbook 即可。通过更换默认的文档类，从而满足使用者的不同需求，目前本模板可以用于以下场景：

- 撰写书籍或者笔记
- 讨论班的 Slide 制作

zL<sup>A</sup>T<sub>E</sub>X 的制作初衷：让使用者可以方便进行书籍和笔记的撰写以及日常汇报 slide 的无缝切换。zL<sup>A</sup>T<sub>E</sub>X 全部由 L<sup>A</sup>T<sub>E</sub>X3 进行编写，采用 `<key-value>` 的方式进行选项和命令的配置，对于作者来说：方便后续的模板拓展和维护；对于用户来说：使用键值对可以减轻用户记忆命令参数这一负担，方便用户使用命令。如果使用者熟悉 L<sup>A</sup>T<sub>E</sub>X，那么花费不到 10min 的时间，你便可以轻松使用本文档类完成如上任务，减少不必要的工作。

## 2 宏包机制

---

zslide-last-page

---

`\pageref{zslide-last-page}`

Updated: 2024-11-05

---

引用当前文档的最后一页，用于 slide 制作时的页码引用。

zL<sup>A</sup>T<sub>E</sub>X 文档类会根据用户指定的选项自动处理和加载对应的宏包，所以 zL<sup>A</sup>T<sub>E</sub>X 文档类在不同的导言区选项声明下加载的宏包和命令是不同的。后文详细地介绍了不同导言区配置以及不同编译引擎下的宏包加载情况。zL<sup>A</sup>T<sub>E</sub>X 始终秉持着最少依赖的原则，能够自己实现的功能，尽量不引入宏包。如部分用户会用到的 lastpage 宏包提供的 `\pageref{LastPage}` 已实现为：`\pageref{zslide-last-page}`（需在 `<slide>=true` 时使用）。

### 2.1 基本宏包

zL<sup>A</sup>T<sub>E</sub>X 会加载一系列的基本宏包，意味着无论用户的导言区如何配置，这部分宏包均会被加载。具体的宏包加载情况如下：

zL<sup>A</sup>T<sub>E</sub>X 默认只加载很少的一部分基础宏包，用户如果想要实现更加个性化的功能还请自行引入相关宏包；在默认情况下本模板即可呈现一个比较好的效果，不熟

l3keys2e	geometry	fancyhdr	graphicx
amsmath	amsfonts	esint	framed
graphicx	xcolor	cleveref	sidenotes
titlesec	titletoc		

表 1: zL<sup>A</sup>T<sub>E</sub>X 文档类基本宏包

悉 L<sup>A</sup>T<sub>E</sub>X 的用户不用担心本模板配置选项过于复杂. 想要马上开始体验? 请参见“[小节 3.3](#)”的最小写作示例.

## 3 安装使用

### 3.1 在线使用

为了让部分用户可以直接体验到 zL<sup>A</sup>T<sub>E</sub>X, 免去“繁杂”的环境配置. 我已将本模板部署在 Overleaf 上, 地址为: [TeXPgae zL<sup>A</sup>T<sub>E</sub>X Project](#), 直接打开此地址即可体验. Github 上的项目内容包含本手册以及 zTikZ 文档的源码与文档. 由于部分的技术原因, zTikZ 请在本地体验.

### 3.2 本地安装

目前本文档类 zL<sup>A</sup>T<sub>E</sub>X 还没有登陆 CTAN, 未来可能也没有这个打算, 本模板还没有完全开发完成. 由于本文档类使用的部分 L<sup>A</sup>T<sub>E</sub>X3 命令在老版本下并不存在, 所以如果你的 T<sub>E</sub>XLive 过于老旧, 则可能出现无法编译的情况. 目前已知 zL<sup>A</sup>T<sub>E</sub>X 文档类在各平台的兼容情况为:

Windows : T<sub>E</sub>XLive 最低版本 2022

Linux : T<sub>E</sub>XLive 最低版本 2022

MacOS : 兼容 MacT<sub>E</sub>X2024(老版也应兼容)

---

ztool

---

\usepackage{ztool}

Updated: 2024-12-05

---

本宏集以独立实现了一个 ztool 模块, 此模块中包含原来已被废弃的 l3sys-shell 中的所有命令. ztool 实现了 box 以及 文件 IO 操作相关的函数. 在 ztool 的协助下, zL<sup>A</sup>T<sub>E</sub>X 能够较少-shell-escape 的调用.

由于 zL<sup>A</sup>T<sub>E</sub>X 还没有传入 CTAN(未来可能会考虑), 所以想要使用此文档类, 可以有如下的两种方法:

- 把此文档类 - zlatex 目录中的所有内容放入当前项目文件夹下
- 在命令行运行命令: `kpsewhich -var-value=TEXMFHOME`, 在 Windows 上一般是: `C:/Users/\meta{name}/texmf/`, 在 Linux 下一般是 `~/texmf/`; 具体路

径以自己的实际情况为准. 在此路径下新建文件夹 `tex/latex/zlatex`, 此文件夹对应的路径记为  $\langle z\text{\LaTeX}\rangle$ ; 然后把 `zlatex` 目录中的所有内容放入  $\langle z\text{\LaTeX}\rangle$  下.

在本手册后续, 我们使用  $\langle z\text{\LaTeX}\rangle$  表示本宏集的根本目录.

### 3.3 最小工作示例

`z\LaTeX` 的最小工作示例如下<sup>1</sup>. 首先是中文写作示例, 默认加载 `article` 文档类, 如果喜欢使用 `book` 文档类, 可以在加载文档类时指定 `class=book`.

```
% compile engine: xelatex
\documentclass[lang=cn]{zlatex}

\begin{document}
% some preface
% \tableofcontents

% wrting your document here ...
\end{document}
```

其次是英文写作示例 (此时为 `book` 文档类), 你需要修改的地方只有两处; 首先就是把语言选项改为 `lang=en`, 其次便是把编译方式改为 `pdf\LaTeX`.

```
% compile engine: pdflatex
\documentclass[class=book]{zlatex}

\title{\{title\}}
\author{\{author\}}
\date{\{date\}}
\begin{document}
\maketitle
\frontmatter
% some preface
% \tableofcontents
% some claim etc.
\mainmatter

% wrting your document here ...
\end{document}
```

在使用 `book` 文档类时, 如果不加载 `\frontmatter` 和 `\mainmatter` 两命令, 那么可能会导致整个文档的页眉, 页脚格式不正确.

---

<sup>1</sup> 导言区的配置可能需要根据自己的实际情况加以调整, 详细配置请参见后文

## 4 zL<sup>A</sup>T<sub>E</sub>X 配置

### 4.1 导言区

---

`\zlatexSetup`    `\zlatexSetup{⟨key-value⟩}`

---

Updated: 2024-11-05

---

zL<sup>A</sup>T<sub>E</sub>X 接受一系列的键值对进行配置，部分的配置尽可以在加载文档类时指定。为区分二者，我们约定：

- 名字后带有 ☆ 号的选项，只能作为宏包/文档类选项，需要在引入宏包/文档类的时候指定；
- 名字后带有 ★ 号的选项，只能通过 zL<sup>A</sup>T<sub>E</sub>X 宏集提供的用户接口 `\zlatexSetup` 来设定
- 名字后不带有特殊符号的选项，既可以作为宏包/文档类选项，也可以通过 `\zlatexSetup` 来设定。

---

`\zlatexOptions`    `\zlatexOptions`

---

Updated: 2024-11-05

---

zL<sup>A</sup>T<sub>E</sub>X 内置命令，用于打印此时文档类 zL<sup>A</sup>T<sub>E</sub>X 接收到的选项，可以在调试模板时使用。使用样例：

```
\zlatexOptions
```

---

**Class Options:** cn - oneside - 12pt

---

zL<sup>A</sup>T<sub>E</sub>X 的配置选项可以在文档类加载时指定，也可以通过命令 `\zlatexSetup` 进行指定。zL<sup>A</sup>T<sub>E</sub>X 的 `⟨key-value⟩` 被划分为两个层级，第一层的 `⟨key⟩` 为：`⟨layout⟩`，`⟨mathSpec⟩`，`⟨font⟩`，`⟨bib_index⟩`，`⟨toc⟩`，`⟨packageOption⟩`，`⟨classOption⟩`，`⟨section⟩`，`⟨lang⟩`，`⟨toc⟩`。其中前 7 个键 `⟨key⟩` 均具有自己的独立子键 `⟨sub-key⟩`；后两者不具有自己的子键，直接指定即可。关于各层 `⟨key-value⟩` 的关系请见图 1。

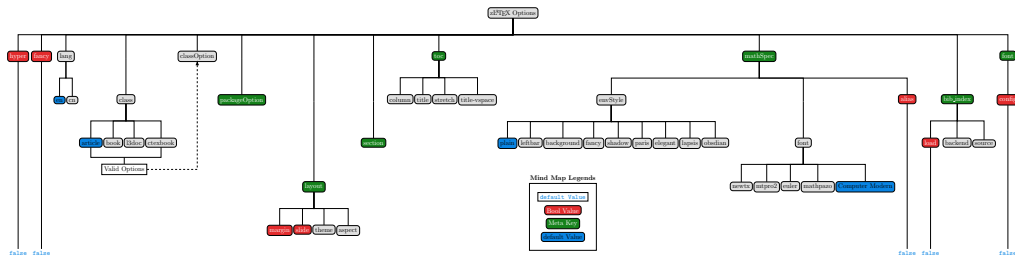


图 1: zL<sup>A</sup>T<sub>E</sub>X 文档类选项示意图

总体而言，zL<sup>A</sup>T<sub>E</sub>X 的文档类选项是比较复杂的，对于刚接触本文档类的用户，不必知晓所有的选项配置，因为在默认的配置下，zL<sup>A</sup>T<sub>E</sub>X 已经能够得到一个效果较佳的文档了。下面我们将详细介绍各个 `⟨key⟩` 的指定方式及其具体含义。

---

<code>\zlatexloadmodule</code>	<code>\zlatexloadmodule{&lt;module name&gt;}</code>
<code>\zlatexloadlibrary</code>	<code>\zlatexloadlibrary{&lt;library name&gt;}</code>

---

Updated: 2024-11-05

可以使用这两个命令用于加载 z<sub>La</sub>T<sub>E</sub>X 的模块和库，所有的 module 默认全部加载，library 默认全部不加载，由用户指定加载。目前已有的 module 和 library 列表如下：

**module 列表:****library 列表:**

- |   |   |
|---|---|
| • <code>zlatex.module.fontcfg.tex</code>  | • <code>zlatex.library.fancy.tex</code>     |
| • <code>zlatex.module.indexref.tex</code> | • <code>zlatex.library.mathalias.tex</code> |
| • <code>zlatex.module.layout.tex</code>   | • <code>zlatex.library.slide.tex</code>     |
| • <code>zlatex.module.pageinfo.tex</code> | • <code>zlatex.library.theme.tex</code>     |
| • <code>zlatex.module.theme.tex</code>    |   |
| • <code>zlatex.module.thm.tex</code>      |   |
| • <code>zlatex.module.titlesec.tex</code> |   |
| • <code>zlatex.module.toc.tex</code>      |   |

各个 module 和 library 的加载方式请参见如下示例：

```
% \documentclass{zlatex}
\zlatexloadlibrary{{fancy}}
\zlatexloadlibrary{{mathalias}}
\zlatexloadlibrary{{slide}}
\zlatexloadlibrary{{theme}}
```

你当然可以编写一个你自己的 module，不妨假设其名称为 `<moduleA>`；将此文件命名为 `zlatex.module.<moduleA>.tex`，然后将其放入路径 `<zLaTEX>/module/` 下，最后使用如下命令加载此 module：

```
\zlatexloadmodule{<moduleA>}
```

---

<code>lang</code>	☆ <code>lang=&lt;cn, en&gt;</code>
-------------------	------------------------------------

---

Updated: 2024-11-05

z<sub>La</sub>T<sub>E</sub>X 目前仅对中英文做了适配，对于法语仅有部分支持。根据不同的文档类语言，z<sub>La</sub>T<sub>E</sub>X 会加载不同的和语言相关的宏包，不同 `<lang>` 设置下，宏包的加载情况为：

<code>lang=en</code>	<code>inputenc(pdf<sub>tex</sub>)</code>	<code>fontenc</code>	<code>babel</code>	<code>microtype</code>
<code>lang=cn</code>	<code>fontspec</code>	<code>ctex</code>		

表 2: z<sub>La</sub>T<sub>E</sub>X 文档类语言宏包

---

**hyper** ☆ `hyper=<true, false>`

---

Updated: 2024-11-05

---

是否开启文档内部的超链接以及 PDF 书签，默认为 **false**。建议在最后的成稿中启用此选项，在草稿阶段置为 **false** 可以加快文档的编译速度。

---

**fancy** ☆ `fancy=<true, false>`

---

Updated: 2024-11-05

---

此选项用于控制文档的外观，包括章节样式，定理类环境样式，默认为 **false**。

---

**class** ☆ `class=<article, book, ctexbook>`

---

Updated: 2024-11-05

---

此选项用于指定加载的基文档类，默认为 **article**。在加载 **ctexbook** 文档类后可以使用此文档类提供的 `\ctexset{}` 命令进行相关选项的配置。

---

**classOption** ☆ `classOption=<class options>`

---

Updated: 2024-11-05

---

此选项接受一个键值对，用于传递基文档类选项，针对默认的 **article** 文档类，此项为 **oneside**, **12pt**。一个简单的设置样例：

```
\documentclass[
  class=article,
  classOption={10pt, leqno, a4paper},
]{zlatex}
```

---

**packageOption** ☆ `packageOption=<key-value>`

---

Updated: 2024-11-20

---

此选项接受一个键值对，用于给不同宏包传递选项，使用样例：

```
\documentclass[
  packageOption={
    fontspec=quiet,
    ctex={scheme=plain, punct=quanjiao},
  },
]{zlatex}
```

---

**toc** `toc=<key-value>`

---

Updated: 2024-12-25

---

此选项用于设置目录的样式，所有可选配置项如下：

```
\zlatexSetup{
  toc={
    column=<int:1>,
    title=<tl:Contents>,
    title-vspace=<dim:-2em>,
    stretch=<float:1>
  }
}
```



若上述的  $\langle column \rangle \geq 2$ , 那么 zL<sup>A</sup>T<sub>E</sub>X 会自动加载 multicol 宏包. 注意: 由于在 L<sup>A</sup>T<sub>E</sub>X 中, 文档类选项不能含有控制序列, 所以如果上述的  $\langle toc \rangle$  的某一个子项中含有控制序列, 那么务必通过命令 `\zlatexSetup` 进行设置, 示例如下:

```
\zlatexSetup{
  toc={
    title=\hfill\large\normalfont CONTENTS {\sffamily\small NEW}\hfill
  }
}
```

---

**font** ☆

**font** =  $\langle true, false \rangle$ 


---

Updated: 2024-12-06

此选项目前在实验性阶段, 主要用于字体配置, 默认为 `false`. 在启用此选项后, zL<sup>A</sup>T<sub>E</sub>X 会自动加载 fontspec 宏包, 此时需更换引擎为 X<sub>L</sub><sup>A</sup>T<sub>E</sub>X 或者 Lua<sup>A</sup>T<sub>E</sub>X.

---

**layout** ☆

**layout** =  $\langle key-value \rangle$ 


---

Updated: 2024-11-05

设置文档布局, 所有可选配置项如下:

```
\documentclass[
  layout={
    margin=\bool{false},
    slide=\bool{false},
    aspect=\float/float{12:9},
    theme=\str{AnnArborDefault}
  },
]{zlatex}
```

在加载 `slide` library 后, 如果设置  $\langle slide=true \rangle$ , 那么此时即可把文档转为 `slide` 模式。

---

**bib\_index**
**bib\_index** =  $\langle key-value \rangle$ 


---

Updated: 2024-12-05

此选项用于控制文档是否生成索引和参考文献, 所有可用的选项为:

```
\zlatexSetup{
  bib_index={
    load=\bool{false},
    source=\str{ref.bib},
    backend=\str{biber}
  }
}
```

$\langle load \rangle$  用于控制是否加载 biblatex 宏包, 默认为 `false`;  $\langle source \rangle$  用于指定参考文献源文件, 默认文件名为: `ref.bib`;  $\langle backend \rangle$  用于指定参考文献的后端, 默认为 `biber`.

---

**mathSpec**

---

Updated: 2024-11-05

---

**mathSpec**=⟨key-value⟩

此键用于配置数学排版相关选项. 所有可用的选项为:

```

\zlatexSetup{
  mathSpec={
    alias=⟨bool: false⟩,
    envStyle=⟨tl: plain⟩,
    font=⟨choice: ncmrm⟩
  }
}

```

⟨*alias*⟩ 默认为 `false`, 当置为 `true` 时, zL<sup>A</sup>T<sub>E</sub>X 会加载 `mathalias` 库, 其中包含一系列命令的简写声明, 如 `\ZZ` 代替 `\mathbb{Z}`; ⟨*envStyle*⟩ 用于指定数学环境的样式, 默认为 `plain`, 为了编译速度考虑, zL<sup>A</sup>T<sub>E</sub>X 在预定了一系列环境的同时, 也在 `theme` 库中声明了其它样式. 样式列表如下:

**thm module 定义样式:**

- `plain`
- `background`
- `leftbar`
- `fancy`

**theme library 定义样式:**

- `shadow`
- `paris`
- `elegant`
- `obsidian`
- `lapsis`

⟨*font*⟩ 用于指定数学公式字体, 预定义的字体有: `newtx`, `euler`, `mtpro2`, `mathpazo`. 其中 `mtpro2` 为付费字体, 需用户手动安装.

## 5 杂项

本小节会列举部分在 `zlatex` 源文件中定义的命令, 这部分命令未迁移到任何的 `module` 或者是 `library` 中.

`\zlatexVerb`

`\zlatexVerb[<format>]{<item>}`

Updated: 2024-11-05

此命令和 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 中的 `\verb` 类似, 用于输出控制序列名称. 和后者类似, 此命令也不能作为任何控制序列的参数. *<format>* 用于指定控制序列的打印格式, 默认为 `\texttt`. 一个使用样例如下:

```
\zlatexVerb{\alpha + \beta}\par
\zlatexVerb[\textsf]{\alpha + \beta}

-----

\alpha + \beta
\alpha + \beta
```

`\zlatexCounterWith`

`\zlatexCounterWith{<child>}{<parent>}`

Updated: 2024-11-05

此命令用于给一个计数器添加一个父计数器, 当 *<parent>* 计数器增加时, *<child>* 计数器会自动重置. 本命令为 `\@addtoreset` 的封装.

`\zlatexFramed`

`\zlatexFramed[<color>]{<name>}`

Updated: 2024-11-05

此命令用于创建一个类似 Markdown 中引用环境, *<color>* 表示环境 *<name>* 的默认颜色, 在使用环境 *<name>* 时可以更改 *<color>* 这一默认的可选参数. 一个使用样例如下:

```
\zlatexFramed[red]{ref}
\begin{ref}This is a simple ref env.\end{ref}
\begin{ref}[green]This is a simple ref env.\end{ref}

-----

This is a simple ref env.

This is a simple ref env.
```

`\_zlatex_quad_dim`

`\_zlatex_quad_dim`

Updated: 2024-12-05

此命令表示当前文档中一个空格的宽度.

## 6 Modules

本节对应的所有 module 默认自动加载, 用户仍可以通过命令 `\zlatexloadmodule` 调用自己编写的 module.

### 6.1 fontcfg

本模块主要用于配置 z $\text{\LaTeX}$  的字体, 尽管 `fontspec` 和 `unicode-math` 已经在很大程度上简化了字体的配置, 但是对于一些用户来说, 仍然会感到困扰. 本模块的目的就是为了简化字体的配置, 让普通的  $\text{\LaTeX}$  用户也能够方便的配置字体, 用上自己喜欢的字体.

---

`\Cinzel`


---

`Updated: 2024-12-05`


---

`\Cinzel`

本命令用于临时切换 Cinzel 字体 (此时需使用  $\text{\XeLaTeX}$  或  $\text{\LuaLaTeX}$ ), 本字体在 `\fancy=true` 时, 会自动应用于 chapter 页的字体.

---

`\blacktriangleright`


---

`Updated: 2024-12-05`


---

`\blacktriangleright`

本命令 (符号) 来自 AMSb 字体, `\slot="49`. 主要用于在 `\slide=true` 时对此符号进行 Patch.

## 6.2 indexref

本模块主要用于给文档增加索引, 参考文献以及超链接支持, 通过本模块, 用户可以方便的添加索引, 参考文献以及超链接.

<code>\hyper@anchor</code>	<code>\hyper@anchor{&lt;destination name&gt;}</code>
New: 2024-12-05	此命令用于创建一个超链接锚点, <code>&lt;destination name&gt;</code> 作为后续超链接命令的跳转目标.
<code>\hyper@link</code>	<code>\hyper@link{&lt;context&gt;}{&lt;destination name&gt;}{&lt;link text&gt;}</code>
New: 2024-12-05	此命令用于创建一个超链接, <code>&lt;link text&gt;</code> 本身作为一个超链接对象, 点击 <code>&lt;link text&gt;</code> 即可跳转到对应的 <code>&lt;destination name&gt;</code> . <code>&lt;context&gt;</code> 表示此链接所属的类型, 默认有: <code>link</code> , <code>url</code> , <code>cite</code> 三种类型.
<code>\hyper@linkstart</code>	<code>\hyper@linkstart{&lt;context&gt;}{&lt;destination name&gt;}</code>
New: 2024-12-05	此命令用于开启一个超链接域, 在此域中可以是任意的文本或其它图片对象. 此命令需结合后续的 <code>\hyper@linkend</code> 命令使用, 此二命令可以和上述的 <code>\hyper@link</code> 命令基本等效.
<code>\hyper@linkend</code>	<code>\hyper@linkend</code>
New: 2024-12-05	用于结束由 <code>\hyper@linkstart</code> 开启的域.
<code>\MakeLinkTarget</code>	<code>\MakeLinkTarget[&lt;prefix&gt;]{&lt;counter&gt;}</code>
<code>\MakeLinkTarget*</code>	<code>\MakeLinkTarget*{&lt;target&gt;}</code>
New: 2024-12-05	此二命令用于在用户层面创建超链接跳转目标, 其中 <code>&lt;prefix&gt;</code> 和 <code>&lt;counter&gt;</code> 可以作为命令 <code>\hyper@link</code> 的参数使用. <code>&lt;counter&gt;</code> 可以为 <code>chapter</code> , <code>section</code> , <code>subsection</code> 等. 针对 <code>\MakeLinkTarget*</code> , 其中 <code>&lt;target&gt;</code> 可以为任意的 Unicode 文本 (但为了兼容性考虑, 请尽量使用 ASCII 字符).
<code>\LinkTargetOn</code>	<code>\LinkTargetOn</code>
<code>\LinkTargetOff</code>	<code>\LinkTargetOff</code>
New: 2024-12-05	此命令常在一个局部中用于取消由 <code>\MakeLinkTarget</code> 或 <code>\refstepcounter</code> 创建的超链接. 在使用 <code>\LinkTargetOff</code> 后, 你仍然可以在一个局部里重新启用超链接然后创建对应的 Target, 示例如下:

```

\LinkTargetOff % suppress anchor in internal refstepcounter
...
\refstepcounter{...}
...
{\LinkTargetOn\MakeLinkTarget*{mytarget}} % create manual anchor for
future reference
...
\LinkTargetOn

```

---

<code>\NextLinkTarget</code>	<code>\NextLinkTarget{&lt;target&gt;}</code>
------------------------------	--

---

New: 2024-12-05

此命令设置下一个由 `\MakeLinkTarget` 或 `\refstepcounter` 创建的 target. 此命令的作用和 `\hypersetup` 中的 `next-anchor` 类似.

---

<code>\SetLinkTargetFilter</code>	<code>\SetLinkTargetFilter{&lt;filter&gt;}</code>
-----------------------------------	---

---

New: 2024-12-05

此命令用于给当前文档中所有的 Link Target 添加一个前缀, 此命令在合并多个不同的 PDF 时是十分有用的.

### 6.3 layout

本模块主要用于设置文章布局，包括纸张大小，页边距等.

## 6.4 pageinfo

本模块主要包含与页面生成以及页面标注相关(页眉页脚)的命令, 如 `\maketitle`, `\zlatexPageMask`; 通过本模块, 用户可以方便的制作自己的独特页面样式以及添加水印.

---

<code>\orimaketitle</code>	<code>\orimaketitle</code>
----------------------------	----------------------------

---

Updated: 2024-11-05	<code>\maketitle</code> 的原始形式.
---------------------	--------------------------------

---



---

<code>\maketitle</code>	<code>\maketitle</code>
-------------------------	-------------------------

---

Updated: 2024-11-05	<code>zL<sup>A</sup>T<sub>E</sub>X</code> 对原始的 <code>\maketitle</code> 进行了重定义, 以适应不同的文档类和页面布局.
---------------------	--

---



---

<code>\Maketitle</code>	<code>\Maketitle[⟨margin:1in⟩]</code>
-------------------------	---------------------------------------

---

New: 2024-12-05	此命令会忽略所有的文档类选项或者是页面布局, 在新的页面布局中插入 <code>\maketitle</code> 的原始定义, <code>⟨margin⟩</code> 表示新的页面布局的 <code>margin</code> , 默认为 <code>1in</code> . 此命令的实现为:
-----------------	--

---

```
\newcommand\Maketitle[1][1in]{
  \newgeometry{margin=#1}
  \orimaketitle
  \restoregeometry
}
```

---

<code>\frontmatter</code>	<code>\frontmatter</code>
---------------------------	---------------------------

---

<code>\mainmatter</code>	<code>\mainmatter</code>
--------------------------	--------------------------

---

Updated: 2024-11-05	此二命令用于设置文档的前言和正文部分, 在 <code>zL<sup>A</sup>T<sub>E</sub>X</code> 中这两个命令已经被重定义, 当加载的 <code>⟨class⟩</code> 为 <code>book</code> 或 <code>ctexbook</code> 时, 这两个命令会自动处理页眉页脚相关设置.
---------------------	--

---



---

<code>\_zlatex_page_annotate:nnnnn</code>	<code>\_zlatex_page_annotate:nnnnn {⟨fore/background⟩}</code>
Updated: 2024-12-05	<code>{⟨position⟩}{⟨anchor⟩}</code>
	<code>{⟨object⟩}{⟨hook range⟩}</code>

---

此命令为 `\zlatexPageMask` 的底层命令, 用户可以依据此命令创建更加具有针对性的水印命令.

---

<code>\zlatexPageMask</code>	<code>\zlatexPageMask[⟨key-value⟩]{⟨item⟩}</code>
------------------------------	---

---

<code>\zlatexPageMask*</code>	
-------------------------------	--

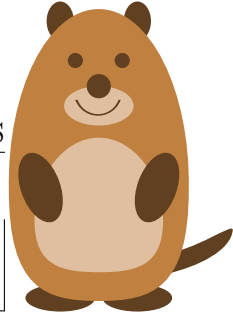
---

Updated: 2024-12-05	命令 <code>\zlatexPageMask</code> 用于给当前页面添加水印, <code>\zlatexPageMask*</code> 用于给当前页面及其之后的所有页面添加水印. <code>⟨item⟩</code> 可以为一段文字, 也可以为一系列的图片 (需要使用 <code>\includegraphics</code> 进行导入). <code>⟨key-value⟩</code> 所有可用的选项为:
---------------------	--

---

```
\zlatexPageMask[
  layer=⟨t1:background, foreground⟩,
  position=⟨t1:⟨dim1, dim2⟩⟩,
  label=⟨t1:DEFAULT⟩,
```





```
anchor=<tl: c, tl, tc, tr, bl, bc, br>
]{<item>}
```

其中  $\langle position \rangle$  以页面的左下角为原点, 向上向右为正方向. 注意: `transparent` 宏包仅可以在 `pdfLATEX` 和 `LuaLATEX` 下正常工作. 一个简单的示例, 给当前页面添加一个水印:

```
% \usepackage{tikzlings}
\zlatexPageMask{
  \transparent{.5}\includegraphics{latex-logo.pdf};
}
\zlatexPageMask[anchor=tr, position={{\zpw, \zph}}]{
  \begin{tikzpicture}[scale=2]
    \marmot
  \end{tikzpicture}
}
```

上述的 `\zpw`, `\zph` 分别表示当前纸张的宽和高.

---

`\zlatexPageMaskRemove`


---

New: 2024-12-05

---

```
\zlatexPageMaskRemove{<foreground/background>}{<label>}
```

此命令用于移除由 `\zlatexPageMask` 命令添加的页面水印,  $\langle label \rangle$  即为 `\zlatexPageMask` 可选参数  $\langle key-value \rangle$  中  $\langle label \rangle$  对应的 `value`. 如果  $\langle label \rangle$  对应的水印并不存在, `zLATEX` 会抛出警告.

The L<sup>A</sup>T<sub>E</sub>X Project

6.5 theme

本模块主要用于文档装饰，在本模块中定义了一系列的颜色主题，这系列主题可以应用于文章中的各个元素，包括但不限于章节标题，定理环境，超链接跳转，(子)目录样式。

\zlatexColorSetup

Updated: 2024-11-05

`\zlatexColorSetup{<key-value>}`

当 `<hype>=true` 时，此命令可以用于设置文档中各种元素的色彩，但仅可在导言区使用。zL<sup>A</sup>T<sub>E</sub>X 把文档中的元素分为如下的 3 类：

- 章节标题类: `chapter`, `chapter-rule`
- 超链接类: `link`, `cite`, `url`
- 定理环境类: `axiom`, `definition`, `theorem`, `proposition`, `lemma`, `corollary`, `proof`, `example`, `solution`

在颜色指定上，zL<sup>A</sup>T<sub>E</sub>X 实现了一套自己的颜色指定方式，在指定色彩时可以不必要提前定义。参见如下使用案例：

```
\zlatexColorSetup{
  chapter = red,
  link = {HTML}{d9d9d9},
  theorem = {RGB}{136, 63, 214}
}
```

zL<sup>A</sup>T<sub>E</sub>X 的默认配色<sup>2</sup>如下：















Struct	chapter	chap- rule	link	url	cite	chap- theme	slide- theme
Color							
MathEnv	axiom	definition	theorem	lemma	corollary	proposition	remark
Color							

表 3: zL<sup>A</sup>T<sub>E</sub>X 文档类默认配色

定理类环境的色彩保存于变量 `zlatex@color@<name>` 中，其中 `<name>` 为对应环境的名称。不推荐用户 使用命令 `\definecolor`, `\colorlet` 直接对这类色彩变量进行重定义，zL<sup>A</sup>T<sub>E</sub>X 鼓励用户通过 `\zlatexColorSetup` 命令进行色彩的重定义。

<sup>2</sup>zchapColor 还未整理，目前只能单独重定义

---

```
\\_zlatex_color_set:n \\_zlatex_color_set:n {<color spec>}
```

---

Updated: 2024-12-05

此命令可以自动解析 `<color spec>`, 并以此创建或定义对应的色彩. `<color spec>` 可以为普通的预定义色彩名, 如 `red`, `orange` 等. 亦或者是 HTML, RGB, CMYK 等色彩模型, 但此时的格式略有不同. 此命令仅能在 `\\keys_define:nn` 中使用, 新定义的色彩名为 `zlatex@color@\\l_keys_key_str`. 关于参数 `<color spec>` 的使用, 可以参见如下示例:

```
\\_zlatex_color_set:n {red}  
\\_zlatex_color_set:n {{HTML}{393D44}}
```

## 6.6 thm

本模块主要用于定理类以及证明类数学环境定制. 本模块提供了丰富的接口以及选项, 与此同时本模块提供了丰富的 Hook, 方便用户直接对环境进行操作.

---

 $\backslash\text{zlatexThmLang}$ 


---

Updated: 2024-11-05

此命令用于设置定理类环境的名称, 目前支持 **cn**, **en**, **fr** 三种语言. 此命令仅能在文档的导言区使用, 但为了说明此命令的使用方法, 在本手册中, 此命令的定义被临时改变了. 一个使用样例如下:

```
\begin{theorem}[AAA-1]
  This is a theorem AAA-1.
\end{theorem}
\zlatexThmLang{fr}
\begin{theorem}[BBB-1]
  This is a france theorem BBB-1.
\end{theorem}
\zlatexThmLang{en}
\begin{theorem}[CCC-1]
  This is a english theorem CCC-1.
\end{theorem}
```

**定理 6.1 (AAA-1)** This is a theorem AAA-1.

**Théorème 6.2 (BBB-1)** This is a france theorem BBB-1.

**Theorem 6.3 (CCC-1)** This is a english theorem CCC-1.

---

 $\backslash\text{zlatexThmCreate}$ 


---

Updated: 2024-12-25

 $\backslash\text{zlatexThmCreate}\{\langle type \rangle\}\{\langle spec \rangle\}$ 

根据  $\langle spec \rangle$  创建一系列类型为  $\langle type \rangle$  的定理环境, 如果对应的环境已存在, 则覆盖其原有的定义.  $\langle type \rangle$  可选 **theorem**, **proof** 两种类型, 针对  $\langle spec \rangle$ , 请参见下面的使用样例:

```
\zlatexThmCreate{theorem}{Zaxiom, Ztheorem=Thm|{\HTML}{a0d911},
Zproposition=Prop|blue}
\zlatexThmCreate{proof}{Zproof, Zexample=EXAMPLE|red, Zsolution=Solution|}
\begin{Zproof}[AAA-2]
  This is an proof AAA-2.
\end{Zproof}
\begin{Zexample}[BBB-2]
  This is an example.
\end{Zexample}
\begin{Ztheorem}[CCC-2]
  This is a theorem
\end{Ztheorem}
```

**Zproof:** This is an proof AAA-2.

**EXAMPLE:** This is an example.

**Thm 6.1 (CCC-2)** This is a theorem

`\zlatexThmCreate` 仅可在导言区使用. 但为了说明此命令的使用方法, 在本手册中, 此命令的定义被临时改变了. 和前面的 `\zlatexColorSetup` 类似, 在指定色彩时可以不必要提前定义.

---

`\zlatexThmNumber` ★ `\zlatexThmNumber`

---

Updated: 2024-11-05

此命令表示对应环境的编号, 类似于 `amsthm` 中的 `\thmnumber`. 用户不应在除 `\zlatexThmTitleFormat` 外的任何地方使用, 在此命令之外, 此命令输出的内容无任何实际意义.

---

`\zlatexThmName` ★ `\zlatexThmName`

---

Updated: 2024-11-05

此命令表示对应环境的名称, 类似于 `amsthm` 中的 `\thmname`. 用户不应在除 `\zlatexThmTitleFormat` 外的任何地方使用, 在此命令之外, 此命令输出的内容无任何实际意义.

---

`\zlatexThmNote` ★ `\zlatexThmNote{\langle prefix \rangle}{\langle suffix \rangle}`

---

Updated: 2024-12-05

此命令表示对应环境的注释, 类似于 `amsthm` 中的 `\thmnote`. 用户不应在除 `\zlatexThmTitleFormat` 外的任何地方使用, 在此命令之外, 此命令输出的内容无任何实际意义.

---

`\zlatexThmTitle` ★ `\zlatexThmTitle`

---

Updated: 2024-11-05

此命令表示对应环境的标题, 包含 `\zlatexThmNumber`, `\zlatexThmName`, `\zlatexThmNote` 三部分. 具体的格式可以使用 `\zlatexThmTitleFormat` 进行指定.

---

`\zlatexThmTitleSwitch` `\zlatexThmTitleSwitch`

---

Updated: 2024-11-05

此命令用于切换 `\zlatexThmTitle` 是否显示, 在自定义环境样式时比较有用. 用户不应该对此命令进行直接的调用.

```
\begin{theorem}[AAA-3]
  This is a theorem AAA-3.
\end{theorem}
\zlatexThmStyleNew{
  ZZZ={begin=, end=, option=\zlatexThmTitleSwitch},
}
\zlatexThmStyle{ZZZ}
\begin{theorem}[BBB-3]
  This is a theorem BBB-3.
\end{theorem}
```

定理 6.4 (AAA-3) This is a theorem AAA-3.  
This is a theorem BBB-3.

关于命令 `\zlatexThmStyle` 的使用可以参见下面的说明.

`\zlatexThmTitleFormat`  
`\zlatexThmTitleFormat*`

Updated: 2024-11-05

`\zlatexThmTitleFormat{<format>}`

此命令用于指定 `\zlatexThmTitle` 的格式, 默认格式为:

```
\zlatexThmName\ \zlatexThmNumber\ \zlatexThmNote{(){} }
```

`\zlatexThmNoteEmptyTF`

Updated: 2024-12-05

`\zlatexThmNoteEmptyTF{<true code>}{<false code>}`

此命令用于判断 `\zlatexThmNote` 是否为空, 如果为空则执行 `<true code>`, 否则执行 `<false code>`. 这个命令在自定义 `\zlatexThmTitle` 时很有用. 一个使用样例 (L<sup>A</sup>T<sub>E</sub>X 内置的 obsidian 定理样式对应的 format):

```
\zlatexThmTitleFormat*{\bfseries
\zlatexThmName\ \zlatexThmNumber
\zlatexThmNoteEmptyTF{}{\} }
\zlatexThmNote{}{}
}
```

`\zlatexThmBefore`

Updated: 2024-12-05

`\zlatexThmBefore{<code>:\par}`

此命令用于把 `<code>` 置于每个环境 (命令 `\_zlatex_thm_warp_start:nnnn`) 之前, 默认为 `\par`. 用户可以把 `<code>` 置为空或者是 `\noindent` 以取消段落缩进.

`\qedsymbol`

Updated: 2024-11-05

`\qedsymbol`

此命令用于输出证明环境的结束符号, 默认为  $\square$ .

`\zlatexThmCnt`

Updated: 2024-11-05

`\zlatexThmCnt{<key-value>}`

此命令用于定义数学类环境的计数器, 仅能在导言区使用, 但为了说明此命令的使用方法, 在本手册中, 此命令的定义被临时改变了. 所有可用的选项为:

```
\zlatexThmCnt{
parent=<counter:section>,
share=<bool:false>
}
```

`<parent>` 用于指定计数器的父计数器, 默认父计数器为 `section`; 当父计数器更新时, 此环境的计数器便会重置; `<share>` 用于控制所有的定理类环境是否共用一个计数器, 默认不共用.

---

`\zlatexThmColorSetup`

---

Updated: 2024-11-05

---

`\zlatexThmColorSetup{⟨key-value⟩}`

此命令和 `\zlatexColorSetup` 类似, 但其仅用于对数学环境的色彩设置, 且仅能在导言区使用. 所有合法的 `⟨key⟩` 选项请参见 `zLATEX` 文档类选项 `mathSpec` 中的数学环境.

---

`\zlatexThmStyle`

---

Updated: 2024-11-05

---

`\zlatexThmStyle{⟨style⟩}`

此命令用于设置定理类环境的样式, 仅能在导言区使用. 可用的 `⟨style⟩` 请参见 `⟨mathSpec⟩` 的说明. 一个基本的使用样例:

```
\begin{theorem}[AAA-4]
  This is a theorem AAA-4.
\end{theorem}
\zlatexThmStyle{fancy}
\begin{theorem}[BBB-4]
  This is a fancy theorem BBB-4.
\end{theorem}
```

定理 6.6 (AAA-4) This is a theorem AAA-4.

定理 6.7 (BBB-4) This is a fancy theorem BBB-4.

---

`\zlatexThmStyleNew`

---

Updated: 2024-12-05

---

`\zlatexThmStyleNew{⟨key-value⟩}`

此命令用于定义新的定理类环境的样式, 仅能在导言区使用. 调用格式为:

```
\zlatexThmStyleNew{
  ⟨style A⟩={
    begin=⟨begin code 1⟩,
    end=⟨end code 1⟩,
    option=⟨option 1⟩,
    preamble=⟨preamble code 1⟩
  },
  ⟨style B⟩={
    begin=⟨begin code 2⟩,
    end=⟨end code 2⟩,
    option=⟨option 2⟩,
    preamble=⟨preamble code 2⟩
  },
  ...
}
```

在声明对应的 `⟨style⟩` 后, 使用命令 `\zlatexThmStyle{⟨style⟩}` 进行切换.

---

`\zlatexThmHook`    `\zlatexThmHook{⟨key-value⟩}`  
`\zlatexThmHook*`   `\zlatexThmHook*{⟨key-value⟩}`

---

Updated: 2024-11-05

此命令用于给已有的定理类环境 Hook 中添加代码, 已有的 Hook: `⟨zlatex/thm/before⟩`, `⟨zlatex/thm/begin⟩`, `⟨zlatex/thm/end⟩`, `⟨zlatex/thm/after⟩`. `\zlatexThmHook` 只应用于下一个环境, `\zlatexThmHook*` 会应用于接下来的所有环境. 各个 Hook 的位置分布如下:

```
(zlatex/thm/before) --> (warper begin)
--> (thm-title)      --> (zlatex/thm/begin)
--> (thm-content)    --> (zlatex/thm/end) -->
(warper end) --> (zlatex/thm/after)
```

这两个命令不支持手动设置 `⟨label⟩`, 针对于 `\zlatexThmHook*`, `zLATEX` 会自动设置 `⟨label⟩`, 其格式为 `thm-hook.⟨Hook Index⟩`. 针对 `⟨key⟩`, 所有合法的选项为: `⟨before⟩`, `⟨begin⟩`, `⟨end⟩`, `⟨after⟩`, 一个使用样例如下:

```
\begin{theorem}[AAA-5]
  This is a theorem AAA-5.
\end{theorem}
\zlatexThmHook{before=ZZa\ , begin=ZZb\ ,}
\begin{theorem}[BBB-5]
  This is a theorem BBB-5.
\end{theorem}
```

---

**定理 6.8 (AAA-5)** This is a theorem AAA-5.  
**ZZa 定理 6.9 (BBB-5)** ZZb This is a theorem BBB-5.

---

`\zlatexThmToc`    `\zlatexThmToc[⟨stretch⟩]`

---

Updated: 2024-12-05

此命令用于打印定理类环境的目录, `⟨stretch⟩` 为任意非负的浮点数, 用于指定定理目录的 `stretch` 值. 使用样例如下:

```
\zlatexThmToc[1.25]
\begin{proposition}[AAA-6]proposition AAA-6\end{proposition}
\begin{lemma}[BBB-6]lemma BBB-6\end{lemma}
\begin{corollary}[CCC-3]corollary CCC-3\end{corollary}
```

---

<b>T 定理 6.1 (AAA-1)</b> . . . . .	20
<b>T Théorème 6.2 (BBB-1)</b> . . . . .	20
<b>T Theorem 6.3 (CCC-1)</b> . . . . .	20
<b>Thm 6.1 (CCC-2)</b> . . . . .	20
<b>T 定理 6.4 (AAA-3)</b> . . . . .	21
<b>T 定理 6.5 (BBB-3)</b> . . . . .	21
<b>T 定理 6.6 (AAA-4)</b> . . . . .	23
<b>T 定理 6.7 (BBB-4)</b> . . . . .	23
<b>T 定理 6.8 (AAA-5)</b> . . . . .	24



<b>T</b> 定理 6.9 (BBB-5) . . . . .	24
<b>P</b> 命题 6.1 (AAA-6) . . . . .	24
<b>L</b> 引理 6.1 (BBB-6) . . . . .	24
<b>C</b> 推论 6.1 (CCC-3) . . . . .	24
<b>R</b> 注记 7.1 (AAA-7) . . . . .	39
<b>A</b> 公理 7.1 (BBB-7) . . . . .	39
<b>P</b> 命题 7.1 (CCC-4) . . . . .	40
<b>L</b> 引理 7.1 (DDD-1) . . . . .	41
<b>D</b> 定义 7.1 (DDD-1) . . . . .	41
命题 6.1 (AAA-6) proposition AAA-6	
引理 6.1 (BBB-6) lemma BBB-6	
推论 6.1 (CCC-3) corollary CCC-3	

---

`\zlatexThmTocAdd` `\zlatexThmTocAdd[⟨key-value⟩]{⟨counter⟩}`

---

Updated: 2024-12-05

此命令用于向定理类环境目录中添加条目,  $\langle counter \rangle$  为计数器名, 可以为 `section`, `subsection` 等.  $\langle key-value \rangle$  为可选参数, 用于指定所添加的条目的其它信息. 一个使用样例如下:

```
\zlatexThmTocAdd[name=SEC-NAME]{section}
```

---

`\zlatexThmTocLevel` `\zlatexThmTocLevel{⟨depth⟩}`

---

Updated: 2024-12-05

此命令用于设置定理类环境目录的最大深度, 仅能在导言区使用, 但为了说明此命令的使用方法, 本手册中此命令的定义被临时改变了.  $\langle depth \rangle$  为一个  $\geq 1$  的整数.

---

`\zlatexThmTocPrefix` `\zlatexThmTocPrefix{⟨prefix⟩}`

---

Updated: 2024-12-05

此命令用于所有定理类环境目录中所有条目的共同前缀, 默认为空.

---

`\zlatexThmTocSymbol` `\zlatexThmTocSymbol{⟨key-value⟩}`

---

Updated: 2024-12-05

此命令用于分别设置所有定理类环境名在目录中的前缀, 仅能在导言区使用, 但为了说明此命令的使用方法, 本手册中此命令的定义被临时改变了. 所有可用的选项为:

```
\zlatexThmTocSymbol{
  ⟨axiom: A ⟩,
  ⟨definition: D ⟩,
  ⟨theorem: T ⟩,
  ⟨lemma: L ⟩,
  ⟨corollary: C ⟩,
  ⟨proposition: P ⟩,
  ⟨remark: R ⟩,
}
```

---

`\zlatexThmTocSymbolClear`

---

Updated: 2024-12-05

`\zlatexThmTocSymbolClear`

此命令用于清除所有由命令 `\zlatexThmTocSymbol` 指定的定理类环境名在目录中的前缀, 自然不包括由 `\zlatexThmTocPrefix` 指定的前缀.

## 6.7 titlesec

本模块的用于定义章节标题样式, 目的是实现 `titlesec` 和 `titletoc` 中的所有功能, 使其能作为上述两宏包的一个可选替代. 后续可能会与 `toc module` 合并.

## 6.8 toc

本模块主要用于自定义目录格式, 目前基于 `titletoc`, 后续可能会与 `titlesec` module 合并.

---

`\zlatexPartialToc`

---

Updated: 2024-11-05

---

`\zlatexPartialToc[depth:2]`

此命令用于输出每一个章节对应的子目录, 此命令目前基于 `titletoc` 宏包. `<depth>` 用于指定子目录最大深度, 默认为 2.

---

`\zlatexStopPartialToc`

---

Updated: 2024-11-05

---

`\zlatexStopPartialToc{chapters,section}{counter}`

此命令用于结束子目录的搜集, 一般情况下, 用户不应该使用此命令.

## 7 Libraries

本节主要介绍 z $\text{\LaTeX}$  中提供的各类 library，这些 library 用于优化用户  $\text{\LaTeX}$  的文档书写和阅读体验。部分 library 是对 z $\text{\LaTeX}$  中原始功能的增强，但与此同时，文档的编译速度势必会稍微减慢，所以请酌情加载这部分 library。

所有的 library 均不默认加载，用户需要使用 `\zlatexloadlibrary{<library name>}` 手动加载，详细的 `<library name>` 请参见命令 `\zlatexloadlibrary` 的说明。

### 7.1 fancy

此 library 用于章节的格式化以及部分的，目前仅对 `\chapter` 进行了重定义。如果此时 `<fancy>=true`，那么在加载此 library 的同时，z $\text{\LaTeX}$  会同时加载 `tcolorbox`，`tikz` 以及 `tikz` 的 `calc` 库。

---

`\numSuffix`


---

`\numSuffix{<number>}`


---

Updated: 2024-11-05

---

此命令用于数字的格式化，其中 `<number>` 为一个任意的整数，一个使用样例如下：

```
\numSuffix{1}, \numSuffix{2}, \numSuffix{25}
```

---

st, nd, th

## 7.2 mathalias

本模块主要为一系列命令的别名定义, 后文称此为 alias, 用于简化用户在数学环境中的命令输入. 此 library 建立了以下几个方面的 alias:

- 数学字体命令
- 各类箭头
- 各类数学算符
- 其余常见符号
- 自动括号命令 (试验阶段)

对于自动括号命令, 目前还很不成熟, 如果不清楚对应的命令原理请勿使用. 正对此特性, 推荐用于使用 `physics 2` 宏包.

---

<code>\F</code>	<code>\F{&lt;math char&gt;}</code>
<code>\R</code>	<code>\R{&lt;math char&gt;}</code>
<code>\K</code>	<code>\K{&lt;math char&gt;}</code>
<code>\C</code>	<code>\C{&lt;math char&gt;}</code>
<code>\B</code>	<code>\B{&lt;math char&gt;}</code>
<code>\S</code>	<code>\S{&lt;math char&gt;}</code>
<code>\FF</code>	<code>\FF{&lt;math char&gt;}</code>

---

Updated: 2024-12-05

以上各命令的原始定义: `\F` 为 `\boldsymbol`, `\R` 为 `\mathrm`, `\K` 为 `\mathfrak`, `\C` 为 `\mathcal`, `\B` 为 `\mathbb`, `\S` 为 `\mathscr`, `\FF` 为 `\mathbf`. 使用示例如下:

Normal Version:  $\mathbf{A} + \mathrm{A} + \mathfrak{a} + \mathcal{A} + \mathbb{A} + \mathscr{A} + \mathbf{A}$  ✓

Alias Version:  $\mathbf{F}{A} + \mathbf{R}{A} + \mathbf{K}{a} + \mathbf{C}{A} + \mathbf{B}{A} + \mathbf{S}{A} + \mathbf{FF}{A}$

---

Normal Version:  $\mathbf{A} + A + \mathfrak{a} + \mathcal{A} + \mathbb{A} + \mathscr{A} + \mathbf{A}$

Alias Version:  $\mathbf{A} + A + \mathfrak{a} + \mathcal{A} + \mathbb{A} + \mathscr{A} + \mathbf{A}$

---

<code>\ma</code>	<code>\ma</code>
<code>\mma</code>	<code>\mma</code>

---

Updated: 2024-12-05

以上各命令的原始定义: `\ma` 为 `\mapsto`, `\mma` 为 `\longmapsto`. 使用示例如下:

Normal Version:  $a \mapsto b, a \longmapsto b$

Alias Version:  $a \ma b, a \mma b$

---

Normal Version:  $a \mapsto b, a \longmapsto b$

Alias Version:  $a \ma b, a \mma b$

---

<code>\la</code>	<code>\la</code>
<code>\La</code>	<code>\La</code>
<code>\nla</code>	<code>\nla</code>
<code>\Nla</code>	<code>\Nla</code>
<code>\lla</code>	<code>\lla</code>
<code>\Lla</code>	<code>\Lla</code>

---

Updated: 2024-12-05

以上各命令的原始定义: `\la` 为 `\leftarrow`, `\La` 为 `\Leftarrow`, `\nla` 为 `\nleftarrow`, `\Nla` 为 `\nLeftarrow`, `\lla` 为 `\longleftarrow`, `\Lla` 为 `\Longleftarrow`. 使用示例如下:

Normal Version:  $a \leftarrow b, a \Leftarrow b, a \nleftarrow b, a \nLeftarrow b, a \longleftarrow b, a \Longleftarrow b$  ✓  
 Alias Version:  $a \la b, a \La b, a \nla b, a \Nla b, a \lla b, a \Lla b$ .

-----  
 Normal Version:  $a \leftarrow b, a \Leftarrow b, a \nleftarrow b, a \nLeftarrow b, a \longleftarrow b, a \Longleftarrow b$   
 Alias Version:  $a \la b, a \La b, a \nla b, a \Nla b, a \lla b, a \Lla b$ .

---

<code>\ra</code>	<code>\ra</code>
<code>\Ra</code>	<code>\Ra</code>
<code>\nra</code>	<code>\nra</code>
<code>\Nra</code>	<code>\Nra</code>
<code>\rra</code>	<code>\rra</code>
<code>\Rra</code>	<code>\Rra</code>

---

Updated: 2024-12-05

以上各命令的原始定义: `\ra` 为 `\rightarrow`, `\Ra` 为 `\Rightarrow`, `\nra` 为 `\nrightarrow`, `\Nra` 为 `\nRightarrow`, `\rra` 为 `\longrightarrow`, `\Rra` 为 `\Longrightarrow`. 使用示例如下:

Normal Version:  $a \rightarrow b, a \Rightarrow b, a \nrightarrow b, a \nRightarrow b, a \longrightarrow b, a \Longrightarrow b$  ✓  
 Alias Version:  $a \ra b, a \Ra b, a \nra b, a \Nra b, a \rra b, a \Rra b$ .

-----  
 Normal Version:  $a \rightarrow b, a \Rightarrow b, a \nrightarrow b, a \nRightarrow b, a \longrightarrow b, a \Longrightarrow b$   
 Alias Version:  $a \ra b, a \Ra b, a \nra b, a \Nra b, a \rra b, a \Rra b$ .

---

<code>\da</code>	<code>\da</code>
<code>\Da</code>	<code>\Da</code>
<code>\nda</code>	<code>\nda</code>
<code>\Nda</code>	<code>\Nda</code>
<code>\dda</code>	<code>\dda</code>
<code>\Dda</code>	<code>\Dda</code>

---

Updated: 2024-12-05

以上各命令的原始定义: `\da` 为 `\leftrightarrow`, `\Da` 为 `\Leftrightarrow`, `\nda` 为 `\nleftrightarrow`, `\Nda` 为 `\nLeftrightarrow`, `\dda` 为 `\longleftrightarrow`, `\Dda` 为 `\Longleftrightarrow`. 使用示例如下:

Normal Version:  $\$a\leftrightharpoonup b, a\Lefttrightharpoonup b, a\leqtrightharpoonup b, a\leqtrightharpoonup b, a\longleftarrow b, a\Longleftarrow b, a\longrightarrow b, a\Longrightarrow b$  ✓  
 Alias Version:  $\$a\lhd b, a\Da b, a\nda b, a\Nda b, a\dd a b, a\Dda b$ .

Normal Version:  $a \leftrightarrow b, a \Leftrightarrow b, a \rightleftharpoons b, a \rightleftarrows b, a \longleftrightarrow b, a \Longleftrightarrow b$

Alias Version:  $a \leftrightarrow b, a \Leftrightarrow b, a \rightleftharpoons b, a \rightleftarrows b, a \longleftrightarrow b, a \Longleftrightarrow b$ .

---

<code>\xla</code>	<code>\xla[⟨above⟩](⟨below⟩)</code>
<code>\xla*</code>	<code>\xla*[⟨above⟩](⟨below⟩)</code>
<code>\Xla</code>	<code>\Xla[⟨above⟩](⟨below⟩)</code>
<code>\Xla*</code>	<code>\Xla*[⟨above⟩](⟨below⟩)</code>
<code>\xxla</code>	<code>\xxla[⟨above⟩](⟨below⟩)</code>
<code>\xxla*</code>	<code>\xxla*[⟨above⟩](⟨below⟩)</code>
<code>\xra</code>	<code>\xra[⟨above⟩](⟨below⟩)</code>
<code>\xra*</code>	<code>\xra*[⟨above⟩](⟨below⟩)</code>
<code>\Xra</code>	<code>\Xra[⟨above⟩](⟨below⟩)</code>
<code>\Xra*</code>	<code>\Xra*[⟨above⟩](⟨below⟩)</code>
<code>\xxra</code>	<code>\xxra[⟨above⟩](⟨below⟩)</code>
<code>\xxra*</code>	<code>\xxra*[⟨above⟩](⟨below⟩)</code>
<code>\hla</code>	<code>\hla[⟨above⟩](⟨below⟩)</code>
<code>\hla*</code>	<code>\hla*[⟨above⟩](⟨below⟩)</code>
<code>\hra</code>	<code>\hra[⟨above⟩](⟨below⟩)</code>
<code>\hra*</code>	<code>\hra*[⟨above⟩](⟨below⟩)</code>

---

Updated: 2024-12-05

以上所有带有 \* 命令中的 ⟨above⟩ 参数均会被放入 \text 命令中, 以上命令的原始定义: \xla 为 \xleftarrow, \Xla 为 \xLeftarrow, \xxla 为 \xLongleftarrow, \xra 为 \xrightarrow, \Xra 为 \xRightarrow, \xxra 为 \xLongrightarrow, \hla 为 \xhookleftarrow, \hra 为 \xhookrightarrow. 使用示例如下:

Normal Version:  $\$ \xleftarrow[b]{a} + \xLeftarrow[b]{a} + \xLongleftarrow[b]{a} + \xrightarrow[b]{a} + \xRightarrow[b]{a} + \xLongrightarrow[b]{a} + \xhookleftarrow[b]{a} + \xhookrightarrow[b]{a}$  ✓  
 Alias Version:  $\$ \xla[a](b) + \Xla[a](b) + \xxla[a](b) + \xra[a](b) + \Xra[a](b) + \xxra[a](b) + \hla[a](b) + \hra[a](b)$  ✓  
 Alias Text Version:  $\$ \xla*[a](b) + \Xla*[a](b) + \xxla*[a](b) + \xra*[a](b) + \Xra*[a](b) + \xxra*[a](b) + \hla*[a](b) + \hra*[a](b)$  ✓

Normal Version:  $\frac{a}{b} + \frac{a}{b} + \frac{a}{b} + \frac{a}{b} + \frac{a}{b} + \frac{a}{b} + \frac{a}{b} + \frac{a}{b}$

Alias Version:  $\frac{a}{b} + \frac{a}{b} + \frac{a}{b} + \frac{a}{b} + \frac{a}{b} + \frac{a}{b} + \frac{a}{b} + \frac{a}{b}$

Alias Text Version:  $\frac{a}{b} + \frac{a}{b} + \frac{a}{b} + \frac{a}{b} + \frac{a}{b} + \frac{a}{b} + \frac{a}{b} + \frac{a}{b}$

---

<code>\A</code>	<code>\A</code>
<code>\E</code>	<code>\E</code>

---

Updated: 2024-12-05

以上两个命令分别表示任意和存在符号, 一个使用样例如下:



Normal Version:  $\forall \varepsilon > 0, \exists \delta$  \\

Alias Version:  $\forall \varepsilon > 0, \exists \delta$

Normal Version:  $\forall \varepsilon > 0, \exists \delta$

Alias Version:  $\forall \varepsilon > 0, \exists \delta$

`\ns`

`\ns`

`\se`

`\se`

`\sse`

`\sse`

Updated: 2024-12-05

以上三个命令的原始定义: `\ns` 为 `\varnothing`, `\se` 为 `\backsim`, `\sse` 为 `\cong`. 一个使用样例如下:

Normal Version:  $\varnothing, \backsim, \cong$  \\

Alias Version:  $\varnothing, \backsim, \cong$

Normal Version:  $\varnothing, \backsim, \cong$

Alias Version:  $\varnothing, \backsim, \cong$

`\dd`

`\dd`

Updated: 2024-12-05

此命令主要用于替代默认的 `\mathrm{dd}`, 与此同时, 其会自动处理左右间隔, 更加规范的处理可以参见 `fixdiff`. 一个简单的使用示例如下:

Normal Version:  $\int x \mathrm{d}x = x^{\int x \mathrm{d}x} = \frac{1}{2}x^2 + C$  ✓

Alias Version:  $\int x \mathrm{d}x = x^{\int x \mathrm{d}x} = \frac{1}{2}x^2 + C$  ✓

Alias Version:  $\int x \mathrm{d}x = x^{\int x \mathrm{d}x} = \frac{1}{2}x^2 + C$  ✓

Normal Version:  $\int x \mathrm{d}x = x^{\int x \mathrm{d}x} = \frac{1}{2}x^2 + C$

Alias Version:  $\int x \mathrm{d}x = x^{\int x \mathrm{d}x} = \frac{1}{2}x^2 + C$

`\CC`

`\CC`

`\RR`

`\RR`

`\NN`

`\NN`

`\ZZ`

`\ZZ`

Updated: 2024-12-05

以上四个命令分别表示复数域, 实数域, 自然数集以及整数集. 一个使用样例如下:

Normal Version:  $\mathbb{C}, \mathbb{R}, \mathbb{N}, \mathbb{Z}$  \\

Alias Version:  $\mathbb{C}, \mathbb{R}, \mathbb{N}, \mathbb{Z}$

Normal Version:  $\mathbb{C}, \mathbb{R}, \mathbb{N}, \mathbb{Z}$

Alias Version:  $\mathbb{C}, \mathbb{R}, \mathbb{N}, \mathbb{Z}$

---

<code>\alt</code>	<code>\alt</code>
<code>\rot</code>	<code>\rot</code>
<code>\div</code>	<code>\div</code>
<code>\curl</code>	<code>\curl</code>
<code>\grad</code>	<code>\grad</code>
<code>\id</code>	<code>\id</code>
<code>\im</code>	<code>\im</code>
<code>\ker</code>	<code>\ker</code>
<code>\cok</code>	<code>\cok</code>
<code>\hom</code>	<code>\hom</code>
<code>\sign</code>	<code>\sign</code>
<code>\trace</code>	<code>\trace</code>

---

Updated: 2024-12-05

以上所有命令均使用 `\DeclareMathOperator` 进行声明, 其会自动处理前后间距. 一个使用样例如下:

Normal Version: <code>\operatorname{alt}</code> , <code>\operatorname{rot}</code> ,	✓
<code>\operatorname{div}</code> , <code>\operatorname{curl}</code> , <code>\operatorname{grad}</code> ,	✓
<code>\operatorname{Id}</code> , <code>\operatorname{Im}</code> , <code>\operatorname{Ker}</code> ,	✓
<code>\operatorname{Cok}</code> , <code>\operatorname{Hom}</code> , <code>\operatorname{sign}</code> ,	✓
<code>\operatorname{trace}</code>	
Alias Version: <code>\alt</code> , <code>\rot</code> , <code>\div</code> , <code>\curl</code> , <code>\grad</code> , <code>\id</code> , <code>\im</code> , <code>\ker</code> , <code>\cok</code> ,	✓
<code>\hom</code> , <code>\sign</code> , <code>\trace</code>	
-----	
Normal Version: alt, rot, div, curl, grad, Id, Im, Ker, Cok, Hom, sign, trace	
Alias Version: alt, rot, div, curl, grad, Id, Im, Ker, Cok, Hom, sign, trace	

---

<code>\zab</code>	<code>\zab&lt;(), [], \{\}</code>
-------------------	-----------------------------------

---

Updated: 2024-12-05

此命令用于处理括号的自动缩放, 一个使用样例如下:

Normal Version: <code>\displaystyle \left( \frac{1}{2} \right)</code> , <code>\left[ \frac{1}{2} \right]</code> , <code>\left\{ \frac{1}{2} \right\}</code>	✓
Alias Version: <code>\displaystyle \zab(\frac{1}{2})</code> , <code>\zab[\frac{1}{2}]</code> , <code>\zab\{\frac{1}{2}\}</code>	
-----	
Normal Version: $\left( \frac{1}{2} \right)$ , $\left[ \frac{1}{2} \right]$ , $\left\{ \frac{1}{2} \right\}$	
Alias Version: $\zab\left( \frac{1}{2} \right)$ , $\zab\left[ \frac{1}{2} \right]$ , $\zab\left\{ \frac{1}{2} \right\}$	

### 7.3 slide

此 library 用于从文档模式切换到 slide 模式, 无需用户对代码进行任何的更改, 仅需在导言区加载此 library 即可.  $\text{z}\text{L}\text{A}\text{T}\text{E}\text{X}$  会自动处理文档的分页.

由于此 library 内部 patch 了很多的  $\text{L}\text{A}\text{T}\text{E}\text{X}$  内部命令, 所以请谨慎加载. 另外, 加载此 library 并不会牺牲太多的编译速度.

---

 $\backslash\text{zslideFrameCntAux:nn}$ 


---

Updated: 2024-11-05

---

 $\backslash\text{zslideFrameCntAux:nn} \{ \langle name \rangle \} \{ \langle number \rangle \}$ 

此命令会向文件  $\backslash\text{jobname.aux}$  中写入一个变量, 其名称为:  $\backslash\text{zsec@}\langle name \rangle@\text{cnt}$ , 其值为:  $\langle number \rangle$ . 此命令在制作目录或向后搜集文档内容时是十分有用的.

---

 $\backslash\_zslide\_status\_bar:nnnn$ 


---

Updated: 2024-11-05

---

 $\backslash\_zslide\_status\_bar:nnnn \{ \langle type \rangle \} \{ \langle coordinate \rangle \} \{ \langle width \rangle \} \{ \langle height \rangle \}$ 

此命令用于创建 slide 的页面状态栏, 其中  $\langle coordinate \rangle$  为坐标, 形如 (10pt, .1\linewidth), 以当前页面的左上角为原点, 向下向右为正方向;  $\langle type \rangle$  为状态栏类型, 目前所有可选值有: UR, UL, BL, BC, BR, sec;  $\langle width \rangle$  为宽度, 接受一个浮点数, 默认以  $\backslash\text{paperwidth}$  为单位.  $\langle height \rangle$  为状态栏的高度, 接受一个合法的 dim 类型值, 如 10pt, 2em 等.

注意: 普通用户不应该直接调用此命令, 此命令主要提供给模板的开发者.

---

 $\backslash\_zslide\_status\_info:nnnn$ 


---

Updated: 2024-11-05

---

 $\backslash\_zslide\_status\_info:nnnn \{ \langle type \rangle \} \{ \langle coordinate \rangle \} \{ \langle width \rangle \} \{ \langle content \rangle \}$ 

此命令用于创建 slide 的页面元信息, 其被置于一个 box 中, 为方便叙述, 我们称其为  $\langle BOX \rangle$ . 其中  $\langle type \rangle$  表示  $\langle BOX \rangle$  在页面上的位置, 可选值有: foot, head;  $\langle coordinate \rangle$  表示  $\langle BOX \rangle$  的左侧坐标, 接受一个浮点数, 以  $\backslash\text{linewidth}$  为单位. 此参数以当前页面的左上角为原点, 向下向右为正方向.;  $\langle width \rangle$  为当前  $\langle BOX \rangle$  的 (弹性) 宽度, 接受一个浮点数, 以  $\backslash\text{linewidth}$  为单位.  $\langle content \rangle$  表示  $\langle BOX \rangle$  中存放的文本或图片内容.

注意: 普通用户不应该直接调用此命令, 此命令主要提供给模板的开发者.

---

 $\backslash\_zslide\_navigate:nnnn$ 


---

Updated: 2024-11-05

---

 $\backslash\_zslide\_navigate:nnnn \{ \langle range \rangle \} \{ \langle target \rangle \} \{ \langle target\ symbol \rangle \} \{ \langle other\ symbol \rangle \}$ 

此命令用于创建 slide 中的导航栏,  $\langle range \rangle$  接受一个正整数, 表示 frame 的总数;  $\langle target \rangle$  为接受一个在  $0 \sim \langle range \rangle$  内的正整数, 表示选定的编号.  $\langle target\ symbol \rangle$  为选定的编号的符号,  $\langle other\ symbol \rangle$  为其它编号的符号.

注意: 普通用户不应该直接调用此命令, 此命令主要提供给模板的开发者.

---

 $\backslash\text{zslideTitle}$ 
 $\backslash\text{zslideAuthor}$ 
 $\backslash\text{zslideDate}$ 


---

Updated: 2024-11-05

---

 $\backslash\text{zslideTitle}$ 
 $\backslash\text{zslideAuthor}$ 
 $\backslash\text{zslideDate}$ 

此三个命令用于分别保存导言区  $\backslash\text{@title}$ ,  $\backslash\text{@author}$ ,  $\backslash\text{@date}$  三个变量的值, 用户可以在正文部分使用此三个变量.

---

 $\backslash\_zslide\_metadate:n$ 


---

Updated: 2024-11-05

---

 $\backslash\_zslide\_metadate:n \{ \langle key \rangle \}$ 

此命令可以根据  $\langle key \rangle$  获取 slide 的 status info 中对应的元信息.

注意: 普通用户不应该直接调用此命令, 此命令主要提供给模板的开发者.

---

 $\backslash\text{zslideFrameTotal}$   $\backslash\text{zslideFrameTotal}\{\langle name \rangle\}$ 


---

Updated: 2024-11-05

用户可以在自定义导航栏时使用此命令, 此命令可以根据  $\langle name \rangle$  来获取  $\backslash\text{jobname.aux}$  中变量  $\backslash\text{zsec}\langle name \rangle\text{cnt}$  的值. 默认返回整个 slide 的 Frame 总数, 第一次编译或者变量  $\backslash\text{zsec}\langle name \rangle\text{cnt}$  不存在时则返回 ??.

---

 $\backslash\text{zslideFrameIndex}$   $\backslash\text{zslideFrameIndex}$ 


---

Updated: 2024-11-05

用户可以在自定义导航栏时使用此命令, 此命令在每一页 Frame 中会返回其对应的 Frame Index. 比如在第一页, 其返回的 Frame Index 为 1.

---

 $\backslash\text{zslideNavigateBall}$   $\backslash\text{zslideNavigateBall}[\langle target symbol \rangle][\langle other symbol \rangle]$ 


---

Updated: 2024-11-05

此命令为上述  $\backslash\text{zslide\_navigate:nnnn}$  的一个具体应用.  $\langle target symbol \rangle$  默认为  $\bullet$ ,  $\langle other symbol \rangle$  默认为  $\circ$ . 二者的意义请参见命令  $\backslash\text{zslide\_navigate:nnnn}$  的说明.

---

 $\backslash\text{zslidelogo}$   $\backslash\text{zslidelogo}[\langle key-value \rangle]\{\langle picture \rangle\}$ 


---

Updated: 2024-11-05

此命令用于设置 slide 的 logo 图标, 仅可在导言区使用.  $\langle picture \rangle$  为图片名称,  $\langle key-value \rangle$  的所有可用选项如下:

```
 $\backslash\text{zslidelogo}[
  position = \langle dim1, dim2 \rangle : (\backslash\text{paperwidth} - \backslash\text{zlatex\_quad\_dim}, 1.5em),
  width = \langle dim : 2.5em \rangle,
  exclude = \langle clist : 1 \rangle
]\{\langle picture \rangle\}$ 
```

$\langle exclude \rangle$  中的页码不会添加 logo, 其余页码会添加 logo.

---

 $\backslash\text{zslideTocPage}$   $\backslash\text{zslideTocPage}$   
 $\backslash\text{thecontentslabel}$   $\backslash\text{thecontentslabel}$   
 $\backslash\text{zslideToclabelSet}$   $\backslash\text{zslideTocSecIcon}$   
 $\backslash\text{zslideTocSecIcon}$   $\backslash\text{zslideTocSubsecIcon}$   
 $\backslash\text{zslideTocSubsecIcon}$   $\backslash\text{zslideToclabelSet}[\langle extra width \rangle]\{\langle item \rangle\}$ 


---

Updated: 2024-11-05

这一组命令主要用于自定义 slide 中的目录, 其中  $\backslash\text{zslideTocPage}$  表示目录项目对应的页码,  $\backslash\text{thecontentslabel}$  表示目录项目的对应的名称.  $\backslash\text{zslideTocSecIcon}$  和  $\backslash\text{zslideTocSubsecIcon}$  表示 slide 模式下目录中 section 和 subsection 对应的 icon. 用户可以在导言区自定义这两个 icon, 默认情况下这两个 icon 的声明及效果如下:

```
 $\backslash\text{ExplSyntaxOn}$ 
Section Icon:  $\backslash\text{box\_move\_up:nn}$  {2pt}
 $\{\backslash\text{hbox:n}$  { $\backslash\text{ztool\_resize\_to\_wd:nn}$ 
 $\{6pt\}$   $\{\backslash(\backslash\text{blacktriangleright})\}$ 
 $\}\backslash\text{par}$ 

Subsection Icon:  $\backslash\text{rule}[1.5pt]\{3pt\}\{3pt\}$ 
 $\backslash\text{ExplSyntaxOff}$ 
```

---

SectionIcon:▶  
SubsectionIcon:■

---

`\zslideToclabelSet` 用于设置 slide 模式下目录中各条目的格式. `<extra width>` 表示 `\thecontentslabel` 右侧额外的间距. `<item>` 可以选择使用 `\thecontentslabel`, `\zslidesecIcon`, `\zslidesubsecIcon` 或这是其它用户自定义的符号.

---

`\zslideDocTextColor`    `\zslideDocTextColor{<color>}`

---

Updated: 2024-11-05

此命令用于覆盖原本的 slide 文本色彩为 `<color>`. 用户不应该滥用此命令.

---

`\zslideDefaultUR`    `\zslideDefaultUR`  
`\zslideDefaultBR`    `\zslideDefaultBR`

---

Updated: 2024-11-05

此二命令表示 slide 模式下, UR, BR 位置处默认的元信息.

---

`\zslideIfPageTF`    `\zslideIfPageTF{<formula>}{<true code>}{<false code>}`

---

Updated: 2024-11-05

此命令此命令在自定义 slide 的元信息时很有用, 其会自动比较当前页码与 `<formula>` 的关系, 然后执行对应的分支. 一个使用样例如下:

```
\zslideThemeUse[
  UR={text=\zslideIfPageTF{=1}{}\{\zslideDefaultUR:\ \zslideNavigateBall}\},
]{AnnArborSpruce}
```

---

`\zslideSetup`    `\zslideSetup[<key>]{<spec>}`

---

Updated: 2024-11-05

此命令用于设置 slide 模式下, 用于调整 z<sub>La</sub>T<sub>E</sub>X 中关于 slide 相关的配置. `<key>` 表示 z<sub>La</sub>T<sub>E</sub>X 中 `<key-value>` 的层级名称. 此参数的所有合法选项仅有 `logo`, 或者置为空. `<spec>` 表示对具体项目的设置, 所有可用的选项如下:

```
\zslideSetup{
  doc = {\bf-color}, <text-color>, <text-style>
  sec = {\bg}, <fg>, <prefix>, <suffix>,
  UL  = {\bg}, <fg>, <text>,
  % 'UR', 'BL', 'BC', 'BR' are the same as 'UL'
  toc = {
    leftmargin = {
      chapter = <dim>,
      section = <dim>,
      subsection = <dim>
    },
    label = {
      % use command \zslideToclabelSet to set
      chapter = <item>,
      section = <item>,
      subsection = <item>
    },
  },
}
```

```

    suffix = {
      chapter = <item>,
      section = <item>,
      subsection = <item>
    },
  }
}

```

---

`\zslideThemeCreate`


---

Updated: 2024-11-05

`\zslideThemeCreate{<name>}spec`

此命令会按照 `<spec>` 创建名为 `<name>` 的 slide 颜色主题。具体的可调整选项请参见命令 `\zslideSetup` 中的 `<spec>` 参数说明。

---

`\zslideThemeUse`


---

Updated: 2024-11-05

`\zslideThemeUse[<key-value>]{<name>}`

此命令仅能在导言区使用，其会根据 `<spec>` 对颜色主题 `<name>` 中的部分配置进行重写，然后再应用 `<name>` 这一 slide 主题。

注意：为了编译速度考虑，用户应在加载 `zlatex` 时便通过 `<theme>` 参数指定 slide 的主题，且 `zLATEX` 仅加载一个主题，命令 `\zslideThemeUse` 更大程度上是处于用户便于修改预定义主题而提供的。

---

`AnnArborDefault`


---

Updated: 2024-11-05

```

\documentclass[layout=slide, theme=AnnArborDefault]zlatex
\zslideThemeUse[<spec>]{AnnArborDefault}

```

可以在加载文档类时选择此主题，还可以使用上述命令根据 `<spec>` 对此主题进行部分配置进行重定义。本主题具体效果请前往 Beamer Theme Matrix 查看。

---

`AnnArborBeaver`


---

Updated: 2024-11-05

```

\documentclass[layout=slide, theme=AnnArborBeaver]zlatex
\zslideThemeUse[<spec>]{AnnArborBeaver}

```

可以在加载文档类时选择此主题，还可以使用上述命令根据 `<spec>` 对此主题进行部分配置进行重定义。本主题具体效果请前往 Beamer Theme Matrix 查看。

---

`AnnArborAlbatross`


---

Updated: 2024-11-05

```

\documentclass[layout=slide, theme=AnnArborAlbatross]zlatex
\zslideThemeUse[<spec>]{AnnArborAlbatross}

```

可以在加载文档类时选择此主题，还可以使用上述命令根据 `<spec>` 对此主题进行部分配置进行重定义。本主题具体效果请前往 Beamer Theme Matrix 查看。

---

`AnnArborSeahorse`


---

Updated: 2024-11-05

```

\documentclass[layout=slide, theme=AnnArborSeahorse]zlatex
\zslideThemeUse[<spec>]{AnnArborSeahorse}

```

可以在加载文档类时选择此主题，还可以使用上述命令根据 `<spec>` 对此主题进行部分配置进行重定义。本主题具体效果请前往 Beamer Theme Matrix 查看。

---

`AnnArborSpruce`


---

Updated: 2024-12-05

```

\documentclass[layout=slide, theme=AnnArborSpruce]zlatex
\zslideThemeUse[<spec>]{AnnArborSpruce}

```

可以在加载文档类时选择此主题，还可以使用上述命令根据 `<spec>` 对此主题进行部分配置进行重定义。本主题具体效果请前往 Beamer Theme Matrix 查看。

## 7.4 theme

本 library 中定义了一系列的定理类主题, 在加载 theme library 的同时, 会自动导入 tcolorbox, tikz 和 pifont 三个宏包. 同时也会加载 tikz 的 fadings, calc 两个库. 如此数量的宏包导入必然会拖慢整个文档的编译, 请酌情考虑加载此 library.

---

shadow

---

 $\backslash\text{zlatexThmStyle}\{\text{shadow}\}$ 


---

Updated: 2024-12-05

---

加载此 library 后即可应用上述样式, 样式预览如下:

```
% \zlatexloadlibrary{mathalias}
\begin{remark}[AAA-7]
As any dedicated reader can clearly see, the Ideal of practical
reason is a representation of, as far as I know, the things in themselves;
\begin{align}
\underset{}{\mathbf{v}} \mathbf{\bigotimes} \mathbf{w} &= \sum_{i=1}^3 \left( a_{i1} u^i v^1 + a_{i2} u^i v^2 + a_{i3} u^i v^3 \right) \\
&= \int x \, dx = \frac{1}{2} x^2 + \mathbf{R}\{C\}
\end{align}
As any dedicated reader can clearly see, the Ideal of practical
reason is a representation of, as far as I know, the things in themselves;%
\end{remark}
```

注记 7.1 (AAA-7) As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves;

$$\mathbf{v} \mathbf{\bigotimes} \mathbf{w} = \sum_{i=1}^3 (a_{i1} u^i v^1 + a_{i2} u^i v^2 + a_{i3} u^i v^3) \quad (7.1)$$

$$= \int x \, dx = \frac{1}{2} x^2 + C \quad (7.2)$$

As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves;

---

paris

---

 $\backslash\text{zlatexThmStyle}\{\text{paris}\}$ 


---

Updated: 2024-12-05

---

加载此 library 后即可应用上述样式, 样式预览如下:

```
% \zlatexloadlibrary{mathalias}
\begin{axiom}[BBB-7]
As any dedicated reader can clearly see, the Ideal of practical
reason is a representation of, as far as I know, the things in themselves;
\begin{align}
\underset{}{\mathbf{v}} \mathbf{\bigotimes} \mathbf{w} &= \sum_{i=1}^3 \left( a_{i1} u^i v^1 + a_{i2} u^i v^2 + a_{i3} u^i v^3 \right) \\
&= \int x \, dx = \frac{1}{2} x^2 + \mathbf{R}\{C\}
\end{align}
```

As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves;  
`\end{axiom}`

**公理 7.1 (BBB-7)** As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves;

$$\mathbf{v} \otimes \mathbf{w} = \sum_{i=1}^3 (a_{i1} u^i v^1 + a_{i2} u^i v^2 + a_{i3} u^i v^3) \quad (7.3)$$

$$= \int x \, dx = \frac{1}{2} x^2 + C \quad (7.4)$$

As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves;

obsidian

`\zlatexThmStyle{obsidian}`

Updated: 2024-12-05

加载此 library 后即可应用上述样式, 样式预览如下:

```
% \zlatexloadlibrary{mathalias}
\begin{proposition}[CCC-4]
As any dedicated reader can clearly see, the Ideal of practical
reason is a representation of, as far as I know, the things in themselves;
\begin{align}
\underset{}{\mathbf{v}} \mathbin{\bigotimes} \mathbf{w} &= \sum_{i=1}^3 \left( a_{i1} u^i v^1 + a_{i2} u^i v^2 + a_{i3} u^i v^3 \right) \\
&= \int x \, dx = \frac{1}{2} x^2 + C
\end{align}
As any dedicated reader can clearly see, the Ideal of practical
reason is a representation of, as far as I know, the things in themselves;
\end{proposition}
```

“命题:7.1 CCC-4 ▽

As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves;

$$\mathbf{v} \otimes \mathbf{w} = \sum_{i=1}^3 (a_{i1} u^i v^1 + a_{i2} u^i v^2 + a_{i3} u^i v^3) \quad (7.5)$$

$$= \int x \, dx = \frac{1}{2} x^2 + C \quad (7.6)$$

As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves;



lapsis

\zlatexThmStyle{lapsis}

Updated: 2024-12-05

加载此 library 后即可应用上述样式, 样式预览如下:

```
% \zlatexloadlibrary{mathalias}
\begin{lemma}[DDD-1]
As any dedicated reader can clearly see, the Ideal of practical
reason is a representation of, as far as I know, the things in themselves;
\begin{align}
\underiset{}{\mathbf{v}} \bigotimes \mathbf{w}
&= \sum_{i=1}^3 \left( a_{i1} u^i v^1 + a_{i2} u^i v^2 + a_{i3} u^i v^3 \right) \\
&= \int x \, dx = \frac{1}{2} x^2 + \mathbf{R}\{C\}
\end{align}
As any dedicated reader can clearly see, the Ideal of practical
\tcblower
\begin{align}
\int x \, dx = \frac{1}{2} x^2 + \mathbf{R}\{C\}
\end{align}
reason is a representation of, as far as I know, the things in themselves;%
\end{lemma}
```

引理 7.1 As any dedicated reader can clearly see, the Ideal of practical reason is a  
DDD-1 representation of, as far as I know, the things in themselves;

$$\mathbf{v} \bigotimes \mathbf{w} = \sum_{i=1}^3 (a_{i1} u^i v^1 + a_{i2} u^i v^2 + a_{i3} u^i v^3) \quad (7.7)$$

$$= \int x \, dx = \frac{1}{2} x^2 + C \quad (7.8)$$

As any dedicated reader can clearly see, the Ideal of practical

$$\int x \, dx = \frac{1}{2} x^2 + C \quad (7.9)$$

reason is a representation of, as far as I know, the things in themselves; ◀

elegant

\zlatexThmStyle{elegant}

Updated: 2024-12-05

加载此 library 后即可应用上述样式, 样式预览如下:

```
% \zlatexloadlibrary{mathalias}
\begin{definition}[DDD-1]
As any dedicated reader can clearly see, the Ideal of practical
reason is a representation of, as far as I know, the things in themselves;
\begin{align}
\underiset{}{\mathbf{v}} \bigotimes \mathbf{w}
&= \sum_{i=1}^3 \left( a_{i1} u^i v^1 + a_{i2} u^i v^2 + a_{i3} u^i v^3 \right) \\
&= \int x \, dx = \frac{1}{2} x^2 + \mathbf{R}\{C\}
\end{align}
As any dedicated reader can clearly see, the Ideal of practical
\tcblower
\begin{align}
\int x \, dx = \frac{1}{2} x^2 + \mathbf{R}\{C\}
\end{align}
reason is a representation of, as far as I know, the things in themselves;%
\end{definition}
```

As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves;%  
`\end{definition}`

As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves;

$$= \int x \, dx = \frac{1}{2}x^2 + C \quad (7.11)$$

As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves;

# Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

Symbols		\Dda	31
-shell-escape	4	\dda	31
??	36	\DeclareMathOperator	34
\_zlatex_quad_dim	11, 36	\definecolor	18
A		\div	34
\A	32	\documentclass	38
\alt	34	E	
AnnArborAlbatross	38	\E	32
AnnArborBeaver	38	elegant	41
AnnArborDefault	38	F	
AnnArborSeahorse	38	\F	30
AnnArborSpruce	38	fancy	7, 8
B		\FF	30
\B	30	font	9
\backsimeq	33	foot	35
basic packages	MMMMI-3	\frontmatter	5, 16
BC	35	G	
bib commands:		\grad	34
bib_index	9	H	
BL	35	head	35
\blacktriangleright	12	\hla	32
\boldsymbol	30	\hla*	32
BR	35	\hom	34
C		\hra	32
\C	30	\hra*	32
\CC	33	hyper	8
\chapter	29	\hypersetup	14
\Cinzel	12	I	
class	8	\id	34
classOption	8	\im	34
\cok	34	\includegraphics	16
\colorlet	18	K	
\cong	33	\K	30
\ctexset{}	8	\ker	34
\curl	34	keys commands:	
D		\keys_define:nn	19
\Da	31	L	
\da	31	\La	31
\dd	33		



<b>T</b>		<code>\xxla*</code> . . . . .	32
T <sub>E</sub> X and L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> commands:		<code>\xxra</code> . . . . .	32
<code>\@addtoreset</code> . . . . .	11	<code>\xxra*</code> . . . . .	32
<code>\@author</code> . . . . .	35	<b>Z</b>	
<code>\@date</code> . . . . .	35	<code>\zab</code> . . . . .	34
<code>\@title</code> . . . . .	35	<code>\zLaTeX</code> . . . . .	3
<code>\hyper@anchor</code> . . . . .	13	zlatex internal commands:	
<code>\hyper@link</code> . . . . .	13	<code>\_zlatex_color_set:n</code> . . . . .	19
<code>\hyper@linkend</code> . . . . .	13	<code>\_zlatex_page_annotate:nnnnn</code> . .	16
<code>\hyper@linkstart</code> . . . . .	13	<code>\_zlatex_thm_warp_start:nnnn</code> . .	22
<code>zlatex@color@⟨name⟩</code> . . . . .	18	<code>\zlatexColorSetup</code> . . . . .	18, 21, 23
<code>\zsec@⟨name⟩@cnt</code> . . . . .	35, 36	<code>\zlatexCounterWith</code> . . . . .	11
<code>\text</code> . . . . .	32	<code>\zlatexFramed</code> . . . . .	11
<code>\thecontentslabel</code> . . . . .	36, 37	<code>\zlatexloadlibrary</code> . . . . .	7, 29
<code>theme</code> . . . . .	7	<code>\zlatexloadmodule</code> . . . . .	7, 12
<code>thm-hook.⟨Hook Index⟩</code> . . . . .	24	<code>\zlatexOptions</code> . . . . .	6
<code>\thmname</code> . . . . .	21	<code>\zlatexPageMask</code> . . . . .	16, 17
<code>\thmnote</code> . . . . .	21	<code>\zlatexPageMask*</code> . . . . .	16
<code>\thmnumber</code> . . . . .	21	<code>\zlatexPageMaskRemove</code> . . . . .	17
<code>toc</code> . . . . .	8	<code>\zlatexPartialToc</code> . . . . .	28
<code>\trace</code> . . . . .	34	<code>\zlatexSetup</code> . . . . .	6, 9
<b>U</b>		<code>\zlatexStopPartialToc</code> . . . . .	28
<code>UL</code> . . . . .	35	<code>\zlatexThmBefore</code> . . . . .	22
<code>UR</code> . . . . .	35	<code>\zlatexThmCnt</code> . . . . .	22
<code>\usepackage</code> . . . . .	4	<code>\zlatexThmColorSetup</code> . . . . .	23
<b>V</b>		<code>\zlatexThmCreate</code> . . . . .	20, 21
<code>\varnothing</code> . . . . .	33	<code>\zlatexThmHook</code> . . . . .	24
<code>\verb</code> . . . . .	11	<code>\zlatexThmHook*</code> . . . . .	24
<b>X</b>		<code>\zlatexThmLang</code> . . . . .	20
<code>\xhookleftarrow</code> . . . . .	32	<code>\zlatexThmName</code> . . . . .	21
<code>\xhookrightarrow</code> . . . . .	32	<code>\zlatexThmNote</code> . . . . .	21, 22
<code>\Xla</code> . . . . .	32	<code>\zlatexThmNoteEmptyTF</code> . . . . .	22
<code>\xla</code> . . . . .	32	<code>\zlatexThmNumber</code> . . . . .	21
<code>\Xla*</code> . . . . .	32	<code>\zlatexThmStyle</code> . . . . .	22, 23, 39–41
<code>\xla*</code> . . . . .	32	<code>\zlatexThmStyleNew</code> . . . . .	23
<code>\xLeftarrow</code> . . . . .	32	<code>\zlatexThmTitle</code> . . . . .	21, 22
<code>\xLeftrightarrow</code> . . . . .	32	<code>\zlatexThmTitleFormat</code> . . . . .	21, 22
<code>\xLongleftarrow</code> . . . . .	32	<code>\zlatexThmTitleFormat*</code> . . . . .	22
<code>\xLongrightarrow</code> . . . . .	32	<code>\zlatexThmTitleSwitch</code> . . . . .	21
<code>\Xra</code> . . . . .	32	<code>\zlatexThmToc</code> . . . . .	24
<code>\xra</code> . . . . .	32	<code>\zlatexThmTocAdd</code> . . . . .	25
<code>\Xra*</code> . . . . .	32	<code>\zlatexThmTocLevel</code> . . . . .	25
<code>\xra*</code> . . . . .	32	<code>\zlatexThmTocPrefix</code> . . . . .	25, 26
<code>\xrightarrow</code> . . . . .	32	<code>\zlatexThmTocSymbol</code> . . . . .	25, 26
<code>\xrightarrow</code> . . . . .	32	<code>\zlatexThmTocSymbolClear</code> . . . . .	26
<code>\xxla</code> . . . . .	32	<code>\zlatexVerb</code> . . . . .	11
		<code>\zph</code> . . . . .	17
		<code>\zpw</code> . . . . .	17

zslide commands:	\zslideIfPageTF .....	37
\_zslide_metadate:n .....	\zslidelogo .....	36
\_zslide_navigate:nnnn .....	\zslideNavigateBall .....	36
\_zslide_status_bar:nnnn .....	\zslidesecIcon .....	37
\_zslide_status_info:nnnn .....	\zslideSetup .....	37, 38
zslide-last-page .....	\zslidesubsecIcon .....	37
\zslideAuthor .....	\zslideThemeCreate .....	38
\zslideDate .....	\zslideThemeUse .....	38
\zslideDefaultBR .....	\zslideTitle .....	35
\zslideDefaultUR .....	\zslideToclabelSet .....	36, 37
\zslideDocTextColor .....	\zslideTocPage .....	36
zslideFrameCntAux commands:	\zslideTocSecIcon .....	36
\_zslideFrameCntAux:nn .....	\zslideTocSubsecIcon .....	36
\zslideFrameIndex .....	ztool .....	4
\zslideFrameTotal .....	\ZZ .....	10, 33