

zL^AT_EX 用户手册

Eureka

2025 年 2 月 26 日

- ▶ 基本介绍
- ▶ 安装使用
 - 在线使用
 - 本地安装
 - 最小工作示例
- ▶ z \LaTeX 配置
 - 前言
 - 宏包机制
 - 基本宏包
 - 导言区
- ▶ 杂项
- ▶ 状态检测
- ▶ Modules
 - fontcfg
 - indexref
 - layout
 - pageinfo
 - theme
 - thm
 - titlesec

- toc
- ▶ Libraries
 - fancy
 - mathalias
 - slide
 - theme

- ▶ ztool
 - l3sys-shell
 - 文件 IO 操作
 - Box 操作
- ▶ TODO

Index

219

基本介绍

 $\backslash\text{zLaTeX}$

Updated: 2024-11-05

 $\backslash\text{zLaTeX}$

用于输出本宏集对应的 logo，使用示例如下：

Hello $\backslash\text{zLaTeX}\{\}$; 你好 $\backslash\text{zLaTeX}\{\}$.

例 1

Hello $\text{z}\text{\LaTeX}$; 你好 $\text{z}\text{\LaTeX}$.

$\text{z}\text{\LaTeX}$ 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 $\langle class \rangle$ 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

z \LaTeX 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。z \LaTeX 全部由 \LaTeX 3 进行编写，采用 $\langle key-value \rangle$ 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 \LaTeX ，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。

安装使用

2.1 在线使用

为了让部分用户可以直接体验到 $\text{z}\LaTeX$ ，免去“繁杂”的环境配置. 我已将本模板部署在 TeXPage 上，地址为: [TeXPage \$\text{z}\LaTeX\$ 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.

2.2 本地安装

目前本文档类 $z\LaTeX$ 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 $\LaTeX 3$ 命令在老版本下并不存在，所以如果你的 \TeX Live 过于老旧，则可能出现无法编译的情况. 目前已知 $z\LaTeX$ 文档类在各平台的兼容情况为：

Windows : \TeX Live 最低版本 2022

Linux : \TeX Live 最低版本 2022

MacOS : 兼容 $\text{MacTeX} 2024$ (老版也应兼容)

`ztool`

`Updated: 2024-12-05`

`\usepackage{ztool}`

本宏集已独立实现了一个 `ztool` 宏包, 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, $\text{z}\text{\LaTeX}$ 能够避免或减少 `-shell-escape` 相关的调用. `ztool` 宏包手册请参见 [节 8](#).

由于 $\text{z}\text{\LaTeX}$ 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/<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 $\langle zT_{\text{E}}X \rangle$; 然后把 `zlatex` 目录中的所有内容放入 $\langle zT_{\text{E}}X \rangle$ 下.

在本手册后续, 我们使用 $\langle zT_{\text{E}}X \rangle$ 表示本宏集
的根目录.

2.3 最小工作示例

z \LaTeX 的最小工作示例如下¹. 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.

```
% !TeX program = XeLaTeX
\documentclass[lang=cn]{zlatex}

\begin{document}
% some preface
% \tableofcontents
```

例 2

¹ 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

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

其次是英文写作示例, 然后更改此时的基本文档类为 book, 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(此为默认选项), 其次便是把编译方式改为 pdf \LaTeX .

```
% !TeX program = pdfLaTeX  
\documentclass[class=book]{zlatex}  
  
\title{<title>}  
\author{<author>}
```

例 3

```
\date{<date>}  
\begin{document}  
\maketitle  
\frontmatter  
  
% some preface  
% \tableofcontents  
% some claim etc.  
\mainmatter  
  
% writing 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.

3.1 前言

在阅读本手册之前, 我们做如下约定:

- 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 z \LaTeX 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。

同时, 针对后续 $\text{z}\LaTeX$ 提供的一系列命令, 我们约定:

- 名字后带有 \star 号的命令, 可以在 x , e , f 型参数中被完全展开,
- 名字后带有 \star 号的命令, 只能在 x , e 型参数中被完全展开, 无法在 f 型参数中被完全展开;

3.2 宏包机制

$\text{z}\LaTeX$ 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 $\text{z}\LaTeX$ 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.

z^lTeX 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{zlatex-last-page}` (在页码正确的情况下, 超链接跳转可能并不正确, 这种情况下可以使用 `zlatex@lastpage` 这一 anchor).

3.3 基本宏包

z^lTeX 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:

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

表 1: $\text{z}\text{\LaTeX}$ 文档类基本宏包

$\text{z}\text{\LaTeX}$ 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, 不熟悉 \LaTeX 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 2.3](#)”的

最小写作示例.

3.4 导言区

`\zlatexSetup`

`\zlatexSetup{⟨key-value⟩}`

Updated: 2024-11-05

z^ΛT_EX 接受一系列的键值对进行配置, 部分的配置仅可以在加载文档类时指定.

`\zlatexOptions`

Updated: 2024-11-05

`\zlatexOptions`

`zATEX` 内置命令，用于打印此时文档类 `zATEX` 接收到的选项，可以在调试模板时使用。使用样例：

`\zlatexOptions`**例 4**

Class Options: cn - onside - 12pt

`zATEX` 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。`zATEX` 的 `⟨key-value⟩` 被划分为两个层级；第一层中的 `⟨layout⟩`, `⟨mathSpec⟩`, `⟨toc⟩`, `⟨packageOption⟩`, `⟨classOption⟩`, `⟨toc⟩`, `⟨font⟩` 均具有自己的独立子键 (`⟨sub-key⟩`),

个 $\langle key \rangle$ 的指定方式及其具体含义.

```
\zlatexloadmodule \zlatexloadmodule{\module name}
```

```
\zlatexloadlibrary \zlatexloadlibrary{\library name}
```

Updated: 2024-11-05

可以使用这两个命令用于加载 z^AT_EX 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:

- zlatex.module.
fontcfg.tex

- zlatex.module.
indexref.tex

- zlatex.module.

- zlatex.library.
fancy.tex

- zlatex.library.
mathalias.tex

- zlatex.library.

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}
```

例 5

你当然可以编写一个你自己的 module, 不妨假设其名称为 $\langle moduleA \rangle$; 将此文件命名为 `zlatex.module. $\langle moduleA \rangle$` . 然后将其放入路径 $\langle zT_{\text{E}}X \rangle / \text{module} /$ 下, 最后使用如下命令加载此 module:

```
\zlatexloadmodule{\moduleA}
```

例 6

 lang ☆

Updated: 2024-11-05

lang= \langle cn, **en** \rangle

z^ΛT_EX 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^ΛT_EX 会加载不同的和语言相关的宏包, 不同 \langle lang \rangle 设置下, 宏包的加载情况为:

 hyper ☆

Updated: 2024-11-05

hyper= \langle true, **false** \rangle

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.

lang=en	inputenc(pdf $\mathrm{T}_{\mathrm{E}}\mathrm{X}$)	fontenc	babel	microtype
lang=cn	fontspec	ctex		

表 2: z $\mathrm{L}_{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 文档类语言宏包

fancy ☆

Updated: 2024-11-05

fancy= $\langle true, \textcolor{red}{false} \rangle$

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=\textcolor{red}{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` ☆

Updated: 2024-11-05

`classOption=<class options>`

此选项接受一个逗号分隔的列表, 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}
```

例 7

`packageOption` ☆

Updated: 2024-11-20

`packageOption=<key-value>`

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}
```

例 8

`toc`

Updated: 2024-12-25

`toc=<key-value>`

此选项用于设置目录的样式，所有可选配置项如下：

例 9

```

\zlatexSetup{
  toc={
    column=\langle int:1 \rangle,
    title=\langle t1:Contents \rangle,
    title-vspace=\langle dim:-2em \rangle,
    stretch=\langle float:1 \rangle
  }
}

```

若上述的 $\langle column \rangle \geq 2$, 那么 $\text{z}\text{\LaTeX}$ 会自动加载 `multicol` 宏包. 注意: 由于在 \LaTeX 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:

例 10

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

font ☆

Updated: 2024-12-06

font = $\langle key-value \rangle$

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， $\text{z}\text{\LaTeX}$ 会自动加载 fontspec 宏包，此时需更换引擎为 $\text{X}\text{\LaTeX}$ 或者 $\text{Lua}\text{\LaTeX}$ 。目前所有可选值如下：

```
\documentclass[
  font={config=<true, false>}
]{zlatex}
```

例 11

 layout


layout=<key-value>

 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>
```

例 12

```
},  
]{zlatex}
```

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

`bib_index`

Updated: 2024-12-05

`bib_index=\langle key-value \rangle`

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=\langle bool:\textcolor{red}{false} \rangle,  
  }  
}
```

例 13


```

source=⟨str:ref.bib⟩,
backend=⟨str:biber⟩
}
}

```

⟨load⟩ 用于控制是否加载 biblatex 宏包, 默认为 false; ⟨source⟩ 用于指定参考文献源文件, 默认文件名为:ref.bib; ⟨backend⟩ 用于指定参考文献的 backend, 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{
```

例 14

```
  mathSpec={
```

```
    alias=\langle bool: false \rangle,
```

```
    envStyle=\langle tl: plain \rangle,
```

```
    font=\langle choice: ncmrm \rangle
```

```
  }
```

```
}
```

$\langle alias \rangle$ 默认为 `false`, 当置为 `true` 时, $\text{z}\text{\LaTeX}$ 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}`; $\langle envStyle \rangle$ 用于指定数学环境的样式, 默认为 `plain`, 为了编译速度考虑, 尽管 $\text{z}\text{\LaTeX}$ 预定义了一系列的环境样式, 但并不会预加载全部的样式, 而是把部分的样式移入了 `theme`

库中. 详细的预定义样式列表如下:

thm module 定义样式: **theme library 定义样式:**

- plain
- background
- leftbar
- fancy
- shadow
- paris
- elegant
- obsidian
- lapsis

$\langle font \rangle$ 用于指定数学公式字体, 预定义的字体有: newtx, euler, mtpo2, mathpazo. 其中 mtpo2

为付费字体，需用户手动安装.

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`

Updated: 2024-11-05

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

此命令和 $\text{\LaTeX}_{2\epsilon}$ 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}
```

例 15

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

`\graphicspath`

New: 2024-11-05

`\graphicspath{\path}`

此命令用于指定图片的搜索路径, 此命令来自 `graphicx` 宏包, 默认设置为:

```
\graphicspath{
  {.figure}{.figures}{.image}{.images}
  {.Pictures}{.picture}{.Pics}{.pics}
  {.graphics}{.graphic}
}
```

例 16

`\zlatexCounterWith`

Updated: 2024-11-05

`\zlatexCounterWith{\langle child \rangle}{\langle parent \rangle}`

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

`\counterwithin`

New: 2025-02-04

`\counterwithin{\langle child \rangle}{\langle parent \rangle}`

此命令用于给一个计数器 $\langle child \rangle$ 设置一个父计数器, 当 $\langle parent \rangle$ 计数器增加时, $\langle child \rangle$ 计数器会自动重置. 本命令和命令 `\@addtoreset` 的作用类似.

`\zlatexFramed`

`\zlatexFramed[$\langle color \rangle$]{ $\langle name \rangle$ }`Updated: 2024-11-05

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

`\zlatexFramed[red]{ref}`**例 17**`\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.

`\c_zlatex_quad_dim`

Updated: 2024-12-05

`\c_zlatex_quad_dim`

此命令表示当前文档中一个空格的宽度.

`zlatex-title-page`
`zlatex-last-page`

New: 2025-01-17

`\pageref{zlatex-title-page}`
`\pageref{zlatex-last-page}`

引用当前文档的最后一页, 可以在制作页眉页脚格式时使用. 但对应的超链接跳转也许并不正确, 此时应使用 `zlatex@lastpage` 这一 anchor. 一个基本的使用样例如下:

```
\pageref{zlatex-title-page}--\pageref{zlatex-例 18
-last-page}
```

0-240

`zlatex@titlepage`
`zlatex@lastpage`

New: 2025-01-17

`\hyper@link{\<context>}{zlatex@titlepage}{\<link
text>}`
`\hyper@link{\<context>}{zlatex@lastpage}{\<link
text>}`

上述两 Targets 由命令 `\hyper@anchor` 设置, 分别应用于引用当前文档的第一页和最后一页, 在 $\text{z}\text{\LaTeX}$ 中, 标题页的页码为 1.

注意: 普通用户不应该直接使用这两个 Targets, 此二 Targets 主要提供给模板的开发者, 用户应使用位于首页和尾页的 `zlatex-title-page` 和 `zlatex-last-page` 两 label.

<code>\zlatexTitle</code>	<code>\zlatexTitle</code>
<code>\zlatexAuthor</code>	<code>\zlatexAuthor</code>
<code>\zlatexDate</code>	<code>\zlatexDate</code>

Updated: 2024-11-05

此三个命令用于分别保存导言区 `\@title`, `\@author`, `\@date` 三个变量的值, 用户可以在正文部分使用此三个变量. 一个使用样例如下:

```
\zlatexTitle\par
\zlatexAuthor\par
\zlatexDate
```

例 19

```
zATEX 用户手册
Eureka
```

2025 年 2 月 26 日

状态检测

因 $\text{z}\text{\LaTeX}$ 的选项配置比较庞大，其中涉及到诸多的宏包和命令的加载，所以在文档编译时，我们需要对文档的各种状态进行检测， $\text{z}\text{\LaTeX}$ 提供了一系列的命令用于检测文档中各个变量状态的命令。

<code>\zlatexHyperTF</code>	★	<code>\zlatexHyperTF{\langle true code \rangle}{\langle false code \rangle}</code>
<code>\zlatexFancyTF</code>	★	此命令用于检测当前文档中是否开启了超链接功能，如果开启了，那么执行 <code>\langle true code \rangle</code> ，否则执行 <code>\langle false code \rangle</code> ，其余命令的使用方法同理。各个检测命令的基本使用样例如下：
<code>\zlatexMarginTF</code>	★	
<code>\zlatexSlideTF</code>	★	
<code>\zlatexFontConfigTF</code>	★	
<code>\zlatexMathAliasTF</code>	★	
<code>\zlatexBibIndexLoadTF</code>	★	
<code>\zlatexThemeLibLoadTF</code>	★	

New: 2025-01-15

例 20

```

\zlatexHyperTF{Hyperref is loaded.}{Hyperref
is not loaded.}\par
\zlatexFancyTF{Fancy is loaded.}{Fancy is
not loaded.}\par
\zlatexMarginTF{Margin does set.}{Margin
does NOT set.}\par
\zlatexSlideTF{Slide is loaded.}{Slide is
not loaded.}\par
\zlatexFontConfigTF{Font config is
loaded.}{Font config is not loaded.}\par
\zlatexMathAliasTF{Math alias is
loaded.}{Math alias is not loaded.}\par
\zlatexBibIndexLoadTF{Bib index is
loaded.}{Bib index is not loaded.}\par

```



```
\zlatexThemeLibLoadTF{Theme library is  
loaded.}{Theme library is not loaded.}
```



Hyperref is loaded.

Fancy is not loaded.

Margin does NOT set.

Slide is loaded.

Font config is not loaded.

Math alias is loaded.

Bib index is not loaded.

Theme library is loaded.

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 $\text{z}\text{\LaTeX}$ 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 \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 Xe^{La}TeX 或 Lua^{La}TeX), 本字体在 $\langle fancy \rangle = \text{true}$ 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 AMSa 字体, $\langle slot \rangle = "49$. 主要用于在 $\langle slide \rangle = \text{true}$ 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.

`\hyper@anchor`

New: 2024-12-05

`\hyper@anchor{⟨destination name⟩}`

此命令用于创建一个超链接锚点, `⟨destination name⟩` 作为后续超链接命令的跳转目标.

`\hyper@link`

New: 2024-12-05

`\hyper@link{\langle context \rangle}{\langle destination name \rangle}{\langle link text \rangle}`

此命令用于创建一个超链接, $\langle link\ text \rangle$ 本身作为一个超链接对象, 点击 $\langle link\ text \rangle$ 即可跳转到对应的 $\langle destination\ name \rangle$. $\langle context \rangle$ 表示此链接所属的类型, 默认有: link, url, cite 三种类型.

`\hyper@linkstart`

New: 2024-12-05

`\hyper@linkstart{\langle context \rangle}{\langle destination name \rangle}`

此命令用于开启一个超链接域, 此域中的内容可以是任意的文本或其它图片对象. 此命令需结合后续的 `\hyper@linkend` 命令使用, 此二命令结合使用时基本和上述的 `\hyper@link` 命令基本等效.

`\hyper@linkend`

New: 2024-12-05

`\hyper@linkend`

用于结束由 `\hyper@linkstart` 开启的域.

`\hyper@linkfile`

New: 2024-12-05

`\hyper@linkfile{\link
text}}{\filename}{\destname}`

此命令用于创建一个超链接, 点击 `\link text` 即可
跳转到对应的 `\filename` 文件中的 `\destname` 处.

<code>\MakeLinkTarget</code>	<code>\MakeLinkTarget[\langle prefix \rangle]{\langle counter \rangle}</code>
------------------------------	---

<code>\MakeLinkTarget*</code>	<code>\MakeLinkTarget*{\langle target \rangle}</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 $\langle prefix \rangle$ 和 $\langle counter \rangle$ 可以作为命令 `\hyper@link` 的参数使用. $\langle counter \rangle$ 可以为 chapter, section, subsection 等. 针对 `\MakeLinkTarget*`, 其中 $\langle target \rangle$ 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).

`\LinkTargetOn``\LinkTargetOn``\LinkTargetOff`

`\LinkTargetOff`New: 2024-12-05

此命令常在一个局部中用于取消由 `\MakeLinkTarget` 或 `\refstepcounter` 创建的 Target. 在使用 `\LinkTargetOff` 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{...}
...
```

例 21


```
{\LinkTargetOn\MakeLinkTarget*{mytarget}} % ✓
  create manual anchor for future reference
...
\LinkTargetOn
```

`\NextLinkTarget`

New: 2024-12-05

`\NextLinkTarget{<target>}`

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.

`\SetLinkTargetFilter`

New: 2024-12-05

`\SetLinkTargetFilter{\langle filter \rangle}`

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, 用户可以方便制作独特的页面样式以及水印添加.

`\orimaketitle`

`\orimaketitle`

Updated: 2024-11-05

`\maketitle` 的原始形式.

`\maketitle`

`\maketitle`

Updated: 2024-11-05

\LaTeX 对原始的 `\maketitle` 进行了重定义, 以适应不同的文档类和页面布局.

`\Maketitle`

New: 2024-12-05

`\Maketitle[\langle margin:1in \rangle]`

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 `\maketitle` 的原始定义， \langle margin \rangle 表示新的页面布局的 margin，默认为 1in. 此命令的实现为：

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

例 22

`\frontmatter` `\frontmatter`
`\mainmatter` `\mainmatter`

Updated: 2024-11-05

此二命令用于设置文档的前言和正文部分，在 $\text{z}\text{L}\text{A}\text{T}\text{E}\text{X}$ 中这两个命令已经被重定义，当加载的 $\langle class \rangle$ 为 `book` 或 `ctexbook` 时，这两个命令会自动处理页眉页脚相关设置.

`_zlatex_page_annotate:nnnnn` `_zlatex_page_annotate:nnnnn`

Updated: 2024-12-05

 $\{ \langle fore/background \rangle \}$
 $\{ \langle position \rangle \} \{ \langle anchor \rangle \}$
 $\{ \langle object \rangle \} \{ \langle hook\ range \rangle \}$

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.

`\zlatexPageMask``\zlatexPageMask*`

Updated: 2024-12-05

`\zlatexPageMask[⟨key-value⟩]{⟨item⟩}`

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

`\zlatexPageMask[`**例 23**`layer=⟨tl:background, foreground⟩,``position=⟨tl:(dim1, dim2)⟩,``label=⟨tl:DEFAULT⟩,``anchor=⟨tl:c, tl, tc, tr, bl, bc, br, etc⟩``]{⟨item⟩}`

其中 $\langle position \rangle$ 以页面的左下角为原点, 向上向右为正方向. $\langle anchor \rangle$ 为两个字符: 一个为水平位置, 另一个为垂直位置. 水平位置包括: 左 (l)、中 (c)、右 (r)、内侧 (i)、外侧 (o); 垂直位置包括: 顶部 (t)、中部 (m)、底部 (b).

注意: `transparent` 宏包仅可以在 `pdf \LaTeX` 和 `Lua \LaTeX` 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-log
  o.pdf}
```

例 24


```
}  
\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.

`\zpw``\zpw``\zph`

`\zph`

New: 2024-12-05

此二命令表示当前纸张的宽和高, 命令原型为 `\paperwidth` 和 `\paperheight`.

`\zlatexPageMaskRemove`

`\zlatexPageMaskRemove{\langle foreground/background \rangle}{\langle label \rangle}`

New: 2024-12-05

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, \LaTeX 会抛出警告.

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 $\langle hype \rangle = \text{true}$ 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. $\text{z}\text{\LaTeX}$ 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`;
- 数学环境类: `axiom`, `definition`, `theorem`, `lemma`, `corollary`, `proposition`, `remark`, `proof`, `exercise`, `example`, `solution`, `problem`.

在颜色指定上, $\text{z}\text{\LaTeX}$ 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

z \LaTeX 部分默认配色² 如下:





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

表 3: z \LaTeX 文档类默认配色

定理类环境的色彩保存于变量 `zlatex@color@<name>`

²zchapColor 还未整理, 目前只能单独重定义

中, 其中 $\langle name \rangle$ 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, $\text{z}\text{\LaTeX}$ 鼓励用户通过 `\zlatexColorSetup` 命令进行色彩的重定义.

注意: `\zlatexThmColorSetup` 仅能用于数学类环境的色彩自定义, 所以对于 `link`, `chapter` 等色彩请使用 `\zlatexColorSetup` 命令进行设置.

`_zlatex_color_set:n`

Updated: 2024-12-05

`_zlatex_color_set:n {<color spec>}`

此命令可以自动解析 $\langle color\ spec \rangle$, 并以此创建或定义对应的色彩. $\langle color\ spec \rangle$ 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 $\langle color\ spec \rangle$ 的使用, 可以参见如下示例:

`\ExplSyntaxOn`
`\keys_define:nn {colorTest}{`
`keyA .tl_set:N =`
`\l__zlatex_keyA_color_tl,`
例 26

```
keyA .code:n = {  
  \_zlatex_color_set:n {#1} },  
}  
\keys_set:nn {colorTest}{keyA={HTML}{d9d9d9}}  
\textcolor{zlatex@color@keyA}{This~is~a~test.}  
\ExplSyntaxOff
```

This is a test.

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.

thm 提供的数学环境主要分为两类:

- 定理类: axiom, definition, theorem, lemma, corollary, proposition, remark;
- 证明类: proof, exercise, example, solution, problem

所以请区分“定理类”和“证明类”两类环境, 以便于正确地使用 thm 提供的各个命令. $\text{z}\text{\LaTeX}$ 的 thm module 中的部分命令或变量也许没有显式地含有 **theorem** 字样, 但是这些命令或变量仍然是属于“定理类”的; 应用于“证明类”环境的命令或变量均显式地含有 **proof** 字样.

 \zlatexThmLang

Updated: 2024-11-05

 $\text{\zlatexThmLang}\{\langle\text{lang}\rangle\}$

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:

例 27

```
\begin{theorem}[zlatexThmLang-1]
  This is a chinese zlatexThmLang-1.
\end{theorem}
\zlatexThmLang{fr}
\begin{theorem}[zlatexThmLang-2]
  This is a france zlatexThmLang-2.
\end{theorem}
\zlatexThmLang{en}
\begin{theorem}[zlatexThmLang-3]
  This is a english zlatexThmLang-3.
\end{theorem}
```

定理 6.1 (zlatexThmLang-1) This is a chinese
zlatexThmLang-1.

Théorème 6.2 (zlatexThmLang-2) This is a
france zlatexThmLang-2.

Theorem 6.3 (zlatexThmLang-3) This is a english
zlatexThmLang-3.

`\zlatexThmNameSet`

New: 2025-02-04

`\zlatexThmNameSet{⟨lang⟩}{⟨key-value⟩}`

此命令用于设置数学环境的名称, 包括“定理类”和“证明类”. 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. 预定义的 $\langle lang \rangle$ 值有: en, cn, fr, 用户可以使用此命令自行配置, 然后使用命令 `\zlatexThmLang{⟨lang⟩}` 进行切换. $\langle key-value \rangle$ 的所有可用键即为所有的预定义数学环境名称. 一个使用样例如下:

`\zlatexThmNameSet{cn}{`
`theorem=新定理,`
例 28

```
proof=证
}
\zlatexThmLang{cn}
\begin{theorem}[zlatexThmNameSet-1]
  This is a theorem zlatexThmNameSet-1.
\end{theorem}
\begin{proof}
  This is a proof.
\end{proof}
```

新定理 6.4 (zlatexThmNameSet-1) This is a theorem zlatexThmNameSet-1.

证: This is a proof.



注意: 此命令需应用于 `\zlatexThmLang` 命令之前, 否则此命令的相关设置将不会生效.

`\zlatexThmCreate`

Updated: 2025-02-04

`\zlatexThmCreate[⟨type:theorem⟩]{⟨spec⟩}`

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. 命令 `\zlatexThmCreate` 仅可在导言区使用, 但为了说明其使用方法, 在本手册中, 此命令的定义被临时改变了. 和前面的 `\zlatexColorSetup` 命令类似, 在指定色彩时可以不必要提前定义. 针对 $\langle spec \rangle$, 请参见下面的使用样例:

例 29

```
\zlatexThmCreate{Zaxiom,  
Ztheorem=Thm|{HTML}{a0d911},  
Zproposition=Prop|blue}  
\zlatexThmCreate[proof]{Zproof,  
Zexample=EXAMPLE|red, Zsolution=Solution|}  
\begin{Zproof}[zlatexThmCreate-1]  
  This is a Zproof zlatexThmCreate-1.  
\end{Zproof}  
\begin{Zexample}[zlatexThmCreate-2]  
  This is a Zexample zlatexThmCreate-2.  
\end{Zexample}  
\begin{Ztheorem}[zlatexThmCreate-3]  
  This is a Ztheorem zlatexThmCreate-3  
\end{Ztheorem}
```



Zproof: This is a Zproof zlatexThmCreate-1.

EXAMPLE: This is a Zexample zlatexThmCreate-2.

Thm 6.1 (zlatexThmCreate-3) This is a Ztheorem zlatexThmCreate-3

`\zlatexThmNumber` ★

Updated: 2024-11-05

`\zlatexThmNumber`

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, 在命令 `\zlatexThmTitleFormat` 之外, 此命令输出的内容无任何实际意义.

`\zlatexThmName` ★

Updated: 2024-11-05

`\zlatexThmName`

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, 在命令 `\zlatexThmTitleFormat` 之外, 此命令输出的内容无任何实际意义.

`\zlatexThmNote` ★

Updated: 2024-12-05

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

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, 在命令 `\zlatexThmTitleFormat` 之外, 此命令输出的内容无任何实际意义.

`\thm@temp@color``\thm@proof@temp@color`

New: 2025-02-04

`\thm@temp@color``\thm@proof@temp@color`

此二命令用于临时保存定理类环境和证明类环境的色彩, 用于在 `\zlatexThmTitleFormat` 中进行色彩切换. 注意: 普通用户在使用这两个命令时, 请将其置于 `\makeatletter` 和 `\makeatother` 之间.

```
\zlatexThmTitle  ★ \zlatexThmTitle
\zlatexThmTitle* ★ \zlatexThmTitle*
```

Updated: 2024-11-05

`\zlatexThmTitle` 命令为定理类环境纯文本标题, 包含 `\zlatexThmNumber`, `\zlatexThmName`, `\zlatexThmNote` 三部分以及一些其它文本. `\zlatexThmTitle*` 为 `\zlatexThmTitle` 的格式化版本 (可能包含 `\bfseries`, `\sffamily` 等格式化命令); 用户在自定义定理类环境样式时应优先使用 `\zlatexThmTitle*`, 此命令生成的定理类环境标题才能被 `\zlatexThmTitleFormat` 控制. 此二命令中文本的具体格式可以使用 `\zlatexThmTitleForma` 进行指定.

`\zlatexThmTitleSwitch`
`\zlatexThmTitleSwitch*`

Updated: 2024-11-05

`\zlatexThmTitleSwitch`

命令 `\zlatexThmTitleSwitch` 用于隐藏定理类环境的标题, 命令 `\zlatexThmTitleSwitch*` 用于显示标题; 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}[zlatexThmTitleSwitch-1]
```

例 30

```
  A theorem zlatexThmTitleSwitch-1.
```

```
\end{theorem}
```

```
\zlatexThmStyleNew{
```

```
  ZZZ={begin=, end=,
```

```
  option=\zlatexThmTitleSwitch},
```

```
}
```



```
\zlatexThmStyle{ZZZ}  
\begin{theorem}[zlatexThmTitleSwitch-2]  
  A theorem zlatexThmTitleSwitch-2.  
\end{theorem}
```

定理 6.5 (zlatexThmTitleSwitch-1) A theorem
zlatexThmTitleSwitch-1.
A theorem zlatexThmTitleSwitch-2.

关于命令 `\zlatexThmStyle` 的使用可以参见下面的说明.

 $\backslash\text{zlatexThmTitleFormat}$
 $\backslash\text{zlatexThmTitleFormat*}$

Updated: 2025-02-04

 $\backslash\text{zlatexThmTitleFormat}[\langle type : \textcolor{red}{theorem} \rangle]\{\langle format \rangle\}$

此命令用于修改类型为 $\langle type \rangle$ 的数学类环境的标题格式, 即命令 $\backslash\text{zlatexThmTitle}$ 的内容, 仅能在导言区使用. $\langle type \rangle$ 的可选值有: theorem, proof, 默认值为 theorem. 命令 $\backslash\text{zlatexThmTitleFormat}$ 仅应用于之后的第一个 (类型为 $\langle type \rangle$ 的) 数学类环境标题样式, 而 $\backslash\text{zlatexThmTitleFormat*}$ 则应用于之后的所有 (类型为 $\langle type \rangle$ 的) 数学类环境. $\langle format \rangle$ 的默认格式为 (未考虑 $\backslash\text{zlatexThmNote}$ 为空的情况下):

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

例 31

此命令的一个简单使用案例如下:

```
\zlatexThmColorSetup{proof=blue!50}
\makeatletter
\zlatexThmTitleFormat{\bfseries\color{\thm@t
emp@color}\zlatexThmName\zlatexThmNote{\{ }\{ \
}}\zlatexThmNumber\
}
\zlatexThmTitleFormat[proof]{\color{\thm@pro
of@temp@color}\bfseries[:\zlatexThmName:] \
}
```

例 32


```

\makeatother
\begin{theorem}[zlatexThmTitleFormat-1]
  A theorem zlatexThmTitleFormat-1.
\end{theorem}
\begin{proof}
  This is a proof.
\end{proof}

```

定理{zlatexThmTitleFormat-1}6.7 A theorem
zlatexThmTitleFormat-1.

[: 证明:] This is a proof. □

此外, 还可以参见命令 `\zlatexThmNoteEmptyTF` 中的使用示例.

注意: 如果 $\langle type \rangle$ 为 `proof`, 那么在 $\langle format \rangle$ 中仅有 `\zlatexThmName` 和 `\thm@proof@temp@color` 可用.

`\zlatexThmNoteEmptyTF`

Updated: 2024-12-05

`\zlatexThmNoteEmptyTF{\langle true code \rangle}{\langle false code \rangle}`

此命令用于判断 `\zlatexThmNote` 是否为空, 如果为空则执行 $\langle true code \rangle$, 否则执行 $\langle false code \rangle$. 这个命令在自定义 `\zlatexThmTitle` 时很有用. 一个使用样例 (z \LaTeX 内置的 `obsidian` 定理样式对应的格式, 具体效果可以参见: [小节 7.4](#)):

例 33

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

`\zlatexThmBefore`

Updated: 2025-02-04

`\zlatexThmBefore[⟨type:theorem⟩]{⟨code:\par⟩}`

此命令用于把 $\langle code \rangle$ 置于每个类别为 $\langle type \rangle$ 的数学环境 (如果 $\langle type \rangle$ 为 `theorem`, 也就是命令 `_zlatex_thm_warp_start:nnnn`; 如果 $\langle type \rangle$ 为 `proof`, 那么就是 `_zlatex_thm_proof_title:`) 之前. $\langle type \rangle$ 的可选值有: `theorem`, `proof`, 默认值为 `theorem`. $\langle code \rangle$ 默认为 `\par`, 用户可以把 $\langle code \rangle$ 置为空, 或设置为 `\noindent` 以取消段落缩进. 一个简单的使用样例如下:

```
\zlatexThmBefore{}
```

```
Inline item:%
```

例 34

```
\begin{theorem}[zlatexThmBefore-1]
  This is a theorem.%
\end{theorem}%
\begin{proposition}[zlatexThmBefore-2]
  This is proposition I.
\end{proposition}
\begin{proof}
  This is a proof.
\end{proof}
```

Inline item: **定理 6.8** (zlatexThmBefore-1) This is
a theorem. **命题 6.1** (zlatexThmBefore-2) This is
proposition I.

证明: This is a proof. □

`\zlatexThmTitleBefore`

New: 2025-02-04

`\zlatexThmTitleBefore`[$\langle type: \text{theorem} \rangle$]{ $\langle code: \backslash noindent \rangle$ }

此命令用于把 $\langle code \rangle$ 置于每个类型为 $\langle type \rangle$ 的数学环境标题之前. $\langle type \rangle$ 的可选值有: theorem, proof, 默认值为 theorem. $\langle code \rangle$ 默认为 `\noindent`, 用户可以把 $\langle code \rangle$ 置为空以保留段落缩进. 一个简单的使用样例如下:

```

\zlatexThmTitleBefore[proof]{[PRF-LIKE]}
\begin{solution}
  This is solution \zlatexThmTitleBefore.
\end{solution}

```

例 35

[PRF-LIKE] 解: This is solution zlatexThmTitleBefore.

 $\backslash\text{qedsymbol}$

Updated: 2024-11-05

 $\backslash\text{qedsymbol}$

此命令用于输出证明环境的结束符号, 默认为 \square .

 $\backslash\text{zlatexThmCnt}$

Updated: 2024-11-05

 $\backslash\text{zlatexThmCnt}\{\langle\text{key-value}\rangle\}$

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{
```

例 36

```
  parent=\langle counter:section\rangle,
```

```
  share=\langle bool:false\rangle
```

```
}
```

$\langle parent \rangle$ 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; $\langle share \rangle$ 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: 2025-01-06

`\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（比如，你不能在此命令中设置 `⟨link⟩` 对应的色彩），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` 中的数学环境或本节开头的介绍，如果其中的 `⟨key⟩` 不在已有的定理类或者是证明类环境（包括由命令 `\zlatexThmCreate` 所创建的环境）中，那么 `zLATEX` 会抛出错误。

使用方法可以参见命令 `\zlatexThmTitleFormat` 中的示例。

`\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，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。由于技术原因，当用户需要加载后续的 theme library 时，必须将命令 `\zlatexThmStyle` 置于 `\zlatexloadlibrary{theme}` 之前。可用的 `<style>` 请参见 `<mathSpec>` 的说明。一个基本的使用样例：

```
\zlatexThmStyle{plain}
\begin{theorem}[zlatexThmStyle-1]
  A `plain' style zlatexThmStyle-1.
\end{theorem}
```

例 37

```
\zlatexThmStyle{leftbar}
\begin{theorem}[zlatexThmStyle-2]
  A `leftbar' style zlatexThmStyle-2.
\end{theorem}
\zlatexThmStyle{background}
\begin{theorem}[zlatexThmStyle-3]
  A `background' style zlatexThmStyle-3.
\end{theorem}
\zlatexThmStyle{fancy}
\begin{theorem}[zlatexThmStyle-4]
  A `fancy' style zlatexThmStyle-4.
\end{theorem}
```

定理 6.9 (zlatexThmStyle-1) A ‘plain’ style zlatexThmStyle-1.

定理 6.10 (zlatexThmStyle-2) A ‘leftbar’ style zlatexThmStyle-2.

定理 6.11 (zlatexThmStyle-3) A ‘background’ style zlatexThmStyle-3.

定理 6.12 (zlatexThmStyle-4) A ‘fancy’ style zlatexThmStyle-4.

`\zlatexThmStyleNew`

Updated: 2024-12-05

`\zlatexThmStyleNew{⟨key-value⟩}`

此命令用于定义新的定理类环境样式, 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. 此命令的调用格式如下:

`\zlatexThmStyleNew{`**例 38**`⟨style A⟩={``begin=⟨begin code 1⟩,``end=⟨end code 1⟩,``option=⟨option 1⟩,``preamble=⟨preamble code 1⟩``},``⟨style B⟩={`

```
begin= $\langle begin code 2 \rangle$ ,  
end= $\langle end code 2 \rangle$ ,  
option= $\langle option 2 \rangle$ ,  
preamble= $\langle preamble code 2 \rangle$   
},  
...  
}
```

在声明对应的 $\langle style \rangle$ 后，在导言区使用命令 $\backslash zlatexThmStyle{\langle style \rangle}$ 进行切换.

`\zlatexThmHook`
`\zlatexThmHook*`

 Updated: 2025-02-04

`\zlatexThmHook[⟨name: theorem⟩]{⟨key-value⟩}`
`\zlatexThmHook*[⟨name: theorem⟩]{⟨key-value⟩}`

此命令用于给已有的 (名称为 $\langle name \rangle$ 的) 定理类环境 Hook 中添加代码, $\langle name \rangle$ 的默认值为 `theorem`. 已有的 Hook: $\langle zlatex/thm/before \rangle$, $\langle zlatex/thm/begin \rangle$, $\langle zlatex/thm/end \rangle$, $\langle zlatex/thm/after \rangle$. `\zlatexThmHook` 只应用于下一个定理类环境, `\zlatexThmHook*` 会应用于接下来的所有定理类环境. 各个 Hook 的位置分布如下:

<pre>(zlatex/thm/before) --> (warper begin) --> (thm-title) --> (zlatex/thm/begin) --> (thm-content) --> (zlatex/thm/end) --></pre>	例 39
---	-------------

```
(warper end) --> (zlatex/thm/after)
```

这两个命令不支持手动设置 $\langle label \rangle$, 针对于 $\backslash zlatexThmHook*$, $z\LaTeX$ 会自动设置 $\langle label \rangle$, 其格式为 $thm-hook.\langle Hook\ Index \rangle$. 针对 $\langle key-value \rangle$, 所有合法的键选项为: $\langle before \rangle$, $\langle begin \rangle$, $\langle end \rangle$, $\langle after \rangle$, 一个使用样例如下:

```
\begin{theorem}[zlatexThmHook-1]
  This is a theorem zlatexThmHook-1.
\end{theorem}
\zlatexThmHook{before=ZZa\ , begin=ZZb\ ,}
\begin{theorem}[zlatexThmHook-2]
```

例 40


```

This is a theorem zlatexThmHook-2.
\end{theorem}

```

定理 6.13 (zlatexThmHook-1) This is a theorem
zlatexThmHook-1.

ZZa **定理 6.14 (zlatexThmHook-2)** ZZb This is a
theorem zlatexThmHook-2.

`\zlatexThmProofHook`
`\zlatexThmProofHook*`

New: 2025-02-04

`\zlatexThmProofHook[⟨name:proof⟩]{⟨key-value⟩}`
`\zlatexThmProofHook*[⟨name:proof⟩]{⟨key-value⟩}`

此命令用于给已有的 (名称为 $\langle name \rangle$ 的) 证明类环境

Hook 中添加代码, $\langle name \rangle$ 的默认值为 `proof`. 已有的

Hook: $\langle zlatex/proof/before \rangle$, $\langle zlatex/proof/begin \rangle$,

$\langle zlatex/proof/end \rangle$, $\langle zlatex/proof/after \rangle$. `\zlatexThmProofHo`

只应用于下一个证明类环境, `\zlatexThmProofHook*`

会应用于接下来的所有证明类环境. 各个 Hook 的位置分布如下:

```
(zlatex/proof/before) --> (proof-title)
--> (zlatex/proof/begin)
--> (proof-content) --> (zlatex/proof/end) -->
```

例 41

```
(env icon) --> (zlatex/proof/after)
```

和 `\zlatexThmHook`, `\zlatexThmHook*` 类似, 此二命令会自动设置对应的 `\label`, 无需用户手动指定. 针对 `\key-value`, 设置方法也是类似. 一个简单的使用样例如下:

```
\zlatexThmProofHook*[solution]{例 42
  before=\noindent\textbf{\color{red}BEFORE},
  begin=\textbf{\color{red}BEGIN},
  end=\textbf{\color{red}END},
  after=\textbf{\color{red}AFTER},
}
\begin{proof}
```

```
This is a proof.  
\end{proof}  
\begin{solution}  
  This is solution I.  
\end{solution}  
\begin{solution}  
  This is solution II.  
\end{solution}
```

证明: This is a proof. □

BEFORE 解: BEGINThis is solution I. **END**

AFTER

BEFORE 解: BEGINThis is solution II. **END**

AFTER

`\zlatexThmToc`

Updated: 2024-12-05

`\zlatexThmToc[stretch:1]`

此命令用于打印定理类环境的目录, $\langle stretch \rangle$ 为任意非负的浮点数, 用于指定定理目录的 stretch 值, 默认值为 1. 使用样例如下:

`\zlatexThmToc[1.25]`

例 43

`\begin{proposition}[zlatexThmToc-1]propositi` ✓

on

`zlatexThmToc-1\end{proposition}` ✓

`\begin{lemma}[zlatexThmToc-2]lemma` ✓

`zlatexThmToc-2\end{lemma}`

```
\begin{corollary}[zlatexThmToc-3]corollary
zlatexThmToc-3\end{corollary}
```



T 定理 6.1 (zlatexThmLang-1)

T Théorème 6.2

(zlatexThmLang-2)

T Theorem 6.3

(zlatexThmLang-3)

T 新 定 理 6.4

(zlatexThmNameSet-1)

Thm 6.1 (zlatexThmCreate-3)

T 定理 6.5

(zlatexThmTitleSwitch-1)

T 定理 6.6

(zlatexThmTitleSwitch-2)

T 定理 6.7

(zlatexThmTitleFormat-1)

T 定理 6.8 (zlatexThmBefore-1)

P 命题 6.1 (zlatexThmBefore-2)

T 定理 6.9 (zlatexThmStyle-1)

T 定理 6.10 (zlatexThmStyle-2)

T 定理 6.11 (zlatexThmStyle-3)

T 定理 6.12 (zlatexThmStyle-4)

T 定理 6.13 (zlatexThmHook-1)

T 定理 6.14 (zlatexThmHook-2)

P 命题 6.2 (zlatexThmToc-1)

L 引理 6.1 (zlatexThmToc-2)

C 推论 6.1 (zlatexThmToc-3)

New:Added Thm ITEM

R 注记 7.1 (zlatexThmStyle-5)

A 公理 7.1 (zlatexThmStyle-6)

P 命题 7.1 (zlatexThmStyle-7)

L 引理 7.1 (zlatexThmStyle-8)

D 定义 7.1 (zlatexThmStyle-9)

命题 6.2 (zlatexThmToc-1) proposition

zlatexThmToc-1

引理	6.1	(zlatexThmToc-2)	lemma
zlatexThmToc-2			
推论	6.1	(zlatexThmToc-3)	corollary
zlatexThmToc-3			

`\zlatexThmTocAdd`

Updated: 2024-12-05

`\zlatexThmTocAdd[⟨level:section⟩][⟨key-value⟩]`

此命令用于向定理类环境目录中添加条目, $\langle level \rangle$ 为计数器名, 表示该条目在目录中的层级. 可以为 `section`, `subsection` 等, 该可选参数默认为 `section`. $\langle key-value \rangle$ 为必选参数, 用于指定所添加的条目的其它信息. 一个使用样例如下:

```
\zlatexThmTocAdd[section]{name=New:Added Thm
ITEM}
```

例 44

`\zlatexThmTocStop`

Updated: 2024-12-05

`\zlatexThmTocStop`

此命令用于停止向定理类环境目录中添加条目，目前处于实验阶段，用户暂时不应使用此命令。

`\zlatexThmTocLevel`

Updated: 2024-12-05

`\zlatexThmTocLevel{⟨depth⟩}`

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， $\langle depth \rangle$ 为一个 ≥ 1 的整数。

`\zlatexThmTocPrefix`

Updated: 2024-12-05

`\zlatexThmTocPrefix{\langle prefix \rangle}`

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`

Updated: 2024-12-05

`\zlatexThmTocSymbol{\langle key-value \rangle}`

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, 命令 `\zlatexThmTocSymbolClear` 可以用于清除其设置的前缀. 所有可用的选项为:

`\zlatexThmTocSymbol{``\langle axiom: A \rangle,``\langle definition: D \rangle,``\langle theorem: T \rangle,`**例 45**

```

\langle lemma: L \rangle,
\langle corollary: C \rangle,
\langle proposition: P \rangle,
\langle remark: R \rangle,
}

```

`\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[$\langle depth \rangle$: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. $\langle depth \rangle$ 用于指定子目录最大深度, 默认为 2.

`\zlatexStopPartialToc`

Updated: 2024-11-05

`\zlatexStopPartialToc{\textcolor{red}{chapters},section}\{\langle counter \rangle\}`

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.

Libraries

本节主要介绍 $\text{z}\text{\LaTeX}$ 中提供的各类 library, 这些 library 用于优化用户 \LaTeX 的文档书写和阅读体验。部分 library 是对 $\text{z}\text{\LaTeX}$ 中原始功能的增强, 但同时, 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* 列表请参见命令 `\zlatexloadlibrary` 的说明.

7.1 fancy

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

 $\backslash\text{zlatexNumSuffix}$

Updated: 2025-02-04

 $\backslash\text{zlatexNumSuffix}\{\langle number \rangle\}$

此命令用于数字的格式化，其中 $\langle number \rangle$ 为一个任意的整数，一个使用样例如下：

$\backslash\text{zlatexNumSuffix}\{1\}$, $\backslash\text{zlatexNumSuffix}\{2\}$, 例 46
 $\backslash\text{zlatexNumSuffix}\{25\}$

st, nd, th

`\zsubtitle`

Updated: 2024-11-05

`\zsubtitle{\<subtitle>}`

当 fancy library 被加载时, 此命令用于设置章节的副标题; 若没有加载, 则此命令无效, 其参数会被吞掉.

`\zchapterLContent`

Updated: 2024-11-05

`\zchapterLContent{\<content>}`

当 fancy library 被加载时, 此命令用于设置章节的左侧内容; 若没有加载, 则此命令无效, 其参数会被吞掉.

`\zchapterRContent`

Updated: 2024-11-05

`\zchapterRContent{\langle content \rangle}`

当 fancy library 被加载时, 此命令用于设置章节的右侧内容; 若没有加载, 则此命令无效, 其参数会被吞掉.

`\zchapterSaying`

Updated: 2024-11-05

`\zchapterSaying[\langle author \rangle]{\langle saying \rangle}`

当 fancy library 被加载时, 此命令用于设置章节的引言. `\langle author \rangle` 为可选参数, 用于指定引言的作者.

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 `libray` 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。针对此特性，推荐用于使用 `pyhsics 2` 宏包.

WARNING: 此 library 加载的大量自定义命令可能会与已有的命令的冲突, 请谨慎加载此 library.

<code>\F</code>	<code>\F{\tokens}</code>
<code>\R</code>	<code>\R{\tokens}</code>
<code>\K</code>	<code>\K{\tokens}</code>
<code>\C</code>	<code>\C{\tokens}</code>
<code>\B</code>	<code>\B{\tokens}</code>
<code>\S</code>	<code>\S{\tokens}</code>
<code>\FF</code>	<code>\FF{\tokens}</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: $\mathbf{A} + \mathrm{A} + \text{例 47}$

$\mathfrak{a} + \mathcal{A} + \mathbb{A} +$ ✓
 $\mathscr{A} + \mathbf{A}$

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

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

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

此 library 定义的一系列命令遵循如下的规则:

- 首字母重复表示对应箭头的加长,
- 首字母大写表示对应箭头的双线版本,
- 前置 n 或 N 表示对应箭头的否定.

`\ma``\mma`

Updated: 2024-12-05

`\ma``\mma`

以上各命令的原始定义: `\ma` 为 `\mapsto`, `\mma` 为 `\longmapsto`. 注意: 此命令及其后续类似命令均表示该命令在未来可能会有改动, 比如未来其可能会接受参数。该命令的使用示例如下:

Normal Version: $a \mapsto b, a \longmapsto b$ **例 48**

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: $\$a\leftarrow b$, $a\Leftarrow b$ 例 49
 b , $a\rightarrowtail b$, $a\Leftarrowtail b$,
 $a\longleftarrow b$, $a\Longleftarrow b$ $\$ \backslash$
 Alias Version: $\$a\la b$, $a\La b$, $a\nla b$,
 $a\Nla b$, $a\lla b$, $a\Lla b$ $\$$.

Normal Version: $a \leftarrow b$, $a \Leftarrow b$, $a \rightarrowtail b$, $a \Leftarrowtail b$, $a \longleftarrow b$, $a \Longleftarrow b$
 b , $a \leftrightsquigarrow b$
 Alias Version: $a \leftarrow b$, $a \Leftarrow b$, $a \rightarrowtail b$, $a \Leftarrowtail b$, $a \longleftarrow b$, $a \Longleftarrow b$, $a \leftrightsquigarrow b$.

<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:

例 50

Normal Version: $\$a\rightarrow b$,

$a\rightarrow b$, $a\rightarrow b$,

$a\rightarrow b$, $a\rightarrow b$,

$a\rightarrow b$ $\$ \backslash$

Alias Version: $\$a\ra b$, $a\Ra b$, $a\hra b$,

$a\Nra b$, $a\rra b$, $a\Rra b$.

Normal Version: $a \rightarrow b, a \Rightarrow b, a \nrightarrow b, a \nRightarrow b, a \longrightarrow b, a \Longrightarrow b$

Alias Version: $a \rightarrow b, a \Rightarrow b, a \nrightarrow b, a \nRightarrow b, a \longrightarrow b, a \Longrightarrow b$.

<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 `\leftrightharrow`,
`\Da` 为 `\Leftrightarrow`, `\nda` 为 `\nleftrightharrow`,
`\Nda` 为 `\nLeftrightarrow`, `\dda` 为 `\longleftrightharrow`,
`\Dda` 为 `\Longleftrightharrow`. 使用示例如下:

Normal Version: $\$a\leftrightharpoons b$, 例 51
 $a\Leftrightarrow b$, $a\mathrel{\nleftarrow{\rightleftarrows}} b$, ✓
 $a\mathrel{\nleftarrow{\rightleftarrows}} b$, $a\longleftarrow b$, ✓
 $a\Longleftarrow b\$ \backslash\backslash$
 Alias Version: $\$a\da b$, $a\Da b$, $a\nda b$, ✓
 $a\Nda b$, $a\dda b$, $a\Dda b\$$.

Normal Version: $a \leftrightarrow b, a \Leftrightarrow b, a \mathrel{\leftrightarrow} b, a \mathrel{\nleftrightarrow} b, a \longleftrightarrow b, a \Longleftrightarrow b$
 $a \Longleftrightarrow b$
 Alias Version: $a \leftrightarrow b, a \Leftrightarrow b, a \mathrel{\leftrightarrow} b, a \mathrel{\nleftrightarrow} b, a \longleftrightarrow b, a \Longleftrightarrow b$.
 $a \Longleftrightarrow b$.

<code>\xla</code>	<code>\xla[\langle above \rangle](\langle below \rangle)</code>
<code>\xla*</code>	<code>\xla*[\langle above \rangle](\langle below \rangle)</code>
<code>\Xla</code>	<code>\Xla[\langle above \rangle](\langle below \rangle)</code>
<code>\Xla*</code>	<code>\Xla*[\langle above \rangle](\langle below \rangle)</code>
<code>\xxla</code>	<code>\xxla[\langle above \rangle](\langle below \rangle)</code>
<code>\xxla*</code>	<code>\xxla*[\langle above \rangle](\langle below \rangle)</code>
<code>\xra</code>	<code>\xra[\langle above \rangle](\langle below \rangle)</code>
<code>\xra*</code>	<code>\xra*[\langle above \rangle](\langle below \rangle)</code>
<code>\Xra</code>	<code>\Xra[\langle above \rangle](\langle below \rangle)</code>
<code>\Xra*</code>	<code>\Xra*[\langle above \rangle](\langle below \rangle)</code>
<code>\xxra</code>	<code>\xxra[\langle above \rangle](\langle below \rangle)</code>
<code>\xxra*</code>	<code>\xxra*[\langle above \rangle](\langle below \rangle)</code>

Updated: 2024-12-05

以上所有带有 `*` 命令中的 `\langle above \rangle` 和 `\langle below \rangle` 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`

例 52

Normal Version: $\xleftarrow[b]{a} + \xLeftarrow[b]{a} + \xLongleftarrow[b]{a} + \xrightarrow[b]{a} + \xRightarrow[b]{a} + \xLongrightarrow[b]{a}$ \checkmark
 $\xLeftarrow[b]{a} + \xLongleftarrow[b]{a} + \xrightarrow[b]{a} + \xRightarrow[b]{a} + \xLongrightarrow[b]{a}$ \checkmark
 $\xLongrightarrow[b]{a}$ $\$ \backslash$

Alias Version: $\xla[a](b) + \Xla[a](b) + \xxla[a](b) + \xra[a](b) + \Xra[a](b) + \xxra[a](b)$ \checkmark
 $\xla[a](b) + \xra[a](b) + \xxra[a](b)$ $\$ \backslash$

Alias Text Version: $\xla*[a](b) + \Xla*[a](b) + \xxla*[a](b) + \xra*[a](b) + \Xra*[a](b) + \xxra*[a](b)$ \checkmark
 $\xla*[a](b) + \xra*[a](b) + \xxra*[a](b)$ $\$$

Normal Version: $\frac{a}{b} + \xleftarrow[b]{a} + \xLongleftarrow[b]{a} + \frac{a}{b} + \xrightarrow[b]{a} + \xRightarrow[b]{a}$
Alias Version: $\xla[b]{a} + \xra[b]{a} + \xxra[b]{a} + \frac{a}{b} + \xRightarrow[b]{a} + \xLongrightarrow[b]{a}$

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

<code>\hla</code>	<code>\hla[\langle above \rangle](\langle below \rangle)</code>
<code>\hla*</code>	<code>\hla*[\langle above \rangle](\langle below \rangle)</code>
<code>\hra</code>	<code>\hra[\langle above \rangle](\langle below \rangle)</code>
<code>\hra*</code>	<code>\hra*[\langle above \rangle](\langle below \rangle)</code>

Updated: 2024-12-05

以上所有带有 `*` 命令中的 `\langle above \rangle` 和 `\langle below \rangle` 参数均会被放入 `\text` 命令中, 以上命令的原始定义:
`\hla` 为 `\xhookleftarrow`, `\hra` 为 `\xhookrightarrow`.
 使用示例如下:

Normal Version: `\hookrightarrow[b]{a}` + **例 53**

`\hookrightarrow[b]{a}`

Alias Version: `\hla[a](b)` + `\hra[a](b)`

Alias Text Version: `\hla*[a](b)` + `\hra*[a](b)`

Normal Version: $\hookrightarrow[b]{a} + \hookrightarrow[b]{a}$

Alias Version: $\hla[b]{a} + \hra[b]{a}$

Alias Text Version: $\hla*[b]{a} + \hra*[b]{a}$

 \forall \exists

Updated: 2024-12-05

 \forall \exists

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: $\forall \epsilon > 0, \exists \delta$ **例 54**

$\exists \delta > 0, \forall \epsilon > 0$

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

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

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

`\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 `\backsimeq`, `\sse` 为 `\cong`. 一个使用样例如下:

Normal Version: \varnothing, \backsimeq , 例 55

\cong

Alias Version: \ns, \se, \sse

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

Alias Version: \ns, \se, \sse

`\dd`

Updated: 2024-12-05

`\dd`

此命令主要用于替代默认的 `\mathrm{d}`, 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: `\displaystyle\int`**例 56**`x\;\mathrm{d}x = x^{\int x\mathrm{d} x} =` ✓`\frac{1}{2}x^2 + \mathrm{C}`Alias Version: `\displaystyle\int x\dd x =` ✓`x^{\int x\dd x} = \frac{1}{2}x^2 + \mathrm{C}`

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

$$\text{Alias Version: } \int x \, dx = x^{\int x dx} = \frac{1}{2}x^2 + C$$

`\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} **例 57**
`\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, 可以使用命令 `\zlatexMathAliasOpSet` 进行重定义. 一个使用样

例如下:

Normal Version: $\backslash\operatorname{alt}$, $\backslash\operatorname{rot}$, $\backslash\operatorname{div}$, $\backslash\operatorname{curl}$, $\backslash\operatorname{grad}$, $\backslash\operatorname{Id}$, $\backslash\operatorname{Im}$, $\backslash\operatorname{Ker}$, $\backslash\operatorname{Cok}$, $\backslash\operatorname{Hom}$, $\backslash\operatorname{sign}$, $\backslash\operatorname{trace}$

Alias Version: $\backslash\operatorname{alt}$, $\backslash\operatorname{rot}$, $\backslash\operatorname{div}$, $\backslash\operatorname{curl}$, $\backslash\operatorname{grad}$, $\backslash\operatorname{id}$, $\backslash\operatorname{im}$, $\backslash\operatorname{ker}$, $\backslash\operatorname{cok}$, $\backslash\operatorname{hom}$, $\backslash\operatorname{sign}$, $\backslash\operatorname{trace}$

Normal Version: alt, rot, div, curl, grad, Id, Im, Ker, Cok, Hom, sign, t
Alias Version: alt, rot, div, curl, grad, Id, Im, Ker, Cok, Hom, sign, tra

`\zlatexMathAliasOpSet`

Updated: 2024-12-05

`\zlatexMathAliasOpSet{⟨key-value⟩}`

此命令用于设置上述各数学算子的名称, 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. 一个使用样例如下:

`\[\alt, \im \]`**例 59**`\zlatexMathAliasOpSet{alt=ALT, im=IM}``\[\alt, \im \]`

alt, Im

ALT, IM

`\zab`

Updated: 2024-12-05

`\zab` $\langle () , [] , \{\} \rangle$

此命令用于处理括号的自动缩放, 该命令现在正处于实验性阶段, 可能存在潜在的问题, 请谨慎使用. 该命令的一个使用样例如下:

Normal Version: `\displaystyle \left(\frac{1}{2} \right), \left[\frac{1}{2} \right], \left\{ \frac{1}{2} \right\}` 例 60 ✓

Alias Version: `\displaystyle \zab(\frac{1}{2}), \zab[\frac{1}{2}], \zab\{\frac{1}{2}\}` ✓

Normal Version: $\left(\frac{1}{2}\right), \left[\frac{1}{2}\right], \left\{\frac{1}{2}\right\}$

Alias Version: $\left(\frac{1}{2}\right), \left[\frac{1}{2}\right], \left\{\frac{1}{2}\right\}$

7.3 slide

此 library 用于从文档模式切换到 slide 模式，无需用户对文档源码进行任何的更改，仅需在导言区加载此 library 即可。z \LaTeX 会自动处理文档的分页。

由于此 library 内部 patch 了很多的 \LaTeX 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。

zslide 中的坐标系统：如果不加说明，那么 zslide 中的坐标系统均以当前页面的左上角为原点，取向上向右为正方向。这就意味着你的纵坐标往往为负值，横坐标往往为正值。

WARNING: 此 library Patch 了大量的原始命令, 在 slide 模式下使用 `\zlatexThmToc` 或 `\zlatexPartialToc` 可能会导致一些奇怪的报错, 比如: ! LaTeX Error: Something's wrong--perhaps a missing \item.; 目前定位到 `\titlecontents{subsection}`, 如果后续自己实现了 `titletoc` 和 `titlesec` 两个宏包, 那么这个问题就比较方便解决了

`\zslideSetup`

Updated: 2024-11-05

`\zslideSetup[⟨key⟩]{⟨spec⟩}`

在加载 slide 库后, 此命令用于调整 $\text{z}\text{\LaTeX}$ 关于 slide 的默认配置. $\langle key \rangle$ 表示 $\text{z}\text{\LaTeX}$ 中 $\langle key-value \rangle$ 的层级名称, 默认为空, 此时即为根目录. $\langle key \rangle$ 的所有合法选项有: doc, sec, UL, toc, 或者直接置为空表示根目录. $\langle spec \rangle$ 表示对具体项目的设置, 在根目录下, 其所有可用的选项为:

`\zslideSetup{`
例 61

```
doc = {⟨bg-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 =  $\langle dim \rangle$ ,  
    section =  $\langle dim \rangle$ ,  
    subsection =  $\langle dim \rangle$   
  },  
  label = {  
    % use command \zslideToclabelSet to set  
    chapter =  $\langle item \rangle$ ,  
    section =  $\langle item \rangle$ ,  
    subsection =  $\langle item \rangle$   
  },  
  suffix = {  
    chapter =  $\langle item \rangle$ ,
```



```
    section = <item>,
    subsection = <item>
  },
}
```

在特定的子目录, 如 $\langle key \rangle = \text{doc}$ 或 $\langle key \rangle = \text{toc/leftmargin}$ 时, 一个设置样例如下:

```
\zslideSetup[doc]{
  bg-color=yellow!20,
  text-color=red
}
\zslideSetup[toc/leftmargin]{
```

例 62

```
chapter=1em,
section=4em,
}
```

```
zslide-title-page \pageref{zslide-title-page}
```

```
zslide-last-page \pageref{zslide-last-page}
```

New: 2024-12-15

引用当前文档的最后一页, 用于 slide 制作时的页码引用. 使用样例如下:

```
\pageref{zslide-title-page}--\pageref{zslide-  
-last-page}
```

例 63

`zslide@titlepage`
`zslide@lastpage`

Updated: 2024-11-05

`\hyper@link{<context>}{zslide@titlepage}{<link
text>}`
`\hyper@link{<context>}{zslide@lastpage}{<link
text>}`


上述两 Targets 由命令 `\hyper@anchor` 设置, 分别应用于引用当前文档的第一页和最后一页, 在 `zslide` 中, 标题页的页码为 0.

注意: 普通用户不应该直接使用这两个 Targets, 此二 Targets 主要提供给模板的开发者, 用户应使用位于首页和尾页的 `zslide-title-page` 和 `zslide-last-page` 两 label.

`zslide@color@title`

New: 2025-02-25

`\color{zslide@color@title}<item>`
`\textcolor{zslide@color@title}{<item>}`

此颜色用于设置 slide 模式下 title 的颜色, 默认为 , 即: HTML:d9d9d9.

`\zslideFrameCntAux:nn`

Updated: 2024-11-05

`\zslideFrameCntAux:nn {<name>}{<number>}`

此命令会向文件 `\jobname.aux` 中写入一个变量, 其名称为: `\zsec@<name>@cnt`, 其值为: `<number>`; `<name>` 一般为一大写罗马数字, 如 I, II, III, IV 等. 此命令在制作进度条或向后搜集文档内容时是十分有用的.

`_zslide_status_bar:nnnn`

Updated: 2024-11-05

`_zslide_status_bar:nnnn`
`{\langle type \rangle}{\langle coordinate \rangle}{\langle width \rangle}{\langle height \rangle}`

此命令用于创建 slide 的页面背景色块, 为方便叙述, 我们称其为 $\langle BOX \rangle$. 其中 $\langle coordinate \rangle$ 表示 $\langle BOX \rangle$ 左下角坐标, 形如 (10pt, -.1\paperwidth), 以当前页面的左上角为原点, 取向上向右为正方向; $\langle type \rangle$ 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 \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: 此命令需放入 shipout/background 或 shipout/foreground 这两个 Hook 中; 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.

`_zslide_status_info:nnnn`

Updated: 2024-11-05

`_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; `\g_zslide_status_info_head_B_dim` 和 `\g_zslide_status_info_foot_B_dim` 两个寄存器存放了 head 和 foot 中文字基线的纵坐标。 $\langle coordinate \rangle$ 表示 $\langle BOX \rangle$ 的左下角坐标，接受一个浮点数，以 `\paperwidth` 为单位。此参数以当前页面的左上角为原点，取向上向右为正方向； $\langle width \rangle$ 为当前 $\langle BOX \rangle$ 的 (弹性) 宽度，接受一个浮点数，以 `\paperwidth` 为单位。 $\langle content \rangle$ 表示 $\langle BOX \rangle$ 中存放的文本或图片内容。

`_zslide_navigate:nnnn`

Updated: 2024-11-05

`_zslide_navigate:nnnn {<range>}{<target>}{<target
symbol>}{<other symbol>}`

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: 此命令需放入 `shipout/background` 或 `shipout/foreground` 这两个 Hook 中; 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.

<code>\g_zslide_status_info_head_B_dim</code>	<code>\g_zslide_status_info_head_B_dim</code>
<code>\g_zslide_status_info_foot_B_dim</code>	<code>:-.9em+2.5pt</code>
<code>\g_zslide_status_info_sec_L_dim</code>	<code>\g_zslide_status_info_foot_B_dim</code>
<code>\g_zslide_status_info_sec_B_dim</code>	<code>:-\paperheight+2.5pt</code>
	<code>\g_zslide_status_info_sec_L_dim... :1cm</code>
	<code>\g_zslide_status_info_sec_B_dim</code>
	<code>:-2.25em</code>

New: 2025-01-14

前两个寄存器存放了 head 和 foot 中文字基线的纵坐标, 前者的默认值为 $-.9em+2.5pt$, 后者的默认值为 $-\text{paperheight}+2.5pt$. `\g_zslide_status_info_sec_L_dim` 中存放了 section 文本距离页面左边界的距离, 默认值为 $1cm$; `\g_zslide_status_info_sec_B_dim` 中存放了 section 文本基线的纵坐标, 默认值为 $-2.25em$.

注意: 普通用户不应该直接修改此系列寄存器

<code>\g_zslide_status_bar_head_H_dim</code>	<code>\g_zslide_status_bar_head_H_dim... :.9em</code>
<code>\g_zslide_status_bar_foot_H_dim</code>	<code>\g_zslide_status_bar_foot_H_dim... :.9em</code>
<code>\g_zslide_status_bar_sec_H_dim</code>	<code>\g_zslide_status_bar_sec_H_dim... :1.8em</code>
<code>\g_zslide_status_bar_sec_B_dim</code>	<code>\g_zslide_status_bar_sec_B_dim.. :-2.7em</code>

New: 2025-01-14

前两个寄存器存放了 slide 中 head 和 foot 对应背景色块的高度, 默认值均为 .9em; `\g_zslide_status_bar_sec_H_dim` 中存放了 section 的背景色块的高度, 默认值为 1.8em; `\g_zslide_status_bar_sec_B_dim` 中存放了 section 的背景色块中文字的基线的纵坐标, 默认值为 -2.7em; 当改变此三个寄存器的值时, 对应色块的基线保持不变, 其高度会做出相应的改变.

注意: 普通用户不应该直接修改此系列寄存器

<code>\zslideTitle</code>	<code>\zslideTitle</code>
<code>\zslideAuthor</code>	<code>\zslideAuthor</code>
<code>\zslideDate</code>	<code>\zslideDate</code>

Updated: 2024-11-05

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

<code>_zslide_metadate:n</code>	<code>_zslide_metadate:n {⟨key⟩}</code>
----------------------------------	--

Updated: 2024-11-05

此命令可以根据 `⟨key⟩` 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.

`\zslideFrameTotal`

Updated: 2024-11-05

`\zslideFrameTotal{⟨name⟩}`

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 `\jobname.aux` 中变量 `\zsec@⟨name⟩@cnt` 的值. 默认返回当前 section 下的 Frame 总数, 第一次编译亦或者是变量 `\zsec@⟨name⟩@cnt` 不存在, 命令 `\zslideFrameTotal` 将会返回 ??.

`\zslideFrameIndex`

Updated: 2024-11-05

`\zslideFrameIndex`

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.

`\zslideNavigateBall`

Updated: 2024-11-05

`\zslideNavigateBall[⟨target symbol⟩][⟨other symbol⟩]`

此命令为上述 `_zslide_navigate:nnnn` 的一个具体应用. `⟨target symbol⟩` 默认为 `•`, `⟨other symbol⟩` 默认为 `○`. 二者的意义请参见命令 `_zslide_navigate:nnnn` 的说明.

`\zslideLogo`

Updated: 2024-11-05

`\zslidelogo[⟨key-value⟩]{⟨picture⟩}`

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

```
\zslideLogo[
```

例 64

```
position = ⟨(dim1,
```

✓

```
dim2):(\paperwidth-\_zlatex_quad_dim,
```

✓

```
1.5em)⟩,
```

```
width = ⟨dim:2.5em⟩,
```

```
exclude = ⟨clist:1⟩
```

```
] {⟨picture⟩}
```

⟨exclude⟩ 中的页码不会添加 logo, 其余页码会添加 logo.

`\zslideTocPage``\thecontentslabel``\zslideToclabelSet``\zslideTocSecIcon``\zslideTocSubsecIcon`

Updated: 2024-11-05

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

这一组命令主要用于自定义 slide 中的目录, 其中

`\zslideTocPage` 表示目录项目对应的页码, `\thecontentslabel`

表示目录项目的对应的名称. `\zslideTocSecIcon`

和 `\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:

例 65

`\ExplSyntaxOn`

```
Section Icon: \box_move_up:nn {2pt}
  {\hbox:n {\ztool_set_to_wd:nn
    {6pt}}{\(\blacktriangleright\)}}
}\par
```

```
Subsection Icon: \rule[2pt]{3pt}{3pt}
```

`\ExplSyntaxOff`

```
SectionIcon:▶
```

```
SubsectionIcon:■
```

`\zslideToclabelSet` 用于设置 slide 模式下目录中各条目的格式. $\langle extra\ width \rangle$ 表示 `\thecontentslabel`

右侧额外的间距. $\langle item \rangle$ 可以选择使用 `\thecontentslabel`, `\zslidesecIcon`, `\zslidesubsecIcon` 或其它用户自定义符号.

`\zslideDocTextColor`

Updated: 2024-11-05

`\zslideDocTextColor{\langle color \rangle}`

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

`\zslideDefaultUR`

`\zslideDefaultBR`

Updated: 2024-11-05

`\zslideDefaultUR`
`\zslideDefaultBR`

此二命令表示 `slide` 模式下, UR, BR 位置处默认的元信息.

`\zslideIfPageTF`

Updated: 2024-11-05

`\zslideIfPageTF{⟨formula⟩}{⟨true code⟩}{⟨false
code⟩}`

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 $\langle formula \rangle$ 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}}{\zslideDefa
ultUR:\
  \zslideNavigateBall}},
]{AnnArborSpruce}
```

例 66

✓
✓

`\zslideThemeCreate`

Updated: 2024-11-05

`\zslideThemeCreate{⟨name⟩}{⟨spec⟩}`

此命令会按照 $\langle spec \rangle$ 创建名为 $\langle name \rangle$ 的 slide 颜色主题, 仅可在导言区使用. 具体的可调整选项请参见命令 `\zslideSetup` 中的 $\langle spec \rangle$ 参数说明.

`\zslideThemeUse``\zslideThemeUse[⟨key-value⟩]{⟨name⟩}`

Updated: 2024-11-05

此命令仅能在导言区使用, 其会根据 $\langle spec \rangle$ 对颜色主题 $\langle name \rangle$ 中的部分配置进行重写, 然后再应用 $\langle name \rangle$ 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 $\langle theme \rangle$ 参数指定 slide 的主题, 且 $\text{z}\text{\LaTeX}$ 仅加载一个主题, 命令 `\zslideThemeUse` 更大程度上是出于方便用户修改预定义主题中的某一特定项目这一目的而提供的.

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, 还可以使用命令 `\zslideThemeCreate` 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 $\langle spec \rangle$ 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 $\langle spec \rangle$ 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[\langle spec \rangle]{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 $\langle spec \rangle$ 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.

如用户在自定义定理类环境样式时需要更改 \LaTeX 的默认配色, 请将 `\zlatex_keys_set:nn` 或其它基于 `\keys_set:nn` 的命令放置于命令 `\zlatexThmStyleNew` 对应样式 `\preamble` 中而非 `\option` 中, 否则 \LaTeX 中的一系列与 `\zlatexColorSetup` 相关的函

数将失去对新定义数学类环境样式的色彩控制能力.

shadow

`\zlatexThmStyle{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}
```

例 67

```
\begin{remark}[zlatexThmStyle-5]
```

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}} \bigotimes \mathbf{w}}
```



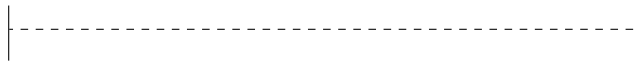
```

& = \sum_{i=1}^3\left(a_{i1}u^{iv^1}+a_{i2}
}u^{iv^2}+a_{i3}u^{iv^3}\right)
\\
& = \int x \, dx = \frac{1}{2} x^2 + \mathcal{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 (zlatexThmStyle-5) 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;

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

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

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

Updated: 2024-12-05

`\zlatexThmStyle{paris}`

加载此 library 后即可应用上述样式, 样式预览如下:

```
% \zlatexloadlibrary{mathalias}
```

例 68

```
\begin{axiom}[zlatexThmStyle-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;

```
\begin{align}
```

```
\underset{}{\mathbf{v}} \bigotimes \mathbf{w}}
```

```

& = \sum_{i=1}^3\left(a_{i1}u^{iv^1}+a_{i2}
}u^{iv^2}+a_{i3}u^{iv^3}\right)
\\
& = \int x \, dx = \frac{1}{2} x^2 + \mathcal{R}\{C\}
\end{align}

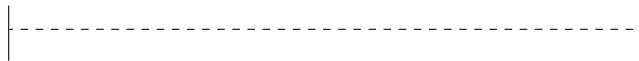
```

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 (zlatexThmStyle-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;

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

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

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

Updated: 2024-12-05

`\zlatexThmStyle{obsidian}`

加载此 library 后即可应用上述样式, 样式预览如下:

```
% \zlatexloadlibrary{mathalias}
```

例 69

```
\begin{proposition}[zlatexThmStyle-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}} \bigotimes \mathbf{w}}
```



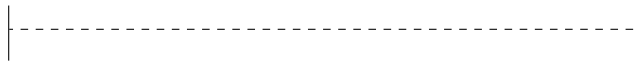
```

& = \sum_{i=1}^3\left(a_{i1}u^{iv^1}+a_{i2}
}u^{iv^2}+a_{i3}u^{iv^3}\right)
\\
& = \int x \, dx = \frac{1}{2} x^2 + \mathcal{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{proposition}



“命题:7.1 zlatexThmStyle-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;

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

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

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

Updated: 2024-12-05

`\zlatexThmStyle{lapsis}`

加载此 library 后即可应用上述样式, 样式预览如下:

```
% \zlatexloadlibrary{mathalias}
```

例 70

```
\begin{lemma}[zlatexThmStyle-8]
```

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}} \bigotimes \mathbf{w}}
```

```
& = \sum_{i=1}^3\left(a_{i1}u^{iv^1}+a_{i2}\right. ✓
      }u^{iv^2}+a_{i3}u^{iv^3}\right) ✓
```

```
\\
```

```
& = \int x \, dx = \frac{1}{2} x^2 + \mathcal{R}\{C\}
```

```
\end{align}
```

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

```
\tcblower
```

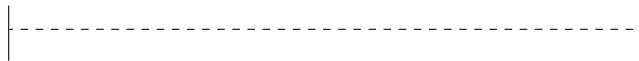
```
\begin{align}
```

```
\int x \, dx = \frac{1}{2} x^2 + \mathcal{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 representation of, as far as I know, the things in themselves;

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

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

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

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

elegant

Updated: 2024-12-05

`\zlatexThmStyle{elegant}`

加载此 library 后即可应用上述样式, 样式预览如下:

```
% \zlatexloadlibrary{mathalias}
```

例 71

```
\begin{definition}[zlatexThmStyle-9]
```

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}} \bigotimes \mathbf{w}}
```



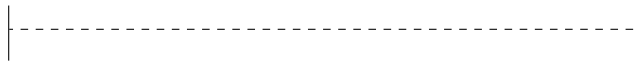
```

& = \sum_{i=1}^3\left(a_{i1}u^{iv^1}+a_{i2}
}u^{iv^2}+a_{i3}u^{iv^3}\right)
\\
& = \int x \, dx = \frac{1}{2} x^2 + \mathcal{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{definition}`



定义 7.1 (zlatexThmStyle-9)

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;

$$\mathbf{v} \otimes \mathbf{w} = \sum_{i=1}^3 (a_{i1}u^i v^1 + a_{i2}u^i v^2 + a_{i3}u^i v^3) \quad (7.10)$$

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

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;

8.1 l3sys-shell

本部分主要介绍 ztool 中实现的原始 l3sys-shell 宏包中的命令。所以使用本部分的命令时需编译 L^AT_EX 文档时启用 `-shell-escape` 参数, 否则此系列命令将不会执行任何操作。

WARNING: 请谨慎使用此部分的命令, 部分不当操作可能导致系统崩溃。

```
\ztool_shell_escape:n \ztool_shell_escape:n {\command}
```

```
\ztool_shell_escape:e
```

Updated: 2024-12-05

当 `-shell-escape` 参数启用时, 此命令会在 shell 中执行 `\command`, 如果 `-shell-escape` 参数未启用, 此命令将不会执行任何操作.

```
\ztool_shell_mkdir:n \ztool_shell_mkdir:n {\dir}
```

```
\ztool_shell_mkdir:e
```

Updated: 2024-12-05

当 `-shell-escape` 参数启用时, 此命令会创建一个目录 `\dir`, 如果 `-shell-escape` 参数未启用, 此命令将不会执行任何操作.

```
\ztool_shell_cp:nn
```

```
\ztool_shell_cp:(ee|ne|en)
```

Updated: 2024-12-05

```
\ztool_shell_cp:nn {<source>}{<target>}
```

当 `-shell-escape` 参数启用时, 此命令将把文件 `<source>` 复制为文件 `<target>`, 如果 `-shell-escape` 参数未启用, 此命令将不会执行任何操作.

```
\ztool_shell_mv:nn
```

```
\ztool_shell_mv:(ee|ne|en)
```

Updated: 2024-12-05

```
\ztool_shell_mv:nn {<source>}{<target>}
```

当 `-shell-escape` 参数启用时, 此命令将把文件 `<source>` 移动到目录 `<target>`, 如果 `-shell-escape` 参数未启用, 此命令将不会执行任何操作.

```
\ztool_shell_rm:n \ztool_shell_rm:n {\file}
```

```
\ztool_shell_rm:e
```

Updated: 2024-12-05

当 `-shell-escape` 参数启用时, 此命令将删除文件 `\file`, 如果 `-shell-escape` 参数未启用, 此命令将不会执行任何操作.

```
\ztool_shell_rmdir:n \ztool_shell_rmdir:n {\dir}
```

```
\ztool_shell_rmdir:e
```

Updated: 2024-12-05

当 `-shell-escape` 参数启用时, 此命令将删除目录 `\dir`, 如果 `-shell-escape` 参数未启用, 此命令将不会执行任何操作.

```
\ztool_get_shell_pwd:N
```

```
\ztool_get_shell_pwd:c
```

Updated: 2024-12-05

```
\ztool_get_shell_pwd:N <tl>
```

当 `-shell-escape` 参数启用时, 此命令将返回当前的工作目录, 并将其存放在 $\langle tl \rangle$ 中, 如果 `-shell-escape` 参数未启用, 此命令将不会执行任何操作.

```
\ztool_shell_split_ls:nN
```

Updated: 2024-12-05

```
\ztool_shell_split_ls:nN {<dir>}{<tl>}
```

当 `-shell-escape` 参数启用时, 此命令将返回目录 $\langle dir \rangle$ 下的所有文件名, 并将其存放在 $\langle tl \rangle$ 中, 如果 `-shell-escape` 参数未启用, 此命令将不会执行任何操作.

8.2 文件 IO 操作

本部分主要介绍 `ztool` 中实现的文件 IO 操作, 包括: 读取文件, 写入文件, 追加文件等操作. 本部分的系列命令均不需要启用 `-shell-escape` 参数.

`\ztool_file_new:nn`

Updated: 2024-12-05

`\ztool_file_new:nn {<boolean>}{<file>}`

此命令用于创建一个名为 `<file>` 的新文件, 如果 `<file>` 不存在, 则会创建一个名为 `<file>` 的新文件. 若文件已存在, 那么当 `<boolean>` 为 `\c_true_bool` 时, **会覆盖原文件**, 否则不会进行任何操作.

```
\ztool_read_file_as_seq:nnN          \ztool_read_file_as_seq:nnN
\ztool_read_file_as_seq:(neN|nnc|nec)  {\langle bool \rangle}{\langle file \rangle}{\langle seq \rangle}
```

Updated: 2024-12-05

此命令用于读取文件 $\langle file \rangle$ 的内容, 并将其存放在 $\langle seq \rangle$ 中, 如果 $\langle file \rangle$ 不存在, 则 $\langle seq \rangle$ 会被置为空. $\langle bool \rangle$ 用于控制是否保留行尾的空格, 可选值有: `\c_true_bool`, `\c_false_bool`, 默认为 `\c_true_bool`; 如果 $\langle bool \rangle$ 为 `\c_true_bool`, 则保留行尾的空格, 否则不保留.

此命令仅在一个组内有效, 且不可嵌套.

```

\ztool_gread_file_as_seq:nnN          \ztool_read_file_as_seq:nnN
\ztool_gread_file_as_seq:(neN|nnc|nec)  {\langle bool \rangle}{\langle file \rangle}{\langle seq \rangle}

```

Updated: 2025-01-05

此命令用于读取文件 $\langle file \rangle$ 的内容, 并将其存放在 $\langle seq \rangle$ 中, 如果 $\langle file \rangle$ 不存在, 则 $\langle seq \rangle$ 会被置为空. $\langle bool \rangle$ 用于控制是否保留行尾的空格, 可选值有: `\c_true_bool`, `\c_false_bool`, 默认为 `\c_true_bool`; 如果 $\langle bool \rangle$ 为 `\c_true_bool`, 则保留行尾的空格, 否则不保留.

此命令作用于全局, 不可嵌套.

<code>\ztool_append_to_file:nn</code>	<code>\ztool_append_to_file:nn</code>
<code>\ztool_append_to_file:(no nf ee)</code>	<code>{\langle file \rangle}{\langle content \rangle}</code>

Updated: 2025-01-05

此命令用于将 $\langle content \rangle$ 追加到文件 $\langle file \rangle$ 中, 如果 $\langle file \rangle$ 不存在, 则会创建一个名为 $\langle file \rangle$ 的新文件, 并将 $\langle content \rangle$ 写入其中.

```
\ztool_replace_file_line:nnn
```

```
\ztool_replace_file_line:(enn|ene|eee)
```

```
\ztool_replace_file_line:nnn
```

```
{\langle file \rangle}{\langle line \rangle}{\langle content \rangle}
```

Updated: 2025-01-05

此命令用于将文件 $\langle file \rangle$ 中的第 $\langle line \rangle$ 行替换为 $\langle content \rangle$, 如果 $\langle file \rangle$ 不存在, 则不会进行任何操作.

```
\ztool_insert_to_file:nnn          \ztool_insert_to_file:nnn
\ztool_insert_to_file:(nen|nfn|een)  {\<file>}{\<line>}{\<content>}
```

Updated: 2025-01-05

此命令用于将 $\langle content \rangle$ 插入到文件 $\langle file \rangle$ 的第 $\langle line \rangle$ 行之前, 如果 $\langle file \rangle$ 不存在, 则不会进行任何操作.

下面一个示例展示了如何使用 `ztool` 中的几个文件 IO 操作命令:

```
\ExplSyntaxOn
```

```
\ztool_file_new:nn {\c_true_bool}{testIO.txt}
```

例 72

```
\seq_new:N \l_ztool_tmp_seq \seq_clear:N ✓  
\l_ztool_tmp_seq  
\ztool_append_to_file:nn {testIO.txt} ✓  
{|APPEND-CONTENT|}  
\ztool_insert_to_file:nnn {testIO.txt} {2} ✓  
{|~~INSERT-CONTENT~~|}  
\ztool_gread_file_as_seq:nnN {\c_false_bool} ✓  
{testIO.txt} \l_ztool_tmp_seq  
\seq_use:Nn \l_ztool_tmp_seq {\par}  
\ExplSyntaxOff  
\inputminted{text}{testIO.txt}
```

|INSERT-CONTENT|
|APPEND-CONTENT|

| INSERT-CONTENT |
| APPEND-CONTENT |

8.3 Box 操作

本部分介绍 ztool 中实现的 Box 操作, 包括 box 的测量以及 box 的简单变换.

 $\backslash\text{ztool_get_ht:Nn}$
 $\backslash\text{ztool_get_ht: (Ne|ce)}$

Updated: 2024-12-05

 $\backslash\text{ztool_get_ht:Nn } \langle dim \rangle \{ \langle content \rangle \}$

此命令用于将 $\langle content \rangle$ 的高度保存在 $\langle dim \rangle$ 这一寄存器中.

 $\backslash\text{ztool_get_wd:Nn}$
 $\backslash\text{ztool_get_wd: (Ne|ce)}$

Updated: 2024-12-05

 $\backslash\text{ztool_get_wd:Nn } \langle dim \rangle \{ \langle content \rangle \}$

此命令用于将 $\langle content \rangle$ 的宽度保存在 $\langle dim \rangle$ 这一寄存器中.

`\ztool_get_dp:Nn`
`\ztool_get_dp:(Ne|ce)`

Updated: 2024-12-05

`\ztool_get_dp:Nn <dim>{\<content>}`

此命令用于将 $\langle content \rangle$ 的深度保存在 $\langle dim \rangle$ 这一寄存器中.

`\ztool_gget_ht:Nn`
`\ztool_gget_ht:(Ne|ce)`

Updated: 2024-12-05

`\ztool_gget_ht:Nn <dim>{\<content>}`

此命令用于将 $\langle content \rangle$ 的高度保存在 $\langle dim \rangle$ 这一寄存器中, 并且此操作是全局的.

`\ztool_gget_wd:Nn`
`\ztool_gget_wd:(Ne|ce)`

Updated: 2024-12-05

`\ztool_gget_wd:Nn <dim>{\<content>}`

此命令用于将 $\langle content \rangle$ 的宽度保存在 $\langle dim \rangle$ 这一寄存器中, 并且此操作是全局的.

`\ztool_gget_dp:Nn`
`\ztool_gget_dp:(Ne|ce)`

Updated: 2024-12-05

`\ztool_gget_dp:nn <dim>{<content>}`

此命令用于将 $\langle content \rangle$ 的深度保存在 $\langle dim \rangle$ 这一寄存器中, 并且此操作是全局的.

`\ztool_set_to_wd:nn`
`\ztool_set_to_wd:(en|ne)`

Updated: 2024-12-05

`\ztool_set_to_wd:nn {<dim>}{<content>}`

此命令用于将 $\langle content \rangle$ 的宽度调整为 $\langle dim \rangle$.

`\ztool_set_to_ht:nn`
`\ztool_set_to_ht:(en|ne)`

Updated: 2024-12-05

`\ztool_set_to_ht:nn {<dim>}{<content>}`

此命令用于将 $\langle content \rangle$ 的高度调整为 $\langle dim \rangle$.

下面一个示例展示了如何使用 `ztool` 中的几个 Box 操作命令:

`\ExplSyntaxOn`

例 73

`% get dim of content`

`\dim_new:N \l_ztool_tmp_H_dim`

`\dim_new:N \l_ztool_tmp_W_dim`

`\ztool_get_ht:Nn \l_ztool_tmp_H_dim`



`{Hello,~world!}`

`\ztool_get_wd:Nn \l_ztool_tmp_W_dim`



`{Hello,~world!}`

`\dim_use:N \l_ztool_tmp_H_dim \quad`



`\dim_use:N \l_ztool_tmp_W_dim\\`

`% set content to dim`

```
Hello,~world!|
```

```
\ztool_set_to_ht:nn {.5cm} {Hello,~world!}|
```

```
\ztool_set_to_wd:nn {25pt} {Hello,~world!}
```

```
\ExplSyntaxOff
```

```
7.31432pt 55.70055pt
```

```
Hello, world!|Hello, world!|Hello, world!
```

TODO

z^ΛT_EX 的开发还远远没有结束，还有很多功能需要完善，这里列出部分将来可能会完善的功能：

1. 封装 geometry 宏包的相关接口，使得用户可以通过 z^ΛT_EX 的接口来设置页面布局.
2. 在独立实现 titlesec 和 titletoc 之前，先暂时把这两个宏包的接口封装一下，放入 z^ΛT_EX 中.
3. 使用 new marker mechanism 来实现 fancyhdr 的相关功能.
4. 自定义 syntax 环境，用于排版代码. (比如给出相关命令的 $\langle key \rangle$ 或 $\langle key \rangle$ 的默认值).

5. 把自己修改的那个 Euler Math 变体配置进 $\text{z}\text{\LaTeX}$, 命名为 `vareuler`, 然后把相关配置写入 `fontcfg` module.
6. 给 `\zlatexPageMask` 命令增加一个 `\langle transparent \rangle` key 以适配不同的对象 (文本, 图片) 以及引擎.
7. 2025-02-04-已完成: 添加一个证明类环境的 `\zlatexThmProofTitileFormat` 接口, 用于设置证明类环境的标题格式.
8. 完善 Metropolis `zslide` 主题, 实现 `zslide` 中的 `\zslideThemeUse` 和 `\zslideColorUse` 接口, 包括二者的自由组合.

9. 添加一个真正的 `\zslideFrameTotal` 命令, 并把现在的 `\zslideFrameTotal` 命令重命名为 `\zslideFrameSecTotal`.
10. 完善 `thm module` 的 `icon` 接口 (类似 `Elegant \LaTeX` 系列), 但此接口仅在用户加载 `theme library` 时才可用.
11. 完善 `thm module` 中 `paris` 主题的分页样式.
12. 使用 `ztool` 缩放 `thm module` 中 `obsidian` 样式标题中的 `icon`.
13. 重新实现部分的 `xcoffins` 宏包中的命令, 目标

为: 实现 `\parbox` 的功能, 并且比之更加的易用.

14. 封装 PlainTeX 中的 `\parshape` 及其相关命令, 使之更加的易用.
15. 封装 `\lastbox` 相关命令, 实现段落的分割和盒子的跨页需求.
16. 在实现跨页盒子的基础上, 手动实现 `framed` 宏包的功能, 在替代该宏包原有功能的基础上, 提供更加易用的接口.
17. 增加一个基于任意变换矩阵的盒子 (内容) 操作命令, 也许是依赖 `l3draw` ?? 或许增加一

个 `\ztool_set_to_wd_ht:nnn` 或 `\ztool_set_wd_ht_plus_dp:nnnn` 命令 ???

18. 提供列表设置的相关命令, 目标是成为宏包 `enumerate` 的一个可选替代. (直接从原始的 `list` 环境出发 ?? 未来会把这部分命令抽离到一个新的单独模块)
19. 在 `pageinfo` 中实现一个增强的 `\marginpar` 命令, 目的是成为 `sidenotes` 宏包的一个可选替代.
20. 实现 `\hyper@icon` 接口, 用于设置文档中的超链接图标. (没有 `icon` 的超链接未免过于单调)

21. 2025-02-05-已完成: 优化 module 和 library 的加载检测机制, 完善相关变量的检测设置, 如在 mathalias 这一 library 中将变量 `\g_zlatex_math_alias_bool` 显示的设置设置为 true.
22. 创建 `\MathAliasOn`, `\MathAliasOff` 两命令用于限制 mathalias library 中命令的使用范围.
23. 在部分 z_{La}T_EX 内置命令的实现中增加 `_zlatex_plus_key_aux:nnn` 命令, 用于在保留原内容的基础上增加内容.
24. 修复 `\zlatexThmTocAdd` 增加的定理条目超链接跳转异常这一问题.

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	A
<code>-shell-escape</code> <i>8, 196–201</i>	<code>\A</code> <i>139</i>
<code>??</code> <i>163</i>	<code>\alt</code> <i>144</i>
<code>_zlatex_quad_dim</code> <i>165</i>	<code>AnnArborAlbatross</code> <i>173</i>

AnnArborBeaver	172	bib commands:	
AnnArborDefault	172	bib_index	32
AnnArborSeahorse	173	BL	157
AnnArborSpruce	174	\blacktriangleright	51
B		\boldsymbol	126
\B	126	bool commands:	
\backsimeq	140	\c_false_bool	202, 203
basic packages	MMMMI-16	\c_true_bool	201–203
BC	157	BR	157
\bfseries	84		

C

<code>\C</code>	126	<code>\cong</code>	140
<code>\CC</code>	142	<code>\counterwithin</code>	39
<code>\chapter</code>	121	<code>\ctexset</code>	26
<code>\Cinzel</code>	51	<code>\curl</code>	144
<code>class</code>	26		
<code>classOption</code>	27		
<code>\cok</code>	144		
<code>\color</code>	156		
<code>\colorlet</code>	70		

D

<code>\Da</code>	133
<code>\da</code>	133
<code>\dd</code>	141
<code>\Dda</code>	133
<code>\dda</code>	133

`\DeclareMathOperator` 144

`\definecolor` 70

`\div` 144

`doc, \sec, \UL, \toc` 151

`\documentclass` 172–174

E

`\E` 139

`elegant` 192

F

`\F` 126

`fancy` 23, 25

`\FF` 126

`font` 30

`foot` 158

`\frontmatter` 12, 62

G

`\grad` 144

`\graphicspath` 38

H

<code>head</code>	158
<code>\hla</code>	137
<code>\hla*</code>	137
<code>\hom</code>	144
<code>\hra</code>	137
<code>\hra*</code>	137
<code>hyper</code>	24
<code>\hypersetup</code>	57

I

<code>\id</code>	144
<code>\im</code>	144
<code>\includegraphics</code>	63

K

<code>\K</code>	126
<code>\ker</code>	144
keys commands:	
<code>\keys_define:nn</code>	71
<code>\keys_set:nn</code>	175

L	
<code>\La</code>	129
<code>\la</code>	129
<code>lang</code>	24
language packages	MMMMI-24
<code>lapis</code>	188
<code>\lastbox</code>	217
<code>layout</code>	31
<code>\Leftarrow</code>	129
<code>\leftarrow</code>	129
<code>\Leftrightarrow</code>	133
<code>\leftrightharpoon</code>	133
<code>\LinkTargetOff</code>	56
<code>\LinkTargetOn</code>	56
<code>\Lla</code>	129
<code>\lla</code>	129
<code>\Longleftarrow</code>	129
<code>\longleftarrow</code>	129
<code>\Longleftrightharpoon</code>	133
<code>\longleftrightharpoon</code>	133

<code>\longmapsto</code>	128	<code>\Maketitle</code>	61
<code>\Longrightarrow</code>	131	<code>\maketitle</code>	60, 61
<code>\longrightarrow</code>	131	<code>\mapsto</code>	128
M			
<code>\ma</code>	128	<code>\marginpar</code>	218
<code>\mainmatter</code>	12, 62	<code>mathalias</code>	23
<code>\makeatletter</code>	83	<code>\MathAliasOff</code>	219
<code>\makeatother</code>	83	<code>\MathAliasOn</code>	219
<code>\MakeLinkTarget</code>	55–57	<code>\mathbb</code>	126
<code>\MakeLinkTarget*</code>	55	<code>\mathbf</code>	126
		<code>\mathcal</code>	126

<code>\mathfrak</code>	126	<code>\Nla</code>	129
<code>\mathrm</code>	126	<code>\nla</code>	129
<code>\mathscr</code>	126	<code>\nLeftarrow</code>	129
<code>mathSpec</code>	33, 97	<code>\nleftarrow</code>	129
<code>\mma</code>	128	<code>\nLeftrightarrow</code>	133
N		<code>\nlefnrightarrow</code>	133
<code>\Nda</code>	133	<code>\NN</code>	142
<code>\nda</code>	133	<code>\noindent</code>	92, 94
<code>next-anchor</code>	57	<code>\Nra</code>	131
<code>\NextLinkTarget</code>	57	<code>\nra</code>	131

<code>\nrightarrow</code>	131	<code>\paperheight</code>	66
<code>\nrightrightarrow</code>	131	<code>\paperwidth</code>	66, 157, 158, 165
<code>\ns</code>	140	<code>\par</code>	92
O			
<code>obsidian</code>	184	<code>\parbox</code>	217
<code>\orimaketitle</code>	60	<code>paris</code>	180
		<code>\parshape</code>	217
P			
<code>packageOption</code>	28	Q	
<code>\pageref</code>	16, 42, 154	<code>\qedsymbol</code>	95

R

<code>\R</code>	126
<code>\Ra</code>	131
<code>\ra</code>	131
<code>\refstepcounter</code>	56, 57
<code>\Rightarrow</code>	131
<code>\rightarrow</code>	131
<code>\rot</code>	144
<code>\RR</code>	142
<code>\Rra</code>	131

<code>\rra</code>	131
-------------------------	-----

S

<code>\S</code>	126
<code>\se</code>	140
<code>sec</code>	157
<code>\SetLinkTargetFilter</code>	58
<code>\sffamily</code>	84
<code>shadow</code>	176
<code>shipout/background</code>	157–159
<code>shipout/foreground</code>	157–159

`\sign` 144
`slide` 23
`\sse` 140

T

T_EX and L^AT_EX 2_ε commands:

`\@addtoreset` 39
`\@author` 44, 162
`\@date` 44, 162
`\@title` 44, 162
`\hyper@anchor` 43, 52, 155

`\hyper@icon` 218
`\hyper@link` ... 43, 53, 55, 155
`\hyper@linkend` 53, 54
`\hyper@linkfile` 54
`\hyper@linkstart` 53, 54
`\thm@proof@temp@color` . 83, 90
`\thm@temp@color` 83
`zlatex@color@⟨name⟩` 69
`zlatex@lastpage` 43
`zlatex@titlepage` 43

<code>\zsec@<name>@cnt</code>	156, 163	<code>\thmnote</code>	82
<code>zslide@color@title</code>	156	<code>\thmnumber</code>	81
<code>zslide@lastpage</code>	155	<code>\titlecontents</code>	150
<code>zslide@titlepage</code>	155	<code>toc</code>	28
<code>\text</code>	135, 137	<code>\trace</code>	144
<code>\textcolor</code>	156		
<code>\thecontentslabel</code>	166–168		
		U	
<code>theme</code>	23, 98	<code>UL</code>	157
<code>thm-hook.<Hook Index></code>	104	<code>UR</code>	157
<code>\thmname</code>	82	<code>\usepackage</code>	8

V			
<code>\varnothing</code>	140	<code>\xLeftarrow</code>	135
<code>\verb</code>	37	<code>\xleftarrow</code>	135
X		<code>\xLongleftarrow</code>	135
		<code>\xLongrightarrow</code>	135
<code>\xhookleftarrow</code>	137	<code>\Xra</code>	135
<code>\xhookrightarrow</code>	137	<code>\xra</code>	135
<code>\Xla</code>	135	<code>\Xra*</code>	135
<code>\xla</code>	135	<code>\xra*</code>	135
<code>\Xla*</code>	135	<code>\xrightarrow</code>	135
<code>\xla*</code>	135	<code>\xrightarrow</code>	135

<code>\xxla</code>	135	zlatex commands:	
<code>\xxla*</code>	135	<code>\zlatex_keys_set:nn</code>	175
<code>\xxra</code>	135	<code>\c_zlatex_quad_dim</code>	41
<code>\xxra*</code>	135	zlatex internal commands:	
		<code>_zlatex_color_set:n</code>	71
		<code>\g_zlatex_math_alias_-</code>	
		<code>bool</code>	219
		<code>_zlatex_page_annotate:nnnnn</code>	
		62
		<code>_zlatex_plus_key_-</code>	
Z			
<code>\zab</code>	147		
<code>\zchapterLContent</code>	122		
<code>\zchapterRContent</code>	123		
<code>\zchapterSaying</code>	123		
<code>\zLaTeX</code>	4		

aux:nnn	219	68, 70, 79, 97, 175
_zlatex_thm_proof_-		\zlatexCounterWith	39
title:	92	\zlatexDate	44
_zlatex_thm_warp_-		\zlatexFancyTF	47
start:nnnn	92	\zlatexFontConfigTF	47
zlatex-last-page	42	\zlatexFramed	40
zlatex-title-page	42	\zlatexHyperTF	47
\zlatexAuthor	44	\zlatexloadlibrary ..	22, 98, 120
\zlatexBibIndexLoadTF	47	\zlatexloadmodule	22, 50
\zlatexColorSetup		\zlatexMarginTF	47

<code>\zlatexMathAliasOpSet</code> ..	144, 146	<code>\zlatexStopPartialToc</code>	119
<code>\zlatexMathAliasTF</code>	47	<code>\zlatexThemeLibLoadTF</code>	47
<code>\zlatexNumSuffix</code>	121	<code>\zlatexThmBefore</code>	92
<code>\zlatexOptions</code>	19	<code>\zlatexThmCnt</code>	95
<code>\zlatexPageMask</code> 60, 62, 63, 66, 215		<code>\zlatexThmColorSetup</code>	70, 97
<code>\zlatexPageMask*</code>	63	<code>\zlatexThmCreate</code>	79, 97
<code>\zlatexPageMaskRemove</code>	66	<code>\zlatexThmHook</code>	103, 107
<code>\zlatexPartialToc</code>	119, 150	<code>\zlatexThmHook*</code>	103, 104, 107
<code>\zlatexSetup</code>	14, 18, 19, 29	<code>\zlatexThmLang</code>	74, 77, 79
<code>\zlatexSlideTF</code>	47	<code>\zlatexThmName</code>	82, 84, 90

<code>\zlatexThmNameSet</code>	77	<code>\zlatexThmTitle</code>	84, 87, 90
<code>\zlatexThmNote</code>	82, 84, 87, 90	<code>\zlatexThmTitle*</code>	84
<code>\zlatexThmNoteEmptyTF</code>	89, 90	<code>\zlatexThmTitleBefore</code>	94
<code>\zlatexThmNumber</code>	81, 84	<code>\zlatexThmTitleFormat</code>	
<code>\zlatexThmProofHook</code>	106		81–84, 87, 97
<code>\zlatexThmProofHook*</code>	106	<code>\zlatexThmTitleFormat*</code>	87
<code>\zlatexThmProofTitleFormat</code>	215	<code>\zlatexThmTitleSwitch</code>	85
<code>\zlatexThmStyle</code>	86,	<code>\zlatexThmTitleSwitch*</code>	85
	98, 102, 176, 180, 184, 188, 192	<code>\zlatexThmToc</code>	109, 150
<code>\zlatexThmStyleNew</code>	101, 175	<code>\zlatexThmTocAdd</code>	114, 219

<code>\zlatexThmTocLevel</code>	115	zslide commands:	
<code>\zlatexThmTocPrefix</code> . . .	116, 117	<code>_zslide_metadate:n</code>	162
<code>\zlatexThmTocStop</code>	115	<code>_zslide_navigate:nnnn</code> . .	
<code>\zlatexThmTocSymbol</code> . . .	116, 117	159, 164
<code>\zlatexThmTocSymbolClear</code>	116, 117	<code>_zslide_status_bar:nnnn</code>	157
<code>\zlatexTitle</code>	44	<code>\g_zslide_status_bar_-</code>	
<code>\zlatexVerb</code>	37	<code>foot_H_dim</code>	161
<code>\zph</code>	65, 66	<code>\g_zslide_status_bar_-</code>	
<code>\zpw</code>	65, 66	<code>head_H_dim</code>	161
		<code>\g_zslide_status_bar_sec_-</code>	

B_dim	161	\g_zslide_status_info_-	
\g_zslide_status_bar_sec_-		sec_L_dim	160
H_dim	161	zslide-last-page	154
_zslide_status_info:nnnn	158	zslide-title-page	154
\g_zslide_status_info_-		\zslideAuthor	162
foot_B_dim	158, 160	\zslideColorUse	215
\g_zslide_status_info_-		\zslideDate	162
head_B_dim	158, 160	\zslideDefaultBR	168
\g_zslide_status_info_-		\zslideDefaultUR	168
sec_B_dim	160	\zslideDocTextColor	168

zslideFrameCntAux commands:	\zslideSetup 151, 170
\zslideFrameCntAux:nn . . . 156	\zslidesubsecIcon 168
\zslideFrameIndex 163	\zslideThemeCreate 170, 172
\zslideFrameSecTotal 216	\zslideThemeUse 171–174, 215
\zslideFrameTotal 163, 216	\zslideTitle 162
\zslideIfPageTF 169	\zslideToclabelSet 166, 167
\zslideLogo 164	\zslideTocPage 166
\zslidelogo 164	\zslideTocSecIcon 166
\zslideNavigateBall 164	\zslideTocSubsecIcon 166
\zslidesecIcon 168	\zsubtitle 122

ztool	8	\ztool_gget_ht:Nn	210
ztool commands:		\ztool_gget_wd:Nn	210
\ztool_append_to_file:nn	204	\ztool_gread_file_as_-	
\ztool_file_new:nn	201	seq:nnN	203
\ztool_get_dp:Nn	210	\ztool_insert_to_file:nnn	206
\ztool_get_ht:Nn	209	\ztool_read_file_as_-	
\ztool_get_shell_pwd:N ..	200	seq:nnN	202, 203
\ztool_get_wd:Nn	209	\ztool_replace_file_-	
\ztool_gget_dp:Nn	211	line:nnn	205
\ztool_gget_dp:nn	211	\ztool_set_to_ht:nn	211

<code>\ztool_set_to_wd:nn</code>	211	<code>\ztool_shell_mkdir:n</code>	197
<code>\ztool_set_to_wd_ht:nnn</code> .	218	<code>\ztool_shell_mv:nn</code>	198
<code>\ztool_set_wd_ht_plus_-</code>		<code>\ztool_shell_rm:n</code>	199
<code>dp:nnnn</code>	218	<code>\ztool_shell_rmdir:n</code>	199
<code>\ztool_shell_cp:nn</code>	198	<code>\ztool_shell_split_ls:nN</code>	200
<code>\ztool_shell_escape:n</code> . . .	197	<code>\ZZ</code>	34, 142