

字帖宏包

陈文建 longaster@163.com

农历二〇二一年一月十一日 v1.2

我欲乘風向北，飛雪落船。
軒轅東游，待我仙蹤。
白龍吟，奈我何。
望嶺，沐日光，滄海萬壑。
青山長，風聲萬壑歸來。
不見天涯人，不回。

我欲乘風向北，飛雪落船。
軒轅東游，待我仙蹤。
白龍吟，奈我何。
望嶺，沐日光，滄海萬壑。
青山長，風聲萬壑歸來。
不見天涯人，不回。

——《少年歌行》

注：本文档中的非商业的字体版权为原字体公司（北京方正有限公司及其它相关公司），此处使用该字体仅为显示该宏包的排版效果，不作商业用途。本文档 zitie.tex 和相应的 zitie.sty 源文件使用 [LPPL 1.3c 协议](#)。

zitie 宏包是用纯 L^AT_EX3 模块编写的(即,不加载 pgf、pstricks 等绘图宏包),也不使用 xcolor 宏包,来生成网格化的字、生成网格化的背景的宏包。绘图功能完全依赖 l3draw,但应注意 l3draw 目前仍然是实验性的,其接口在后续版本可能无法使用。

L^AT_EX 2_ε 版本应在 2020-10-01 及以后。

目前仅支持 X_YL^AT_EX,待后续版本支持 Lua_TE_X,不支持 pdf_TE_X。

目录

第 1 节 zitie 宏包的加载和基本命令	2	第 5 节 编程接口	9
第 2 节 选项	3	第 6 节 TODO	10
第 3 节 background	8	版本历史	11
第 4 节 zhlipsum, 中文乱数假文	9	代码索引	11

第 1 节 zitie 宏包的加载和基本命令

加载宏包:`\usepackage[⟨key-val⟩]{zitie}`。⟨key-val⟩ 选项见第 2 节。

本宏包将加载 xeCJK 宏包 [1]。

<code>\zitienewfont</code>	<code>\zitienewfont {⟨font family⟩ {⟨font name⟩ , ... }</code>
<code>\zitieCJKfamily</code>	<code>\zitienewfont {⟨font family⟩ {⟨font name⟩ {⟨font features⟩} , ... }</code>
	<code>\zitienewfont * {⟨font family⟩ {⟨font name⟩ {⟨font features⟩}</code>
	<code>\zitieCJKfamily {⟨font family⟩}</code>
	<code>\zitieCJKfamily+ {⟨font family⟩}</code>
	<code>\zitieCJKfamily- {⟨font family⟩}</code>

这个命令用来加载和设置需要的字体,⟨font family⟩ 为在 `\framezi` 等命令的 font 选项中使用名字,也可通过 `\zitieCJKfamily` 来在文档中使用该字体。⟨font name⟩ 为字体的名字,⟨features⟩, 为 fontspec 宏包 [2] 和 xeCJK 宏包支持的字体特性。

`\zitieCJKfamily` 和 xeCJK 宏包的 `\CJKfamily` 作用完全相同,只不过首先判断该 ⟨font family⟩ 是否被 `\zitienewfont` 声明,若没有,则再使用 `\CJKfamily` 进行处理。

<code>\framezi</code>	<code>\framezi * [⟨key-val⟩] {⟨字或词句⟩}</code>
<code>\framerange</code>	<code>\framerange [⟨key-val⟩] {⟨逗号分隔的16进制序列⟩}</code>
<code>\framezifile</code>	<code>\framezifile * [⟨key-val⟩] {⟨文件名⟩}</code>

以上命令用来生成网格化的字。

⟨key-val⟩ 为第 2 节定义的键。

⟨字或词句⟩ 应仅包含类代码为 11 或 12 的字符,即正常情况下输入的字符,不应包含 T_EX 控制序列即其它特殊字符。`\framezi` 带星号的版本接收一个控制序列,这个序列包含想要网格化的字。

⟨逗号分隔的 16 进制序列⟩ 应使用如下的形式: "4E00 -> "4E27, "4E30 或 一 -> 丢, 丰, 与其它设置字符序列的宏包相似(如 xeCJK 宏包的 `\xeCJKDeclareSubCJKBlock`)。注意,由于网格化操作需要大量的计算,因此单个 ⟨range⟩ 不应太大,依据编译环境而定,一般不应超过 2000 个,否则可能无法成功编译。其它命令中也是如此。

⟨文件名⟩ 为想要网格化的文字的文件名。`\framezifile` 带 * 的版本,可在 ⟨key-val⟩ 中使用 filepath 设置文件搜索路径,详见后节的说明。带星号和不带星号的版本实现略有差异,速度也可能不一样。

当连续多次使用 `\framezi` 等命令时,为去掉后面的空白,可使用 % 注释符。

```

zitie/framezi/before
zitie/framezi/after
zitie/framerange/before
zitie/framerange/after
zitie/framerange/range
zitie/framezifile/before
zitie/framezifile/after

```

这三个命令还各自定义了钩子(hook),其中带有 after 为 reversed,即先添加的后执行。before 为选项设置完成后执行,after 为整个命令执行完之后执行。其中 zitie/framerange/range 为每一个列表均执行一次,包括无效的和最后一次。

可以通过 \AddToHook、\AddToHookNext、\RemoveFromHook 来添加和移除,详细请参考 source2e.pdf[3] 或 lthooks[4] 的说明文档。

```

\begin{zitieframe} [{<key-val>}]
...
\end{zitieframe}
\begin{zitieframe} [{<initial material>}] [{<key-val>}]
...
\end{zitieframe}

```

可以在 zitieframe 环境中使用分段(显式的或隐式的,特殊命令仍然不可使用)。

<initial material> 为在选项和钩子执行完毕之后要执行的内容。这里的内容不会被网格化。

zitie/zitieframe/par 钩子为环境中的每个段落分段后要执行的命令。

第2节 选项

以下选项在方框化命令的宏中是局部设置的。

basechar	basechar = <CJK char>
fontsize	fontsize = <fontsize command>
zihao	zihao = <字号>

初始值 = 好
初始值 = \normalsize

basechar 设置当 resize 为 base 类时的基字符,这个用来计算缩放比例,基字符不同时,即使给定相同的缩放比例,其实际缩放比例也可能不同。可使用后文介绍的 \zitiebasechar 来引用当前的 basechar 值。

fontsize 设置计算基字符时的字大小,默认值也为 \normalsize。zihao 设置计算缩放比例时的字号大小,必须加载 ctexsize 或 ctex 宏包才可使用。

basechar 和 fontsize 应使用 \zitiesetup 或宏包选项全局地设置,在 \framezi 中使用该选项,则该修改在之后的 \framezi 中并不会生效,即设置是局部的。

当修改字体大小时,若要在 \framezi 等命令的外部使用其宽高寄存器 (\zitiebasecharwidth、\zitiebasecharheight),则应重新设置 basechar 或 fontsize。 \framezi 等命令将自动重新计算,但不会改变全局的值。

punctuation	punctuation = {ignore leave onlast scale ...}	初始值 = ignore
punctuation*	punctuation* = {{replace}}	

zitie 宏包对字符类型进行区分,对 CJK 字符和标点符号采用不同的处理方式。该选项设置标点符号采用何种处理方式。初始情况下为 `ignore`,即忽略该字符。

`leave` 选项把该标点符号原样输出,不对其进行任何处理,字体和其它属性和正文相同。

`onlast` 选项把标点符号放到 0 宽度的盒子中,并且忽略它的深度和高度,看起来就像是在最后一个字的方框中。

`scale` 选项把标点符号按最后一个字的缩放倍数进行缩放,再将其进行输出。

选项 `punctuation*` 定义处理标点符号的方式为其值,可接收一个参数。如设置 `punctuation*=\fbox{#1}`,则将标点符号放入 `\fbox` 内。

若 `punctuation` 并不存在,则将其值视为已定义的宏。如,假设 `@gobble` 值并不存在,若使用 `punctuation=@gobble`,则标点符号处理命令为 `\@gobble`,这是 $\text{\LaTeX} 2_{\epsilon}$ 定义的命令,其作用与 `punctuation=ignore` 相同。

还可使用 `\zitienuwrule` 定义新的标点符号处理方式。详见 `\zitienuwrule` 命令的说明。

实际上,`onlast` 和 `scale` 就是由 `\zitienuwrule` 定义的,这个键的处理器接收一个标点符号作为其参数。具体的定义见 `\zitienuwrule` 命令的说明。

T_Xhackers note: 按照 `xeCJK` 宏包的字符分类,`zitie` 宏包将字符类别为 `FullLeft`、`FullRight`、`HalfLeft`、`HalfRight` 视为标点符号。将 CJK 视为 CJK 字符。

于是可以通过修改标点符号的字符类别为 CJK 来使用 CJK 的处理规则,但是要注意,一般标点符号的宽高与常用字符的宽高并不一致。某些字符也不是全高的,这时,可用后文介绍的 `holdbasecharheight` 键来保证至少有 `basechar` 字符的高度。

```
\xeCJKDeclareCharClass{CJK}{。}
\framezi[resize=real,punctuation=leave,holdbasecharheight=false]{好的。一点、}
\framezi[resize=real,punctuation=leave,holdbasecharheight]{好的。一点、}
\xeCJKResetPunctClass
\framezi[resize=real,punctuation=leave,holdbasecharheight=false]{好的。一点、}
\framezi[resize=real,punctuation=leave,holdbasecharheight]{好的。一点、}
```

好的。一点好的。一点
好的。一点好的。一点

frametype	frametype = {none 口 十 田 米 咪 ...}	初始值 = none
resize	resize = {none real base square ...}	初始值 = none

`frametype` 设置方框样式。可用值的效果正如选项值: `口`—仅方框, `十`—仅中间的横线和竖线, `田`—常见的田字格, `米`—十字格再加上斜的两条线, `咪`—常见的米字格。还可自定义方框,详见第 5 节。以下 `frametype` 分别为 `none`, `口`, `十`, `田`, `米`, `咪`。

无 口 十 田 米 咪

`resize` 设置缩放方式。`real`,使用字符实际宽高缩放,`base`,使用 `basechar` 设置的字符的宽高缩放,`square`,使得字符的宽高相等再进行缩放。还可自定义缩放方式,详见第 5 节。

以下为宽度设置为 1cm,`resize` 分别为 `none`, `real`, `base`, `square` 时的缩放情况。

咪 实 基 方

<code>xscale</code>	<code>xscale = <scale ratio></code>	初始值 = 1
<code>yscale</code>	<code>yscale = <scale ratio></code>	初始值 = 1
<code>scale</code>	<code>scale = <scale ratio></code>	
<code>width</code>	<code>width = <dim></code>	
<code>height</code>	<code>height = <dim></code>	
<code>holdbasecharheight</code>	<code>holdbasecharheight = true false</code>	初始值 = false
<code>holdbasecharwidth</code>	<code>holdbasecharwidth = true false</code>	初始值 = false
<code>holdbasechar</code>	<code>holdbasechar = true false</code>	

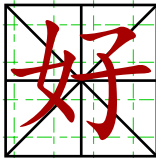
设置缩放比例和盒子宽高。

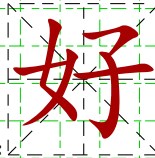
宽高具有更高的优先级,即若比例和宽高都设置了,则使用宽高来计算。宽高为二者都为0cm 视为未设置,二者有一大于0cm,视为设置了宽高。


此处的宽高和最终的宽高可能略有差异,最终的宽高保存在 `\zitiewidth` 和 `\zitieheight` dim 寄存器中。

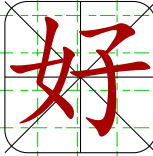
`holdbasecharheight` 设置为 true 时,将保证单个盒子至少有 `basechar` 的高度,
`holdbasecharwidth` 设置为 true 时,将设置盒子宽度为 `basechar` 的宽度。`holdbasechar` 将同时设置二者的值。


<code>linewidth</code>	<code>linewidth = <dim></code>	初始值 = 0.4pt
<code>dashpattern</code>	<code>dashpattern = {<dim1>, <dim2>, ... }</code>	
<code>framearc</code>	<code>framearc = <dim></code>	
<code>framearc*</code>	<code>framearc* = {{<dim1>}} {{<dim2>}}</code>	

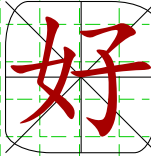
`linewidth=1pt:` 

`dashpattern={1mm,2mm,3mm,4mm}:` 

`framearc=3mm,frametype=□:` 

`framearc=3mm,frametype=咪:` 

`framearc*={3mm}{6mm},frametype=□:` 

`framearc*={3mm}{6mm},frametype=咪:` 

<code>framecolor</code>	<code>framecolor = <color expr></code>	初始值 = black
<code>framecolor*</code>	<code>framecolor* = {{models}} {{values}}</code>	
<code>charcolor</code>	<code>charcolor = <color expr></code>	初始值 = black
<code>charcolor*</code>	<code>charcolor* = {{models}} {{values}}</code>	
<code>fillcolor</code>	<code>fillcolor = <color expr></code>	
<code>fillcolor*</code>	<code>fillcolor* = {{models}} {{values}}</code>	

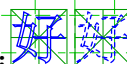
命名的颜色仅支持 L^AT_EX3 定义的 **black**, **white**, **red**, **green**, **blue**, **cyan**, **magenta** 和 **yellow**。

颜色模型和表达式也应使用 L^AT_EX3 支持的模型和表达式, 详见 [interface3.pdf\[5\]](#) 文档。

若要去掉 `fillcolor`, 应将其设置为空, 即 `fillcolor={}`, 而不是将其设置为白色。

<code>charstroke</code>	<code>charstroke = <none solid dashed whitesolid whitedashed invisible ...></code>	初始值 = none
<code>charstrokespecial</code>	<code>charstrokespecial = <pdf literal></code>	
<code>\zitiestrokechars</code>	<code>\zitiestrokechars {<pdf literal>} {<typeset material>}</code>	

设置字符的外框, 类似微软 Word 中字符的轮廓特效:



初始值 `none` 什么也不做。 `solid` 设置外轮廓为 0.25bp 的实线, `dashed` 设置外轮廓为 0.25bp 的虚线。这两个不会进行填充操作, 即背景色是什么颜色, 则字显示的就是背景, 字轮廓的颜色为 `charcolor` 的颜色。 `whitesolid` 和 `whitedashed` 在 `solid` 和 `dashed` 的基础上将字填充为白色。 `invisible` 将字符设置为不可见, 但不影响背景和网格的显示, 隐藏的字仍然可被复制。

`charstrokespecial` 将 `<pdf literal>` 为用在 `\special{pdf:code ...}` 中 `pdf` 的指令, 其中, `charstroke=solid` 相当于 `charstrokespecial={1 Tr 0.25 w [] 0 d 1 J}`, `charstroke=dashed` 相当于 `charstrokespecial={1 Tr 0.25 w [0.8] 0 d 1 J}`, `charstroke=whitesolid` 相当于

`charstrokespecial={2 Tr 0.25 w [] 0 d 1 J 1 -1 -1 rg}`, `charstroke=whitedashed` 相当于 `charstrokespecial={2 Tr 0.25 w [0.8] 0 d 1 J 1 -1 -1 rg}`。

`<pdf literal>` 可用的算符参考 [pdf reference\[7\]](#)。

于是可将字符渲染模式设置为 3 来隐藏字:

`\framezi[charstroke=solid]{字}`:



`\framezi[charstrokespecial={3 Tr}]{字}`:



还可使用 `\zitienuwrule` 来定义新的 `charstroke` 处理规则。这个键的处理器接收当前的 `<CJK char>` 作为其参数。具体的定义见 `\zitienuwrule` 的说明。

`\zitiestrokechars` 使用 `<pdf literal>` 处理 `<typeset material>`。

于是上述隐藏字的效果可用 `\zitienuwrule` 和 `\zitiestrokechars` 来定义一个 `invisible` 规则:

```
\zitienuwrule{charstroke}{invisible}{\zitiestrokechars{3 Tr}{#1}}
\framezi[width=1cm,charstroke=invisible]{字}
```



这与上面的效果是相同的。实际上, `zitie` 就是这样实现的。

<code>font</code>	<code>font = {font family}</code>	初始值 = 宋体
<code>savefontname</code>	<code>savefontname = true false</code>	初始值 = false
<code>fallback</code>	<code>fallback = true false</code>	初始值 = false
<code>fontbackfont</code>	<code>fontbackfont = {(fallback font list)}</code>	
<code>fontbackfont*</code>	<code>fontbackfont+= {(fallback font list)}</code>	

`font` 设置盒子中使用的字体, 若将其设置为空, 即 `font={}` 或 `font`, 则将使用正文字体。

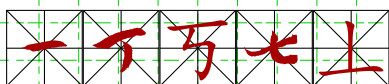
`savefontname` 设置是否保存字体名字, 若设置为真, 则将 `font` 键指定的字体名字保存在 `\zitiefontname` 中, 以便在后文使用, 如在标点符号的处理中或页眉页脚中。

`fallbackfont` 设置当前字体中不存在此字形时, 要使用哪些字体, 字体必须被 `\zitienewfont` 或 `\newCJKfontfamily` 声明。 `fallbackfont+` 将 `{(fallback font list)}` 局部的添加到之前的 `(fallback font list)` 中。

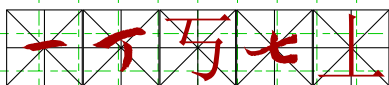
注意, `fallbackfont` 特性和 `xeCJK` 的 `FallBack` 并不冲突, 当 `fallback font list` 中的所有字体没有对应的字形时, 才会使用 `FallBack` 定义的字体。这一特性可通过设置 `fallback=false` 来关闭。

如在 `FZSunGuoTingCaoShuFU` (方正孙过庭草书) 字体中没有“ㄣ”, 虽然在在导言区使用 `\zitienewfont{ {孙过庭草繁}{FZSunGuoTingCaoShuFU}{FallBack=SimSun} }` 设置其 `FallBack` 字体为 `SimSun` (宋体), 但若设置 `fallbackfont={楷体}`, `fallback=true` 则会在楷体中查找是否有该字形, 然后才在宋体中查找:

`\framezi[fallbackfont={楷体}, width=1cm, font=孙过庭草繁, fallback, holdbasecharheight]{一丁ㄣ七上}`



`\framezi[width=1cm, font=孙过庭草繁, fallback, holdbasecharheight]{一丁ㄣ七上}`



<code>filepath</code>	<code>filepath = { (filepath1), (filepath2), ... }</code>
<code>filepath+</code>	

`filepath` 为 `\framezifile*` 添加文件搜索路径。其值为逗号分隔的列表。 `filepath+` 将新的搜索路径局部地添加到原有的路径上。

<code>\zitiesetup</code>	<code>\zitiesetup {(key-val)}</code>
<code>\zitiecolorlet</code>	<code>\zitiecolorlet {(name)} [(model)] {(value)}</code>

`\zitiesetup` 为 `zitie` 宏包的设置命令, 这个命令设置的值在当前组中有效。

`\zitiecolorlet` 定义新的颜色。注意, 若加载 `xcolor` 宏包, 使用 `\colorlet` 命令定义的颜色无法在 `zitie` 宏包中使用, 必须使用 `\zitiecolorlet` 来定义颜色。

<code>\zitienuwrule</code>	<code>\zitienuwrule [(arg nums)] {(option key)} {(rule)} {(replace)}</code>
----------------------------	---

为 `zitie` 宏包的键 `(option key)` 定义新的处理规则。将忽略 `(option key)` 和 `(rule)` 两端的空格。

`(arg nums)` 为键预先定义的参数数目。若未设置 `(arg nums)`, 则将其设置为 1。

`zitie` 宏包使用

```
\zitienuwrule{punctuation}{onlast}
{\penalty10000 \smash{\makebox[Opt]{%
\color{zitiecharcolor}\zitieCJKfamily{\zitiefontname}#1}}}
```

和

```
\zitienuwrule{punctuation}{scale}{
\hbox_set:Nn \l_tmpa_box
{ \color_select:n { zitiecharcolor } \zitieCJKfamily{\zitiefontname} #1 }
\box_scale:Nnn \l_tmpa_box \zitiexscaleratio \zitieyscaleratio
\box_use_drop:N \l_tmpa_box
}
```

定义了两个 `punctuation` 处理规则: `onlast` 和 `scale`, 这样就可以使用 `punctuation=onlast` 和 `punctuation=scale` 来对标点符号进行处理。

使用

```

\zitieneurule {charstroke} {solid}
{ \zitiestrokechars { 1 Tr 0.25 w [] 0 d 1 J } {#1} }
\zitieneurule {charstroke} {dashed}
{ \zitiestrokechars { 1 Tr 0.25 w [0.8] 0 d 1 J } {#1} }
\zitieneurule {charstroke} {whitesolid}
{ \zitiestrokechars { 2 Tr 0.25 w [] 0 d 1 J 1 1 1 rg } {#1} }
\zitieneurule {charstroke} {whitedashed}
{ \zitiestrokechars { 2 Tr 0.25 w [0.8] 0 d 1 J 1 1 1 rg } {#1} }
\zitieneurule {charstroke} {invisible} { \zitiestrokechars { 3 Tr } {#1} }

```

定义了五个 `charstroke` 处理规则,这样就可以使用 `charstroke=solid, ...` 来处理字符。

```

\zitiebasechar
\zitiebasecharwidth
\zitiebasecharheight
\zitiewidth
\zitieheight
\zitieboxwd
\zitieboxht
\zitieboxdp
\zitiefontname
\zitiexscaleratio
\zitieyscaleratio

```

`\zitiebasechar`、`\zitiebasecharwidth`、`\zitiebasecharheight` 是局部设置的。其它的为全局设置的接口。

```

zitieframecolor
zitiecharcolor
zitiefillcolor

```

这些是全局保存的颜色名,可在 `xcolor` 宏包的 `\color` 命令和 `LATEX3` 命令 `\color_select:n` 中使用。

第3节 background

在宏包加载时使用 `enable-background` 以启用该特性。

```

\zitiebackground \zitiebackground [{background key-val}]

```

使用背景。

background 可用的键值:(设置是全局的)

<code>true</code>	<code>true</code> 和 <code>on</code> 作用相同,表示设置网格背景。
<code>on</code>	<code>false</code> 和 <code>off</code> 作用相同,表示取消网格背景。
<code>false</code>	<code>next</code> 表示下一次设置背景,即当前页设置背景。这个选项也将取消之前设置的背景。
<code>off</code>	
<code>next</code>	

<code>colboxes</code>	<code>colboxes = <integer></code>	初始值 = 1
<code>rowboxes</code>	<code>rowboxes = <integer></code>	初始值 = 1
<code>framewidth</code>	<code>framewidth = <dim></code>	
<code>frameheight</code>	<code>frameheight = <dim></code>	
<code>boxwidth</code>	<code>boxwidth = <dim></code>	
<code>boxheight</code>	<code>boxheight = <dim></code>	
<code>onpaper</code>	<code>onpaper</code>	
<code>ontext</code>	<code>ontext</code>	

设置背景格子的大小。

`boxwidth` 和 `boxheight` 具有更高的优先级,若设置了,则背景每个格子的宽高为所设置的值,否则,使用 `<framewidth>`、`<frameheight>` 及 `<colboxes>`、`<rowboxes>` 来计算大小。

`onpaper` 设置 `frameheight=\paperheight`, `framewidth=\paperwidth`, `ontext` 设置 `frameheight=\textheight`, `framewidth=\textwidth`。

```
frametype
linewidth
framecolor
framecolor*
fillcolor
fillcolor*
dashpattern
```

和\framezi 等命令的选项相同,但二者互不影响。

颜色分别保存在 zitiebackgroundframecolor, zitiebackgroundfillcolor 颜色名中。可在\color 中直接使用,也可在\color_select:n 中使用。

```
xrange xrange = { <left> , <right> }
yrange yrange = { <top> , <bottom> }
```

设置网格显示的范围。注意这是实验性的,效果可能并不好。

本文档的设置为:

```
\ztiebackground[
  on, linewidth=0.2pt, dashpattern={1mm, 5mm, 2mm, 1mm},
  frametype=田, framecolor=green!80!black,
  colboxes=20, rowboxes=30,
]
```

background 的键值也可通过 \ztiesetup{background={{<background key-val>}}} 来设置。

第4节 zhlipsum, 中文乱数假文

ztie 宏包完全兼容 zhlipsum 宏包 [6]。使用\newzhlipsum 命令定义的假文也可使用。在加载宏包时使用 enable-zhlipsum 以启用该特性。

```
\framezhlipsum <key-val> {<paragraph list>} [{<name>}]
ztie/framezhlipsum/before
ztie/framezhlipsum/after
ztie/framezhlipsum/paragraph
```

<key-val> 为\framezi 第2节定义的选项,<paragraph list> 为段落列表,支持如下形式:

6-8, -3, 9。<name> 为假文的名字。可用值为 simp, trad, nanshanjing, xiangyu, zhufu, aspirin, 详细说明参考 zhlipsum 宏包的说明文档。

定义了 zitie/framezhlipsum/before, zitie/framezhlipsum/after, zitie/framezhlipsum/paragraph 三个钩子。其中 zitie/framezhlipsum/paragraph 为假文间要执行的代码,如有5段,则在这5个段落中间格执行一次,即执行4次。其它两个钩子的用法和前述相似。

第5节 编程接口

```
\ztie_new_frame_construct:nn
\ztie_frame_type:n
\ztie_frame_type_c:n
\ztie_new_resize_method:nn
\ztie_new_font:n
\ztie_new_font:nnn
```

创建新的构造器、resize 方法和字体。

```
\ztie_new_process_rule:nnn
\ztie_new_process_rule:nnnn
\ztie_processor:n
\ztie_processor_c:n
\ztie_processor:nnn
\ztie_processor_c:nnn
```

创建和使用处理器。

```
\zitie_color_set:nn
\zitie_color_set:nnn
```

颜色设置。

```
\zitie_cjk_glyph_use:nN
\zitie_token_class_dispatch:Nnnnn
\zitie_token_class_dispatch_o:Nnnnn
\zitie_token_class_dispatch_f:Nnnnn
```

字形选择和字符类别判断。

```
\zitie_token_if_punctuation:NTF
\zitie_token_if_punctuation_o:NTF
\zitie_token_if_punctuation_f:NTF
```

是否为标点符号。

```
\zitie_single_construct:N
\zitie_single_construct:nN
```

构造单个字的方框。

```
\zitie_background_new_frame_construct:nn
\zitie_background_frame_type:n
\zitie_background_frame_type_c:n
```

创建和使用新的 background frame 类型。

第 6 节 TODO

- 实现更快的、消耗资源更少的版本。
- 实现非单字处理的版本。
- 支持 Lua \TeX 。
- 优化背景模块。
- 增加间距功能。
- 更多网格效果。
- 支持 debug 信息和拼音。
- 更多缩放和变换效果。
- 对字符类别进行更多的分类处理。
- 支持透明度, (需要 `l3opacity`)。
- 等。

参考文献

- [1] [CTEX.ORG](https://ctex.org), *xeCJK* 宏包
- [2] WILL ROBERTSON, With contributions by Khaled Hosny, Philipp Gesang, Joseph Wright, and others. *The fontspec package*
- [3] JOHANNES BRAAMS, DAVID CARLISLE, ALAN JEFFREY, LESLIE LAMPORT, FRANK MITTELBACH, CHRIS ROWLEY, RAINER SCHÖPF, *The L^AT_EX2_ε Sources*,
- [4] FRANK MITTELBACH, *The lthooks package*
- [5] THE L^AT_EX PROJECT, *The L^AT_EX3 Interfaces*
- [6] 曾祥东, zhlipsum: 中文乱数假文 (Lorem ipsum)
- [7] ADOBE® SYSTEMS INCORPORATED, *Adobe® Portable Document Format*

版本历史

v1.2

(2021/09/23)

General: charstroke 新增 invisible 值。 5

不再加载 ctexsize 宏包。不再设置宋体为主字体 2

修改了一些文档错误,将 \zitie_color_select:... 修

改为 \zitie_color_set:... 2

新增 \zitiebasechar、\zitiebasecharwidth、

\zitiebasecharheight 7

新增 fontsize 选项,不再使用 zihao 选项作为默认字

号选项。 3

新增 holdbasecharwidth、holdbasecharheight、

holdbasechar 选项。 4

代码索引

意大利体的数字表示描述对应索引项的页码;带下划线的数字表示定义对应索引项的代码行号;罗马字体的数字表示使用对应索引项的代码行号。

A

\AddToHook 2

\AddToHookNext 2

B

basechar 3

boxheight 8

boxwidth 8

C

charcolor 5

charcolor* 5

charstroke 5

charstrokespecial 5

\CJKfamily 2

colboxes 8

color commands:

\color_select:n 7,8

D

dashpattern 4,8

F

fallback 6

false 7

filepath 6

filepath+ 6

fillcolor 5,8

fillcolor* 5,8

font 6

fontbackfont 6

fontbackfont* 6

fontsize 3

framearc 4

framearc* 4

framecolor 5,8

framecolor* 5,8

frameheight 8

\framerange 2

frametype 4,8

framewidth 8

H

\framezhlipsum 8

\framezi 2,3,8

\framezifile 2,6

H

height 4

holdbasechar 4

holdbasecharheight 4

holdbasecharwidth 4

L

linewidth 4,8

N

\newCJKfontfamily 6

\newzhlipsum 8

next 7

O

off 7

on 7

onpaper 8

ontext 8

P

punctuation 3

punctuation* 3

R

\RemoveFromHook 2

resize 4

rowboxes 8

S

savefontname 6

scale 4

T

TeX and L^AT_EX 2_ε commands:

\@gobble 3

\color 7,8

\colorlet 6

\fbox	3	\zitie_single_construct:N	9
\normalsize	3	\zitie_single_construct:nN	9
\special	5	\zitie_token_class_dispatch:Nnnnn	9
\usepackage	2	\zitie_token_class_dispatch_f:Nnnnn	9
\zitiebasecharheight	3	\zitie_token_class_dispatch_o:Nnnnn	9
\zitiebasecharwidth	3	\zitie_token_if_punctuation:NTF	9
\zitiefontname	6	\zitie_token_if_punctuation_f:NTF	9
\zitieheight	4	\zitie_token_if_punctuation_o:NTF	9
\zitiewidth	4	zitie/framerange/after	2
true	7	zitie/framerange/before	2
W			
width	4	zitie/framerange/range	2
X			
\xeCJKDeclareSubCJKBlock	2	zitie/framezhlipsum/after	8
xrange	8	zitie/framezhlipsum/before	8
xscale	4	zitie/framezhlipsum/paragraph	8
Y			
yrange	8	zitie/framezi/after	2
yscale	4	zitie/framezi/before	2
Z			
zihao	3	zitie/framezifile/after	2
zitie commands:		zitie/framezifile/before	2
\zitie_background_frame_type:n	9	zitie/zitiefame/after	3
\zitie_background_frame_type_c:n	9	zitie/zitiefame/before	3
\zitie_background_new_frame_construct:nn	9	zitie/zitiefame/par	3
\zitie_cjk_glyph_use:nN	9	\zitiebackground	7
\zitie_color_set:nn	9	\zitiebasechar	3,7
\zitie_color_set:nnn	9	\zitiebasecharheight	7
\zitie_frame_type:n	9	\zitiebasecharwidth	7
\zitie_frame_type_c:n	9	\zitieboxdp	7
\zitie_new_font:n	9	\zitieboxht	7
\zitie_new_font:nnn	9	\zitieboxwd	7
\zitie_new_frame_construct:nn	9	zitiecharcolor	7
\zitie_new_process_rule:nnn	9	\zitieCJKfamily	2
\zitie_new_process_rule:nnnn	9	\zitiecolorlet	6
\zitie_new_resize_method:nn	9	zitiefillcolor	7
\zitie_processor:n	9	\zitiefontname	7
\zitie_processor:nnn	9	zitieframe	3
\zitie_processor_c:n	9	zitieframecolor	7
\zitie_processor_c:nnn	9	\zitieheight	7
		\zitieneufont	2,6
		\zitieneufont	3,5,6
		\zitiesetup	3,6,8
		\zitiestrokechars	5,6
		\zitiewidth	7
		\zitiexscaleratio	7
		\zitieyscaleratio	7