

字帖宏包

陈文建(Longaster)

农历二〇二一年八月十五 v1.1

我欲乘風向北飛，雲借船風初之見來。
軒轅東游得仙子，萬品絕盡歸來。
玄龍吟奈我，海裏長風萬人不回。
望巔青不見天涯人。

我欲乘風向北飛，雲借船風初之見來。
軒轅東游得仙子，萬品絕盡歸來。
玄龍吟奈我，海裏長風萬人不回。

——《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),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_YT_EX,待后续版本支持 LuaT_EX,不支持 pdfT_EX。

目录

第 1 节 zitie 宏包的加载和基本命令	2	第 5 节 编程接口	8
第 2 节 选项	3	第 6 节 TODO	8
第 3 节 background	7	代码索引	10
第 4 节 zhlipsum, 中文乱数假文	7		

第 1 节 zitie 宏包的加载和基本命令

加载宏包:`\usepackage[⟨key-val⟩]{zitie}`。⟨key-val⟩ 选项见以下几节。

若要加载 ctex 宏包,请先加载 ctex 然后再加载 zitie 宏包。若加载本宏包时未加载 ctex 宏包,则会加载 xeCJK 宏包和 ctexsize 宏包。如果是 Windows 平台,还将设置宋体为主字体。

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

⟨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 1000 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 %。

```

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 来添加和执行,详细请参考 source2e.pdf 或 lthooks 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 = <CJK char>
zihao = <字号>

```

初始值 = 好
初始值 = 4

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。

zihao 设置计算缩放比例时的字号大小。

```

punctuation = {ignore|leave|onlast|scale|...}
punctuation* = {replace}

```

初始值 = ignore

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 L^AT_EX 2_ε 定义的命令,其作用与 punctuation=ignore 相同。

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

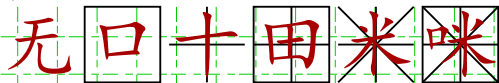
实际上,onlast 和 scale 就是由 \zitieneurule 定义的,这个键的处理器接收一个标点符号作为其参数。具体的定义见 \zitieneurule 命令的说明。

TpXhackers 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。

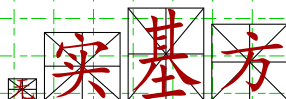
frametype	frametype = <none 口 十 田 米 咪 ...>
resize	resize = <none real base square ...>

初始值 = none
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。



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 时的缩放情况。

xscale	xscale = <scale ratio>
yscale	yscale = <scale ratio>
scale	scale = <scale ratio>
width	width = <dim>
height	height = <dim>

初始值 = 1

初始值 = 1

设置缩放比例和盒子宽高。

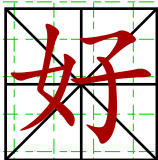
宽高具有更高的优先级, 即若比例和宽高都设置了, 则使用宽高来计算。宽高为三者都为 0cm 视为未设置, 二者有一大于 0cm, 视为设置了宽高。

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。

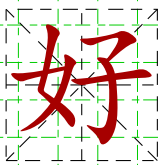
linewidth	linewidth = <dim>
dashpattern	dashpattern = { <-dim1, dim2, ...> }
framearc	framearc = <dim>
framearc*	framearc* = { {{dim1}} {{dim2}} }

初始值 = 0.4pt

linewidth=1pt:



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



framearc=3mm,frametype=口:

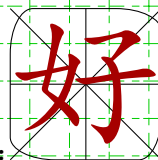


framearc=3mm,frametype=咪:



framearc*={3mm}{6mm},frametype=口:





framearc*={3mm}{6mm},frametype= 咪:

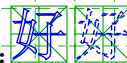
framecolor	framecolor = {color expr}	初始值 = black
framecolor*	framecolor* = {{models}} {{values}}	
charcolor	charcolor = {color expr}	初始值 = black
charcolor*	charcolor* = {{models}} {{values}}	
fillcolor	fillcolor = {color expr}	
fillcolor*	fillcolor* = {{models}} {{values}}	

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 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。

charstroke	charstroke = {none solid dashed whitesolid whitedashed ...}	初始值 = none
charstrokespecial	charstrokespecial = {pdf literal}	
\zitiestrokechars	\zitiestrokechars {{pdf literal}} {{typeset material}}	

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。

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}。

还可使用 \ziti newrule 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 \ziti newrule 的说明。

\zitiestrokechars 使用 {pdf literal} 处理 {typeset material}。

font	font = {font family}	初始值 = 宋体
savefontname	savefontname = true false	初始值 = false
fallback	fallback = true false	初始值 = false
fontbackfont	fontbackfont = {{fallback font list}}	
fontbackfont*	fontbackfont+ = {{fallback font list}}	

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, 以便在后文使用, 如在标点符号的处理中。

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 \ziti newfont 或 \newCJKfontfamily 声明。fallbackfont+ 将 {{fallback font list}} 添加的之前的 {fallback font list} 中。

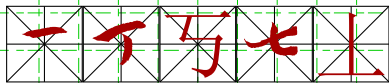
注意, fallbackfont 特性和 xeCJK 的 FallBack 并不冲突, 当 fallback font list 中的所有字体没有对应的字形时, 才会使用 FallBack 定义的字体。这一特性可通过 fallback=false 来关闭。

如在 FZSunGuoTingCaoShuFU (方正孙过庭草书) 字体中没有“乚”, 虽然在在导言区使用 \ziti newfont{ {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, 然后再在宋体中查找:

\framezi[fallbackfont={楷体},width=1cm,font=孙过庭草繁,fallback]{一丁乚乚}



```
\framezi[width=1cm,font=孙过庭草繁,fallback]{一丁万七上}
```



```
filepath filepath = <filepath1, filepath2, ...>
```

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

```
\zitiesetup \zitiesetup <{key-val}>
\zitiecolorlet \zitiecolorlet <{<name>}> [<model>] <{<value>}>
```

用于设置选项,定义新的颜色。注意,若加载 `xcolor` 不能使用 `\colorlet` 命令,必须使用 `\zitiecolorlet` 来设置颜色。

```
\zitieneurule \zitieneurule [<arg nums>] <{<option key>}> <{<rule>}> <{<replace>}>
```

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` 宏包使用了

```
\zitieneurule{punctuation}{onlast}
{ \penalty10000 \smash{\makebox[0pt]{%
\color{zitiecharcolor}\zitieCJKfamily{\zitiefontname}#1}}}
```

和

```
\zitieneurule{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} }
```

定义了四个 `charstroke` 处理规则,这样就可以使用 `charstroke=solid, ...` 来处理字符。

这些命令为全局设置的接口。

```
\zitiewidth
\zitieheight
\zitieboxwd
\zitieboxht
\zitieboxdp
\zitiefontname
\zitiexscaleratio
\zitieyscaleratio
```

```
zitieframecolor
zitiecharcolor
zitiefillcolor
```

这些是全局保存的颜色名,可在 `xcolor` 宏包的 `\color` 命令和 \LaTeX 3 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。

<code>\zitiebackground</code>	<code>\zitiebackground [{background key-val}]</code>
-------------------------------	--

使用背景。

`background` 可用的键值: (设置是全局的)

<code>true</code> <code>on</code> <code>false</code> <code>off</code> <code>next</code>	<code>true</code> 和 <code>on</code> 作用相同, 表示设置网格背景。 <code>false</code> 和 <code>off</code> 作用相同, 表示取消网格背景。 <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>colboxes</code> <code>rowboxes</code> <code>framewidth</code> <code>frameheight</code> <code>boxwidth</code> <code>boxheight</code> <code>onpaper</code> <code>ontext</code>	<code>colboxes</code> = $\langle integer \rangle$ <code>rowboxes</code> = $\langle integer \rangle$ <code>framewidth</code> = $\langle dim \rangle$ <code>frameheight</code> = $\langle dim \rangle$ <code>boxwidth</code> = $\langle dim \rangle$ <code>boxheight</code> = $\langle dim \rangle$ <code>onpaper</code> <code>ontext</code>	初始值 = 1 初始值 = 1
---	---	--------------------

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 $\langle framewidth \rangle$ 、 $\langle frameheight \rangle$ 及 $\langle colboxes \rangle$ 、 $\langle rowboxes \rangle$ 来计算大小。

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

<code>frametype</code> <code>linewidth</code> <code>framecolor</code> <code>framecolor*</code> <code>fillcolor</code> <code>fillcolor*</code> <code>dashpattern</code>	和 <code>\framezi</code> 等命令的选项相同, 但三者互不影响。 颜色分别保存在 <code>zitiebackgroundframecolor</code> , <code>zitiebackgroundfillcolor</code> 颜色名中。可在 <code>\color</code> 中直接使用, 也可在 <code>\color_select:n</code> 中使用。
--	---

<code>xrange</code> <code>yrange</code>	<code>xrange</code> = { $\langle left \rangle$, $\langle right \rangle$ } <code>yrange</code> = { $\langle top \rangle$, $\langle bottom \rangle$ }
--	--

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:

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

第4节 zhlipsum, 中文乱数假文

`zitie` 宏包完全兼容 `zhlipsum` 宏包。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          \framezhlipsum [{key-val}] {{paragraph list}} [(name)]
zitie/framezhlipsum/before
zitie/framezhlipsum/after
zitie/framezhlipsum/paragraph
```

$\langle key-val \rangle$ 为 \framezi 第 2 节定义的选项, $\langle paragraph list \rangle$ 为段落列表, 支持如下形式: 6-8, -3, 9。 $\langle name \rangle$ 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

```
\zitie_new_frame_construct:nn
\zitie_frame_type:n
\zitie_frame_type_c:n
\zitie_new_resize_method:nn
\zitie_new_font:n
\zitie_new_font:nnn
```

创建新的构造器、resize 方法和字体。

```
\zitie_new_process_rule:nnn
\zitie_new_process_rule:nnnn
\zitie_processor:n
\zitie_processor_c:n
\zitie_processor:nnn
\zitie_processor_c:nnn
```

创建和使用处理器。

```
\zitie_color_select:nn
\zitie_color_select:nnn
```

颜色选择。

```
\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。

- 支持 `LuaTeX`。
- 优化背景模块。
- 增加间距功能。
- 更多网格效果。
- 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。

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		\newzhlipsum		7
\AddToHook	2	next		7
\AddToHookNext	2	O		
B		off		7
basechar	3	on		7
boxheight	7	onpaper		7
boxwidth	7	ontext		7
C		P		
charcolor	5	punctuation	3	
charcolor*	5	punctuation*	3	
charstroke	5	R		
charstrokespecial	5	\RemoveFromHook	2	
\CJKfamily	2	resize	4	
colboxes	7	rowboxes	7	
color commands:		S		
\color_select:n	6,7	savefontname	5	
D		scale	4	
dashpattern	4,7	T		
F		TeX and L ^A T _E X 2 _ε commands:		
fallback	5	\@gobble	3	
false	7	\color	6,7	
filepath	6	\colorlet	6	
filepath+	6	\fbbox	3	
fillcolor	5,7	\special	5	
fillcolor*	5,7	\usepackage	2	
font	5	\zitieheight	4	
fontbackfont	5	\zitiewidth	4	
fontbackfont*	5	true	7	
framearc	4	W		
framearc*	4	width	4	
framecolor	5,7	X		
framecolor*	5,7	\xeCJKDeclareSubCJKBlock	2	
\framefile	6	xrange	7	
frameheight	7	xscale	4	
\framerange	2	Y		
frametype	4,7	yrange	7	
framewidth	7	yscale	4	
\framezhlipsum	8	Z		
\framezi	2,7,8	zihao	3	
\framezifile	2	zitie commands:		
H		\zitie_background_frame_type:n	8	
height	4	\zitie_background_frame_type:c:n	8	
L		\zitie_background_new_frame_construct:nn	8	
linewidth	4,7	\zitie_cjk_glyph_use:nN	8	
N		\zitie_color_select:nn	8	
\newCJKfontfamily	5	\zitie_color_select:nnn	8	

\zitie_frame_type:n	8	zitie/framezi/after	2
\zitie_frame_type_c:n	8	zitie/framezi/before	2
\zitie_new_font:n	8	zitie/framezifile/after	2
\zitie_new_font:nnn	8	zitie/framezifile/before	2
\zitie_new_frame_construct:nn	8	zitie/zitieframe/after	3
\zitie_new_process_rule:nnn	8	zitie/zitieframe/before	3
\zitie_new_process_rule:nnnn	8	zitie/zitieframe/par	3
\zitie_new_resize_method:nn	8	\zitiebackground	7
\zitie_processor:n	8	\zitieboxdp	6
\zitie_processor:nnn	8	\zitieboxht	6
\zitie_processor_c:n	8	\zitieboxwd	6
\zitie_processor_c:nnn	8	zitiecharcolor	6
\zitie_single_construct:N	8	\zitieCJKfamily	2
\zitie_single_construct:nN	8	\zitiecolorlet	6
\zitie_token_class_dispatch:Nnnnn	8	zitiefillcolor	6
\zitie_token_class_dispatch_f:Nnnnn	8	\zitiefontname	6
\zitie_token_class_dispatch_o:Nnnnn	8	zitieframe	3
\zitie_token_if_punctuation:NTF	8	zitieframecolor	6
\zitie_token_if_punctuation_f:NTF	8	\zitieheight	6
\zitie_token_if_punctuation_o:NTF	8	\zitienewfont	2,5
zitie/framerange/after	2	\zitienewrule	3,5,6
zitie/framerange/before	2	\zitiesetup	6
zitie/framerange/range	2	\zitiestrokechars	5
zitie/framezhlipsum/after	8	\zitiewidth	6
zitie/framezhlipsum/before	8	\zitiexscaleratio	6
zitie/framezhlipsum/paragraph	8	\zitieyscaleratio	6