zTool 接口文档

Eureka

由于本人时间有限，目前此宏包的开发暂停.

2025 年 5 月 29 日

总目录

1 基本介绍

zTeX 宏集已独立实现了一个 ztool 宏包, 此宏包中包含原来已被废弃的 l3sys-shell 中的所有命令. 除此之外, ztool 提供了 box 操作, 文件 IO 以及基本图形绘制相关的函数. 在 ztool 的协助下, zTeX 能够避免或减少命令行 `-shell-escape` 参数或其它相关宏包的调用 (如 robust-externalize 宏包).

本宏包在 Github 上的地址如下:

https://github.com/zongpingding/zTeX_bundle

该仓库中包含本宏集的源码与用户手册; 当前宏集的稳定版本于半年之前发布, 最新的开发版请切换到 “dev” 分支; 本手册适用于当前最新的开发版.

2 宏包选项

ztool 分为了“shell-escape，file-io，box，zdraw”四个库，每一个库之间互不影响，均可单独加载. 默认不加载任意的 ztool 库.

ztool/shell-escape
ztool/file-io
ztool/box
ztool/zdraw

shell-escape = ⟨***false***|*true*⟩ ..... 初始值：false
file-io = ⟨***false***|*true*⟩ ..... 初始值：false
box = ⟨***false***|*true*⟩ ..... 初始值：false
zdraw = ⟨***false***|*true*⟩ ..... 初始值：false

New: 2025-05-22

这四个选项为 ztool 宏包的选项，可以在加载 ztool 宏包时使用，一个基本的使用样例如下，该示例加载了 ztool 的 shell-escape 库和 box 库:

```
\usepackage[shell-escape, box=true]{ztool}
```
例 1

\ztoolloadlib

\ztoolloadlib {⟨*library*⟩}

New: 2025-05-22

此命令用于加载 ztool 库，⟨*library*⟩ 为库的名称，可选值有：“shell-escape，file-io，box，zdraw”.

一个基本的使用样例如下，该示例加载了 ztool 的 shell-escape 库和 box 库:

```
\ztoolloadlib{shell-escape, box}
```
例 2

3 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在编译 LaTeX 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, 部分不当操作可能会导致一些无法挽救的后果.**

\ztool_shell_escape:n
\ztool_shell_escape:e

Updated: 2024-12-05

\ztool_shell_escape:n {⟨*command*⟩}

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:e

Updated: 2024-12-05

\ztool_shell_mkdir:n {⟨*dir*⟩}

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:e

Updated: 2024-12-05

\ztool_shell_rm:n {⟨*file*⟩}

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:e

Updated: 2024-12-05

\ztool_shell_rmdir:n {⟨*dir*⟩}

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 ⟨*tl*⟩ 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

\ztool_shell_split_ls:nN {⟨*dir*⟩}⟨*tl*⟩

Updated: 2024-12-05

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

4 File IO

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 `\ztool_file_new:nn {⟨bool⟩}{⟨file⟩}`

Updated: 2024-12-05

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 ⟨bool⟩ 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 {⟨bool⟩}{⟨file⟩}⟨seq⟩`
\ztool_read_file_as_seq:(neN|nnc|nec)

Updated: 2024-12-05

此命令用于读取文件 ⟨file⟩ 的内容, 并将其存放在 ⟨seq⟩ 中, 如果 ⟨file⟩ 不存在, 则 ⟨seq⟩ 会被置为空. ⟨bool⟩ 用于控制是否保留行尾的空格, 可选值有:\c_true_bool, \c_false_bool; 如果 ⟨bool⟩ 为 \c_true_bool, 则保留行尾的空格, 否则不保留. **注意**: ⟨seq⟩ 中的内容对应的 catcode 不变, 且此命令仅在当前组生效.

\ztool_gread_file_as_seq:nnN `\ztool_read_file_as_seq:nnN {⟨bool⟩}{⟨file⟩}⟨seq⟩`
\ztool_gread_file_as_seq:(neN|nnc|nec)

Updated: 2025-01-05

此命令用于读取文件 ⟨file⟩ 的内容, 并将其存放在 ⟨seq⟩ 中, 如果 ⟨file⟩ 不存在, 则 ⟨seq⟩ 会被置为空. ⟨bool⟩ 用于控制是否保留行尾的空格, 可选值有:\c_true_bool, \c_false_bool; 如果 ⟨bool⟩ 为 \c_true_bool, 则保留行尾的空格, 否则不保留. **注意**: ⟨seq⟩ 中的内容对应的 catcode 不变, 且此命令仅在当前组生效.

\ztool_write_seq_to_file:nNn `\ztool_write_seq_to_file:nNn {⟨bool⟩}⟨seq⟩{⟨file⟩}`
\ztool_write_seq_to_file:(nNe|nNV)

New: 2025-05-27

此命令用于将 ⟨seq⟩ 按行写入到文件 ⟨file⟩ 中, 如果 ⟨file⟩ 不存在, 则会创建一个名为 ⟨file⟩ 的新文件; 若 ⟨file⟩ 已经存在, 则可以使用 ⟨bool⟩ 控制当前的写入模式: ⟨bool⟩ 为 \c_true_bool 时, **覆盖写入**; ⟨bool⟩ 为 \c_false_bool 时, 追加写入; 如果 ⟨seq⟩ 为空, 则不会进行任何操作.

\ztool_append_to_file:nn `\ztool_append_to_file:nn {⟨file⟩}{⟨content⟩}`
\ztool_append_to_file:(no|nf|ee)

Updated: 2025-01-05

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

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

\ztool_replace_file_line:nnn {⟨*file*⟩}{⟨*line*⟩}{⟨*content*⟩}

Updated: 2025-01-05

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

\ztool_insert_to_file:nnn
\ztool_insert_to_file:(nen|nfn|een)

\ztool_insert_to_file:nnn {⟨*file*⟩}{⟨*line*⟩}{⟨*content*⟩}

Updated: 2025-01-05

此命令用于将 ⟨*content*⟩ 插入到文件 ⟨*file*⟩ 的第 ⟨*line*⟩ 行之前, 如果 ⟨*file*⟩ 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:

例 3

```
\ExplSyntaxOn
\ztool_file_new:nn {\c_true_bool}{testIO.txt}
\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} {1} {|INSERT-~-CONTENT|}
\ztool_append_to_file:nn {testIO.txt} {|APPEND-CONTENT-II|}
\ztool_replace_file_line:nnn {testIO.txt} {3} {|REPLACE-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|
|REPLACE-CONTENT|

```
|INSERT- -CONTENT|
|APPEND-CONTENT|
|REPLACE-CONTENT|
```

5 盒子操作

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.

\ztool_get_ht:Nn
\ztool_get_ht:(Ne|ce)

\ztool_get_ht:Nn ⟨*dim*⟩{⟨*content*⟩}

此命令用于将 ⟨*content*⟩ 的高度保存在 ⟨*dim*⟩ 这一寄存器中.

Updated: 2024-12-05

\ztool_get_ht_plus_dp:Nn
\ztool_get_ht_plus_dp:(Ne|ce)

\ztool_get_ht:Nn ⟨*dim*⟩{⟨*content*⟩}

此命令用于将 ⟨*content*⟩ 的高度和深度的和保存在 ⟨*dim*⟩ 这一寄存器中.

Updated: 2024-12-05

\ztool_get_wd:Nn
\ztool_get_wd:(Ne|ce)

\ztool_get_wd:Nn ⟨*dim*⟩{⟨*content*⟩}

此命令用于将 ⟨*content*⟩ 的宽度保存在 ⟨*dim*⟩ 这一寄存器中.

Updated: 2024-12-05

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

\ztool_get_dp:Nn ⟨*dim*⟩{⟨*content*⟩}

此命令用于将 ⟨*content*⟩ 的深度保存在 ⟨*dim*⟩ 这一寄存器中.

Updated: 2024-12-05

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

\ztool_gget_ht:Nn ⟨*dim*⟩{⟨*content*⟩}

此命令用于将 ⟨*content*⟩ 的高度保存在 ⟨*dim*⟩ 这一寄存器中, 并且此操作是全局的.

Updated: 2024-12-05

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

\ztool_gget_wd:Nn ⟨*dim*⟩{⟨*content*⟩}

此命令用于将 ⟨*content*⟩ 的宽度保存在 ⟨*dim*⟩ 这一寄存器中, 并且此操作是全局的.

Updated: 2024-12-05

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

\ztool_gget_dp:nn ⟨*dim*⟩{⟨*content*⟩}

此命令用于将 ⟨*content*⟩ 的深度保存在 ⟨*dim*⟩ 这一寄存器中, 并且此操作是全局的.

Updated: 2024-12-05

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

\ztool_set_to_wd:nn {⟨*dim*⟩}{⟨*content*⟩}

此命令用于将 ⟨*content*⟩ 的宽度调整为 ⟨*dim*⟩, 然后排版出来.

Updated: 2024-12-05

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

\ztool_set_to_ht:nn {⟨*dim*⟩}{⟨*content*⟩}

此命令用于将 ⟨*content*⟩ 的高度调整为 ⟨*dim*⟩, 然后排版出来.

Updated: 2024-12-05

\ztool_autoset_to_wd_and_ht:nnn
\ztool_autoset_to_wd_and_ht:(nne|een|eee)
Updated: 2025-04-29

\ztool_autoset_to_wd_and_ht:nn {⟨*width*⟩}{⟨*height*⟩}{⟨*content*⟩}

此命令用于将 ⟨*content*⟩ 的宽度调整为 $\min(\langle width\rangle, \langle height\rangle)$, 然后排版出来.

\ztool_rotate:nn
\ztool_rotate:(en|ne|ee)
New: 2025-04-29

\ztool_rotate:nn {⟨*angle*⟩}{⟨*content*⟩}

此命令用于将 ⟨*content*⟩ 旋转 ⟨*angle*⟩ 度, 然后排版出来.

\ztool_scale_to_wd:nn
\ztool_scale_to_wd:(en|ne|ee)
New: 2025-04-29

\ztool_scale_to_wd:nn {⟨*dim*⟩}{⟨*content*⟩}

此命令用于将 ⟨*content*⟩ 的宽度调整为 ⟨*dim*⟩, 但是不对盒子的高度做任何的调整, 然后排版出来.

\ztool_scale_to_ht:nn
\ztool_scale_to_ht:(en|ne|ee)
New: 2025-04-29

\ztool_scale_to_ht:nn {⟨*dim*⟩}{⟨*content*⟩}

此命令用于将 ⟨*content*⟩ 的高度 + 深度整体调整为 ⟨*dim*⟩, 但是不对盒子的宽度做任何的调整, 然后排版出来.

\ztool_scale_to_wd_and_ht:nnn
\ztool_scale_to_wd_and_ht:(nno|nne|eee)
New: 2025-04-29

\ztool_scale_to_wd_and_ht:nnn {⟨*width*⟩}{⟨*height*⟩}{⟨*content*⟩}

此命令用于将 ⟨*content*⟩ 的宽度调整为 ⟨*width*⟩, 高度 + 深度整体调整为 ⟨*height*⟩, 然后排版出来.

\ztool_box_item_align:Nnnn
\ztool_box_item_align:(cnnn|Nnno|cnno|Nnen|Nnee)
Updated: 2025-05-13

\ztool_box_item_align:Nnnn ⟨*cmd*⟩{⟨*width*⟩}{⟨*content*⟩}{⟨*align*⟩}

此命令用于将 ⟨*content*⟩ 的宽度调整为 ⟨*width*⟩, 然后排版出来, ⟨*align*⟩ 用于控制对齐方式, 可选值有: `left`, `center`, `right`, `scatter`. ⟨*cmd*⟩ 为一个命令, 其接受一个参数, 它将应用到 ⟨*content*⟩ 的每一个 Token 上. **注意**: ⟨*content*⟩ 中的空格会被忽略, 如果需要空格, 请使用 "\␣" 或 "~" 替代.

\ztool_fp_to_rad:n
New: 2025-05-12

\ztool_fp_to_rad:n {⟨*angle*⟩}

此命令用于将 ⟨*angle*⟩ 从弧度制转换为角度制.

\ztool_affine_transformation:Nnnnn
\ztool_affine_transformation:(Neeee|cnnnn|ceeee)

New: 2025-05-12

\ztool_affine_transformation:Nnnnn ⟨*coffin*⟩{⟨a⟩}{⟨b⟩}{⟨c⟩}{⟨d⟩}

此命令用于对 ⟨*coffin*⟩ 进行任意的仿射变换 (线性变换), 具体的使用方法可以参见前述的 ztoolboxaffine 命令; 上述参数对应的仿射变换矩阵 Λ 为

$$\Lambda = \begin{bmatrix} a & c \\ b & d \end{bmatrix}.$$

关于上述函数 \ztool_affine_transformation:Nnnnn 的一些技术细节: 给定任意一个仿射变换 Λ, 不妨设

$$\Lambda = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}.$$

我们可以做如下的分解 (与 SVD 分解类似), 令 $m = 2x$, 则有:

$$\begin{aligned} \Lambda &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & m \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \\ &= \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} \cos\omega & -\sin\omega \\ \sin\omega & \cos\omega \end{bmatrix} \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}. \end{aligned} \tag{5.1}$$

我们给出如下的记号:

- $\mathbf{T}_1(\theta)$: 旋转矩阵, 绕原点逆时针旋转 θ 角;
- $\mathbf{T}_2(x)$: 缩放矩阵, 把 x 轴方向的所有向量变为原来的 x 倍;
- $\mathbf{T}_3(y)$: 缩放矩阵, 把 y 轴方向的所有向量变为原来的 y 倍;

那么我们可以认为 $\{\mathbf{T}_1(\theta), \mathbf{T}_2(x), \mathbf{T}_3(y)\}$ 就是 $A_{2\times 2}$ 的基. 所以我们可以把上面的 方程 (5.1) 写成如下表达式:

$$\Lambda = \mathbf{T}_1(\theta) \cdot \mathbf{T}_1(\phi) \cdot \mathbf{T}_2(S_x) \cdot \mathbf{T}_3(S_y) \cdot \mathbf{T}_1(\omega) \cdot \mathbf{T}_2(s_x) \cdot \mathbf{T}_3(s_y). \tag{5.2}$$

根据矩阵乘法的结果, 我们可以知道上述的 m, s_x, S_x, ϕ 等参数如下:

$$s_x = \sqrt{A_{11}^2 + A_{21}^2}, \qquad \theta = \arctan\left(\frac{A_{21}}{A_{11}}\right).$$

s_y 和 m 的求解结果如下:

$$ms_y = A_{12}\cos\theta + A_{22}\sin\theta, \qquad s_y = \begin{cases} \dfrac{ms_y\cos\theta - A_{12}}{\sin\theta} & \text{如果 } \sin\theta \neq 0, \\ \dfrac{A_{22} - ms_y\sin\theta}{\cos\theta} & \text{如果 } \sin\theta = 0; \end{cases}$$

那么此时很容易知道 $m = ms_y/s_y$. 对 shear matrix 的分解结果如下:

$$S_x = \sqrt{\frac{m^2}{4}+1} - \frac{m}{2}, \qquad S_y = \sqrt{\frac{m^2}{4}+1} + \frac{m}{2},$$
$$\phi = -\frac{\pi}{4} - \frac{1}{2}\arctan(\frac{m}{2}), \qquad \omega = \frac{\pi}{4} - \frac{1}{2}\arctan(\frac{m}{2}).$$

最后我们只需要从右到左将这一系列的变换应用到 ⟨*box*⟩ 上即可. 从上面也可以看出, 命令 `\ztool_affine_transformation:Nnnnn` 仅依赖于 LaTeX3 中的 `\coffin_scale:Nnn` 和 `\coffin_rotate:Nn` 两个函数. 命令 `\ztool_affine_transformation:Nnnnn` 实现过程中相关的参考链接如下:

- `https://math.stackexchange.com/a/3521141/1235323`;
- `https://math.stackexchange.com/a/281087/1235323`.

> 如果原 TeX 引擎提供了 shear transformation 相关的 primitive, 那么上述对 shear matrix 的分解就是不必要的. 部分的引擎中原始提供了仿射变换矩阵这一 primitive, 比如 pdfTeX 中的 `\pdfsetmatrix` 命令.

下面的示例展示了如何使用这一章节中的几个 Box 操作命令:

例 4

```
\ExplSyntaxOn
\setlength{\fboxsep}{0pt}
% get dim of content
\dotfill\par
\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\par

% set content to dim
\dotfill\par
Hello,~world|
\ztool_set_to_ht:nn {.5cm} {Hello,~world}|
\ztool_set_to_wd:nn {40pt} {Hello,~world}\par

% scale one dimension
\dotfill\par
\ztool_scale_to_wd:nn {2em}{\fbox{AA}}\par
\ztool_scale_to_wd:nn {2em}{\fbox{AAA}}\par
\ztool_scale_to_wd:nn {2em}{\fbox{AAAAA}}\par
\ztool_scale_to_ht:nn {2.5em}{\fbox{\vbox{\hbox{A}}}}\quad
\ztool_scale_to_ht:nn {2.5em}{\fbox{\vbox{\hbox{A}\hbox{A}}}}\quad
\ztool_scale_to_ht:nn {2.5em}{\fbox{\vbox{\hbox{A}\hbox{A}\hbox{A}}}}\quad
\ztool_scale_to_ht:nn
{2.5em}{\fbox{\vbox{\hbox{A}\hbox{A}\hbox{A}\hbox{A}}}}\par
```

```
% box item align
\dotfill\par
\def\boxItemCmd#1{\textcolor{blue}{|#1|}}
\underline{
  \ztool_box_item_align:Nnnn \boxItemCmd{15em}{{Tom}{Amy}{Jennery}}{scatter}
}\par
\underline{
  \ztool_box_item_align:Nnnn \boxItemCmd{15em}{{Tom} {Amy}\ {Jennery}}{center}
}\par

% affine transform
\dotfill\par
\hcoffin_set:Nn \l_tmpa_coffin {\rule{2em}{2em}}
\coffin_typeset:Nnnnn \l_tmpa_coffin {l}{b}{0pt}{0pt}
\ztool_affine_transformation:Nnnnn \l_tmpa_coffin {1}{0}{.5}{1}
\coffin_typeset:Nnnnn \l_tmpa_coffin {l}{b}{0pt}{0pt}
\ExplSyntaxOff
```

7.54619pt 58.58835pt

Hello, world|Hello, world|Hello, world

|Tom| |Amy| |Jennery|

|Tom||Amy|| ||Jennery|

6 zdraw

这部分主要包含一些图像绘制命令, 这系列的命令并不依赖于 tikz 宏包, 它们的主要依赖项如下:

- LaTeX 2_ε 内置 picture 环境;
- pict2e : LaTeX 2_ε 内置 picture 环境的增强版, 提供了更好的绘图功能;
- bxeepic: 可以用于提供 dash line 支持, 目前还未引入该宏包.

zpic `\begin{zpic}[⟨key-value⟩] ⟨draw commands⟩ \end{zpic}`

New: 2025-05-13

此环境基于 LaTeX 2_ε 内置 picture 环境定义,

ztool/draw/picture/unit	unit = ⟨长度⟩	初始值: 1cm
ztool/draw/picture/width	width = ⟨浮点数⟩	初始值: 0
ztool/draw/picture/height	height = ⟨浮点数⟩	初始值: 0
ztool/draw/picture/xoffset	xoffset = ⟨浮点数⟩	初始值: 0
ztool/draw/picture/yoffset	yoffset = ⟨浮点数⟩	初始值: 0
ztool/draw/picture/opacity-color	opacity-color = ⟨颜色⟩	初始值: white

上述的 ⟨*opacity-color*⟩ 选项用于设置当前 zpic 环境中的 “透明” 色彩, 也就是和当前文档默认背景色相同的色彩; 所以可能会出现 ⟨*opacity-color*⟩ 覆盖到其它 object 上的情况.

\put `\put (⟨x, y⟩) {⟨content⟩}`

New: 2025-05-13

此命令即为 LaTeX 2_ε 内置 picture 环境中的 \put 命令. **注意**: 此命令需要在 picture 或 zpic 环境中使用.

\zline `\zline [⟨key-value⟩](⟨coor-1⟩)(⟨coor-2⟩)`

New: 2025-05-13

此命令用于绘制一条从 ⟨*coor-1*⟩ 到 ⟨*coor-2*⟩ 的线段, ⟨*key-value*⟩ 用于设置线条的属性, 可用选项请参见后续的 ⟨*parent=ztool/draw/picture/line*⟩.

ztool/../line/draw	draw = ⟨颜色⟩	初始值: black
ztool/../line/width	width = ⟨长度⟩	初始值: .4pt
ztool/../line/dash	dash = ⟨*true*|***false***⟩	初始值: false

上述 ⟨*width*⟩ 用于设置线条的宽度, ⟨*draw*⟩ 用于设置线条的颜色, ⟨*dash*⟩ 用于设置线条是否为虚线. **注意**: 目前 ⟨*dash*⟩ 选项还未适配, 处于不可用的状态.

\zvector `\zvector [⟨key-value⟩](⟨coor-1⟩)(⟨coor-2⟩)`

New: 2025-05-13

此命令用于绘制向量, 该向量的起点为 ⟨*coor-1*⟩, 终点为 ⟨*coor-2*⟩; ⟨*key-value*⟩ 用于设置该向量的外观属性, 其继承自 ⟨*parent=ztool/draw/picture/line*⟩, 其余的可用选项请参见后续 ⟨*parent=ztool/draw/picture/line/vector*⟩.

ztool/../vector/> > = ⟨***latex***|*pst*⟩ ..... 初始值: latex

此选项用于控制箭头的样式, 默认为 LaTeX 样式, 即 \ltxarrows; ⟨*pst*⟩, 即 PsTricks, 对应于 \pstarrows 命令.

\zdraw \zdraw [⟨*key-value*⟩](⟨*coor-1*⟩)...(⟨*coor-n*⟩);

New: 2025-05-13

此命令将绘制一条从点 ⟨*coor-1*⟩ 到点 ⟨*coor-n*⟩ 的折线段, ⟨*key-value*⟩ 继承自 ⟨*parent=ztool/draw/picture/line*⟩, 可以用于设置线条的属性, 额外可用的选项请参见后续的 ⟨*parent=ztool/draw/picture/zdraw*⟩.

注意: 此命令末尾的 ";" 是不能省略的, 否则会报错.

ztool/../zdraw/vector vector = ⟨***false***|*true*⟩ ..... 初始值: false
ztool/../zdraw/cycle cycle = ⟨***false***|*true*⟩ ..... 初始值: false
ztool/../zdraw/fill fill = ⟨***false***|*true*| 颜色⟩ ..... 初始值: false
ztool/../zdraw/shift shift = {⟨浮点数, 浮点数⟩} ..... 初始值: {0, 0}

当 ⟨*fill*⟩ 设置为 true 时, ⟨*cycle*⟩ 会自动设置为 true; ⟨*vector*⟩ 用于设置是否将每一个子线段替换为向量. ⟨*shift*⟩ 分别表示 x 和 y 方向的偏移量. **注意**: ⟨*shift*⟩ 选项中的 {} 不能省略.

\zarc \zarc[⟨*key-value*⟩](⟨浮点数, 浮点数⟩)

New: 2025-05-13

此命令用于绘制一个圆弧, (⟨浮点数, 浮点数⟩) 为其圆心, 默认绘制 $\frac{1}{4}$ 圆弧; ⟨*key-value*⟩ 继承自 ⟨*parent=ztool/draw/picture/line*⟩, 可以用于设置线条的属性, 额外可用的选项请参见后续的 ⟨*parent=ztool/draw/picture/zarc*⟩.

ztool/../zarc/radius radius = ⟨浮点数⟩ ..... 初始值: .5
ztool/../zarc/start start = ⟨浮点数⟩ ..... 初始值: 0
ztool/../zarc/end end = ⟨浮点数⟩ ..... 初始值: 90
ztool/../zarc/fill fill = ⟨***false***|*true*| 颜色⟩ ..... 初始值: false

⟨*start*⟩ 按照逆时针旋转到角度 ⟨*end*⟩ 结束; ⟨*radius*⟩ 为圆弧的半径; ⟨*fill*⟩ 用于设置圆弧的填充颜色.

\zcircle \zcircle[⟨*key-value*⟩](⟨浮点数, 浮点数⟩)

New: 2025-05-13

此命令基于上述的 \zarc 命令, 默认情况下将以 (⟨浮点数, 浮点数⟩) 为圆心绘制一个完整的圆; ⟨*key-value*⟩ 和上述的 \zrac 命令中的 ⟨*key-value*⟩ 选项相同,

\zrectangle \zrectangle[⟨*key-value*⟩](⟨*coor-1*⟩)(⟨*coor-2*⟩)

New: 2025-05-13

此命令用于绘制矩形, (⟨*coor-1*⟩) 和 (⟨*coor-2*⟩) 为矩形对角线的两个端点坐标; ⟨*key-value*⟩ 继承自 ⟨*parent=ztool/draw/picture/line*⟩, 其余的 ⟨*key-value*⟩ 请参见后续 ⟨*parent=ztool/draw/picture/zrectangle*⟩.

ztool/../zrectangle/arc arc = ⟨浮点数⟩ ..... 初始值: 0
ztool/../zrectangle/fill fill = ⟨***false***|*true*| 颜色⟩ ..... 初始值: false

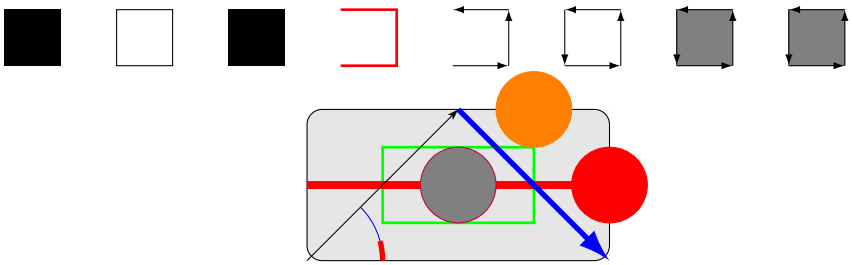
⟨*fill*⟩ 用于设置矩形的填充颜色, ⟨*arc*⟩ 用于设置矩形圆角对应的半径.

下面给出一些绘图示例, 方便读者理解上述绘图命令的基本使用方法:

例 5

```
\mbox{}\vskip2em
\begin{zpic}[unit=2em]
  \zdraw[fill, cycle] (0, 0)(1, 0)(1, 1)(0, 1);
  \zdraw[cycle, shift={2, 0}] (0, 0)(1, 0)(1, 1)(0, 1);
  \zdraw[fill, shift={4, 0}] (0, 0)(1, 0)(1, 1)(0, 1);
  \zdraw[draw=red, width=1pt, shift={6, 0}] (0, 0)(1, 0)(1, 1)(0, 1);
  \zdraw[vector, shift={8, 0}] (0, 0)(1, 0)(1, 1)(0, 1);
  \zdraw[vector, cycle, shift={10, 0}] (0, 0)(1, 0)(1, 1)(0, 1);
  \zdraw[vector, fill, shift={12, 0}] (0, 0)(1, 0)(1, 1)(0, 1);
  \zdraw[vector, cycle, fill, shift={14, 0}] (0, 0)(1, 0)(1, 1)(0, 1);
\end{zpic}

\vskip2cm
\begin{zpic}[unit=2cm, xoffset=2]
  % 1. rectangle
  \zrectangle[arc=.1, fill=gray!20](0, 0)(2, 1)
  \zrectangle[draw=green, width=1pt](.5, .25)(1.5, .75)
  % 2. line / vecter
  \zline[width=3pt, draw=red](0, .5)(2, .5)
  \zvector[>=pst](0, 0)(1, 1)
  \zvector[draw=blue, width=2pt](1, 1)(2, 0)
  % 3. arc / circle
  \zarc[draw=blue, end=45](0, 0) % fill=<empty>
  \zarc[draw=blue, width=2pt, end=15, fill=, draw=red](0, 0)
  \zcircle[radius=.25, fill, draw=purple](1, .5)
  \zcircle[radius=.25, fill=orange, draw=none](1.5, 1)
  \zcircle[radius=.25, fill=red, draw=](2, .5)
\end{zpic}
```



7 TODO

ztool 在将来也许会有改动, 这里列出部分将来可能会完善的功能 (☐ – 未完成; ☑ – 已完成; ☒ – 不考虑该功能):

- ☐ 重新实现 xsimverb 宏包中的 `\xsim_file_write_start:nn` 和 `\xsim_file_write_stop:` 命令, 使其和 ztool 宏包适配.
- ☑ 2025-05-22-已完成:修复 `\ztool_append_to_file:nn` 文件首行空行的问题.

8 zTool 源码

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% ztool.sty                                                                   %
%% Copyright 2024, 2025 Zongping Ding.                                         %
%                                                                              %
% This work may be distributed and/or modified under the conditions of the     %
% LaTeX Project Public License, either version 1.3 of this license or any     %
% later version.                                                               %
% The latest version of this license is in                                     %
%                    http://www.latex-project.org/lppl.txt                     %
% and version 1.3 or later is part of all distributions of LaTeX               %
% version 2005/12/01 or later.                                                 %
%                                                                              %
% This work has the LPPL maintenance status `maintained'.                      %
%                                                                              %
% The Current Maintainer of this work is Zongping Ding.                        %
%                                                                              %
% ztool.sty consists of the parts:                                             %
%                    l3sys-shell,                                              %
%                    file IO,                                                  %
%                    box manipulation,                                         %
%                    zdraw.                                                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\NeedsTeXFormat{LaTeX2e}
\ProvidesExplPackage{ztool}{2025/05/20}{1.0.1}{A~pre-release~tool~package~for~LaTeX}


%%%%%    l3keys intial patch begin   %%%%%
% 1. https://github.com/latex3/latex3/issues/1738
% 2. https://tex.stackexchange.com/q/742604/294585
\cs_set_protected:Npn \__keys_initialise:n #1
  {
    \exp_after:wN \__keys_find_key_module:wNN
      \l_keys_path_str \s__keys_stop
      \l_keys_key_tl \l_keys_key_str
    \tl_set_eq:NN \l_keys_key_tl \l_keys_key_str
    \tl_set:Nn \l_keys_value_tl {#1}
    \cs_if_exist:cTF { \c__keys_code_root_str \l_keys_path_str }
      {
        \str_clear:N \l__keys_inherit_str
        \__keys_execute:nn \l_keys_path_str {#1}
      }
      {
        \cs_if_exist:cT
          { \c__keys_inherit_root_str \__keys_parent:o \l_keys_path_str }
          { \__keys_execute_inherit: }
      }
  }
%%%%%    l3keys intial patch end   %%%%%


```

```
\clist_new:N \g__ztool_library_loaded_clist
\clist_gclear:N \g__ztool_library_loaded_clist
\bool_new:N \g__ztool_lib_user_load_dupulicate_bool
\bool_gset_false:N \g__ztool_lib_user_load_dupulicate_bool
\cs_new_nopar:Npn \__ztool_load_library:n #1
  {
    \clist_map_inline:nn {#1} {
      \clist_if_in:NnTF \g__ztool_library_loaded_clist {##1} {
        \msg_set:nnn {ztool} {library-loaded}
          {
            ztool~library~"##1"~already~loaded,ignored~loading.
            \msg_line_context:
          }
        \bool_if:NT \g__ztool_lib_user_load_dupulicate_bool
          {
            \msg_warning:nnn {ztool} {library-loaded} {##1}
          }
      }{
        \file_if_exist:nTF {library/ztool.library.##1.tex}{
          \clist_gput_right:Nn \g__ztool_library_loaded_clist {##1}
          \makeatletter\file_input:n {library/ztool.library.##1.tex}
        }{
          \msg_set:nnn {ztool} {library-not-found} {ztool~library~`##1'~not~found.}
          \msg_error:nnn {ztool} {library-not-found} {##1}
        }
      }
    }
  }
\NewDocumentCommand\ztoolloadlib{m}
  {
    \__ztool_load_library:n {#1}
    \bool_gset_true:N \g__ztool_lib_user_load_dupulicate_bool
    \ExplSyntaxOff
  }
\keys_define:nn { ztool }
  {
    shell-escape .code:n    = { \__ztool_load_library:n {shell-escape} },
    file-io      .code:n    = { \__ztool_load_library:n {file-io} },
    box          .code:n    = { \__ztool_load_library:n {box} },
    zdraw        .code:n    = { \__ztool_load_library:n {zdraw} },
  }
\ProcessKeyOptions [ ztool ]
```

```
\ProvidesExplFile{ztool.library.shell-escape.tex}{2025/05/21}{1.0.1}{shell-escape~library~for~ztool}


% ==> l3sys-shell tool
% NOTE: Copy of the original 'l3sys-shell' + some modifications
% windows path handle
\cs_new:Npn \ztool_sys_path_to_win:N #1
  {
    \quark_if_nil:NF #1 {
      \token_if_eq_meaning:NNTF #1 /
        { \c_backslash_str }
        {#1}
      \ztool_sys_path_to_win:N
    }
  }
\cs_new:Npn \ztool_sys_path_to_win:w #1 ~ #2 \q_stop
  {
    \ztool_sys_path_to_win:N #1 \q_nil
    \tl_if_empty:nF {#2}
      {
        \c_space_tl
        \__sys_path_to_win:w #2 \q_stop
      }
  }
\cs_new:Npn \ztool_sys_path_to_win:n #1
  {
    \exp_after:wN \ztool_sys_path_to_win:w \tl_to_str:n {#1} ~ \q_stop
  }
% respective commands
\cs_new_protected:Npn \ztool_shell_escape:n #1
  {
    \sys_if_shell_unrestricted:T
      { \sys_shell_now:n {#1} }
  }
\cs_generate_variant:Nn \ztool_shell_escape:n {e}
\cs_new_protected:Npe \ztool_shell_mkdir:n #1
  {
    \ztool_shell_escape:e {
      \sys_if_platform_unix:T
        {mkdir~-p~\exp_not:N \tl_to_str:n {#1}}
      \sys_if_platform_windows:T
        {mkdir~ \exp_not:N \ztool_sys_path_to_win:n {#1}}
    }
  }
\cs_new_protected:Npe \ztool_shell_cp:nn #1#2
  {
    \ztool_shell_escape:e {
      \sys_if_platform_unix:T
        {
          cp~-f~ \exp_not:N \tl_to_str:n {#1} ~
            \exp_not:N \tl_to_str:n {#2}
```

```
        }
      \sys_if_platform_windows:T
        {% can NOT use wildcards in CMD
          copy~/y~ \exp_not:N \ztool_sys_path_to_win:n {#1} ~
            \exp_not:N \ztool_sys_path_to_win:n {#2}
        }
    }
  }
\cs_new_protected:Npe \ztool_shell_mv:nn #1#2
  {
    \ztool_shell_escape:e {
      \sys_if_platform_unix:T
        {
          mv~ \exp_not:N \tl_to_str:n {#1} ~
            \exp_not:N \tl_to_str:n {#2}
        }
      \sys_if_platform_windows:T
        {
          copy~/y~ \exp_not:N \ztool_sys_path_to_win:n {#1} ~
            \exp_not:N \ztool_sys_path_to_win:n {#2}
            \token_to_str:N & \token_to_str:N &
            del~/f~/q~\exp_not:N \ztool_sys_path_to_win:n {#1}
        }
    }
  }
\cs_new_protected:Npe \ztool_shell_rm:n #1
  {
    \ztool_shell_escape:e {
      \sys_if_platform_unix:T
        { rm~-f~ \exp_not:N \tl_to_str:n {#1}  }
      \sys_if_platform_windows:T
        { del~/f~/q~ \exp_not:N \ztool_sys_path_to_win:n {#1} }
    }
  }
\cs_new_protected:Npe \ztool_shell_rmdir:n #1
  {
    \ztool_shell_mkdir:n {#1}
    \ztool_shell_escape:e {
      \sys_if_platform_unix:T
        { rm~-rf~ \exp_not:N \tl_to_str:n {#1} }
      \sys_if_platform_windows:T
        { rmdir~/s~/q~ \exp_not:N \ztool_sys_path_to_win:n {#1} }
    }
  }
\tl_new:N \l__ztool_shell_tmp_tl
\cs_new_protected:Npe \ztool_get_shell_pwd:N #1
  {
    \exp_not:N \sys_get_shell:nnN
      {
        \sys_if_platform_unix:T { pwd }
        \sys_if_platform_windows:T { cd }
      }{
```

```
        \char_set_catcode_other:N \exp_not:N \\
        \char_set_catcode_other:N \exp_not:N \#
        \char_set_catcode_other:N \exp_not:N \~
        \char_set_catcode_other:N \exp_not:N \%
        \char_set_catcode_space:N \exp_not:N \ %
        \tex_endlinechar:D -1 \scan_stop:
      }
    \exp_not:N \l__ztool_shell_tmp_tl
    \str_set:NV #1 \exp_not:N \l__ztool_shell_tmp_tl
  }
\cs_new_protected:Npe \ztool_shell_split_ls:nN #1#2
  {
    \exp_not:N \sys_get_shell:nnN
      {
        \sys_if_platform_unix:T { ls~-1~ #1 }
        \sys_if_platform_windows:T { dir~/b~ #1 }
      }{
        \ExplSyntaxOff
        \char_set_catcode_other:N \exp_not:N \\
        \char_set_catcode_other:N \exp_not:N \#
        \char_set_catcode_other:N \exp_not:N \~
        \char_set_catcode_other:N \exp_not:N \%
        \char_set_catcode_other:n { 13 }
      }
      \exp_not:N \l__ztool_shell_tmp_tl
    \str_set:NV \exp_not:N \l__sys_tmp_tl \exp_not:N \l__sys_tmp_tl
    \seq_set_split:NnV #2
      { \char_generate:nn { `\^^M } { 12 } }
      \exp_not:N \l__ztool_shell_tmp_tl
    \seq_pop_right:NN #2 \exp_not:N \l__sys_tmp_tl
  }
\cs_generate_variant:Nn \ztool_shell_mkdir:n {e}
\cs_generate_variant:Nn \ztool_shell_cp:nn { ee, ne, en }
\cs_generate_variant:Nn \ztool_shell_mv:nn { ee, ne, en }
\cs_generate_variant:Nn \ztool_shell_rm:n { e, f, o }
\cs_generate_variant:Nn \ztool_shell_rmdir:nn { e, f, o }
\cs_generate_variant:Nn \ztool_get_shell_pwd:N {c}
\cs_generate_variant:Nn \ztool_shell_split_ls:nN {nc}
```

```
\ProvidesExplFile{ztool.library.file-io.tex}{2025/05/27}{1.0.1}{file-io~library~for~ztool}


% ==> file IO operations
% 1. create a new file
% 2. append to a file
% 3. read from file / write to file
\ior_new:N \g_ztool_file_read_ior
\ior_new:N \g_ztool_file_append_ior
\iow_new:N \g_ztool_file_append_iow
\tl_new:N \l_ztool_current_line
\str_clear:N \l_ztool_file_ori_content_str
\seq_new:N \l_ztool_file_seq
\seq_new:N \l__ztool_tmp_seq
\cs_generate_variant:Nn \seq_use:Nn { Ne }

\cs_new_protected:Npn \ztool_read_file_as_seq:nnN #1#2#3
  {% #1: bool(True to keep spaces, False to trim); #2: file name; #3: seq
    \seq_clear:N #3
    \file_if_exist:nT {#2}
      {
        \ior_open:Nn \g_ztool_file_read_ior {#2}
        \ior_map_inline:Nn \g_ztool_file_read_ior
          {
            \bool_if:nTF {#1}
              { \seq_put_right:Nn #3 {##1} }
              { \seq_put_right:Ne #3 {\tl_trim_spaces:n {##1}} }
          }
        \ior_close:N \g_ztool_file_read_ior
      }
  }
\cs_new_protected:Npn \ztool_gread_file_as_seq:nnN #1#2#3
  {% #1: bool(True to keep spaces, False to trim); #2: file name; #3: seq
    \seq_gclear:N #3
    \file_if_exist:nT {#2}
      {
        \ior_open:Nn \g_ztool_file_read_ior {#2}
        \ior_map_inline:Nn \g_ztool_file_read_ior
          {
            \bool_if:nTF {#1}
              { \seq_gput_right:Nn #3 {##1} }
              { \seq_gput_right:Ne #3 {\tl_trim_spaces:n {##1}} }
          }
        \ior_close:N \g_ztool_file_read_ior
      }
  }
\cs_generate_variant:Nn \ztool_read_file_as_seq:nnN { ne, nnc, nec }
\cs_generate_variant:Nn \ztool_gread_file_as_seq:nnN { ne, nnc, nec }

\cs_new_protected:Npn \ztool_file_new:nn #1#2
  {% #1: \c_true_bool to allow overwrite; #2: file name
```

```
    \bool_if:nT {#1}
      {
        \iow_open:Nn \g_ztool_file_append_iow {#2}
        \iow_close:N \g_ztool_file_append_iow
      }
  }
\cs_new_protected:Npn \ztool_append_to_file:nn #1#2
  {% #1: file name; #2: content
    \seq_clear:N \l_ztool_file_seq
    \file_if_exist:nF {#1}{ \ztool_file_new:nn {\c_true_bool}{#1} }
    \ior_open:Nn \g_ztool_file_append_ior {#1}
    \ior_str_map_inline:Nn \g_ztool_file_append_ior
      {
        \seq_put_right:Nn \l_ztool_file_seq
          { ##1 }
      }
    \iow_open:Nn \g_ztool_file_append_iow {#1}
    \seq_if_empty:NF \l_ztool_file_seq
      {
        \iow_now:Ne \g_ztool_file_append_iow
          { \seq_use:Ne \l_ztool_file_seq {\iow_newline:} }
      }
    \iow_now:Ne \g_ztool_file_append_iow {#2}
    \iow_close:N \g_ztool_file_append_iow
  }
\cs_generate_variant:Nn \ztool_append_to_file:nn { no, nf, ne, ee }

\cs_new_protected:Npn \ztool_write_seq_to_file:nNn #1#2#3
  {% #1:bool; #2:seq; #3:file name
    \seq_clear:N \l__ztool_tmp_seq
    \bool_if:nTF { #1 }
      {
        \seq_set_eq:NN \l_ztool_file_seq #2
      }{
        \ztool_read_file_as_seq:nnN
          { \c_true_bool }{ #3 }
          \l__ztool_tmp_seq
        \seq_concat:NNN \l_ztool_file_seq \l__ztool_tmp_seq #2
      }
    \file_if_exist:nF {#3}{ \ztool_file_new:nn {\c_true_bool}{#3} }
    \iow_open:Nn \g_tmpa_iow { #3 }
    \seq_if_empty:NF \l_ztool_file_seq
      {
        \iow_now:Ne \g_tmpa_iow
          { \seq_use:Ne \l_ztool_file_seq { \iow_newline: } }
      }
    \iow_close:N \g_tmpa_iow
  }
\cs_generate_variant:Nn \ztool_write_seq_to_file:nNn { nNe, nNV }

\cs_new_protected:Npn \ztool_replace_file_line:nnn #1#2#3
  {% #1:file name; #2:line index; #3:replacement
```

```
    \seq_clear:N \l_ztool_file_seq
    \file_if_exist:nT {#1}{
      \ior_open:Nn \g_ztool_file_read_ior {#1}
      \ior_str_map_inline:Nn \g_ztool_file_read_ior
        {
          \seq_put_right:Nn \l_ztool_file_seq {##1}
        }
      \ior_close:N \g_ztool_file_read_ior
      \seq_set_item:Nnn \l_ztool_file_seq {#2}
        { #3 }
      \iow_open:Nn \g_ztool_file_append_iow {#1}
      \seq_if_empty:NF \l_ztool_file_seq
        {
          \iow_now:Ne \g_ztool_file_append_iow
            { \seq_use:Ne \l_ztool_file_seq {\iow_newline:} }
        }
      \iow_close:N \g_ztool_file_append_iow
    }
  }
\cs_generate_variant:Nn \seq_set_item:Nnn { Nne }
\cs_generate_variant:Nn \ztool_replace_file_line:nnn { e, ene, eee }
\cs_new_protected:Npn \ztool_insert_to_file:nnn #1#2#3
  {% #1:file name; #2:line index; #3:content
    \seq_clear:N \l_ztool_file_seq
    \file_if_exist:nT {#1}{
      \ior_open:Nn \g_ztool_file_read_ior {#1}
      \ior_str_map_inline:Nn \g_ztool_file_read_ior
        {
          \seq_put_right:Nn \l_ztool_file_seq {##1}
        }
      \ior_close:N \g_ztool_file_read_ior
      \tl_set:No \l_ztool_current_line
        { \seq_item:Nn \l_ztool_file_seq {#2} }
      \seq_set_item:Nne \l_ztool_file_seq {#2}
        { #3\iow_newline:\l_ztool_current_line }
      \iow_open:Nn \g_ztool_file_append_iow {#1}
      \iow_now:Ne \g_ztool_file_append_iow
        { \seq_use:Ne \l_ztool_file_seq {\iow_newline:} }
      \iow_close:N \g_ztool_file_append_iow
    }
  }
\cs_generate_variant:Nn \ztool_insert_to_file:nn { ne, nf, ee }
```

```
\ProvidesExplFile{ztool.library.box.tex}{2025/05/21}{1.0.1}{box~library~for~ztool}


% ==> box manipulation tool
\cs_set:Nn \__ztool_leave_vmode:
  { \ifvmode \leavevmode \fi }
% catch box dimension
\box_new:N \l_ztool_measure_box
\cs_new:Npn \ztool_box_set_to:NNn #1#2#3 {
  \hbox_set:Nn \l_ztool_measure_box {#3}
  \dim_set:Nn #2 {#1 \l_ztool_measure_box}
  \box_set_eq:NN \l_ztool_measure_box \c_empty_box
}
\cs_new:Npn \ztool_box_gset_to:NNn #1#2#3 {
  \hbox_set:Nn \l_ztool_measure_box {#3}
  \dim_gset:Nn #2 {#1 \l_ztool_measure_box}
  \box_set_eq:NN \l_ztool_measure_box \c_empty_box
}
\cs_new:Npn \ztool_get_ht:Nn
  { \ztool_box_set_to:NNn \box_ht:N }
\cs_new:Npn \ztool_get_ht_plus_dp:Nn
  { \ztool_box_set_to:NNn \box_ht_plus_dp:N }
\cs_new:Npn \ztool_get_wd:Nn
  { \ztool_box_set_to:NNn \box_wd:N }
\cs_new:Npn \ztool_get_dp:Nn
  { \ztool_box_set_to:NNn \box_dp:N }
\cs_new:Npn \ztool_gget_ht:Nn
  { \ztool_box_gset_to:NNn \box_ht:N }
\cs_new:Npn \ztool_gget_wd:Nn
  { \ztool_box_gset_to:NNn \box_wd:N }
\cs_new:Npn \ztool_gget_dp:Nn
  { \ztool_box_gset_to:NNn \box_dp:N }
\cs_generate_variant:Nn \ztool_get_ht:Nn  { Ne, ce }
\cs_generate_variant:Nn \ztool_get_ht_plus_dp:Nn  { Ne, ce }
\cs_generate_variant:Nn \ztool_get_wd:Nn  { Ne, ce }
\cs_generate_variant:Nn \ztool_gget_ht:Nn { Ne, ce }
\cs_generate_variant:Nn \ztool_gget_wd:Nn { Ne, ce }


%% modify box content
% 1. auto scale and rotate (smaller of two)
\cs_new_protected:Npn \ztool_autoset_to_wd_and_ht:nnn #1#2#3
  {% #1:width; #2:height; #3:object
    \hbox_set:Nn \l_tmpa_box {#3}
    \box_autosize_to_wd_and_ht:Nnn \l_tmpa_box {#1}{#2}
    \__ztool_leave_vmode:
    \box_use:N \l_tmpa_box
  }
\cs_new_protected:Npn \ztool_rotate:nn #1#2
  {% #1:angle; #2:object
    \hbox_set:Nn \l_tmpa_box {#2}
    \box_rotate:Nn \l_tmpa_box {#1}
```

```
    \__ztool_leave_vmode:
    \box_use:N \l_tmpa_box
  }
\cs_generate_variant:Nn \ztool_rotate:nn { e, ne, ee }
\cs_generate_variant:Nn \ztool_autoset_to_wd_and_ht:nnn { nne, een, eee }

% 2. width/height scale to same time
\cs_new_protected:Npn \ztool_set_to_wd:nn #1#2
  {% #1:width; #2:object
    \hbox_set:Nn \l_tmpa_box {#2}
    \box_resize_to_wd:Nn \l_tmpa_box {#1}
    \__ztool_leave_vmode:
    \box_use:N \l_tmpa_box
  }
\cs_new_protected:Npn \ztool_set_to_ht:nn #1#2
  {% #1:height; #2:object
    \hbox_set:Nn \l_tmpa_box {#2}
    \box_resize_to_ht:Nn \l_tmpa_box {#1}
    \__ztool_leave_vmode:
    \box_use:N \l_tmpa_box
  }
\cs_generate_variant:Nn \ztool_set_to_wd:nn { e, ne, ee }
\cs_generate_variant:Nn \ztool_set_to_ht:nn { e, ne, ee }

% 3. only scale one dimension
% NOTE: if boxwdcontent ≤ given dim, no manipulation
\cs_new_protected:Npn \ztool_scale_to_wd:nn #1#2
  {
    \hbox_set:Nn \l_tmpa_box {#2}
    \dim_set:Nn \l_tmpa_dim { \box_wd:N \l_tmpa_box }
    \fp_set:Nn \l_tmpa_fp
      {
        \fp_eval:n { min(1, \dim_ratio:nn {#1}{\l_tmpa_dim}) }
      }
    \box_scale:Nnn \l_tmpa_box {\l_tmpa_fp}{1}
    \__ztool_leave_vmode:
    \box_use:N \l_tmpa_box
  }
\cs_new_protected:Npn \ztool_scale_to_ht:nn #1#2
  {% take depth into consideration
    \hbox_set:Nn \l_tmpa_box {#2}
    \dim_set:Nn \l_tmpa_dim { \box_ht_plus_dp:N \l_tmpa_box }
    \fp_set:Nn \l_tmpa_fp
      {
        \fp_eval:n { min(1, \dim_ratio:nn {#1}{\l_tmpa_dim}) }
      }
    \box_scale:Nnn \l_tmpa_box {1}{\l_tmpa_fp}
    \__ztool_leave_vmode:
    \box_use:N \l_tmpa_box
  }
\cs_new_protected:Npn \ztool_scale_to_wd_and_ht:nnn #1#2#3
  {% take depth into consideration
```

```
    \hbox_set:Nn \l_tmpa_box {#3}
    \dim_set:Nn \l_tmpa_dim { \box_wd:N \l_tmpa_box }
    \dim_set:Nn \l_tmpb_dim { \box_ht_plus_dp:N \l_tmpa_box }
    \fp_set:Nn \l_tmpa_fp
      {
        \fp_eval:n { min(1, \dim_ratio:nn {#1}{\l_tmpa_dim}) }
      }
    \fp_set:Nn \l_tmpb_fp
      {
        \fp_eval:n { min(1, \dim_ratio:nn {#2}{\l_tmpb_dim}) }
      }
    \box_scale:Nnn \l_tmpa_box {\l_tmpa_fp}{\l_tmpb_fp}
    \__ztool_leave_vmode:
    \box_use:N \l_tmpa_box
  }
\cs_generate_variant:Nn \ztool_scale_to_wd:nn { e, ne, ee }
\cs_generate_variant:Nn \ztool_scale_to_ht:nn { e, ne, ee }
\cs_generate_variant:Nn \ztool_scale_to_wd_and_ht:nnn { nne, nno, eee }


%% box content align
\seq_new:N \l__ztool_boxitem_seq
\cs_set_protected:Npn \ztool_box_item_align:Nnnn #1#2#3#4
  {% #1:cmd, #2:width, #3:object, #4:align format(left, right, scatter, center)
    \hb@xt@#2{
      \tl_map_inline:nn {#3}
        {
          \seq_put_right:No \l__ztool_boxitem_seq {\exp_not:N #1{##1}}
        }
      \str_case:nnF { #4 }
        {
          {  left  }{ \seq_use:Nn \l__ztool_boxitem_seq {}\hfill }
          {  right }{ \hfill\seq_use:Nn \l__ztool_boxitem_seq {} }
          { scatter}{ \seq_use:Nn \l__ztool_boxitem_seq {\hfill} }
          { center }{ \hfill\seq_use:Nn \l__ztool_boxitem_seq {}\hfill }
          { tower  }
            {
              \edef\seq@count{\seq_count:N \l__ztool_boxitem_seq}
              \seq_map_indexed_inline:Nn \l__ztool_boxitem_seq
                {% ##1: index, ##2: content
                  %% Method II: plain
                  \edef\item@width{\dim_eval:n {#2/(\seq@count+1)}}
                  \hskip\item@width\clap{##2}
                }\hskip\item@width\hss
            }
          { custom }
            {
              \def\total@width{#2}
              \def\align@cmd{#1}
              \def\align@object{#3}
              \def\align@format{#4}
              \tl_use:N \l__ztex_boxitem_align_custom_tl
```

```
            }
        }{\relax}
    }
    \seq_clear:N \l__ztool_boxitem_seq
  }
\cs_generate_variant:Nn \ztool_box_item_align:Nnnn { c, Nnno, cnno, Nne, Nnee }


%% affine transformation
% REF:
% 1. https://math.stackexchange.com/a/3521141/1235323
% 2. https://math.stackexchange.com/a/281087/1235323
\cs_new:Npn \ztool_fp_to_rad:n #1
  { \fp_eval:n {#1/pi*180} }
\cs_new:Npn \ztool_matrix_det:nnnn #1#2#3#4
  {
    \fp_eval:n { #1*#4 - #2*#3 }
  }
% (translation) + x-scale + y-scale + rotate
\fp_new:N \g_affine_precision_fp
\fp_set:Nn \g_affine_precision_fp {0.0001}
\fp_new:N \l__affine_@@_a_fp
\fp_new:N \l__affine_@@_b_fp
\fp_new:N \l__affine_@@_c_fp
\fp_new:N \l__affine_@@_d_fp
\msg_set:nnn { ztool }{affine-det-zero}
  {
    current~determination~of~the~affine~transformation~
    matrix~equals~to~zero,~give~up~this~transformation
  }

\coffin_new:N \l__affine_trans_coffin
\cs_generate_variant:Nn \coffin_typeset:Nnnnn { Nxxxx }
\cs_new:Npn \ztool_affine_transformation:Nnnnn #1#2#3#4#5
  {% #1:box; #2:a_11; #3:a_21; #4:a_12; #5:a_22.
    \fp_compare:nNnT
      { abs(\ztool_matrix_det:nnnn {#2}{#3}{#4}{#5}) }
        < { \g_affine_precision_fp }
      { \prg_map_break:Nn \l__affine_matrix_det_zero
        { \msg_warning:nn { ztool }{affine-det-zero} }}
    \fp_set:Nn \l__affine_@@_a_fp {#2}
    \fp_set:Nn \l__affine_@@_b_fp {#3}
    \fp_set:Nn \l__affine_@@_c_fp {#4}
    \fp_set:Nn \l__affine_@@_d_fp {#5}
    \__box_affine_transform:N #1
    \prg_break_point:Nn \l__affine_matrix_det_zero { }
    \coffin_typeset:Nxxxx \l__affine_trans_coffin
      { \l__ztool_affine_pole_a_tl }{ \l__ztool_affine_pole_b_tl }
      { \l__ztool_affine_xoffset_dim }{ \l__ztool_affine_yoffset_dim }
  }
\cs_generate_variant:Nn \ztool_affine_transformation:Nnnnn { Neeee, cnnnn, ceeee }
\cs_new:Npn \__box_affine_transform:N #1
```

```
  {
    % transform debug
    \bool_if:NT \g_ztool_affine_debug_bool
      {
        \noindent\dotfill\[\begin{bmatrix}
          \fp_use:N \l__affine_@@_a_fp & \fp_use:N \l__affine_@@_c_fp\\
          \fp_use:N \l__affine_@@_b_fp & \fp_use:N \l__affine_@@_d_fp
        \end{bmatrix}\]
      }
    % get affine parameters
    \__affine_trans_get_sx:
    \__affine_trans_get_theta:
    \__affine_trans_get_sy:
    \__affine_trans_get_Sx:
    \__affine_trans_get_Sy:
    \__affine_trans_get_phi:
    \__affine_trans_get_omega:
    % start transform box/coffin
    \coffin_scale:Nnn #1
      { \l__box_affine_sx_fp }
      { \l__box_affine_sy_fp }
    \coffin_rotate:Nn #1
      { \ztool_fp_to_rad:n {\l__box_affine_omega_fp} }
    \coffin_scale:Nnn #1
      { \l__box_affine_Sx_fp }
      { \l__box_affine_Sy_fp }
    \coffin_rotate:Nn #1
      { \ztool_fp_to_rad:n {\l__box_affine_phi_fp} }
    \coffin_rotate:Nn #1
      { \ztool_fp_to_rad:n {\l__box_affine_theta_fp} }
  }
\keys_define:nn { ztool / affine }
  {
    debug   .bool_gset:N = \g_ztool_affine_debug_bool,
    debug   .initial:n   = false,
    debug   .default:n   = true,
    pole-1  .tl_set:N    = \l__ztool_affine_pole_a_tl,
    pole-2  .tl_set:N    = \l__ztool_affine_pole_b_tl,
    pole-1  .initial:n   = { l },
    pole-2  .initial:n   = { b },
    xoffset .dim_set:N   = \l__ztool_affine_xoffset_dim,
    yoffset .dim_set:N   = \l__ztool_affine_yoffset_dim,
    xoffset .initial:n   = { 0pt },
    yoffset .initial:n   = { 0pt },
  }
\NewDocumentCommand{\ztoolboxaffine}{O{}m>{\SplitList{,}}m}
  {% #1:key-value; #2:content; #3:matrix.
    \group_begin:
      \keys_set:nn { ztool / affine } {#1}
      \hcoffin_set:Nn \l__affine_trans_coffin {#2}
      \ztool_affine_transformation:Nnnnn \l__affine_trans_coffin #3
    \group_end:
```

```
  }
% internal affine transform functions
\cs_new:Nn \__ztool_affine_debug_fp:N
  {
    \bool_if:NTF \g_ztool_affine_debug_bool
      { \string #1 % \show #1
        ~=~\fp_use:N #1\\
      }{ \relax }
  }
\fp_new:N \l__box_affine_sx_fp
\cs_new:Nn \__affine_trans_get_sx:
  {
    \fp_set:Nn \l__box_affine_sx_fp
      { \fp_eval:n {sqrt(\l__affine_@@_a_fp^2 + \l__affine_@@_b_fp^2)} }
    \__ztool_affine_debug_fp:N \l__box_affine_sx_fp
  }
\fp_new:N \l__box_affine_theta_fp
\cs_new:Nn \__affine_trans_get_theta:
  {
    \fp_set:Nn \l__box_affine_theta_fp
      { \fp_eval:n {atan(\l__affine_@@_b_fp/\l__affine_@@_a_fp)} }
    \__ztool_affine_debug_fp:N \l__box_affine_theta_fp
  }
\fp_new:N \l__box_affine_msy_fp
\cs_new:Nn \__affine_trans_get_msy:
  {
    \fp_set:Nn \l__box_affine_msy_fp
      { \fp_eval:n {
        \l__affine_@@_c_fp*cos(\l__box_affine_theta_fp)
        +
        \l__affine_@@_d_fp*sin(\l__box_affine_theta_fp)
      } }
    \__ztool_affine_debug_fp:N \l__box_affine_msy_fp
  }
\fp_new:N \l__box_affine_sy_fp
\cs_new:Nn \__affine_trans_get_sy:
  {
    \__affine_trans_get_msy:
    \bool_if:nTF
      {
        \fp_compare_p:nNn { abs(sin(\l__box_affine_theta_fp)) }
          < {\c_zero_fp + \g_affine_precision_fp}
      }{
        \fp_set:Nn \l__box_affine_sy_fp
          {
            ( \l__affine_@@_d_fp -
            \l__box_affine_msy_fp*sin(\l__box_affine_theta_fp) )
            / cos(\l__box_affine_theta_fp)
          }
      }{
        \fp_set:Nn \l__box_affine_sy_fp
          {
```

```
            ( \l__box_affine_msy_fp*cos(\l__box_affine_theta_fp) -
            \l__affine_@@_c_fp )
            / sin(\l__box_affine_theta_fp)
          }
      }
    \__ztool_affine_debug_fp:N \l__box_affine_sy_fp
  }
\fp_new:N \l__box_affine_m_fp
\cs_new:Nn \__affine_trans_get_m:
  {
    \fp_set:Nn \l__box_affine_m_fp
      { \l__box_affine_msy_fp / \l__box_affine_sy_fp }
    \__ztool_affine_debug_fp:N \l__box_affine_m_fp
  }
\fp_new:N \l__box_affine_Sx_fp
\fp_new:N \l__box_affine_Sy_fp
\cs_new:Nn \__affine_trans_get_Sx:
  {
    \__affine_trans_get_m:
    \fp_set:Nn \l__box_affine_Sx_fp
      { sqrt(\l__box_affine_m_fp^2/4 + 1) - \l__box_affine_m_fp/2 }
    \__ztool_affine_debug_fp:N \l__box_affine_Sx_fp
  }
\cs_new:Nn \__affine_trans_get_Sy:
  {
    \fp_set:Nn \l__box_affine_Sy_fp
      { sqrt(\l__box_affine_m_fp^2/4 + 1) + \l__box_affine_m_fp/2 }
    \__ztool_affine_debug_fp:N \l__box_affine_Sy_fp
  }
\fp_new:N \l__box_affine_phi_fp
\fp_new:N \l__box_affine_omega_fp
\cs_new:Nn \__affine_trans_get_phi:
  {
    \fp_set:Nn \l__box_affine_phi_fp
      { -pi/4 - 1/2*atan(\l__box_affine_m_fp/2) }
    \__ztool_affine_debug_fp:N \l__box_affine_phi_fp
  }
\cs_new:Nn \__affine_trans_get_omega:
  {
    \fp_set:Nn \l__box_affine_omega_fp
      { pi/4 - 1/2*atan(\l__box_affine_m_fp/2) }
    \__ztool_affine_debug_fp:N \l__box_affine_omega_fp
  }
```

```
\ProvidesExplFile{ztool.library.zdraw.tex}{2025/05/21}{1.0.1}{zdraw~library~for~ztool}


% ==> ztool draw (based on package 'pict2e' and 'picture' env)
\RequirePackage{pict2e}
\cs_new:Npn \_@@_begin_picture:nnnn #1#2#3#4
  { \begin{picture}
      (\fp_eval:n {#1}, \fp_eval:n {#2})
      (\fp_eval:n {-#3}, \fp_eval:n {-#4}) }
\cs_new:Nn \_@@_end_picture:
  { \end{picture} }
\cs_new:Npn \__@@_pic_put:nnn #1#2#3
  { \put(\fp_eval:n {#1}, \fp_eval:n {#2}){ #3 } }
\cs_generate_variant:Nn \_@@_begin_picture:nnnn { VVVV, eeee }
\cs_generate_variant:Nn \__@@_pic_put:nnn { VVV, een }

% picture environment alias
\keys_define:nn { ztool / draw / picture }
  {
    unit     .dim_set:N = \l__pic_unit_dim,
    unit     .initial:n = { 1cm },
    width    .fp_set:N  = \l__pic_width_fp,
    width    .initial:n = 0,
    height   .fp_set:N  = \l__pic_height_fp,
    height   .initial:n = 0,
    xoffset  .fp_set:N  = \l__pic_xoffset_fp,
    xoffset  .initial:n = 0,
    yoffset  .fp_set:N  = \l__pic_yoffset_fp,
    yoffset  .initial:n = 0,
    opacity-color .tl_set:N  = \l__pic_opacity_color_tl,
    opacity-color .initial:n = { white },
  }
\NewDocumentEnvironment{zpic}{O{}}
  {
    \group_begin:
    \keys_set:nn { ztool / draw / picture } {#1}
    \setlength\unitlength{ \l__pic_unit_dim }
    \_@@_begin_picture:VVVV
      \l__pic_width_fp  \l__pic_height_fp
      \l__pic_xoffset_fp\l__pic_yoffset_fp
  }{
    \_@@_end_picture:
    \group_end:
  }


% picture commands alias
\cs_new:Npn \__coor_st:n #1
  { \clist_item:nn {#1}{1} }
\cs_new:Npn \__coor_nd:n #1
  { \clist_item:nn {#1}{2} }
```

```
\cs_new:Npn \__coor_rd:n #1#2
  { \clist_item:nn {#1}{3} }
\cs_new:Npn \__coor_st_nd:n #1
  {
    {\clist_item:nn {#1}{1}}
    {\clist_item:nn {#1}{2}}
  }
\cs_new:Npn \__coor_st_nd_rd:n #1
  {
    {\clist_item:nn {#1}{1}}
    {\clist_item:nn {#1}{2}}
    {\clist_item:nn {#1}{3}}
  }
\cs_generate_variant:Nn \__coor_st:n { V, e }
\cs_generate_variant:Nn \__coor_nd:n { V, e }
\cs_generate_variant:Nn \__coor_rd:n { V, e }
\cs_generate_variant:Nn \__coor_st_nd:n { V, e }
\cs_generate_variant:Nn \__coor_st_nd_rd:n { V, e }

\bool_new:N \l__ztool_invalid_color_bool
\cs_new:Npn \__color_safe_use:n #1
  {
    \__color_if_valid:nT {#1}
      { \color{#1} }
  }
\prg_new_conditional:Npnn \__color_if_valid:n #1 {p, T, F, TF}
  {
    \def\ztool@targer@color{#1}
    \def\ztool@color@none{none}
    \bool_if:eTF
      {
        \tl_if_empty_p:e {#1} ||
        \tl_if_eq_p:NN \ztool@color@none \ztool@targer@color
      }{ \prg_return_false: }
      { \prg_return_true: }
  }
\prg_generate_conditional_variant:Nnn \__color_if_valid:n
  { V, e }{ p, T, F, TF }
\cs_generate_variant:Nn \__color_safe_use:n { V, e }


% --> line/vector
\fp_new:N \l__draw_vector_slope_fp
\fp_new:N \l__draw_vector_normal_fp
\fp_new:N \l__draw_vector_xsep_fp
\cs_new:Npn \__@@_pic_line:nnn #1#2#3
  {% #1:x; #2:y; #3:x-distance NOT the length
    \line(\fp_eval:n {#1}, \fp_eval:n {#2})
      { \fp_eval:n {#3} }
  }
\cs_new:Npn \__@@_pic_vector:nnn #1#2#3
  {% #1:x; #2:y; #3:x-distance NOT the length
```

```
    \vector(\fp_eval:n {#1}, \fp_eval:n {#2})
      { \fp_eval:n {#3} }
  }
\keys_define:nn { ztool / draw / picture / line }
  {
    draw   .tl_set:N  = \l__pic_line_draw_color_tl,
    draw   .initial:n = { black },
    % color  .meta:n    =  draw = #1 , % alias for 'draw'
    width  .dim_set:N = \l__pic_line_width_dim,
    width  .initial:n = { .4pt },
    dash   .bool_set:N = \l__pic_line_dash_bool,
    dash   .initial:n = { false },
  }
\cs_new_protected:Nn \__pic_set_line_width:
  {
    \linethickness{ \l__pic_line_width_dim }
  }
\cs_new_protected:Nn \__pic_set_line_color:
  {
    \__color_safe_use:V \l__pic_line_draw_color_tl
  }
\cs_new_protected:Nn \__pic_set_fill_color:
  {
    \__color_safe_use:V \l__pic_region_fill_color_tl
  }
\def\z@pic@vector@style{\ltxarrows}
\keys_define:nn { ztool / draw / picture }
  {
    vector .inherit:n = { ztool/draw/picture/line },
  }
\keys_define:nn { ztool / draw / picture / vector }
  {
    >           .choice:,
    > / latex   .code:n = {\def\z@pic@vector@style{\ltxarrows}},
    > / pst     .code:n = {\def\z@pic@vector@style{\pstarrows}},
    > / unknown .code:n =
      {
        \msg_set:nnn { ztool }{unknown-arrow-style}
          { Unknown~arrow~style,~use~'latex'~or~'pst'. }
        \msg_error:nn { ztool }{unknown-arrow-style}
      }
  }
\cs_new_protected:Npn \ztool_pic_line_vector:nnnn #1#2#3#4
  {% #1:line/vector; #2:key-value; #3:start coor; #4:end coor;
    \group_begin:
    \keys_set:nn { ztool / draw / picture / #1 }{#2}
    \fp_set:Nn \l__draw_vector_slope_fp
      { (\__coor_nd:n {#4} - \__coor_nd:n {#3})
      / (\__coor_st:n {#4} - \__coor_st:n {#3}) }
    \fp_set:Nn \l__draw_vector_xsep_fp
      { abs(\__coor_st:n {#4} - \__coor_st:n {#3}) }
    \z@pic@vector@style
```

```
    \__pic_set_line_width:
    \exp_last_unbraced:Ne \__@@_pic_put:nnn {\__coor_st_nd:n {#3}}
      {
        \__pic_set_line_color:
        \cs:w __@@_pic_#1:nnn\cs_end: {1}
          { \l__draw_vector_slope_fp }
          { \l__draw_vector_xsep_fp  }
      }
    \group_end:
  }
\NewDocumentCommand{\zline}{O{}d()d()}
  {
    \ztool_pic_line_vector:nnnn {line}{#1}{#2}{#3}
  }
\NewDocumentCommand{\zvector}{O{}d()d()}
  {
    \ztool_pic_line_vector:nnnn {vector}{#1}{#2}{#3}
  }


% --> \zdraw -- similar to \tikz command in tikz
% NOTE: these line/vector commands are identical to
%  1. \Line (x1,y1)(x2,y2),          \Vector(x1,y1)(x2,y2)
%  2. \polyline (x1,y1)···(xn,yn), \polyvector(x1,y1)···(xn,yn)
%  3. \polygon (x1,y1)···(xn,yn), when set 'cycle',
%     \polygon*(x1,y1)···(xn,yn), when set 'fill' (auto cycle).
%  4. Trim leading space after '\polygon' or '*' to avoid error !!
\cs_new:Npn \__@@_pic_Line:nnnn #1#2#3#4
  { \Line (#1, #2)(#3, #4) }
\cs_new:Npn \__@@_pic_Vector:nnnn #1#2#3#4
  { \Vector (#1, #2)(#3, #4) }
\cs_new:Npn \__@@_pic_polyline:n #1
  {
    \tl_set:Ne \l_tmpa_tl {\tl_trim_spaces:e {#1}}
    \exp_last_unbraced:NV \polyline \l_tmpa_tl
  }
\cs_new:Npn \__@@_pic_polyvector:n #1
  {
    \tl_set:Ne \l_tmpa_tl {\tl_trim_spaces:e {#1}}
    \exp_last_unbraced:NV \polyvector \l_tmpa_tl
  }
\cs_new:Npn \__@@_pic_polygon:nn #1#2
  {
    \tl_set:Ne \l_tmpa_tl {\tl_trim_spaces:e {#1}}
    \tl_set:Ne \l_tmpb_tl {\tl_trim_spaces:e {#2}}
    \tl_set:Ne \l_tmpa_tl { \l_tmpa_tl\l_tmpb_tl }
    \exp_last_unbraced:NV \polygon \l_tmpa_tl
  }
\cs_generate_variant:Nn \__@@_pic_polygon:nn { nV, ne }
\tl_new:N \l__pic_region_fill_color_tl
\bool_new:N \l__pic_region_fill_bool
\keys_define:nn { ztool / draw / picture / region }
```

```
  {
    fill   .choices:nn  = { true, false }{
      \use:c { bool_set_ \l_keys_choice_tl :N }
        \l__pic_region_fill_bool
    },
    fill   .initial:n   = { false },
    fill   .default:n   = { true  },
    fill / unknown .code:n = {
      \tl_if_empty:eF \l_keys_value_tl
        { \bool_set_true:N \l__pic_region_fill_bool }
      \tl_set:Ne \l__pic_region_fill_color_tl { \l_keys_value_tl }
    },
  }
\keys_define:nn { ztool / draw / picture }
  {
    zdraw   .inherit:n   =   {
      ztool/draw/picture/line,
      ztool/draw/picture/vector,
      ztool/draw/picture/region,
    },
  }
\keys_define:nn { ztool / draw / picture / zdraw }
  {
    vector  .bool_set:N = \l__pic_draw_vector_bool,
    vector  .initial:n  = { false },
    cycle   .bool_set:N = \l__pic_draw_cycle_bool,
    cycle   .initial:n  = { false },
    shift   .tl_set:N   = \l__pic_draw_shift_tl,
    shift   .initial:n  = { 0, 0 },
  }
\cs_new:Npn \__region_fill_color_miss:n #1
  {
    \bool_if:eT {
      \l__pic_region_fill_bool &&
      \tl_if_empty_p:N \l__pic_region_fill_color_tl
    }{ \tl_set:Nn \l__pic_region_fill_color_tl {#1} }
  }
\cs_new_protected:Npn \ztool_pic_draw:nw #1#2;
  {% #1:key-value; #2:coors list (use ';' to end scan just like tikz)
    \group_begin:
    \keys_set:nn { ztool / draw / picture / zdraw }{#1}
    \__region_fill_color_miss:n { gray }
    \edef\coors@first
      {
        \exp_last_unbraced:Ne
          \__coors_list_first:w {\tl_trim_spaces:e {#2}}
          \scan_stop:
      }
    \edef\draw@flag
      {
        \tl_map_function:nN {
          \l__pic_draw_vector_bool
```

```
          \l__pic_draw_cycle_bool
          \l__pic_region_fill_bool
        } \int_eval:n
      }
    \__@@_pic_put:nnn
      { \__coor_st:V \coors@first + \__coor_st:V \l__pic_draw_shift_tl }
      { \__coor_nd:V \coors@first + \__coor_nd:V \l__pic_draw_shift_tl }
      {
        \__pic_set_line_width:
        \__pic_set_line_color:
        \exp_after:wN \int_case:nnF \exp_after:wN {
            \exp_after:wN \int_from_bin:n \exp_after:wN
              { \draw@flag }
          }{
            {0}{ \__@@_pic_polyline:n    {#2} }
            {1}{ \__@@_pic_polygon:nn {*}{#2} }
            {2}{ \__@@_pic_polygon:ne { }{#2} }
            {3}{ \__@@_pic_polygon:nn {*}{#2} }
            {4}{ \__@@_pic_polyvector:n  {#2} }
            {5}{
              \__pic_set_fill_color:
              \__@@_pic_polygon:nn {*}{#2}
              \__pic_set_line_color:
              \exp_args:Ne \__@@_pic_polyvector:n {#2(\coors@first)}
            }
            {6}{ \exp_args:Ne \__@@_pic_polyvector:n {#2(\coors@first)} }
            {7}{
              \__pic_set_fill_color:
              \__@@_pic_polygon:nn {*}{#2}
              \__pic_set_line_color:
              \exp_args:Ne \__@@_pic_polyvector:n {#2(\coors@first)}
            }
          }{\relax}
      }
    \group_end:
  }
\cs_new:Npn \__coors_list_first:w (#1)#2\scan_stop:
  { #1 }
\NewDocumentCommand{\zdraw}{O{}}
  { \ztool_pic_draw:nw {#1} }


% --> arc / circle
\cs_new:Npn \__@@_pic_arc:nnnn #1#2#3#4
  {% #1:fill bool; #2:start angle; #3:end angle; #4:radius
    \arc #1[\fp_eval:n {#2}, \fp_eval:n {#3}]
      { \fp_eval:n {#4} }
  }
\cs_new:Npn \__@@_pic_circel:nn #1#2
  {% #1:fill bool; #2:radius
    \__@@_pic_arc:nnnn {#1}{0}{360}{#2}
  }
```

```
% --> circle
\keys_define:nn { ztool / draw / picture }
  {
    arc   .inherit:n   =    {
      ztool/draw/picture/line,
      ztool/draw/picture/region,
    },
  }
\keys_define:nn { ztool / draw / picture / arc }
  {
    radius .fp_set:N   = \l__pic_arc_radius_fp,
    radius .initial:n  = .5,
    start  .fp_set:N   = \l__pic_arc_start_fp,
    start  .initial:n  = 0,
    end    .fp_set:N   = \l__pic_arc_end_fp,
    end    .initial:n  = 90,
  }
\prg_generate_conditional_variant:Nnn
  \bool_if:n { e } { p, T, F, TF }
\cs_new_protected:Npn \ztool_pic_arc:nn #1#2
  {% #1:key-value; #2:coor
    \group_begin:
    \keys_set:nn { ztool / draw / picture / arc }{#1}
    \__region_fill_color_miss:n { gray }
    \__color_if_valid:VF \l__pic_region_fill_color_tl
      { \bool_set_false:N \l__pic_region_fill_bool }
    \exp_last_unbraced:Ne \__@@_pic_put:nnn {\__coor_st_nd:n {#2}}
      {
        \__pic_set_line_width:
        \bool_if:eT \l__pic_region_fill_bool
          {
            \__pic_set_fill_color:
            \exp_args:Ne \__@@_pic_arc:nnnn {*}
              { \fp_use:N \l__pic_arc_start_fp  }
              { \fp_use:N \l__pic_arc_end_fp    }
              { \fp_use:N \l__pic_arc_radius_fp }
          }
        % NOTE: border must over the fill
        \__pic_set_line_color:
        \exp_args:Ne \__@@_pic_arc:nnnn {}
          { \fp_use:N \l__pic_arc_start_fp  }
          { \fp_use:N \l__pic_arc_end_fp    }
          { \fp_use:N \l__pic_arc_radius_fp }
      }
    \group_end:
  }
\NewDocumentCommand{\zarc}{O{}d()}
  {% #1:key-value; #2:coor
    \ztool_pic_arc:nn {#1}{#2}
  }
```

```
\NewDocumentCommand{\zcircle}{O{}d()}
  {
    \ztool_pic_arc:nn {start=0, end=360, #1}{#2}
  }


% --> oval / rectangle
% \oval[⟨arc⟩](⟨full-x-width, full-y-width⟩)[⟨part⟩]
% part: (l, r) × (t, b)
\cs_new:Npn \__@@_pic_oval:nnnn #1#2#3#4
  {% #1:arc; #2:part; #3:x-width; #4:y-width;
    \oval
      [\fp_eval:n {#1}]
      (\fp_eval:n {#3}, \fp_eval:n {#4})
      [ #2 ]
  }
\keys_define:nn { ztool / draw / picture }
  {
    rectangle   .inherit:n  =  {
      ztool/draw/picture/line,
      ztool/draw/picture/region,
    },
  }
\keys_define:nn { ztool / draw / picture / rectangle }
  {
    arc     .fp_set:N   = \l__pic_rec_arc_fp,
    arc     .initial:n  = 0,
  }
\int_new:N \l__pic_rec_quadrant_index_int
\cs_new_protected:Npn \ztool_pic_rectangle:nnn #1#2#3
  {% #1:key-value; #2:start coor; #3:end coor;
    \group_begin:
    \keys_set:nn { ztool / draw / picture / rectangle }{ fill=false }
    \keys_set:nn { ztool / draw / picture / rectangle }{ #1 }
    \edef\rec@arc   { \fp_use:N  \l__pic_rec_arc_fp    }
    \edef\rec@width { \fp_eval:n {\__coor_st:n {#3} - \__coor_st:n {#2}} }
    \edef\rec@height{ \fp_eval:n {\__coor_nd:n {#3} - \__coor_nd:n {#2}} }
    \__region_fill_color_miss:n { gray }
    \__color_if_valid:VF \l__pic_region_fill_color_tl
      {
        \bool_set_false:N \l__pic_region_fill_bool
        \prg_map_break:Nn \l__ztool_pic_rec_fill {}
      }
    %% begin fill rounded rectangle
    \__@@_pic_put:nnn {\__coor_st:n {#2}}{\__coor_nd:n {#2}}
      {
        \__pic_set_fill_color:
        \rule
          {\fp_eval:n {\rec@width *\dim_to_decimal:n {\l__pic_unit_dim}}pt}
          {\fp_eval:n {\rec@height*\dim_to_decimal:n {\l__pic_unit_dim}}pt}
      }
    \int_set:Nn \l__pic_rec_quadrant_index_int { 0 }
```

```
    \tl_map_inline:nn
      {
        {\__coor_st:n {#2}+\rec@width-\rec@arc, \__coor_nd:n
        {#2}+\rec@height-\rec@arc}
        {\__coor_st:n {#2}+\rec@arc,            \__coor_nd:n
        {#2}+\rec@height-\rec@arc}
        {\__coor_st:n {#2}+\rec@arc,            \__coor_nd:n {#2}+\rec@arc}
        {\__coor_st:n {#2}+\rec@width-\rec@arc, \__coor_nd:n {#2}+\rec@arc}
      }{
        \int_incr:N \l__pic_rec_quadrant_index_int
        \edef\qu@drant@index{\int_use:N \l__pic_rec_quadrant_index_int}
        \exp_last_unbraced:Ne \__@@_pic_put:nnn {\__coor_st_nd:n {##1}}
          {
            \__color_safe_use:V \l__pic_opacity_color_tl
            \__@@_pic_arc:nnnn {*}
              { (\qu@drant@index-1)*90 }
              { \qu@drant@index*90     }
              { sqrt(2)*\rec@arc       }
            \__pic_set_fill_color:
            \__@@_pic_arc:nnnn {*}{0}{360}{\rec@arc}
          }
      }
    %% end fill rounded rectangle
    \prg_break_point:Nn \l__ztool_pic_rec_fill { }
    \__@@_pic_put:nnn {\__coor_st:n {#2}+\rec@width/2}{\__coor_nd:n
    {#2}+\rec@height/2}
      {
        \__pic_set_line_color:
        \__pic_set_line_width:
        \__@@_pic_oval:nnnn
          { \rec@arc }{ }
          { \rec@width  }
          { \rec@height }
      }
    \group_end:
  }
\NewDocumentCommand{\zrectangle}{O{}d()d()}
  {
    \ztool_pic_rectangle:nnn { #1 }{#2}{#3}
  }
```

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.