

# 几何库 geometry 函数介绍

松坂さとう

## 功能型函数

`round(num[, decimal])`--保留指定小数位数

第一个参数填数字；第二个参数填要保留的小数位数，不填默认取整。

`random_N(min[, max][, variance][, expectation])`--正态分布随机数发生器

第一个参数填最小值；第二个参数填最大值，不填默认范围从 0 到最小值；第三个参数填方差，不填默认 $((max-min)/6)^2$ ；第四个参数填期望，不填默认范围内的平均值。

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

正态分布概率密度函数

`random_S(min, max, segments)`--分段随机

第一个参数填最小值；第二个参数填最大值；第三个参数填分段数，有几段就产生几个随机数。函数返回一个装着随机数的表。

`random_P(min, max, step)`--生成一个可指定步长的随机数

第一个参数填最小值；第二个参数填最大值；第三个参数填步长。

`reverse_color(color)`--计算一种颜色的反色

唯一一个参数为颜色字符串。

`gradient_color(c1, c2[, pct][, accele])`--计算两种颜色的渐变色

前两个参数填颜色字符串；第三个参数填渐变程度，范围[0,1]，0 为 c1,1 为 c2，不填默认 0.5；第四个参数填加速度，不填默认 1。

`color_brightness(color, pct)`--控制颜色的亮度

第一个参数填颜色字符串；第二个参数填程度，范围 0 到正无穷，0 为黑色，0~1 之间颜色变暗，大于 1 颜色变亮，最亮为白色。

### **color\_alpha(color,pct)--根据颜色设定透明度**

第一个参数填颜色字符串，颜色越浅越不透明，颜色越深越透明；第二个参数填程度，范围 0 到正无穷，0 为完全不透明，数字越大越透明，直至完全透明。

### **timing(num,func,...)--函数测速**

第一个参数填函数要运行的次数；第二个参数填函数名；后面的参数依次填要测速的函数的参数。函数返回精确到毫秒的总运行时间和精确到微秒的单次运行时间。

## **生成绘图型函数**

### **circle(diameter[,clockwise])--固定直径圆形，可指定路径方向**

第一个参数填圆的直径；第二个参数可选，填 0 或 1（0 代表路径顺时针，1 代表路径逆时针），不填默认顺指针。

### **random\_circle(min,max[,clockwise])--随机范围直径圆形，可指定路径方向**

第一个参数填圆的最小直径；第二个参数填圆的最大直径；第三个参数可选，填 0 或 1（0 代表路径顺时针，1 代表路径逆时针），不填默认顺指针。

### **annulus(r,R)--生成圆环**

第一个参数填内半径；第二个参数填外半径。

### **ellipse(x\_length,y\_length[,clockwise])--椭圆，可指定路径方向**

第一个参数填椭圆的长轴长度；第二个参数填椭圆的短轴长度；第三个参数可选，填 0 或 1（0 代表路径顺时针，1 代表路径逆时针），不填默认顺指针。

### **isosceles\_triangle(length,height[,clockwise])--固定底高等腰三角形，可指定路径方向**

第一个参数填等腰三角形的底的长度；第二个参数填等腰三角形的高；第三个参数可选，填 0 或 1（0 代表路径顺时针，1 代表路径逆时针），不填默认顺指针。

**regular\_triangle(length[,clockwise])**--固定大小正三角形，可指定路径方向

第一个参数填这个正三角形的最小外接圆的直径；第二个参数可选，填 0 或 1 ( 0 代表路径顺时针，1 代表路径逆时针 )，不填默认顺指针。

**square(length[,clockwise])**--固定边长正方形，可指定路径方向

第一个参数填正方形的边长；第二个参数可选，填 0 或 1 ( 0 代表路径顺时针，1 代表路径逆时针 )，不填默认顺指针。

**random\_square(min,max[,clockwise])**--随机范围边长正方形，可指定路径方向

第一个参数填正方形的最小边长；第二个参数填正方形的最大边长；第三个参数可选，填 0 或 1 ( 0 代表路径顺时针，1 代表路径逆时针 )，不填默认顺指针。

**rectangle(length,height[,clockwise])**--固定长宽矩形，可指定路径方向

第一个参数填长方形的长；第二个参数填长方形的宽；第三个参数可选，填 0 或 1 ( 0 代表路径顺时针，1 代表路径逆时针 )，不填默认顺指针。

**random\_rectangle(l\_min,l\_max[,h\_min][,h\_max][,clockwise])**--随机范围长宽矩形，可指定路径方向

第一个参数填矩形的最小长度；第二个参数填矩形的最大长度；第三个参数填矩形的最小高度，不填默认等于最小长度；第四个参数填矩形的最大高度，不填默认等于最大长度；第五个参数可选，填 0 或 1 ( 0 代表路径顺时针，1 代表路径逆时针 )，不填默认顺指针。

**rounded\_rectangle(length[,height][,roundx][,roundy][,clockwise])**--固定长宽圆角矩形，可指定路径方向

第一个参数填圆角矩形的长度；第二个参数填圆角矩形的高度，不填默认等于长度；第三个参数填圆角矩形的圆角长度，不填默认等于长度和高度中较小者的 1/5；第四个参数填圆角矩形的圆角高度，不填默认等于圆角长度；第五个参数可选，填 0 或 1 ( 0 代表路径顺时针，1 代表路径逆时针 )，不填默认顺指针。

**ran\_rou\_rect(l\_min,l\_max[,h\_min][,h\_max][,rx][,ry][,clockwise])**--随机范围长宽圆角矩形，可指定路径方向

第一个参数填圆角矩形的最小长度；第二个参数填圆角矩形的最大长度；第三个参数填圆角矩形的最小高度，不填默认等于最小长度；第四个参数填圆角矩形的最大高度，不填默认等于最大长度；第五个参数填圆角矩形的圆角长度，不填默认等于生成长度和生成高度中较小者的 1/5；第六个参数填圆角矩形的圆角高度，不填默认等于圆角长度；第七个参数可选，填 0 或 1（0 代表路径顺时针，1 代表路径逆时针），不填默认顺指针。

**rhombus(length,height[,clockwise])**--固定长高菱形，可指定路径方向  
第一个参数填菱形的横向长度；第二个参数可选，填菱形的纵向高度，不填默认等于菱形的横向长度；第三个参数可选，填 0 或 1（0 代表路径顺时针，1 代表路径逆时针），不填默认顺指针。

**parallelogram(length,height,incline[,directivity][,clockwise])**--固定长高平行四边形，可指定倾斜量、倾斜方向和路径方向  
第一个参数填平行四边形的长；第二个参数填平行四边形的高；第三个参数填倾斜量；第四个参数可选，填 0 或 1（0 代表左倾斜，1 代表右倾斜），不填默认右倾斜；第五个参数可选，填 0 或 1（0 代表路径顺时针，1 代表路径逆时针），不填默认顺指针。

**star(length[,curvature][,clockwise])**--星形，可指定形状和路径方向  
第一个参数填星形的长宽；第二个参数控制星形的“胖瘦”，它不能超过长宽的一半，不填默认等于长宽的 1/10；第三个参数可选，填 0 或 1（0 代表路径顺时针，1 代表路径逆时针），不填默认顺指针。

**pentagram(length[,clockwise][,proportion])**--五角星形，可指定路径方向和形状

第一个参数填这个五角星形的最小外接圆的直径；第二个参数可选，填 0 或 1（0 代表路径顺时针，1 代表路径逆时针），不填默认顺指针；第三个参数可选，填这个五角星形的最大内切圆的半径与最小外接圆的半径的比值。它控制这个五角星的“胖瘦”，可填的范围是(0,sin54° ] (sin54° ≈ 0.809)。当填入的值小于等于 0 时取 0.001；大于 sin54° 时取 sin54° ；不填默认为

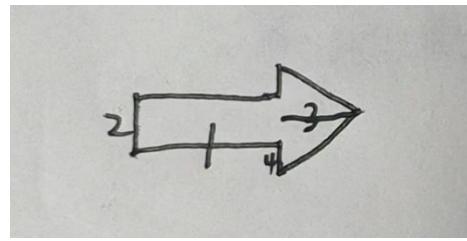
$\sin 18^\circ / \sin 54^\circ$  (此值约为 0.379, 此时形状为正五角星形)。填入的值越接近 0, 五角星越“瘦”; 越接近  $\sin 54^\circ$ , 五角星越“胖”。

**regular\_hexagon(length[, clockwise])**--固定边长正六边形, 可指定路径方向

第一个参数填这个正六边形的最小外接圆的直径; 第二个参数可选, 填 0 或 1 (0 代表路径顺时针, 1 代表路径逆时针), 不填默认顺指针。

**arrow(length1, length2[, length3][, length4][, direction][, clockwise])**--箭头, 可指定指向和路径方向

前四个参数与如图所示, 后四个参数为可选参数。 $length_3$  的默认值为  $length_1$  的一半,  $length_4$  的默认值为  $length_2$  的一半。第五个参数填 1 或 2 或 3 或 4, 方向分别对应上下左右, 不填默认 4; 第六个参数可选, 填 0 或 1 (0 代表路径顺时针, 1 代表路径逆时针), 不填默认顺指针。



**note([x])**--七个音符, 可指定任意一个

该函数只有一个参数, 填 1~7。每个值与音符的对应关系如下:

- 1: 高分音符
- 2: 二分音符
- 3: 四分音符
- 4: 八分音符
- 5: 两个八分音符 (用符杠连接两个八分音符的符干)
- 6: 十六分音符
- 7: 两个十六分音符 (用符杠连接两个十六分音符的符干)

参数也可以不填, 不填默认 7。一般配合 `math.random(7)` 使用。

**structural\_formula([n])**--八个化学结构式, 可指定任意一个

该函数只有一个参数, 填 1~8。对于有机化合物分子, 通常采用键线式表示其结构, 每个值与化学物质的对应关系如下:

- 1: 布洛芬
- 2: 维生素 C
- 3: 麦角酸二乙酰胺

4: 维生素 A

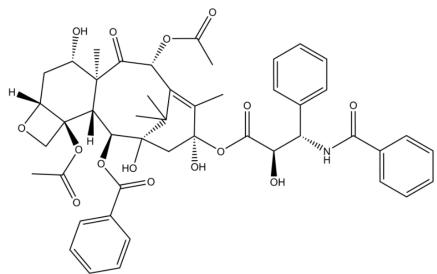
5: 前列环素

6: 红霉素 A

7: 紫杉醇

8: 环孢菌素 A

参数也可以不填，不填默认 7。一般配合  
math.random(8)使用。随着数字的增  
大，对应物质的分子结构会越来越复杂。



紫杉醇的分子结构式

**慎用旋转类效果（表示原子的字母也会跟着旋转）**

（在添加该函数后，函数库体积从 45kb 激增至 138kb……）

**binary\_digit(digit)**--生成指定位数的随机二进制数字绘图

唯一一个参数为要生成的二进制数字的位数。  
**10100001**

**sector(r[,s\_ang][,e\_ang][,clockwise])**--生成扇形

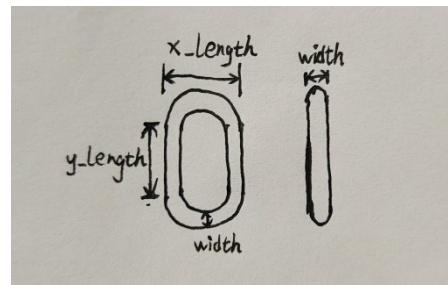
第一个参数填扇形的半径；第二个参数填以 y 轴上半轴为 0 度的开始角度，不填默认 0；第三个参数填结束角度，不填默认 360， $0 < e\_ang - s\_ang \leq 360$ ；第四个参数为角度增大的方向，填 0 为顺指针，填 1 为逆时针，不填默认 0。

**clip\_blinds(length,height,num,pct[,angle][,x][,y][,direction][,mode])**--生成百叶窗绘图，用于 clip 效果

第一个参数填覆盖范围的长度；第二个参数填覆盖范围的高度；第三个参数填百叶窗的条数；第四个参数填展开程度，范围[0,1]，0 为未展开，1 为完全展开；第五个参数填百叶窗的旋转角度，不填默认 0；第六个参数填百叶窗的横向平移距离，不填默认 0；第七个参数填百叶窗的纵向平移距离，不填默认 0；第八个参数控制展开方向，填 0 或 1，0 代表从左向右展开，1 代表从中间向两侧展开，不填默认 0；第九个参数为返回值模式，填 0 或 1，0 代表函数返回字符串，1 代表函数返回表，不填默认 0。

**chain(num[,x\_length][,y\_length][,width][,first])**--生成直线锁链绘图

第一个参数填链环的个数；第二个参数填 0 环的横向长度，默认值为 30；第三个参数填 0 环平直部分的纵向长度，默认值为 20；第四个参数填 1 环的宽度，默认值为 5；第五个参数指定第一个链环是 0 环还是 1 环，填 0 或 1，默认值为 0。



## 操作绘图型函数

**translate(ass\_shape[,x\_incline][,y\_incline])--平移绘图**

第一个参数填绘图代码（支持整数和小数）；第二个参数填 x 方向上的位移（水平向右为正方向），不填默认 0；第三个参数填 y 方向上的位移（水平向下为正方向），不填默认 0。

**zoom(ass\_shape[,x\_zoom][,y\_zoom][,zoom\_center][,zoom\_middle])-缩放绘图**

第一个参数填绘图代码（支持整数和小数）；第二个参数填横向缩放的百分比，不填默认 100；第三个参数纵向缩放的百分比，不填默认等于横向缩放的百分比；第四个参数填缩放中心的 x 坐标，不填默认 0；第五个参数填缩放中心的 y 坐标，不填默认 0。

**spin(ass\_shape[,x\_angle][,y\_angle][,z\_angle][,spin\_center][,spin\_middle])--旋转绘图**

第一个参数填绘图代码（支持整数和小数）；第二个参数填绕 x 轴的旋转角，不填默认 0；第三个参数填绕 y 轴的旋转角，不填默认 0；第四个参数填绕 z 轴的旋转角，不填默认 0；第五个参数填旋转中心的 x 坐标，不填默认 0；第六个参数填旋转中心的 y 坐标，不填默认 0。（注：该函数与用于旋转的标签不同，没有“近大远小”的视觉效果。）

**translate\_tbl(ass\_table[,x\_incline][,y\_incline])--平移绘图表中的每个绘图**

第一个参数填装着绘图代码的表；第二个参数填 x 方向上的位移（水平向右为正方向），不填默认 0；第三个参数填 y 方向上的位移（水平向下为正方向），不填默认 0。

**zoom\_tbl(ass\_table[,x\_zoom][,y\_zoom][,zoom\_center][,zoom\_middle])**--缩放绘图表中的每个绘图

第一个参数填装着绘图代码的表；第二个参数填横向缩放的百分比，不填默认 100；第三个参数纵向缩放的百分比，不填默认等于横向缩放的百分比；第四个参数填缩放中心的 x 坐标，不填默认 0；第五个参数填缩放中心的 y 坐标，不填默认 0。

**spin\_tbl(ass\_table[,x\_angle][,y\_angle][,z\_angle][,spin\_center][,spin\_middle])**--旋转绘图表中的每个绘图

第一个参数填装着绘图代码的表；第二个参数填绕 x 轴的旋转角，不填默认 0；第三个参数填绕 y 轴的旋转角，不填默认 0；第四个参数填绕 z 轴的旋转角，不填默认 0；第五个参数填旋转中心的 x 坐标，不填默认 0；第六个参数填旋转中心的 y 坐标，不填默认 0。(注：该函数与用于旋转的标签不同，没有“近大远小”的视觉效果。)

**round\_tbl(ass\_table[,decimal])**--给表里的绘图坐标保留指定小数位数

第一个参数填装着绘图代码的表；第二个参数填要保留的小数位数，不填默认取整。

**rect\_pos(x1,y1,x2,y2)**--矩形边界定位

前两个参数填矩形左上角的点的坐标；后两个参数填矩形右下角的点的坐标。  
函数返回一个在矩形四条边上的任意点坐标。

**disassemble(ass\_shape)**--拆解单 m 绘图

唯一一个参数为一个绘图代码，函数返回一个装着所有单 m 绘图的表。

**ring(ass\_shape,n)**--旋转绘图并连接

第一个参数填绘图代码，第二个参数填一圈有多少个。

**reverse\_tbl(tbl)**--反转表中元素顺序

参数填一个表。函数返回元素顺序被反转的表。

**shuffle(tbl)**--打乱表中元素顺序

参数填一个表。函数返回元素顺序被打乱的表。

**part(tbl, level[, mode])**--随机显示表中一部分比例的绘图

第一个参数填装着绘图代码的表；第二个参数填一个 0~1 范围内的数字，它表示要显示的绘图代码的比例；第三个参数为返回值模式，填 0 或 1。0 代表函数返回字符串；1 代表函数返回表。不填默认 0。

**arrange(ass\_shape,line\_number,x\_incline[,line][,y\_incline][,first\_proportion][,last\_proportion][,line\_x\_incline][,mode])**

--[[生成规律排列的绘图 参数:图形,单行个数,x 偏移量,总行数,y 偏移量,第一行缩放比例,最后一行缩放比例,偶数行

初始 x 偏移量,模式]]

第 1 个参数 ( 图形 ): 坐标为整数小数均可。

第 2 个参数 ( 单行个数 ): 一行有几个图形。

第 3 个参数 ( x 偏移量 ): 每行相邻两个图形几何中心的 x 坐标的差值。

第 4 个参数 ( 总行数, 可选 ): 一共有多少行, 不填默认 1。

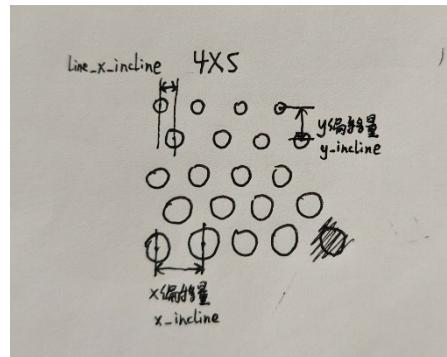
第 5 个参数 ( y 偏移量, 可选 ): 相邻两行的图形的 y 坐标的差值, 不填默认等于 x 偏移量。

第 6 个参数 ( 第一行缩放比例, 可选 ): 第一行的图形与填入图形的缩放百分比, 不填默认 100。

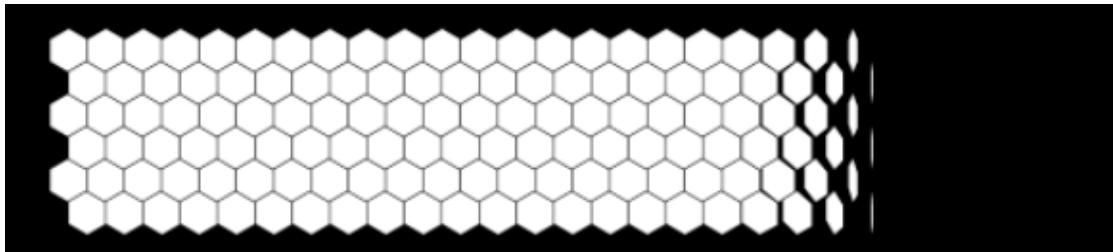
第 7 个参数 ( 最后一行缩放比例, 可选 ): 最后一行的图形与填入图形的缩放百分比, 不填默认等于第一行缩放比例。

第 8 个参数 ( 偶数行初始 x 偏移量, 可选 ): 第二行第一个图形与第一行第一个图形的 x 坐标的差值, 不填默认 0。

第 9 个参数 ( 返回值模式, 可选 ): 填 0 或 1。0 代表函数返回字符串; 1 代表函数返回表。不填默认 0。



**overturn(ass\_shape,line\_number,x\_incline,line,y\_incline)**--做密



## 铺正六边形和菱形翻转效果

第1个参数(图形): 填入一个正六边形或菱形的绘图代码。

第2个参数(单行个数): 一行有几个图形。

第3个参数(x偏移量): 每行相邻两个图形几何中心的x坐标的差值。

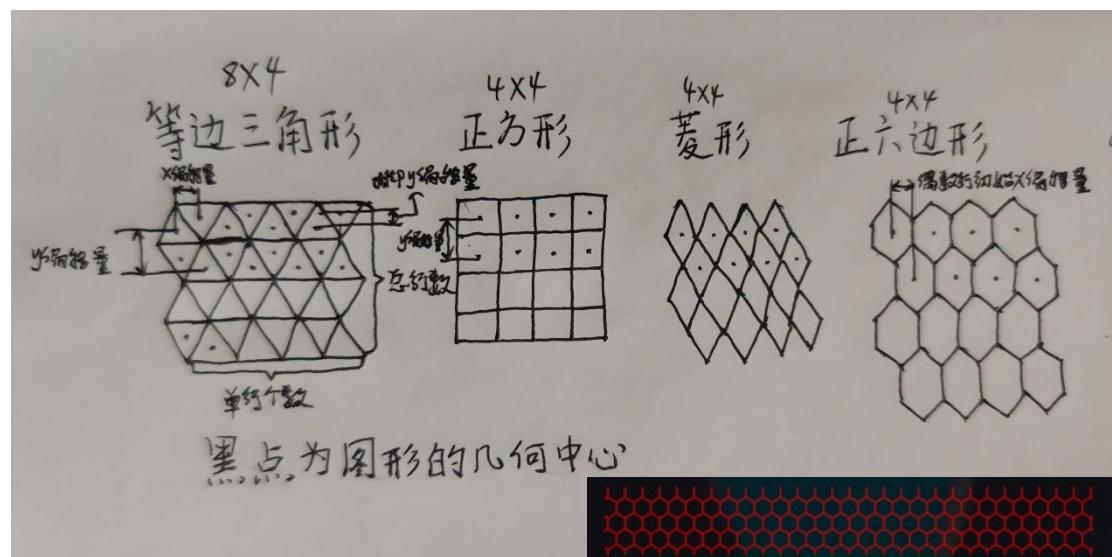
第4个参数(总行数): 一共有多少行(如果你填入的是奇数, 最终也会生成刚好大于这个奇数的偶数行)。

第5个参数(y偏移量): 第二行与第一行的图形的几何中心的y坐标的差值。

函数返回一个表, 里面装着一个固定的绘图代码和定位信息。调用绘图代码时要用 `a.s` 的形式, 调用定位信息时要用 `a[j].x` 和 `a[j].y` 的形式(`a`的命名可以更换)。

```
tessellation(shape,line_number,x_incline,line,y_incline[,line_x_incline][,first_overturn][,adjacent_overturn][,adjacent_y_incline][,mode])
```

--[[生成密铺状态的可密铺图形 参数:图形,单行个数,x偏移量,总行数,y偏移量,偶数行初始x偏移量,偶数行第一个图形翻转状态,每行相邻两个图形的翻转状态,每行相邻两个图形的y偏移量,模式]]



第1个参数(图形): 支持整数和小数, 但正六边形和正三角形最好用小数。

(注: geometry 中所有用算法生成的图形坐标都是小数。)

第2个参数(单行个数): 一行有几个图形。

第3个参数(x偏移量): 每行相邻两个图形几何中心的x坐标的差值。

第4个参数(总行数): 一共有多少行。

第5个参数(y偏移量): 相邻两行中翻转状态相同的图形的y坐标的差值。

第6个参数(偶数行初始x偏移量, 可选): 第二行第一个图形与第一行第一

个图形的 x 坐标的差值。不填默认 0。

第 7 个参数 (偶数行第一个图形翻转状态, 可选): 填 0 或 1 或 2。0 代表图形绕 y 轴横向翻转 180 度; 1 代表图形绕 x 轴纵向翻转 180 度; 2 代表不翻转。不填默认 2。

第 8 个参数 (每行相邻两个图形的翻转状态, 可选): 填 0 或 1。0 代表图形绕 x 轴纵向翻转 180 度; 1 代表不翻转。不填默认 1。

第 9 个参数 (每行相邻两个图形的 y 偏移量, 可选): 每行相邻两个图形的 y 坐标的差值。不填默认 0。

第 10 个参数 (返回值模式, 可选): 填 0 或 1。0 代表函数返回字符串; 1 代表函数返回表。不填默认 0。

**Zpix(shape, width, height)**--返回字体所有可能的像素点坐标表和零矩阵

**※函数必须用在 syl 模式下※**

第一个参数填文字绘图; 第二个参数填 syl.width; 第三个参数填 syl.height。

**FOT\_DotMincho12\_Std\_M(shape, width, height)**--返回字体所有可能的像素点坐标表和零矩阵

**※函数必须用在 syl 模式下※**

第一个参数填文字绘图; 第二个参数填 syl.width; 第三个参数填 syl.height。

**pt\_in\_str\_l\_shape(ass\_shape, pt\_x, pt\_y)**--判断点是否包含在全直线绘图内

参考了多华宫前辈的代码。第一个参数填文字绘图; 第二个参数填要判断的点的 x 坐标; 第三个参数填要判断的点的 y 坐标。函数返回布尔值, true 表示在绘图内, false 表示不在绘图内。

**tbl\_pt\_in\_str\_l\_shape(ass\_shape, tbl[, matrix])**--移除表中不包含在全直线绘图内的点并返回结果和二值矩阵

第一个参数填文字绘图; 第二个参数填像素字体函数返回的坐标表; 第三个参数填像素字体函数返回的零矩阵, 这个参数可以不填, 不填就不会返回二值矩阵。二值矩阵中 0 表示没有像素点, 1 表示有像素点。

**plies(matrix)**--根据二值矩阵计算每个像素点下有几个空位

参数填 `tbl_pt_in_str_l_shape` 返回的二值矩阵。返回的表中的数字与每个像素点一一对应，表示第几个像素点下有几个空位。

**isolated\_point(matrix)**--根据二值矩阵判断孤立像素点并返回索引表

参数填 `tbl_pt_in_str_l_shape` 返回的二值矩阵。返回的索引表中装着所有孤立像素点是从左到右从上到下的第几个点的索引。

**blob(matrix)**--查找二值矩阵的连通域

参数填 `tbl_pt_in_str_l_shape` 返回的二值矩阵。返回的连通域表中第一个子表装着第一个连通域的索引，第二个子表装着第二个连通域的索引，以此类推。

**num\_in\_tbl(tbl,num)**--查找给定数字是否在表中

第一个参数填数组表；第二个参数填数字。

**num\_in\_nested(tbl,num)**--查找给定数字在嵌套表的第几个表里

第一个参数填数组表；第二个参数填数字。数字在嵌套表的第几个表里函数就返回几，如果没有找到则返回 nil。

**close(ass\_shape)**--封闭绘图

唯一一个参数为绘图字符串，函数返回封闭路径的绘图。

**topmost(ass\_shape)**--求全直线绘图的所有最顶端的点

唯一一个参数为全直线绘图字符串，函数返回装着所有最顶端的点坐标的表。

调用定位信息时要用 `a[j].x` 和 `a[j].y` 的形式（`a` 的命名可以更换）。