# pygame.draw.circle()

格式：pygame.draw.circle(画布对象,(r,g,b),(core.x,core.y),半径,[width])

(r,g,b)：表示颜色，(core.x,core.y)表示圆心坐标，width：可选，表示线的粗细

若 width == 0, (默认) 填充圆形

若 width > 0, 用于指示线条粗细

若 width < 0, 不会绘制任何内容

注意：

当使用 width > 1时，边缘线将仅向内增长。

返回值：返回值：（[Rect](https://www.pygame.org/docs/ref/rect.html#pygame.Rect)）a rect bounding the changed pixels

# pygame.draw.rect()

格式:

pygame.draw.rect(画布,(r,g,b,[a]),(let,top,width,height),[width]),

(let,top,width,height):一个4元组表示的矩形，包含位置和大小

width：可选，值线宽

返回值：返回值：（[Rect](https://www.pygame.org/docs/ref/rect.html#pygame.Rect)）一个包围改变像素的矩形。如果没有画出任何东西，则包围矩形的位置将是给定 rect 参数的位置，其宽度和高度将为0

# 3.pygame.draw.polygon()

格式：

pygame.draw.polygon(画布,(r,g,b,[a]),((x,y),(x,y),(x,y)…),[width])

((x,y),(x,y),(x,y)…)：顶点元组，至少3个以上

返回值：（[Rect](https://www.pygame.org/docs/ref/rect.html#pygame.Rect)）一个包围更改像素的矩形，如果没有绘制任何内容，则包围矩形的位置将是 points 参数中第一个点的位置（浮点值将被截断），其宽度和高度将为0

# 4.pygame.draw.ellipse()

格式：

pygame.draw.ellipse(画布,(r,g,b[a]),(rect.left,rect.top,width,height),[width])

(rect.left,rect.top,width,height)：是一个矩形的位置和大小

返回值：（[Rect](https://www.pygame.org/docs/ref/rect.html#pygame.Rect)）一个包围改变像素的矩形。如果没有画出任何东西，则包围矩形的位置将是给定 rect 参数的位置，其宽度和高度将为0。

# 5.pygame.draw.arc()

格式：

pygame.draw.arc(画布,(r,g,b[a]), start\_angle, stop\_angle, [width])

画布,(r,g,b[a]),同上。

start\_angle, stop\_angle: 两个角度参数以弧度表示，并指示弧的开始和停止位置

**若start\_angle < stop\_angle, 从start\_angle 到stop\_angle以逆时针方向画弧**

**若 start\_angle > stop\_angle, 将 tau (tau == 2 \* pi)添加到 stop\_angle：**

**如果生成的终止角度值大于 start\_angle，则上述start\_angle < stop\_angle情况适用，；否则将不绘制任何内容**

**若start\_angle == stop\_angle, 不绘制任何内容**

width：线宽，可选

返回值：（[*Rect*](https://www.pygame.org/docs/ref/rect.html#pygame.Rect)）一个包围改变像素的矩形。如果没有画出任何东西，则包围矩形的位置将是给定 rect 参数的位置，其宽度和高度将为0

# 6.pygame.draw.line()

格式：

pygame.draw.line(画布,(r,g,b[a]),(start.x,start.y),(end.x,end.y),[width])

(start.x,start.y): 起点坐标。

(end.x,end.y): 终点坐标

**width (int) -- （可选）用于指示线条粗细**

**若 width >= 1, 则用于线条粗细 (默认值是1)**

**若 width < 1, 不绘制任何内容**

**注意：当使用 width > 1时，线条将按如下方式增长：**

**对于奇数 width 值，每一条线的宽度度随着原始行位于中心而增大。**

**对于偶数 width 值，每条线的宽度随着原始线从中心偏移而增大（因为没有绘制精确的中心线）。因此，斜率<1（horizontal-ish）的线将比原始线（y方向）多1个像素厚。斜率>=1（vertical-ish）的线在原始线的右侧（x方向）将多出1个厚度像素。**

返回值：（[*Rect*](https://www.pygame.org/docs/ref/rect.html#pygame.Rect)）一个包围更改像素的矩形，如果没有绘制任何内容，则包围矩形的位置将是 start\_pos 参数值（浮点值将被截断），其宽度和高度将为0

提示：**TypeError** -- 如果 start\_pos or end\_pos 不是两个数字的序列

# 7.pygame.draw.lines()

功能：绘制多个连续直线段.

格式：

**pygame.draw.lines(画布,(r,g,b[a]),closed,points,[width])**

**closed** (*bool*) -- 若为True ，则在 points 序列的第一个点和最后一个点之间绘制一条附加线段

**points (tuple(coordinate) or list(coordinate)) -- 一个含有2个或更多 (x, y) 坐标的序列，序列中的每个坐标必须是2个整数/浮点数格式的tuple/list/pygame.math.Vector2，同时相邻坐标将通过线段连接。【例如，对于点[(x1, y1), (x2, y2), (x3, y3)]线段将从(x1, y1)绘制到(x2, y2)并从(x2, y2) 绘制到(x3, y3)，此外，如果闭合参数为真，则另一线段将从(x3, y3)绘制到(x1, y1)】**

返回值：（[*Rect*](https://www.pygame.org/docs/ref/rect.html#pygame.Rect)）一个包围更改像素的矩形，如果没有绘制任何内容，则包围矩形的位置 points 参数中第一个点的位置（浮点值将被截断），其宽度和高度将为0

提示：

**ValueError** -- 若 len(points) < 2 (必须至少有2个点)

**TypeError** -- 若 points 不是序列或 points 不包含数字对

Changed in pygame 2.0.0：添加了对关键字参数的支持。

# 8.pygame.draw.aaline()

功能：在给定Surface上绘制一条抗锯齿直线。

格式：

pygame.draw.aaline(画布,(r,g,b[a]),(start.x,start.y),(end.x,end.y),[blend=1])

**blend** (*int*) -- （可选）如果非零（默认值），线条将与Surface的现有像素阴影混合，否则将覆盖它们

**返回值：（**[***Rect***](https://www.pygame.org/docs/ref/rect.html#pygame.Rect)**）一个包围更改像素的矩形，如果没有绘制任何内容，则包围矩形的位置将是 start\_pos 参数值（浮点值将被截断），其宽度和高度将为0**

**提示：TypeError -- 如果 start\_pos or end\_pos 不是两个数字的序列**

# 9.pygame.draw.aalines()

功能：在给定的Surface上绘制连续的抗锯齿直线序列。

格式：

pygame.draw.aalines(画布,(r,g,b[a]),points,[blend=1])

**points (tuple(coordinate) or list(coordinate)) -- 一个含有2个或更多 (x, y) 坐标的序列，序列中的每个坐标必须是2个整数/浮点数格式的tuple/list/pygame.math.Vector2，同时相邻坐标将通过线段连接。【例如，对于点[(x1, y1), (x2, y2), (x3, y3)]线段将从(x1, y1)绘制到(x2, y2)并从(x2, y2) 绘制到(x3, y3)，此外，如果闭合参数为真，则另一线段将从(x3, y3)绘制到(x1, y1)】**

**blend (int) -- （可选）如果非零（默认值），线条将与Surface的现有像素阴影混合，否则将覆盖它们**

**返回值：（Rect）一个包围更改像素的矩形，如果没有绘制任何内容，则包围矩形的位置 points 参数中第一个点的位置（浮点值将被截断），其宽度和高度将为0**

**提示：**

**ValueError -- 若 len(points) < 2 (必须至少有2个点)**

**TypeError -- 若 points 不是序列或 points 不包含数字对**

**Changed in pygame 2.0.0：添加了对关键字参数的支持。**

**绘图模块的示例代码**

**# 导入名为'pygame'的函数库**

**import pygame**

**from math import pi**

**# 初始化游戏引擎**

**pygame.init()**

**# 以RGB格式定义将使用的颜色**

**BLACK = ( 0, 0, 0)**

**WHITE = (255, 255, 255)**

**BLUE = ( 0, 0, 255)**

**GREEN = ( 0, 255, 0)**

**RED = (255, 0, 0)**

**# 设置屏幕的高度和宽度**

**size = [400, 300]**

**screen = pygame.display.set\_mode(size)**

**pygame.display.set\_caption("Example code for the draw module")**

**#L循环，直到用户单击“关闭”按钮。**

**done = False**

**clock = pygame.time.Clock()**

**while not done:**

**# 将while循环限制为每秒最多10次**

**# Leave this out and we will use all CPU we can.**

**clock.tick(10)**

**for event in pygame.event.get(): # 捕获用户动作**

**if event.type == pygame.QUIT: # 若用户点击“关闭”**

**done=True # 标记已完成动作，以便退出此循环**

**# 所有绘图代码都发生在for循环之后，但是在 while done==False主循环中。**

**# 清除屏幕并设置屏幕背景**

**screen.fill(WHITE)**

**# 在屏幕上画一条从(0, 0)到(50, 30)的绿线**

**# 5像素宽**

**pygame.draw.line(screen, GREEN, [0, 0], [50,30], 5)**

**# 在屏幕上画三条黑线，每一条5像素宽**

**# 'False' 表示第一个和最后一个点没有连接**

**pygame.draw.lines(screen, BLACK, False, [[0, 80], [50, 90], [200, 80], [220, 30]], 5)**

**# 在屏幕上画一条从(0, 50)到(50, 80)的绿线**

**# 因为它是一条抗锯齿线，所以宽度为1像素**

**pygame.draw.aaline(screen, GREEN, [0, 50],[50, 80], True)**

**# 绘制矩形轮廓**

**pygame.draw.rect(screen, BLACK, [75, 10, 50, 20], 2)**

**# 绘制实心矩形**

**pygame.draw.rect(screen, BLACK, [150, 10, 50, 20])**

**# 画一个圆角矩形**

**pygame.draw.rect(screen, GREEN, [115, 210, 70, 40], 10, border\_radius=15)**

**pygame.draw.rect(screen, RED, [135, 260, 50, 30], 0, border\_radius=10, border\_top\_left\_radius=0, border\_bottom\_right\_radius=15)**

**# 使用矩形作为外边界绘制椭圆轮廓**

**pygame.draw.ellipse(screen, RED, [225, 10, 50, 20], 2)**

**# 使用矩形作为外边界绘制实心椭圆**

**pygame.draw.ellipse(screen, RED, [300, 10, 50, 20])**

**# 使用polygon命令绘制三角形**

**pygame.draw.polygon(screen, BLACK, [[100, 100], [0, 200], [200, 200]], 5)**

**# 画一条弧线作为椭圆的一部分**

**# 使用弧度确定要绘制的角度**

**pygame.draw.arc(screen, BLACK,[210, 75, 150, 125], 0, pi/2, 2)**

**pygame.draw.arc(screen, GREEN,[210, 75, 150, 125], pi/2, pi, 2)**

**pygame.draw.arc(screen, BLUE, [210, 75, 150, 125], pi,3\*pi/2, 2)**

**pygame.draw.arc(screen, RED, [210, 75, 150, 125], 3\*pi/2, 2\*pi, 2)**

**# 画一个圆**

**pygame.draw.circle(screen, BLUE, [60, 250], 40)**

**# 绘制一个象限圆**

**pygame.draw.circle(screen, BLUE, [250, 250], 40, 0, draw\_top\_right=True)**

**pygame.draw.circle(screen, RED, [250, 250], 40, 30, draw\_top\_left=True)**

**pygame.draw.circle(screen, GREEN, [250, 250], 40, 20, draw\_bottom\_left=True)**

**pygame.draw.circle(screen, BLACK, [250, 250], 40, 10, draw\_bottom\_right=True)**

**# 继续用画的东西更新屏幕。**

**# MUST在所有其他绘图命令之后发生**

**pygame.display.flip()**

**# Be IDLE friendly**

**pygame.quit()**

**整理自：**[**https://www.pygame.org/docs/ref/draw.html**](https://www.pygame.org/docs/ref/draw.html)