

0.什么是turtle

Turtle库是Python语言中一个很流行的绘制图像的函数库，想象一个**小乌龟**，在一个横轴为x、纵轴为y的坐标系原点，(0,0)位置开始，它根据一组函数指令的控制，在这个平面坐标系中移动，从而在它爬行的路径上绘制



了图形。

复习:import

1.基操上线

1.1 画布 canvas

```
turtle.screensize(canvwidth=None, canvheight=None, bg=None)
```

```
turtle.screensize(800,600, "green")
```

```
turtle.screensize() #返回默认大小(400, 300)
```

```
turtle.setup(width=0.5, height=0.75, startx=None, starty=None)
```

#参数: width, height: 输入宽和高为整数时, 表示像素; 为小数时, 表示占据电脑屏幕的比例,
(startx, starty): 这一坐标表示矩形窗口左上角顶点的位置, 如果为空, 则窗口位于屏幕中心。

#参数分别为画布的宽(单位像素), 高, 背景颜色。

1.2 画笔

```

turtle.pensize()
#设置画笔的宽度;

turtle.pencolor()
#没有参数传入，返回当前画笔颜色，传入参数设置画笔颜色，可以是字符串如"green", "red",也可以是RGB 3元组。

turtle.speed(speed)
#设置画笔移动速度，画笔绘制的速度范围[0,10]整数，数字越大越快。

```

```

#原谅色
turtle.colormode(255)
turtle.pencolor(126,200,174)

turtle.pencolor('#7ec8ae')

```

1.3 绘图指令

操纵海龟绘图有着许多的命令，这些命令可以划分为3种：一种为**运动命令**，一种为**画笔控制命令**，还有一种是**全局控制命令**。

1.4 画笔运动命令

命令	说明
<code>turtle.forward(distance)</code>	向当前画笔方向移动distance像素长度
<code>turtle.backward(distance)</code>	向当前画笔相反方向移动distance像素长度
<code>turtle.right(degree)</code>	顺时针移动degree°
<code>turtle.left(degree)</code>	逆时针移动degree°
<code>turtle.pendown()</code>	移动时绘制图形，缺省时也为绘制
<code>turtle.goto(x,y)</code>	将画笔移动到坐标为x,y的位置
<code>turtle.penup()</code>	提起笔移动，不绘制图形，用于另起一个地方绘制
<code>turtle.circle()</code>	画圆，半径为正(负)，表示圆心在画笔的左边(右边)画圆
<code>setx()</code>	将当前x轴移动到指定位置
<code>sety()</code>	将当前y轴移动到指定位置
<code>setheading(angle)</code>	设置当前朝向为angle角度
<code>home()</code>	设置当前画笔位置为原点，朝向东。
<code>dot(r)</code>	绘制一个指定直径和颜色的圆点

1.5 画笔控制命令

命令	说明
<code>turtle.fillcolor(colorstring)</code>	绘制图形的填充颜色
<code>turtle.color(color1, color2)</code>	同时设置 <pencolor=color1, fillcolor="color2</td"></pencolor=color1,>
<code>turtle.filling()</code>	返回当前是否在填充状态
<code>turtle.begin_fill()</code>	准备开始填充图形
<code>turtle.end_fill()</code>	填充完成
<code>turtle.hideturtle()</code>	隐藏画笔的turtle形状
<code>turtle.showturtle()</code>	显示画笔的turtle形状

1.5 全局控制命令

命令	说明
<code>turtle.clear()</code>	清空turtle窗口，但是turtle的位置和状态不会改变
<code>turtle.reset()</code>	清空窗口，重置turtle状态为起始状态
<code>turtle.undo()</code>	撤销上一个turtle动作
<code>turtle.isvisible()</code>	返回当前turtle是否可见
<code>stamp()</code>	复制当前图形
<code>turtle.write(s [,font=("font-name",font_size,"font_type")])</code>	写文本，s为文本内容，font是字体的参数，分别为字体名称，大小和类型；font为可选项，font参数也是可选项

1.6 其他命令

命令	说明		
turtle.mainloop()或turtle.done()	启动事件循环 -调用Tkinter的mainloop函数。 必须是乌龟图形程序中的最后一个语句。		
turtle.mode(mode=None)	设置乌龟模式（“standard”，“logo”或“world”）并执行重置。如果没有给出模式，则返回当前模式。		
	模式	初始龟标题	正角度
	standard	向右（东）	逆时针
	logo	向上（北）	顺时针
turtle.delay(delay=None)	设置或返回以毫秒为单位的绘图延迟。		
turtle.begin_poly()	开始记录多边形的顶点。当前的乌龟位置是多边形的第一个顶点。		
turtle.end_poly()	停止记录多边形的顶点。当前的乌龟位置是多边形的最后一个顶点。将与第一个顶点相连。		
turtle.get_poly()	返回最后记录的多边形。		

```
turtle.circle(radius, extent=None, steps=None)

#描述：以给定半径画圆

#参数：
radius(半径)#半径为正(负)，表示圆心在画笔的左边(右边)画圆；

extent(弧度) #(optional);

steps (optional) # (做半径为radius的圆的内切正多边形，多边形边数为steps)

circle(50) # 整圆；

circle(50,steps=3) # 三角形；

circle(120, 180) # 半圆
```

Further Reading

[更多详情](#)

[学会Python后，我画画是这样的](#)

[Turtle绘图篇](#)

[孤陋寡闻了吧？Python 居然可以做这30件神奇好玩的事情（附教程）](#)

[python表白实现代码（可视化与动画版）](#)

[情人节来了，教你个用 Python 表白的技巧](#)

[我用 Python 找出了删除我微信的所有人并将他们自动化删除了](#)