

第九章

python标准库概览

本章大纲

1.标准库: turtle库(必选)

2.标准库: random库(必选)、time库(可选)

1. turtle的概述

- 1、turtle（海龟）是Python重要的标准库之一，它能够进行基本的图形绘制。
- 2、turtle库绘制图形有一个基本框架：一个小海龟在坐标系中爬行，其爬行轨迹形成了绘制图形。对于小海龟来说，有“前进”、“后退”、“旋转”等爬行行为，对坐标系的探索也通过“前进方向”、“后退方向”、“左侧方向”和“右侧方向”等小海龟自身角度方位来完成。

2. turtle库的引用方式

第一种，`import turtle`，则对turtle库中函数调用采用`turtle.<函数名>()`形式。

第二种，`from turtle import *`，则对turtle库中函数调用直接采用`<函数名>()`形式，不在使用 `turtle.`作为前导。

第三种，`import turtle as t`，则对turtle库中函数调用采用更简洁的`t.<函数名>()`形式，保留字`as`的作用是将turtle库给予别名`t`。

3.窗口函数

`turtle.setup(width, height, startx, starty)` 窗体函数

作用：设置主窗体的大小和位置 参数：

width：窗口宽度，如果值是整数，表示的像素值；如果值是小数，表示窗口宽度与屏幕的比例；

height: 窗口高度，如果值是整数，表示的像素值；如果值是小数，表示窗口高度与屏幕的比例；

startx：窗口左侧与屏幕左侧的像素距离，如果值是None，窗口位于屏幕水平中央；

starty：窗口顶部与屏幕顶部的像素距离，如果值是None，窗口位于屏幕垂直中央

4.常用状态函数

函数	描述
pendown()	放下画笔
penup()	拿起画笔
pensize()	设置画笔线条的粗细
pencolor()	设置画笔颜色
color()	设置画笔和填充颜色
begin_fill()	填充图形前，调用该方法
end_fill()	填充图形结束
filling()	返回填充状态，True为填充
clear()	清空当前窗口

reset()	清空当前窗口，并重置位置
write(str,font=None)	输出font字体的字符串
screensize()	设置画布的长和宽
hideturtle()	隐藏画笔的turtle形状
showturtle	显示画笔形状
Isvisible()	如果画笔可见则显示true

turtle中的画笔（即小海龟）可以通过一组函数来控制，其中turtle.penup()和turtle.pendown() 是一组，它们分别表示画笔的抬起和落下，函数定义如下：

1、turtle.penup() 别名 turtle.pu(), turtle.up()

作用：抬起画笔，之后，移动画笔不绘制形状

参数：无

turtle.pendown() 别名 turtle.pd(), turtle.down()

作用：落下画笔，之后，移动画笔将绘制形状

参数：无

`turtle.pensize()`函数用来设置画笔尺寸

`turtle.pensize(width)` 别名 `turtle.width()`

作用：设置画笔宽度，当无参数输入时返回当前画笔宽度

参数： `width`：设置的画笔线条宽度，如果为None或者为空，函数则返回当前画笔宽度。

`turtle.pencolor()`函数给画笔设置颜色

`turtle.pencolor(colorstring)` 或者 `turtle.pencolor((r,g,b))`

作用：设置画笔颜色，当无参数输入时返回当前画笔颜色

参数： `colorstring`：表示颜色的字符串，例如："purple"、"red"、"blue"等 (r,g,b): 颜色对应RGB的01数值，例如：1, 0.65, 0

5.常用运动函数

函数	描述
forward()	沿着当前指定方向前进距离
backward()	沿着当前相反方向后退指定距离
setheading(angle)	设置当前朝向为angle角度
circle(r,e)	绘制一个半径r和角度e得圆或弧形
undo()	撤销画笔最后一步动作
right(angle)	向右旋转angle角度
left(angle)	向左旋转angle角度
goto(x,y)	移动到绝对坐标 (x,y) 处
speed()	设置画笔得绘制速度，参数为0-10
dot(d,color)	绘制一个直径d和颜色color的圆点
home()	设置当前画笔位置为原点，朝向东

`turtle.fd()`函数最常用，它控制画笔向当前行进方向前进一个距离

画笔状态函数

`turtle.fd(distance)` 别名 `turtle.forward(distance)`

作用：向小海龟当前行进方向前进`distance`距离

参数： `distance`：行进距离的像素值，当值为负数时，表示向相反方向前进。

`turtle.seth()`函数用来改变画笔绘制方向

`turtle.seth(to_angle)` 别名 `turtle.setheading(to_angle)`

作用：设置小海龟当前行进方向为`to_angle`，该角度是绝对方向角度值。参

数： `to_angle`：角度的整数值

`turtle.circle()`函数用来绘制一个弧形

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

作用：根据半径`radius`绘制`extent`角度的弧形。

参数：`radius`：弧形半径，当值为正数时，半径在小海龟左侧，当值为负数时，半径在小海龟右侧；`extent`：绘制弧形的角度，当不给该参数或参数为`None`时，绘制整个圆形。

1. random库的概述

使用random库主要目的是生成随机数

这个库提供了不同类型的随机数函数，其中最基本的函数是 `random.random()`，它生成一个 $[0.0, 1.0)$ 之间的随机小数，所有其他随机函数都是基于这个函数扩展而来。

2. random库常用函数

函数	描述
seed(a=None)	初始化随机数种子，默认值为当前系统时间
random()	生成一个[0.0, 1.0)之间的随机小数
randint(a, b)	生成一个[a,b]之间的整数
getrandbits(k)	生成一个k比特长度的随机整数
randrange(start, stop[, step])	生成一个[start, stop)之间以step为步数的随机整数
uniform(a, b)	生成一个[a, b]之间的随机小数
choice(seq)	从序列类型(例如：列表)中随机返回一个元素
shuffle(seq)	将序列类型中元素随机排列，返回打乱后的序列
sample(pop, k)	从pop类型中随机选取k个元素，以列表类型返回

1. time库概述

处理时间是程序最常用的功能之一，time库是Python提供的处理时间标准库。time库提供系统级精确计时器的计时功能，可以用来分析程序性能，也可让程序暂停运行时间。

```
>>> import time
>>> print(time.localtime())
time.struct_time(tm_year=2020, tm_mon=4, tm_mday=17,
tm_hour=10, tm_min=12, tm_sec=39, tm_wday=4, tm_yday=
108, tm_isdst=0)
```

time库的功能主要分为3个方面：时间处理、时间格式化和计时。

- 时间处理主要包括4个函数：time.time()、time.gmtime()、time.localtime()、time.ctime()。
- 时间格式化主要包括3个函数：time.mktime()、time.strftime()、time.strptime()。
- 计时主要包用到1个函数：time.sleep()

使用time.time()获取当前时间戳

```
>>> import time
>>> time.time()
1587089623.4982438
```

使用time.gmtime(secs)获取当前时间戳对应的struct_time对象

```
>>> import time
>>> time.gmtime()
time.struct_time(tm_year=2020, tm_mon=4, tm_mday=17,
tm_hour=2, tm_min=14, tm_sec=25, tm_wday=4, tm_yday=1
08, tm_isdst=0)
```

使用time.localtime(secs)获取当前时间戳对应的本地时间的struct_time对象

```
>>> import time
>>> print(time.localtime())
time.struct_time(tm_year=2020, tm_mon=4, tm_mday=17,
tm_hour=10, tm_min=12, tm_sec=39, tm_wday=4, tm_yday=
108, tm_isdst=0)
```

注意结果与gmtime的区别，UTC时间已自动转换为北京时间。

使用time.ctime(secs)获取当前时间戳对应的 易读字符串表示，内部会调用time.localtime() 函数以输出当地时间。

```
>>> import time
>>> time.ctime()
'Fri Apr 17 10:16:03 2020'
```

time库使用time.mktime()、time.strptime()、time.strftime()进行时间格式化。

使用time.mktime(t) 将struct_time对象t转换 为时间戳，注意t代表当地时间。struct_time 对象的元素如下

下标	属性	值
0	tm_year	年份，整数
1	tm_mon	月份[1, 12]
2	tm_mday	日期[1, 31]
3	tm_hour	小时[0, 23]
4	tm_min	分钟[0, 59]
5	tm_sec	秒[0, 61]
6	tm_wday	星期[0, 6]（0表示星期一）
7	tm_yday	该年第几天[1, 366]
8	tm_isdst	是否夏令时，0否，1是，-1未知

调用time.mktime(t)函数

```
>>> t = time.localtime()
>>> time.mktime(t)
1587089801.0
>>> time.ctime(time.mktime(t))
'Fri Apr 17 10:16:41 2020'
```

time.strftime()函数是时间格式化最有效的方法，几乎可以以任何通用格式输出时间。该方法利用一个格式字符串，对时间格式进行表达。

```
>>> lctime = time.localtime()
>>> lctime
time.struct_time(tm_year=2020, tm_mon=4, tm_mday=17,
tm_hour=10, tm_min=17, tm_sec=44, tm_wday=4, tm_yday=
108, tm_isdst=0)
>>> time.strftime('%Y-%m-%d %H:%M:%S', lctime)
'2020-04-17 10:17:44'
```

strftime()方法的格式化控制符

格式化字符串	日期/时间	值范围和实例
%Y	年份	0001~999,1995
%m	月份	01~12,8
%B	月名	January~December,April
%b	月名缩写	Jan~Dec,Apr
%d	日期	01~31,25
%A	星期	Monday~Sunday,Wednesday
%a	星期缩写	Mon~Sun,Wed
%H	小时（24小时）	00~23,18
%I	小时（12小时）	01~12,7
%p	上/下午	AM,PM,AM
%M	分钟	00~59,30
%S	秒	00~59,30

strptime()方法与strftime()方法完全相反，用于提取字符串中时间来生成struct_time对象，可以很灵活的作为time模块的输入接口

```
>>> timeString = '2020-04-17 10:17:44'
>>> time.strptime(timeString, '%Y-%m-%d %H:%M:%S')
time.struct_time(tm_year=2020, tm_mon=4, tm_mday=17,
tm_hour=10, tm_min=17, tm_sec=44, tm_wday=4, tm_yday=
108, tm_isdst=-1)
```