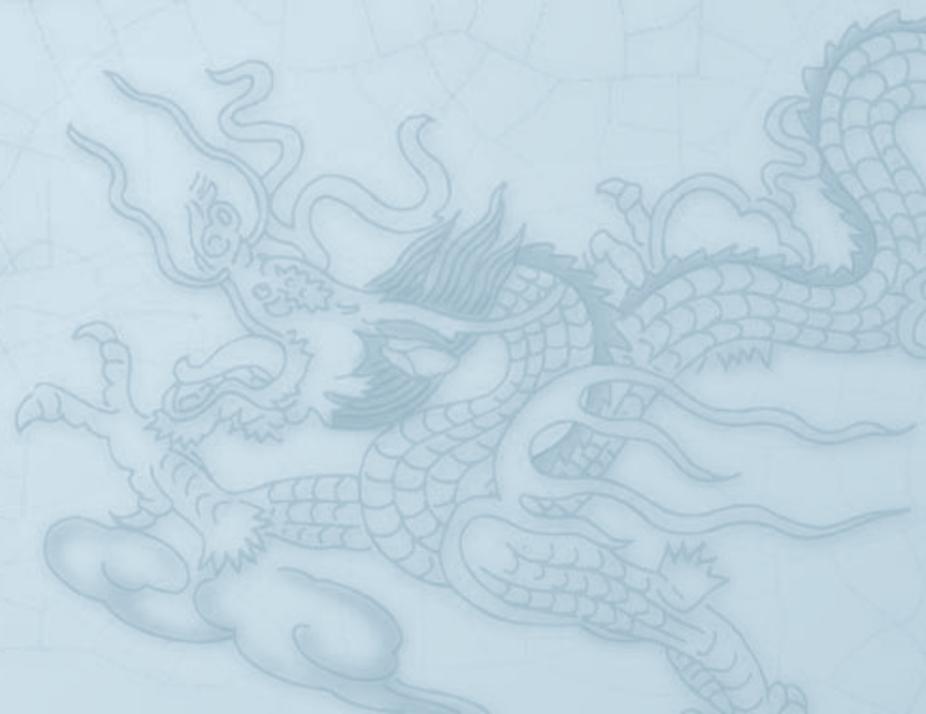


Python的基本图形绘制

--turtle库的应用



Python基本图形绘制

- 2.1 深入理解Python语言
- 2.2 实例: Python蟒蛇绘制
- 2.3 模块: turtle库的使用
- 2.4 turtle程序语法元素分析



深入理解Python语言

- 编程语言的江湖
- Python语言的特点
- "超级语言"的诞生



编程语言有哪些？

**C, java,C++, C#, CSS, Fortran, Go, HTML,
Java,JavaScript, Lisp, Lua, Matlab, Object C,
Pascal, Perl, PHP,PostScript, Python, Ruby,
Scala, SQL, Swift, VBA,VB.NET, Verilog,
VHDL, Visual Basic.....**

编程语言，也是一个江湖！

不同编程语言的目的和适用对象

编程语言	学习内容	语言本质	解决问题
C	指针、内存、数据类型	理解计算机系统结构	性能
Java	对象、跨平台、运行时	理解主客体关系	跨平台
C++	对象、多态、继承	理解主客体关系	大规模程序
VB	对象、按钮、文本框	理解人机交互逻辑	桌面应用
Python	编程逻辑、第三方库	理解问题求解	各类问题

各编程语言所处历史时期和使命不同，Python是**计算时代演进**的选择！

Python特点与优势

1. 语法简洁



- C代码量的10%
- 强制可读性
- 较少的底层语法元素
- 多种编程方式
- 支持中文字符

2. 生态高产



- >15万第三方库
- 快速增长的计算生态
- 避免重复工作量
- 开放共享
- 跨操作系统平台

Python的计算生态

- ◆ Python计算生态 = 标准库 + 第三方库
- ◆ 标准库：随解释器直接安装到操作系统中的功能模块
- ◆ 第三方库：需要经过安装才能使用的功能模块
- ◆ - 库Library、包Package、模块Module，统称模块

“超级语言”的特点

- ◆ 具有庞大计算生态，可以很容易利用已有代码功能
- ◆ 编程思维不再是刀耕火种，而是集成开发。
- ◆ **Python: 唯一的“超级语言”!**

2.2 实例：Python蟒蛇绘制

- ◆ 用程序绘制一条蟒蛇，蟒蛇的基本形状



用程序绘制一条蟒蛇

- ◆ - 问题1: 如何打开一个绘图窗体? 如何在窗体上绘图?
- ◆ - 问题2: 如何绘制直线如何绘制曲线?
- ◆ 开始编程啦



2.3 turtle库的应用

- ◆ - **turtle**库概述
- ◆ - **turtle**绘图窗体布局
- ◆ - **turtle**空间坐标体系
- ◆ - **turtle**角度坐标体系
- ◆ - **RGB**色彩体系

1.turtle库概述

- ◆ turtle(海龟)库是turtle绘图体系的python实现；
- ◆ - turtle绘图体系： 1969年诞生，主要用于程序设计入门级图形绘制；
- ◆ - Python语言的标准库之一。

turtle的原 (wan) 理 (fa)

turtle(海龟)是一种真实的存在；

- 可以想象有一只海龟，开始在窗体（画布）正中心；如果它在画布上游走就会留下轨迹；
- 走过的轨迹形成了绘制的图形；
- 海龟由程序控制，可以变换颜色、改变宽度等。

turtle的魅力

Python还能画画



Beijing



2.turtle绘图窗体布局

Setup()函数显示并设置绘图窗体的大小及窗体在屏幕上的相对位置:`turtle.setup(width,height,startx,starty)`



2.turtle绘图窗体布局



显示并设置画布的大小和背景色：

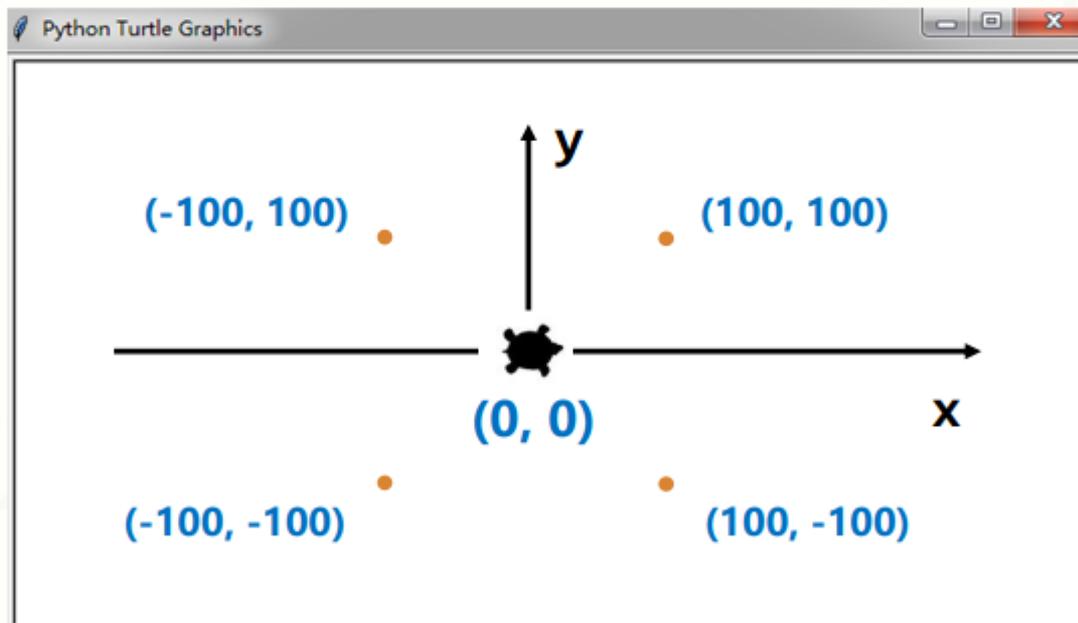
```
turtle.screenize(canvwidth=400,canvheight=400,bg="red")
```

注意：窗体和画布不是一个概念。

画布大于窗体时会出现滚动条。

3.turtle空间坐标体系—绝对坐标

绝对坐标

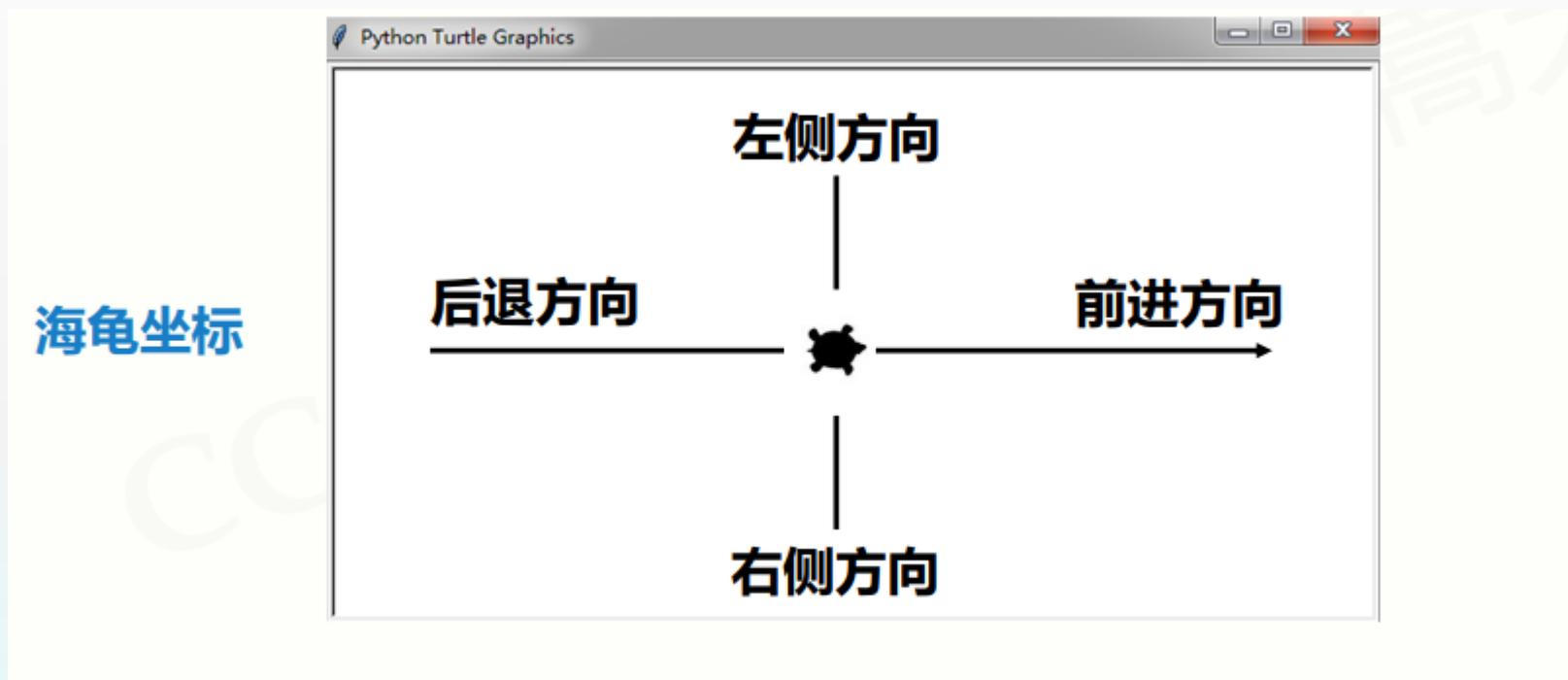


以画布正中心点为原点，向右直线为X正轴，向上为Y正轴，形成一个平面坐标系。画布上每一个点都有一对坐标值。

Turtle.goto()函数

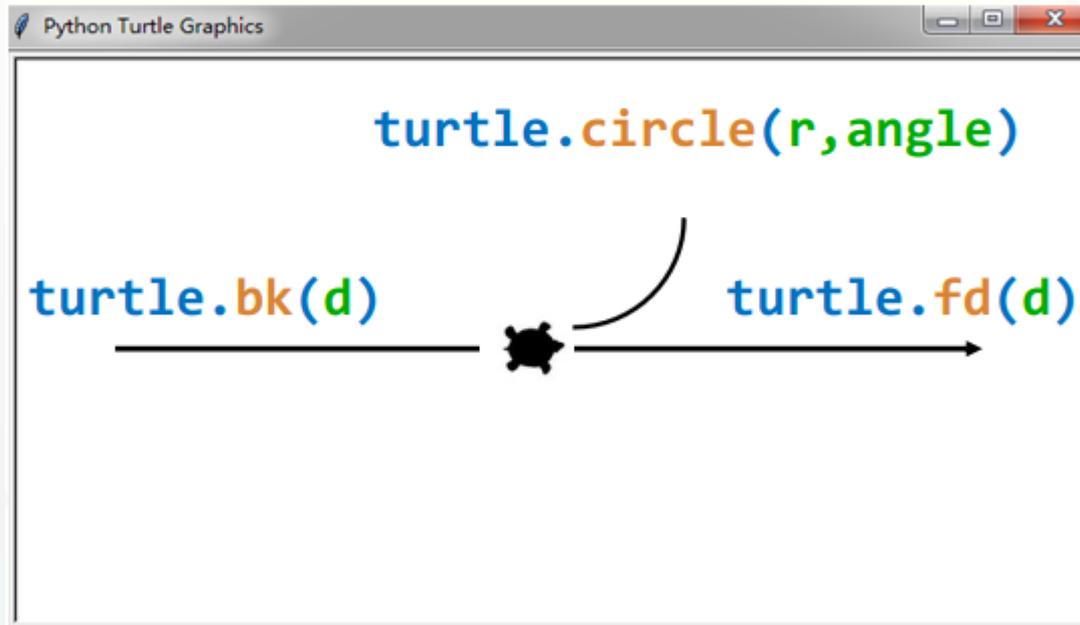
- ◆ 表示海龟在平面上沿直线移动的位置；
- ◆ 用法：turtle.**goto**(x,y)。
- ◆ x,y表示画布上的坐标位置，单位是像素。

turtle空间坐标体系—海龟坐标



海龟视角下的空间坐标

海龟运动函数—参照海龟坐标



`turtle.forward(d)` 或者 `turtle.fd(d)` # 直行到 d 像素处。

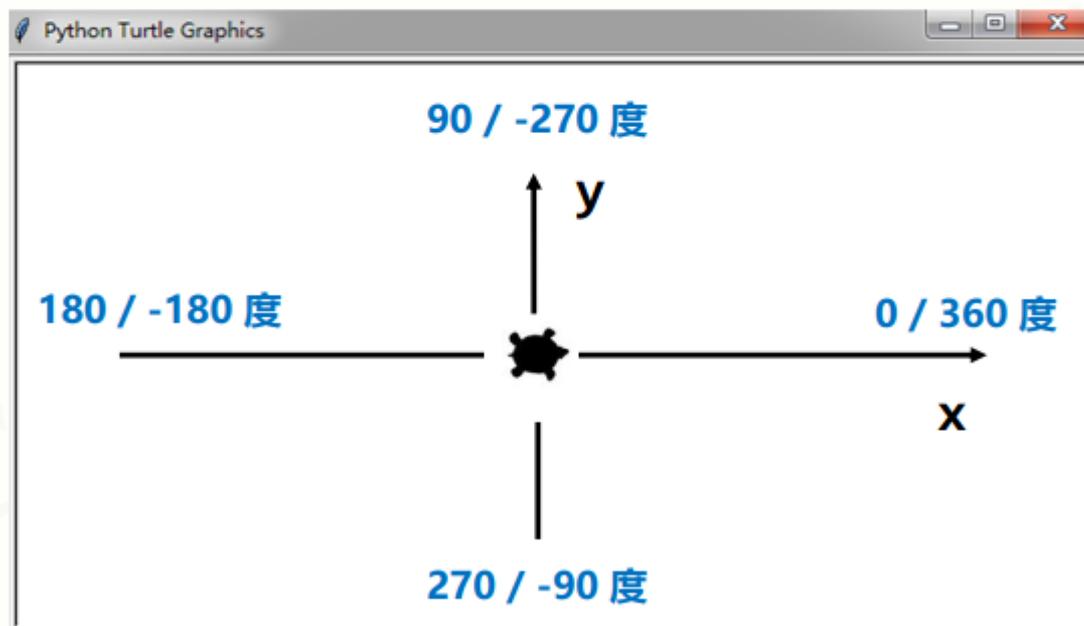
`turtle.circle(r, angle)` # 以海龟左侧，垂直海龟当前方向， r 像素处为圆点，以逆时针方向画个 $angle$ 度的弧。

当 r 为负值，则以海龟右侧，垂直海龟当前方向 r 像素出为圆心，以逆时针方向画个 $angle$ 度的弧。

当 $angle$ 为负值时，以顺时针方向画弧。当 $angle$ 省略时，画圆。

4.turtle角度坐标体系—绝对角度

绝对角度



Turtle的绝对角度坐标体系

turtle.seth()函数

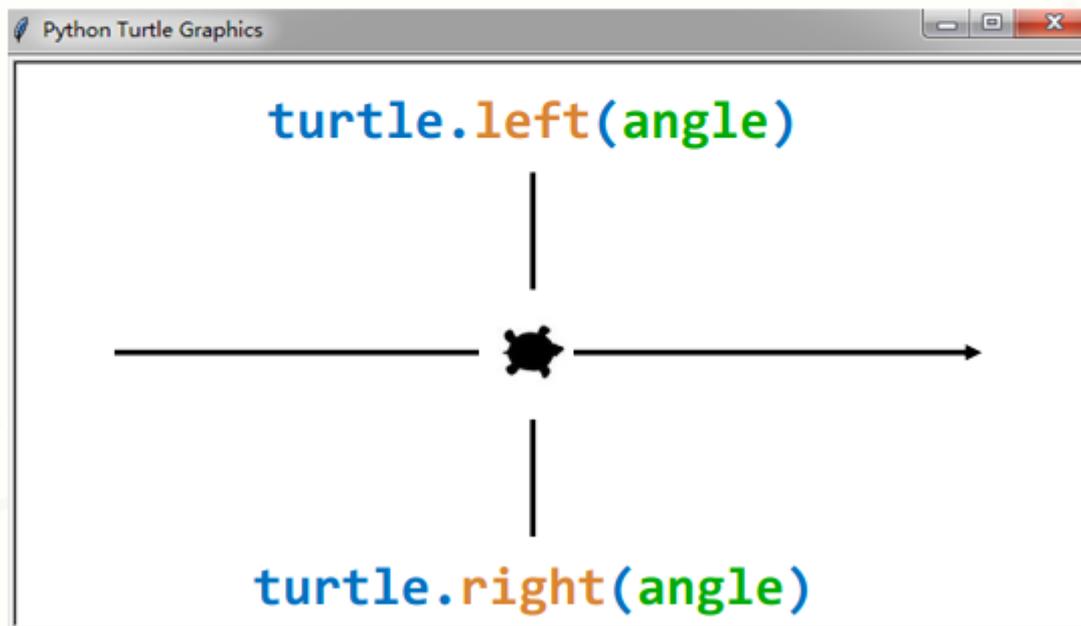
- ◆ **setheading(angle)** 或 **turtle.seth(angle)** : 改变海龟的行进方向, angle表示绝对的角度坐标值。
- ◆ **turtle.seth()** 函数只改变海龟的行进方向, 但不能行进。



seth(45)

Turtle角度坐标体系--海龟角度

海龟角度



`turtle.left(angle)`:在海龟**当前方向**上向左转一个角度。
`turtle.right(angle)`:在海龟**当前方向**上右转一个角度。

5.RGB色彩模式

由三种颜色构成的万物色

- ◆ - RGB指红绿蓝三个通道的颜色组合
- ◆ - 覆盖视力所能感知的所有颜色
- ◆ - RGB每色取值范围0-255整数或0-1小数。

turtle的RGB色彩模式

默认采用小数值，可切换为整数值。

◇ **turtle.colormode(mode)**

◇ 参数**mode**可以取值如下：

- **1.0**：RGB小数值模式

- **255**：RGB整数值模式

小结

- **turtle库的海龟绘图法**
- **turtle.setup()**调整绘图窗体在电脑屏幕中的布局
- **绝对坐标&海龟坐标**
- **绝对角度&海龟角度**
- **RGB色彩体系，整数值&小数值，色彩模式切换**

2.4 turtle程序语法元素分析

- 库引用方法
- Turtle画笔的属性设置
- turtle画笔控制函数
- turtle运动控制函数
- turtle方向控制函数
- range()函数

1.引用库

-使用import保留字完成，用法：

import <库名>

函数调用如下：

<库名>.<函数名>(<函数参数>)

import更多用法

- ◆ 使用from和import保留字共同完成：
- ◆ **from <库名> import <函数名>** 或
- ◆ **from <库名> import ***
- ◆ 采用<函数名>(<函数参数>)编码风格。

import的第三种用法

使用import和as保留字共同完成：

import <库名> as <库别名>

采用<库别名>.<函数名>(<函数参数>) 编码

2.turtle画笔的属性

- ◆ 画笔设置后一直有效，直至下次重新设置
- ◆ **--turtle.pensize(width) 别名 turtle.width(width)**
- ◆ 画笔宽度，海龟的腰围
- ◆ **--turtle.pencolor(color) color为颜色字符串或r,g,b值**
- ◆ 画笔颜色，海龟在涂装
- ◆ **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。**

3.turtle画笔控制函数

画笔操作后一直有效，一般成对出现。

-turtle.penup() 别名 turtle.pu()

抬起画笔，海龟在飞行

-turtle.pendown() 别名 turtle.pd()

落下画笔，海龟在爬行，能留下痕迹

3.turtle画笔控制函数

- ◆ `Turtle.pencolor(color)`:设置画笔的颜色
- ◆ `turtle.fillcolor(color)`: 绘制图形的填充颜色。
- ◆ `turtle.color(color1, color2)`: 同时设置
pencolor=color1, fillcolor=color2
- ◆ `turtle.begin_fill()`: 准备开始填充图形
- ◆ `turtle.end_fill()`: 填充完成
- ◆ `turtle.done()`:停止画笔绘制,但绘图窗体不关闭

Turtle的颜色

`turtle.pencolor(color)` 或者 `turtle.color(color1,color2)`

的参数可以有三种形式

颜色字符串：**`turtle.pencolor("purple")`**

RGB的小数值：**`turtle.pencolor(0.63, 0.13, 0.94)`**

RGB的整数值：**`turtle.pencolor(125,125,0)`**

3.turtle运动控制函数

◇ 控制海龟行进：走直线 & 走曲线

turtle.forward(d) 别名 **turtle.fd(d)** 向前行进，海龟走直线

- **d**: 行进距离，可以为负数

turtle.circle(r, angle) 根据半径r绘制angle角度的弧形

- **r**: 默认圆心在海龟左侧r距离的位置；

- **angle**: 绘制角度，默认值是360度整圆

4.turtle方向控制函数

控制海龟面对方向: 绝对角度 & 海龟角度

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

改变海龟行进的方向

-angle: 行进方向的绝对角度。

`turtle.left(angle)` 海龟向左转

`turtle.right(angle)` 海龟向右转

--angle: 在海龟当前行进方向上旋转的角度

5. Turtle文本输出函数

- ◆ `write(arg,align='left',font=('arial',8,'normal'))`
- ◆ `arg`表示要输出的文本。
- ◆ `align`:定位海龟和文字的位置。
- ◆ `font`:设置字体、字号等。

6.range()函数

产生循环计数序列

- ◆ `range(N)`: 产生一个0-N的半闭包区间
- ◆ `range(M,N)`: 产生一个M到N的半闭包区间
- ◆ `range(M,N,K)`: 产生一个M到N，步长为k的半闭包区间

小结

- 库引用: **import**、 **from...import**、 **import...as**
- penup()**、 **pendown()**、 **pensize()**、 **pencolor()**
- **fd()**、 **circle()**、 **seth()**、 **left()**、 **right()**
- 循环语句: **for**和**in**、 **range()**函数