第 2 周：Python 基本图像绘制

2.1 深入理解 Python 语言

1.编程语言的多样核心

（1）C 语言

学习内容：指针、内存、数据类型；

语言本质：理解计算机系统结构；

解决问题：性能。

（2）Java 语言

学习内容：对象、跨平台、运行时；

语言本质：理解主客体关系；

解决问题：跨平台。

（3）C++语言

学习内容：对象、多态、继承；

语言本质：理解主客体关系；

解决问题：大规模程序。

（4）Python

学习内容：编程逻辑、第三方库；

语言本质：理解问题求解；

解决问题：各类问题。

2.Python 语言的特点

（1）Python 语言是**通用**、**脚本**、**开源**、**跨平台**、**多模型**语言。

（2）Python 语法简洁（具有强制可读性，Python 具有较少的底层语法元素，支持多种编程方式，支持中文字符）、生态高产（具有大量的第三方库，因此有快速增长的计算生态，可以避免重复，并且开放共享、支持跨平台）。

2.2 Python 蟒蛇绘制

1.问题分析

用程序绘制一条蟒蛇。

第一步是设计蟒蛇的基本形状。

-问题 1：计算机绘图的原理是什么？（一段程序为何能够产生窗体？为何能在窗体上绘制图形？）

-问题 2：Python 蟒蛇绘制从哪里开始呢？（如何绘制一条线？如何绘制一个弧形？如何绘制一条蟒蛇？）

2.实例编写

代码：

```
import turtle//程序关键。引入了一个绘图库就是 turtle
```

```
turtle.setup(650,350,200,200)//定义窗口的大小和位置
turtle.penup()//首先将海龟抬起来
turtle.fd(-250)//向左移 250 像素，不留痕迹
turtle.pendown()//落下笔
turtle.pensize(25)//设置笔的大小
turtle.pencolor("red")//设置笔的颜色
turtle.seth(-40)//设置角度为-40 度
//设置圆心为左侧 40 像素，角度 80，然后再右侧 40 像素绘制
for i in range(4):
    turtle.circle(40,80)
    turtle.circle(-40,80)
turtle.circle(40,80/2)//变为直线
turtle.fd(40)//继续前行
turtle.circle(16,180)//然后转个半圆上来
turtle.fd(40*2/3)//往前走走
turtle.done()//结束
```

运行结果：

图 1

3.举一反三

Python 蟒蛇绘制是各类图形绘制问题的代表，学会了该绘制，就可以实现圆形绘制、五角星绘制、国旗绘制、机器猫绘制。

2.3 模块 1：turtle 模块的使用

1.turtle 库基本介绍

是 turtle 绘图体系的 python 体现，主要用于程序设计入门，属于 python 的标准库。

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。

（1）turtle 的原理：

有一只海龟，在窗体正中心，在画布上游走，走过的轨迹行程了图形。海龟轨迹的颜色、大小等都是由程序控制。

2.turtle 绘图窗体布局

（1）turtle 的绘图窗体

turtle 的一个画布空间最小单位是像素，窗体与屏幕的关系如图 2 所示。

图 2

turtle.setup(width, height, startx, starty)

该函数设置窗体的大小和位置，4 个参数中的后两个可选，该函数不是必须的。

3.turtle 空间坐标体系

包括**绝对坐标**和**相对坐标**。

绝对坐标体系中，绘图窗体的中心坐标是（0,0），然后向右、向上为正方向分别产生 x

轴和 y 轴。

turtle.goto(x, y)

让海龟直接到某个位置，不考虑海龟的初始位置。

海龟坐标（相对坐标）是指以海龟本身视角来判断前后左右。常用函数有：

turtle.circle(r, angle)//以左侧某一个点为圆心曲线运行

turtle.bk(d)//反方向

turtle.fd(d)//正向运行

4.turtle 角度坐标体系

绝对角度坐标体系如图 3 所示，我们可以使用以下代码来设置角度：

turtle.seth(angle)//改变海龟行进方向，只改变方向不运动

图 3

海龟角度，即以海龟的视角来判断角度，用到两个语句：

turtle.left(angle)//海龟向左转角度

turtle.right(angle)//海龟向右角度

2.4 turtle 语言元素分析

1.库引用与 import

若我们不想使用 turtle.<函数名>，而是直接想使用<函数名>，那么可以使用：

from turtle import *

使用原来方式的好处是不会出现函数重名，第二种的好处是更简洁。

也可以使用

import <库名> as <库别名>

调用是，使用：

<库别名>.<函数名>

2.turtle 画笔控制函数

在例子代码中，以下代码使用了画笔控制函数：

turtle.penup()

turtle.pendown()

turtle.pensize(25)

turtle.pencolor("red")

画笔操作后一直有效，一般成对出现即包括 turtle.penup()（别名 turtle.pu()）和 turtle.pendown()（别名 turtle.pd()）。当执行 penup 的时候，海龟在飞行，因此不会画出轨迹；执行 pendown，海龟落下，因此后来再画就有轨迹。

turtle.pensize(width)别名 turtle.width(width)，设置画笔的宽度，设置后一直有效，直到下次重新设置。

turtle.pencolor(color)绘制画笔颜色，color 参数有三种形式：

（1）字符串：如 turtle.pencolor("red");

（2）RGB 小数值：turtle.pencolor(0.63,0.13,0.94)

（3）RGB 元组值：turtle.pencolor((0.63,0.19,0.94))

3.turtle 运动控制函数

控制海龟行进方向：走直线或走曲线。

turtle.forward(d)//向前行进，海龟走之间

turtle.fd(d)//别名。同时，d 为行进距离，可以为负数
turtle.circle(r, extent=None)//根据 r 绘制 extent 角度的弧形
//r：默认圆心在海龟左侧 r 距离的位置

4.turtle 方向控制函数

turtle.setheading(angle)//别名 turtle.seth(angle)
//改变行进方向，angle 改变角度，此处角度为绝对角度

5.基本循环语句

已经学习过，不再整理

这里注意一件事，print("Hello:", i)这样输出的结果 hello:和数字之间是又空格的，要会使用！

6."Python 蟒蛇绘制"代码分析