# Python实验报告2

## 实验二：

简单Python程序

## 实验目的：

熟悉Python语言基本元素，了解Python语言函数库Turtle。

## 实验要求：

1. 熟悉掌握Python语言基本元素，能仿照例题温度转换程序编写汇率兑换程序。

2. 基本掌握Turtle库的常用函数和用法，能绘制简单的图形。

## 实验过程：

从实际问题入手，以简单的温度转换程序为例，逐行逐句地分析Python语言的基本元素。通过讲解Python蟒蛇绘制实例，介绍Python语言函数库turtle及其基本用法。

2.1 实例1的修改。改造实例代码1.1，采用eval(input(<提示内容>))替换现有输入部分，并使输出的温度值为整数。

代码改写如下：

*SymbolStr = input("请输入温度体系符号: ")*

*If SymbolStr in ['F', 'f']:*

*t = eval(input("请输入温度值: "))*

*C = (t - 32) / 1.8*

*print("转换后的温度是:{:.0f}C".format(C))*

*elif SymbolStr in ['C', 'c']:*

*t = eval(input("请输入温度值: "))*

*F = t \* 1.8 + 32*

*print("转换后的温度是:{:.0f}F".format(F))*

*else:*

*print("输入符号错误")*

2.2 汇率兑换程序。按照温度转换程序的设计思路，按照1美元=6人民币汇率编写一个美元和人民币的双向兑换程序。

编写代码：

*exchangeRate = 6*

*currencyStr = input("请输入带货币符号的金额(美元$,人民币￥): ")*

*if currencyStr[-1] == "$":*

*d = eval(currencyStr[0:-1])*

*y = d \* exchangeRate*

*print("{}$ = {} ￥".format(d, y))*

*elif currencyStr[-1] == '￥':*

*y = eval(currencyStr[0:-1])*

*d = y / exchangeRate*

*print("{}￥ = {:.2f}$".format(y, d))*

*else:*

*print("输入有错误")*

2.3实例2的修改。改造实例代码2.1，绘制一条彩色蟒蛇，即在绘制Python蟒蛇的每个小段时，画笔的绘制颜色会发生变化。

编写代码：

import turtle

snakeColor = ["black","purple","red","green","blue"]

turtle.setup(650, 350, 200, 200)

turtle.penup()

turtle.fd(-250)

turtle.pendown()

turtle.pensize(25)

turtle.seth(-40)

for i in range( 4):

turtle.pencolor(snakeColor[i])

turtle.circle(40, 80)

turtle.circle(-40, 80)

i = i + 1

turtle.pencolor(snakeColor[i % 5 ])

turtle.circle(40, 80/ 2)

turtle.fd(40)

i = i + 1

turtle.pencolor(snakeColor[i % 5])

turtle.circle(16, 180)

turtle.fd(40 \* 2/3)

2.4 等边三角形的绘制。使用turtle库中的turtle.fd()函数和turtle.seth()函数绘制一个等边三角形，效果如图。

编写代码：

import turtle

turtle.setup(650, 550, 0, 0)

turtle.seth(0)

turtle.fd(250)

turtle.seth(120)

turtle.fd(250)

turtle.seth(240)

turtle.fd(250)

2.5 叠加等边三角形的绘制。使用turtle库中的turtle.fd()函数和turtle.seth()函数绘制一个叠加等边三角形，效果如图。

编写代码：

import turtle as t

t.setup(600,400,100,100)

t.seth(240)

t.fd(100)

t.left(120)

t.fd(100)

t.left(120)

t.fd(100)

t.right(120)

t.fd(100)

t.right(120)

t.fd(100)

t.left(120)

t.fd(100)

t.left(120)

t.fd(200)

t.left(120)

t.fd(100)

t.seth(0)

t.done()

2.6 无角正方形的绘制。利用turtle库绘制一个没有角的正方形，效果如图所示。

编写代码：

import turtle

turtle.setup(400,400,40,40)

theta = 270

turtle.penup()

for i in range(4):

turtle.seth(theta)

turtle.fd(10)

turtle.pendown()

turtle.fd(40)

turtle.penup()

turtle.fd(10)

theta = theta + 90

2.7 六角形的绘制。利用turtle库绘制一个六角形，效果如图。

编写代码：

import turtle

import math

def drawTriangle(edge, theta, increment):

for i in range(3):

turtle.seth(theta)

turtle.fd(edge)

theta = theta + increment

px = -150

py = 0

turtle.setup(800, 600)

turtle.penup()

turtle.setx(px)

turtle.sety(py)

turtle.pendown()

drawTriangle(180, 30, 240)

turtle.penup()

turtle.setx(px + 60 \* math.cos(math.pi / 6))

turtle.sety(py - 90)

turtle.pendown()

drawTriangle(180, 30, 120)

## 实验思考：

对于python程序的编写要使用英文编写。要回正确引用turtle函数库。