# Python实验报告2

## 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：

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：

TempStr=input("请输入带有符号的温度值：")

if TempStr[-1] in ['F','f']:

    C=(eval(TempStr[0:-1])-32)/1.8

    print("转换后的温度是{:.0f}C".format(C))

elif TempStr[-1] in ['c','C']:

    F=1.8\*eval(TempStr[0:-1])+32

print("转换后的温度是{:.0f}".format(F))

else:

print("输入格式错误")

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：

TempStr=input("请输入带有符号的金钱数额:")

if TempStr[-1] in ['D','d']:

    Y=6\*eval(TempStr[0:-1])

    print("转换后的钱数是{:.2f}Y".format(Y))

elif TempStr[-1] in ['Y','y']:

    D=eval(TempStr[0:-1])/6

    print("转换后的钱数是{:.2f}D".format(D))

else:

    print("输入错误")

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.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。

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：**

认真了解python语言的基本要素并正确使用