# Python前三次实验报告

## 实验一、熟悉Python语言开发环境

实验目的：了解Python语言，熟悉课程所使用的Python语言开发环境

实验要求：能编辑、运行给定的代码，熟练掌握集成开发环境

实验学时：2学时

实验内容：

1. 完成教材29页程序练习题1.1-1.6

1.1

str1 = input("请输入一个人的名字：")

str2 = input("请输入一个国家名字：")

print("世界这么大，{}想去{}看看。".format(str1,str2))

1.2

adius = 25 # 圆的半径是25

area = 3.1415 \* radius \* radius # 输入计算圆面积的公式

print(area)

print("{:.2f}".format(area)) # 只输出两位小数

1.3

from datetime import datetime # 引用datetime 库

now = datetime.now() # 获得当前日期和时间信息

print(now)

now.strftime("%x") # 输出其中的日期部分

now.strftime("%X") # 输出其中的时间部分

1.4

for i in range(1,10):

for j in range(1,i+1):

print("{}\*{}={:2} ".format(j,i,i\*j), end='')

print('')

小结：这6道题的代码展示了Python语言的一些基本语法，我们将在后续课程进行学习，要注意python语言中的缩进很重要，一旦写错了，程序就会出语法错误.

2、完成程序练习题1.7-1.8

小结：这两道题展示了Python语言中的绘图功能，可以修改程序中的数据，观察绘制出的图形的不同点。

## 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

实验学时：2学时

实验内容：

#e1.1TempConvert.py

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{:.2f}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{:.2f}F".format(F))

else:

print("输入格式错误")

#e1.2TempConvert.py

TempStr = input("请输入带有符号的温度值: ")

while TempStr[-1] not in ['N','n']:

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

C = (eval(TempStr[0:-1]) - 32)/1.8

print("转换后的温度是{:.2f}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{:.2f}F".format(F))

else:

print("输入格式错误")

TempStr = input("请输入带有符号的温度值: ")

#e1.3TempConvert.py

def tempConvert(ValueStr):

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

C = (eval(ValueStr[0:-1]) - 32)/1.8

print("转换后的温度是{:.2f}C".format(C))

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

F = 1.8\*eval(ValueStr[0:-1]) + 32

print("转换后的温度是{:.2f}F".format(F))

else:

print("输入格式错误")

TempStr = input("请输入带有符号的温度值: ")

tempConvert(TempStr)

## 实验三、基本数据类型

实验目的：掌握数值运算操作和字符类型转换函数

实验要求：

1、掌握Python内置的数值运算和字符函数；

2、掌握字符串类型及其操作和格式化

实验学时：2学时

实验内容：

#e3.1DayDayUp365.py

import math

dayup = math.pow((1.0 + 0.001), 365) # 每天提高0.001

daydown = math.pow((1.0 - 0.001), 365) # 每天荒废0.001

print("向上: %.2f, 向下: %.2f."%(dayup, daydown))

#e3.2DayDayUp365.py

import math

dayup = math.pow((1.0 + 0.005), 365) # 每天提高0.005

daydown = math.pow((1.0 - 0.005), 365) # 每天荒废0.005

print("向上: %.2f, 向下: %.2f."%(dayup, daydown))