

第二章：Python 程序实例解析

1. 请使用 turtle 库函数绘制一个包含 9 个同心圆的靶盘

```
import turtle
def bapan(radius):
    turtle.penup()
    turtle.goto(0,-radius)
    turtle.pendown()
    turtle.circle(radius,360)

turtle.pensize(2)
turtle.pencolor('red')
for i in range(10,100,10):
    bapan(i)
turtle.done()
```

程序练习题 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正数

```
TempStr = eval('input("请输入需要转换的温度：")')
if TempStr[-1] in ['F','f']:
    C=int((eval(TempStr[0:-1])-32)/1.8)
    print("转换后的温度速{:d}C".format(C))
elif TempStr[-1] in ['C','c']:
    F=int(1.8*eval(TempStr[0:-1])+32)
    print("转换后的温度速{:d}F".format(F))
else:
    print("ERROR")
```

程序练习题 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=6.36 人民币汇率编写一个美元和人民币双向兑换程序。

```
value = input("请输入要转换的数额, 务必在输入数值前输入 USD 或 RMB 以表明货币类型:\n")
result = 0
if value[0:3] == "RMB":
    result = eval(value[3:]) / 6.36
    print("USD{:.2f}".format(result))
elif value[0:3] == "USD":
    result = (eval(value[3:])* 6.36)
    print("RMB{:.2f}".format(result))
```

程序练习题 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 蟒蛇的每一小段，画笔的绘制颜色会发生变化。

```
import turtle
color=['grey','darkgreen','gold','violet']
turtle.setup(650, 350, 200, 200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(14)
turtle.seth(-40)
for i in range(4):
    turtle.pencolor(color[i])
    turtle.circle(40, 80)
    turtle.circle(-40, 80)
turtle.pencolor("red")
turtle.circle(40, 80/2)
turtle.pencolor("blue")
turtle.fd(40)
turtle.pencolor("pink")
turtle.circle(16, 180)
turtle.fd(40 * 2/3)
turtle.done()
```

程序练习题 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。

```
import turtle
def Dengbian(N):
    turtle.fd(N)
    turtle.seth(120)
    turtle.fd(N)
    turtle.seth(240)
    turtle.fd(N)
    turtle.done()
Bc= input("请输入边长")
Dengbian(eval(Bc))
```

程序练习题 2.8 正方形螺旋线的绘制。利用 turtle 库绘制一个正方形螺旋线。

```
import turtle
turtle.pensize(2)
turtle.pencolor("blue")
for i in range(90):
    turtle.forward(2*i)
    turtle.left(90)
turtle.done()
```

第三章：基本数据类型

实例 3.2 反转：凯撒密码解密

```
miwen = input("请输入密文： ")
for p in miwen:
    if ord("a")<=ord(p)<=ord("z"):
        print(chr(ord("a")+(ord(p)-ord("a")-3)%26),end='')
    else:
        print(p,end='')

```

习题 3.1 重量计算。月球上物体的体重是地球上的 16.5%没加入你在地球上每年增长 0.5kg，编写程序输出未来 10 年你在地球和月球上的体重状况。

```
weight = input("请输入你的体重（以千克为单位） ")
weight = eval(weight)
wei_mon = weight*0.165
print("你目前在地球上的体重为{:.2f}kg， 你目前在月球的体重为{:.2f}kg"\
      .format(weight,wei_mon))
for i in range(11):
    weight +=0.5
wei_mon = weight*0.165
print("你 10 年后在地球上的体重为{:.2f}kg， 你 10 年后在月球的体重为{:.2f}kg"\
      .format(weight,wei_mon))

```

习题 3.4 回文数判断。设 n 是以任意自然数，如果 n 的个位数字反向排列所得到的结果与 n 相等，则 n 被称为回文数。从键盘输入一个 5 位数字，判断其是不是回文数。

```
num = input("请输入一个五位数字")
if(len(num)!=5):
    print("不是回文数")
else:
    if((eval(num[0])==eval(num[4])) and (eval(num[1])==eval(num[3]))):
        print("是回文数")
    else:
        print("不是回文数")

```

第四章：程序的控制结构

习题 4.1（按照习题 4.4 和 4.5 来做）猜数字游戏。让计算机随机产生一个预设数字，范围在 0-9 以内，让用户通过键盘输入所猜的数字，如果大于预设的数，显示“遗憾，太大了”；如果小于预设的数，显示“遗憾，太小了”，如此循环，直到猜中该数，显示“预测 N 次，终于猜中了！”。当用户输入的不是整数时，程序会自动退出执行并显示“输入内容必须为整数！”

```
from random import *
def input_num():
    you_num = input("请输入您猜的数字")

```

```

    you_num = eval(you_num)
    return you_num

num = randint(0,9)
you_num = input_num()
print(num)
n=1
try:
    while(num!=you_num):
        n += 1
        if(you_num<num):
            print("遗憾, 太小了")
        if(you_num>num):
            print("遗憾, 太大了")
        you_num = input_num()
    print("预测{}次, 终于猜中了! ".format(n))
except:
    print("输入内容必须为整数! ")

```

习题 4.2 统计不同字符个数。用户从键盘输入一行字符, 统计并输出其中英文字符、数字和其它字符的个数。

```

import string
from collections import namedtuple
def str_count(s):
    count_en = count_dg = count_sp = count_zh = count_oth = 0
    s_len = len(s)
    for c in s:
        #英文
        if c in string.ascii_letters:
            count_en += 1
        #数字
        elif c.isdigit():
            count_dg += 1
        #空格
        elif c.isspace():
            count_sp += 1
        #中文
        elif c.isalpha():
            count_zh += 1
        #其它特殊字符
        else:
            count_oth += 1
    total_chars = count_zh + count_en + count_sp + count_dg + count_oth
    if total_chars == s_len:
        return namedtuple('Count', ['total', 'zh', 'en', 'space', 'digit', 'other'])(s_len, count_zh,

```

```
count_en, count_sp, count_dg, count_oth)
    else:
        print('Something is wrong!')
        return None
    return None
```

```
s = input("请输入你想统计的字符串。")
count = str_count(s)
print(s, end='\n\n')
print('该字符串共有 {} 个字符, 其中有 {} 个汉字, {} 个英文, {} 个空格, {} 个数字, {} 个
标点符号。'.format(count.total, count.zh, count.en, count.space, count.digit, count.othher))
```

习题 4.6 羊车门问题。有 3 扇关闭的门，一扇门后面听着汽车，其余门后是山羊，只有主持人直到每扇门后面有什么，参赛者可以选择一扇门，在开启它之前，主持人会开启另外一扇门，露出门后的山羊，然后允许参赛者更换自己的选择。请问：才塞着更换选择后能否增加猜中汽车的机会？其使用 random 库对这个随机事件进行预测，分别给出参赛者改变选择和坚持选择获胜的概率。

```
from random import*
TIMES = input("请输入你想做的测试次数")
TIMES = eval(TIMES)
my_first_choice_n=0#初始化不改选择的次数
my_change_choice_n=0#初始化更改选择的次数
for i in range(TIMES):
    a="羊 1","羊 2","车"
    car_inDoor=choice(a)
    my_guess=choice(a)
    if car_inDoor==my_guess:
        my_first_choice_n+=1
    else:
        my_change_choice_n+=1
print("不改选择:{}".format(my_first_choice_n/TIMES))
print("更改选择:{}".format(my_change_choice_n/TIMES))
```

习题 4.7 用异常处理改造实例 1，使其能够接受并处理用户的任何输入。

```
TempStr = input("请输入带有符号的温度值：")
try:
    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=(eval(TempStr[0:-1]))*1.8+32# [0:-1]代表从第一位到倒数第二位
        print("转换后的温度是{:.2f}F".format(F))
    else:
        print("Format Error!!")
```

```
except:  
    print("输入错误, 请严格按照规定输入信息!")
```