Python 语言程序设计基础(第 2 版)

全答案

（Ver. 2.0，2018 年 5 月）

嵩天 礼欣 黄天羽著

（本文档由该书原作者提供，有任何修改意见请反馈：黄天羽 huangtianyu@bit.edu.cn。）

目录

第 1 章 程序设计基本方法

1.1 计算机的概念

[1.1]: 计算机是根据指令操作数据的设备，它的两个显著特点是功能性和可编程性。

[1.2]: 吉尔德定律（Gilder’s Law）：主干网的带宽每 6 个月增长一倍；

梅特卡尔夫定律（Metcalfe’s Law）：网络的价值同用户数量的平方成正比；

库梅定律（Koomey’s Law）：每 18 个月相同计算量所需要消耗的能量会减少一半；

尼尔森定律（Nielsen’s Law）：用户带宽将以平均每年 50%的增幅增长。

[1.3]: 近十年提出的技术名词有普适计算、云计算、物联网、深度学习、量子计算、大数据、区块链、边缘计算、软件定义网络等。

1.2 程序设计语言

[1.4]: 机器语言，也就是二进制代码语言，全部由 0 和 1 组成。

[1.5]: 编译执行： 优点：编译产生的目标代码执行更快，在同类型操作系统上使用灵活；

缺点：对不同类型操作系统支持较差，比如 windows 和 linux；

解释执行： 优点：可在不同类型操作系统上运行，源代码保留，方便纠错与维护；

缺点: 执行效率较低，源代码保留使程序容易遭到破坏。

[1.6]: 工作中处理数据、方便及时获取信息、创造新的工作机会等。

1.3 Python 语言概述

[1.7]: 降低学习成本、有利于程序演进、推动了互联网的进步。

[1.8]: 平台无关，粘性扩展，强制可读，支持中文，模式多样，类库丰富等。

[1.9]: print("祖国，你好")

1.4 Python 语言开发环境配置

[1.10]: 在 print()参数列表中添加 end = ""，表示以空字符结尾，替换默认的换行结尾。

[1.11]: 可以使用 turtle 库绘制图形

[1.12]: 可以使用 datetime 或 time 库来获取日期时间

1.5 程序的基本编写方法

[1.13]: Input: 给出一个问题及回答者的答案

Process: 将回答者答案与人类答案进行比较

Output: 回答者是人或计算机

[1.14]: Python 语言能够帮助求解问题中的计算部分。

[1.15]: 调试指排除程序错误，此时程序输出是不正确的。测试指在程序正确输出后对其他特性诸如性能，安全性进行进一步探究和改进，此时程序的输出是正确的。

[1.16]: B

1.6 Python 语言的版本更迭

[1.17]: Python 2 输出是 print "祖国，你好"，Python 3 输出是 print("祖国，你好")。

[1.18]: Python 2 中的 input()返回类型取决于输入类型，Python 3 中无论输入什么，input()返回的都是字符串。

[1.19]: (1) 观察它们的 print 用法

(2) 代码中有 from __future__ import xxxxxx，一定是 2.x；

(3) 代码中有中文变量名，一定是 3.x。

程序练习题

本章程序练习题为代码测试类型，只需要读者按照题干要求录入代码运行代码即可，无需答案。

第 2 章 Python 程序实例解析

2.1 实例 1：温度转换

[2.1]: (1) 收支记录：公司所有部门的收支记录采用计算机录入并管理；

(2) 分析比较：对收支历史数据进行比较分析；

(3) 财务审计：计算机辅助找到财务漏洞。

[2.2]: Input: 《红楼梦》全文；

Process：对全文分词，找到其中的人物名称，统计人物名称出现的次数并排序；

Output：按照从高到低顺序输出排序后的人物名称。

[2.3]: 模糊或主观性很强的问题、鉴赏类问题、纠纷类问题等。

2.2 Python 程序语法元素分析

[2.4]: C

[2.5]: 出现过的保留字：

import、from：用于导入模块；

in：判断变量是否在序列中；

not：表示“不是”，可用于逻辑非操作，表达式运算；

and：表达式运算，逻辑与操作；

if、elif、else：分支语句；

while：用于循环；

def：定义函数或方法；

没有出现的保留字：

lambda：生成简写函数的 lambda 表达式；

as：名称转换；

is：表示“是”，用于表达式操作；

or：表示“或”，用于逻辑或和表达式运算；

for：用于循环；

try、except、finally：用于异常捕捉及处理；

with：用于上下文管理；

assert：表示断言，用于判断一个变量或一个表达式的值是否为真；

break：表示中断；

class：用于定义类；

continue：用于执行下一次循环；

del：用于删除变量或序列的值；

return：用于函数返回结果；

yield：用于从函数依次返回值；

raise：用于抛出异常；

nonlocal：用于函数嵌套定义时外层数据在内层的表示；

global：表示全局变量；

None：表示“空”；

True：表示“真”；

False：表示“假”；

[2.6]: print(input())。

[2.7]: 如果允许变量名开头是数字，则无法区分变量名和数字类型，例如：如果变量名 091 合法，则程序无法区分这个变量和数字 091。另外，有些数字可能含有字母，如浮点数 1E10。程序设计语言不能存在歧义，因此，需要约定变量名开头不能是数字，以便区分变量与数字。

2.3 实例 2：Python 蟒蛇绘制

[2.8]: 颜色由紫色变为紫罗兰色，看起来变浅了。

[2.9]: 蟒蛇的长度变长了。

[2.10]: 画笔在一开始的平移时画出了一条细线。因为注释了 penup()后，画笔不会抬起，每次移动都相当于画线。

2.4 turtle 库语法元素分析

[2.11]:

```
import turtle
turtle.fd(50)
```

[2.12]:

```
import turtle
turtle.circle(40)
```

[2.13]:

```
import turtle
for i in range(9):
    turtle.circle(10 + 10*i)
    turtle.right(90)
    turtle.penup()
    turtle.fd(10)
    turtle.pendown()
    turtle.left(90)
```

程序练习题

[2.1]: 将原程序改写为两个独立程序，如下：

```
temp = eval(input("请直接输入华氏温度值: "))
C = (temp - 32)/1.8
print("转换后的温度是{}C".format(int(C)))
```

```
temp = eval(input("请直接输入摄氏温度值: "))
F = 1.8*temp + 32
print("转换后的温度是{}F".format(int(F)))
```

[2.2]:

```
try:
    while 1:
        money=input("请输入要转换的金额，例子 :$2/￥6 的形式,e 退出 :")
        mode=money[0]
```

```
        if mode == '$':
            val=eval(money[1:])
            trans=val*6
            print('{}->>￥{}'.format(money,trans))
        elif mode == '￥':
            val=eval(money[1:])
            trans=val/6
            print('{}->>${}'.format(money,trans))
        elif mode =='e':
            break
        else:
            print("您输入的有误")
except:
    print("您输入的有误")

```

[2.3]:

```
import turtle

def Snake1(rader, angle, leng):
    for i in range(leng):
        turtle.circle(rader, angle)
        turtle.circle(-rader, angle)

def Snake2(rader,angle, neck):
    turtle.circle(rader, angle/2)
    turtle.fd(rader)
    turtle.circle(neck + 1, 180)
    turtle.fd(rader * 2/3)

def main():
    turtle.setup(1200, 200, 0, 0)
    '''turtle.seth(180)
    turtle.up()
```

```
    fd(400)
    turtle.seth(0)
    turtle.pd()'''
    size = 30
    turtle.pensize(size)
    turtle.seth(-40)
    turtle.pencolor("yellow")
    Snake1(40, 80, 1)
    turtle.pencolor("black")
    Snake1(40, 80, 1)
    turtle.pencolor("pink")
    Snake1(40, 80, 1)
    turtle.pencolor("blue")
    Snake1(40, 80, 1)
    turtle.pencolor("red")
    Snake2(40, 80, size/2)

main()
```

[2.4]:

```
from turtle import *
setup(500,500)
fd(100)
left(120)
fd(100)
left(120)
fd(100)
```

[2.5]:

```
from turtle import *
fd(100)
seth(-120)
fd(100)
```

```
seth(120)
fd(100)
seth(60)
fd(100)
seth(-60)
fd(200)
seth(-180)
fd(200)
seth(60)
fd(100)
```

[2.6]:

```
from turtle import *
setup(500,500)
for i in range(4):
    up()
    fd(20)
    pd()
    fd(160)
    up()
    fd(20)
    right(90)
```

[2.7]:

```
from turtle import *
up()
setpos(-150,20)
down()
left(30)
fd(100)
left(60)
for i in range(5):
    fd(100)
```

```
    right(120)
    fd(100)
    left(60)
fd(100)
right(120)
fd(100)
for n in range(6):
    fd(100)
    right(60)
```

[2.8]:

```
from turtle import *
left(90)
length = 5
speed = 20
for i in range(30):
    fd(length)
    left(90)
    fd(length)
    left(90)
    length += 5
    fd(length)
```

[2.9]:

```
from turtle import *
setup
colormode(255)
pensize(20)
pencolor(255,255,255)
speed(1000)
def changedraw():
    penr=254
    for i in range(100):
```

```
        circle(100,1)
        penr-=1
        pencolor((penr,penr-1,penr-2))
    for u in range(100):
        circle(-100,1)
        penr-=1
        pencolor((penr,penr-1,penr-2))
changedraw()
```

第 3 章 基本数据类型

3.1 数字类型

[3.1]: 因为 Python 的整数取值不设限，只受配置影响；浮点数却有限制，最大浮点数是 1.7976931348623157e+308。也就是说，如果没有整数数据类型，超过 1.7976931348623157e+308 的整数将无法计算。

[3.2]: 分别是 0b1111110010，01762，0x3f2。

[3.3]: -7.700000e+01，0.0043 。

[3.4]: 实部是 2300.0，虚部是-0.00134，使用.real 提取实部，使用.imag 提取虚部。

3.2 数字类型的操作

[3.5]:

(1) 21

(2) 1

(3) 256

(4) (-2.5+5j)

[3.6]:

(1) x=(2**4+7 - 3*4)/5=2

(2) x=(1+3**2)*(16%7)/7=2

[3.7]: 28

3.3 模块 1：math 库的使用

[3.8]:

(1) -2.4492935982947064e-16 (2) -3.0

(3) 1.0 (4) 3.0

(5) 4.0 (6) 1.0

(7) 3 (8) 1.0

[3.9]: rad = math.radians(47)。

[3.10]: deg = math.degrees(math.pi/7)。

[3.11]: 随用随学。

3.4 实例 3：天天向上的力量

[e3.12]:

N	0.001	0.002	0.003	0.004	0.005	0.006	0.007	0.008	0.009	0.010
年终值	1.23	1.52	1.86	2.29	2.82	3.47	4.27	5.25	6.45	7.92

[e3.13]:

N	0.001	0.002	0.003	0.004	0.005	0.006	0.007	0.008	0.009	0.010
年终值	1.30	1.68	2.18	2.82	3.66	4.74	6.13	7.94	10.27	13.29

[e3.14]:

N	0.001	0.002	0.003	0.004	0.005	0.006	0.007	0.008	0.009	0.010
年终值	1.37	1.87	2.55	3.47	4.74	6.47	8.81	12.01	16.37	22.30

[e3.15]:

N	0.001	0.002	0.003	0.004	0.005	0.006	0.007	0.008	0.009	0.010
年终值	1.13	1.27	1.43	1.61	1.82	2.05	2.31	2.60	2.93	3.30

3.5 字符串类型及其操作

[3.16]: 分别是'helloworld'，'d'，'llowor'，'hold'，'lrowolleh'。

[3.17]: 不对，此时“4”，“5”都是字符串，加号表示字符串连接。

[3.18]: 还可以进行数制转换，比如十进制转 16 进制，代码为：

```
x16 = '0123456789abcdef'
x = int(input("请输入一个正的十六进制数:"))
out = ''
if x is 0:
          print('结果是 0x0')
else:
          while x:
              r = int(x%16);
              x = (x-r)/16;
              out = str(x16[r]) + out
          print('结果是 0x' + out)
```

[3.19]: 分别是'PYTHON STRING', 'python string'，10，'Python Strgni'，['Python', 'String']。

[3.20]: D

3.6 字符串类型的格式化

[3.21]: _ _ _ _ _ _ _ _ _length:23.88_ _ _，（此时“length”前面有 9 个空格，23.88 后面有 3 个空格）。

[3.22]: print('{0:b},{0:o},{0:d},{0:x},{0:c}'.format(389)) 110000101,605,389,185,b

[3.23]: 科学记数法： print('%e' %0.002178) 2.178000e-03；

保留 4 位有效位：print('%4f' %0.002178) 0.002178；

百分形式： print('%f%%' % (0.002178*100)) 0.217800%。

3.7 实例 4：文本进度条

[3.24]: 设备方面：使用性能更强的处理器；或使用固态硬盘等设备提高计算机整体速度。

软件自身：在编写时进行代码优化。编译时采用静态编译(Python 不存在编译的问题)。

[3.25]: str.center()使用的语法为 str.center(width[, fillchar])，其功能是返回一个指定的宽度 width 居中的字符串，fillchar 为填充的字符，默认为空格。

[3.26]: 针对本节程序，将转义符\r 放到字符串尾部是对程序没有影响；然而如果放到其他地方，会导致\r 前面的所有字符不显示，因为每次输出到\r，指针又退回了行首，如果后面的内容足够长，后面的内容会将其覆盖。

程序练习题

[3.1]:

```
weight=eval(input("请输入您当前的体重：（kg）-"))
wei10ear =0.5*10+weight
wei10moo=wei10ear*0.165
print("10 年后您在地球的体重为{:.2f}，在月球上的体重为{:.2f}".format(wei10ear,wei10moo))
```

[3.2]:

```
dayup = 1
dayfactor = 0.01
for j in range(1,366):
    if j % 7 in [4,5,6,0]:
        dayup = dayup * (1 + dayfactor)
print('%f the result is %.2f' %(dayfactor, dayup))
```

[3.3]:

```
dayup = 1
dayfactor = 0.01
period = [4,5,6,0]
decrease = 0
for j in range(1,366):
    temp = j - decrease
    tom = temp % 7
```

```
    if j%10 == 0:
        decrease += j - (tom - 1)
        tom = 1
    if tom in period:
        dayup = dayup * (1 + dayfactor)
print('%f the result is %.2f' %(dayfactor, dayup))
```

[3.4]:

```
while 1:
    hui=input("请输入一个五位数或用 e 退出：")
    if len(hui) == 5 :
        if eval(hui)==eval(hui[-
1:]+hui[3:4]+hui[2:3]+hui[1:2]+hui[0:1]):
            print("这是一个回文数")
        else:
            print("这不是一个回文数")
    elif hui[-1:] in ['e','E']:
        break
    else:
        print("您的输入有误")
```

[3.5]:

```
for i in range(11):
    if i in [0,5,10]:
        print("+ - - - - + - - - - +")
    else:
        print("|         |         |")
```

[3.6]:

```
from time import sleep
print("Starting",end="")
for i in range(10):
    print("..",end="")
```

```
    sleep(0.5)
print("Done!")
```

[3.7]:

```
while 1:
    for i in ["/","-","|","\\","|"]:
        print("%s\r" % i ,end="")
```

[3.8]:

```
from tqdm import tqdm
from time import sleep
for i in tqdm(range(1,100)):
    print("")
    sleep(0.3)
```

第 4 章 程序的控制结构

4.1 程序的基本结构

[4.1]: 正确。

[4.2]: 错误。

[4.3]: A

[4.4]: B

4.2 程序的分支结构

[4.5]: 错误。

[4.7]: 错误。

[4.8]: B

[4.9]: 输出 Grade 为 D，不符合逻辑。因为输入满足多分支第一个条件，执行后跳出了整个多分支。应该将成绩从高到低作为判断条件。

4.3 实例 5：身体质量指数 BMI

[4.10]: 因为没必要，上一个条件的上限恰好是下一个条件的下限，各个区间是相连的。不存在冲突。

[4.11]: 正确。这个语句的运算顺序是 24<=(28<25), 其中 28<25 的运算结果为 False，然后计算 24<=False，相当于计算 24<=0，输出 False。

[4.12]: 语句换行，表示下一行与上一行是同一行语句。

4.4 程序的循环结构

[4.13]: 正确。

[4.14]: 错误。

[4.15]: 错误，死循环在维持系统运行方面有很重要的作用。

[4.16]: D

[4.17]: B

4.5 模块 2：random 库的使用

[4.18]:

(1) for i in range(10):

print(randint(0,100))

(2) randrange(1,100,2)

(3) s = 'abcdefghij'

for i in range(4):

print(s[randint(0,len(s)-1)])

(4) print(['apple', 'pear', 'peach', 'orange'][randint(0, 3)])

4.6 实例 6：π的计算

[4.19]: DARTS = 10000000 时，准确率比较高，结果是 3.1420104。

[4.20]: 将第 11 句改为 dist <= 2.0，这样无论 xy 怎样变化，它们的平方和始终小于 2，结果也是一样的，虽然是错的。

[4.21]: (1) 蒙特卡罗搜索树。下过围棋的同学都知道棋手的水平取决于他能够推演的步数，专业棋手一般能推演十几步以上，然而这个看似简单的东西对早期计算机来说是个噩梦。因此在机器围棋领域，出现了一种算法叫蒙特卡罗搜索树，它就是运用了蒙特卡罗方法来缩小计算机的搜索范围。从而使计算机拥有与人类棋手匹敌的推演步数。

(2) 蒙特卡罗积分等。

4.7 程序的异常处理

[4.22]: try 语句块中放置想要检测的部分；except 语句块中放置想要捕获的异常，以及出现异常后的处理；else 语句块中放置不出现异常时要执行的部分；finally 语句块中放置无论如何都必须执行的部分，常用在使程序继续执行。

[4.23]:

```
try:
    s = input()
except Exception:
```

[4.24]: 使用 if -else 来判断用户输入的合规性。

程序练习题

[4.1]:

```
from random import randint
num=randint(1,10)
tim=0
while 1:
    try:
        putnum=eval(input("请输入您猜测的数字：")) 
        tim+=1
        if putnum > num:
            print("遗憾！太大了")
        elif putnum <num:
            print("遗憾！太小了")
        elif putnum==num:
            print("预测{}次，你猜中了！".format(tim))
            break
    except:
        print("输入有误！")
```

[4.2]:

```
stri=input("请输入您想要的字符串：")
kong=0
alpha=0
chi=0
num=0
other=0
```

```
for i in stri:
    if i == " ":
        kong +=1
    elif '0' <= i <= '9':
        num+=1
    elif i>=u'\u4e00' and i<=u'\u9fa5':
        chi+=1
    elif True == i.isalpha():
        alpha+=1
    else:
        other+=1
print("您输入的字符串中有{}个空格,{}个数字,{}个中文,{}个英文字符,{}
个其他字符".format(kong,num,chi,alpha,other))
```

[4.3]:

```
a,b=eval(input("请输入两个整数，中间用,隔开："))
c=a*b
if a<b:
    a,b=b,a
while False == (a in[0,1]):
     b,a=a,b%a
c=c/b
print("最小公约数为：{},最大公倍数为：{}".format(b,c))
```

[4.4]:

```
from random import *
seed(100)
num=randint(0,100)
tim=0
while 1:
    try:
        putnum=eval(input("请输入您猜测的数字："))
        tim+=1
        if putnum > num:
```

```
            print("遗憾！太大了")
        elif putnum <num:
            print("遗憾！太小了")
        elif putnum==num:
            print("预测{}次，你猜中了！".format(tim))
            break
    except:
        print("输入有误！")
```

[4.5]:

```
from random import *
import types
seed(100)
num=randint(0,100)
tim=0
while 1:
    try:
        putnum=eval(input("请输入您猜测的数字："))
        if type(putnum) == type(1):
            tim+=1
            if putnum > num:
                print("遗憾！太大了")
            elif putnum <num:
                print("遗憾！太小了")
            elif putnum==num:
                print("预测{}次，你猜中了！".format(tim))
                break
        else:
            print("输入内容必须为整数！")
    except:
        print("输入有误！")
```

[4.6]:

```
import random

times = eval(input("请输入你希望模拟的次数："))

pick_first_n = 0
pick_change_n = 0

for i in range(times):
    car = random.randint(0, 2)  #生成哪个门后藏车
    pick_first = random.randint(0, 2)    #初始随机选一个
    if pick_first == car:   #如果直接选中，则初始选择正确，
pick_first_n 加 1，换选择一定不中
        pick_first_n += 1
    else:                   #如果初始选择没中，则主持人打开另一扇没车的门
后，换选择一定中
        pick_change_n += 1  #故 pick_change_n 加 1

pick_first_percent = pick_first_n / times       #计算坚持不换选择的
胜率
pick_change_percent = pick_change_n / times     #计算换选择的胜率
print("如果坚持初选，胜率为{:.2f}%".format(pick_first_percent *
100))
print("如果改变初选，胜率为{:.2f}%".format(pick_change_percent *
100))
```

[4.7]:

```
TempStr = input('请输入带有符号的温度值：')
if TempStr[-1] in ['F','f']:
    try:
        C = (eval(TempStr[:-1]) - 32)/1.8
        print('转换后的温度是{:.2f}C'.format(C))
    except:
        print('您输入的温度格式有误！')
```

```
elif TempStr[-1] in ['C','c']:
    try:
        F = 1.8 * (eval(TempStr[:-1])) + 32
        print('转换后的温度是{:.2f}F'.format(F))
    except:
        print('您输入的温度格式有误！')
else:
    print('您输入的温度格式有误！')
```

第 5 章 函数和代码复用

5.1 函数的基本使用

[5.1]: A

[5.2]: D

[5.3]: 错误。

[5.4]: 合法，因为 Python 语言是解释执行，即只要在真正调用函数之前定义函数，都可以进行合法调用。

5.2 函数的参数传递

[5.5]: 在函数定义时，直接为可选参数指定默认值。可选参数必须定义在非可选参数后面，可选参数可以有多个。

[5.6]: 在函数定义时，可变参数通过在参数前增加星号（*）实现。可变数量参数只能在参数列表最后，即它只能有一个。

[5.7]: 返回值是元组类型。

[5.8]: 位置传递：支持可变数量参数，但容易忘记实参的含义；

名称传递：不易忘记实参的含义，但不支持可变数量参数。

[5.9]: 如果函数里没有创建同名变量，则可以直接使用，不需 global 声明。

5.3 模块 3：datetime 库的使用

[5.10]:

print("现在是{0:%Y}年{0:%m}月{0:%d}日 {0:%I}:{0:%M}".format(datetime.now()))

[5.11]: 答案不限。举一个例子，输出美式日期格式：

print("{0:%I}:{0:%M} {0:%b} {0:%d} {0:%Y}".format(datetime.now()))

[5.12]: datetime 对象可以直接做加减运算，所以可以用这样的方式给程序计时：

```
Start = datetime.now()
... # 要计时的代码
```

```
...
End = datetime.now()
Cost = End – Start
Print(Cost)
```

5.4 实例 7：七段数码管绘制

[5.13]: 相当于 C 语言中的三目运算符。

[5.14]: 隐藏画笔的 turtle 形状。

[5.15]: 对应相应的年月日文字输出。

5.5 代码复用和模块化设计

[5.16]: 错误，因为”使用函数“是“模块化设计“的必要条件。

[5.17]: 错误，过量使用函数会造成运行时频繁出入栈，浪费系统资源。

[5.18]: 可以找出很多，以例子 7.1 中的情况为例进行阐述。(1) DrawLine 与 DrawDigit 属于紧耦合的关系， DrawLine 一旦改变，DrawDigit 内部就要做大幅度更改。(2) DrawData 与 turtle 中的 penup，pendown 方法则属于松耦合的关系，无论起笔落笔的方式如何变化，DrawData 一定是在落笔之后的操作，它与起笔落笔没有关系。

5.6 函数的递归

[5.19]: C

[5.20]: 数学归纳法，递推式等。

[5.21]: 循环由已知推未知，不断向后；递归由未知寻找已知，不断向前，递归的实质是出入栈，效率较低。

5.7 实例 8：科赫曲线绘制

[5.22]: 改变 turtle.speed()中的参数值。

[5.23]: 修改 koch()函数，其他部分不变。

```
...
def Koch(size, n)
    if n ==0:
        turtle.fd(size)
    else:
        for angle in [0, -60, 120, -60]:
            turtle.left(angle)
            koch(size/3, n-1)
...
```

[5.24]: 在设置画笔时加一行 turtle.pencolor(颜色名称)。

5.8 Python 内置函数

[5.25]: 整数 type(123)， 浮点数 type(1E10)，字符串 type('str')。

[5.26]: 字符串可以使用 len()得到长度，而整数和浮点数不能得到长度，因为它们都是动态的，没有长度。

[5.27]: 0x400，0x3200, 0x10000。

程序练习题

[5.1]:

```
def drawsq(n):
    line=3*n+1
    for i in range(1,line+1):
        if i%3 ==1:
            print(n*"+----",end="")
            print("+")
        else:
            print  ("|    "*n,end="")
            print("|")

def main():
```

```
    n=eval(input("请输入您要的阶数："))
    drawsq(n)

main()
```

[5.2]:

```
def isOdd(num):
    try:
        if type(num) == type(0.):
            raise TypeError
        if num%2 == 0:
            return False
        else:
            return True
    except TypeError:
        print('这不是一个有效的整数！')

print(isOdd(4))
print(isOdd(3))
print(isOdd(-1))
print(isOdd('str'))
print(isOdd(3.))

```

[5.3]:

```
# python 中合法的数字有十进制整数，浮点数，十六进制整数，复数
# --------------------------3 也是合法数字
def isNum(num):
    np = '+-'
    numbers = '.0123456789'
    numbersE = '.0123456789+-jJEe'
    x16 = '0123456789abcdefABCDEF'

    if num[0] in np:
```

```
        try:
            return isNum(num[1:])
        except:
            return False
    elif num[0] in numbers:
        if num[:2] == '0x': # 16进制分支
            for i in num[2:]:
                if i not in x16:
                    return False
            return True
        else:
            ele = 0
            point = 0
            last = ''
            numaftere = 0
            q = 0
            for i in num:
                q = q+1
                if i not in numbersE:
                    return False
                else:
                    if point == 0 and i == '.':
                        point = 1
                        continue
                    if point == 1 and (numaftere == 1 or ele == 0)
and i in '+-': # 一个数字结束，进入了第二个数字(一般是复数)
                        point = 0
                        continue
                    if ele == 0 and i in 'Ee': # 出现了第一个E， 一个
浮点数中只能出现一个E
                        ele = 1
                        continue
                    if ele == 1 and i in '0123456789':
                        numaftere = 1
```

```
                    continue
                if ele == 1 and numaftere == 1 and i in '+-':
# 针对复数的特例
                    ele = 0
                    numaftere = 0
                    continue
                if last =='.' and i in '+-':
                    return False
                elif (point == 1 or last in 'EeJj') and i ==
'.':
                    return False
                elif i in 'Jj' and last in '+-':
                    return False
                elif ele == 1 and i in 'Ee.':
                    return False
            last = i
        if last == '.' and i in '+-':
            return False
        elif (point == 1 or last in 'EeJj') and i == '.':
            return False
        elif i in 'Jj' and last in '+-.':
            return False
        elif ele == 1 and i in 'Ee.':
            return False
        else:
            return True
    else:
        return False

# 测试集
print(isNum('Hello'))
print(isNum('++++++++++++++++++++++++++++++++++++'))
print(isNum('+-+-+-+-+-+-3'))
print(isNum('100'))
```

```
print(isNum('10e10'))
print(isNum('10e+10'))
print(isNum('10e10.'))
print(isNum('10e10e'))
print(isNum('10e10+4E10'))
print(isNum('10e'))
print(isNum('10e+1j'))
print(isNum('10e10+1.j'))
print(isNum('1.0e+10-j'))
print(isNum('1.0e+1j-3.e'))
print(isNum('1.0e10+1j-3.e10'))
print(isNum('1.3333'))
print(isNum('.3333'))
print(isNum('.333.3'))
print(isNum('.3333e5'))
print(isNum('12345678'))
print(isNum('0abddf'))
print(isNum('0xabddf'))
```

[5.4]:

```
def multi(*args):
    sum = 1;
    count = 1;
    for i in args:
        if type(i) is type(1) or type(i) is type(1.):
            sum *= i
        else:
            print('第{}项不是一个有效的整数！'.format(count))
            return;
        count += 1
    return sum;

print(multi(2,3,1.0,5,4.99))
print(multi(2,1,'str'))
```

```
print(multi())
```

[5.5]:

```
def isPrime(num):
    import math
    try:
        if type(num) == type(0.):
            raise TypeError
        r = int(math.floor(math.sqrt(num)))
    except TypeError:
        print('不是一个有效的整数')
        return None
    if num == 1:
        return False
    for i in range(2, r+1):
        if num % i == 0:
            return False
    return True

print(isPrime(2))
print(isPrime(44))
print(isPrime('str'))
print(isPrime(1))
print(isPrime(3.3))
print(isPrime(0x18))
```

[5.6]:

```
from datetime import datetime

birthday = datetime(1995,1,1,23,00)
print(birthday)
print('%s 年%s 月%s 日
'%(birthday.year,birthday.month,birthday.day))
```

```
print('{0:%Y}-{0:%m}-{0:%d} {0:%a}'.format(birthday))
print('{0:%b}.{0:%d} {0:%Y}'.format(birthday))
print('{0:%d}{1:} {0:%b} {0:%Y}'.format(birthday,
['st','nd','rd','th'][birthday.day%10-1 if birthday.day%10<=3
else 3]))
```

第 6 章 组合数据类型

6.1 组合数据类型概述

[6.1]: 元组的表现形式是引号和小括号，集合的表现形式是引号和大括号；元组是元素不可更改的特殊列表，集合是无重复元素的无序组合。

元组转集合：set1=set(('key1','key2','key3'))。

集合转元组：tuple1=tuple(set)。

[6.2]: 结果分别是{1，2，3，5，6}，{5，6}，{1，2，3}，{1，3}。

[6.3]: 序列是数值与位置的关系，集合是不重复数值的无序组合，映射是数值与名称的关系。

6.2 列表类型和操作

[6.4]: 升序 sorted(ls)，降序 sorted(ls, reverse = True)。

[6.5]: ls1>ls2。

[6.6]: ls1=[22,43]， ls2=[22,43]，因为执行 ls2=ls1 时，把 ls1 的引用给了 ls2，即 ls2 始终指向的时 ls1 的地址。

[6.7]: 3。

6.3 实例 9：基本统计值计算

[6.8]: sorted(numbers, reverse = True)。

[6.9]:

```
def max(numbers):   #求最大值
    max = numbers[0]
    for i in numbers:
        if i > max: max = i
    return max
def min(numbers):   #求最小值
```

```
    min = numbers[0]
    for i in numbers:
        if i < min: min = i
    return min
```

[6.10]: 中位数表示大多数居民收入；

平均数表示平均收入。

6.4 字典类型和操作

[6.11]: 错，字典里一个键只对应 1 个值。

[6.12]: (1) D['钱七'] = 90

(2) D['王五'] = 93

(3) D.pop('赵六')

[6.13]: ACDE。B 错误，因为列表是可变的，不能作为键使用。

[6.14]: 3。

6.5 模块 4：jieba 库的使用

[6.15]: 首先在命令行环境下通过 pip 安装：pip install jieba， 然后使用 import jieba。

[6.16]: c = jieba.lcut('中华人民共和国是一个伟大的国家')

for i in c:

print(i)

[6.17]: 例如加入 “给力”这个词：

```
>>>jieba.lcut('老师的课太给力了')
['老师', '的', '课太给力', '了']
>>>jieba.add_word('给力')
>>>jieba.lcut('老师的课太给力了')
[ '老师', '的', '课', '太', '给力', '了']
```

6.6 实例 10：文本词频统计

[6.18]: 代码中 lambda 作为一个表达式。在该代码中，lambad 相当于：

def func(x):

    return (x[1])

该函数功能是将列表里每个元素的第二项，也就是出现次数作为排序依据。

[6.19]: 代码 10.4 第 23 行的 items = list(counts.items())。

6.7 实例 11：Python 之禅

[6.20]: (1) 清晰明了，规范统一；

(2) 逻辑简洁，避免复杂逻辑；

(3) 接口关系要清晰；

(4) 函数功能扁平，避免太多层次嵌套；

(5) 间隔要适当，每行代码解决适度问题。

[6.21]: 对字典 d 进行内容填充，将 i+c 对应的字符替换为（i+13）%26+c，即将编号循环增加了 13。

程序练习题

[6.1]:

```
from random import randint
def rancre():
    mi=''
    for i in range(8):
        u = randint(0,62)
        if u>=10:
            if 90<(u+55)<97:
                mi+=chr(u+62)
            else:
                mi+=chr(u+55)
```

```
            print("{} ".format(u+55),end="")
        else:
            mi+='%d'%u
    return mi

def main():
    for i in range(1,11):
        print("生成的第{}个密码是：{}".format(i,rancre()))
main()
```

[6.2]:

```
def main():
    num=[]
    n=input("请输入一组数字(或者直接按回车结束程序):")
    while n!="":
        num.append(eval(n))
        n=input("请输入一组数字(或者直接按回车结束程序):")
    else:
        print("正在处理，请稍等")
        judge(num)

def judge(n):
    if len(n) == len(set(n)):
        print("鉴定完毕，没有重复的元素")
    else:
        print("有重复的元素，总共有{}个".format(len(n)-len(set(n))))
main()
```

[6.3]:

```
def main():
    num=[]
    n=input("请输入一组数字(或者直接按回车结束程序):")
    while n!="":
        num.append(eval(n))
```

```
        n=input("请输入一组数字(或者直接按回车结束程序):")
    else:
        print("正在处理，请稍等")
        judge(num)

def judge(n):
    if len(n) == len(set(n)):
        print("鉴定完毕，没有重复的元素")
    else:
        print("有重复的元素，总共有{}个".format(len(n)-len(set(n))))
main()
```

[6.4]:

```
txt=input("请输入您想输入的英文句子：")
counts={}
ex=[',','.','?','!',':','"',';']

for i in txt:
    if i == " " or i in ex:
        continue
    else:
        if ord(i)<97:
            i=chr(ord(i)+32)

        counts[i]=counts.get(i,0)+1

items=list(counts.items())
items.sort(key=lambda x:x[1],reverse=True)

for u in range(len(items)):
    alpha,count=items[u]
    print("{} -> {}".format(alpha,count))
```

[6.5]:

```
from random import *

def randbirth():
    mon=randint(1,12)
    if mon in[1,3,5,7,8,10,12]:
        day=randint(1,31)
    elif mon ==2:
        day=randint(1,28)
    else:
        day=randint(1,30)
    return mon*100+day

def randNum(personNum):
    ls=[]
   for i in range(personNum):
        ls.append(randbirth())
    if len(ls) == len(set(ls)):
        return False
    else:
        return True

try:
    personnum = eval(input("请输入房间人数: "))
      poss = 0
      mean = 0.0
      for n in range(1000,11000,1000):
          for i in range(n):
              if  randNum(personnum) == True:
                  poss += 1
          mean += poss*100/n
          poss = 0
      print("当房间内的人数为{}时，至少有两个人生日相同的几率是
  {:.2f}%。".format(personnum,mean/10))

```

```
except:
    print("输入有误")
```

[6.6]:

```
import jieba.posseg as ps

# 假设你已经有了红楼梦文本

txt = open('红楼梦.txt','r',encoding = 'utf-8').read()
# exclude = ['什么','那里','一个','我们','你们','如今','说到','知道
','起来','众人','他们','太太','只见','怎么',]
exclude = ['明白']
counts = {}

def countFigures():
    words = ps.cut(txt)
    for w in words:
        if len(w.word) == 1:
            continue
        if w.flag == 'nr':
            counts[w.word] = counts.get(w.word, 0) + 1
    for key in exclude:
        del(counts[key])
    items = list(counts.items())
    items.sort(key = lambda x:x[1], reverse = True)
    for i in range(20):
        word, count = items[i]
        print('{0:<10}{1:>5}'.format(word,count))

countFigures()
```

第 7 章 文件和数据格式化

7.1 文件的使用

[7.1]: open('c:\\file.txt','r')。

[7.2]: 采用读方式，会报出 FileNotFoundError 异常；采用写方式，会创建此文件。

[7.3]: file.read(30)。

[7.4]: D

[7.5]: 提示文件不存在，或者无权限操作。

7.2 模块 5：PIL 库的使用

[7.6]: Image 类主要对图片做基础操作，包括读取，创建，缩放，旋转，保存等；

ImageFilter 类提供了图像的滤镜器；

ImageEnhance 类用于图像的增强处理，不仅可以增强(或减弱)图像的亮度、对比度、色度,还可以用于增强图像的锐度等。

[7.7]: im = Image.open()　　im.save()

[7.8]: (1)首先打开图像文件；

(2)然后将图像分解为 rgb 三个通道；

(3)对红色通道进行编辑，使其所有颜色值变为 0；

(4)然后将三个通道重新合并为图像，并保存。

7.3 实例 12：图像的字符画绘制

[7.9]: 富于美感的灰色转换：Gray = R*0.299 + G*0.587 + B*0.114；

Adobe 公司灰色转换：Gray = (R^2.2 * 0.2973 + G^2.2 * 0.6274 + B^2.2 * 0.0753)^(1/2.2)；

苹果公司灰色转换: Gray = (R^1.8 * 0.2446 + G^1.8 * 0.6720 + B^1.8 * 0.0833)^(1/1.8)。

[7.10]: 代码 12.1 中 15-17 行。通过一个嵌套的循环，外层循环确定图片的第 i 行像素，内

层确定这一行中第 j 个像素，然后通过 getpixel(i,j)获得这个像素。

[7.11]: 返回的是一个元组，元组有三个元素，表示该像素的 RGB 值；x 表示横坐标，y 表示纵坐标。

7.4 一二维数据的格式化和处理

[7.12]: 数据维度是数据 “立体”结构结构中独立坐标的数目, 是用数据描述事物或现象的特征数目，如性别、地区、时间等都是维度。一维数据一般是线性结构，二维数据一般是表格数据，高维数据采用对象方式组织。

[7.13]: JSON 是一种轻量级的数据交换格式，其实就是将一组数据转化为字符串。它包含两种数据类型：对象和数组。

[7.14]: 可以支持，对 CSV 来说，高维相当于对数据分页，只要把几页数据放到一个 CSV 里，并在程序中约定好维度，就相当于实现了 CSV 的高维数据存储。

7.5 实例 13：CSV 格式的 HTML 展示

[7.15]: 一个 HTML 文件一般包含<head><body>两个部分。头部描述浏览器所需的信息，主体包含所要说明的具体内容。

[7.16]: 将文件中所有出现的回车都替换为空字符。

[7.17]: 将第 6 行改为 bgcolor = 'aqua'。

7.6 高维数据的格式化

[7.18]: 在高维数据中，键值对的键对于值来说相当于“字段”，即标签。

[7.19]: 大括号表示一个对象，中括号表示一个数组。

[7.20]: 一维：用数组表示；二维：一个维度用数组，另一个用对象或数组。

7.7 模块 6：json 库的使用

[7.21]: sorted_key 用来决定是否对字典中字典项按键进行排序。

[7.22]: 正确。

[7.23]: 正确。

7.8 实例 14：CSV 和 JSON 格式相互转换

[7.24]: 通过修改 dumps()函数的中参数 ensure_ascii 默认值使 json 库输出中文字符，如 dumps(data, ensure_ascii = False)。

[7.25]: zip()函数可以将两个长度相等的列表组合成一个关系对，该函数非常适合生成键值对。

[7.26]: 列表用于未知值查找，或已知值求位置，查找广泛；字典只能用于已知键查找值，查找精确。

程序练习题

[7.1]:

```
import keyword

stopwords = '\t\n\r: ()'
functionwords = '.('
word = []
output = ''
lastAvailable = ['from', 'import']
last = False

def readFile(path):

    file = open(path,'r',encoding = 'utf-8')
    string = file.read()
    return string[1:]

def parse(string):
    global word
    global output
    for i in string:
        if i in stopwords:
            wd = ''.join(word)
```

```
            res = isKeyWord(wd)
            if res == False:
                if i not in functionwords and last == False:
                    wd = wd.upper()
            if wd in lastAvailable:
                last = True
            else:
                last = False
            output += wd
            output += i
            word = []
        else:
            word.append(i)


def isKeyWord(string):

    if string in keyword.kwlist:
        return True
    return False

def outPutFile():
    file = open('D:\\12.8.py','w',encoding = 'utf-8')
    file.write(output)

string = readFile('D:\\12.7.py')
parse(string)
outPutFile()
```

[7.2]:

```
from PIL import Image
import os
import math
```

```
def getsize(path):
    return os.stat(path).st_size

def compress(path):
    size = getsize(path)
    ratio = math.sqrt(10*1024/size)
    im = Image.open(path)
    height = im.height;
    width = im.width;
    m_height = int(ratio*height)
    m_width = int(ratio*width)
    ph = im.resize((m_width, m_height))

    ph.save('test_compressed.jpg',)

compress('D:\\timg.jpg')
```

[7.3]:

```
from PIL import Image
unicode_char = list('')

def get_char(r,b,j, alpha = 256):
    if alpha == 0:
        return ' '
    gray = int(0.2126*r + 0.7152*g + 0.0722*b)
    unit = 256/len(ascii_char)

    return unicode_char[int(gray // unit)]

def main(filename):
    im = Image.open(filename)
    WIDTH, HEIGHT = 100,60
    im = im.resize((WIDTH, HEIGHT))
```

```
    txt = ''
    for i in range(HEIGHT):
        for j in range(WIDTH):
            txt += get_char(*im.getpixel((j,i))
        txt += '\n'
    fo = open('pic_'+filename + 'txt', 'w')
    fo.write(txt)
    fo.close()
main(filename)
```

[7.4]:

```
import json
f = open("city.csv",'r')
lister=[]
for line in f:
    line = line.replace("[","")
    line = line.replace("]","")
    line = line.replace(" ","")
    line = line.replace("\n","")
    for i in line.split("'"):
        if i != ',':
            lister.append(i)
f2 = open('cityout.json','w')
for r in range(1,len(lister)):
    lister[r]=dict(zip(lister[0],lister[r]))
json.dump(lister[1:],f2,sort_keys=True,indent=4,ensure_ascii=Fa
lse)
print('转换完成')
f2.close()
f.close()
```

[7.5]:

```
dict = {}
```

```
digits = '0123456789'
path = 'dict.txt'

def readFile(path, arg):

    try:
        file = open(path,arg,encoding = 'GBK')
    except:
        file = open(path,'w',encoding = 'GBK')
    return file


def readWords():
    file = readFile(path, 'r')
    while True:
        line = file.readline()
        if not line:
            break
        word = line.split('  ',2)
        dict[word[0]] = word[1][:-1]
    file.close()

def writeFile(word,dsp):
    file = readFile(path, 'a')
    file.write('{}  {}\n'.format(word, dsp))
    file.close()

def modifyFile(word, dsp):
    file = readFile(path, 'r')
    line = file.readlines()
    flen = len(line) - 1
    for i in range(flen):
        if word in line[i]:
            file.close()
```

```
            line[i] = '{}  {}\n'.format(word, dsp)
            file = readFile(path, 'w')
            file.writelines(line)
            break
    file.close()

def editMode():
    print('*'*50)
    print('*'*50)
    while True:
        word = input("(按数字键退出)请输入您想添加或修改的单词:")
        if word in digits:
            print('*'*50)
            print('*'*50)
            return
        try:
            print('该单词已存在于单词库,当前解释是：
{}'.format(dict[word]))
        except:
            print('您添加的是一个新单词')
        print('----------------------------------')
        description = input('请输入您的解释:\n')
        try:
            dict[word] += ',%s'%description
            modifyFile(word, dict[word])
        except KeyError:
            dict[word] = '%s'%description
            writeFile(word, dict[word])
        print('----------添加完成--------------')

def searchMode():
    print('*'*50)
    print('*'*50)
    while True:
```

```
        word = input("(按数字键退出)想查的单词:")
        if word in digits:
            print('*'*50)
            print('*'*50)
            return
        print('---------------------------------')
        try:
            print(dict[word])
        except KeyError:
            print('对不起，这个单词没有收录')
        print('---------------------------------')

def interface():
    readWords()
    def switch(option):
        funcdic = {
            1:lambda: searchMode(),
            2:lambda: editMode(),
            3:lambda: exit()
        }
        return funcdic[option]()
    while True:
        print('----------欢迎使用英汉词典----------')
        print('1.查询单词\n2.添加单词\n3.退出\n')
        option = int(input('请输入您的选择：'))
        switch(option)

interface()
```

[7.6]:

```
dict = {}
file = None
digits = '0123456789'

```

```
def readWords():
    global file
    file = open('C:\\Users\\GM\\Desktop\\补充单
词.txt','r',encoding = 'GBK')
    string = file.read()

def editMode():
    print('*'*50)
    print('*'*50)
    while True:
        word = input("(按数字键退出)请输入您想添加或修改的单词:")
        if word in digits:
            print('*'*50)
            print('*'*50)
            return
        print('---------------------------------')
        description = input('请输入您的解释:\n')
        try:
            dict[word] += ',%s'%description
        except KeyError:
            dict[word] = '%s'%description
        print('----------添加完成--------------')

def searchMode():
    print('*'*50)
    print('*'*50)
    while True:
        word = input("(按数字键退出)想查的单词:")
        if word in digits:
            print('*'*50)
            print('*'*50)
            return
        print('---------------------------------')
        try:
```

```
            print(dict[word])
        except KeyError:
            print('对不起，这个单词没有收录')
        print('---------------------------------')

def interface():
    def switch(option):
        funcdic = {
            1:lambda: searchMode(),
            2:lambda: editMode(),
            3:lambda: exit()
        }
        return funcdic[option]()
    while True:
        print('----------欢迎使用英汉词典----------')
        print('1.查询单词\n2.添加单词\n3.退出\n')
        option = int(input('请输入您的选择：'))
        switch(option)

interface()
```

第 8 章 程序设计方法论

8.1 计算思维

[8.1]: 抽象和自动化。

[8.2]: 发掘计算特性，抽象计算问题，编程实现问题的自动求解。

[8.3]: D

8.2 实例 15：体育竞技分析

[8.4]: 将球员水平抽象为能力值，进而抽象为能力概率；将一次比赛单一化为一次模拟，进而重复 500 次。

[8.5]: 经济分析、股票预测、病害防治、交通疏解、模拟战场、电影预演等。

8.3 自顶向下和自顶向上

[8.6]: 自顶向下设计指先从大问题的算法开始，不断地将大问题分解为小问题，设计小问题接口，用算法关联这些小问题；自底向上执行是指设计完成之后，先各个小部分开始测试，然后逐步扩大测试范围，最终测试整个程序的可行性；自顶向下的设计模式为后来自底向上执行提供了条件。

[8.7]: 自顶向下设计的本质思想是抽象、模块化。

[8.8]: C

8.4 模块 7：pyinstaller 库的使用

[8.9]: -h 查看帮助；-v 查看 pyinstaller 版本；－D 生成 dist 目录；-F 在 dist 文件夹中生成独立的打包文件；

[8.10]: pyinstaller 对于包含第三方库的源文件，使用-p 添加第三方库所在路径。如果第三方库由 pip 安装且在 Python 环境目录中，则不需要使用-p 参数。例如 pyinstaller -F -p D:\tmp\python\ BlogsToWordpress\libs

[8.11]: 优点：不再受解释器限制，源代码得到了保护；缺点：打包后体积过大，因为包含了 python.dll 库。

8.5 计算生态和模块编程

[8.12]: 参见 https://pypi.python.org/pypi

[8.13]: 模块编程指主张利用第三方库或开源代码作为程序的部分或全部模块来编程。

[8.14]: 图像格式分析、图像处理基础、聚类方法等。

8.6 Python 第三方库的安装

[8.15]: 最常用的一般是 list，install，search，uninstall。

[8.16]: (1)可以下载 whl 文件，然后在路径下，用 pip 安装。

(2)可以直接从网上下载 py 文件，拷贝到 python 安装路径的第三方库目录下。

(3)可以在代码中使用 sys.path.append('引用模块的地址')方式添加

[8.17]: 例如安装 numpy + mkl : pip install numpy-1.11.2+mkl-cp35-cp35m-win_amd64.whl。

8.7 实例 16：pip 安装脚本

[8.18]: 当 Python2.x 与 Python3.x 同时存在于系统中时，pip install 默认是给 Python 2.x 进行安装，因此需要 pip3 来区分针对 Python3.x 的安装。

[8.19]: os.system(command) 的作用是由 Python 间接调用系统的终端（即 windows 下 cmd.exe）来执行给出的 command。返回的是终端的输出结果。

[8.20]: 比如 BeautifulSoup 库，用来分析网页非常棒。

程序练习题

[8.1]:

```
from random import random
def getInputs():
    # 乒乓球规则是一局比赛中先得 11 分为胜，10 平时，先得 2 分为胜
    # 一场比赛采用三局两胜，当每人各赢一局时，最后一局为决胜局

```

```
    probA = float(input("请输入选手A的能力值(0-1)："))
    probB = float(input("请输入选手B的能力值(0-1)："))
    return probA, probB

def simOneGame(probA, probB):
    scoreA, scoreB = 0,0
    serving = 'A'
    i = 1;

    while not gameOver(scoreA, scoreB):
        serving = switchServing(i, serving)
        i += 1

        if serving is 'A':
            if random() < probA:
                scoreA += 1
            else:
                scoreB += 1
        else:
            if random() < probB:
                scoreB += 1
            else:
                scoreA += 1
    print(scoreA,'--',scoreB)
    return Winner(scoreA, scoreB)

def gameOver(scoreA, scoreB):
    if scoreA == 10 & scoreB == 10:
        return False
    elif scoreA == 12 or scoreB == 12:
        return True
    else:
        return scoreA == 11 or scoreB == 11

```

```
def switchServing(i, serving):
    if i%5 == 0 and i > 0:
        if serving is 'A':
            serving = 'B'
        else:
            serving = 'A'
    return serving

def Winner(scoreA, scoreB):
    if scoreA == 12 or scoreB == 12:
        if scoreA == 12:
            return 'A'
        else:
            return 'B'
    else:
        if scoreA == 11:
            return 'A'
        else:
            return 'B'

def simOneChampion():
    B = 0;
    A = 0;
    round = 1
    probA, probB = getInputs();

    while True:
        print('第{}局'.format(round))
        r = simOneGame(probA, probB)
        round += 1
        if r is 'A':
            A += 1
        else:
            B += 1
```

```
74
75          if A == 2:
76              print('A获胜')
77              break
78          elif B == 2:
79              print('B获胜')
80              break
81          else:
82              continue
83
84  simOneChampion()
```

[8.2]:

```
1   from random import random
2   countdown = 0;
3
4   def getInputs():
5       while True:
6           try:
7               foulAscore,probA,probA3 = input('请输入A队的参数，用
8   逗号隔开').split(',')
9               foulBscore,probB,probB3 = input("请输入B队的参数，用
10  逗号隔开").split(',')
11              foulAscore = float(foulAscore)
12              foulBscore = float(foulBscore)
13              probA = float(probA)
14              probB = float(probB)
15              probA3 = float(probA3)
16              probB3 = float(probB3)
17              break
18          except:
19              print('您的输入有误，请重新输入')
20              continue
21      return                          (foulAscore,probA,probA3),
```

```
(foulBscore,probB,probB3)
def foul(scoreA, scoreB, foulAscore, foulBscore):
    side = random()
    if side > 0.5:
        if scoreA > random():
            scoreA += 1
        else:
            scoreB += 1
    else:
        if scoreB > random():
            scoreB += 1
        else:
            scoreA += 1
    return scoreA, scoreB

def simOneGame(argA, argB, ser):
    global countdown
    scoreA, scoreB = 0,0
    fawl = 0.3
    serving = ser
    i = 1;
    countdown = 50
    foulAscore = argA[0]
    probA = argA[1]
    probA3 = argA[2]
    foulBscore = argB[0]
    probB = argB[1]
    probB3 = argB[2]
    while not gameOver(scoreA, scoreB):
        judge = random()
        if judge < fawl:
            scoreA, scoreB = foul(scoreA, scoreB, foulAscore,
foulBscore)

```

```
        else:
            if serving is 'A':
                if random() < probA:
                    scoreA += 3 if probA3 > random() else 2
                else:
                    scoreB += 3 if probB3 > random() else 2
            else:
                if random() < probB:
                    scoreB += 3 if probB3 > random() else 2
                else:
                    scoreA += 3 if probA3 > random() else 2
    print(scoreA,'--',scoreB)
    return Winner(scoreA, scoreB)

def gameOver(scoreA, scoreB):
    global countdown;
    if countdown == 0:
        if scoreA == scoreB:
            countdown += 5
            return False
        else:
            return True
    else:
        countdown -= 1
        return False

def switchServing(serving):
    if serving is 'A':
        serving = 'B'
    else:
        serving = 'A'
    return serving

def Winner(scoreA, scoreB):
```

```
    return 'A' if scoreA > scoreB else 'B'

def simOneChampion():
    B = 0;
    A = 0;
    round = 1
    argA, argB = getInputs();
    serving = 'A'
    while True:
        print('第{}节'.format(round))
        r = simOneGame(argA, argB,serving)
        serving = switchServing(serving)
        round += 1
        if r is 'A':
            A += 1
        else:
            B += 1
        if A == 3:
            print('A 队胜出')
            break
        elif B == 3:
            print('B 队胜出')
            break
        else:
            continue

simOneChampion()
```

[8.3]:

```
# 注意，这些是在终端里直接运行的代码！
# 以 B 站视频为例
# 常规下载
you-get http://www.bilibili.com/video/av3567324
# 仅查看视频信息
```

```
you-get -i http://www.bilibili.com/video/av3567324
# 将视频保存到特定位置,例如，windows 下保存到 E 盘根目录
you-get -o E:/ http://www.bilibili.com/video/av3567324

# 注意！
# （1）各大视频网站需要登录观看的视频无法直接下载
# （2）有些网站视频需要设置代理
```

[8.4]:

```
import jieba.posseg as ps
import jieba
import matplotlib.pyplot as plt
from scipy.misc import imread
from wordcloud import WordCloud,STOPWORDS,ImageColorGenerator

txt = open('三国演义.txt','r',encoding = 'utf-8').read()
counts = {}
def wordAndFrequency(items):
    total = 0
    word_frequency = []
    for i in range(20):
        word, count = items[i]
        a = [word, count]
        total += a[1]
        word_frequency.append(a)
    for i in range(20):
        word_frequency[i][1] = word_frequency[i][1] / total
        word_frequency[i][1] = int(word_frequency[i][1] * 100)
        word_frequency[i][0].encode('utf-8')
        word_frequency[i] = tuple(word_frequency[i])
        print(word_frequency[i])
    return word_frequency

def countFigures():
```

```
    words = ps.cut(txt)
    for w in words:
        if len(w.word) == 1:
            continue
        if w.flag == 'nr':
            rword = w.word
            if rword == '玄德' or rword == '玄德曰':
                rword = '刘备'
            if rword == '孔明' or rword == '孔明曰':
                rword = '诸葛亮'
            if rword[-1:] == '兵':
                continue
            counts[rword] = counts.get(rword, 0) + 1
    items = list(counts.items())
    items.sort(key = lambda x:x[1], reverse = True)
    for i in range(20):
        word, count = items[i]
        print('{0:<10}{1:>5}'.format(word, count))
    return wordAndFrequency(items)

def generateWordCloud(word_frequency):
    sanguo_color = imread('三国演义.jpg')
    wc = WordCloud(font_path='C:/Windows/Fonts/msyh.ttc',
                   background_color="black",   #可以选择 black 或
white
                   mask = sanguo_color,
                   random_state=42,
                   margin=5, width=1800, height=800) # 长宽度控制清
晰程度
    wc.generate_from_frequencies(word_frequency)
    image_colors = ImageColorGenerator(sanguo_color)
    plt.imshow(wc.recolor(color_func=image_colors))
    plt.axis('off')
    plt.figure()
```

```
    plt.show()
word_frequency = countFigures()
generateWordCloud(word_frequency)
```

第 9 章 科学计算和可视化

9.1 问题概述

[9.1]: 科学计算可以用于工程计算，如室内位置定位、动画解算、图形计算等。

[9.2]: 首先安装 numpy+mkl，scipy，然后再安装 matplotlib。

9.2 模块 8：numpy 库的使用

[9.3]: np.ones((x,y), dtype) 创建元素类型为 dtype 的全 1 矩阵；np.empty((x,y), dtype) 创建元素类型为 dtype 的全零矩阵；np.linspace(x,y,n)常见一个由 x 到 y，等分成 n 个元素的数组等方法。

[9.4]: np.sqrt(ndarray)。

[9.5]: 对 ndarray 数组降维可使用 flatten()函数，返回折叠后的一维数组，它的维数就降到了一维。

9.3 实例 17：图像的手绘效果

[9.6]: 三维。

[9.7]: 使用 Image.open().convert('L ')方法可以将彩色图片转换成灰度图片。通过 np.array()将它转换成 ndarray 矩阵，再通过索引处理每一个像素。

[9.8]: fromarray(ndarray)函数可以把 ndarray 类型转换成图像，前提是 ndarray 是个整型矩阵。

9.4 模块 9：matplotlib 库的使用

[9.9]: 使用如下代码可以在屏幕上画出上下两个坐标系图：

```
plt.subplot(211)
plt.xlabel()
...
plt.subplot(212)
```

```
plt.xlabel()
...
plt.show()
```

[9.10]: 坐标系标题、坐标轴、坐标轴对应的标签、单位名称、刻度值。

[9.11]: 设函数为 f（x），则绘制曲线的精简代码可为：plt.plot([f(i) for i in range(1,10)])。

9.5 实例 18：科学坐标图绘制

[9.12]: 代码 18.1 第 13 行的含义为：在点（1，cos2pi）处添加标注。

[9.13]: 代码 18.1 第 32 行图注中的内容在第 13 行被设定。

[9.14]: y 曲线的颜色参数由第 7 行传入，在第 8 行 color = pcolor 设置，形状由 linestyle= “-”设置。z 曲线的颜色形状在第 9 行的第 2 个参数“b--”设置，b 表示 blue，--表示虚线。

9.6 实例 19：多级雷达图绘制

[9.15]: 代码 19.1 第 14 行的含义为：将屏幕划分为 1 行 1 列，即不对屏幕划分，polar = True 表示绘制一个极坐标图。

[9.16]: 代码 19.1 第 18 行的含义为：绘制标题'DOTA 能力值雷达图'于屏幕坐标系中，左下角为(0，0)，x = 0.52，y = 0.95 表示标题位于中部靠上的位置。ha = 'center '表示文本居中。

[9.17]: 代码 19.2 第 27 行的含义为：设置标注，标注内容为 data_labels 内容，位置为（0.94，0.8），文本行间距为 0.1。

程序练习题

[9.1]:略。

[9.2]:略。

[9.3]:

```
from PIL import Image
import numpy as np
```

```
3
4  # 浮雕效果
5  # 自定义你的图像名称
6
7  vec_el = np.pi/8.
8  vec_az = np.pi/8.
9
10 depth = 10.
11 im = Image.open('pic.jpg').convert('L')
12 a = np.asarray(im).astype('float')
13 grad = np.gradient(a)
14 grad_x, grad_y = grad
15 grad_x = grad_x*depth/50
16 grad_y = grad_y*depth/50
17 dx = np.cos(vec_el)*np.cos(vec_az)
18 dy = np.cos(vec_el)*np.cos(vec_az)
19 dz = np.sin(vec_el)
20
21 A = np.sqrt(grad_x ** 2 + grad_y **2 + 1.)
22 uni_x = grad_x/A
23 uni_y = grad_y/A
24 uni_z = 1./A
25 a2 = 255*(dx*uni_x + dy*uni_y + dz*uni_z)
26 a2 = a2.clip(0,255)
27
28 im2 = Image.fromarray(a2.astype('uint8'))
29 im2.save('pic_process.jpg')
```

[9.4]:略。

[9.5]:

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3  import matplotlib
```

```
matplotlib.rcParams['font.family'] = 'SimHei'
matplotlib.rcParams['font.sans-serif'] = ['SimHei']
radar_labels = np.array(['爆发力','技巧熟练度','气势',\
                         '稳定性','速度'])
nAttr = 5
data = np.array([[0.8,0.9,0.9,0.75],
                 [0.85,0.87,0.9,0.9],
                 [0.9,0.6,1.0,0.6],
                 [0.8,0.7,0.7,0.9],
                 [0.6,0.9,0.9,0.92]]
                 )
data_labels = ('马龙','张继科','马琳',\
               '王励勤')
angles = np.linspace(0, 2*np.pi, nAttr, endpoint = False)
data = np.concatenate((data, [data[0]]))
angles = np.concatenate((angles, [angles[0]]))
fig = plt.figure(facecolor = 'white')
plt.subplot(111, polar = True)
plt.plot(angles, data, 'o-', linewidth = 1.5, alpha = 0.2)
plt.fill(angles, data, alpha = 0.2)
plt.thetagrids(angles*180/np.pi, radar_labels, frac =1.2)
plt.figtext(0.05, 0.97, '中国国家队乒乓球运动员参数雷达图',ha =
'left', size = 15)
legend = plt.legend(data_labels, loc = (0.9,0.92), labelspacing
= 0.1)
#plt.setp(legend.get_texts(), fontsize = 'small')
#plt.grid(True)
plt.savefig('holland_radar.JPG')
plt.show()
```

第 10 章 网络爬虫和自动化

10.1 问题概述

[10.1]: 可通过大量获取网页内容来为数据挖掘和机器学习提供训练数据。

10.2 模块 10：requests 库的使用

[10.2]: HTTP 协议中 get 功能单纯的从服务器获取数据，服务器中网站数据不可能变化；post 功能可以通过提交表单修改网站的数据（比如注册），网站数据可能变化。

[10.3]: 函数 post(url 链接，{key：value}) 这个字典中包含要提交的数据，比如 username：Jim、Password:123456 等。

[10.4]: 长度差异可能是编码造成的，text 返回 unicode，content 返回 str。在 Python3.x 环境下长度相同。

10.3 模块 11：beautifulsoup4 库的使用

[10.5]: C。

[10.6]: 假设 BeautifulSoup 对象名称为 soup，

(1) print(soup.head)；

(2) print(soup.body)；

(3) soup.find('p',{ 'id': 'China'})；

(4) soup.find_all(string = re.compile('[\u4e00-\u9fa5'])) 。

10.4 实例 20：中国大学排名爬虫

[10.7]: 可以在输出格式上做进一步改进，按照特定字段排序输出，如根据培养规模排序输出、根据高校名称排序输出等。

[10.8]: 将第 26 行改为 u = allUniv[len(allUniv)-1-i]，将第 34 行改为 main(50)，就可以输出后 50 位大学。

[10.9]: 假设输出山东省的大学排名，将 printUnivList(num)修改为：

```
def printUnivList(num):
    print("{:^4}{:^10}{:^5}{:^8}{:^10}".format('排名','学校名称','省市','总分','培养规模'))
    for i in range(num):
        u = allUniv[i]
        if str(u[2]) == '山东省':
            print("{:^4}{:^10}{:^5}{:^8}{:^10}".format(u[0],u[1],u[2],u[3],u[6]))
        else:
            continue;
```

程序练习题

[10.1]:

```
import requests
from bs4 import BeautifulSoup
allUniv = []
def getHTMLText(url):
    try:
        r = requests.get(url, timeout = 30)
        r.raise_for_status()
        r.encoding = 'utf-8'
        return r.text
    except:
        return ''

def fillUnivList(soup):
    data = soup.find_all('tr')
    for tr in data:
        ltd = tr.find_all('td')
```

```
        if len(ltd) == 0:
            continue
        singleUniv = []
        for td in ltd:
            singleUniv.append(td.string)
        allUniv.append(singleUniv)
def printUnivList(num):
    Univ = sorted(allUniv, key = lambda location: location[2])
    print("{1:^2}{2:{0}^10}{3:{0}^6}{4:{0}^4}{5:{0}^10}".form
at(chr(12288),'排名','学校名称','省市','总分','培养规模'))
    for i in range(num):
        u = Univ[i]

    print("{1:^4}{2:{0}^10}{3:{0}^5}{4:{0}^8.1f}{5:{0}^10}".f
ormat(chr(12288),u[0],u[1],u[2],eval(u[3]),u[6]))

def main(num):
    url='http://www.zuihaodaxue.cn/zuihaodaxuepaiming2016.htm
l'
    html = getHTMLText(url)
    soup = BeautifulSoup(html, 'html.parser')
    fillUnivList(soup)
    print(len(allUniv))
    printUnivList(len(allUniv))

main(10)
```

[10.2]:

```
import requests
from bs4 import BeautifulSoup

# USNEWS 的大学排名默认是卡片显示，为了处理方便，这里将它设置成表格显示
# USNEWS 的大学排名分页，在每一页的读取里加入了读取下一页网址的功能，就
是代码中的 Button 部分
```

```
# 回避其他因素，只进行了美国大学排名。


def getHtml(url):
    response = requests.get(url)
    response.raise_for_status()
    response.encoding = 'utf-8'
    return response.text

def parseHtml():
    url =
'http://colleges.usnews.rankingsandreviews.com/best-
colleges/rankings/national-universities?_sort=rank&_sort-
direction=asc&_mode=table'
    while True:
        text = getHtml(url)
        soup = BeautifulSoup(text)
        button   =   soup.find_all('a',   {'class':   'button
secondary radius shadow full-width '})
        if len(button) == 0:
            break
        url = button[0]['href']
        university = soup.find_all('tr', {'data-parent-
id':'search-application-results-view'})
        rankUniversities(university)

def rankUniversities(universities):
    for i in universities:
        name = i.div.a.string
        rank = i.span.string[1:4]
        if rank == 'ank' or rank == 'nra':
            rank = 'N/A'
        if rank[-1] == 'i':
            rank = rank[:-1]
```

```
        enroll = [j for j in i.find_all('td',{'class':'text-
right'})][1].div.string
        print('{:<55}{:<5}{:<7}'.format(name, rank,
enroll.replace('\n','').replace(' ','')))

parseHtml()
```

[10.3]:

```
from bs4 import BeautifulSoup
import requests

def getHTML(url):
    response = requests.get(url)
    response.raise_for_status()
    response.encoding = 'utf-8'
    return response.text

def findLinks(text):
    soup = BeautifulSoup(text)
    title = soup.find_all('li', {'class':'base_name'})[0].string
    time = soup.find_all('li',{'class':'base_pub'})[0].string
    info = soup.find_all('li', {'class':'base_what'})[0].string
    print('{}-{}{}'.format(title, time, info))
    series = soup.find_all('a',{'site':True})
    z = 0;
    print('当前剧集：')
    for i in series:
        print('{:^3}'.format(i.string),end = '')
        z = z+1
        if z%6 == 0:
            print('\n')

Text =
getHTML('http://www.soku.com/detail/show/XNjA4NTI=?spm=a2h0k.81
```

```
91414.0.0')
findLinks(text)
```

[10.4]:

```
import requests


def getRobotTXT(url):
    response = requests.get(url)
    response.raise_for_status()
    return response.text


def analyseTXT(url,text):
    try:
        file = open('ban.txt','a')
    except:
        file = open('ban.txt','w')
    s = text.split('User-agent:')
    for i in range(1,len(s)):
        sub = s[i].split('\r\n')
        string = '对于爬虫代理：{} ,以下网址不能访问：
'.format(sub[0])
        print(string)
        file.write(string+'\r\n')
        for j in range(1,len(sub)):


            if sub[j][:9] == 'Disallow:':
                string = url+sub[j][10:]
                print(string)
                file.write(string+'\r\n')
    file.close()
url = 'https://www.baidu.com'
robots = '/robots.txt'
text = getRobotTXT(url+robots)
analyseTXT(url, text)
```

[10.5]:

```
import requests
import bs4
import urllib.robotparser

def backhtml(url):
    htmlback=requests.get(url)
    htmlback_encoding=htmlback.apparent_encoding
    soup=bs4.BeautifulSoup(htmlback.text,'html.parser')
    print('网页代码如下：')
    print(soup.prettify())

def robotana(url,urlwant,strin):
    rp = urllib.robotparser.RobotFileParser()
    rp.set_url(url)
    rp.read()
    if rp.can_fetch("*", urlwant) == True:
        print('robots 协议未禁止')
        backhtml('http://' + urlwant)
    else:
        print("十分抱歉，对方网页禁止了您的访问；请看以下规则")
        print(strin)

url=input("请输入您要爬取的网页,不用输'http'：")
realurl=url.split('/',1)[0]
roboturl='http://'+realurl+'/robots.txt'
robotread=requests.get(roboturl)
robotread.encoding=robotread.apparent_encoding
if robotread.status_code != 200:
    print('您爬取的网站没有设置 robots 规则，正在返回网页源代码')
    backhtml('http://'+url)
else:
    print("正在分析规则..")
```

```
    robotana(roboturl,url,robotread.text)
```

[10.6]:

```
import requests
import json
import re
import os
import random

folder_path = 'haha'
def getBaiduImage(url):
    try:
        rsp = requests.get(url, timeout = 10)
        rsp.raise_for_status()

    except:
        print('对不起，百度图片访问失败！程序退出')
        return
    data = json.loads(rsp.text)
    downLoadImage(data)

def tryPic(url):
    form = url[-4:]
    pat = '.*/(.*?)' + form
    pattern = re.compile(pat, re.S)
    filename = re.findall(pattern, url)
    return filename[0], form

def downLoadImage(data):
    global folder_path
    os.mkdir(folder_path)
    imgs = data['data']
    count = 0

```

```
    for i in imgs:
        try:
            url = i['download_url']
            img = requests.get(url, timeout = 10)
            img = img.content
            image,form = tryPic(url)
            FileName = folder_path +'\\'+image+form
            file =  open(FileName,'bw')
            file.write(img)
            print('No.%d success' % count)
        except KeyError:
            print('No.%d failed'%count)

        count += 1

# 用女明星的图片进行测试
url =
'http://image.baidu.com/channel/listjson?pn=0&rn=30&tag1=%E6%98
%8E%E6%98%9F&tag2=%E5%85%A8%E9%83%A8&ftags=%E5%A5%B3%E6%98%8E%E
6%98%9F##内地&ie=utf8'+'1'+'&rn=60&ie=utf8&oe=utf-
8&'+str(random.random())
getBaiduImage(url)
```

[10.7]:

```
import requests
import time
def visitBaidu():
    url = 'https://www.baidu.com/'
    count = 0
    fail = 0
    start = time.time()
    while True:
        if int(time.time()-start) == 30:
            break
```

```
        try:
            r = requests.get(url, timeout = 30)
            r.raise_for_status()
            count += 1
        except:
            fail += 1
            continue
    print('30秒内，访问百度{}次成功 {}次失败'.format(count, fail))

visitBaidu()
```