

《高级开发技术（Python）》

实验指导手册

（学生版 V2.1）

2022 年 8 月 20 日

目 录

一、课程简介	3
二、教学目的和要求	3
三、实验教学中应注意的问题.....	3
四、实验教学内容	4
实验 01、选择与循环（8 题）	4
实验 02、内置函数（5 题）	8
实验 03、列表（10 题）	11
实验 04、元组、字典和集合（10 题）	15
实验 05、函数（10 题）	19
实验 06、字符串、正则表达式（8 题）	24
实验 07、面向对象程序设计（6 题）	27
实验 08、文件与文件夹操作（10 题）	36
实验 09、数据库编程（4 题）	43
实验 10、科学计算（numpy、scipy）	51
实验 11、数据可视化 matplotlib（3 题）	52
实验 12、数据分析 pandas（2 题）	59
实验 13、机器学习（4 题）	62

一、课程简介

课程名称：高级开发技术/ Advanced Programming Technology

实验学时：16 学时

课程类型：专业任选课

适用专业：智能科学与技术专业、三年级（第五学期）、本科

实验环境：Windows 操作系统，Python 3.*，Anaconda3 Spyder

方案版本：V2.1（2022.08）

Python 语言是一种开源的编程语言，可运行于各种操作系统之上，被广泛应用于科学计算及可视化、数据分析与处理、图像处理、人工智能、深度学习、工业控制等领域。因此高级开发技术课程选用 Python 语言，重点学习 python 高级开发技术。本课是智能科学与技术专业学生专业任选课，学习内容包括 Python 运算符、表达式、内置函数、列表、元组、字典、集合等基本数据结构；选择结构和循环结构等基本控制语句；字符串与正则表达式；函数和模块；面向对象程序设计；文件与文件夹操作；异常处理等基本知识，以及可选择性学习数据库、科学计算、数据分析、数据可视化、机器学习等内容。

通过学习本课，学生将掌握 Python 的基本理论，具备利用 Python 解决实际问题的能力，为未来胜任科学计算、数据分析、人工智能、机器学习领域的编程工作奠定基础。

二、教学目的和要求

本课程是智能科学与技术专业的一门专业任选课，大三上学期开出，是培养学生面向数据分析、人工智能、机器学习编程能力的重要课程。

本课程采用 Python 语言介绍高级开发技术。根据本课程需要支撑的毕业要求，确定本课程的教学目标：使学生掌握 Python 语言的基本理论、基本知识、基本技能和基本方法，能够在软件开发过程中，针对特定需求综合数据结构、算法复杂性分析、Python 语言等知识设计、实现相关算法，解决实际问题。从知识角度，使学生理解 Python 的编程模式；掌握 Python 语言的基本语法、内置函数、列表、元组、字典、集合的作用和用法；掌握字符串与正则表达式的使用方法；掌握函数和类的设计与使用；能够在软件开发过程中，针对特定需求，运用 Python 语言设计并实现文件与文件夹操作、数据库编程；在科学计算、数据分析、数据可视化、机器学习、人工智能等现代信息领域，理解 Python 语言的特点和优势，能够选用合适的 Python 技术、扩展库、资源和工具，综合运用，解决特定的工程问题。

三、实验教学中应注意的问题

Python 编程模式中非常重要的一条是代码简单化、问题简单化，同时应保证代码具有

较强的可读性。实验教学是学生熟悉、理解和熟练运用课堂理论教学过程中所学知识的重要环节。在实验教学过程中，一定不要在 Python 程序中带有其他编程语言的痕迹，要尽量从最简单的角度去思考和解决问题、实现自己的想法和思路，保证代码的优雅、简洁，让代码更加 Pythonic。

使用 Python 编程解决问题应充分借鉴和使用成熟的标准库和扩展库，尽量避免自主编写完整的业务逻辑，部分具体操作完全可以借助于 Python 标准库和扩展库来实现，大幅度提高开发效率。

四、实验教学内容

实验 01、选择与循环（8 题）

【实验类型】设计型

【实验学时】2

【实验目标】

- 1) 熟练掌握 Python 选择结构与循环结构语法，理解缩进的作用。
- 2) 熟练运用 for 循环和 while 循环，理解带有 else 子句的循环语句执行流程。
- 3) 理解 break 语句和 continue 语句在循环中的作用。
- 4) 理解并能够熟练设计条件表达式。

【学习重点】

and 不一定是布尔值、for 和 while 中 else 用法。

【实验内容】

（一）必做部分

1、编写程序，运行后用户输入 4 位整数作为年份，判断其是否为闰年。

提示：如果年份能被 400 整除，则为闰年；如果年份能被 4 整除但不能被 100 整除也为闰年。

```

.
////////////////////////////////////+
+
x = input('Please input an integer of 4 digits meaning the year:')
x = int(x)
if x%400==0 or (x%4==0 and not x%100==0):
    print('Yes')
else:
    print('No')
+
////////////////////////////////////+

```

2、编写程序，用户从键盘输入小于 1000 的整数，对其进行因式分解。例如， $10=2 \times 5$ ， $60=2 \times 2 \times 3 \times 5$ 。

```

.
////////////////////////////////////+
+
x = input('Please input an integer less than 1000:')
x = eval(x)
t = x
i = 2
result = []
while True:
    if t==1:
        break
    if t%i == 0:
        result.append(i)
        t = t/i
    else:
        i += 1
print(x, '=', '*' .join(map(str,result)))
+
////////////////////////////////////+

```

3、10 的阶乘（循环 crr118）

编写程序，计算 $1 \times 2 \times 3 \times \dots \times 10$ 。

```

fac=1
for i in range(1,11):
    fac*=i
print("1*2*3...*10=",fac)

```

4、编写程序，至少使用 2 种不同的方法计算 100 以内所有奇数的和。

```

print(sum([i for i in range(1,100) if i%2==1]))
print(sum(range(1,100)[::2]))

```

5、编写程序，实现分段函数计算，如下表所示。

x	y
$x < 0$	0
$0 \leq x < 5$	x
$5 \leq x < 10$	$3x-5$
$10 \leq x < 20$	$0.5x-2$
$20 \leq x$	0

```

x = input('Please input x:')
x = eval(x)

if 0 <= x < 5:
    print(x)
elif 5 <= x < 10:
    print(3*x-5)
elif 10 <= x < 20:
    print(0.5*x-2)
else:
    print(0)

```

6、计算糖果总数（循环 crr127）

编写程序，计算糖果总数。假设有一盒糖果，按照如下方式从中取糖果： 1 个 1 个地取，正好取完。 2 个 2 个地取，还剩 1 个。 3 个 3 个地取，正好取完。 4 个 4 个地取，还剩 1 个。 5 个 5 个地取，还差 1 个。 6 个 6 个地取，还剩 3 个。 7 个 7 个地取，正好取完。 8 个 8 个地取，还剩 1 个。 9 个 9 个地取，正好取完。 请问：这个盒子里一共有多少个糖果？

```
n=1
while 1:
    if n%2==1 and n%3==0 and n%4==1 and n%5==4 and n%6==3 and n%7==0 and
n%8==1 and n%9==0:
        print("There are {} candies in this box.".format(n))
        break
    else:
        n=n+1
```

（二）选做部分（涉及列表）

7、编写程序，生成一个包含 50 个随机整数的列表，然后删除其中所有奇数。

提示：从后向前删除。

```

////////////////////////////////////
import random
x = [random.randint(0,100) for i in range(50)]
print(x)
for i in range(len(x))[:-1]:
    if x[i]%2 == 1:
        del x[i]
print(x)
////////////////////////////////////

```

8、编写程序，生成一个包含 20 个随机整数的列表，然后对其中偶数下标的元素进行降序排列，奇数下标的元素不变。

```

////////////////////////////////////
import random
x = [random.randint(0,100) for i in range(20)]
print(x)
x[::2] = sorted(x[::2], reverse=True)
print(x)
////////////////////////////////////

```

实验 02、内置函数（5 题）

【实验类型】验证型，设计型

【实验学时】2

【实验目标】

- 1) 掌握 Anaconda3 开发环境安装与使用。
- 2) 掌握 Python 程序的创建和运行方式，包括在开发环境中直接运行 Python 程序和在外

命令提示符环境中交互式运行方式。

- 3) 掌握 Python 基本输入输出函数、代码编写规范和注释语句。
- 4) 掌握 Python 标准库和扩展库的导入和使用方式。
- 5) 理解 pip 工具的用法,能够安装新模块、查看已安装模块、升级或卸载已安装模块。
- 6) 理解和掌握 Python 程序的__name__属性。
- 7) 掌握 Python 运算符、常用内置函数的用法。
- 8) 初步掌握列表、元祖、字典、集合的概念和基本用法。
- 9) 初步掌握 lambda 表达式作为函数参数用法。

【实验内容】

(一) 课外自行完成部分

- 1、在 windows 操作系统下,完成 Anaconda3 安装和基本使用。采用最新版本。
- 2、启动 Spyder 开发环境,并熟悉其使用方法(直接运行方式、命令提示符下交互式运行方式)。
 - (1) 使用 print 函数输出字符串“Hello World!”;
 - (2) 使用 print 函数输出表达式 9999**99(9999 的 99 次方)的值。
 - (3) 参照教材介绍的续行符、注释符,在 Spyder 中编写代码予以练习。
- 3、以安装中文分词扩展库 jieba 为例,练习使用 pip 命令在线和离线安装 jieba 扩展库,以及卸载 jieba 扩展库。
- 4、以 random 标准库、numpy 扩展库为例,练习标准库和扩展库的导入方法,以及导入后常用函数的用法。注意: random 标准库中各个函数经常用到。
- 5、理解 Python 程序既可以直接运行,也可以当作模块来使用,以及这两种方式下,__name__变量值得含义。
- 6、在 spyder 环境中,运行并验证下列内置函数。
 - help、id
 - bin、oct、hex、ord、chr
 - list、tuple、dict、set、frozenset、type、isinstance
 - max、min、sum
 - input、print
 - sorted、enumerate、map、reduce、filter、range、zip、eval

（二）必做部分

1、输入任意大的自然数，输出各位数字之和。

```
////////////////////////////////////  
  
#编写程序，输入任意大的自然数，输出各位数字之和  
num = input('请输入一个自然数：')  
print(sum(map(int, num)))  
  
////////////////////////////////////
```

2、圆柱体计算，结果保留两位小数（数值 crr11）

利用 Python 计算，结果保留两位小数：底面半径为 66，高为 24.2 的圆柱体的体积和表面积。结果保留 2 位小数。

```
pi=3.1415  
r=66  
h=24.2  
v=pi*r*r*h  
s=pi*r*r*2+2*pi*r*h  
print("The volume is {:.2f}".format(v))  
print("Surface area is {:.2f}".format(s))
```

3、三门课成绩（混合运算和类型转换 crr18）

小明参加语文，数学和英语考试，输入小明的 3 门成绩，求 3 门成绩的和，平均以及最高和最低分是多少？如果三门课程以权重 0.5，0.3 和 0.2 计入总分，求小明的最终总评成绩是多少？

```
Chinese=eval(input('Chinese scores'))  
Math=eval(input('Maths scores'))  
English=eval(input('English scores'))  
print('Sum score',Chinese+Math+English)  
print('Average score {:.2f}'.format((Chinese+Math+English)/3))  
print('Highest score',max(Chinese,Math,English))  
print('Lowest score',min(Chinese,Math,English))  
print('Total {:.2f}'.format(Chinese*0.5+Math*0.3+English*0.2))
```

4、找零钱（数值 crr41）

商店需要找钱给顾客，现在只有 50 元、5 元和 1 元的人民币若干张。输入一个整数金额值，给出找钱的方案，假设人民币足够多，且优先使用面额大的钱币。

```
money = eval(input(" "))  
m50 = money // 50    # 计算需要的 50 元面额的纸币数量  
money = money % 50    # 使用 50 元面额钱币后剩下的金额  
m5 = money // 5  
money = money % 5  
m1 = money  
print("The number of 50 yuan denomination notes is ", m50)  
print("The number of 5 yuan denomination notes is ", m5)  
print("The number of 1 yuan denomination notes is ", m1)
```

5、进制转换

编写程序，输入一个自然数，输出它的二进制、八进制、十六进制表示形式。（3500）

```
num = int(input('请输入一个自然数: '))  
print('二进制: ', bin(num))  
print('八进制: ', oct(num))  
print('十六进制: ', hex(num))
```

实验 03、列表（10 题）

【实验类型】验证型，设计型

【实验学时】2

【实验目标】

- 1) 熟悉 Python 列表创建与删除。
- 2) 熟练掌握 Python 列表元素增加、删除的方法。
- 3) 熟练掌握列表推导式，能够使用列表推导式生成符合特定条件的列表。
- 4) 熟练掌握切片的用法，能够使用切片访问列表元素，能够使用切片为列表增加、删除、修改元素。
- 5) 熟悉+、*运算符对列表的作用。
- 6) 理解和熟练运用成员测试运算符 in 在遍历序列元素。
- 7) 熟练掌握列表排序方法 sort()和内置排序函数 sorted()的用法。

【学习重点】

- 使用 range 对象创建列表、列表元素访问、列表推导式、列表切片。

【实验内容】

（一）必做部分

1、输入一个包含若干整数的列表，输出新列表，要求新列表中的所有元素来自于输入的列表，并且降序排列。

```
lst=eval(input('输入一个包含若干整数的列表lst:'))  
print(sorted(lst,reverse=True))
```

2、列表生成式（crr22）。使用列表生成式生成列表，其元素为 100 以内所有能被 3 整除的数。

```
x=[i for i in range(1,100) if i%3==0]  
print(x)
```

3、给出一个包含若干整数的列表[23,16,18,19,76,121,33,57,80]，输出一个新列表，要求新列表中只包含原列表中的偶数。提示：可以使用 list、filter、lambda 函数完成。

```
lst=[23,16,18,19,76,121,33,57,80]  
print(list(filter(lambda x:x%2==0,lst)))
```

4、给出一个包含若干整数的列表[23,16,18,19,76,121,33,57,80]，输出列表中的所有整数连乘的结果。提示：可以使用 reduce、lambda 函数完成。

```
from functools import reduce  
lst=[23,16,18,19,76,121,33,57,80]  
print(reduce(lambda x,y:x*y,lst))
```

5、列表切片

编写程序，用户输入一个列表和两个整数作为下标，然后用切片获取并输出介于两个下标之间的元素组成的子列表。例如，用户输入[1,2,3,4,5,6]和 2，5 后，程序输出[3,4,5,6]。

```

x = input('Please input a list:')
x = eval(x)
start, end = eval(input('Please input the start and the end positions:'))
print(x[start:end+1])

```

6、单词长度（crr23）

输入一句英文句子，例如：the scenery along the should be and the mood at the view，输出其中最长的单词及长度。

```

sen=input("输入一句英文句子:")
words=sen.split()
length=[len(word) for word in words]
maxlen=max(length)
print("最长单词{} 有{}字母".format(words[length.index(maxlen)],maxlen))

```

7、输入两个分别包含 2 个整数的列表，分别表示城市中两个地点的坐标，输出两个点之间的曼哈顿距离。

```

lstA=eval(input('输入一个包含 2 个整数的列表 lstA:'))
lstB=eval(input('输入一个包含 2 个整数的列表 lstB:'))
print(sum(map(lambda x,y:abs(x-y),lstA,lstB)))

```

8、高考录取率（列表与元组 crr39）

根据十年高考录取率表创建列表，并完成如下操作：

- ① 计算十年平均录取率。
- ② 找出录取率最高的年份。

表 1 十年高考录取率表

年份	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
录取率	57%	56%	57%	62%	69%	72%	75%	76%	74.3%	74%

```

rate_years=[(2006,57),(2007,56),(2008,57),(2009,62),(2010,69),\
             (2011,72),(2012,75),(2013,76),(2014,74.3),(2015,74)]
rates=[item[1] for item in rate_years]

avg=sum(rates)/len(rates)
print("平均录取率 {:.2f}%".format(avg))

topYear=rate_years[rates.index(max(rates))]
print("录取率最高的年份 {}, 录取率{}%".format(topYear[0],topYear[1]))

```

(二) 选做部分

9、列表二分查找 (crr65)

二分查找是一个经典的算法，用来在有序的一组数中快速找到待查找的数。所谓“二分”，就是每次操作都将查找范围一分为二，即将查找区间缩小一半，直到找到或查询了所有区间都没有找到要查找的数据为止。利用二分查找法，查找[34, 64, 67, 72, 73, 82, 83, 85, 87, 88, 90, 91, 96, 98]中指定数字的索引号。

```

ls=[34, 64, 67, 72, 73, 82, 83, 85, 87, 88, 90, 91, 96, 98]
x=int(input("请输入待查找的数:"))
low=0
high=len(ls)-1

while low<=high:
    mid=(low+high)//2
    if ls[mid]<x:
        low=mid+1
    elif ls[mid]>x:
        high=mid-1
    else:
        print("The number {} is found, and the index is {}".format(x,mid));
        break;
else:
    print("There is no {} in the list.".format(x))

```

10、编写程序，输入一个大于 2 的自然数，然后输出小于该数字的所有素数组成的列表。
要求：利用列表、内置函数 `enumerate()`、`filter()`来实现。

```
////////////////////////////////////  
  
    maxNumber = int(input("请输入一个大于 2 的自然数: "))  
    lst = list(range(2, maxNumber))  
    #最大整数的平方根  
    m = int(maxNumber**0.5)  
    for index,value in enumerate(lst):  
        #如果当前数字已大于最大整数的平方根，结束判断  
        if value > m:  
            break  
        #对该位置之后的元素进行过滤  
        lst[index+1:] = filter(lambda x: x%value != 0, lst[index+1:])  
    print(lst)  
  
////////////////////////////////////
```

实验 04、元组、字典和集合（10 题）

【实验类型】验证型，设计型

【实验学时】2

【实验目标】

- 1) 熟悉 Python 元组、字典、集合的创建与删除。
- 2) 熟练掌握生成器推导式用法。
- 3) 理解和熟练运用成员测试运算符 `in` 在遍历序列元素。
- 4) 掌握序列解包的用法。

【学习重点】

- 元组创建与访问、生成器推导式、字典的 `get()`方法、集合运算、序列解包。

【实验内容】

（一）必做部分

1、餐厅下午茶（列表与元组 crr66）

某餐厅推出了优惠下午茶套餐活动。顾客可以以优惠的价格从给定的糕点和给定的饮料中各选一款组成套餐。已知，指定的糕点包括松饼(Muffins)、提拉米苏(Tiramisu)、芝士蛋

糕(Cheese Cake)和三明治(Sandwich); 指定的饮料包括红茶(Black tea,)、咖啡(Coffee)和橙汁(Orange Juice)。请问, 可以搭配出哪些套餐供客户选择? 请依次打印输出各种套餐。

```
snacks = ['Muffins', 'Tiramisu', 'Cheese Cake', 'Sandwich']  
drinks = ['Black tea', 'Coffee', 'Orange Juice']  
  
menus = []  
for snack in snacks:  
    for drink in drinks:  
        menu = (snack, drink)  
        menus.append(menu)  
for menu in menus:  
    print(menu)
```

2、生成 1000 个 0~100 之间的随机整数, 采用集合统计每个元素的出现次数。

```
////////////////////////////////////  
import random  
x = [random.randint(0,100) for i in range(1000)]  
  
# 使用集合  
s = set(x)  
for v in s:  
    print(v, ': ', x.count(v))  
print('='*30)  
////////////////////////////////////
```

3、输入两个分别包含若干整数的列表 lstA 和 lstB, 输出一个字典, 要求使用列表 lstA 中的元素作为键, 列表 lstB 中元素作为值, 并且最终字典中的元素数量取决于 lstA 和 lstB 中元素最少的列表的数量。

```
////////////////////////////////////  
lstA=eval(input('输入一个包含若干整数的列表 lstA:'))  
lstB=eval(input('输入一个包含若干整数的列表 lstB:'))  
result=dict(zip(lstA,lstB))  
print(result)  
////////////////////////////////////
```


4、输入一个字符串，输出其中出现次数最多的字符及其出现的次数。要求使用字典完成。

```
////////////////////////////////////  
  
data=input('输入一个字符串')  
d=dict()  
for ch in data:  
    d[ch]=d.get(ch,0)+1  
mostCommon=max(d.items(),key=lambda item:item[1])  
print(mostCommon)  
  
////////////////////////////////////
```

5、籍贯分布（字典的整体操作 crr92）

小夏和小迪接到一个调研任务，需要按省份统计班级同学的籍贯分布情况。他们决定两人分头统计男生和女生的籍贯分布，最后再汇总结果。

已知小夏统计的女生籍贯分布是：dicGirls={'Jiangsu':3,'Zhejiang':2,'Jilin':1}；小迪统计的男生籍贯分布是：dicBoys={'Jiangsu':8,'Zhejiang':5,'Shandong':5,'Anhui':4,'Fujian':2}。请编写程序将两人的调研结果合并并且输出。

```
////////////////////////////////////  
  
dicGirls={'Jiangsu':3,'Zhejiang':2,'Jilin':1}  
dicBoys={'Jiangsu':8,'Zhejiang':5,'Shandong':5,'Anhui':4,'Fujian':2}  
dic=dicBoys.copy()  
for k,v in dicGirls.items():  
    dic[k]=dic.get(k,0)+v  
print(dic)  
  
////////////////////////////////////
```

（二）选做部分

6、输入两个集合 setA 和 setB，分别输出它们的交集、并集和差集 setA-setB。

```
////////////////////////////////////
```

```

#编写程序, 输入两个集合 setA 和 setB, ↵
#分别输出它们的交集、并集和差集 setA-setB↵
setA = eval(input('请输入一个集合: ')) ↵
setB = eval(input('再输入一个集合: ')) ↵
print('交集: ', setA & setB) ↵
print('并集: ', setA | setB) ↵
print('setA-setB: ', setA - setB)↵
↵
////////////////////////////////////

```

7、输入包含若干集合的列表，输出这些集合的并集。要求使用 `reduce()`函数和 `lambda` 表达式完成。

```

////////////////////////////////////

from functools import reduce↵
lst=eval(input('输入一个包含若干集合的列表 lst:'))↵
print(reduce(lambda x,y:x | y,lst))↵
↵
////////////////////////////////////

```

8、设计一个字典，并编写程序。用户输入内容作为键，然后输出字典中对应的值。如果用户输入的键不存在，则输出“输入的键不存在”。

```

////////////////////////////////////

d = {1:'a', 2:'b', 3:'c', 4:'d'}↵
v = input("Please input a key:")↵
v = eval(v)↵
print(d.get(v, '您输入的的键不存在'))↵
↵
////////////////////////////////////

```

9、编写程序，输入一个大于 2 的自然数，输出小于该数字的所有素数组成的集合。要求：利用集合对象的 `discard()`方法来实现。

```

////////////////////////////////////

```

```

maxNumber = int(input('请输入一个大于 2 的自然数: '))
numbers = set(range(2, maxNumber))

# 最大数的平方根, 以及小于该数字的所有素数
m = int(maxNumber**0.5)+1
primesLessThanM = [p for p in range(2, m)
if 0 not in [p%d for d in range(2, int(p**0.5)+1)]]

# 遍历最大整数平方根之内的自然数
for p in primesLessThanM:
    for i in range(2, maxNumber//p+1):
        # 在集合中删除该数字所有的倍数
        numbers.discard(i*p)
print(numbers)

////////////////////////////////////

```

10、国土面积（字典的整体操作 crr90）

已知字典{'Canada': 997.1, 'Russia': 1707.5, 'China': 960.1}，按照国家名的升序输出 Russia、Canada、China 三个国家和对应的国土面积

```

dicCountries={'Canada': 997.1, 'Russia': 1707.5, 'China': 960.1}
ls=sorted(dicCountries)

for country in ls:
    print(country, dicCountries[country])

```

实验 05、函数（10 题）

【实验类型】设计型

【实验学时】2

【实验目标】

- 1) 掌握 Python 函数定义的语法。
- 2) 掌握默认值参数、关键参数和可变长度参数的用法。
- 3) 理解 return 语句的作用；理解变量作用域、局部变量、全局变量的概念。
- 4) 理解并能够运用 lambda 表达式，尤其是 lambda 表达式做其他函数参数的用法。

【实验内容】

（一）必做部分

1、判断奇数（函数 crr171）

编写 isOdd() 函数，该函数应有一个整数参数，如果该参数为奇数，函数返回 True，否则返回 False。并编写主程序调用该函数进行测试。

```
def isOdd(n):  
    return n % 2 != 0  
n = int(input('输入一个整数: '))  
print(f'{n} is odd') if isOdd(n) else print(f'{n} is not odd')
```

2、闰年（函数 crr174）

编写一个程序，在主程序中求 1900 ~ 2020 年中所有的闰年，每行输出 5 个年份。闰年即能被 4 整除但不能被 100 整除，或者能被 400 整除的年份。要求定义一个函数 isLeapYear(year)，该函数用来判断某年是否为闰年，是闰年则函数返回 True，否则返回 False。并编写主程序调用该函数进行测试。

```
def isLeapYear(year):  
    return year % 400 == 0 or (year % 4 == 0 and year % 100 != 0)  
count = 0  
for n in range(1900, 2021):  
    if isLeapYear(n) is True:  
        print(n, end='\t')  
        count += 1  
        if count % 5 == 0:  
            print()
```

3、编写函数，判断一个整数是否为素数，并编写主程序调用该函数。

```
////////////////////////////////////
```

```

↵
import math↵
↵
def isPrime(v):↵
    n = int(math.sqrt(v)+1)↵
    for i in range(2,n):↵
        if v%i==0:↵
            return 'No'↵
    return 'Yes'↵
↵
print(isPrime(37))↵
print(isPrime(60))↵
print(isPrime(113))↵
↵

```

//

4、编写函数，接收一个字符串，分别统计大写字母、小写字母、数字、其他字符的个数，并以元组的形式返回结果。

//

```

def demo(v):↵
    capital = little = digit = other = 0↵
    for i in v:↵
        if 'A'<=i<='Z':↵
            capital += 1↵
        elif 'a'<=i<='z':↵
            little += 1↵
        elif '0'<=i<='9':↵
            digit += 1↵
        else:↵
            other += 1↵
    return (capital,little,digit,other)↵
↵
x = 'capital = little = digit = other = 0'↵
print(demo(x))↵

```

//

5、（可变长度参数）编写函数，可以接收任意多个整数并输出其中的最大值和所有整数之和。

//

```
↵  
def demo(*v):↵  
    print(v)↵  
    print(max(v))↵  
    print(sum(v))↵  
↵  
demo(1,2,3)↵  
demo(1,2,3,4)↵  
demo(1,2,3,4,5)↵
```

//

（二）选做部分

6、lambda()函数应用（crr80）

利用 lambda()函数按行输出列表[3, 5, -7, -1, 0, -9]中所有的负数。

```
f = lambda x: x<0↵  
list = [3, 5, -7, 4, -1, 0, -9]↵  
for i in filter(f, list):↵  
    print(i)↵
```

7、完数（函数 crr177）

在主程序中输入一个整数 n，判断该数是否为完数。所谓完数（perfect number），是一些特殊的自然数，它所有的真因子（即除了自身以外的约数）的和，恰好等于它本身。如 $6 = 1 + 2 + 3$ ，所以 6 是完数。要求定义一个函数，用来判断某个整数是否为完数，是完数则函数返回 1，否则返回 0。最后在主程序中测试该函数。是完数输出"YES"，不是输出"NO"。并编写主程序调用该函数进行测试。

```

def fun(n):
    s = 0
    for i in range(1, n):
        if n % i == 0:
            s = s + i
    return 1 if s == n else 0
#def fun2(n):
#    ls = [i for i in range(1, n) if n % i == 0]
#    return 1 if sum(ls) == n else 0
n = int(input('请输入一个整数: '))
print('YES' if fun(n) == 1 else print("NO"))

```

8、递推法（book 实验 9）

假设一段楼梯共 15 个台阶，小明一步最多能上 3 个台阶。使用递推法计算小明上这段楼梯一共有多少种方法。编写主程序调用该函数进行测试。

```

def climbStairs1(n):
    #递推法
    a = 1
    b = 2
    c = 4
    for i in range(n-3):
        c, b, a = a+b+c, c, b
    return c
print(climbStairs1(15))

```

9、递归法（book 实验 9）

假设一段楼梯共 15 个台阶，小明一步最多能上 3 个台阶。使用递归法计算小明上这段楼梯一共有多少种方法。编写主程序调用该函数进行测试。

```
def climbStairs2(n):
    #递归法
    first3 = {1:1, 2:2, 3:4}
    if n in first3.keys():
        return first3[n]
    else:
        return climbStairs2(n-1) + \
            climbStairs2(n-2) + \
            climbStairs2(n-3)
print(climbStairs2(15))
```

10、编写函数，模拟内置函数 sum()。

```
////////////////////////////////////
.
def Sum(v):
    s = 0
    for i in v:
        s += i
    return s
x = [1,2,3,4,5]
print(Sum(x))
x = (1,2,3,4,5)
print(Sum(x))
+1
////////////////////////////////////
```

实验 06、字符串、正则表达式（8 题）

【实验类型】验证型、设计型

【实验学时】2

【实验目标】

- 1) 熟练使用运算符%和 format()方法进行字符串格式化。
- 2) 熟练掌握字符串分割、连接、排版、替换、查找、删除指定字符等方法。
- 3) 了解正则表达式语法、常用元字符。

- 4) 熟悉常用正则表达式的写法。
- 5) 掌握 Python 正则表达式模块 re 的用法。

【实验内容】

1、输出随机字符串

编写程序，输出由英文字母大小写或数字组成的长度为 10 且不重复的随机字符串。

```
import string
import random
x = string.ascii_letters + string.digits
print("".join(random.sample(x, 10)))
```

2、货币的转换（字符串 crr107）

编写程序，实现货币的转换。从键盘输入美元的币值，转换为人民币的币值并输出，结果保留 2 位小数。假设美元兑换人民币的汇率是 6.868。运行效果如下。

输入：请输入要兑换的美元，以\$结束：3.5\$

输出：24.04

```
my=input("请输入要兑换的美元，以$结束：")
rmb=float(my[0:-1])*6.868
print("{:.2f}".format(float(rmb)))
```

3、凯撒加密（book 实验 19）

凯撒加密算法的原理：把明文中每个英文字母替换为该字母在字母表中后面第 k 个字母。如果后面 k 个字母超出字母表的范围，则字母表首尾相接，也就是字母 Z 的下一个字母是 A，字母 z 的下一个字母是 a。要求明文中的大写字母、小写字母分别进行处理。编写程序，输入一个字符串作为明文和一个整数作为凯撒加密算法的密钥，最后输出加密后的结果。

提示：利用字符串和切片操作。

```

import string
def kaisa(s, k):
    lower = string.ascii_lowercase    #小写字母
    upper = string.ascii_uppercase    #大写字母
    before = string.ascii_letters
    after = lower[k:] + lower[:k] + upper[k:] + upper[:k]
    table = ''.maketrans(before, after)    #创建映射表
    return s.translate(table)

s = "Python is a greate programming language. I like it!"

print(kaisa(s,3))

```

4、字符替换（曹洁）

使用正则表达式，编写程序将所有地址中的 ROAD 写成 RD.

例如，字符串 “this is road,This is ROAD,This is Road”，替换成 “this is RD.,this is RD.,this is RD.”

```

import re
s='this is road .....,This is ROAD,This is Road'
x=re.sub(r'\broad\b','RD.',s.lower())
print(x)

```

5、检测字母或数字（曹洁）

使用正则表达式，编写程序检测字符串是否是由字母或者数字组成。

```

import re
pattern=re.compile('[a-zA-Z0-9]+$')
str1='abcd425A'
str2="".join(pattern.findall(str1))
print(str2==str1)

```

6、纠正字母

假设有一段英文“I am a teacher,I am man, and I am 38 years old.I am not a busInessman.”，其中有单词 busInessman 中间的字母“i”误写为“I”，请编写程序使用正则表达式进行纠正。运行效果如下。

```

import re
x = "I am a teacher.I am man, and I am 38 years old.I am not a businessman."
print(x)
result = re.sub(r'\B\b', 'i', x)
print(result)

```

7、输出英文中所有长度为 3 个字母的单词

编写程序，用户输入一段英文，然后输出这段英文中所有长度为 3 个字母的单词。

```

import re
x = input('Please input a string:')
pattern = re.compile(r'\b[a-zA-Z]{3}\b')
print(pattern.findall(x))

```

8、查找并输出字符串 S 中 AABB 形式的词语，如踏踏实实、密密麻麻等等。

假定有一个字符串 s= '例如踏踏实实、密密麻麻、简简单单、时时刻刻是一些。。。'，请使用 re 模块查找并输出其中所有 AABB 形式的词语。

```

import re
s = '例如踏踏实实、密密麻麻、简简单单、时时刻刻是一些。。。'
pattern = r'([\u4e00-\u9fa5])\2([\u4e00-\u9fa5])\3'
r = re.findall(pattern, text)
if r:
    for word in r:
        print(word[0])

```

实验 07、面向对象程序设计（6 题）

【实验类型】设计型

【实验学时】2

【实验目标】

- 1) 理解 Python 类定义语法。
- 2) 理解类成员、实例成员和私有成员、公有成员的概念。
- 3) 掌握 Python 类中属性的定义，以及属性处理方式。

【学习重点】

成员方法、属性。

【实验内容】

1、定义一个圆柱体类 Cylinder

定义一个圆柱体类 Cylinder，包含底面半径和高两个属性（数据成员）；包含一个可以计算圆柱体体积的方法。然后编写相关程序测试相关功能。

```
class Cylinder:
    num=3.14
    def __init__(self,a,b):
        self.r=a
        self.h=b
    def volume(self):
        r=float(self.r)
        h=float(self.h)
        self.v=r*r*h*Cylinder.num
    def print(self):
        print(self.v)
#主函数
a=input("请输入圆柱体的半径:")
b=input("请输入圆柱体的高:")
c1=Cylinder(a,b)
c1.volume()
print("圆柱体的体积为:",end=' ')
c1.print()
```

2、根据人的特征，参照下面要求，定义一个 Person 类（曹洁）

该类根据人的特征，除了姓名、年龄、性别三个属性外，还有一个 show 方法，用于显示姓名、年龄、性别。

```
class Person:
    def __init__(self,name,age,sex):
        self.name = name
        self.age = age
        self.sex = sex
    def setName(self,name):
        self.name = name
    def getName(self):
        return self.name
    def setAge(self,age):
        self.age = age
    def getAge(self):
        return self.age
    def setSex(self,sex):
        self.sex = sex
    def getSex(self):
        return self.sex
    def show(self):
        return 'name:{0}, age:{1}, sex:{2}'.format(self.name,self.age,self.sex)
```

3、根据下面提示，继承 Person 类生成 Teacher 类（曹洁）

Teacher 类继承了 Person 类的姓名、年龄、性别三个属性外，还有教授的课程（course）、教师工资（salary）。此外，教师类还应该有上课 setCourse（）、涨工资 setSalary（）这样行为。


```

class Stack:
    def __init__(self, size=10): #创建栈对象并进行初始化，默认栈大小为 10
        self.__content = [] # 使用列表存放栈的元素
        self.__size = size # 初始栈大小
        self.__current = 0 # 栈中元素个数初始化为 0
    def empty(self): #清空栈
        self.__content = []
        self.__current = 0
    def isEmpty(self): #测试栈是否为空
        return not self.__content
    def setSize(self, size): #调整栈的大小，可以增大或缩小栈空间
        # 如果缩小空间时指定的新大小，小于已有元素个数
        # 则删除指定大小之后的已有元素
        if size < self.__current:
            for i in range(size, self.__current)[::-1]:
                del self.__content[i]
            self.__current = size
        self.__size = size
    def isFull(self): #测试栈是否已满
        return self.__current == self.__size
    def push(self, v): #入栈
        if self.__current < self.__size:
            self.__content.append(v)
            self.__current = self.__current + 1 # 栈中元素个数加 1
        else:
            print("Stack Full!")
    def __str__(self):
        return str(self.__content)
    __repr__ = __str__
    def pop(self): #出栈
        if self.__content:
            self.__current = self.__current - 1 # 栈中元素个数减 1
            return self.__content.pop()
        else:
            print("Stack is empty!")
    def show(self): #显示当前栈对象中的元素
        print(self.__content)
    def showRemainderSpace(self): #显示栈对象剩余空间大小
        print("Stack can still PUSH", self.__size - self.__current, "elements.")
sk = Stack(6)

```



1	2	3
4	5	6
7	8	9
10	11	12
13	14	15
16	17	18
19	20	21
22	23	24
25	26	27
28	29	30
31	32	33
34	35	36
37	38	39
40	41	42
43	44	45
46	47	48
49	50	51
52	53	54
55	56	57
58	59	60
61	62	63
64	65	66
67	68	69
70	71	72
73	74	75
76	77	78
79	80	81
82	83	84
85	86	87
88	89	90
91	92	93
94	95	96
97	98	99
100	101	102
103	104	105
106	107	108
109	110	111
112	113	114
115	116	117
118	119	120
121	122	123
124	125	126
127	128	129
130	131	132
133	134	135
136	137	138
139	140	141
142	143	144
145	146	147
148	149	150
151	152	153
154	155	156
157	158	159
160	161	162
163	164	165
166	167	168
169	170	171
172	173	174
175	176	177
178	179	180
181	182	183
184	185	186
187	188	189
190	191	192
193	194	195
196	197	198
199	200	201
202	203	204
205	206	207
208	209	210
211	212	213
214	215	216
217	218	219
220	221	222
223	224	225
226	227	228
229	230	231
232	233	234
235	236	237
238	239	240
241	242	243
244	245	246
247	248	249
250	251	252
253	254	255
256	257	258
259	260	261
262	263	264
265	266	267
268	269	270
271	272	273
274	275	276
277	278	279
280	281	282
283	284	285
286	287	288
289	290	291
292	293	294
295	296	297
298	299	300
301	302	303
304	305	306
307	308	309
310	311	312
313	314	315
316	317	318
319	320	321
322	323	324
325	326	327
328	329	330
331	332	333
334	335	336
337	338	339
340	341	342
343	344	345
346	347	348
349	350	351
352	353	354
355	356	357
358	359	360
361	362	363
364	365	366
367	368	369
370	37	

////////////////////////////////////

1	2	3
4	5	6
7	8	9
10	11	12
13	14	15
16	17	18
19	20	21
22	23	24
25	26	27
28	29	30
31	32	33
34	35	36
37	38	39
40	41	42
43	44	45
46	47	48
49	50	51
52	53	54
55	56	57
58	59	60
61	62	63
64	65	66
67	68	69
70	71	72
73	74	75
76	77	78
79	80	81
82	83	84
85	86	87
88	89	90
91	92	93
94	95	96
97	98	99
100	101	102
103	104	105
106	107	108
109	110	111
112	113	114
115	116	117
118	119	120
121	122	123
124	125	126
127	128	129
130	131	132
133	134	135
136	137	138
139	140	141
142	143	144
145	146	147
148	149	150
151	152	153
154	155	156
157	158	159
160	161	162
163	164	165
166	167	168
169	170	171
172	173	174
175	176	177
178	179	180
181	182	183
184	185	186
187	188	189
190	191	192
193	194	195
196	197	198
199	200	201
202	203	204
205	206	207
208	209	210
211	212	213
214	215	216
217	218	219
220	221	222
223	224	225
226	227	228
229	230	231
232	233	234
235	236	237
238	239	240
241	242	243
244	245	246
247	248	249
250	251	252
253	254	255
256	257	258
259	260	261
262	263	264
265	266	267
268	269	270
271	272	273
274	275	276
277	278	279
280	281	282
283	284	285
286	287	288
289	290	291
292	293	294
295	296	297
298	299	300
301	302	303
304	305	306
307	308	309
310	311	312
313	314	315
316	317	318
319	320	321
322	323	324
325	326	327
328	329	330
331	332	333
334	335	336
337	338	339
340	341	342
343	344	345
346	347	348
349	350	351
352	353	354
355	356	357
358	359	360
361	362	363
364	365	366
367	368	369
370	37	


```

# 头节点
head = Node(0)
p = head
for i in range(1, 10):
    # 依次生成 10 个数字，并创建相应的节点
    # 把节点连接到链表的尾部
    n = Node(i)
    p.insertAfter(n)
    p = n

p = head
# 遍历链表节点，在值为 3 的节点后面插入值为 3.5 的新节点
while True:
    if p.data == 3:
        p.insertAfter(Node(3.5))
        break
    else:
        p = p.next

p = head
# 删除节点 7 后面的节点
while True:
    if p.data == 7:
        p.deleteAfter()
        break
    p = p.next
    if p.next == None:
        break

p = head
# 遍历链表并输出每个节点的值
while True:
    print(p.data)
    if p.next == None:
        break
    p = p.next

////////////////////////////////////

```

6、设计一个三维向量类，并实现向量的加法、减法以及向量与标量的乘法和除法运算

////////////////////////////////////

CH

```
class Vector3:
    def __init__(self, x=0, y=0, z=0):
        self.X = x
        self.Y = y
        self.Z = z

    def __add__(self, n):
        r = Vector3()
        r.X = self.X + n.X
        r.Y = self.Y + n.Y
        r.Z = self.Z + n.Z
        return r

    def __sub__(self, n):
        r = Vector3()
        r.X = self.X - n.X
        r.Y = self.Y - n.Y
        r.Z = self.Z - n.Z
        return r

    def __mul__(self, n):
        r = Vector3()
        r.X = self.X * n
        r.Y = self.Y * n
        r.Z = self.Z * n
        return r

    def __truediv__(self, n):
        r = Vector3()
        r.X = self.X / n
        r.Y = self.Y / n
        r.Z = self.Z / n
        return r

    def __floordiv__(self, n):
        r = Vector3()
        r.X = self.X // n
        r.Y = self.Y // n
        r.Z = self.Z // n
        return r
```



```

def show(self):
    print((self.X,self.Y,self.Z))
+
+
v1 = Vector3(1,2,3)+
v2 = Vector3(4,5,6)+
v3 = v1+v2+
v3.show()+
v4 = v1-v2+
v4.show()+
v5 = v1*3+
v5.show()+
v6 = v1/2+
v6.show()+
////////////////////////////////////

```

实验 08、文件与文件夹操作（10 题）

【实验类型】验证型，设计型

【实验学时】2

【实验目标】

- 1) 掌握 Python 读写文本文件、二进制文件的方法。
- 2) 掌握 struct 和 pickle 模块的使用方法。
- 3) 掌握 os 模块的用法。
- 4) 掌握 os.path 模块的用法。
- 5) 掌握 shutil 模块的用法。

【实验内容】

（一）文件操作

1、文本文件读写基础。编写程序，在 D 盘根目录下创建一个文本文件 test.txt，并向其中写入字符串 hello world。

```

////////////////////////////////////

```

```
fp = open(r'D:\test.txt','a+')↵
print('hello world',file=fp)↵
fp.close()↵
```

////////////////////////////////////

2、编写一个程序 `demo.py`，要求运行该程序后，生成 `demo_new.py` 文件，其中内容与 `demo.py` 一样，只是在每一行的后面加上行号。要求行号以 `#` 开始，并且所有行的 `#` 符号垂直对齐。

////////////////////////////////////

```
filename = demo.py'↵
with open(filename, 'r') as fp: ↵
    lines = fp.readlines()↵
maxLength = len(max(lines, key=len))↵
↵
lines = [line.rstrip().ljust(maxLength)+'#+str(index)+'\n' for index, ↵
        line in enumerate(lines)]↵
with open(filename[:-3]+'_new.py', 'w') as fp: ↵
    fp.writelines(lines)↵
↵
```

////////////////////////////////////

3、编写程序，使用 `pickle` 模块将包含学生成绩的字典保存为二进制文件，然后再读取内容并显示。

////////////////////////////////////

////////////////////////////////////

4、计算文件 MD5 的值。MD5 是一种常用的哈希算法，不论原始信息长度如何，总是计算得到一个固定长度的二进制字符串。该算法对原文的改动非常敏感，因此该算法被用于检验信息是否被修改过，用于文件完整性检验，或者数字签名。

Python 标准库 `hashlib` 中 `md5()` 函数可以用来计算字节串的 MD5 值。编写一个程序，要求输入一个文件名，然后输出该文件的 MD5 值，如果文件不存在就给出相应提示。

////////////////////////////////////

7、编写程序，用户输入一个目录和一个文件名，搜索该目录及其子目录中是否存在该文件。

```
////////////////////////////////////  
  
import os  
+  
directory = input("请输入文件夹: ")  
filename = input("请输入文件名: ")  
+  
paths = os.walk(directory)  
for root, dirs, files in paths:  
    if filename in files:  
        print("Yes")  
        break  
    else:  
        print("No")  
+  
////////////////////////////////////
```

8、编写程序，实现磁盘垃圾文件清理功能。要求程序运行时，通过命令行参数指定要清理的文件夹，然后删除该文件夹及其子文件夹中所有扩展名为 tmp、log、obj、txt 以及大小为 0 的文件。

```
////////////////////////////////////
```

```

from os.path import isdir, join, splitext, getsize +
from os import remove, listdir +
import sys +
+
# 指定要删除的文件类型 +
filetypes = ['.tmp', '.log', '.obj', '.txt'] +
+
def delCertainFiles(directory): +
    if not isdir(directory): +
        return +
    for filename in listdir(directory): +
        temp = join(directory, filename) +
        if isdir(temp): +
            delCertainFiles(temp) +
        elif splitext(temp)[1] in filetypes or getsize(temp) == 0: +
            # 删除指定类型的文件或大小为 0 的文件 +
            remove(temp) +
            print(temp, 'deleted....') +
        +
    directory = 'c:\\dgg' +
    delCertainFiles(directory) +

```

////////////////////////////////////

9、在某个文件夹中有若干子文件夹，在每个子文件夹中都有一些扩展名为 pptx 的 powerpoint2007 文件。编写程序，统计指定文件夹内所有 PPTX 文件的幻灯片总数量。

提示：需要安装扩展库 python-pptx，利用 os，os.path 模块。

////////////////////////////////////


```

import os
import os.path
import pptx
total=0
def pptCount(path):
    global total
    for subpath in os.listdir(path):
        subpath=os.path.join(path,subpath)
        if os.path.isdir(subpath):
            pptCount(subpath)
        elif subpath.endswith('.pptx'):
            print(subpath)
            presentation=pptx.Presentation(subpath)
            total+=len(presentation.slides)
    +
x='c:\\d\\f'
pptCount(x)
print(total)
+
////////////////////////////////////

```

10、编写程序，模拟生成若干同学的成绩并写入 Excel 文件，其中学生姓名和课程名称均可重复，也就是允许出现同一门课程的多次成绩，最后统计所有学生每门课程的最高成绩，并写入新的 Excel 文件。

实验步骤：

首先，在命令提示符环境使用 `pip install openpyxl` 命令安装扩展库 `openpyxl`。然后编写代码。

```

////////////////////////////////////

from random import choice, randint
from openpyxl import Workbook, load_workbook

#生成随机数据
def generateRandomInformation(filename):
    workbook = Workbook()
    worksheet = workbook.worksheets[0]
    worksheet.append(['姓名','课程','成绩'])

    #中文名字中的第一、第二、第三个字
    first = '赵钱孙李'
    middle = '伟昀琛东'
    last = '坤艳志'

```

```

subjects = ('语文','数学','英语')
for i in range(200):
    name = choice(first)
    #按一定概率生成只有两个字的中文名字
    if randint(1,100)>50:
        name = name + choice(middle)
    name = name + choice(last)
    #依次生成姓名、课程名称和成绩
    worksheet.append([name, choice(subjects), randint(0, 100)])
#保存数据，生成 Excel 2007 格式的文件
workbook.save(filename)

def getResult(oldfile, newfile):
    #用于存放结果数据的字典
    result = dict()

    #打开原始数据
    workbook = load_workbook(oldfile)
    worksheet = workbook.worksheets[0]

    #遍历原始数据
    for row in worksheet.rows:
        if row[0].value == '姓名':
            continue
        #姓名,课程名称,本次成绩
        name, subject, grade = map(lambda cell:cell.value, row)

        #获取当前姓名对应的课程名称和成绩信息
        #如果 result 字典中不包含，则返回空字典
        t = result.get(name, {})
        #获取当前学生当前课程的成绩，若不存在，返回 0
        f = t.get(subject, 0)
        #只保留该学生该课程的最高成绩
        if grade > f:
            t[subject] = grade
            result[name] = t

    workbook1 = Workbook()
    worksheet1 = workbook1.worksheets[0]
    worksheet1.append(['姓名','课程','成绩'])

    #将 result 字典中的结果数据写入 Excel 文件
    for name, t in result.items():
        print(name, t)

```

```

        for subject, grade in t.items():
            worksheet1.append([name, subject, grade])
        workbook1.save(newfile)
    if name == ' main ':
        oldfile = r'd:\test.xlsx'
        newfile = r'd:\result.xlsx'
        generateRandomInformation(oldfile)
        getResult(oldfile, newfile)

```

//

实验 09、数据库编程（4 题）

【实验类型】设计型

【实验学时】2

【实验目标】

- 1) 掌握运用扩展库 openpyxl 操作 Excel 文件。
- 2) 掌握 SQL 语句的编写。
- 3) 掌握生成器函数的编写和使用。
- 4) 掌握标准库 sqlite3 的用法。

【实验内容】

1、创建 sqlite3 数据库，并使用 Python 访问。

- 1) 创建一个数据库 stuinfo，并在其中创建数据库表 student，表中包含 stuid(学号)、stuname(姓名)、birthday(出生日期)、sex(性别)、address(家庭地址)、rxrq(入学日期) 6 列，其中 stuid 设为主键，并添加 5 条记录。
- 2) 将所有记录的 rxrq 属性更新为 2017-9-1。
- 3) 查询性别为“女”的所有学生的 stuname 和 address 字段值。

//

```

import sqlite3
con=sqlite3.connect("sales.db")
cur=con.cursor()
cur.execute("DROP table book")#第一次运行时这句删掉，之后再运行加上。
con.execute("create table book(stuid primary key,stuname varchar(20),birthday integer(20),sex
varchar(20),address varchar(20),rxrq integer(20))")
cur.execute("insert into book(stuid,stuname,birthday,sex,address,rxrq) values (1,'张一
',19980301,'男','黄河路 1 号',20160901)")
cur.execute("insert into book(stuid,stuname,birthday,sex,address,rxrq) values (2,'张二
',19980302,'女','黄河路 2 号',20160902)")
cur.execute("insert into book(stuid,stuname,birthday,sex,address,rxrq) values (3,'张三
',19980303,'男','黄河路 3 号',20160903)")
cur.execute("insert into book(stuid,stuname,birthday,sex,address,rxrq) values (4,'张四
',19980304,'男','黄河路 4 号',20160904)")
cur.execute("insert into book(stuid,stuname,birthday,sex,address,rxrq) values (5,'张五
',19980305,'女','黄河路 5 号',20160905)")
print('记录计入完成')
cur.execute('select * from book')
values = cur.fetchall()
for row in values:
    print(row)
print('记录更新完成')
cur.execute("Update book set rxrq=20170901")
cur.execute('select * from book')
values = cur.fetchall()
for row in values:
    print(row)
print("性别为女的名字和地址信息为： ")

cur.execute('select stuname,sex,address from book')
values = cur.fetchall()
for row in values:
    if row[1]=='女':
        print(row[0],row[2])
cur.close()
con.commit()
con.close()

```

////////////////////////////////////

2、创建 MySQL 数据库，并使用 Python 访问。

提示：需要安装 **pymysql**

创建一个数据库 Test，包含订单表(Order0)、订单明细表(Order1)、商品类别表(Category)三个表。

其中，Order0 中主要包含 OrderID、订单日期、客户名称、联系人、联系电话、收货地址。OrderID 为主键。

Order1 中主要包含 ID、OrderID、商品名称、类别 ID、规格、单位、数量、单价。ID 为主键，OrderID 为外键。类别 ID 也为外键。

Category 中主要包含类别 ID、类别名称。

使用 Python 实现增删改查功能。

```
////////////////////////////////////
import pymysql
conn = pymysql.connect(host='127.0.0.1',
                        user='root',
                        password='123456',
                        database='mysql',
                        charset='UTF8MB4')

cursor = conn.cursor()
def doSQL(sql):
    cursor.execute(sql)
    conn.commit()
# 删除数据库
doSQL('DROP DATABASE IF EXISTS onelinelearning;')

# 创建数据库
doSQL('CREATE DATABASE IF NOT EXISTS onelinelearning;')

# 删除数据表
doSQL('DROP TABLE IF EXISTS questions')

# 创建数据表
sql = '''
CREATE TABLE IF NOT EXISTS questions(
id INT auto_increment PRIMARY KEY,
wenti CHAR(200) NOT NULL UNIQUE,
daan CHAR(50) NOT NULL
) ENGINE=innodb DEFAULT CHARSET=UTF8MB4;
'''
```

```

doSQL(sql)
# 删除所有数据
doSQL('DELETE FROM questions;')
↵
# 插入数据
for i in range(10):
    sql = 'INSERT INTO questions(wenti,daan) VALUES("测试问题{0}","答案{0}");'.format(i)
    cursor.execute(sql)
conn.commit()
# 修改数据
doSQL('UPDATE questions set daan="被修改了" WHERE wenti="测试问题 6";')
# 删除指定的数据
doSQL('DELETE FROM questions WHERE daan="答案 8";')
↵
# 查询并输出数据
sql = 'SELECT * FROM questions'
cursor.execute(sql)
for row in cursor.fetchall():
    print(row)
# 关闭游标和连接
cursor.close()
conn.close()
.
////////////////////////////////////

```

3、创建 MongoDB 数据库，并使用 Python 访问。

从豆瓣电影（movie.douban.com）中爬取某部电影的评论，并存储到 MongoDB 数据库中，实现评论增删、改、查文档功能，具体功能自行定义。

```

import pymongo
import csv
import random
import ssl
import time
from urllib import request
from bs4 import BeautifulSoup
.

```

```

def getComment(url):
    # 火狐 谷歌 Safari 的头部
    header1 = {
        "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36",
        "Host": "movie.douban.com"
    }
    # ie 浏览器的请求头
    header2 = {
        "User-Agent": "Mozilla / 5.0 (Windows NT 10.0; WOW64; Trident / 7.0; rv: 11.0) like
Gecko",
        "Host": "movie.douban.com"
    }
    headlist = [header1, header2]
    index = random.randint(0, 1)
    # index 是一个随机数
    realHeader = headlist[index]
    req = request.Request(url, headers=realHeader)
    # 处理 https 协议的, http, https (更加安全协议) 可以解决 ssl 报错问题
    context = ssl._create_unverified_context()
    # 获取 网页的 文本对象
    html = request.urlopen(req, context=context)
    # 获取网页源码
    contents = html.read().decode("utf-8")
    return contents

```

```

def read_data(contents):
    soup = BeautifulSoup(contents, 'lxml')
    # 找到了大的容器, 里面包含很多 item 项
    comments_div = soup.select_one("#comments") # id 返回 Tag 类型
    # print(comments_div)
    # 获取大的容器中的每个小项, 返回一个结果集列表
    comment_list = comments_div.select('.comment-item') # 返回 ResultSet
    # print(comment_list)
    comment_dict_list = []
    # 遍历每个 item 项,
    for comment in comment_list:
        # find 与 find_all
        # attribute: 属性
        comment_div = comment.find('div', attrs={'class': 'comment'})
        # print(comment_div)
        comment_info = comment_div.find('span', attrs={'class': 'comment-info'})
        # 评价的用户名
        username = comment_info.find('a').text
        # 获取星级

```

```

        star = comment_info.find_all('span')[1].get('title')
        # 评论内容
        comment_text = comment_div.find('span', attrs={'class': 'short'}).text
        # 字典存放
        comment_dict = {'username': username, 'star': star, 'comment': comment_text}
        # 将字典存到列表中
        comment_dict_list.append(comment_dict)
        # [{}, {}, {}, {}, {}]
    # 返回字典
    return comment_dict_list

def get_collection():
    client=MongoClient('mongodb://localhost:27017/')
    db=client['douban']
    collection=db.movies
    return collection

# 将其保存到 csv 文件中
def save_comment_csv(comment_dict_list,collection):
    for comment_dict in comment_dict_list:
        collection.insert(comment_dict)

if __name__ == '__main__':
    collection = get_collection()
    for i in range(20):
        start = str(i * 20) # 20 40 60 80 100
        url = 'https://movie.douban.com/subject/34841067/comments?start=' + start +
        '&limit=20&sort=new_score&status=P'
        # 将 新的 url 以参数的形式赋值给 getComment()
        contents = getComment(url)
        # 解析爬取到的数据
        comment_dict_list = read_data(contents)
        # 保存到一个 csv 文件中
        save_comment_csv(comment_dict_list,collection)
        print('保存成功第' + str(i + 1) + '页内容')
        # 延时
        time.sleep(5)

```

4、Python 操作 Excel。

生成 50 个 Excel 文件，每个文件中包含 5 列数据，其中每个单元格内的内容随机生成，并且每个 Excel 文件的数据行数不相同。然后创建一个 SQLite 数据库，其结构与 Excel 文件相符合，最后把前面生成的 50 个 Excel 文件中的数据导入到这个数据库中。

提示：利用扩展库 openpyxl，标准库 string、os、os.path、sqlite3、time。

需要 sqlite 数据库：dataxlsx.db

Py 文件：test_xlsx2sqlite.py

////////////////////////////////////

```
from random import choice, randrange
from string import digits, ascii_letters
from os import listdir, mkdir
from os.path import isdir
import sqlite3
from time import time
from openpyxl import Workbook, load_workbook

def generateRandomData():
    """ 生成测试数据，共 50 个 Excel 文件，每个文件有 5 列随机字符串"""
    # 如果不存在子文件夹 xlsx，就创建
    if not isdir('xlsx'):
        mkdir('xlsx')

    # total 表示记录总条数
    global total

    # 候选字符集
    characters = digits+ascii_letters
```

```

# 生成 50 个 Excel 文件
for i in range(50):
    xlsName = 'xlsxs\\'+str(i)+'.xlsx'

    # 随机数, 每个xlsx文件的行数不一样
    totalLines = randrange(10**2)

    # 创建 Workbook, 获取第 1 个 Worksheet
    wb = Workbook()
    ws = wb.worksheets[0]

    # 写入表头
    ws.append(['a', 'b', 'c', 'd', 'e'])
    # 随机数据, 每行 5 个字段, 每个字段 30 个字符
    for j in range(totalLines):
        line = [" ".join((choice(characters) for ii in range(30))) for jj in range(5)]
        ws.append(line)

        total += 1

    # 保存xlsx文件
    wb.save(xlsName)

def eachXlsx(xlsxFn):
    """针对每个xlsx文件的生成器"""
    # 打开Excel文件, 获取第1个Worksheet
    wb = load_workbook(xlsxFn)
    ws = wb.worksheets[0]
    for index, row in enumerate(ws.rows):
        # 忽略表头
        if index == 0:
            continue
        yield tuple(map(lambda x: x.value, row))

```

```

↵
def xlsx2sqlite():↵
    """从批量 Excel 文件中导入数据到 SQLite 数据库"""↵
    # 获取所有 xlsx 文件名↵
    xlsxs = ('xlsxs\\'+fn for fn in listdir('xlsxs'))↵
    ↵
    # 连接数据库, 创建游标↵
    with sqlite3.connect('dataxlsx.db') as conn:↵
        cur = conn.cursor()↵
        for xlsx in xlsxs:↵

        # 批量导入, 减少提交事务的次数, 可以提高速度↵
        sql = 'INSERT INTO fromxlsx VALUES(?,?,?,?)'↵
        cur.executemany(sql, eachXlsx(xlsx))↵
        conn.commit()↵

    ↵
    # 用来记录生成的随机数据的总行数↵
    total = 0↵
    ↵
    # 生成随机数据↵
    generateRandomData()↵
    ↵
    # 导入数据, 并测试速度↵
    start = time()↵
    xlsx2sqlite()↵
    delta = time()-start↵
    ↵
    print('导入用时: ', delta)↵
    print('导入速度 (条/秒): ', total/delta)↵

    //////////////////////////////////////

```

实验 10、科学计算（numpy、scipy）

【实验类型】验证型，设计型

【实验学时】2

【实验目标】

- 1) 掌握扩展库 numpy、scipy 的安装。
- 2) 掌握使用扩展库 numpy 进行数组和矩阵运算方法。
- 3) 掌握使用扩展库 scipy 进行科学计算的基本操作。

4) 掌握使用扩展库 matplotlib 进行数据可视化的基本操作。

【实验内容】

(一) 运行两份 PPT (numpy、scipy) 中全部实例代码，阅读并读懂全部代码。

(二) 在线观看并实操算法工程师 flare_zhao 的《人工智能数学基础与 Python 实战》课程中的第一章、第二章、第三章

网址: <https://www.imooc.com/learn/1253>

实验 11、数据可视化 matplotlib (3 题)

【实验类型】设计型

【实验学时】2

【实验目标】

- 1) 掌握 matplotlib 模块的安装与使用。
- 2) 掌握曲线与散点图的绘制。
- 3) 掌握图形显示模式的设置方法。

【实验内容】

1、折线图。

某商品进价 49 元，售价 75 元，现在商场新品上架搞促销活动，顾客每多买一件就给优惠 1%，但是每人最多可以购买 30 件。对于商场而言，活动越火爆商品单价越低，但总收入和盈利越多。对于顾客来说，虽然买的越多单价越低，但是消费总金额却是越来越多的，并且购买太多也会因为用不完而导致过期不得不丢弃造成浪费。现在要求计算并使用折线图可视化顾客购买数量 num 与商家收益、顾客总消费以及顾客省钱情况的关系，并标记商场收益最大的批发数量和商场收益。

参考代码：

```
import matplotlib.pyplot as plt
import matplotlib.font_manager as fm

# 进价与零售价
basePrice, salePrice = 49, 75

# 计算购买 num 个商品时的单价，买的越多，单价越低
def compute(num):
    return salePrice * (1-0.01*num)
```

```

# numbers 用来存储顾客购买数量
# earns 用来存储商场的盈利情况
# totalConsumption 用来存储顾客消费总金额
# saves 用来存储顾客节省的总金额
numbers = list(range(1, 31))
earnings = []
totalConsumption = []
saves = []
# 根据顾客购买数量计算三组数据
for num in numbers:
    perPrice = compute(num)
    earnings.append(round(num*(perPrice-basePrice), 2))
    totalConsumption.append(round(num*perPrice, 2))
    saves.append(round(num*(salePrice-perPrice), 2))

# 绘制商家盈利和顾客节省的折线图，系统自动分配线条颜色
plt.plot(numbers, earnings, label='商家盈利')
plt.plot(numbers, totalConsumption, label='顾客总消费')
plt.plot(numbers, saves, label='顾客节省')

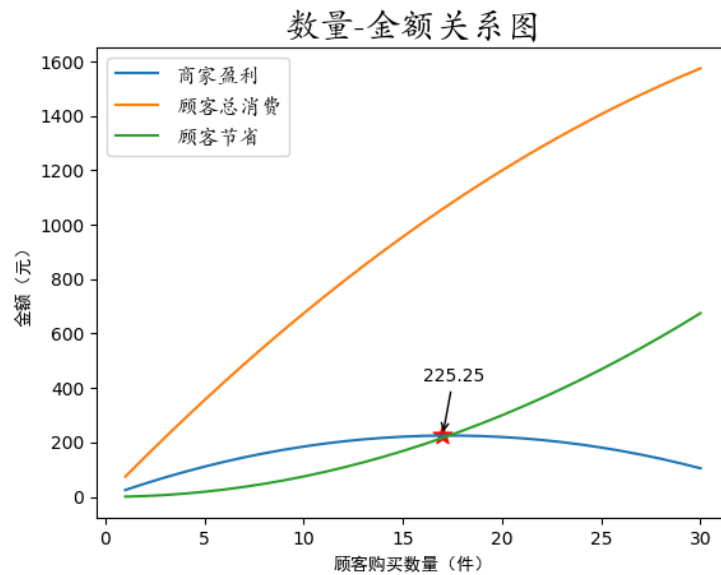
# 设置坐标轴标签文本
plt.xlabel('顾客购买数量（件）', fontproperties='simhei')
plt.ylabel('金额（元）', fontproperties='simhei')
# 设置图形标题
plt.title('数量-金额关系图', fontproperties='stkaity', fontsize=20)

# 创建字体，设置图例
myfont = fm.FontProperties(fname=r'C:\Windows\Fonts\STKAITI.ttf',
                           size=12)
plt.legend(prop=myfont)

# 计算并标记商家盈利最多的批发数量
maxEarn = max(earnings)
bestNumber = numbers[earnings.index(maxEarn)]
# 散点图，在相应位置绘制一个红色五角星
plt.scatter([bestNumber], [maxEarn], marker='*', color='red', s=120)
# 使用 annotate()函数在指定位置进行文本标注
plt.annotate(xy=(bestNumber, maxEarn), # 箭头终点坐标
             xytext=(bestNumber-1, maxEarn+200), # 箭头起点坐标
             s=str(maxEarn), # 显示的标注文本
             arrowprops=dict(arrowstyle="->")) # 箭头样式

# 显示图形
plt.show()

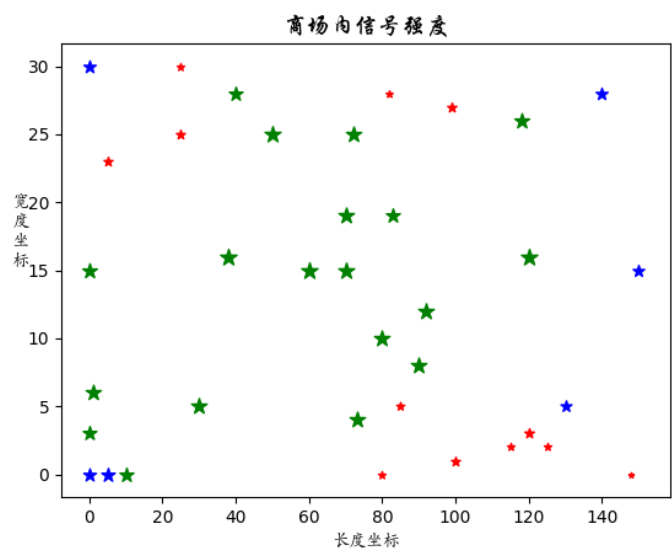
```



2、散点图

某商场开业三个月后，有顾客反应商场一楼部分位置的手机信号不好，个别收银台有时无法正常使用微信支付或支付宝，商场内也有些位置无法正常使用微信。为此，商场安排工作人员在不同位置对手机信号强度进行测试以便进一步提高服务质量和用户体验，测试数据保存于文件“D:\服务质量保证\商场一楼手机信号强度.txt”中，文件中每行使用逗号分隔的三个数字分别表示商场内一个位置的 x、y 坐标和信号强度，其中 x、y 坐标值以商场西南角为坐标原点且向东为 x 正轴（共 150 米）、向北为 y 正轴（共 30 米），信号强度以 0 表示无信号、100 表示最强。

编写程序，使用散点图对该商场一楼所有测量位置的手机信号强度进行可视化，既可以直观地发现不同位置信号的强度以便分析原因，也方便观察测试位置的分布是否合理。在散点图中，使用横轴表示 x 坐标位置、纵轴表示 y 坐标位置，使用五角星标记测量位置，五角星大小表示信号强度，五角星越大表示信号越强，反之表示信号越弱。同时，为了更好的可视化效果，信号强度高于或等于 70 的位置使用绿色五角星，低于 70 且高于或等于 40 的使用蓝色五角星，低于 40 的位置使用红色五角星。



参考代码：

```

import matplotlib.pyplot as plt
xs = []
ys = []
strengths = []
# 读取文件中的数据
with open('商场一楼手机信号强度.txt') as fp:
    for line in fp:
        x, y, strength = map(int, line.split(','))
        xs.append(x)
        ys.append(y)
        strengths.append(strength)

# 绘制散点图，s 指大小，c 指颜色，marker 指符号形状
for x, y, s in zip(xs, ys, strengths):
    if s < 40:
        color = 'r'
    elif s < 70:
        color = 'b'
    else:
        color = 'g'
    plt.scatter(x, y, s=s, c=color, marker='*')

plt.xlabel('长度坐标',
           fontproperties='stkaiti', # 设置中文字体
           fontsize=10)             # 设置字号
plt.ylabel('宽\n度\n坐\n标',
           fontproperties='stkaiti',
           fontsize=10,
           rotation='horizontal')   # 设置文字方向
plt.title('商场内信号强度',
          fontproperties='stxingkai',
          fontsize=14)

plt.show()

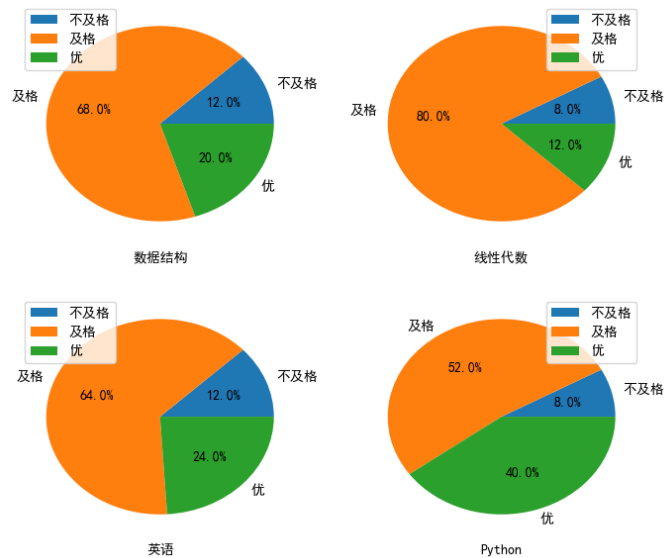
```

//////////商场一楼手机信号强度//////////文件//////////

```

0,0,60
0,3,70
5,0,68
10,0,73
0,30,56
0,15,80

```

参考代码:

```
from itertools import groupby
import matplotlib.pyplot as plt

# 设置图形中使用中文字体
plt.rcParams['font.sans-serif'] = ['simhei']

# 每门课程的成绩
scores = {'数据结构': [89, 70, 49, 87, 92, 84, 73, 71, 78, 81, 90, 37,
                        77, 82, 81, 79, 80, 82, 75, 90, 54, 80, 70, 68, 61],
          '线性代数': [70, 74, 80, 60, 50, 87, 68, 77, 95, 80, 79, 74,
                        69, 64, 82, 81, 78, 90, 78, 79, 72, 69, 45, 70, 70],
          '英语': [83, 87, 69, 55, 80, 89, 96, 81, 83, 90, 54, 70, 79,
                  66, 85, 82, 88, 76, 60, 80, 75, 83, 75, 70, 20],
          'Python': [90, 60, 82, 79, 88, 92, 85, 87, 89, 71, 45, 50,
                    80, 81, 87, 93, 80, 70, 68, 65, 85, 89, 80, 72, 75]}

# 自定义分组函数，在下面的 groupby() 函数中使用
def splitScore(score):
    if score >= 85:
        return '优'
    elif score >= 60:
        return '及格'
    else:
        return '不及格'
```

```

# 统计每门课程中优、及格、不及格的人数
# ratios 的格式为 {'课程名称': {'优':3, '及格':5, '不及格':1},...}
ratios = dict()
for subject, subjectScore in scores.items():
    ratios[subject] = {}
    # groupby()函数需要对原始分数进行排序才能正确分类
    for category, num in groupby(sorted(subjectScore), splitScore):
        ratios[subject][category] = len(tuple(num))
# 创建 4 个子图
fig, axs = plt.subplots(2,2)
axs.shape = 4,
# 依次在 4 个子图中绘制每门课程的饼状图
for index, subjectData in enumerate(ratios.items()):
    # 选择子图
    plt.sca(axs[index])
    subjectName, subjectRatio = subjectData
    plt.pie(list(subjectRatio.values()),          # 每个扇形对应的数值
            labels=list(subjectRatio.keys()),    # 每个扇形的标签
            autopct='%1.1f%%')                  # 百分比显示格式
    plt.xlabel(subjectName)
    plt.legend()
    plt.gca().set_aspect('equal')               # 设置纵横比相等
plt.show()

```

实验 12、数据分析 pandas（2 题）

【实验类型】设计型

【实验学时】2

【实验目标】

- 1) 掌握扩展库 pandas、matplotlib 的安装。
- 2) 掌握使用扩展库 pandas 进行数据分析的基本操作。
- 3) 掌握使用扩展库 matplotlib 进行数据可视化的基本操作。

【实验内容】

1、运行下面的程序，在当前文件夹中生成饭店营业额模拟数据文件 **data.csv**。

```
import csv
import random
import datetime

with open('data.csv', 'w', newline='') as fp:
    # 创建 csv 文件写入对象
    wr = csv.writer(fp)
    wr.writerow(['日期', '销量']) # 写入表头
    # 生成模拟数据
    startDate = datetime.date(2020, 1, 1)
    # 生成 365 个模拟数据，可以根据需要进行调整
    for i in range(365):
        # 生成一个模拟数据，写入 csv 文件
        amount = random.randint(0, 1000)
        if len(str(startDate)) > 0:
            wr.writerow([str(startDate), amount])
        # 下一天
        startDate = startDate + datetime.timedelta(days=1)
```

```

↵
#coding:utf-8↵
import pandas as pd↵
import matplotlib.pyplot as plt↵
import numpy as np↵
plt.rcParams['font.sans-serif']=['SimHei'] #用来正常显示中文标签↵
plt.rcParams['axes.unicode_minus']=False #用来正常显示负号↵
# 读取数据，丢弃缺失值↵
df = pd.read_csv('data.csv',encoding='cp936') ↵
df = df.dropna() #删除空值↵
#按月汇总↵
df1 = df[:]↵
df1['月'] = df1['日期'].map(lambda x: x[5:7])↵
df1 = df1.groupby('月',as_index=False).agg(np.sum)↵
print(df1)↵
↵

# 生成营业额折线图 ↵
plt.figure(figsize=(8,4)) #创建大小为 800*400 绘图对象↵
plt.plot(df1['月'],df1['销量'],'b--',linewidth=1.0,label='营业额')↵
plt.xlabel('月份') #设置 X 轴的文字↵
plt.ylabel('营业额（元）') #设置 Y 轴的文字↵
plt.title('按月份汇总营业额折线图') #设置图表的标题↵
plt.legend() #显示图例↵
plt.grid(True)↵
plt.savefig('first.jpg')↵
plt.show()↵
↵

# 按月统计，生成柱状图↵
plt.figure(figsize=(8,4))↵
plt.bar(df1['月'],df1['销量']) ↵
plt.savefig('second.jpg')↵
plt.show()↵
↵

# 查找涨幅最大的月份，写入文件↵
df2 = df1.drop('月', axis=1).diff() ↵
m = df2['销量'].nlargest(1).keys()[0] ↵
with open('maxMonth.txt', 'w') as fp:↵
    fp.write(df1.loc[m, '月'])↵
↵

# 按季度统计，生成饼状图↵
plt.figure(figsize=(8,4))↵

```

```

one = df1[:3]['销量'].sum()
two = df1[3:6]['销量'].sum()
three = df1[6:9]['销量'].sum()
four = df1[9:12]['销量'].sum()
print([one, two, three, four])
labels = ["一季度", "二季度", "三季度", "四季度"]
facts = [one, two, three, four]
plt.pie(x = facts, labels = labels, autopct = "%.0f%%", shadow= True)
plt.savefig('third.jpg')
plt.show()

```

//

实验 13、机器学习（4 题）

【实验类型】设计型

【实验学时】2

【实验目标】

- 1) 掌握扩展库 sklearn、iris 安装和使用。
- 2) 查阅文献，掌握常用有监督和无监督机器学习算法的原理。
- 3) 掌握课堂上介绍的线性回归、KNN、KMean 算法，以及本实验中的朴素贝叶斯算法、决策树算法。
- 4) 理解各算法中各参数的含义以及对结果的影响。

【实验内容】

1、朴素贝叶斯算法

- 在线观看并实操算法工程师 flare_zhao 的《人工智能数学基础与 Python 实战》课程中的“第 4 章 概率分析”，实践“4-4 朴素贝叶斯判断客户消费意愿”。
- 网址：<https://www.imoooc.com/learn/1253>

2、线性回归

#柳毅线性回归视频

结合 PPT 中的线性回归实例，阅读并理解下列代码。

//

```

import matplotlib.pyplot as plt

import numpy as np

from sklearn import datasets, linear_model

from sklearn.metrics import mean_squared_error

#糖尿病人数据

diabetes = datasets.load_diabetes()

#查看 diabetes 结构

print(diabetes.feature_names)

diabetes_X = diabetes.data[:, np.newaxis, 2]

#diabetes_X = diabetes_X.reshape(len(diabetes_X), 1)

diabetes_X_train = diabetes_X[:-20]

diabetes_X_test = diabetes_X[-20:]

diabetes_y_train = diabetes.target[:-20]

diabetes_y_test = diabetes.target[-20:]

#创建一个线性回归对象

regr = linear_model.LinearRegression()

#用训练数据训练模型

regr.fit(diabetes_X_train, diabetes_y_train)

#用训练好的模型对测试集上的数据进行预测

diabetes_y_pred = regr.predict(diabetes_X_test)

print('Coefficients: \n', regr.coef_)

print("Mean squared error: %.2f"

      % mean_squared_error(diabetes_y_test, diabetes_y_pred))

#可视化展示

plt.scatter(diabetes_X_test, diabetes_y_pred, color='red')

plt.scatter(diabetes_X_test, diabetes_y_test, color='black')

plt.plot(diabetes_X_test, diabetes_y_pred, color='blue', linewidth=3)

plt.show()

////////////////////////////////////

```

3、决策树算法

- 数据工程师途索的《Python 数据分析-基础技术篇》中“6-3 scikit-learn 实现决策树”
- 网址: <https://www.imoooc.com/learn/843>

查阅资料, 根据掌握的决策树知识, 阅读并理解下列代码。

```
////////////////////////////////////  
#step1 数据预处理  
from sklearn.datasets import load_iris  
from sklearn.model_selection import train_test_split  
data = load_iris()  
train_data, test_data, train_target, test_target = \  
train_test_split(data.data, data.target, test_size=0.2, random_state=1)  
iris = load_iris()  
#step2 建模  
from sklearn import tree  
clf=tree.DecisionTreeClassifier(criterion='entropy') #使用分类器, 采用熵增益  
clf.fit(train_data,train_target)  
#step3 预测  
y_pred=clf.predict(test_data)  
#step4 验证  
from sklearn import metrics  
print(metrics.accuracy_score(y_true=test_target,y_pred=y_pred))  
print(metrics.confusion_matrix(y_true=test_target,y_pred=y_pred))  
#以图形方式输出结果  
import matplotlib.pyplot as plt  
plt.figure(dpi=300)  
tree.plot_tree(clf, filled=True)  
plt.show()  
#以文本方式输出结果  
r = tree.export_text(clf, feature_names=iris['feature_names'])  
print(r)  
#以文件方式输出结果  
with open('tree.dot','w') as fw:  
    tree.export_graphviz(clf,out_file=fw)  
////////////////////////////////////
```

4、k-means 算法

在课堂上介绍的 k-means 聚类实例基础上, 编写程序, 使用 k-means 聚类方法对已知数据进行聚类, 然后对未知样本寻找相似元素。

```
////////////////////////////////////
```



```

from numpy import array
from sklearn.cluster import KMeans

#训练数据
X = array([[1,1,1,1,1,1,1],
           [2,3,2,2,2,2,2],
           [3,2,3,3,3,3,3],
           [1,2,1,2,2,1,2],
           [2,1,3,3,3,2,1],
           [6,2,30,3,33,2,71]])

#建模
kmeansPredicter = KMeans(n_clusters=3).fit(X)

#训练数据分类
category = kmeansPredicter.predict(X)
print('训练数据分类情况：', category)

#测试
print('1'+('='*30))
X_test=[[1,2,3,3,1,3,1]]
result = kmeansPredicter.predict(X_test)
print('预测结果：', result)
print('相似元素：\n', X[category==result])
print('2'+('='*30))
X_test=[[5,2,23,2,21,5,51]]
result = kmeansPredicter.predict(X_test)
print('预测结果：', result)
print('相似元素：\n', X[category==result])

////////////////////////////////////

```