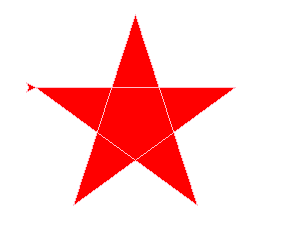
**实验七 模块、包和库**

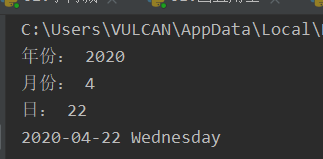
1. 使用turtle模块绘制一个五角星。

import turtle  
turtle.color("pink", "red")  
turtle.begin\_fill()  
  
# 绘画的图形  
for \_ in range(5):  
 turtle.forward(200)  
 turtle.right(144)  
turtle.end\_fill()  
input()



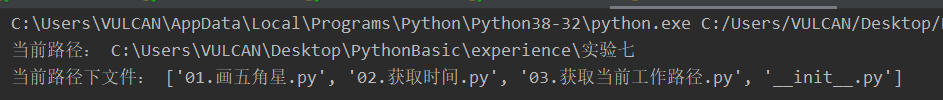
1. 使用datetime 模块获取当前时间，并指出当前时间的年、月、日、周数，以及当天是该周的第几天。

import datetime,time  
# 获取当前时间  
now = datetime.datetime.now()  
# 格式化时间字符串  
str\_time = now.strftime("%Y-%m-%d %A")  
year=now.year  
month=now.month  
day=now.day  
print("年份：",year)  
print("月份：",month)  
print("日：",day)  
print(str\_time)



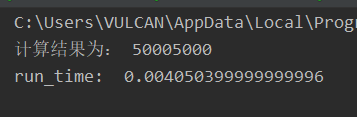
1. 使用OS模块获取当前项目的工作路径，并输出当前路径下的所有文件

import os, os.path  
  
a = os.getcwd()  
b = [f for f in os.listdir(a )]  
  
print("当前路径：",a)  
print("当前路径下文件：",b)



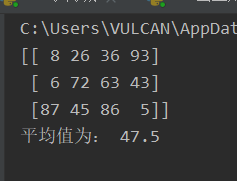
1. 编写计算 的函数，使用Timeit模块计算该函数的执行时间

import timeit  
import time  
  
def calculate():  
 sum=0  
 for i in range(1,10001):  
 sum+=i  
 print("计算结果为：",sum)  
  
if \_\_name\_\_=="\_\_main\_\_":  
 # calculate()  
 run\_time = timeit.timeit(stmt=calculate, number=1)  
 print('run\_time: ', run\_time)



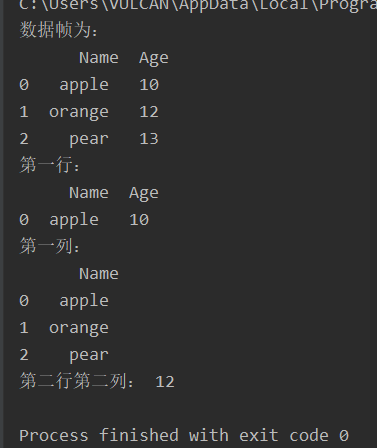
1. 使用RANDOM 模块和NumPy 库生成一个3行4列的多维数组，数组中每一个元素为1—100之间的随机整数，然后求数组所有元素的平均值

import numpy as np  
  
data = np.random.randint(1,100,size=[3,4]) #创建一个2行3列的随机数组  
print(data)  
average=0  
average\_all=0  
sum=0  
for i in data:  
 sum+=i  
average=sum/4  
sum=0  
for i in average:  
 sum+=i  
average\_all=sum/3  
print("平均值为：",average\_all)



1. 使用Pandas库创建一个数据帧（DataFrame），然后输出某行、某列和某个单元格的数据

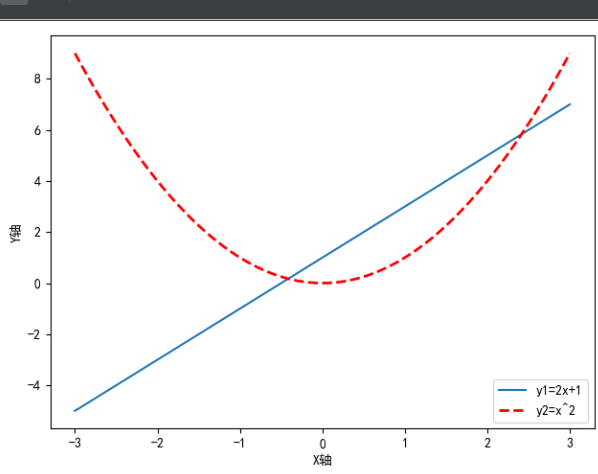
import numpy as np  
import pandas as pd  
  
data = [['apple',10],['orange',12],['pear',13]]  
df = pd.DataFrame(data,columns=['Name','Age'])  
print ("数据帧为：\n",df)  
num1 = df.iloc[[0]]  
print("第一行：\n",num1)  
num2=df.loc[:,['Name']]  
print("第一列：\n",num2)  
detail=df.iat[1,1] #选取第二行第二列，用于已知行、列位置的选取。  
print("第二行第二列：",detail)



1. 使用Matplotlib 绘制 的图形，并设置坐标轴的名称和图例

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  
  
x = np.linspace(-3,3,50)  
y1 = 2\*x+1  
y2 = x\*\*2  
  
#绘制在同一个figure中  
plt.figure()  
plt.xlabel('X轴')  
plt.ylabel('Y轴')  
l1, = plt.plot(x,y1,label='y1=2\*x+1')#放在handles中l1一定要加，  
l2, = plt.plot(x,y2,color='red',linewidth = 2.0,linestyle = '--',label='y2=x^2')  
  
plt.legend(handles=[l1, l2,],labels=['y1=2x+1','y2=x^2'],loc='best')  
plt.show()

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  
  
x = np.linspace(-3,3,50)  
y1 = 2\*x+1  
y2 = x\*\*2  
  
#绘制在同一个figure中  
plt.figure()  
plt.xlabel('X轴')  
plt.ylabel('Y轴')  
l1, = plt.plot(x,y1,label='y1=2\*x+1')#放在handles中l1一定要加，  
l2, = plt.plot(x,y2,color='red',linewidth = 2.0,linestyle = '--',label='y2=x^2')  
  
plt.legend(handles=[l1, l2,],labels=['y1=2x+1','y2=x^2'],loc='best')  
plt.show()



1. 编写程序，实现对一篇中文文章进行分词和统计，结果使用词云图展示

**import** re *# 正则表达式库***import** collections *# 词频统计库***import** numpy **as** np *# numpy数据处理库***import** jieba *# 结巴分词***import** wordcloud *# 词云展示库***from** PIL **import** Image *# 图像处理库***import** matplotlib.pyplot **as** plt *# 图像展示库  
  
# 读取文件*fn = open(**'file.txt'**) *# 打开文件*string\_data = fn.read() *# 读出整个文件*fn.close() *# 关闭文件  
  
# 文本预处理*pattern = re.compile(**u'\t|\n|\.|-|:|;|\)|\(|\?|"'**) *# 定义正则表达式匹配模式*string\_data = re.sub(pattern, **''**, string\_data) *# 将符合模式的字符去除  
  
# 文本分词*seg\_list\_exact = jieba.cut(string\_data, cut\_all = **False**) *# 精确模式分词*object\_list = []  
remove\_words = [**u'的'**, **u'，'**,**u'和'**, **u'是'**, **u'随着'**, **u'对于'**, **u'对'**,**u'等'**,**u'能'**,**u'都'**,**u'。'**,**u' '**,**u'、'**,**u'中'**,**u'在'**,**u'了'**,  
 **u'通常'**,**u'如果'**,**u'我们'**,**u'需要'**,**u'《'**,**u'》'**] *# 自定义去除词库***for** word **in** seg\_list\_exact: *# 循环读出每个分词* **if** word **not in** remove\_words: *# 如果不在去除词库中* object\_list.append(word) *# 分词追加到列表  
  
# 词频统计*word\_counts = collections.Counter(object\_list) *# 对分词做词频统计*word\_counts\_top10 = word\_counts.most\_common(10) *# 获取前10最高频的词*print (word\_counts\_top10) *# 输出检查  
  
# 词频展示*mask = np.array(Image.open(**'zyx.jpg'**)) *# 定义词频背景*wc = wordcloud.WordCloud(  
 font\_path=**'simhei.ttf'**, *# 设置字体格式* mask=mask, *# 设置背景图* max\_words=200, *# 最多显示词数* max\_font\_size=100 *# 字体最大值*)  
  
wc.generate\_from\_frequencies(word\_counts) *# 从字典生成词云*image\_colors = wordcloud.ImageColorGenerator(mask) *# 从背景图建立颜色方案*wc.recolor(color\_func=image\_colors) *# 将词云颜色设置为背景图方案*plt.imshow(wc) *# 显示词云*plt.axis(**'off'**) *# 关闭坐标轴*plt.show() *# 显示图像*



1. 自定义一个模块，然后在其他源文件中进行调用、测试

*#自定义的模板***def** add(a, b):  
 **return** a + b

*# 调用同一目录下的calculator.py的add()方法***from** calculator **import** add  
print(add(4, 5))

