

鄂尔多斯应用技术学院实验报告

ORDOS INSTITUTE OF TECHNOLOGY

Experiment Report

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验科目(Subject) | <Python 语言基础> | | 系 别(Dept.) | 数学与计算机工程系 | |
| 年 级(Grade) | 2020级计算机科学与技术 | | 专 业(Major) | 计算机科学与技术 | |
| 姓 名(Name) | 陈腾飞 | 学 号(NO.) | 20207110036 | 指导教师(Teacher) |  |

鄂尔多斯应用技术学院

2022年09月

**实验分数：**

|  |  |  |
| --- | --- | --- |
| 实验项目 | 占比 | 得分 |
| 实验一：基本数据类型的使用 | 15% |  |
| 实验二：程序控制结构 | 15% |  |
| 实验三：函数及代码复用 | 15% |  |
| 实验四：组合数据类型的应用 | 20% |  |
| 实验五：数据分析与可视化综合实验‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬ | 15% |  |
| 实验六：直辖市近十年GDP总值并可视化排名 | **20%** |  |
| 总分 | **100%** |  |

**书写规范要求：**

1．实验目的：清晰的说明本实验的目的和主要实验方法。

2．实验内容与步骤：较详细地描述实验过程，并对数据、实验结果进行分析。

3．实验心得：完成实验后自己的一些想法和体会，以及自己对实验方法的理解。

**评分标准：**

（一）A（85分以上）

　　1．报告中对实验过程叙述详细、概念正确，语言表达准确，结构严谨，条理清楚，逻辑性强，自己努力完成，没有抄袭。

　　2．对实验过程中存在问题分析详细透彻、规范、全面；

　　3．实验心得体会深刻、有创意，论述合理详细，有自己的个人见解和想法，能结合实例论述，提出问题并给出解决方法。

（二）B（75-85分）

　　1．报告中对实验过程叙述较详细、概念正确，语言表达准确，结构严谨，条理清楚，逻辑性强，自己努力完成，没有抄袭。

　　2．对实验过程中存在问题分析详细透彻、规范、全面；能结合实例内容描述正确。

　　3．实验心得体会深刻、有创意，论述合理详细，有自己的个人见解和想法。

（三）C（60-75分）

　　1．报告中对实验过程叙述较详细，自己努力完成，没有抄袭。

　　2．对实验过程中存在问题有简单分析和描述，不全面。

　　3．实验心得体会不够深刻，缺乏创意。

（四）D（60分以下，或具备下面一项者为不及格）

　　1．没有交报告。

　　2．基本上是抄袭。

3．内容太空泛，太简单。

**实验一基本数据类型的使用（课堂实践）**

**实验目的：**

1、掌握Python的基本数据类型的概念和使用。

2、运用Python的标准数学库进行数值计算。

3、掌握字符串类型的格式化操作方法和应用。

**实验内容：**

**1、天天向上的力量**

这是"实例"题，与课上讲解实例相同，请作答检验学习效果。

-A君：一年365天，每天进步1%，不停歇

-B君：一年365天，每周工作5天休息2天，休息日下降1% ，要多努力呢？

(1)

(2)

import math

factor = 0.01

def dayup(factor):

    basic = 1.0

    for i in range(365):

        if(i % 7 in [6,0] ):

            basic \*= 1 - factor

        else:

            basic \*= 1 + factor

    return basic

while (dayup(factor) < math.pow(1.01,365)):

        factor += 0.01

print(factor)



**2、这是一个关于"字符串之基本操作"的小练习，覆盖字符串索引、切片和基本操作方法，包含两部分内容：跟随练习和小测验**。

(1)在"跟随练习"环节，请使用IDLE编辑器的交互模式，逐一输入以下语句，观察并比较输出结果。

(2)在"小测验"环节，请在本题目对应的"提交代码"页面按要求输入代码，"保存并提交"后查看结果。请使用IDLE编辑器的交互模式（含有>>>提示符），按顺序逐一输入以下语句（共20个）

>>>a = "PYPY"

>>>b = '语言'

>>>a + b

'PYPY语言'

>>>print(a, b)

PYPY 语言

>>>a += b

>>>a

'PYPY语言'

>>>a[4:0]

''

>>>a[:-1:2]

'PP语'

>>>a[::-1]

'言语YPYP'

>>>a[::-2]

'言YY'

>>>a[::]

'PYPY语言'

>>>"Py" in a

False

>>>"Py" not in a

True

>>>b in a

True

>>>a\*3

'PYPY语言PYPY语言PYPY语言'

>>>a\*0

''

>>>a\*-2

''

>>>c = "\'\"\\{\}\\*"

'\'"\\{\\}\\\*'

>>>print(c)

'"\{\}\\*

>>>print(c+"\t\\")

'"\{\}\\* \

小测验：

请将如下代码粘贴至"提交代码"页面，"保存并提交"后查看结果，请解释其中多个空行的缘由。注意，此时代码不是交互模式，而是文件模式。

a = "Python基础语法\n精讲"

print(a[:-2])

print(a[-3::]\*2)



**3、地球数据计算**

地球的半径是6371km，假设地球是一个规则的球体，完成以下计算并依次分行输出结果，结果保留2位小数，输出格式请参考输入输出示例。π值引用math库中常数pi的值。

（1）计算地球表面积（表面积公式S = 4πR2）

（2）计算地球体积（体积公式是V = 4πR3/3）

（3）计算地球赤道的周长（圆周长公式是L = 2πR）

（4）假设有一根绳子正好可以紧贴地球绕赤道一周，紧密的捆绑住地球。现在将绳子延长1米，仍围绕地球赤道一周，假设绳子与地球表面间空隙均匀，计算绳子与地球之间的空隙大小。判断一只正常大小的老鼠能否从这个空隙穿过？

提示：

**输出示例**：

地球表面积为XXXX.XX平方米

地球体积为XXXX.XX立方米

地球赤道周长为XXXX.XX米

空隙大小为XXXX.XX米

老鼠可以从空隙中钻过/老鼠无法通过空隙

可假设老鼠身体截面为圆柱，且圆柱最粗处直径为10cm，若空隙大小10cm，输出：“老鼠可以从空隙中钻过”，否则输出：“老鼠无法通过空隙”。根据判定结果进行输出时，可以用if条件表达式进行判定，满足条件时的输出放在if分支下的缩进语句块中；不满足条件的输出放在与if对齐的else分支下的缩进语句块里。

编程模板如下：

# -------- ------- --------

# @File : 地球数据计算.py

# @Author :

# @Contact: vasp@qq.com

# @Company: 鄂尔多斯应用技术学院

# @Version: 1.0

# @Modify : 2022/10/30 23:23

# Python程序设计基础，高等教育出版社

# -------- ------- --------

import math

radius = 6371 \* 1000

# 1. 计算地球表面积（表面积公式S = 4π(R\*\*2)

#=======================================================

# 补充你的代码

#####

print(f'地球表面积为{surface\_area}平方米')

# 2. 计算地球体积（体积公式是V = 4π(R\*\*3)/3）

#####

# 3. 计算地球赤道的周长（圆周长公式是L = 2πR）

#####

# 4.计算绳子与地球之间的空隙大小，结果保留2位小数

####

# 5.判断老鼠是否可以从空隙中钻过

if #####

#####

else:

print('老鼠无法通过空隙')

#=======================================================

1.程序代码：

import math

radius = 6371 \* 1000

Pi = math.pi

#表面积

def area(r):

    return 4 \* Pi \* r \* r

#体积

def volume(r):

    return 4 \* Pi \* pow(r,3) / 3

#周长

def perimeter(r):

    return 2 \* Pi \* r

def mouse(r):

    if (r - (perimeter(r) + 1) / 2 / Pi) > 0.1:

        res = "老鼠能穿过"

    else:

        res = "老鼠不能穿过"

    return  res

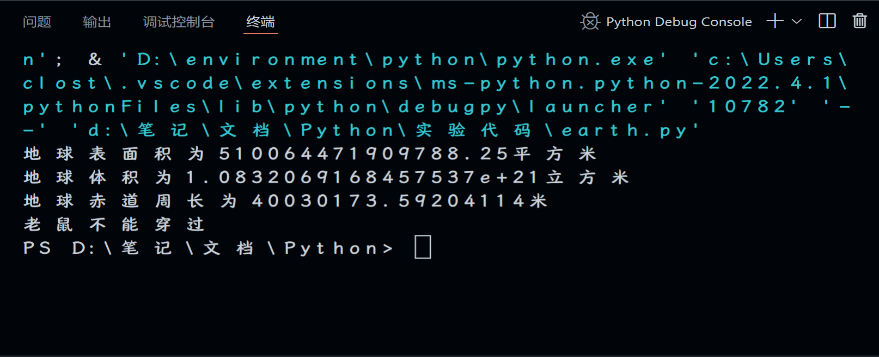
print("地球表面积为{}平方米".format(area(radius)))

print("地球体积为{}立方米".format(volume(radius)))

print("地球赤道周长为{}米".format(perimeter(radius)))

print(mouse(radius))

2.程序运行截图：



|  |  |
| --- | --- |
| 成绩： | 评阅人： |

成绩：

**实验二 程序控制结构**

**实验目的：**

1、熟练运用选择结构与循环结构解决实际问题。

2、注意选择结构嵌套时代码的缩进与对齐。

3、理解带else 子句的循环结构执行流程。

4、理解条件表达式value1 if condition else value2 的用法。

5、理解使用异常处理结构约束用户输入的用法。

6、理解带else 子句的异常处理结构的执行流程。

**实验内容：**

1、你刚刚大学毕业，在北京找到了一份很好的工作，决定开始存钱买房。由于北京的房价很高，你要攒几年钱才能付得起房子的首付。

现根据以下假定来计算你需要多长时间才能攒够首付款：

(2)将首付款占总房价的比例称为portion\_down\_payment。为简单起见，假设portion\_down\_payment = 0.30（30%）。

(3)将存款金额称为current\_savings。你的存款从0元开始。

(4)假设你的年薪是annual\_salary，按12个月平均发放，单位是元。

(5)假设你每个月都要拿出一定百分比的工资来存首付。称为portion\_saved，此值为一个表示百分比的整数，例如50表示50%。

写程序来计算你需要多少个月才能攒够钱付定金，不足一个月按一个月计算。

要求使用下面的输入输出代码，填充完整并通过测试用例。

#=======================================================

total\_cost = float(input()) # total\_cost为当前房价

annual\_salary = float(input()) # 年薪

portion\_saved = float(input()) /100

# 月存款比例，输入30转为0.30（即30%）

# 根据首付款比例计算首付款down\_payment，根据月存款比例计算月存款额monthly\_deposit

#========================

# 补充你的代码

#========================

print('首付',down\_payment)

print('月存款', monthly\_deposit)

# 计算多少个月才能存够首付款，结果为整数，不足1月按1个月计算，即向上取整

#===========================

# 补充你的代码

#===========================

print(f'需要{number\_of\_months}个月可以存够首付')

#=======================================================

输入输出格式

输入为三行，每行一个浮点数，分别代表房价、年薪、月存工资的百分数。

|  |  |
| --- | --- |
| 示例1：  输入：  2500000  120000  70  输出：  首付 750000.00 元  月存款 7000.00 元  需要108个月可以存够首付 | 示例2：  输入：  2500000  100000  80  输出：  首付 750000.00 元  月存款 6666.67 元  需要113个月可以存够首付: |

1.程序代码：

import math

#房子总价，首付占比，当前存款，年薪，月供占工资百分比

total\_cost = float(input("请输入房子的总价"))

portion\_down\_payment = 0.3

current\_savings = 0

annual\_salary = float(input("请输入你的年薪"))

portion\_saved = float(input("请输入你每月拿出工资的百分比")) /100

print('首付',round(total\_cost \* portion\_down\_payment,2))

print('月存款',round(annual\_salary / 12 \* portion\_saved,2))

number\_of\_months = math.ceil(total\_cost \* portion\_down\_payment / (annual\_salary / 12 \* portion\_saved))

print(f'需要{number\_of\_months}个月可以存够首付')

2.程序运行截图：



‬2、猜数游戏

编写程序模拟猜数游戏。程序运行时，系统生成一个随机数，然后提示用户进行猜测，并根据用户输入进行必要的提示（猜对了、太大了、太小了），如果猜对则提前结束程序，如果次数用完仍没有猜对，提示游戏结束并给出正确答案。

**知识拓展：**

1、猜数游戏续。改编程序练习题4.1，让计算机能够随机产生一个预设数字，范围在0～100之间，其他游戏规则不变。

2、猜数游戏续。对于程序练习题4.4程序，当用户输入的不是整数（如字母、浮点数等）时，程序会终止执行退出。改编该程序，当用户输入出错时给出“输入内容必须为整数！”的提示，并让用户重新输入。

1.程序代码：

from random import \*

def guess():

guessNum = randint(0,100)

#10次猜的机会

guessCounter = 0

while True:

if guessCounter >= 3:

print("很遗憾，次数用尽,正确答案是{}".format(guessNum))

break

guessCounter += 1

try:

inputNum = int(input("请输入你要猜的数"))

except :

print("ERROR!!!!输入的不是整数")

return

if inputNum > guessNum:

print("猜大了")

elif inputNum < guessNum:

print("猜小了")

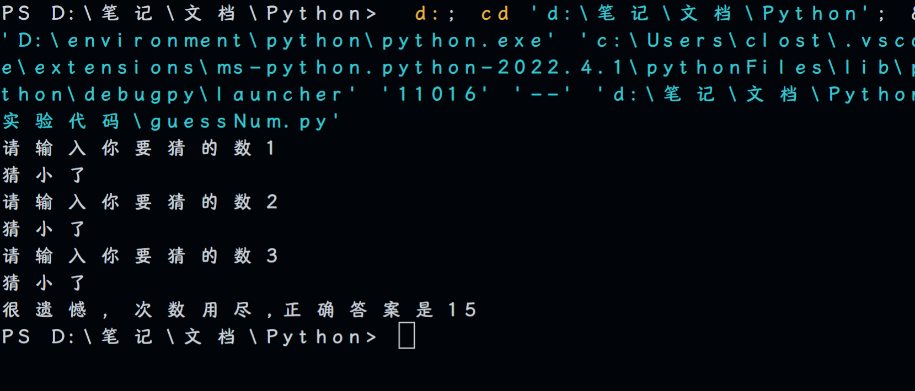
else:

print("恭喜你经过{}次猜对了".format(guessCounter))

break

guess()

2.程序运行截图：



|  |  |
| --- | --- |
| 成绩： | 评阅人： |

成绩：

**实验三 函数及代码复用**

**实验目的：**

1、掌握函数的定义和调用方法。

2、理解函数的参数传递过程以及变量的作用范围。

3、掌握lambda函数。

4、理解函数递归的定义和使用方法。

**实验内容及步骤：**

1、递归法绘制科赫曲线。请根据课本中内容，将“科赫曲线绘制”程序逐行输入其中，按自己的思路进行修改并运行效果，体会递归思想。



1.程序代码：

import turtle

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)

def main():

turtle.setup(800,400)

turtle.speed(0)

turtle.penup()

turtle.goto(-300,-50)

turtle.pendown()

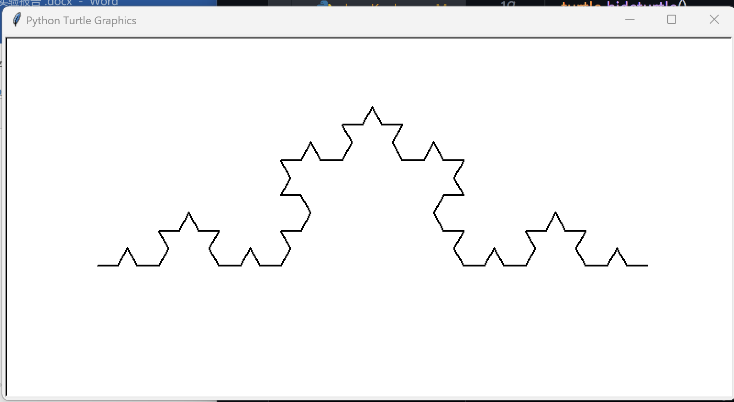
turtle.pensize(2)

koch(600,3)

turtle.hideturtle()

main()

2.程序运行截图：



2.汉诺塔问题。

汉诺塔问题。法国数学家爱德华·卢卡斯曾编写过一个印度的古老传说：在世界中心贝拿勒斯（在印度北部）的圣庙里，一块黄铜板上插着三根宝石针。印度教的主神梵天在创造世界的时候，在其中一根针上从下到上地穿好了由大到小的64片金片，这就是所谓的汉诺塔。不论白天黑夜，总有一个僧侣在按照下面的法则移动这些金片：一次只移动一片，不管在哪根针上，小片必须在大片上面。僧侣们预言，当所有的金片都从梵天穿好的那根针上移到另外一根针上时，世界就将在一声霹雳中消灭，而梵塔、庙宇和众生也都将同归于尽……

分析：我们把n个金片从A柱移动至C柱的问题可以表示为：Hanio(n,A,C,B);

该问题可以分解成以下子问题：

第一步：将n-1个金片从A柱移动至B柱（借助C柱为过渡柱）

第二步：将A柱底下最大的金片移动至C柱

第三步：将B柱的n-1个金片移至C柱（借助A柱为过渡柱）

利用递归思想，解决问题汉诺塔问题。

1.程序代码：

def hanoi(n,a,b,c):

#n = 1时候直接a盘到c盘退出程序

if n == 1:

print("{}-->{}".format(a,c))

return

#把上面n-1个看做一个整体借助b盘。把n-1个从a盘移动到c盘

hanoi(n-1,a,c,b)

#然后把最后一个从a盘移动到c盘

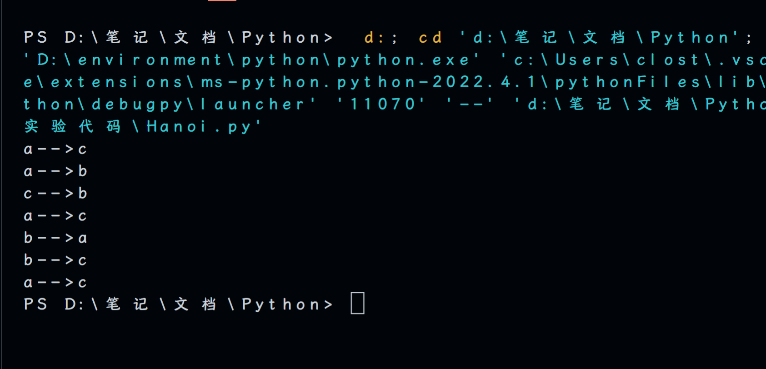
print("{}-->{}".format(a,c))

#然后把b盘上剩余中的n-1借助c盘从b盘移动到a盘

hanoi(n-1,b,a,c)

hanoi(3,"a","b","c")

2.程序运行截图：



|  |  |
| --- | --- |
| 成绩： | 评阅人： |

成绩：

**实验四：组合数据类型的应用**

**实验目的：**

1、了解3类基本组合数据类型。

2、理解列表概念并掌握Python中列表的使用。

3、理解字典概念并掌握Python中字典的使用。

4、运用jieba库进行中文分词并进行文本词频统计。

**实验内容：**

**1、基本统计值计算**

这是"实例"题，与课上讲解实例相同，请作答检验学习效果。

输入示例：

‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬1.程序代码：‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬

from math import sqrt

def getNum():

nums = []

iNumStr = input("请输入数字(直接输入回车退出):")

while iNumStr != "":

nums.append(eval(iNumStr))

iNumStr = input("")

return nums

# 平均值

def mean(numbers):

s = 0.0

for num in numbers:

s += num

return s / len(numbers)

# 标准差

def dev(numbers, mean):

sdev = 0.0

for num in numbers:

sdev += (num - mean) \*\* 2

return sqrt(sdev / (len(numbers) - 1))

#中位数

def median(numbers):

new = sorted(numbers)

size = len(numbers)

if size % 2 == 0:

med = (new[size//2-1] + new[size//2]) /2

else:

med = new[size//2]

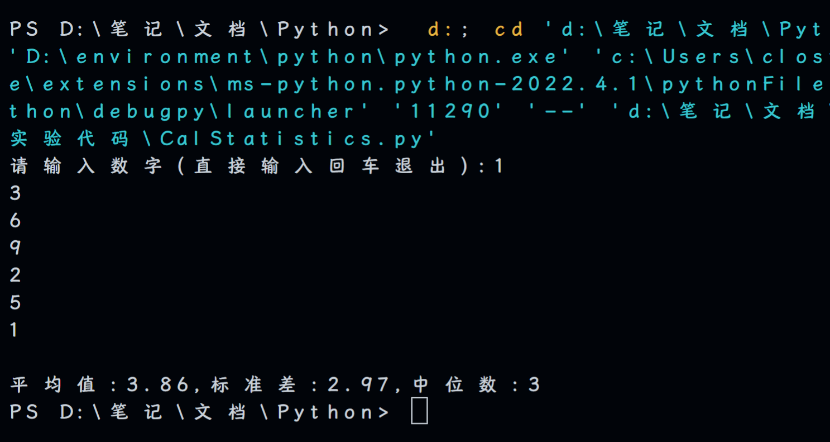
return med

n = getNum()

m = mean(n)

print("平均值:{:.3},标准差:{:.3},中位数:{}".format(m,dev(n,m),median(n)))

2.程序运行截图：



**2、文本词频统计 -- Hamlet**

这是"实例"题，与课上讲解实例相同，请作答检验学习效果。

(1) 单词不区分大小写，即单词的大小写或组合形式一样；

‬(2) 请在文本中剔除如下特殊符号：!"#$%&()\*+,-./:;<=>?@[\]^\_ {|}~‬

(3) 输出10个单词，每个单词一行；

(4) 输出单词为小写形式。

1.程序代码：

from cgitb import text

from itertools import count

def getText():

txt = open("file\Hamlet.txt","r").read().lower()

for ch in '!"#$%&()\*+,-./:;<=>?@[\]^\_ {|}~':

txt = txt.replace(ch," ")

return txt

hamletTxt = getText()

words = hamletTxt.split()

counts = { }

for word in words:

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

items = list(counts.items())

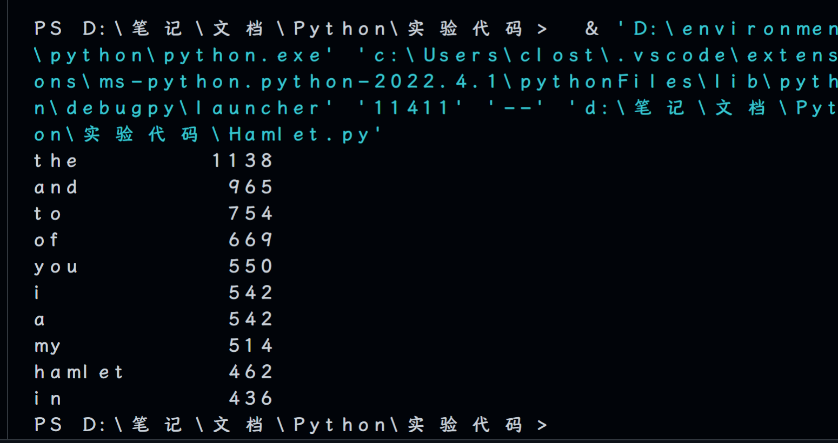
items.sort(key=lambda x:x[1],reverse=True)

for i in range(10):

word,count = items[i]

print("{0:<10}{1:>5}".format(word,count))

2.程序运行截图：

****

1.程序代码：

import jieba

txt = open("file\沉默的羔羊.txt",encoding="utf-8").read()

words = jieba.lcut(txt)

counts = { }

for word in words:

if len(word) == 1:

continue

else:

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

items = list(counts.items())

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

for i in range(15):

word,count = items[i]

print("{0:<10}{1:>5}".format(word,count))

2.程序运行截图：



|  |  |
| --- | --- |
| 成绩： | 评阅人： |

成绩：

**实验五 ：数据分析与可视化综合实验**

**实验目的：**

1、熟练安装扩展库numpy、pandas、matplotlib。

2、熟悉使用扩展库pandas 进行数据分析的基本操作。

3、熟悉使用扩展库matplotlib 进行数据可视化的基本操作。

**实验内容：**

编写程序，使用Numpy库的array函数创建二维数组，参照输出示例输出数组的维度、数组类型、数组元素个数和每个元素的大小。

创建的数组为：

[[ 1 2 3 4]

[ 4 5 6 7]

[ 7 8 9 10]]

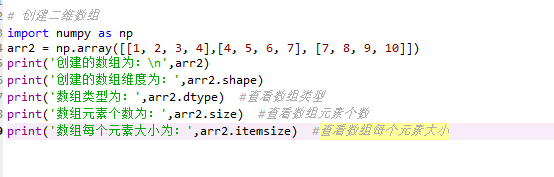
创建的数组维度为： (3, 4)

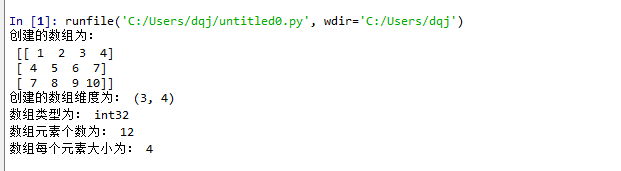
数组类型为： int64

数组元素个数为： 12

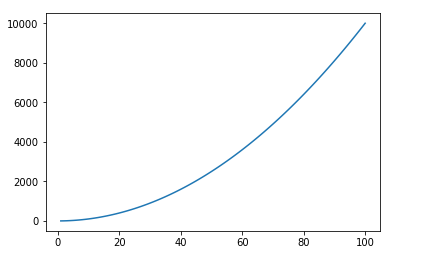
数组每个元素大小为： 8

参考代码：

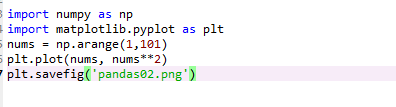




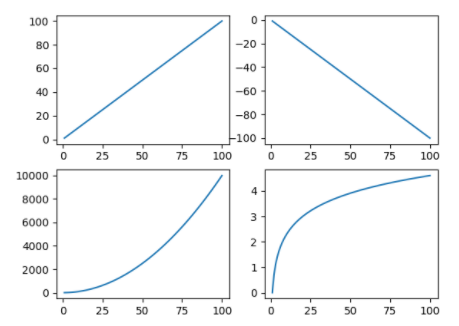
2、matplotlib绘制图形基础绘制y=x\*x图像。参考图1输出示例，输出y=x\*x图像。



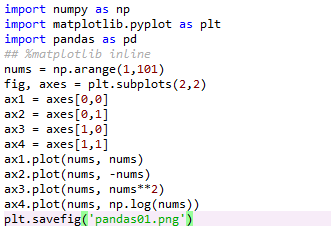
参考代码：



3、参考图2输出示例，输出图片。



参考代码：



1. 程序代码

import numpy as np

arr2 = np.array([[1,2,3,4],[4,5,6,7],[7,8,9,10]])

print("创建的数组为:\n",arr2)

print("创建的数据的维度为:",arr2.shape)

print("数组类型为:",arr2.dtype)

print("数组元素个数为",arr2.size)

print("数组每个元素大小为:",arr2.itemsize)

import imp

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

nums = np.arange(1,101)

fig,axes = plt.subplots(2,2)

ax1 = axes[0,0]

ax2 = axes[0,1]

ax3 = axes[1,0]

ax4 = axes[1,1]

ax1.plot(nums,nums)

ax2.plot(nums,-nums)

ax3.plot(nums,nums\*\*2)

ax4.plot(nums,np.log(nums))

plt.savefig('outPutFile/pandas01.png')

1. 运行截图

|  |  |
| --- | --- |
| 成绩： | 评阅人： |

成绩：

**实验六 ：爬取各省直辖市近十年GDP总值并可视化排名**

1、实验目的

爬取网络数据并进行可视化是在实际工程实践中常用的技能，本实验是一次实用性强的综合性实验。利用python的第三方库的进行网站数据的爬取与可视化，

2、实验内容

本综合实验涉及到的第三方库为time,re,selenium,pandas, bs4，Matplotlib;学生需熟悉掌握第三方库的安装与使用，熟悉使用selenium库的引擎启动器和ActionChains类的使用与原理。爬取过程：动态模拟鼠标+静态分析，可视化过程：数据处理+matplotlib，要求把全国各省直辖市近十年GDP总值排序并以折线与柱状图的形式可视化排名，通过本实验，学生可熟练使用第三方库从网站进行数据爬取，初步清洗爬取到的数据，利用第三方库进行可视化。

3、爬取各省直辖市近十年GDP总值并可视化排名

4、数据爬取

* 1. 网站数据爬取过程中遇到的问题是如何解决的？
  2. 调试的经验和体会。

5、可视化数据爬取结果

1） 网站数据爬取过程中遇到的问题是如何解决的？

2） 调试的经验和体会。

3） 尝试第三种方式可视化排名。

6、附录多种算法实现清单

带注释和功能模块说明的源程序清单（若程序短，可以附报告中，也可不附。但源程序都得另交一个或多个完整的文件）。

import json

import time

import requests

import openpyxl

import warnings

from pyecharts import options as opts

from pyecharts.charts import Bar, Map, Timeline

import pandas as pd

#获取当前时间戳

def getTime():

return int(round(time.time()\*1000))

# 获取数据

def getData():

url = 'https://data.stats.gov.cn/easyquery.htm?cn=E0103'

headers = {

"user-agent":'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36 Edg/107.0.1418.56'

}

params= {'m': 'QueryData', 'dbcode': 'fsnd', 'rowcode': 'reg', 'colcode': 'sj',

'wds': '[{"wdcode":"zb","valuecode":"A020101"}]', 'dfwds': '[{"wdcode":"sj","valuecode": "LAST20"}]',

'k1': str(getTime())

}

return json.loads(requests.get(url, headers=headers, verify=False, params=params).text)

#数据写入excel表

def formatData():

rawData = getData()["returndata"]

dataJson = rawData["datanodes"]

nameJson = rawData["wdnodes"][1]["nodes"]

yearJson = rawData["wdnodes"][2]["nodes"]

dataList = []

nameList = []

yearList = []

for item in dataJson:

dataList.append(item["data"]["strdata"])

for item in nameJson:

nameList.append(item["cname"])

for item in yearJson:

yearList.append(item["cname"])

wb = openpyxl.Workbook()

ws = wb.active

ws.column\_dimensions.width = 60

#第一行数据

ws.cell(1,1,'地区')

for i in range(2,len(yearList)):

ws.cell(1,i,yearList[i-2])

#第一列数据

for i in range(2,len(nameList)):

ws.cell(i,1,nameList[i-2])

# 具体数据

for i in range(2,len(nameList)):

for j in range(2,len(yearList)):

ws.cell(i,j,dataList[(len(yearList))\*(i-2) + j - 2])

wb.save('outPutFile/GdpData.xlsx')

#可视化数据

datas = pd.read\_excel('outPutFile/GdpData.xlsx')

def get\_gdp\_bar1(datas):

c = (

Bar()

.add\_xaxis(datas['地区'].values.tolist())

.add\_yaxis('全国各省GDP(亿元)', datas['2021年'].values.tolist())

.set\_global\_opts(

title\_opts=opts.TitleOpts(title='2021年全国各省GDP(亿元)'),

datazoom\_opts=[opts.DataZoomOpts(), opts.DataZoomOpts(type\_='inside')],

)

.render('outPutFile/2021年全国各省GDP\_Bar1.html')

)

def get\_gdp\_bar2(datas):

sort\_info = datas.sort\_values(by='2021年', ascending=True)

c = (

Bar()

.add\_xaxis(sort\_info['地区'].values.tolist())

.add\_yaxis('全国各省GDP(亿元)', sort\_info['2021年'].values.tolist())

.reversal\_axis()

.set\_series\_opts(label\_opts=opts.LabelOpts(position="right"))

.set\_global\_opts(

title\_opts=opts.TitleOpts(title='2021年全国各省GDP(亿元)'),

datazoom\_opts=[opts.DataZoomOpts(is\_show=True

, orient='vertical')],

)

.render('outPutFile/2021年全国各省GDP\_Bar2.html')

)

def get\_gdp\_map1(datas):

datas['地区'].replace(regex=True, inplace=True, to\_replace=['省', '市', '维吾尔自治区', '回族自治区', '壮族自治区', '自治区'], value=r'')

map = (

Map()

.add('全国各省GDP(亿元)', datas[['地区', '2021年']].values.tolist(), 'china')

.set\_global\_opts(

title\_opts=opts.TitleOpts(title='2021年全国各省GDP(亿元)'),

visualmap\_opts=opts.VisualMapOpts(max\_=110000),

)

)

map.render('outPutFile/2021年全国各省GDP\_Map.html')

def get\_gdp\_map2(datas):

datas['地区'].replace(regex=True, inplace=True, to\_replace=['省', '市', '维吾尔自治区', '回族自治区', '壮族自治区', '自治区'], value=r'')

tl = Timeline()

# 播放的速度，单位毫秒（ms）

tl.add\_schema(play\_interval=300, symbol='emptydiamond')

for i in range(2004, 2022):

map0 = (

Map()

.add('全国各省GDP(亿元)', datas[['地区', str(i) + '年']].values.tolist(), 'china')

.set\_global\_opts(

title\_opts=opts.TitleOpts(title='{}年全国各省GDP(亿元)'.format(i)),

visualmap\_opts=opts.VisualMapOpts(max\_=110000,is\_piecewise=True),

)

)

tl.add(map0, '{}年'.format(i))

tl.render('outPutFile/2004-2021年全国各省GDP.html')

warnings.filterwarnings('ignore')

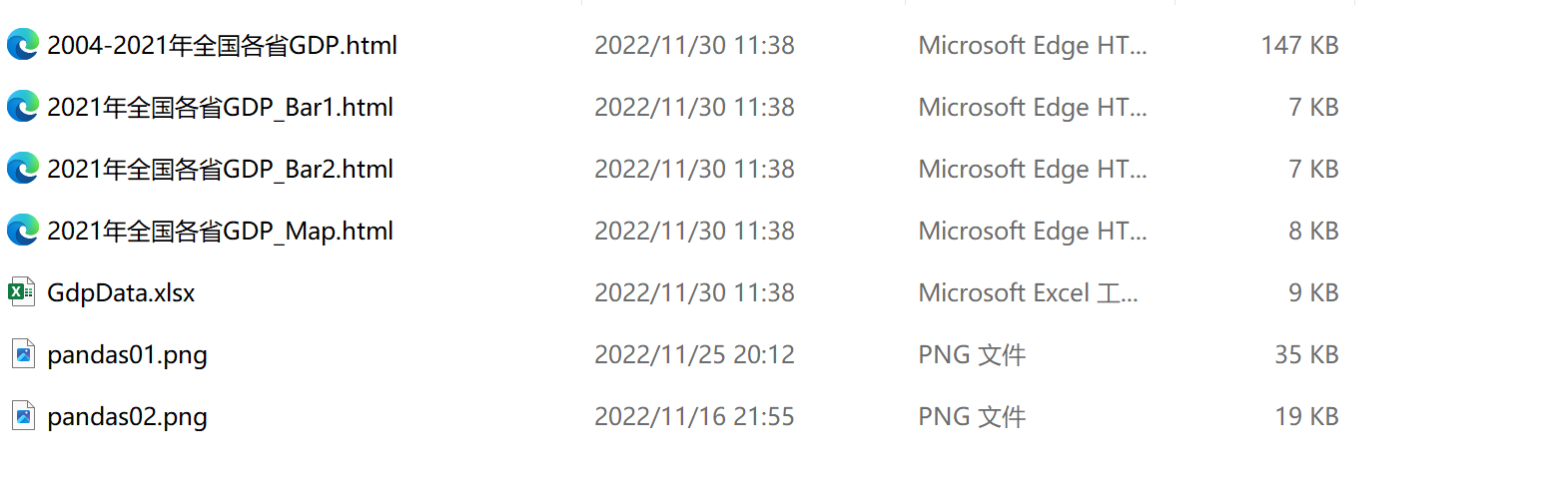
formatData()

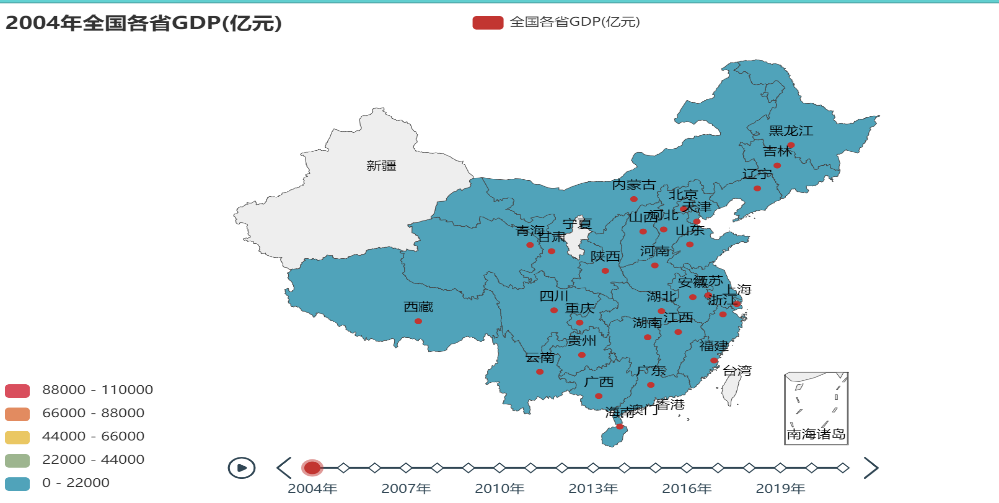
get\_gdp\_bar1(datas)

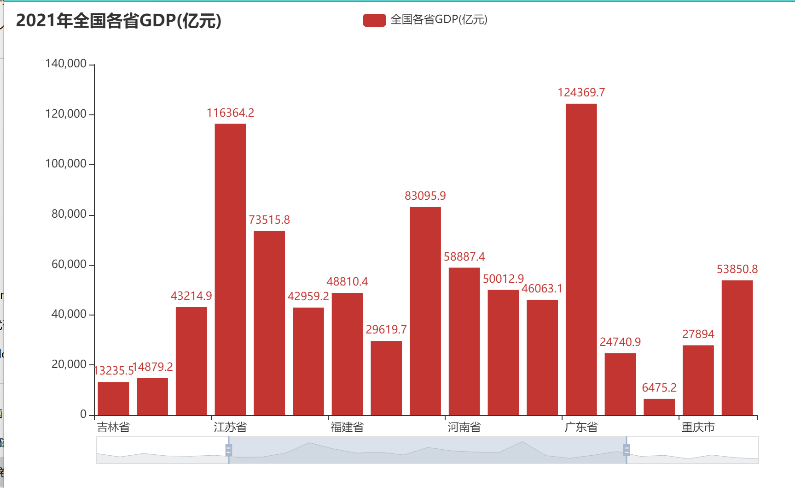
get\_gdp\_bar2(datas)

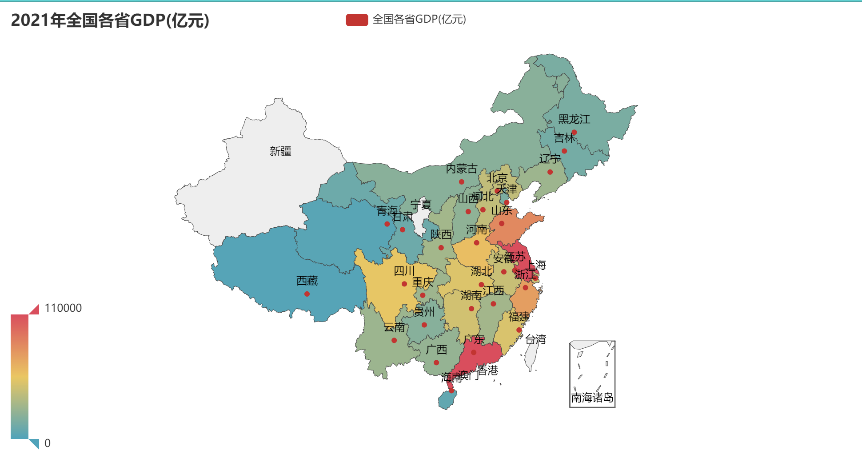
get\_gdp\_map1(datas)

get\_gdp\_map2(datas)









|  |  |
| --- | --- |
| 成绩： | 评阅人： |

成绩：