山东大学 软件工程 学院

2019-2020学年第1学期

Python程序设计与大数据分析 课程实验报告<模板>

**注：实验报告电子版文件名命名方式：学号+班级+姓名+实验\*。**

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：201800301249 | 姓名：王帅 | | 班级： 软件2018.4班 |
| 实验题目：组合数据类型应用练习 | | | |
| 实验学时：4 | | 实验日期： 2020年4月1日 | |
| 一、实验类型：验证型 二、建议学时：4 三、实验目的：  1.掌握Python语言开发环境IDEL的使用；  2.了解Python语言程序的基本结构  四、实验内容： 1、阅读教材第6章：组合数据类型 2、验证型练习：  2.1、验证序列类型、集合类型、列表类型、字典类型的相关操作方法和函数（教材 P156-167 表6.1~表6.5） （注：该部分练习不用提交）  2.2、 验证6.5 jieba库常用的分词函数(表6.6)（注：该部分练习不用提交）  2.3、验证6.6 实例10：文本词频统计中实例代码10.1-10.4 3、设计型练习：  教材第6章中的程序练习题选: 选做习题6.1、6.2、6.4、6.6。  五、实验教材：《Python 语言程序设计基础（第2版）》 | | | |
| 硬件环境：  个人电脑 | | | |
| 软件环境：  IDLE | | | |
| 实验步骤与内容：  实验步骤：   1. 打开IDLE，根据实验要求，熟悉其使用 2. 根据书上的代码，深入自己对python的理解。   实验内容：教材PPT第1、2、3章种的范例程序代码 | | | |
| 结论分析与体会：  Python确实是一个比大多数语言都要方便简单的语言，简洁明了，大大减少了编写代码的时间。  虽然语法较为简单，但是自己的特有方法并不少，都是使用起来可以使我们的编程更加简便的方法，应当合理使用。  注意在使用IDLE时控制缩进，否则可能会发生较大的错误。 | | | |

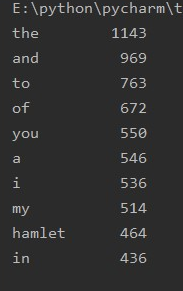
以下分别粘贴实验中验证型题目中通过的程序代码（详细编号）与运行结果；

粘贴实验中设计型题目中调试通过的程序代码（详细编号）与运行结果

10.1、

def getText():  
 txt = open("E:\python\hamlet.txt", "r").read()  
 txt = txt.lower()  
 for ch in '!"#$%&()\*+,-./:;<=>?@[\\]^\_‘{|}~':  
 txt = txt.replace(ch, " ") # 将文本中特殊字符替换为空格  
 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))

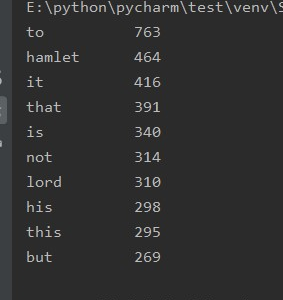
结果：



10.2、

excludes = {"the", "and", "of", "you", "a", "i", "my", "in"}  
  
  
def getText():  
 txt = open("hamlet.txt", "r").read()  
 txt = txt.lower()  
 for ch in '!"#$%&()\*+,-./:;<=>?@[\\]^\_‘{|}~':  
 txt = txt.replace(ch, " ") # 将文本中特殊字符替换为空格  
 return txt  
  
  
hamletTxt = getText()  
words = hamletTxt.split()  
counts = {}  
for word in words:  
 counts[word] = counts.get(word, 0) + 1  
for word in excludes:  
 del (counts[word])  
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))

结果：



10.3、

import jieba  
  
excludes = {} # {"将军","却说","丞相"}  
txt = open("三国演义.txt", "r", encoding='utf-8').read()  
words = jieba.lcut(txt)  
counts = {}  
for word in words:  
 if len(word) == 1: # 排除单个字符的分词结果  
 continue  
 else:  
 counts[word] = counts.get(word, 0) + 1  
for word in excludes:  
 del (counts[word])  
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))

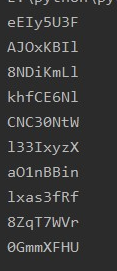
10.4、

import jieba  
  
excludes = {"将军", "却说", "荆州", "二人", "不可", "不能", "如此"}  
txt = open("三国演义.txt", "r", encoding='utf-8').read()  
words = jieba.lcut(txt)  
counts = {}  
for word in words:  
 if len(word) == 1:  
 continue  
 elif word == "诸葛亮" or word == "孔明曰":  
 rword = "孔明"  
 elif word == "关公" or word == "云长":  
 rword = "关羽"  
 elif word == "玄德" or word == "玄德曰":  
 rword = "刘备"  
 elif word == "孟德" or word == "丞相":  
 rword = "曹操"  
 else:  
 rword = word  
 counts[rword] = counts.get(rword, 0) + 1  
for word in excludes:  
 del (counts[word])  
items = list(counts.items())  
items.sort(key=lambda x: x[1], reverse=True)  
for i in range(5):  
 word, count = items[i]  
 print("{0:<10}{1:>5}".format(word, count))

6.1、

import random  
  
element = ["a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v",  
 "w",  
 "x", "y", "z",  
 "A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V",  
 "W",  
 "X", "Y", "Z",  
 "0", "1", "2", "3", "4", "5", "6", "7", "8", "9"]  
  
for i in range(10):  
 for j in range(8):  
 print(random.choice(element), end='')  
 print()

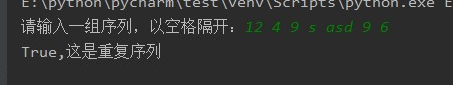
结果：



6.2、

def check(a): # 定义函数Lbpd(a)  
 a = a.split(" ") # 对参数a按照空格进行分词  
 if len(a) == len(set(a)): # 利用集合的不重复性，比较列表a和集合a的长度  
 return "False,这是非重复序列" # 如相同则返回非重复序列  
  
 else:  
 return "True,这是重复序列" # 如不同则返回重复序列  
  
  
t = f = 0 # 重复序列和非重复序列次数初始赋值  
  
a = input("请输入一组序列，以空格隔开：")  
print(check(a)) # 调用函数，打印函数参数为a时的返回值  
if check(a) == "True,这是重复序列": # 统计重复序列次数  
 t += 1  
if check(a) == "False,这是非重复序列": # 统计非重复序列次数  
 f += 1

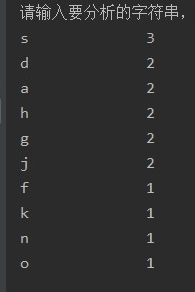
结果：



6.4、

str = input("请输入要分析的字符串，回车表示结束：")  
# 创建字典类型保存结果  
counts = {}  
# 扫描字符串，统计出现频率  
for ch in str:  
 counts[ch] = counts.get(ch,0) + 1  
# 改变类型为列表类型，按照出现频率降序排列  
items = list(counts.items())  
# 利用sort函数排序  
items.sort(key= lambda x : x[1],reverse= True)  
# 打印输出  
for i in range(len(items)):  
 word, count = items[i]  
 print("{0:<10}{1:>5}".format(word, count))

结果：



6.6、

import jieba  
  
path = '红楼梦.txt'  
# 根据路径以utf-8的格式读取文件内容  
txt = open(path, 'r', encoding='utf-8').read()  
words = jieba.lcut(txt)  
# 通过结果分析，记录需要排除的一些不是人名的名词  
excludes = ['这会子', '怎么样', '为什么', '周瑞家',  
 '贾母笑', '悄悄的', '大学生', '小说网', '电子书']  
# 定义空的词典类型  
counts = {}  
for word in words:  
 if len(word) == 1 or len(word) == 2:  
 continue  
 else:  
 counts[word] = counts.get(word, 0) + 1  
for word in excludes:  
 del counts[word]  
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))

结果：

