山东大学网络空间安全学院

Python高级程序设计 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202100460065 | 姓名：李昕 | | 班级：密码 21 二班 |
| 实验题目：熟悉Python开发环境、运算符、内置函数以及Python序列应用 | | | |
| 实验学时：2 | | 实验日期：2022.9.27 | |
| 实验目的：熟悉开发环境的基本操作、熟练运用Python 运算符和内置函数，熟悉Python列表、元组、字典、集合的相关操作，配置和熟悉实验平台，完成实验环境及实验工具的熟悉。编写代码，完成对Python列表、元组、字典、集合的相关操作功能的验证。 | | | |
| 硬件环境：  AMD Ryzen 7 5800H with Radeon Graphics     3.20 GHz  机带 RAM 16.0 GB (13.9 GB 可用) | | | |
| 软件环境：  操作系统：windows 11  编译器：IDLE | | | |
| 实验步骤与内容：  本次实验的实现可以分为八个问题：   1. .若未安装Python则安装Python；掌握Python代码交互式运行方式；掌握Python基本输入输出函数或语句的用法；了解Python代码编写规范；熟悉Python标准库和扩展库的导入和使用方式； 2. .编写程序，输入任意大的自然数，输出各位数字之和； 3. .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； 4. .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 5. .熟悉Python列表元素增加、删除；+、\*运算符对列表、元组对象的作用；Python列表中间位置插入和删除元素；运用成员测试运算符in在遍历序列元素；序列解包；列表排序方法sort()和内置排序函数sorted()的用法；字典对象的get()方法 6. .编写程序。构造一个列表，元素为1000以内的所有素数 7. .编写代码。生成随机密码并验证其是否为强密码，若字符串中同时包含英文字母大小写、数字和标点符号则认为是强密码。 8. .编写程序。根据下图生成一个字典，包括演员名字、饰演角色、配音演员。然后打印出刘昊然扮演的角色是谁。创建一个备份字典。假如江疏影有事情不能参加本次拍摄，请在演员表中去除他的信息。然后更替为张静初。增加新的角色，如下图所示. 打印出备份字典中的所有演员及其角色，并统计有多少个角色。   【输出任意大自然数各位数字之和】  可以它分成多个子问题：  1.用input读取键盘输入的任意大自然数  2.自定义求各位数字之和的函数，返回值用print打印  3.运用求模运算将该数值的各个位数从低位开始逐位相加，并将最后的值return返回  代码实现：   1. def sum(a):  *#求任意大自然数的每位的和* 2. b=0 3. while(a/10>0): 4. b=b+a % 10 5. a=a//10 6. return b 7. a=input("请输入一个任意大的自然数:") 8. a=int(a) 9. print(sum(a))   运行结果：    【输出两个集合的交集、并集和差集】  有两种方法可以解决集合问题：  法一：  Python集合内置支持交集、并集、差集运算  代码实现：   1. b={1,2,3,5,6,8,9} 2. c={2,3,4,6,7,9,10,15} 3. print(b & c) *#交集* 4. print(b | c) *#并集* 5. print(b.difference(c)) *#差集*   法二：  1.交集：用set函数定义一个新的空集合k，将同时存在于两个集合中的元素使用add方法添加到k集合中，完成后将k作为返回值返回；  2.并集：定义一个新的空集合k，将存在于两个集合中的元素不重复的使用add方法添加到k集合中，完成后将k作为返回值返回；  3.差集：定义一个新的空集合k，将存在于A中但不存在于B中的元素使用add方法添加到k集合中，完成后将k作为返回值返回  代码实现：   1. def AjiaoB(b,c):  *#交集* 2. k=set() 3. for i in b: 4. if i in c: 5. k.add(i) 6. return k 7. def AbingB(b,c):  *#并集* 8. k=set() 9. for i in b: 10. if i not in k: 11. k.add(i) 12. for i in c: 13. if i not in k: 14. k.add(i) 15. return k 16. def AjianB(b,c):  *#差集* 17. k=set() 18. for i in b: 19. if i not in c: 20. k.add(i) 21. return k 22. print(AjiaoB(b,c)) 23. print(AbingB(b,c)) 24. print(AjianB(b,c))   运行结果：    【输出二进制、八进制和十六进制表示】  法一：  可以将进制转换问题划分为几个子问题：  1.使用input函数读取一个十进制表示的自然数，并将字符串转化为整数  2.自定义函数，调用bin()、oct()、hex()将数字用对应进制输出  代码实现：   1. def jinzhi(d):  *#输出二进制、八进制和十六进制表示* 2. print(bin(d)) 3. print(oct(d)) 4. print(hex(d)) 5. d=int(input("请输入一个自然数：")) 6. jinzhi(d)   运行结果：    法二：  1.利用字符串格式化将对应进制输出  代码实现：   1. def jinzhi(d):  *#输出二进制、八进制和十六进制表示* 2. print(bin(d)) 3. print(oct(d)) 4. print(hex(d)) 5. d=int(input("请输入一个自然数：")) 6. jinzhi(d) 7. print("{0:b}".format(int(d))) *#二进制* 8. print("%o" % d) *#八进制* 9. print("%x" % d) *#十六进制*   运行结果：    【输出包含1000以内的所有素数的列表】  可以它分成两个子问题：  1.自定义一个函数函数，返回一个元素为1000以内的所有素数的列表  2.用print将返回值打印出来  代码实现：   1. def list1():  *#构造一个元素为1000以内的所有素数的列表* 2. k=list() 3. for i in range(0,1000): 4. for j in range(2,i//2+1): 5. if i%j==0: 6. break 7. else: 8. k.append(i) 9. return k 10. print(list1())   运行结果：    【生成随机密码并判断是否是强密码】  可以将密码生成、判断问题划分为两个子问题：  1.引入random模块，生成8位随机字符串；  2.通过判断ASCII范围判定是否具备强密码的条件，将不同的情况（大写、小写、数字、字符）写入列表c，若最后c的长度不小于3则为强密码。  代码实现：   1. import random 2. def suiji():  *#生成随机密码* 3. k=list() 4. c=list() 5. for i in range(1,8): 6. a=random.randint(33,126) 7. if 47<a<58: 8. b=1 9. elif 64<a<91: 10. b=2 11. elif 96<a<123: 12. b=3 13. else: 14. b=4 15. if b not in c: 16. c.append(b) 17. k.append(chr(a)) 18. k=''.join(k) 19. if len(c)>=3: 20. print("是强密码：",k) 21. else: 22. print("不是强密码：",k) 23. suiji()   运行结果：      【输出演员、角色信息】  将该问题划分为两个子问题：  1.{\*\*addict，\*\*bdict}的形式可以生成一个字典副本，将字典合并  2.使用del删除字典中指定键的元素  代码实现：   1. def actor(a,b): 2. adict={\*\*a,\*\*b} *#字典合并* 3. print("刘昊然饰演的是：",adict['刘昊然']) 4. print("江疏影有事不能参加本次拍摄，更替为张静初") 5. del adict['江疏影'] 6. adict.update({'张静初':'宫羽衣'}) 7. print(adict) 8. adict1={"宣言":"白鹿颜", "魏千翔": "百里宁卿\_姜广涛", "刘冠成": "拓跋山月\_杨默", "江涛":"翼天瞻", "董勇":"吕嵩", "杨新鸣":"大合萨\_郭正建"} 9. adict2={"张智尧":"白毅\_宝木中阳", "陈昊宇":"小舟公主", "杨玏": "吕鹰扬", "吴佳怡": "嬴玉"} 10. adict={\*\*adict,\*\*adict1,\*\*adict2} 11. print("所有演员及角色：") 12. print(adict) 13. print("一共有",len(adict),"个角色") 14. dict1 = {"刘昊然": "吕归尘", "宋祖儿": "羽然", "陈若轩": "姬野\_许凯", "张志坚":"雷碧城", "李光洁": "息衍", "许晴": "白凌波"} 15. dict2 = {"江疏影":"宫羽衣\_韩啸","王鸥":"苏瞬卿","张丰毅":"嬴无翳\_宣晓明","张嘉译":"百里景洪"} 16. actor(dict1,dict2)   执行结果： | | | |
| 结论分析与体会：  通过本次练习，我熟悉了Python列表元素增加、删除；+、\*运算符对列表、元组对象的作用；Python列表中间位置插入和删除元素；运用成员测试运算符in在遍历序列元素；序列解包；列表排序方法sort()和内置排序函数sorted()的用法；字典对象的get()方法。通过六个程序的编写，我复习了字典的元素的添加与修改、读取、创建与删除等操作。 | | | |