

实验三 Python列表

班级： 21计科4班

学号： B20210202307

姓名： 左莉

Github地址： https://github.com/1128zl/git_python_practice 

CodeWars地址： <https://www.codewars.com/users/1128zl> 

实验目的

1. 学习Python的简单使用和列表操作
2. 学习Python中的if语句

实验环境

1. Git
2. Python 3.10
3. VSCode
4. VSCode插件

实验内容和步骤

第一部分

Python列表操作

完成教材《Python编程从入门到实践》下列章节的练习：

- 第3章 列表简介
- 第4章 操作列表
- 第5章 if语句

第二部分

在Codewars网站注册账号，完成下列Kata挑战：

第一题：3和5的倍数 (Multiples of 3 or 5)

难度： 6kyu

如果我们列出所有低于 10 的 3 或 5 倍数的自然数，我们得到 3、5、6 和 9。这些数的总和为 23. 完成一个函数，使其返回小于某个整数的所有是3 或 5 的倍数的数的总和。此外，如果数字为负数，则返回 0。

注意：如果一个数同时是3和5的倍数，应该只被算一次。

提示：首先使用列表解析得到一个列表，元素全部是3或者5的倍数。使用sum函数可以获取这个列表所有元素的和。

代码提交地址： <https://www.codewars.com/kata/514b92a657cdc65150000006>

第二题：重复字符的编码器 (Duplicate Encoder)

难度： 6kyu

本练习的目的是将一个字符串转换为一个新的字符串，如果新字符串中的每个字符在原字符串中只出现一次，则为"("，如果该字符在原字符串中出现多次，则为")"。在判断一个字符是否是重复的时候，请忽略大写字母。

例如：

```
1  "din"      => "(((("
2  "recede"   => "())())"
3  "Success"  => ")())())"
4  "(( @"     => "))((("
```

代码提交地址: <https://www.codewars.com/kata/54b42f9314d9229fd6000d9c>

第三题：括号匹配 (Valid Braces)

难度： 6kyu

写一个函数，接收一串括号，并确定括号的顺序是否有效。如果字符串是有效的，它应该返回True，如果是无效的，它应该返回False。 例如：

```
1  "(){}[]" => True
2  "([{}])" => True
3  "()" => False
4  "[()]" => False
5  "[({})]()" => False
```

提示： python中没有内置堆栈数据结构，可以直接使用list来作为堆栈，其中append方法用于入栈，pop方法可以出栈。

代码提交地址 <https://www.codewars.com/kata/5277c8a221e209d3f6000b56>

第四题：从随机三元组中恢复秘密字符串(Recover a secret string from random triplets)

难度： 4kyu

有一个不为你所知的秘密字符串。给出一个随机三个字母的集合，恢复原来的字符串。

这里的三个字母的组合被定义为三个字母的序列，每个字母在给定的字符串中出现在下一个字母之前。"whi"是字符串 "whatisup" 的一个三个字母的组合。

作为一种简化，你可以假设没有一个字母在秘密字符串中出现超过一次。

对于给你的三个字母的组合，除了它们是有效的三个字母的组合以及它们包含足够的信息来推导出原始字符串之外，你可以不做任何假设。特别是，这意味着秘密字符串永远不会包含不出现在给你的三个字母的组合中的字母。

测试用例：

```
1 secret = "whatisup"
2 triplets = [
3     ['t','u','p'],
4     ['w','h','i'],
5     ['t','s','u'],
6     ['a','t','s'],
7     ['h','a','p'],
8     ['t','i','s'],
9     ['w','h','s']
10 ]
11 test.assertEqual(recoverSecret(triplets), secret)
12
```

代码提交地址：<https://www.codewars.com/kata/53f40dff5f9d31b813000774/train/python>

提示：

- 利用集合去掉triplets中的重复字母，得到字母集合letters，最后的secret应该由集合中的字母组成，secret长度也等于该集合。

```
1 letters = {letter for triplet in triplets for letter in triplet }
2 length = len(letters)
```

- 创建函数check_first_letter(triplets, first_letter)，检测一个字母是不是secret的首字母，返回True或者False。
- 创建函数remove_first_letter(triplets, first_letter)，从三元组中去掉首字母，返回新的三元组。
- 遍历字母集合letters，利用上面2个函数得到最后的结果secret。

第五题：去掉喷子的元音 (Disemvowel Trolls)

难度： 7kyu

喷子正在攻击你的评论区! 处理这种情况的一个常见方法是删除喷子评论中的所有元音(字母: a,e,i,o,u), 以消除威胁。 你的任务是写一个函数, 接收一个字符串并返回一个去除所有元音的新字符串。 例如, 字符串 "This website is for losers LOL!" 将变成 "Ths wbst s fr lsrs LL!".

注意: 对于这个Kata来说, y不被认为是元音。 代码提交地址:
<https://www.codewars.com/kata/52fba66badcd10859f00097e>

提示:

- 首先使用列表解析得到一个列表, 列表中所有不是元音的字母。
- 使用字符串的join方法连结列表中所有的字母, 例如:

```
1 last_name = "lovelace"
2 letters = [letter for letter in last_name if letter not in 'aeiouy']
3 print(letters) # ['l', 'o', 'v', 'e', 'l', 'a', 'c', 'e']
4 name = ''.join(letters) # name = "lovelace"
```

第三部分

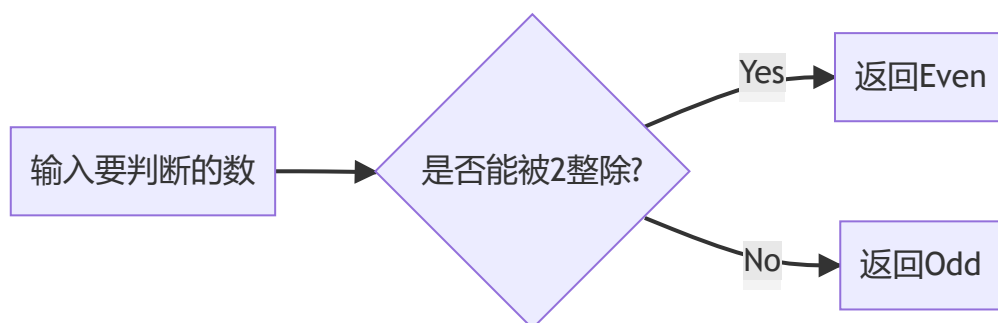
使用Mermaid绘制程序流程图

安装VSCode插件:

- Markdown Preview Mermaid Support
- Mermaid Markdown Syntax Highlighting

使用Markdown语法绘制你的程序绘制程序流程图 (至少一个), Markdown代码如下:

程序流程图



查看Mermaid流程图语法-->[点击这里](#)

使用Markdown编辑器 (例如VScode) 编写本次实验的实验报告, 包括实验过程与结果、实验考查和实验总结, 并将其导出为 PDF格式 来提交。

实验过程与结果

请将实验过程与结果放在这里, 包括:

- 第一部分 Python列表操作和if语句

第三章：

遍历姓名列表并问候

```
1 names = ['helen', 'jack', 'rose', 'jackson']
2 print(names[0])
3 print(names[1])
4 print(names[2])
5 print(names[3])
6 print(names)
7
8 mesg="how are you"
9 print(f"{names[0].title()}, {mesg} ?")
10 print(f"{names[1].title()}, {mesg} ?")
11 print(f"{names[2].title()}, {mesg} ?")
12 print(f"{names[3].title()}, {mesg} ?")
```

邀请嘉宾共进晚餐

```
1 persons = ['易烊千玺', '康康', '卢洋洋']
2 mesg1 = "今天有时间一起吃饭吗"
3 mesg2 = "有事不能赴约"
4 # print("初步邀请嘉宾名单：")
5 print(f"\n{persons[0]}, {mesg1} ?")
6 print(f"{persons[1]}, {mesg1} ?")
7 print(f"{persons[2]}, {mesg1} ?")
8
9 # 3.9 清点晚餐嘉宾
10 print(len(persons))
11
12 # 未能赴约嘉宾
13 print(f"\n{persons[2]} {mesg2}。")
14 persons[2]='周雨彤'
15
16 # print("\n最终邀请嘉宾名单：")
17 print(f"\n{persons[0]}, {mesg1} ?")
18 print(f"{persons[1]}, {mesg1} ?")
19 print(f"{persons[2]}, {mesg1} ?")
20
21 #添加三位嘉宾
22 persons.insert(0, '沈月')
23 persons.insert(2, '王敬轩')
24 persons.append("林允")
25
26 # print(persons)
27
28 print(f"\n{persons[0]}, {mesg1} ?")
29 print(f"{persons[1]}, {mesg1} ?")
```

```

30 print(f"{persons[2]}, {mesg1} ?")
31 print(f"{persons[3]}, {mesg1} ?")
32 print(f"{persons[4]}, {mesg1} ?")
33 print(f"{persons[5]}, {mesg1} ?")
34
35 print("\n由于食材原因，目前只能邀请两位嘉宾。")
36
37 # 逐步删减嘉宾，以确保最后只邀请两名嘉宾前来
38 while(len(persons)>2):
39
40     people=persons.pop()
41     print(f"{people}, {'很抱歉本次无法邀请您来共进晚餐'}。")
42
43
44 #发送信息，再次确认
45 print(f"\n{persons[0]}, {'您依然在受邀之列'}。")
46 print(f"{persons[1]}, {'您依然在受邀之列'}。")
47
48 del persons[0]
49 del persons[0]
50
51

```

管理列表--放眼世界

```

1  places = ['shanghai', 'beijing', 'suzhou', 'chongqing', 'xian']
2
3  print(places)
4  print()
5
6  # sorted() 字母顺序打印
7  print(sorted(places))
8  print()
9
10 #sorted() 字母顺序反向打印
11 print(sorted(places, reverse=True))
12 print()
13
14 #检查是否places本身的顺序被改变
15 print(places)
16 print()
17
18 # reverse()修改places顺序
19 places.reverse()
20
21 print(places)
22 print()
23
24 places.reverse()
25
26 # reverse() 将places顺序改回来

```

```

27 print(places)
28 print()
29
30 '''
31
32 #sort() 字母顺序
33
34 '''
35 # print(places.sort())
36 # 因为places.sort(reverse=True)和places.sort()没有返回值，所以打印的就是None
37 places.sort()
38 print()
39
40 # 检查sort()方法可不可以对本身顺序造成不可逆的改变
41 print(places)
42 print()
43
44 # sorted() 字母顺序反向打印
45 # print(places.sort(reverse=True))
46 places.sort(reverse=True)
47
48 print(places)
49

```

第四章:

动物 (for循环遍历打印)

```

1 animals=['panda','dog','cat','horse']
2
3 for animal in animals :
4     print(animal)
5     print(f"{'A'} {animal} {'would make a great pet'}.")
6
7 print(f"{'Any of these animals would make a great per'}!")

```

100万求和

```

1 values =list(range(1,1000001))
2 # print(values)
3 print(min(values),max(values),sum(values))

```

立方

```

1 list_3=[value**3 for value in range(1,11)]
2 for value in list_3:
3     print(value)

```

切片

```
1
2 list_3=[value**3 for value in range(1,11)]
3
4 print('The first three items in the list are:')
5 print(list_3[0:3])
6 #print(list_3[:3])
7
8 print('The items from the middle of the list are:')
9 print(list_3[3:6])
10
11 print('The last three items in the list are:')
12 print(list_3[-3:])
13 #print(list_3[7:])
```

自助餐

```
1 foods=('noodles','cake','rice','apple')
2 for food in foods:
3     print(food)
4 # foods[1]='ice cream'
5
6 foods=('ice cream','cake','orange','apple')
7 for food in foods:
8     print(food)
```

第五章:

条件测试

```
1 m='CAR'
2 print('subaru'=='car')
3 print()
4
5 print('subaru'=='subaru')
6 print()
7
8 print(m.lower()=='subaru')
9 print()
10
11 print(m.lower()=='car')
12 print()
13
14 print(5 in range(1,5) and 5%2)
15
16 print(5 in range(1,5) or 4 in range(1,5))
```


外星人颜色

```
1
2 alien_color='yellow'
3 if(alien_color=='green'):
4     print("You win 5 scores!")
5 elif(alien_color=='red'):
6     print("You win 10 scores!")
7 else :
8     print("You win 15 scores!")
```

用户名

```
1 users_name=['jack','helen','lily','admin']
2
3 for name in users_name:
4     if(name=='admin'):
5         print("Hello admin,would you like to see a status report?")
6     else:
7         print(f"{'Hello'} {name.title()},{'thank you for logging in again.'}")
8 users_name.pop()
9 users_name.pop()
10 users_name.pop()
11 users_name.pop()
12 if(users_name==[]):
13     print("We need to find some users.")
14
```

• 第二部分 Codewars Kata挑战

第一题:

```
1 def solution(number):
2
3     sum = 0
4     #保存累加和
5
6     #判断是否是3和5的倍数，是就加
7     if number > 0:
8         for i in range(0, number ):
9             if i % 3 == 0 or i % 5 == 0:
10                 sum = sum + i
11
12     return sum
```

第二题:

```

1 def duplicate_encode(word):
2     #字符串全部小写
3     word = word.lower()
4     result = ""
5     for char in word:
6         #利用count函数计算某个字符在字符串中的出现次数
7         if word.count(char) > 1:
8             result += ")"
9         else:
10            result += "("
11    return result

```

第三题:

```

1 def valid_braces(string):
2
3     list=[]
4
5     #将第一个字符放入列表
6     list.append(string[0])
7
8     #遍历剩余字符，如果和列表的最后一个匹配，就将列表最后一个元素删除，
9     #反之则将该字符放入列表最后
10
11    for index in range(1,len(string)):
12        # 分为列表为空，列表有元素且可匹配，列表有元素不可匹配三种情况
13
14        if len(list)==0:
15            list.append(string[index])
16
17        elif((list[-1]=='(' and string[index]==')') or (list[-1]=='[' and
string[index]==']') or (list[-1]=='{' and string[index]=='}')):
18            list.pop()
19
20        else:
21            list.append(string[index])
22    return list==[]

```

第四题:

```

1 def recoverSecret(triplets):
2     'triplets is a list of triplets from the secret string. Return the string.'
3
4     #将三元组中出现的字符全部放入到lic列表中
5     lic = []
6     for i in range(len(triplets)):
7         for j in range(len(triplets[i])):
8             lic.append(triplets[i][j])
9

```

```

10     lic = set(lic)
11     lic = list(lic)
12
13     #三元组中的字符进行比较决定是否要交换位置
14     while True:
15         cnt = 0
16         for i in range(len(triplets)):
17             for j in range(len(triplets[i]) - 1):
18                 a = triplets[i][j]
19                 b = triplets[i][j + 1]
20                 index_a = lic.index(a)
21                 index_b = lic.index(b)
22                 if index_a > index_b:
23                     temp = lic[index_b]
24                     lic[index_b] = lic[index_a]
25                     lic[index_a] = temp
26                     cnt += 1
27
28     if cnt == 0:
29         return ''.join(lic)

```

第五题:

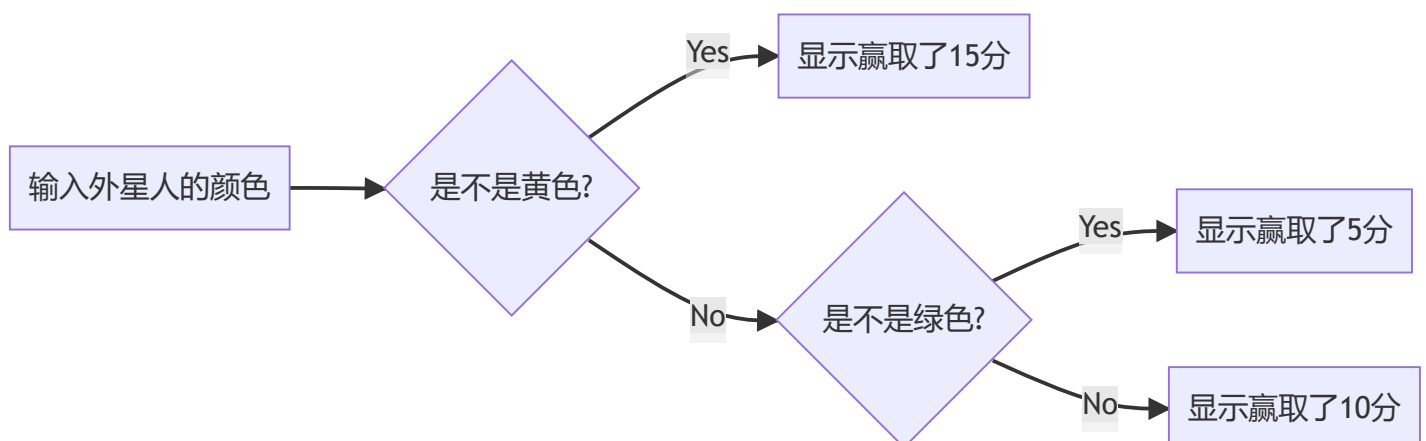
```

1  def disemvowel(string_):
2      #设置列表保存非元音字符
3      list=[]
4
5      for index in range(len(string_)):
6          if string_[index].lower() not in ['a', 'e', 'i', 'o', 'u']:
7              #依次判断决定是否添加
8              list.append(string_[index])
9      string_ = ''.join(list)
10     return string_

```

• 第三部分 使用Mermaid绘制程序流程图

根据外星人颜色来决定得分情况



注意代码需要使用markdown的代码块格式化，例如Git命令行语句应该使用下面的格式：

Git命令

显示效果如下：

```
1 | git init
2 | git add .
3 | git status
4 | git commit -m "first commit"
```

如果是Python代码，使用代码块格式，显示效果如下：

```
1 | def add_binary(a,b):
2 |     return bin(a+b)[2:]
```

代码运行结果的文本可以直接粘贴在这里。

注意：不要使用截图，Markdown文档转换为Pdf格式后，截图可能会无法显示。

实验考查

请使用自己的语言并使用尽量简短代码示例回答下面的问题，这些问题将在实验检查时用于提问和答辩以及实际的操作。

1. Python中的列表可以进行哪些操作？

可以对列表元素进行排序，修改，删除，也可以添加新的元素加入列表。

2. 哪两种方法可以用来对Python的列表排序？这两种方法有和区别？

使用sorted()进行列表排序不会影响列表的原始顺序，而使用sort()进行列表排序会影响列表的原始顺序。

3. 如何将Python列表逆序打印？

使用reverse()函数。

4. Python中的列表执行哪些操作时效率比较高？哪些操作效率比较低？是否有类似的数据结构可以用来替代列表？

索引，赋值，切片等操作效率比较高，插入，删除，搜索等操作效率比较低，可以采用链表结构来代替列表，以提高操作效率。

5. 阅读《Fluent Python》Chapter 2. An Array of Sequence - Tuples Are Not Just Immutable Lists小节（p30-p35）。总结该小节的主要内容。

元组是使用圆括号进行创建的，不可以单独针对元组的某个元素来改变值，只能够重新定义元组。对于元组的元素，依然可以使用索引来访问。

实验总结

本次实验过程中，不仅了解了新的数据结构--元组，也对于列表的各种操作进行了学习并加以运用。整个实验中，对于三元组的卡塔挑战存在一点的困难，最后在不断的学习与尝试下得到了解决。