

实验三 Python列表

班级： 21计科1

学号： B20210302115

姓名： 文嘉

Github地址： https://github.com/Da-BuLiu/python_course

CodeWars地址： <https://www.codewars.com/users/Da-BuLiu>

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：

```
"din"      => "(((  
"recede"   => "())()  
"Success"  => ")())()  
"(( @"     => "))((
```

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：

```
"(){}[]" => True  
"([{}])" => True  
"{}" => False  
"[]" => False  
"[({})]()" => 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：

```
secret = "whatisup"
triplets = [
    ['t', 'u', 'p'],
    ['w', 'h', 'i'],
    ['t', 's', 'u'],
    ['a', 't', 's'],
    ['h', 'a', 'p'],
    ['t', 'i', 's'],
    ['w', 'h', 's']
]
test.assert_equals(recoverSecret(triplets), secret)
```

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。

```
letters = {letter for triplet in triplets for letter in triplet}
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：

```
last_name = "lovelace"
letters = [letter for letter in last_name ]
print(letters) # ['l', 'o', 'v', 'e', 'l', 'a', 'c', 'e']
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

显示效果如下：

```
graph LR
    A[Start] --> B{Is it?}
    B -->|Yes| C[OK]
    C --> D[Rethink]
    D --> B
    B -.->|No| E[End]
```

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](#)
- [第二部分 Codewars Kata挑战](#)

第一题：3和5的倍数（Multiples of 3 or 5）

```
def solution(number):  
    if number < 0:  
        return 0  
  
    list = []  
  
    for i in range(1,number):  
        if i % 3 == 0 or i % 5 == 0:  
            list.append(i)  
  
    return sum(list)
```

Time: 518ms Passed: 46 Failed: 0

Test Results:

> Fixed tests

▼ Random tests

> Random tests (35 of 35 Assertions)

Completed in 2.93ms

You have passed all of the tests! :)

第二题：重复字符的编码器 (Duplicate Encoder)

```
def duplicate_encode(word):  
    word = word.lower()  
  
    new_word = ""  
  
    for ch in word:  
        if word.count(ch) > 1:  
            new_word += ")"  
        else:  
            new_word += "("  
  
    return new_word
```

Time: 560ms Passed: 48 Failed: 0

Test Results:

▼ Duplicate Encoder

- Basic Test Cases (6 of 6 Assertions)
- Tests with '(' and ')' (2 of 2 Assertions)
- And now... some random tests ! (40 of 40 Assertions)

Completed in 5.64ms

You have passed all of the tests! :)

第三题：括号匹配 (Valid Braces)

```
def valid_braces(string):  
  
    stack = []  
    brackets= {'{':'}, '(':')', '[':']'}  
    for ch in string:  
        if ch in brackets:  
            stack.append(ch)  
        elif not stack or ch != brackets[stack.pop()]:  
            return False  
  
    return not stack
```

Time: 503ms Passed: 13 Failed: 0

Test Results:

▼ Valid Braces

➤ sample Tests (13 of 13 Assertions)

Completed in 0.13ms

You have passed all of the tests! :)

第五题：去掉喷子的元音 (Disemvowel Trolls)

```
def disemvowel(string_):  
    words = ['a', 'e', 'i', 'o', 'u']  
    new_string = ''  
    for char in string_:  
        if char.lower() not in words:  
            new_string += char  
  
    return new_string
```



- 第三部分 使用Mermaid绘制程序流程图

第一题：3和5的倍数 (Multiples of 3 or 5)

```
graph TD
  A[Start] --> B{number < 0}
  B -- Yes --> C[Return 0]
  B -- No --> D[Create an empty list]
  D --> E{ i < number }
  E -- Yes --> F{ i % 3 == 0 or i % 5 == 0 }
  F -- Yes --> G[Append i to the list]
  G --> D
  F -- No --> E
  E -- No --> H[Return the sum of the list]
  H --> I[End]
```

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：

```
git init
git add .
```



```
git status
git commit -m "first commit"
```

如果是Python代码，应该使用下面代码块格式，例如：

 Python代码

显示效果如下：

```
def add_binary(a,b):
    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？

Python中的列表可以进行以下操作：

访问元素：通过索引访问列表中的元素，可以使用正数索引从左到右访问元素，使用负数索引从右到左访问元素。

修改元素：通过索引可以修改列表中的元素。

增加元素：可以使用append()方法将元素添加到列表的末尾，使用insert()方法在指定位置插入元素，使用extend()方法将另一个列表的元素添加到当前列表的末尾。

删除元素：使用del语句可以删除列表中的元素，使用remove()方法可以按值删除元素，使用pop()方法可以删除并返回指定索引位置的元素。

切片操作：可以使用切片操作来获取列表的子列表，可以指定起始索引、结束索引和步长。

搜索元素：可以使用in关键字来判断一个元素是否在列表中。

长度计算：可以使用len()函数获取列表的长度。

遍历列表：可以使用for循环遍历列表中的元素。

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？

Python中对列表进行排序的两种方法是：

使用sort()方法：该方法会直接对原列表进行排序，不会返回新的列表。

使用sorted()函数：该函数会返回一个新的排序后的列表，不会修改原列表。

这两种方法的区别在于，sort()方法是就地排序，会修改原列表的顺序，而sorted()函数会返回一个新的排序后的列表，不会修改原列表。

3. 如何将Python列表逆序打印？

要将Python列表逆序打印，可以使用`reverse()`方法将列表进行反转，然后使用`for`循环遍历列表并打印每个元素。

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

在Python中，对列表进行索引和切片操作的效率比较高，因为列表是基于数组实现的，可以通过索引直接访问元素。而对列表进行插入和删除操作的效率比较差，因为需要移动其他元素来保持列表的连续性。

可以使用`collections.deque`类来替代列表。`deque`是双端队列，支持高效的插入和删除操作，可以在列表的两端进行操作，而不需要移动其他元素。

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。

列表和元组是Python中最常用的序列类型，列表是可变的，元组是不可变的。
元组可以作为字典的键，而列表不能。
列表和元组都可以通过索引和切片进行访问，但元组不支持修改操作。
可以使用`tuple()`函数将列表转换为元组，使用`list()`函数将元组转换为列表。
列表和元组都可以进行拆包操作，将序列中的元素分别赋值给多个变量。
列表推导式是一种快速创建列表的方法，类似于数学中的集合推导式。
元组作为函数的返回值可以让函数返回多个值。
元组可以用于交换变量的值，可以用于函数参数的解构赋值。
元组可以用于命名元组，提高代码的可读性。
元组可以用于表示不可变的记录，列表则更适合表示可变的集合

实验总结

通过本次实验我学习了Python中的列表操作跟`if`语句，并且通过了Codewars Kata的一些相应挑战，提高了自己的编程能力。