

实验五 Python数据结构与数据模型

班级： 21计科4班

学号： B20210302426

姓名： 陈佩儿

Github地址： https://github.com/shaliey/python_course

CodeWars地址：

<https://www.codewars.com/users/shaliey>

实验目的

1. 学习Python数据结构的高级用法
2. 学习Python的数据模型

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

在[Codewars网站](https://www.codewars.com/)注册账号，完成下列Kata挑战：

第一题：停止逆转我的单词

难度：6kyu

编写一个函数，接收一个或多个单词的字符串，并返回相同的字符串，但所有5个或更多的字母单词都是相反的（就像这个Kata的名字一样）。传入的字符串将只由字母和空格组成。只有当出现一个以上的单词时，才会包括空格。

例如：

```
spinWords( "Hey fellow warriors" ) => returns "Hey wollef sroirraw"  
spinWords( "This is a test") => returns "This is a test"  
spinWords( "This is another test" )=> returns "This is rehtona test"
```

代码提交地址：

<https://www.codewars.com/kata/5264d2b162488dc400000001>

提示：

- 利用str的split方法可以将字符串分为单词列表

例如：

```
words = "hey fellow warrior".split()  
# words should be ['hey', 'fellow', 'warrior']
```

- 利用列表推导将长度大于等于5的单词反转(利用切片word[::-1])
- 最后使用str的join方法连结列表中的单词。

第二题：发现离群的数(Find The Parity Outlier)

难度：6kyu

给你一个包含整数的数组（其长度至少为3，但可能非常大）。该数组要么完全由奇数组成，要么完全由偶数组成，除了一个整数N。请写一个方法，以该数组为参数，返回这个“离群”的N。

例如：

```
[2, 4, 0, 100, 4, 11, 2602, 36]
# Should return: 11 (the only odd number)
```

```
[160, 3, 1719, 19, 11, 13, -21]
# Should return: 160 (the only even number)
```

代码提交地址：

<https://www.codewars.com/kata/5526fc09a1bbd946250002dc>

第三题： 检测Pangram

难度：6kyu

pangram是一个至少包含每个字母一次的句子。例如，"The quick brown fox jumps over the lazy dog"这个句子就是一个pangram，因为它至少使用了一次字母A-Z（大小写不相关）。

给定一个字符串，检测它是否是一个pangram。如果是则返回 True，如果不是则返回 False。忽略数字和标点符号。

代码提交地址：

<https://www.codewars.com/kata/545cedaa9943f7fe7b000048>

第四题： 数独解决方案验证

难度：6kyu

数独背景

数独是一种在 9x9 网格上进行的 game。游戏的目标是用 1 到 9 的数字填充网格的所有单元格，以便每一列、每一行和九个 3x3 子网格（也称为块）中的都包含数字 1 到 9。更多信息请访问：

<http://en.wikipedia.org/wiki/Sudoku>

编写一个函数接受一个代表数独板的二维数组，如果它是一个有效的解决方案则返回 true，否则返回 false。数独板的单元格也可能包含 0，这将代表空单元格。包含一个或多个零的棋盘被认为是无效的解决方案。棋盘总是 9 x 9 格，每个格只包含 0 到 9 之间的整数。

代码提交地址：

<https://www.codewars.com/kata/63d1bac72de941033dbf87ae>

第五题： 疯狂的彩色三角形

难度： 2kyu

一个彩色的三角形是由一排颜色组成的，每一排都是红色、绿色或蓝色。连续的几行，每一行都比上一行少一种颜色，是通过考虑前一行中的两个相接触的颜色而产生的。如果这些颜色是相同的，那么新的一行就使用相同的颜色。如果它们不同，则在新的一行中使用缺失的颜色。这个过程一直持续到最后一行，只有一种颜色被生成。

例如：

Colour here:	G G	B G	R G	B R
Becomes colour here:	G	R	B	G

一个更大的三角形例子：

```
R R G B R G B B
R B R G B R B
G G B R G G
G R G B G
B B R R
B G R
R B
G
```

你将得到三角形的第一行字符串，你的工作是返回最后的颜色，这将出现在最下面一行的字符串。在上面的例子中，你将得到 "RRGBRBBB"，你应该返回 "G"。

限制条件： $1 \leq \text{length}(\text{row}) \leq 10 \times 5$

输入的字符串将只包含大写字母'B'、'G'或'R'。

例如：

```
triangle('B') == 'B'
triangle('GB') == 'R'
triangle('RRR') == 'R'
triangle('RGBG') == 'B'
triangle('RBRGBRB') == 'G'
triangle('RBRGBRBGGRRRBGBBBGG') == 'G'
```

代码提交地址：

<https://www.codewars.com/kata/5a331ea7ee1aae8f24000175>

提示：请参考下面的链接，利用三进制的特点来进行计算。

<https://stackoverflow.com/questions/53585022/three-colors-triangles>

第二部分

使用Mermaid绘制程序流程图

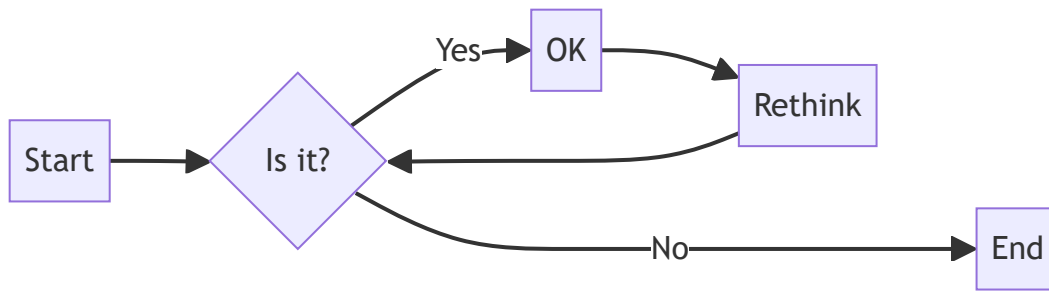
安装VSCode插件：

- 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：

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

显示效果如下：



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 Codewars Kata挑战](#)

第一题：停止逆转我的单词

```
def spin_words(sentence):  
    # Your code goes here  
    results=[]  
    words = sentence.split()  
    for word in words:  
        if len(word)>=5:  
            results.append(word[::-1])  
        else:  
            results.append(word)  
    return ' '.join(results)
```

第二题：发现离群的数(Find The Parity Outlier)

```
def find_outlier(integers):
    odds = []
    evens = []
    for integer in integers:
        if integer % 2 == 0:
            evens.append(integer)
        else:
            odds.append(integer)
    if len(evens)==1:
        return evens[0]
    else:
        return odds[0]
```

第三题：检测Pangram

```
def is_pangram(s):
    # 将输入字符串转换为小写
    s = s.lower()

    # 创建一个包含所有字母的集合
    alphabet = set("abcdefghijklmnopqrstuvwxyz")

    # 遍历输入字符串中的每个字符
    for char in s:
        # 如果字符是字母，从集合中移除
        if char.isalpha():
            alphabet.discard(char)

    # 检查集合是否为空，如果为空，说明是pangram
    return not alphabet
```

第四题：数独解决方案验证

```
def validate_sudoku(board):
    elements = set(range(1, 10))
    # row
    for b in board:
        if set(b) != elements:
            return False
    # column
    for b in zip(*board):
        if set(b) != elements:
            return False
    # magic squares
    for i in range(3, 10, 3):
        for j in range(3, 10, 3):
            if elements != {(board[q][w]) for w in range(j-3, j) for q in range(i-3, i)}:
                return False
    return True
```

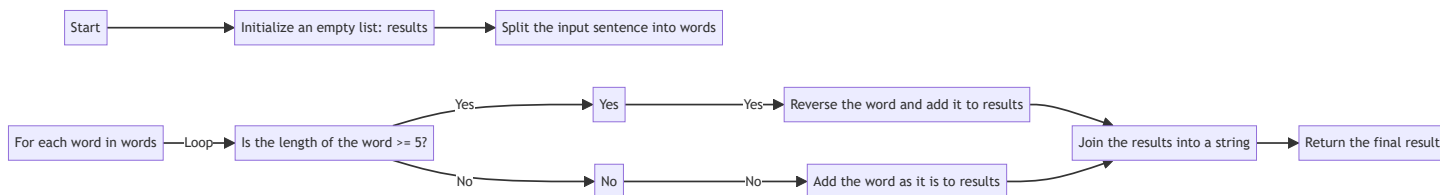
第五题：疯狂的彩色三角形

```
def triangle(row):
    while len(row) > 1:
        next_row = ""
        for i in range(len(row) - 1):
            if row[i] == row[i + 1]:
                next_row += row[i]
            else:
                if row[i] != "R" and row[i + 1] != "R":
                    next_row += "R"
                elif row[i] != "G" and row[i + 1] != "G":
                    next_row += "G"
                else:
                    next_row += "B"
        row = next_row
    return row
```

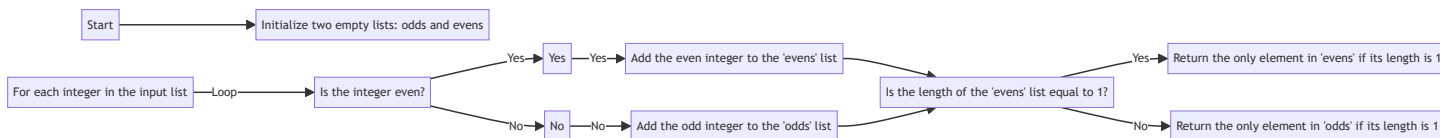
- [第二部分 使用Mermaid绘制程序流程图](#)

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：

第一题：停止逆转我的单词



第二题：发现离群的数(Find The Parity Outlier)



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

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. 集合 (set) 类型有什么特点？它和列表 (list) 类型有什么区别？

- 集合是一种无序、不重复的数据集合，不支持索引访问，用花括号 {} 表示。
- 列表是有序的数据集合，可以包含重复元素，用方括号 [] 表示。
- 区别：集合中元素无序，不重复；列表中元素有序，可重复。

2. 集合 (set) 类型主要有那些操作？

- 添加元素： `add()`
- 删除元素： `remove()` , `discard()`
- 集合运算：交集、并集、差集
- 成员检查： `in`

3. 使用 * 操作符作用到列表上会产生什么效果？为什么不能使用 * 操作符作用到嵌套的列表上？使用简单的代码示例说明。

- * 操作符用于重复列表中的元素，创建一个新列表，但不会影响原始列表。
- 不能用于嵌套列表，因为它只是复制了外部列表的引用，而不是创建内部列表的深拷贝。

```
# 列表重复
my_list = [1, 2]
repeated_list = my_list * 3
print(repeated_list) # 输出: [1, 2, 1, 2, 1, 2]

# 不能用于嵌套列表
nested_list = [[1, 2], [3, 4]]
repeated_nested_list = nested_list * 3
print(repeated_nested_list)
# 输出: [[1, 2], [3, 4], [1, 2], [3, 4], [1, 2], [3, 4]]
# 内部列表仍是原始引用, 改变内部列表一个会影响其他
```

4. 总结列表、集合、字典的解析 (comprehension) 的使用方法。使用简单的代码示例说明。

- **列表解析**: 通过一行代码生成新列表。

```
numbers = [1, 2, 3, 4, 5]
squared = [x**2 for x in numbers]
```

- **集合解析**: 通过一行代码生成新集合。

```
numbers = [1, 2, 2, 3, 4]
unique_squares = {x**2 for x in numbers}
```

- **字典解析**: 通过一行代码生成新字典。

```
names = ['Alice', 'Bob', 'Charlie']
name_lengths = {name: len(name) for name in names}
```

这些解析方法允许您在一行代码中创建新的数据结构，并且通常比使用传统的循环更简洁和高效。

实验总结

总结一下这次实验你学习和使用到的知识，例如：编程工具的使用、数据结构、程序语言的语法、算法、编程技巧、编程思想。

这次实验帮助我更好地理解和运用集合、字典和列表这三种常见的数据结构。我学会了它们的特点、操作和解析方法，这对于处理各种编程任务非常有帮助。这些数据结构是编程中的基本工具，掌握它们对于编写高效、清晰的代码至关重要。

1. 集合 (Set) :

- 集合是一种无序、不重复的数据结构，使用花括号 {} 表示。
- 集合主要用于存储唯一的元素，不允许重复。

- 常用操作包括添加元素、删除元素、集合运算（交集、并集、差集）以及成员检查。
- 集合解析用于通过一行代码生成新的集合。

2. 字典 (Dictionary) :

- 字典是一种键-值对的数据结构，使用花括号 {} 表示，每个键对应一个值。
- 字典允许快速查找值，因为它们使用键作为索引。
- 常用操作包括添加键值对、删除键值对、获取值、遍历键或值。
- 字典解析用于通过一行代码生成新的字典。

3. 列表 (List) :

- 列表是有序的数据结构，使用方括号 [] 表示，允许包含重复元素。
- 列表常用于存储一系列项目，支持索引访问、添加元素、删除元素、切片等操作。
- 列表解析用于通过一行代码生成新的列表。