

# 实验五 Python数据结构与数据模型

---

班级： 21计科1班

学号： B20210302128

姓名： 肖锟

Github地址： <https://github.com/xiaokun8888/python.git> 

CodeWars地址： <https://www.codewars.com/users/xk666> 

---

## 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](#)  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：

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

```
3 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:

```
1 words = "hey fellow warrior".split()  
2 # 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:

```
1 [2, 4, 0, 100, 4, 11, 2602, 36]  
2 # Should return: 11 (the only odd number)  
3  
4 [160, 3, 1719, 19, 11, 13, -21]  
5 # 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：

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

一个更大的三角形例子：

1	R R G B R G B B
2	R B R G B R B
3	G G B R G G
4	G R G B G
5	B B R R
6	B G R
7	R B
8	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：

```
1 triangle('B') == 'B'
2 triangle('GB') == 'R'
3 triangle('RRR') == 'R'
4 triangle('RGBG') == 'B'
5 triangle('RBRGBRBR') == 'G'
6 triangle('RBRGBRBRGRRRBGBBBGG') == '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?}

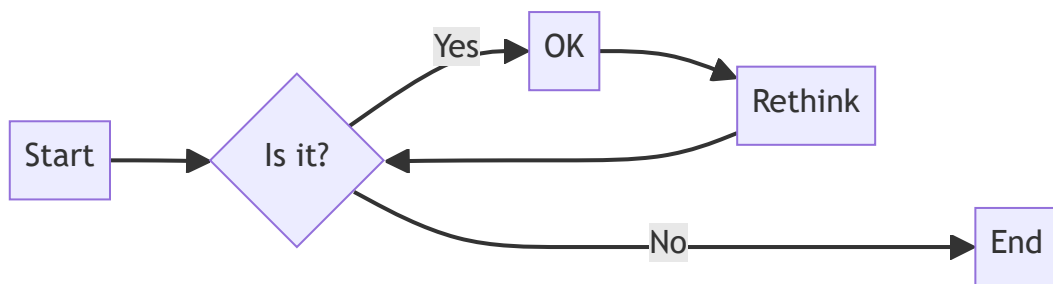
B --> |Yes| C[OK]

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：

```
```bat
git init
git add .
git status
git commit -m "first commit"
```
```

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，应该使用下面代码块格式，例如：

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

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，因为Markdown文档转换为Pdf格式后，截图会无法显示。**

## 第二部分 Codewars Kata挑战

### 第一题：停止逆转我的单词

```
1  def spin_words(sentence):
2      # Your code goes here
3      result=[]
4      spinWords=sentence.split()
5      for i in spinWords:
6          if len(i)>=5 :
7              result.append(i[::-1])
8          else:
```

```
9         result.append(i)
10     result=" ".join(result)
11     return result
```

split函数通过指定分隔符对字符串进行切片，该题用的是空字符串来切片。  
i[::-1]表示将i的值进行反转。

## 第二题：发现离群的数

```
1 def find_outlier(integers):
2     j=[]
3     o=[]
4     for i in integers:
5         if i%2==0:
6             j.append(i)
7         else:
8             o.append(i)
9     if len(j)==1:
10        return j[0]
11    if len(o)==1:
12        return o[0]
```

## 第三题：检测Pangram

```
1 def is_pangram(s):
2     s=s.lower()
3     for c in 'abcdefghijklmnopqrstuvwxyz':
4         if c in s:
5             i=1
6         else:
7             return False
8     return True
```

lower()函数是让字符串的字母都变为小写。

## 第四题：数独解决方案验证

```
1 def validate_sudoku(board):
2     elements=set(range(1,10))
3     for b in board:
4         if set(b)!=elements:
5             return False
6     for b in zip(*board):
7         if set(b)!=elements:
8             return False
9     for i in range(0,7,3):
10        for j in range(0,7,3):
```

```
11         if elements != {(board[w][q]) for q in range(j,j+3) for w in range(i,i+3)}:  
12             return False  
13     return True
```

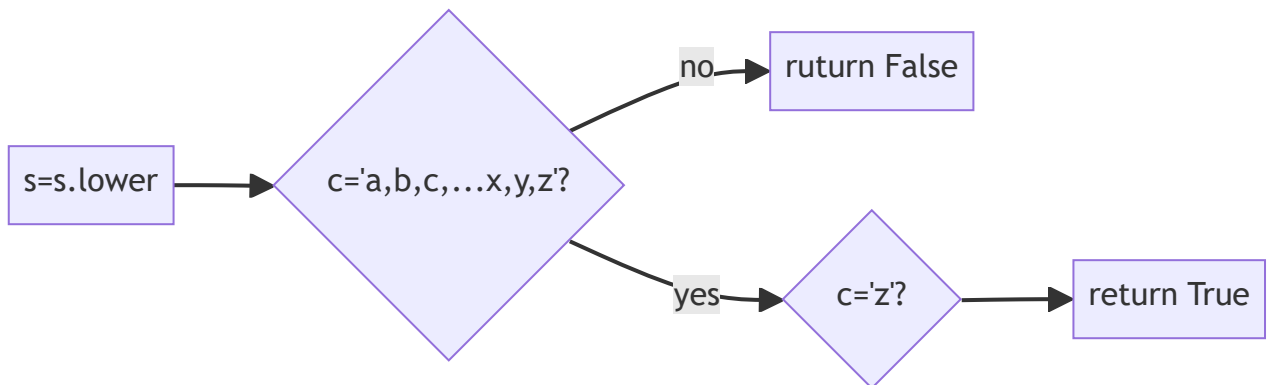
集合set是一个无序的不重复元素序列。  
zip函数是给矩阵转置。

### 第五题： 疯狂的彩色三角形

pass

## 第三部分 使用Mermaid绘制程序流程图

### 第三题： 检测Pangram



## 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set是一个无序的不重复元素序列,用 {} 创建，也可用 set() 函数。

区别是list是一个有序的，可重复的集合用 [] 来创建。

2. 集合（set）类型主要有那些操作？

集合支持常见的集合操作，如添加元素、删除元素、交集、并集、差集等。一些常见的方法包括 add(), remove(), union(), intersection(), difference() 等。

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。



使用 `*` 操作符可以复制列表中的元素，重复多次。例如，`[1, 2] * 3` 会生成 `[1, 2, 1, 2, 1, 2]`。不能直接使用 `*` 操作符作用到嵌套的列表上，因为 `*` 操作符只复制了列表中的元素的引用，而不是创建新的嵌套列表。修改一个嵌套列表中的元素会影响所有重复的元素。

```
1 # 使用 * 操作符重复列表中的元素
2 original_list = [1, 2]
3 repeated_list = original_list * 3
4 print(repeated_list) # 输出: [1, 2, 1, 2, 1, 2]
5
6 # 不能直接使用 * 操作符重复嵌套的列表
7 nested_list = [[1, 2]]
8 repeated_nested_list = nested_list * 3
9 repeated_nested_list[0][0] = 99
10 print(repeated_nested_list) # 输出: [[99, 2], [99, 2], [99, 2]]
```

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：用于生成新的列表，通过对原列表的每个元素应用某种操作或筛选条件来创建新的列表。

示例：

```
1 Copy code
2 original_list = [1, 2, 3, 4, 5]
3 squared_list = [x**2 for x in original_list] # 生成原列表每个元素的平方
```

集合解析：类似于列表解析，但用于创建新的集合，保证集合中的元素唯一。

示例：

```
1 Copy code
2 original_list = [1, 2, 2, 3, 3, 4, 5, 5]
3 unique_set = {x for x in original_list} # 创建包含原列表唯一元素的集合
```

字典解析：用于创建新的字典，通过对原字典的键值对应用某种操作或筛选条件来生成新的字典。

示例：

```
1 Copy code
2 original_dict = {'a': 1, 'b': 2, 'c': 3}
3 squared_dict = {key: value**2 for key, value in original_dict.items()} # 创建新字典，值为原字典值的平方
```

## 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。

本次实验我学习了set集合和它的函数的使用，还了解了lower(),zip(),split()等函数。对字典和列表也有了更深入的理解。越来越明白列表，集合，字典解析的含义，以及其使用方法了。总的来说，通过这次实验，我的python水平有了一定的提高。