

# 实验五 Python数据结构与数据模型

---

班级： 21计科3

学号： B20230302320

姓名： 彭钰淇

Github地址： [ReSakura01/PythonCourse: MyPythonCourse \(github.com\)](https://github.com/ReSakura01/PythonCourse)

CodeWars地址： [ReSakura | Codewars](https://www.codewars.com/users/ReSakura)

## 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?}
```

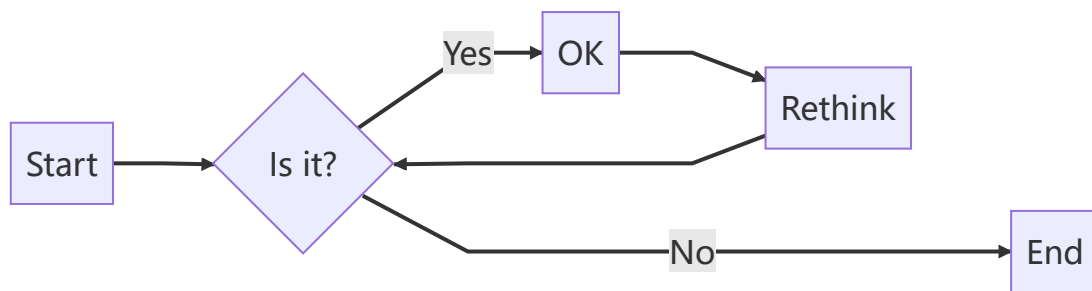
```
B -->|Yes| C[OK]
```

```
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：

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

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，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（set）的特点：无序性。唯一性。可变性。

与列表（list）的区别：列表有序，集合无序。列表允许重复元素，集合不允许。

集合性能更高效用于查找元素是否存在。列表是有序的可变序列，集合是无序的可变集合。

2. 集合（set）类型主要有那些操作？

添加元素：使用add和update方法。

移除元素：使用remove、discard、pop和clear方法。

大小和成员检查：使用len来获取集合大小，使用in来检查元素是否存在。

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\*操作符作用于列表时，会将列表中的元素重复指定的次数。这操作被称为列表的重复。这个操作符不会直接应用于嵌套的列表，因为它只重复顶层的元素，而不会递归地应用于嵌套的子列表。

```
# 列表的重复操作
original_list = [1, 2, 3]
repeated_list = original_list * 3 # 重复3次
print(repeated_list)
# 输出: [1, 2, 3, 1, 2, 3, 1, 2, 3]

# 嵌套列表的重复操作
nested_list = [[1, 2], [3, 4]]
repeated_nested_list = nested_list * 2 # 重复2次
print(repeated_nested_list)
# 输出: [[1, 2], [3, 4], [1, 2], [3, 4]]
```

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 (List Comprehension) :

- 用于创建新的列表，通过迭代、过滤和转换现有的可迭代对象。
- 语法: `[expression for item in iterable if condition]`

```
# 创建一个包含1到5的平方的列表
squares = [x**2 for x in range(1, 6)]
print(squares) # 输出: [1, 4, 9, 16, 25]
```

集合解析 (Set Comprehension) :

- 用于创建新的集合，通过迭代、过滤和转换现有的可迭代对象。
- 语法: `{expression for item in iterable if condition}`

```
# 创建一个包含1到10之间偶数的集合
evens = {x for x in range(1, 11) if x % 2 == 0}
print(evens) # 输出: {2, 4, 6, 8, 10}
```

字典解析 (Dictionary Comprehension) :

- 用于创建新的字典，通过迭代、过滤和转换现有的可迭代对象的键值对。
- 语法: `{key_expression: value_expression for item in iterable if condition}`

```
# 创建一个字典，将1到5的数字映射到它们的平方
squares_dict = {x: x**2 for x in range(1, 6)}
print(squares_dict) # 输出: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
```

## 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。

能够熟练地使用python的一些基本数据结构了，比如set, dictionary.....