

实验五 Python数据结构与数据模型

班级： 21计科3

学号： B20210302304

姓名： 潘振宇

Github地址： <https://github.com>

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

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：

```
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：

 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 Codewars Kata挑战](#)

第一题

```
def spin_words(sentence):
    words=sentence.split()

    new_words=[word[::-1] if len(word)>=5 else word for word in words]

    result=" ".join(new_words)
    return result
```

第二题

```
def find_outlier(integers):
    odd_count=0
    even_count=0

    last_odd=None
    last_even=None

    for i in integers:
        if i%2==0:
            even_count+=1
            last_even=i
        else:
            odd_count+=1
            last_odd=i

    if odd_count>1 and even_count==1:
        return last_even
    if even_count>1 and odd_count==1:
        return last_odd
```

第三题

```
def is_pangram(s):
    s=s.lower()

    for c in 'abcdefghijklmnopqrstuvwxyz':
        if c not in s:
            return False

    return True
```

第四题

```
def validate_sudoku(board):
    elements = set(range(1, 10))

    for b in board:
        if set(b) != elements:
            return False

    for b in zip(*board):
        if set(b) != elements:
            return False

    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：

```

def triangle(row):
    while len(row) > 1:
        new_row = ""
        for i in range(len(row) - 1):
            new_row += get_missing_color(row[i], row[i + 1])
        row = new_row
    return row

def get_missing_color(color1, color2):
    if color1 == color2:
        return color1
    colors = {'R', 'G', 'B'}
    colors.remove(color1)
    colors.remove(color2)
    return colors.pop()

```

- [第二部分 使用Mermaid绘制程序流程图](#)

第二题

```

flowchart LR
    A[Start] --> B{i in integers?}
    B -->|Yes| C{i%2==0?}
    C -->|Yes| D[even_count+=1,last_even=i]
    C -->|No| F[odd_count+=1,last_odd=i]
    D --> K{odd_count>1 and even_count==1?}
    F --> K
    K -->|Yes| M[return last_even]
    K -->|No| N{even_count>1 and odd_count==1?}
    N -->|Yes| S[return last_odd]
    N -->|No| B
    B -->|No| E[End]

```

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？

(1) 集合的特点：

- 唯一性：集合中的元素必须是唯一不重复的。
- 无序性：集合中的元素是无序的，即没有固定的顺序。
- 可变性：可以通过添加和删除元素等操作来改变集合。
- 集合支持各种数学集合操作，如并集、交集、差集等。

(2) 集合与列表的区别：

- 集合的元素无序且唯一，列表的元素有序且允许重复。
- 可以通过索引访问列表中的元素，而集合不可以。
- 集合支持集合操作，如并集和交集，而列表不提供这些操作。
- 集合通常用于需要唯一性并且不关心元素顺序的情况，而列表适用于有序的、允许重复的情况。

2. 集合 (set) 类型主要有那些操作？

(1) 添加元素：

- `add(item)`: 向集合中添加一个元素，如果该元素存在，则忽略。
- `update(iterable)`: 向集合中添加多个元素。

(2) 删除元素：

- `remove(item)`: 从集合中删除指定的元素，若元素不存在，则会引发`KeyError`异常。
- `pop()`: 从集合中随机删除一个元素，若集合为空，则会引发`KeyError`异常。
- `clear()`: 清空整个集合。

(3) 集合操作：

- `union(other_set)`: 返回一个两个集合所有元素的集合。
- `intersection(other_set)`: 返回两个集合的交集。

(4) 集合比较：

- `issubset(other_set)`: 检查一个集合是否是另一个集合的子集。
- `issuperset(other_set)`: 检查一个集合是否是另一个集合的超集。

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。

(1) * 操作符用于列表可以将列表中的元素重复多次，创建一个新的列表。

(2) * 操作符对于嵌套列表会导致意外结果，因为它只是复制嵌套列表引用，而没有复制列表的内容。

(3) 代码说明：

- * 操作符用于列表：

```
original_list = [1, 2, 3]
repeated_list = original_list * 3 # 重复列表元素三次
print(repeated_list)

#输出如下
[1, 2, 3, 1, 2, 3, 1, 2, 3]
```

- * 操作符用于嵌套列表:

```
nested_list = [[1, 2]] * 3
print(nested_list)
```

#输出如下

```
[[1, 2], [1, 2], [1, 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。

(1) 列表解析:

- 列表解析是一种非常简洁的语法, 用于创建新的列表。它允许通过在一行代码中使用表达式和循环来生成列表。
- 代码示例:

```
# 生成平方数列表
squares = [x**2 for x in range(5)]
print(squares)
# 输出: [0, 1, 4, 9, 16]

# 过滤奇数的平方数列表
odd_squares = [x**2 for x in range(5) if x % 2 != 0]
print(odd_squares)
# 输出: [1, 9]

# 使用条件表达式
result = [x if x % 2 == 0 else x*2 for x in range(5)]
print(result)
# 输出: [0, 2, 2, 6, 4]
```

(2) 集合解析:

- 集合解析的用法类似于列表解析, 但其输出为集合。
- 代码示例:

```
# 生成集合的平方
square_set = {x**2 for x in range(5)}
print(square_set)
# 输出: {0, 1, 4, 9, 16}

# 过滤奇数的平方
odd_square_set = {x**2 for x in range(5) if x % 2 != 0}
print(odd_square_set)
# 输出: {1, 9}
```


(3) 字典解析:

- 允许我们以紧凑的方式创建字典。
- 代码示例:

```
# 生成字典的平方
square_dict = {x: x**2 for x in range(5)}
print(square_dict)
# 输出: {0: 0, 1: 1, 2: 4, 3: 9, 4: 16}

# 过滤奇数的平方
odd_square_dict = {x: x**2 for x in range(5) if x % 2 != 0}
print(odd_square_dict)
# 输出: {1: 1, 3: 9}
```

实验总结

通过这次实验，我深入了解了Python集合的基本用法。我学会了如何利用集合这种无序且可变的容器类型，并掌握了集合的基本操作和简单应用。集合适用于存储唯一且无序的元素，这一点和字典不同，后者适用于存储键值对。这次学习让我更清楚地理解了不同数据结构的特点和用途，为我解决问题时的数据选择提供了更多思路。