



实验五 Python数据结构与数据模型

班级：21计科04

学号：B20210302404

姓名：刘卓涵

Github地址：<https://github.com/dawn1234ar/python-homework>

CodeWars地址：<https://www.codewars.com/users/dawn123>

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](https://www.codewars.com/)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 "RRGBRGBB"，你应该返回 "G"。

限制条件：1 <= length(row) <= 10 ** 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：

显示效果如下：

```
flowchart 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

1.

The screenshot shows the Codewars interface for the kata "Stop gninnipS My sdroW!". The title is highlighted in yellow. Below the title, it shows 3083 stars, 549 likes, 91% of 24,264 votes, 82,896 of 240,901 attempts, and 5 issues reported. The language is Python with a score of 3.11. The left sidebar shows the test results: "Test Results: Stop gninnipS My sdroW! Single word (3 of 3 Assertions) Multiple words (2 of 2 Assertions) Random testing (20 of 20 Assertions) Completed in 1.24ms". The main area shows the "Solution" tab with the following code:

```
7     if len(ret) < 5:
8         ans += ret
9     else:
10        ans += ''.join(reversed(ret))
11        ans += x
12        ret = ''
13    else:
14        ret += x
15
16    if len(ret) < 5:
17        ans += ret
18        ret = ''
19    else:
20        ans += ''.join(reversed(ret))
21
```

Below the solution, there is a "Sample Tests" section with the following code:

```
7     def _():
8         test.assert_equals(spin_words("Welcome"), "emocleW")
9         test.assert_equals(spin_words("to"), "to")
10        test.assert_equals(spin_words("CodeWars"), "sraWedeC")
11
12    @test.it("Multiple words")
13    def _():
14        test.assert_equals(spin_words("Hey fellow warriors"), "Hey wollef sroirraw")
15        test.assert_equals(spin_words("This sentence is a sentence"), "This ecnetnes is a ecnetnes")

```

At the bottom, there are buttons for "SKIP", "UNLOCK SOLUTIONS", "DISCUSS (379)", "RESET", "TEST", and "SUBMIT". A message at the bottom says "Good Job! You may take your time to refactor/comment your solution. Submit when ready."

```
''' def spin_words(sentence): # Your code goes here ans = " ret = " for x in sentence: if x == ' ':
if len(ret) < 5: ans += ret else: ans += ".join(reversed(ret)) ans += x ret = " else: ret += x
```

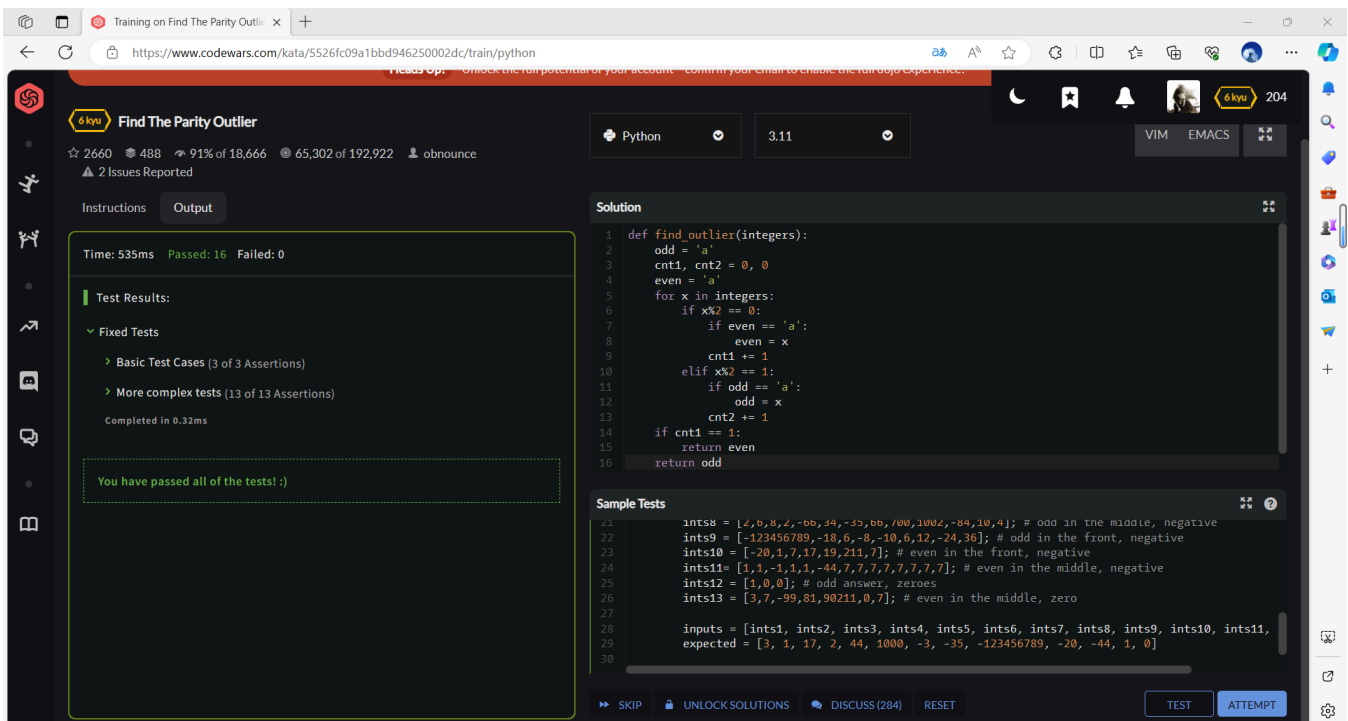
```
if len(ret) < 5:
    ans += ret
    ret = ''
else:
    ans += ''.join(reversed(ret))

return ans
```

2.

```
```py
```

```
def find_outlier(integers):
 odd = 'a'
 cnt1, cnt2 = 0, 0
 even = 'a'
 for x in integers:
 if x%2 == 0:
 if even == 'a':
 even = x
 cnt1 += 1
 elif x%2 == 1:
 if odd == 'a':
 odd = x
 cnt2 += 1
 if cnt1 == 1:
 return even
 return odd
```



Training on Find The Parity Outlier: x +

https://www.codewars.com/kata/5526fc09a1bbd946250002dc/train/python

Find The Parity Outlier

☆ 2660 488 91% of 18,666 65,302 of 192,922 obnounce

2 Issues Reported

Instructions Output

Time: 535ms Passed: 16 Failed: 0

Test Results:

Fixed Tests

- Basic Test Cases (3 of 3 Assertions)
- More complex tests (13 of 13 Assertions)

Completed in 0.32ms

You have passed all of the tests! :)

Solution

```
1 def find_outlier(integers):
2 odd = 'a'
3 cnt1, cnt2 = 0, 0
4 even = 'a'
5 for x in integers:
6 if x%2 == 0:
7 if even == 'a':
8 even = x
9 cnt1 += 1
10 elif x%2 == 1:
11 if odd == 'a':
12 odd = x
13 cnt2 += 1
14 if cnt1 == 1:
15 return even
16 return odd
```

Sample Tests

```
21 ints8 = [2,0,8,2,-66,34,-35,66,700,1002,-84,10,4]; # odd in the middle, negative
22 ints9 = [-123456789,-18,6,-8,-10,6,12,-24,36]; # odd in the front, negative
23 ints10 = [-20,1,7,17,19,211,7]; # even in the front, negative
24 ints11 = [1,1,-1,1,1,-44,7,7,7,7,7,7]; # even in the middle, negative
25 ints12 = [1,0,0]; # odd answer, zeroes
26 ints13 = [3,7,-99,81,90211,0,7]; # even in the middle, zero
27
28 inputs = [ints1, ints2, ints3, ints4, ints5, ints6, ints7, ints8, ints9, ints10, ints11,
29 ints12, ints13]
30 expected = [3, 1, 17, 2, 44, 1000, -3, -35, -123456789, -20, -44, 1, 0]
```

SKIP UNLOCK SOLUTIONS DISCUSS (284) RESET TEST ATTEMPT

3.

Training on Detect Pangram | Co... x +

https://www.codewars.com/kata/545cedaa9943f7fe7b000048/train/python

6 kyu Detect Pangram

☆ 2243 🗳 531 📈 92% of 11,963 🌐 60,684 of 153,785 👤 anindyabd

4 Issues Reported

Python 3.11

VIM EMACS

5 kyu 265

Instructions Output

Time: 531ms Passed: 8 Failed: 0

Test Results:

✓ Fixed tests

- Test pangrams (5 of 5 Assertions)
- Test non-pangrams (3 of 3 Assertions)

Completed in 0.12ms

You have passed all of the tests! :)

Solution

```
1 def is_pangram(s):
2 s = s.lower()
3 for x in 'qwertyuiopasdfghjklzxcvbnm':
4 if x not in s:
5 return False
6
7 return True
```

Outstanding! You may take your time to refactor/comment your solution. Submit when ready.

Sample Tests

```
1 from solution import is_pangram
2 import codewars_test as test
3
4 @test.describe("sample tests")
5 def sample_tests():
6
7 @test.it("Should return true for a pangram")
8 def test_pangram():
9 test.assert_equals(is_pangram("The quick, brown fox jumps over the lazy dog!"), True)
10
```

SKIP UNLOCK SOLUTIONS DISCUSS (235) RESET TEST SUBMIT

```
py def is_pangram(s): s = s.lower() for x in 'qwertyuiopasdfghjklzxcvbnm': if x not in s: return False
```

```
return True
```



4.

```
```py
```

```
def validate_sudoku(board):
    ret = [[board[x + a][y + b] for a in range(0, 3) for b in range(0, 3)] for x in range(0, 9, 3)]
    l = [1, 2, 3, 4, 5, 6, 7, 8, 9]
    for arr in ret:
        arr.sort()
        if arr != l:
            return False
    xz = [[board[x][y] for x in range(0, 9)] for y in range(0, 9)]
    for arr in xz:
        arr.sort()
        if arr != l:
            return False
    zx = [[board[y][x] for x in range(0, 9)] for y in range(0, 9)]
    for arr in zx:
        arr.sort()
        if arr != l:
            return False
    return True
```

The screenshot shows a web browser window displaying a Codewars kata page titled "Sudoku board validator". The page has a dark theme. At the top, there's a navigation bar with a "Heads Up!" message and a user profile section showing "5 kyu" and "265" points. Below the navigation bar, the kata title "Sudoku board validator" is displayed along with statistics: 138 stars, 36 solved, 94% of 252 users solved it, and 865 of 1,834 users reported an issue. The page is divided into three main sections. The left section, titled "Test Results", shows "Time: 496ms", "Passed: 36", and "Failed: 0". It also indicates "Fixed tests (36 of 36 Assertions)" and "Completed in 2.55ms". A green box contains the message "You have passed all of the tests! :)". The middle section, titled "Solution", displays a Python code snippet for the `validate_sudoku` function, which is identical to the code provided in the first block. The right section, titled "Sample Tests", shows a test suite for the function. At the bottom of the page, there are buttons for "SKIP", "UNLOCK SOLUTIONS", "DISCUSS (25)", "RESET", "TEST", and "ATTEMPT".

5. ```

- [第二部分 使用Mermaid绘制程序流程图](#第二部分)

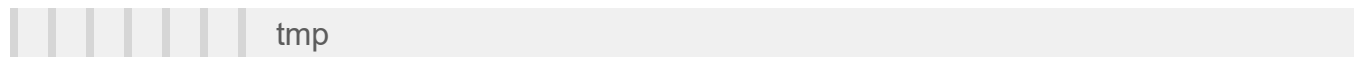
注意代码需要使用markdown的代码块格式化，例如Git命令行语句应该使用下面的格式：

```
```mermaid
flowchart LR
 A[长字符串sentence, 空列表 ans] --> B[将sentence拆分为单词, i = 0]
 B --> C{第i个单词长度大于5}
 C --> |YES| D[将第i个单词翻转加入ans中]
 D --> E{是否是最后一个单词}
 C ----> |No| E[将第i个单词直接加入ans中]
 E --> |YES| F[返回ans]
 E --> |NO| G[i++, ans尾部加空格]
 G --> C
```
```

<<<<<< HEAD

Alt text

Alt text



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：

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？

列表 (list) 是一个有序的数据集合，可以使用索引来访问元素，并且可以修改列表中的元素。

集合 (set) 是一个无序的不重复数据集合，不能使用索引来访问元素，并且可以进行集合运算（并集，交集等）

2. 集合（set）类型主要有那些操作？

- 1、创建集合
- 2、去重
- 3、集合元素的增、删以及判断元素是否在集合中存在
 - 3.1> 增加元素
 - 3.2> 删除元素
- 4、判断元素是否在集合中存在
- 5、计算集合元素个数

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。

序列数据类型（可变和不可变）支持重复操作符。* 重复操作符 * 创建该对象的多个副本并将它们连接在一起。当与整数一起使用时，* 执行乘法，但当与列表、元组或字符串一起使用时，它执行重复操作。

```
inputList = [5, 6, 7]

print(inputList * 2)
```

```
[5, 6, 7, 5, 6, 7]
```

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：通过解析表达式从一个可迭代对象生成一个新的列表

集合解析：通过解析表达式从一个可迭代对象生成一个新的集合

字典解析：通过解析表达式从一个可迭代对象生成一个新的字典

```
l=[chr(i+ord('A')) for i in range(26) if i%2==0 if i%3==0 if i%4==0 ]#结果为: ['A', 'M', 'Y']  
  
d={i:chr(ord('a')+i) for i in [1,1,2,3]} #结果d的值为: {1: 'b', 2: 'c', 3: 'd'}
```

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。

通过本次实验，我对列表的使用有了一定的了解，并对python语法更加熟练。