实验三 Python列表 班级: 21计科1

学号: B20210302112

姓名: 张梽棋

Github地址: https://github.com/yourusername/python\_course

实验目的 学习Python的简单使用和列表操作 学习Python中的if语句 实验环境 Git Python 3.10 VSCode VSCode 插件 实验内容和步骤 第一部分 Python列表操作

完成教材《Python编程从入门到实践》下列章节的练习:

第3章 列表简介 第4章 操作列表 第5章 if语句 第二部分 在Codewars网站注册账号,完成下列Kata挑战:

第一题: 3和5的倍数 (Multiples of 3 or 5) 难度: 6kyu

如果我们列出所有低于 10 的 3 或 5 倍数的自然数,我们得到 3、5、6 和 9。这些数的总和为 23. 完成一个函数,使其返回小于某个整数的所有是3 或 5 的倍数的数的总和。此外,如果数字为负数,则返回 0。

注意: 如果一个数同时是3和5的倍数,应该只被算一次。

代码: def min\_common\_multiple(x, y): # 判断最大值 if x > y: max\_value = x else: max\_value = y

```
while max_value <= x * y:
    if max_value % x == 0 and max_value % y == 0:
        break
    max_value += 1
return max_value</pre>
```

print(min\_common\_multiple(10, 18)) print(min\_common\_multiple(11, 13))

第二题: 重复字符的编码器 (Duplicate Encoder) 难度: 6kyu

本练习的目的是将一个字符串转换为一个新的字符串,如果新字符串中的每个字符在原字符串中只出现一次,则为"(",如果该字符在原字符串中出现多次,则为")"。在判断一个字符是否是重复的时候,请忽略大写字母。

例如:

```
"din" => "(((" "recede" => "()()()" "Success" => ")())())" "(( @" => "))((" 代码: function duplicate_encode($word){ $word = str_split(strtolower($word)); $str = ""; foreach($word as $key){ (count(array_keys($word,$key))>1) ? $str .= ")" : $str .= "("; } return $str; }
```

第三题: 括号匹配 (Valid Braces) 难度: 6kyu

写一个函数,接收一串括号,并确定括号的顺序是否有效。如果字符串是有效的,它应该返回True,如果是无效的,它应该返回False。 例如:

```
"(){}[]" => True "([{}])" => True "(}" => False "[(])" => False "[({})](]" => False
```

代码: def is\_valid(s: str) -> bool: stack = [] map = {")": "(", "}": "{", "]": "["} for char in s: if char in map: top = stack.pop() if stack else '#' if map[char] != top: return False else: stack.append(char) return not stack

第四题: 从随机三元组中恢复秘密字符串(Recover a secret string from random triplets) 难度: 4kyu

有一个不为你所知的秘密字符串。给出一个随机三个字母的组合的集合,恢复原来的字符串。

这里的三个字母的组合被定义为三个字母的序列,每个字母在给定的字符串中出现在下一个字母之前。"whi "是字符串 "whatisup "的一个三个字母的组合。

作为一种简化,你可以假设没有一个字母在秘密字符串中出现超过一次。

对于给你的三个字母的组合,除了它们是有效的三个字母的组合以及它们包含足够的信息来推导出原始字符串之外,你可以不做任何假设。特别是,这意味着秘密字符串永远不会包含不出现在给你的三个字母的组合中的字母。

### 测试用例:

secret = "whatisup" triplets = [ ['t','u','p'], ['w','h','i'], ['t','s','u'], ['a','t','s'], ['h','a','p'], ['t','i','s'], ['w','h','s'] ] test.assert\_equals(recoverSecret(triplets), secret) 代码: def recoverSecret(triplets):  $r = list(\{i \text{ for } t \text{ in } triplets \text{ for } t \text{ in } t \text{ for } t \text{ for } t \text{ in } t \text{ for } t \text{ in } t \text{ for } t \text$ 

```
def fix(t, a, b):
    if t.index(a) > t.index(b):
        t.remove(a)
        t.insert(t.index(b), a)
```

第五题: 去掉喷子的元音 (Disemvowel Trolls) 难度: 7kyu

喷子正在攻击你的评论区! 处理这种情况的一个常见方法是删除喷子评论中的所有元音(字母: a,e,i,o,u),以消除威胁。 你的任务是写一个函数,接收一个字符串并返回一个去除所有元音的新字符串。 例如,字符串 "This website is for losers LOL!" 将变成 "Ths wbst s fr lsrs LL!". 注意: 对于这个Kata来说,y不被认为是元音。 代码: last\_name="This website is for losers LOL!" letters = list(last\_name.lower()) print(letters) result =[ c for c in last\_name.lower() if c not in 'aeiou'] print(result) print(''.join(result))

### 第三部分 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:

#### 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 Python列表操作和if语句 第二部分 Codewars Kata挑战 第三部分 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:

### Git命令

## 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代码

## 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, Markdown文档转换为Pdf格式后, 截图可能会无法显示。

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。

Python中的列表可以进行哪些操作? 1: 创建列表 2: 访问列表 3: 列表统计 4: 列表更新 5: 列表排序

哪两种方法可以用来对Python的列表排序?这两种方法有和区别?

Python中拥有内置函数实现排序,可以直接调用它们实现排序功能

Python列表有一个内置的list.sort()方法可以直接修改列表。还有一个 sorted() 内置函数,它会从一个可迭代对象构建一个新的排序列表。 sort()是应用在 list上的方法,sorted()可以对所有可迭代的对象进行排序操作。 list 的 sort() 方法返回的是对已经存在的列表进行操作,无返回值,而内建函数 sorted() 方法返回的是一个新的 list,而不是在原来的基础上进行的操作

如何将Python列表逆序打印? 方法一: 使用reverse()方法 方法二: 使用切片 (slice) 语法

Python中的列表执行哪些操作时效率比较高?哪些操作效率比较差?是否有类似的数据结构可以用来替代列表?效率高:在列表末尾添加元素,使用 append()方法;在列表末尾添加多个元素,使用 extend()方法;效率低:插入或删除元素:由于列表是可变的,插入或删除元素时需要移动其他元素,因此效率较低。查找元素:由于列表是线性结构,查找元素时需要遍历整个列表,效率也较低。

实验总结 我学会了Python列表的打印操作,利用集合去掉重复字母,去掉一个字符串里面的元音字母,还有列表的一些排序,删除,增加的操作,更加深刻的理解了Python代码的使用规则。