

# 实验五 Python数据结构与数据模型

---

班级： 21计科03

学号： B20210302321

姓名： 向钟源

Github地址： [Ch1rs\(github.com\)](https://github.com/Ch1rs)

CodeWars地址： [Ch1rs\\_X | Codewars](https://www.codewars.com/kata/5264d2b162488dc400000001)

---

## 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('RBRGBRBGRRRBGBBBGG') == '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

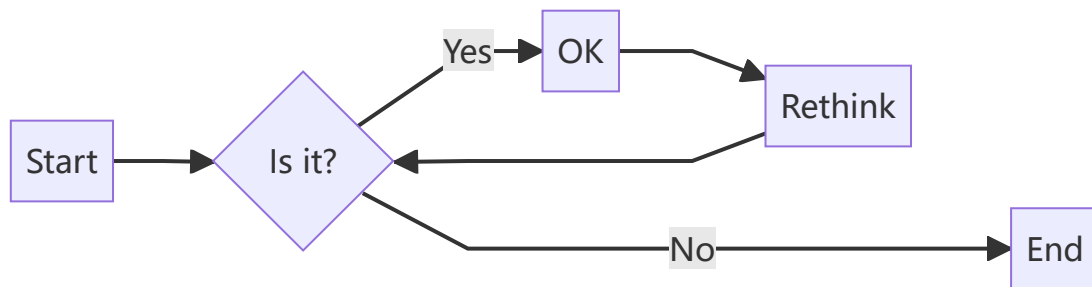
使用Mermaid绘制程序流程图

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：



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：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：

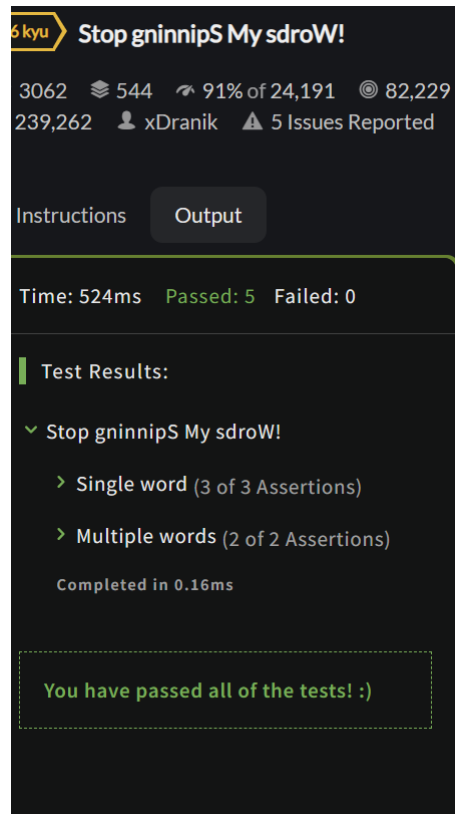
```
words = "hey fellow warrior".split()
# words should be ['hey', 'fellow', 'warrior']
```

- 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。

```
def spin_words(sentence):
    # Your code goes here
    words = sentence.split()

    for i in range(len(words)):
        if len(words[i]) >= 5:
            words[i] = words[i][::-1]

    return ' '.join(words)
```



## 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)
```

solution:

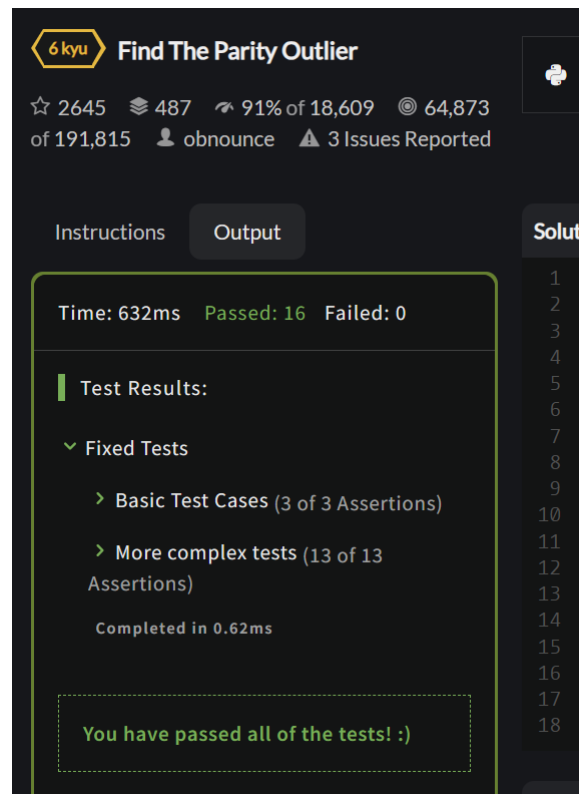
```
def find_outlier(integers):
    even_count = 0
    odd_count = 0
    last_even = 0
    last_odd = 0
```

```

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

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

```



### 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。

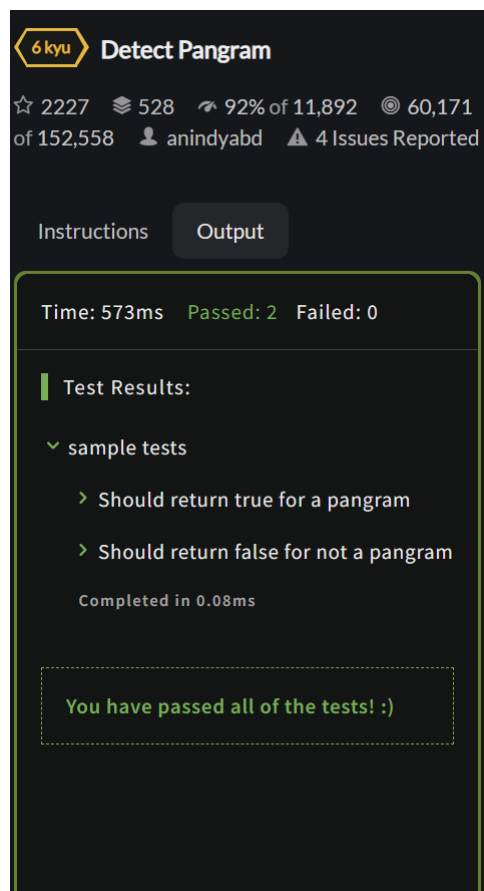
```

import string

def is_pangram(sentence):
    # 将句子中的字母转为小写，并去除标点和空格
    sentence = sentence.lower()
    sentence = ''.join(filter(str.isalpha, sentence))

    # 利用 set 对字母去重，检查长度是否为 26（表示包含了 A-Z 所有字母）
    return len(set(sentence)) == 26

```



#### 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。游戏的目的是用 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>

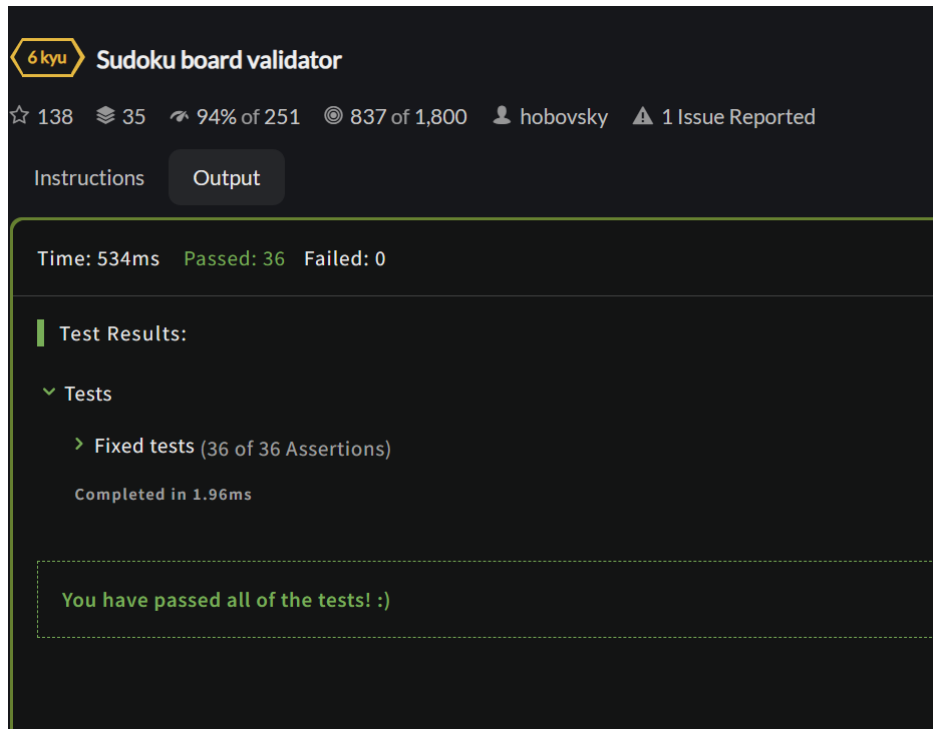
编写一个函数接受一个代表数独板的二维数组，如果它是一个有效的解决方案则返回 true，否则返回 false。数独板的单元格也可能包含 0，这将代表空单元格。包含一个或多个零的棋盘被认为是无效的解决方案。棋盘总是 9 x 9 格，每个格只包含 0 到 9 之间的整数。

solution:

```
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
```



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'

```

solution:

```

def triangle(row):
    if len(row) == 1:
        return row

    new_row = []
    for i in range(len(row) - 1):
        if row[i] == row[i+1]:
            new_row.append(row[i])
        else:
            colors = set('RGB')
            colors.difference_update({row[i], row[i+1]})
            new_row.append(list(colors)[0])

    return triangle(''.join(new_row))

```

2 kyu

Insane Coloured Triangles

☆ 971

🔖 159

👤 95% of 614

🕒 1,713 of 3,316

👤 Bubbler

🚩 1 Issue Reported

Instructions

Output

Time: 480ms Passed: 6 Failed: 0

Test Results:

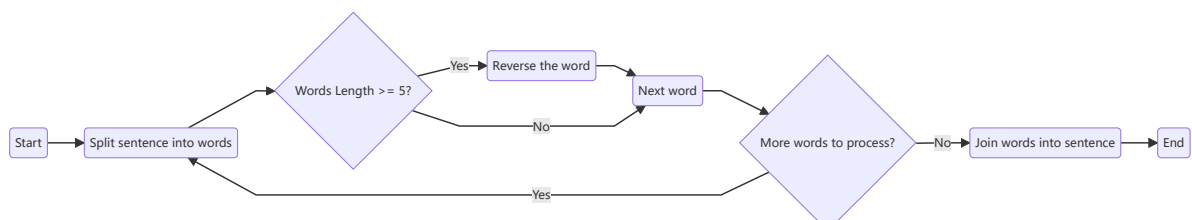
▼ Insane Coloured Triangles

► Basic Tests (6 of 6 Assertions)

You have passed all of the tests! :)

- [第二部分 使用Mermaid绘制程序流程图](#)

第一题的流程图:



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？

**答：集合（Set）和列表（List）是Python中常用的数据类型，它们有几个重要的区别：**

1. **有序性：**列表是有序的数据结构，元素按照插入顺序存储并保持顺序。而集合是无序的，元素存储顺序并不重要，且集合不支持索引。在集合中，无法通过索引来获取元素。
2. **唯一性：**集合中的元素是唯一的，不允许重复值。当你试图向集合中添加已经存在的元素时，集合不会报错，但不会改变。列表可以包含重复的元素。

3. **可变性**：列表是可变的数据结构，可以改变、添加或删除列表中的元素。集合是可变的，可以向集合添加或删除元素，但集合中的元素本身是不可变的。也就是说，无法改变集合中的元素。

2. 集合 (set) 类型主要有那些操作？

答：

1. **增加元素**：

- `add(element)`：向集合中添加一个元素。
- `update(iterable)`：将一个可迭代对象中的元素添加到集合中。

2. **移除元素**：

- `remove(element)`：移除集合中指定的元素，如果元素不存在，会抛出 `KeyError` 错误。
- `discard(element)`：移除集合中指定的元素，如果元素不存在，不会抛出错误。
- `pop()`：移除并返回集合中的任意一个元素。

3. **基本集合操作**：

- `union(set)`：返回两个集合的并集。
- `intersection(set)`：返回两个集合的交集。
- `difference(set)`：返回一个集合中存在，但另一个集合中不存在的元素。
- `symmetric_difference(set)`：返回两个集合中各自独有的元素的集合。
- `issubset(set)`：判断一个集合是否是另一个集合的子集。
- `issuperset(set)`：判断一个集合是否是另一个集合的超集。

4. **其他操作**：

- `clear()`：清空集合中的所有元素。
- `copy()`：复制集合。
- `len(set)`：返回集合中元素的个数。

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。

答：在Python中，`*` 操作符用于列表时，会复制列表中的元素，生成一个新的列表。这意味着将列表乘以一个整数时，会重复该列表中的元素。例如，`[1, 2, 3] * 3` 会生成 `[1, 2, 3, 1, 2, 3, 1, 2, 3]`。

但是，如果尝试在嵌套列表上使用 `*` 操作符，会导致意外的结果。这是因为 `*` 操作符仅仅复制了嵌套列表中的引用，而不是复制其内容。因此，在修改其中一个嵌套列表时，其他嵌套列表也会被修改，因为它们共享相同的引用。

```
# 示例 1: * 操作符应用在列表上
original_list = [1, 2, 3]
new_list = original_list * 3
print(new_list) # 输出: [1, 2, 3, 1, 2, 3, 1, 2, 3]

# 示例 2: * 操作符应用在嵌套的列表上
nested_list = [[0, 1]] * 3
print(nested_list) # 输出: [[0, 1], [0, 1], [0, 1]]

# 修改第一个嵌套列表的元素
nested_list[0][1] = 5
print(nested_list) # 输出: [[0, 5], [0, 1], [0, 1]]
```

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。

答:

#### 列表解析 (List Comprehensions) :

列表解析允许通过一种紧凑的方式创建新的列表。基本语法是使用方括号来包裹一个表达式, 后面跟随一个 for 循环。

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

#### 集合解析 (Set Comprehensions) :

集合解析允许以类似的方式创建新的集合, 使用花括号来包裹表达式。

```
# 创建一个包含平方数的集合
squares_set = {x**2 for x in range(1, 6)}
print(squares_set) # 输出: {1, 4, 9, 16, 25}
```

#### 字典解析 (Dictionary Comprehensions) :

字典解析允许你通过一个键值对表达式来创建新的字典。

```
# 创建一个将数字映射到其平方的字典
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}
```

```
# 列表解析中的条件语句示例
even_squares = [x**2 for x in range(1, 6) if x % 2 == 0]
print(even_squares) # 输出: [4, 16]
```

## 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。

答: 无论是列表、集合还是字典解析, 它们都提供了一种简明扼要的方法来构建这些数据结构, 有助于简化代码并提高可读性。