

一、字符串相关练习题

1. 字符串的基本概念与索引切片

- 定义字符串 `s = "I love Python"`，分别打印该字符串的第3个字符、第7个字符、倒数第2个字符。
- 对字符串 `s = "Data Science"` 进行切片操作，获取子串 `"Science"`。
- 定义字符串 `text = "abcdefghij"`，使用切片获取从第2个字符到第6个字符的子串（包含第2和第6个字符）。
- 定义字符串 `str_example = "HelloWorld"`，获取从第2个字符开始，步长为2的子串并打印。
- 对于字符串 `s = "Programming"`，打印其前5个字符组成的子串。

2. 字符串常用操作与方法

- 定义字符串 `str1 = "good"`，`str2 = "morning"`，将它们拼接成一个新字符串 `"good morning"` 并打印。
- 统计字符串 `s = "She sells seashells by the seashore."` 中字母 `'s'` 出现的次数（不区分大小写）。
- 将字符串 `s = "python is easy"` 中的所有字母转换为大写形式并打印。
- 去除字符串 `s = " Hello Python "` 两端的空格并打印结果。
- 查找字符串 `s = "Python Programming"` 中子串 `"Program"` 第一次出现的索引位置并打印。
- 把字符串 `s = "12345"` 转换为整数类型，再转换回字符串类型并打印，观察结果。

3. 字符串不可变性与转义字符

- 尝试修改字符串 `s = "example"` 的第4个字符为 `'a'`，观察会出现什么错误，并解释原因。
- 打印包含单引号的字符串 `'He said: "It\'s a good day."'`，使用转义字符实现。
- 打印如下格式的字符串：要求使用转义字符实现换行。

```
Title: Python Basics
Author: John
```

- 解释字符串不可变性的含义，并举例说明（比如尝试对字符串进行修改操作）。
- 打印字符串 `s = "C:\\Users\\Desktop\\file.txt"`，体会转义字符在文件路径中的使用。

二、列表和集合相关练习题

1. 列表 (list)

- 创建一个包含字符串 `"apple"`、`"banana"`、`"orange"` 的列表，然后在列表末尾添加字符串 `"grape"` 并打印列表。
- 创建列表 `nums = [8, 2, 5, 2, 7, 3]`，删除列表中值为 `2` 的第一个元素，并打印修改后的列表。
- 对列表 `nums = [9, 4, 6, 1, 3]` 进行排序，使其按升序排列并打印。
- 创建列表 `colors = ["red", "green", "blue"]`，在列表索引为1的位置插入字符串 `"yellow"` 并打印列表。

- 复制列表 `original = [10, 20, 30]` 到新列表 `new_list`，然后修改 `new_list` 的第一个元素为 `5`，分别打印 `original` 和 `new_list`，观察变化。
- 统计列表 `nums = [1, 3, 5, 3, 7, 3]` 中元素 `3` 出现的次数。

2. 集合 (set)

- 创建两个集合 `set1 = {1, 3, 5, 7}`，`set2 = {5, 7, 9, 11}`，求它们的交集并打印。
- 创建集合 `set3 = {2, 4, 6}`，向集合中添加元素 `8` 并打印集合。
- 创建集合 `set4 = {12, 14, 16, 18}`，删除集合中的元素 `16` 并打印集合。
- 创建集合 `set5 = {1, 2, 3}`，`set6 = {3, 4, 5}`，求它们的并集并打印。
- 创建集合 `set7 = {10, 20, 30}`，判断元素 `20` 是否在集合中，打印判断结果。
- 解释集合中元素的特性（唯一性、无序性等），并通过示例说明（比如创建包含重复元素的列表，转换为集合后观察结果）。

3. 列表与集合的对比及转换

- 详细说明列表和集合的主要区别（从元素是否可重复、是否有序、可进行的操作等方面）。
- 将列表 `list1 = [4, 5, 5, 6, 6, 6]` 转换为集合，观察结果并打印。
- 将集合 `set8 = {20, 30, 40}` 转换为列表并打印。
- 有列表 `list2 = [1, 2, 3, 4]` 和集合 `set9 = {3, 4, 5, 6}`，求列表和集合的交集（可先将列表转换为集合再操作）并打印。
- 尝试向列表和集合中分别添加重复元素，观察两者的不同表现并解释原因。

三、字典和元组相关练习题

1. 字典 (dict)

- 创建一个字典，包含键值对：`"name": "Bob"`，`"age": 30`，`"city": "Shanghai"`，然后打印该字典。
- 向上述字典中添加键值对 `"job": "engineer"` 并打印字典。
- 获取上述字典中键为 `"city"` 对应的值并打印。
- 创建字典 `dict_scores = {"math": 90, "english": 85, "science": 95}`，修改键 `"english"` 对应的值为 `88` 并打印字典。
- 删除字典 `dict_scores` 中键为 `"math"` 的键值对，打印删除后的字典。
- 遍历字典 `dict_example = {"a": 1, "b": 2, "c": 3}`，打印所有的键和值。

2. 元组 (tuple)

- 创建一个元组 `tup = (6, 7, 8, 9, 10)`，尝试修改元组的第 4 个元素为 `15`，观察会出现什么错误，并解释原因。
- 创建元组 `tup1 = (20, 30)`，`tup2 = (40, 50)`，将它们连接成一个新元组并打印。
- 创建元组 `tup3 = (7, 14, 21, 28)`，获取元组中第 3 个到第 4 个元素（包含第 3 和第 4 个元素）组成的新元组并打印。
- 定义元组 `tup4 = (1, 2, 3)`，打印元组的长度。
- 有元组 `tup5 = (5, 10, 15)`，判断元素 `10` 是否在元组中，打印判断结果。
- 解释元组不可变性的含义，与字符串不可变性进行对比说明。

3. 字典与元组的对比及转换

- 详细说明字典和元组的主要区别（从元素是否可修改、是否有序、存储形式等方面）。

2. 将字典 `dict2 = {"x": 10, "y": 20}` 转换为包含其键值对的元组列表（如 `[("x", 10), ("y", 20)]`）并打印。
3. 尝试将元组 `tup6 = (("m", 100), ("n", 200))` 转换为字典并打印。
4. 有字典 `dict3 = {"a": 1, "b": 2}` 和元组 `tup7 = ("a", "b", "c")`，遍历元组，若元素是字典的键，则打印该键对应的值。
5. 举例说明在什么场景下适合使用字典，什么场景下适合使用元组。