

Day6-Python 列表操作与练习

一、学习目标

掌握列表的创建、索引和切片，综合运用完成数据筛选与排序任务，掌握列表的常用方法 (append/insert/remove/pop/sort/reverse/count/index)

二、知识点讲解

1. 列表的创建、索引和切片

- 创建：使用方括号 [] 创建列表
- 索引：访问单个元素（从 0 开始）
- 切片：访问部分内容，语法：[开始:结束:步长]

```
# 创建一个包含 5 个整数的列表
numbers = [1, 2, 3, 4, 5]
print(numbers) # 输出整个列表
```

```
# 访问第一个元素（索引 0）
print(numbers[0]) # 第一个元素
```

```
# 访问最后一个元素（索引 -1）
print(numbers[-1]) # 最后一个元素
```

```
# 切片：从索引 1 到 3（不包含索引 4）
print(numbers[1:4]) # [2, 3, 4]
```

```
# 切片：每隔一个元素取一个
print(numbers[::2]) # [1, 3, 5]
```

```
# 切片：倒序列表
print(numbers[::-1]) # [5, 4, 3, 2, 1]
```

2. 列表的常用方法

- `append(x)`: 在末尾添加元素
- `insert(i, x)`: 在指定位置插入元素
- `remove(x)`: 删除第一个匹配的元素
- `pop(i)`: 删除并返回指定位置的元素
- `sort()` / `reverse()`: 排序 / 反转列表
- `count(x)`: 统计元素出现次数
- `index(x)`: 返回元素的索引位置

定义一个水果列表

```
fruits = ["apple", "banana", "cherry"]
```

`append`: 在末尾添加 "orange"

```
fruits.append("orange")
```

```
print(fruits) # ['apple', 'banana', 'cherry', 'orange']
```

`insert`: 在索引 1 插入 "pear"

```
fruits.insert(1, "pear")
```

```
print(fruits) # ['apple', 'pear', 'banana', 'cherry', 'orange']
```

`remove`: 删除第一个匹配到的 "banana"

```
fruits.remove("banana")
```

```
print(fruits) # ['apple', 'pear', 'cherry', 'orange']
```

`pop`: 删除并返回最后一个元素

```
print(fruits.pop()) # 返回 'orange'
```

```
print(fruits) # ['apple', 'pear', 'cherry']
```

`reverse`: 反转列表

```
fruits.reverse()
```

```
print(fruits) # ['cherry', 'pear', 'apple']
```

```
# sort: 按字母排序
```

```
fruits.sort()
```

```
print(fruits) # ['apple', 'cherry', 'pear']
```

```
# count:统计元素出现次数
```

```
print(fruits.count("pear")) # 1
```

```
# index:返回元素的索引位置
```

```
print(fruits.index("pear")) # 2
```

三、 作业安排

任务描述:

1. 创建一个包含 1 到 10 的整数的列表
2. 删除其中的偶数
3. 将列表倒序排列

```
# 创建 1 到 10 的整数列表
```

```
nums = list(range(1, 11))
```

```
print("原列表:", nums)
```

```
# 删除偶数 (使用列表推导式)
```

```
nums = [x for x in nums if x % 2 != 0]
```

```
print("删除偶数后:", nums)
```

```
# 倒序排列
```

```
nums.reverse()  
print("倒序后:", nums)
```

四、 总结

列表是有序可变的序列，可以存储不同类型的数据，索引从 0 开始，切片左闭右开，可设置步长。
常用方法：append、insert、remove、pop、sort、reverse 等，列表推导式可用于快速生成和过滤列表

