

# Day5-Python 字符串操作与练习

## 一、学习目标

理解字符串的不可变性与基本结构（索引/切片），掌握拼接、重复、常用方法 strip/upper/lower/split/join/replace，能够用链式调用完成文本清洗和格式化，了解常见易错点并能避免。

## 二、知识点讲解

### 1. 字符串的拼接与重复

拼接 (+)：将多个字符串合并为一个新字符串；重复 (\*)：将原字符串按次数复制并拼接成新字符串。

```
a = "hello"
b = "world"
print(a + " " + b) # 拼接 -> hello world
print(a * 3)       # 重复 -> hellohellohello
```

注意：大量拼接建议用 join，提高效率。

### 2. 字符串的索引与切片

索引用于访问单个字符；切片可访问字符串的部分内容，语法 [开始:结束:步长]，左闭右开。

```
s = "Python String"
print(s[0]) # 第一个字符 P
print(s[-1]) # 最后一个字符 g
print(s[0:6]) # 从索引 0 到 5 -> Python
print(s[::2]) # 每隔一个取一个 -> Pto tig
print(s[::-1]) # 反转字符串 -> gnirtS nohtyP
```

提示：切片返回新字符串，不会改变原字符串；负步长可反向遍历。

### 3. 常用字符串方法

- strip(): 去除首尾空白字符
- upper()/lower(): 大小写转换

- `split()`: 分割字符串
- `join()`: 合并字符串列表
- `replace(a, b)`: 替换内容

```
text = " Hello Python " # 定义一个字符串, 前后有两个空格
print(text.strip()) # strip() 去掉字符串首尾的空格 → "Hello Python"
print(text.upper()) # upper() 转成全大写 → " HELLO PYTHON " (空格保留)
print(text.lower()) # lower() 转成全小写 → " hello python " (空格保留)
print(text.split()) # split() 按空格拆分成列表 → ["Hello", "Python"]
print("-".join(["Python", "is", "fun"])) # join() 用 "-" 把列表拼接成字符串 → "Python-is-fun"
print(text.replace("Python", "World")) # replace() 把 "Python" 替换成 "World" → " Hello
World "
```

### 三、 作业安排

#### 任务描述:

1. 定义一个字符串 `s = " Python is powerful. "`
2. 去除两端空格 → 转为大写 → 按空格拆分成列表 → 用连字符-连接, 展示链式方法调用的组合应用。

```
s = " Python is powerful. "
result = s.strip().upper().split()
result = "-".join(result)
print(result) # PYTHON-IS-POWERFUL.
```

### 四、 总结

字符串是不可变序列, 索引从 0 开始, 切片左闭右开, 支持负步长。拼接用 `+` 或 `join`; 文本清洗常用 `strip/split/join`, 大小写用 `upper/lower/title`。链式调用可简化多步处理, 提高代码可读性。