
字符串不是背方法

第 4 章 | 真题中的精确匹配、文本清洗与最终格式化

Python 零基础培训

100 分钟课堂

本课只解决竞赛最需要的五件事

1. 区分字符串与数值;
2. 精确匹配前先检查空白和符号;
3. 会用 `strip/replace/split/join`;
4. 会写 f-string, 也能看懂 `format`;
5. 坚持“清洗、计算、选择、最后格式化”。

三套真题怎样用到字符串？

套题	直接证据	本课能力
名仕 科图 青娅	URL、字段名、数字文本 性别匹配、百分比格式 项目匹配、固定两位	定义与类型转换 精确匹配与格式化 文本筛选与交付

字符串不单独出题，却藏在采集、筛选、分组和交付中。

引号决定“这是文本”

`"125" * 3` `# "125125125"`

`125 * 3` `# 375`

`"125" == 125` `# False`

只有业务上需要计算时才转换：`int("125")` 或 `float("19.9")`。

精确匹配：多一个空格也不相等

```
gender = " 女 "  
print(repr(gender))           # ' 女 '  
print(gender == " 女")       # False  
print(gender.strip() == " 女") # True
```

科图题中的女性占比依赖精确匹配。 肉眼判断前先用 repr 看隐藏字符。

字符串不可变：方法返回新值

```
label = " 支付宝  "
label.strip()
print(repr(label))  # 仍有空格
```

```
label = label.strip()
```

调用方法不等于原变量自动改变。要重新赋值或直接使用返回值。

本课最重要的四个普通字符串方法

方法	作用	返回结果
<code>strip()</code>	删除两端空白	新字符串
<code>replace()</code>	替换指定子串	新字符串
<code>split()</code>	按分隔符拆分	子串列表
<code>join()</code>	用连接符拼接	新字符串

清洗必须有业务规则，不能看到符号就随意替换。

替换、分割与拼接

```
project = " 加： 期初现金 "  
project = project.strip().replace(":", ": ")
```

```
parts = "2024-East-0018".split("-")  
code = "_".join(parts)
```

先预测返回类型：split 返回列表，join 返回字符串。

会写 f-string，也要看懂 format

```
value = 66.6667
"{:.2f}%".format(value)    # 科图参考代码
f"{value:.2f}%"            # 推荐写法
```

两种写法都得到 **66.67%**。课堂统一练 f-string，不要求背旧式百分号格式化。

round 不等于固定显示两位

```
amount = 12.3  
round(amount, 2)      # 数值 12.3  
f"{amount:.2f}"       # 文本 "12.30"
```

青娅题若严格要求“不足两位用 0 补足”，最终交付副本需要格式化。

为什么必须最后格式化？



字符串按字符比较： "9.50%" 可能被判断为大于 "10.00%"。

一个字符串与一整列文本不一样

```
text = text.strip()
```

```
df[" 性别"] = (  
    df[" 性别"].astype("string").str.strip()  
)
```

普通文本用方法；pandas Series 通过 `.str` 访问文本方法。

科图微案例：固定处理顺序

1. 清理付款方式两端空白；
2. 按年份和区域分组；
3. 计算组内付款方式占比；
4. 用 `idxmax()` 每组选一行；
5. 最后把占比格式化为百分比文本。

不能在选最大值之前加百分号。

科图：清洗与最后格式化

```
work[" 付款方式"] = (  
    work[" 付款方式"].astype("string").str.strip()  
)  
result = result.loc[  
    result.groupby([" 年", " 区域分布"])[" 占比"].idxmax()  
].copy()  
result[" 占比"] = result[" 占比"].map(  
    lambda x: f"{x:.2f}%"  
)
```

青娅微案例：项目名先清洗再筛选

- ▶ 目标项目名称很长，必须精确一致；
- ▶ 先用 `.str.strip()` 清理项目列；
- ▶ 用 `.isin([...])` 一次匹配两个目标值；
- ▶ 金额先填缺失、保持数值；
- ▶ 只对最终导出副本固定两位。

青娅： 计算副本与导出副本分开

```
selected = work.loc[work[" 项目"].isin(targets)].copy()
selected[amount_cols] = (
    selected[amount_cols].fillna(0).round(2)
)
```

```
export_rows = selected.copy()
for col in amount_cols:
    export_rows[col] = export_rows[col].map(
        lambda x: f"{x:.2f}"
    )
```

教材内容怎样取舍？

必讲

- ▶ 定义与类型边界
- ▶ 精确匹配
- ▶ strip、replace
- ▶ split、join
- ▶ f-string 与 format 识读
- ▶ pandas.str

删减/按需查阅

- ▶ 旧式百分号格式化
- ▶ find 的细节参数
- ▶ 全套大小写方法
- ▶ 字符串对齐
- ▶ 教材文字排版案例

离场检验：只问真正影响得分的内容

- ▶ 为什么 " 女 " 匹配不到 " 女 " ？
- ▶ 一个字符串和一整列文本怎样删除两端空白？
- ▶ `round(12.3, 2)` 与固定两位输出有什么不同？
- ▶ 为什么百分比必须最后格式化？
- ▶ 科图和青娅微案例的完整处理顺序是什么？