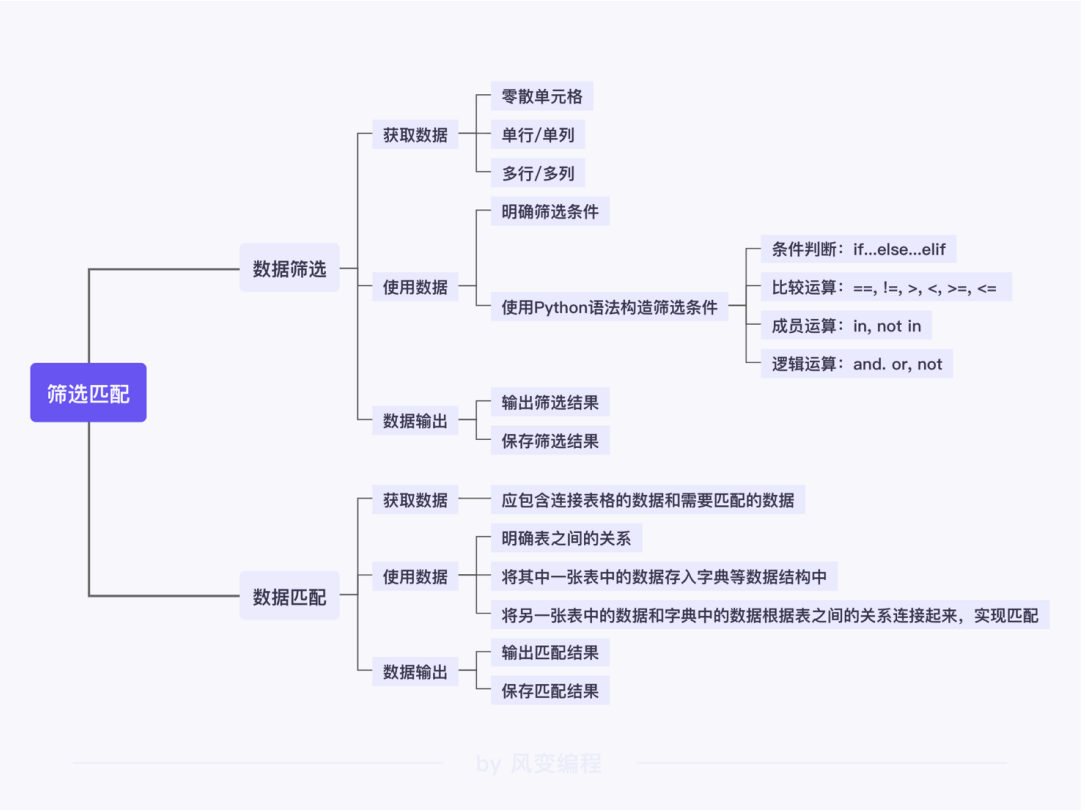


# 高效办公第4关笔记



本关两个项目案例代码，分别解决Excel常见场景中的数据筛选问题和数据匹配问题。

**数据筛选**要求我们在表中筛选出**符合条件的**数据。

**数据匹配**需要我们在多个表之间**匹配相关**的数据。

## 一、案例1 – 数据筛选

### 目标

【10月考勤统计.xlsx】工作簿中，保存了公司一百名员工的迟到信息，这些信息包含了迟到时间和迟到次数，**需要将迟到时间超过45分钟且迟到过3次以上的员工筛选出来，并结果保存为【10月迟到人员信息.xlsx】。**

分解为三个步骤：获取数据，使用数据和数据输出。

### 获取数据

需要获取两部分数据：表头数据和表头以外的所有数据。代码构成👉

```

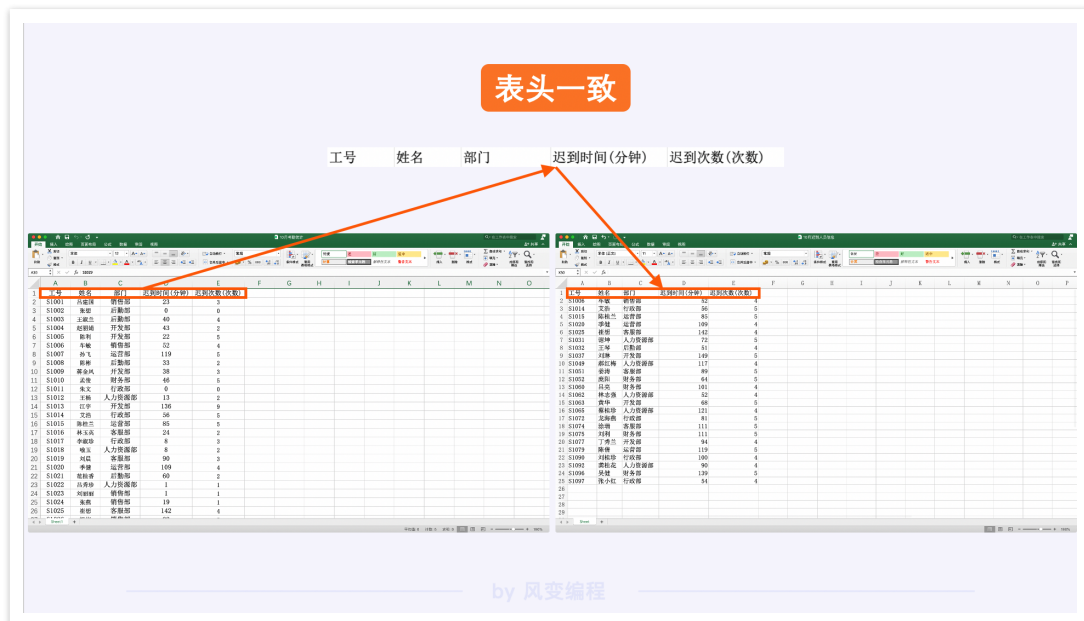
1 from openpyxl import load_workbook, Workbook
2
3 # 打开【10月考勤统计.xlsx】工作簿
4 wb = load_workbook('./material/10月考勤统计.xlsx')
5 # 获取活动工作表
6 ws = wb.active
7 # 获取表头
8
9 late_header = []
10 for cell in ws[1]:
11     late_header.append(cell.value)
12
13
14 # 新建工作簿
15 new_wb = Workbook()
16 # 获取新工作簿中的工作表
17 new_ws = new_wb.active
18
19
20 # 将表头写入新工作簿的工作表中
21 new_ws.append(late_header)

```

因为我们新写入的数据，也同样是需要将相对应的表头的，所以先将表头读取出来，先写入到新的表格中等待备用。

同样在筛选数据之前，完成新的表格建立，方便后续的数据的写入。

【在电脑上可双击可放大图片】



## 使用数据（数据筛选）

通过分析任务需求可以总结出三个筛选条件：

- ① 迟到时间大于45分钟。
- ② 迟到次数大于3次以上。

③ 同时满足上面两个条件。

借助条件判断语句，比较运算符，成员运算符和逻辑运算符等Python基础知识，实现对于数据的筛选。

构造筛选条件的常见Python基础语法		
Python基础语法	筛选条件	示例
条件判断	条件	If 条件
比较运算	条件：A等于B	If A == B
	条件：A不等于B	If A != B
	条件：A大于B	If A > B
	条件：A小于B	If A < B
	条件：A大于等于B	If A >= B
	条件：A小于等于B	If A <= B
成员运算	条件：A在指定的序列B中	If A in B
	条件：A不在指定的序列B中	If A not in B
逻辑运算	条件1与条件2同时成立	If 条件1 and 条件2
	条件1成立或条件2成立	If 条件1 or 条件2
	不满足条件1	If not 条件1

by 风变编程

代码构成👉

```
1 # 从第二行开始遍历表格
2 for row in ws.iter_rows(min_row=2, values_only=True):
3     # 取出姓名，迟到时间和迟到次数
4     name = row[1]
5     time = row[3]
6     number = row[-1]
7     # 判断是否迟到
8     if time > 45 and number > 3:
9         print('{}迟到了{}分钟，迟到了{}次'.format(name, time, number))
10    # 将迟到人员信息写入新工作簿的工作表中
11    new_ws.append(row)
```

设置min\_row=2是为了将除了表头以外的所有的数据读取出来，方便后续的数据筛选，而我们都知使用iter\_rows()方法读取出来的数据，是以元祖进行封装的，所以需要相对应的偏移量进行取值。

而为什么迟到的次数，用的是负索引，这里有一个知识点，正索引是从左到右从0开始正着数，而负索引是从右到左倒着数，好比偏移量0永远代表的是第1位元素一样，偏移量-1永远代表的最后一位元素。

相对应这里而言，你写成number = row[4]也是可行的。

正索引	0	1	2	3	4
负索引	-5	-4	-3	-2	-1
数据	工号	姓名	部门	迟到时间	迟到次数

数据输出

将筛选结果保存起来，存储到【10月迟到人员信息.xlsx】中。代码构成📌

```
1 # 将新工作簿保存为【10月迟到人员信息.xlsx】
2 new_wb.save('./material/10月迟到人员信息.xlsx')
```

## 二、案例2 – 数据匹配

### 目标

匹配两张表格中指定的迟到次数。

### 获取数据

需要获取【10月考勤统计.xlsx】和【迟到次数月度统计（10月更新）.xlsx】中需要匹配的数据，至少应包括工号与迟到次数这两列数据。

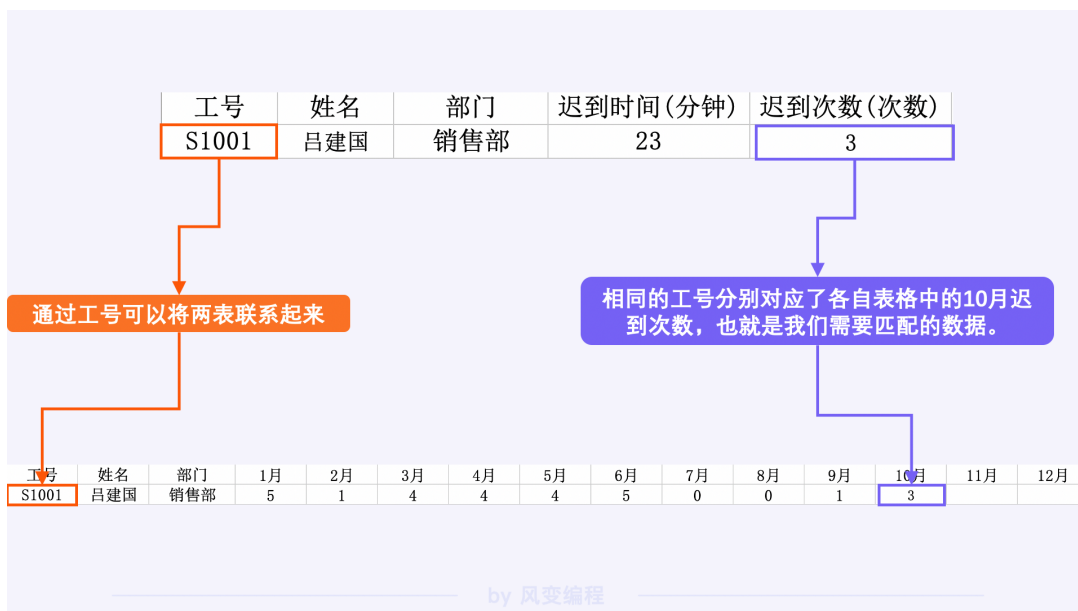


代码构成📌

```
1 from openpyxl import load_workbook
2 # 打开工作簿【10月考勤统计.xlsx】，获取活动工作表
3 wb = load_workbook('./material/10月考勤统计.xlsx')
4 ws = wb.active
5 # 创建迟到人员字典
6 info_dict = {}
```

这里的代码构成很简单，先打开两表中任意一个表格，并创建一个字典，用于创建迟到人员的字典。这里的话，算是一个字典的应用，其出发点、思考点都是针对于本题而言，当成是一个思路拓展，后续有类似的需要，写类似的代码时，可以直接用这个思路。

### 使用数据（数据匹配）



通过表格可以依靠工号连接起来。也就是说，相同的工号都可以对应各自表格中的十月份迟到次数。代码构成👉

```

1 # 循环读取除表头外的表格数据
2 for row in ws.iter_rows(min_row=2, values_only=True):
3     # 取出员工工号
4     staff_id = row[0]
5     # 取出迟到次数
6     staff_late = row[-1]
7     # 将信息添加入字典，字典格式为{'员工工号': '迟到次数'}
8     info_dict[staff_id] = staff_late

```

选择使用字典后，这里就先遍历出【10月考勤统计.xlsx】表单中的所有工号、迟到次数，是这么组成{'员工工号': '迟到次数'}，等待备用。

```

1 # 打开工作簿【迟到次数月度统计（10月更新）.xlsx】，获取活动工作表
2 monthly_wb = load_workbook('./material/迟到次数月度统计（10月更新）.xlsx')
3 monthly_ws = monthly_wb.active
4
5 # 循环读取除表头外的表格数据
6 for monthly_row in monthly_ws.iter_rows(min_row=3, max_col=13,
7 values_only=True):
8     # 取出员工工号
9     member_id = monthly_row[0]
10    # 取出十月份的迟到次数
11    member_late = monthly_row[-1]
12    # 匹配迟到次数是否相等
13    if member_late != info_dict[member_id]:
14        print('工号{}迟到情况不匹配，请核查后更新'.format(member_id))

```

这里做了两个动作，获取到【迟到次数月度统计（10月更新）.xlsx】这个工作簿，并利用循环将表头以外的信息取到，再取到的同时，进行判断。程序逻辑是这样的。

①取到【迟到次数月度统计（10月更新）.xlsx】中的工号(member\_id)

②取到【迟到次数月度统计（10月更新）.xlsx】中的迟到次数(member\_late)

③来判断

```
if member_late != info_dict[member_id]:  
    print('工号{}迟到情况不匹配, 请核查后更新'.format(member_id))
```

用【迟到次数月度统计（10月更新）.xlsx】中的迟到次数(member\_late)

与

【10月考勤统计.xlsx】的迟到次数进行比较。

④如果错误就输出相对应的工号。

这里有个点需要理解：

怎么取出【10月考勤统计.xlsx】的迟到次数的？

因为【迟到次数月度统计（10月更新）.xlsx】与【10月考勤统计.xlsx】的工号一致，所

从【迟到次数月度统计（10月更新）.xlsx】取出来的工号是通过字典【key】的方式取出相对应的值的，刚刚等待备用的字典为此而来。

## 数据输出

如果相同工号对应的迟到次数不相同，就把工号输出在终端上，提醒相关人员去核实情况。