

## Day10-for 和 while 循环

### 一、学习目标

通过今天的学习，你将掌握 for 循环的基本语法与常见用途，while 循环的使用场景与控制方式，如何使用 break 和 continue 控制循环流程，和循环中的常见错误。

### 二、知识点讲解

#### 1. 为什么要用循环？

在编程中，我们经常需要重复执行某段代码。 例如：

- 打印 1 到 10 的所有数字
- 遍历列表中的每个元素
- 持续接收用户输入直到输入“退出”

使用循环语句可以轻松解决这个问题。

#### 2. for 循环：遍历已知序列；如图 1 所示。

##### ☑ 适用场景：

- 遍历列表、字符串、元组、范围（range）等可迭代对象
- 已知循环次数

```
#遍历列表
fruits = ["苹果", "香蕉", "橙子"]

for fruit in fruits:
    print("我喜欢吃", fruit)
```

✓ 0.0s

我喜欢吃 苹果  
我喜欢吃 香蕉  
我喜欢吃 橙子

图 1: for 循环, 遍历列表

3. while 循环: 条件满足时重复执行; 如图 2 所示。

✓ 适用场景:

- 循环次数未知
- 只要某个条件为真就继续执行

```
#计数循环
i = 1
while i <= 5:
    print(i)
    i += 1 # 千万别忘了更新变量!

#必须手动更新变量, 否则会无限循环!
```

✓ 0.0s

1  
2  
3  
4  
5

图 2: while 循环, 计数循环

#### 4. break: 提前退出循环; 如图 3 所示。

✓ 作用:

- 立即终止整个循环, 跳出循环体

```
#找到目标后停止搜索
numbers = [1, 3, 5, 7, 9, 11, 13]

for num in numbers:
    if num == 7:
        print("找到了! 是", num)
        break # 找到就退出, 不再继续
    print("正在检查", num)
```

✓ 0.0s

正在检查 1  
正在检查 3  
正在检查 5  
找到了! 是 7

图 3: break 退出

#### 5. continue: 跳出当前轮, 进入下一轮; 如图 4 所示。

✓ 作用:

- 跳过本次循环的剩余语句, 直接进入下一次循环

```
#只打印偶数
for i in range(1, 6):
    if i % 2 == 1: # 如果是奇数
        continue # 跳过
    print(i)
✓ 0.0s

2
4
```

图 4: continue

## 6. 嵌套循环（进阶）

可以在一个循环内部再写一个循环。

✧ 注意：内层循环每完成一次，外层循环才进行下一步。

## 7. 常见错误：如表 1 所示。

忘记冒号	for i in range(5)	加上:
缩进错误	代码块不对齐	统一用 4 空格
while 无限循环	忘记更新变量	确保条件能变为 False
使用=而不是==	while x = 5:	改为==
break/continue 位置错误	在不该跳过的地方跳过	理清逻辑顺序

表 1: 常见错误

### 三、作业安排

使用 Vscode 或者 PyCharm 创建一个 Jupyter Notebook 文件, 并通过编写代码完成下面的小任务, 最终将成功运行后的整个 Vscode/Pycharm 界面截图发至打卡营群内。

- 1.使用 for 循环打印 1 到 100 中所有 3 的倍数。
- 2.使用 while 循环求 1 到 100 的总和。(使用 ipynb 格式的文件)

### 四、总结

今天学习了 for 和 while 循环, break 和 continue。要合理使用 break 和 continue, 同时要避免无限循环。