M7 · 第 19 讲

读懂报错与迭代修复

把红字变成下一步动作

阅读约 30 分钟

练习约 30 分钟

产出：一份可复现的报错说明

本节目录

开场：看到红字就慌，于是全部重写

本课目标

一、只看三处

二、四类错误

三、最小复现

四、怎么把报错交给 AI

五、修复顺序

六、版本与回滚

七、30 分钟练习

八、检查清单

九、延伸阅读

开场：看到红字就慌，于是全部重写

脚本一跑，终端红了一片。你的第一反应是“AI 写的什么破代码”，然后把整段删掉重来。三轮之后，问题还在。

报错不是失败，是**系统在告诉你下一步查哪里**。这一讲只练一件事：把报错读成动作。

本课目标

- 从 traceback 里找到真正有用的两行；
- 分清四类错误；
- 用最小复现定位问题；
- 给 AI 提供足够信息，而不是一句“报错了”。

一、只看三处

一条 Python 报错通常很长，但有用的只有三处：

1. **最后一行**：错误类型与原因；
2. **倒数第二段**：出错的文件与行号；
3. **中间调用链**：哪一层函数传进去的。

示例：

```
Traceback (most recent call last):
  File "count.py", line 7, in <module>
    rows = list(csv.DictReader(f))
KeyError: 'city'
```

最后一行告诉你：字段 city 不存在；倒数第二段告诉你：第 7 行读表时出的问题。

二、四类错误

类型	典型报错	先查什么
语法	SyntaxError、IndentationError	行号附近，缩进/括号/引号
依赖	ModuleNotFoundError	是否安装、虚拟环境是否选对
路径	FileNotFoundError	当前目录、文件名、权限
逻辑	KeyError、IndexError、TypeError	数据长什么样、类型对不对

前两类看环境，后两类看数据。分清楚，才不会乱改代码。

三、最小复现

不要在一次运行里放 200 个文件。先做最小复现：

- 只留 1 个输入文件；
- 只跑出错的那一步；
- 打印中间结果（前 3 行、字段名、类型）。

三行调试常常比读十页代码有用：

```
print(rows[:3])
print(rows[0].keys())
print(type(rows[0]))
```

四、怎么把报错交给 AI

差的问法：“你的代码报错了，帮我看看。”

好的问法，一次给全：

> 环境：macOS，Python 3.11，已安装 pandas 2.2。 目标：读 orders.csv 统计城市。 代码：（粘贴完整文件） 报错：（粘贴完整 traceback） 我已确认：文件存在，表头是 city,amount；用 3 行样本也会报错。 请先判断原因，再给最小修改，不要重写整个文件。

信息越完整，改得越准。

五、修复顺序

1. 先让程序能跑（哪怕只处理 1 条）；
2. 再让结果正确（对照手工答案）；
3. 再处理异常（空值、格式错、权限）；
4. 最后整理结构（函数、日志、参数）。

顺序反了，会在没跑通的代码上做无意义的优化。

六、版本与回滚

- 每次能跑通，就复制一份“可用版本”备份；
- 大改之前先提交或另存；
- 改坏了直接回到上一版，不要在现场硬修；
- 记录每次改了什么，一行就够。

七、30 分钟练习

1. 故意制造一个 KeyError（改错字段名）（5 分钟）；
2. 只读最后一行与行号，写下你的判断（5 分钟）；
3. 用 print 打印数据结构（10 分钟）；
4. 按“环境 + 代码 + 报错 + 已确认”重问一次（10 分钟）。

八、检查清单

- [ ] 能指出报错的类型与行号；
- [ ] 分得清环境问题还是数据问题；
- [ ] 会做最小复现；
- [ ] 给 AI 的信息包含环境与已排查项；
- [ ] 每次跑通都有备份。

九、延伸阅读

- 知识库 2.0 AI 编程
- 知识库 3.2 工具：辅助学习工具和资源
- 知识库 2.3 藏经阁：个人专栏

> 下一讲：把一个想法部署上线。你会知道静态页、Serverless、服务器各自适合什么。
