

# 回顾：第 1 周的挑战

## 还记得这些痛点吗？

- `cv2.imread()` 返回 `None`，找不到原因
- 图像显示颜色异常
- 数组越界、维度不匹配
- API 参数太多，记不住

## 典型错误代码

```
1 # 死循环！
2 while img is not None:
3     cv2.imshow('image', img)
4     if cv2.waitKey(1) == 'q':
5         break
```

## 传统解决方式：

- ① 翻文档 ([docs.opencv.org](https://docs.opencv.org))
- ② 搜索 StackOverflow
- ③ 问同学/老师
- ④ 试错（耗费大量时间）

## AI 辅助的新方式

- 直接问 AI
- 解释错误原因
- 给出修改建议
- 降低学习门槛！

## Token 预测

+

## 模式匹配

为什么 AI 擅长语法但不一定懂逻辑？

- **擅长：**基于海量代码库的模式复刻
- **不擅长：**理解你的具体业务逻辑

# 幻觉专题

## 什么是 AI 幻觉？

AI 自信满满地生成看似合理但实际上错误或不存在的信息。

**案例：AI 发明了一个不存在的 OpenCV 函数**

## AI 的“创意”代码

```
1 import cv2
2
3 img = cv2.imread('exam.jpg')
4
5 # 不存在的函数！
6 fixed = cv2.auto_fix_exposure(img)
```

## 如何识别幻觉？

- 查官方文档 ([docs.opencv.org](https://docs.opencv.org))

# 编程范式的演进

## 1.0 阶段：查阅文档/StackOverflow

- 手动搜索
- 阅读文档
- 试错调试

效率：两星

## 2.0 阶段：AI 辅助

- 从“写作者”转变为“审查者”
- AI 生成初稿，人审查修改
- 快速原型验证

效率：四星

## 3.0 阶段：AI 原生？

- AI 直接理解需求生成完整系统
- 人主要负责需求定义和架构设计
- 可能的未来方向

效率：五星？

核心结论：程序员的核心竞争力转变

从“记忆语法”转向“问题定义”与“代码审查”

