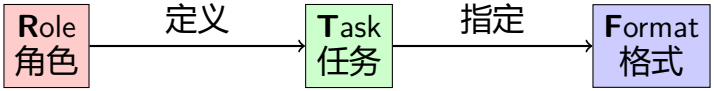
结构化 Prompt 框架：RTF 模式



RTF 模板示例

Role (角色)：你是一个资深计算机视觉算法专家，精通 OpenCV 和图像处理。

Task (任务)：实现试卷图像的二值化，要求能自适应处理光照不均的情况。

Format (格式)：

- 返回带有详细 Docstring 的 Python 函数
- 包含输入输出示例
- 列出关键参数的调优建议

RTF 框架实战对比

不好的 Prompt

帮我写个代码处理图像。

问题分析：

- 没有定义角色
- 任务模糊
- 没有格式要求

好的 Prompt (RTF)

Role：你是一位有 10 年经验的计算机视觉工程师，精通 OpenCV。

Task：请编写一个 Python 函数，实现答题卡图像的自适应二值化。

Format：

- 提供完整的、可运行的代码
- 包含详细的函数文档
- 解释关键参数的设置依据

进阶技巧：Chain-of-Thought (思维链)

什么是思维链?

让 AI "**一步一步思考**" (Step by Step)，而不是直接给出答案。

直接给答案

Prompt： 用 OpenCV 实现透视变换矫正倾斜的试卷。

问题：

- AI 直接扔出代码
- 学生不理解原理
- 换个场景就不会了

思维链引导

Prompt：
请帮我实现试卷透视变换矫正。请按以下步骤思考：
Step 1： 透视变换的数学原理是什么？需要哪些参数?
Step 2： 如何从图像中自动找到试卷的四个角点?
Step 3： 请写出完整的 Python 实现代码

思维链实战：准确率对比

测试任务：用 OpenCV 实现自适应阈值处理

Prompt 类型	直接提问	思维链引导
代码完整度	70%	95%
参数解释清晰度	一般	详细
边界情况处理	很少提及	全面考虑
实际运行成功率	60%	90%

思维链 Prompt 模板

请帮我解决 [问题]。请按以下步骤思考：
Step 1：分析问题的核心要点是什么？
Step 2：有哪些可能的解决方案？各自的优缺点？
Step 3：请给出推荐的实现代码

进阶技巧：Few-shot (少样本提示)

什么是 Few-shot?

给 AI 几个 **正确的例子**，让它模仿你的风格或格式处理新任务。

示例场景： OpenCV 图像转换

Few-shot Prompt

请模仿以下示例的代码风格和注释规范：

示例 1 - 灰度化：

```
# 读取图像并转换为灰度图
# 参数: image_path: 图像文件路径
# 返回: gray_image: 灰度图像数组
def load_and_gray(image_path):
```

待处理任务：

继续 Prompt

示例 2 - 高斯模糊：

```
# 对图像进行高斯模糊
# 参数: image: 输入图像
# 返回: blurred: 模糊后的图像
def gaussian_smooth(image):
    blurred =
cv2.GaussianBlur(image, (5,5), 0)
```

Few-shot 效果对比

Prompt 类型	无 Few-shot	有 Few-shot
代码风格一致性	随机，不稳定	与示例高度一致
注释完整度	较简略	详细，符合示例规范
参数说明	缺失或不清晰	完整的 Docstring
错误处理	经常遗漏	按示例模式添加
代码可读性	一般	优秀

Few-shot 使用技巧

1. **示例数量：** 2-3 个示例通常足够
2. **示例质量：** 确保示例是正确的、高质量的
3. **格式一致：** 示例之间保持风格一致
4. **明确指令：** 告诉 AI " 请按照示例的风格"

上下文窗口管理

问题：为什么代码太长了 AI 就会" 忘掉" 前面的内容？

- 大模型有 **上下文窗口限制**（通常 4K-128K tokens）
- 超出限制后，模型会" 遗忘" 最早的内容
- 导致前后文不一致、回答质量下降

各模型上下文限制：

模型	上下文
GPT-4	8K/32K
GPT-4o	128K
Claude 3.5	200K
DeepSeek	64K
通义千问	128K

精简 Prompt 的技巧：

1. **只提供必要代码**：不要贴整个文件
2. **使用摘要**：" 前面我们讨论了 X，现在要解决 Y"
3. **分段处理**：长任务拆分成多个短任务
4. **定期总结**：让 AI 总结当前进度