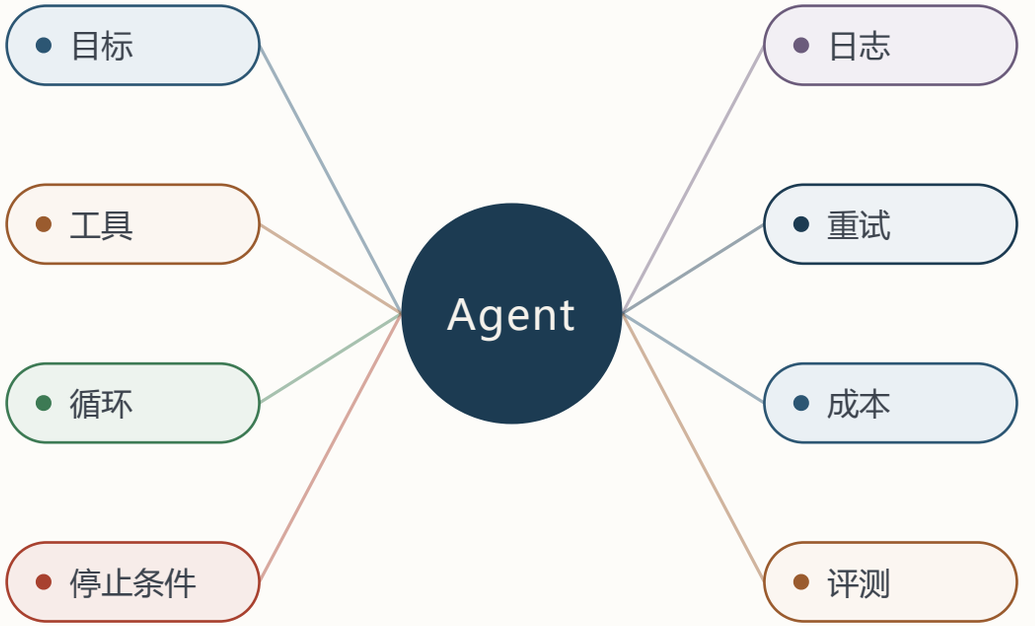
COURSE 05 · AI AGENT

AI AGENT 落地实操

从一次对话到一个能干活、能兜底的系统



AI 教学课件工厂 · 系列课程 05

面向技术与产品团队 / 120 分钟

什么才算一个 AGENT

能调用工具、能根据结果调整下一步，才叫智能体。

01 对话模型

输入一段文字，输出一段文字，过程一次结束

02 智能体

观察、决策、调工具、看结果、再决策，形成循环

03 关键差别不是模型更聪明

而是把执行权交给了循环

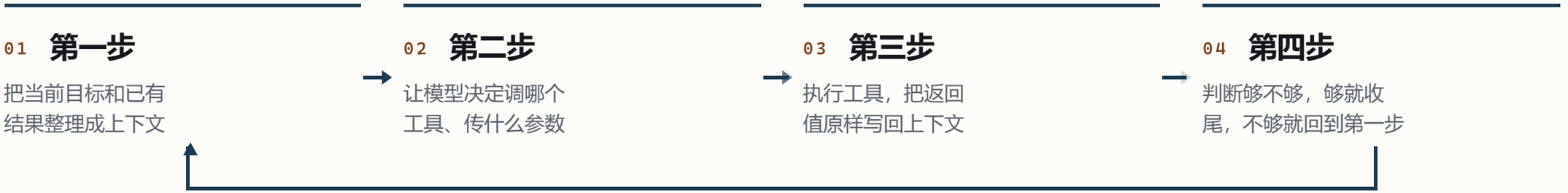
04 有循环就必须有停止条件

否则它会一直跑下去

一句话判别：如果拿掉工具调用，它就退化成一次普通问答。

工具调用与循环，拆开看只有四步

别被框架名词绕晕，最小实现就是四步加一个判断。



最容易被忽略的是第三步：返回值要原样写回，不要先替模型「总结」一遍。

什么时候不该上 AGENT

能用一次问答解决的，别引入循环。

01 任务是单步的、确定的：直接调用，连模型都不一定需要

02 出错代价高且不可回滚：转账、删数据、对外发消息

03 没有清晰的成功判据：循环不知道什么时候该停

04 延迟敏感：每多一轮工具和推理，用户就多等一截

01
单步任务
直接写函数

02
不可回滚
转账 删数据

03
没有判据
不知道何时停

04
延迟敏感
每轮都要等

判断口诀：单步用函数，多步才用智能体；高风险动作一律加人工确认。

最小可用架构：五个部件

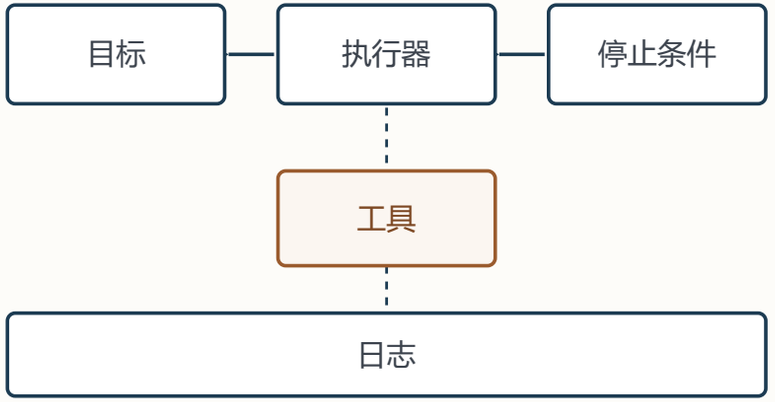
先跑通一个闭环，再谈记忆、多智能体和花哨能力。

01 目标与上下文：这一轮要干什么、已经知道什么

02 工具注册表：名称、参数说明、返回样例、失败语义

03 执行器：真正跑工具，负责超时、权限与幂等

04 停止条件：最大步数、超时、成功判据、人工接管入口



第五个部件是日志，它不算业务逻辑，但没有它，前面四个出了问题你都看不见。

工具描述比提示词更重要

模型选错工具，八成是工具说明写得像给人看的注释。

01 写清适用与不适用：什么情况下不要调这个工具

02 给一个真实返回样例，比三行抽象描述有用

03 参数用枚举和单位，别写「适当」「合理」这类词

04 失败要可区分：参数错、无权限、超时、空结果分开说

边界
什么情况下不要调它
适用性

样例
一个真实返回胜过三行描述
返回体

参数
用枚举和单位，别写「适当」
入参

失败
参数错 / 无权限 / 超时 分开说
错误

把工具说明当 API 文档写，并且每次改工具都同步改说明。

状态、记忆与失败重试

循环能跑起来之后，真正的难点是让它跑得稳。

01 状态外置

把进度存下来，重启后能从断点继续

02 记忆分层

本轮上下文、任务摘要、长期知识分开存

03 重试分级

参数错改参数，超时换工具，逻辑错回退一步

04 所有外部调用都要能重放

避免重复发送、重复扣款

先设计失败路径，再设计成功路径——线上出问题时，你只会用到前者。

没有轨迹日志，就没法复现问题

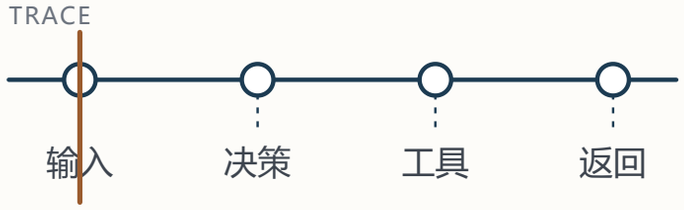
智能体的 bug 往往不在某一行代码，而在某一次决策。

01 每一步记四件事：输入、模型决策、工具返回、耗时

02 给每次运行一个 trace id，串起所有步骤与外部系统

03 敏感字段脱敏后再落盘，日志本身也是数据资产

04 保留可回放能力：拿到同一份轨迹就能重跑一遍



用户报「结果不对」时，你要能回答：它第几步、看到了什么、才做了那个决定。

成本、延迟与评测集

上线前先量三件事，不然优化没有方向。

01 单次任务平均步数：步数比模型单价更影响成本

02 端到端延迟分布：看九十分位，别只看平均值

03 评测集：二十到五十条真实任务，含成功与失败判据

04 每次改提示词或工具，都跑一遍评测集再发版

01
平均步数
决定成本

02
延迟
只看 P90

03
评测集
20–50 条

04
发版前
跑一遍

别用「感觉变好了」发版；一个固定的小评测集，比三轮人工试用可靠。

把循环管住，比把模型换大更重要

今天带走三样东西：四步循环、五个部件、一个评测集。



课后作业：挑一个真实的多步任务，写出工具注册表和停止条件，先不接模型，用手模拟跑一遍循环。