

# 精华

## 从传统应用到 基于大模型应用 设计思想的35讲

《AI智能体系列知识分享》

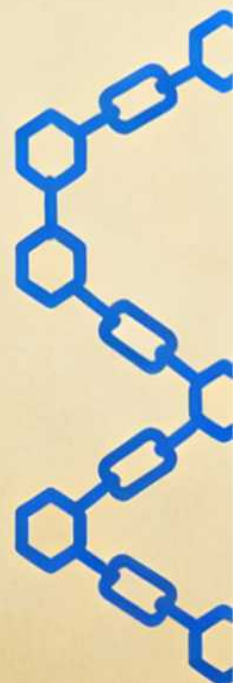
openClaw龙虾是把打开智能体  
普及的生态钥匙

康乐 - 2026-03



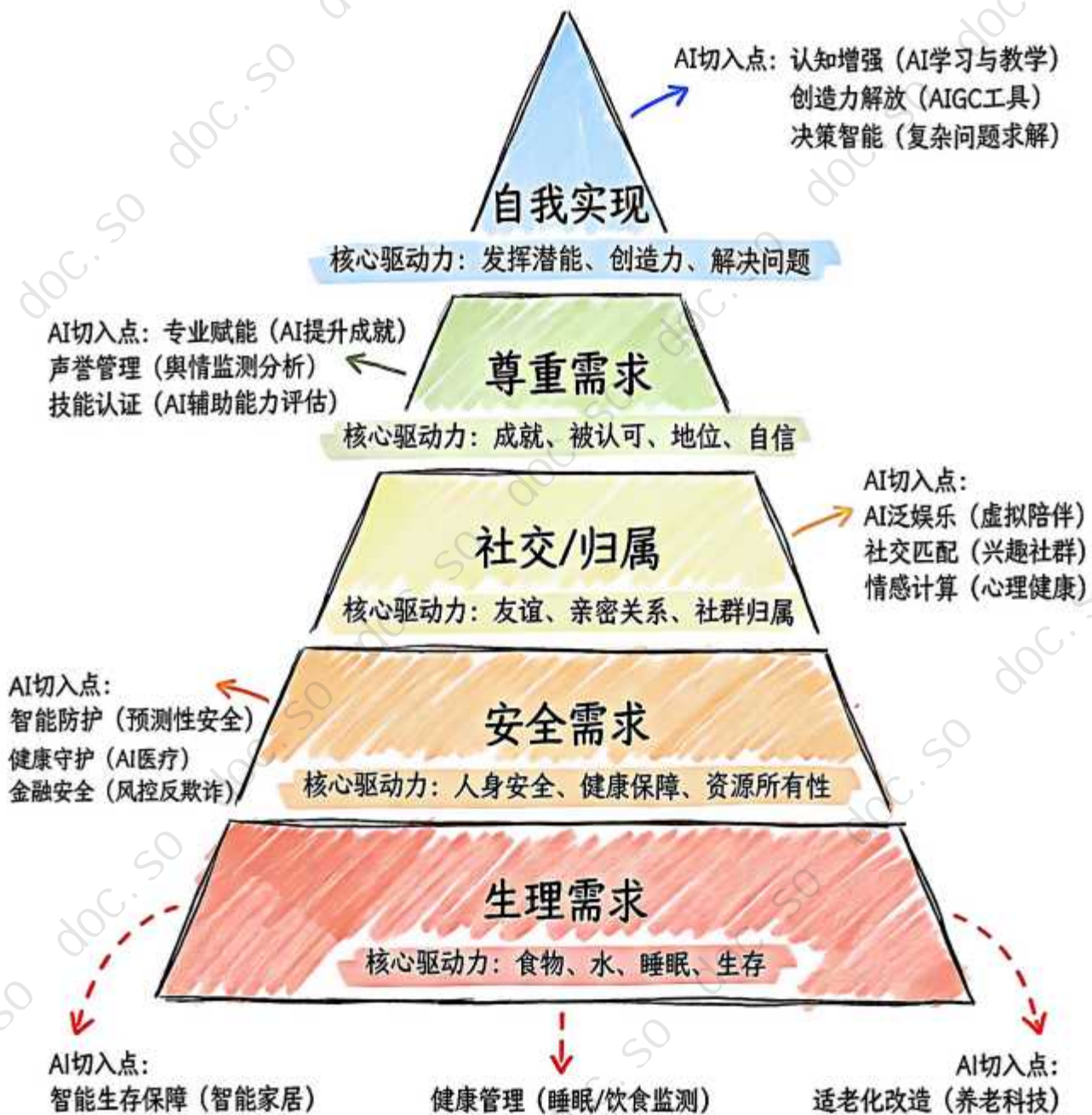
扫码下载完整版

无偿分享在：<http://doc.so> 学习交流



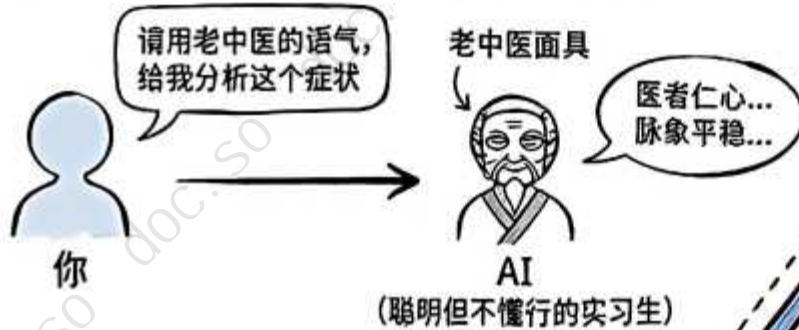
# 2026一人公司需洞察的AI机会

## 从马斯洛需求层次理论找AI切入点



# 关于AI训练这回事

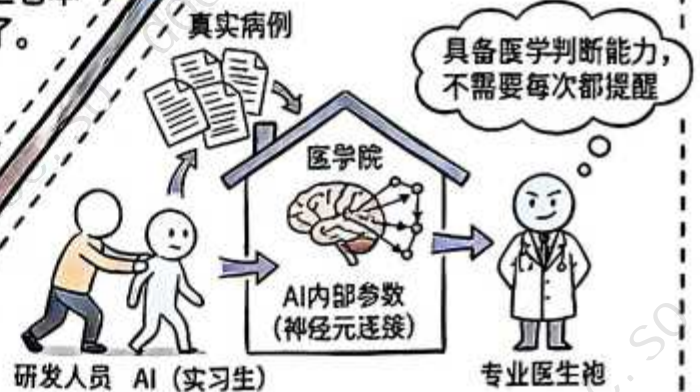
## 普通人的“训练”（提示词工程）



实际上只是在教AI怎么说话，没改变AI本身。就像跟实习生对话，AI立刻切换风格，但它本身不会变成中医，下次你不提醒它就忘了。

- 0 零成本：不需要显卡、不需要数据
- 🦎 随时变：今天让AI当医生，明天当律师，改几个字就行
- 🐟 效果有限：AI的“底子”没变，复杂专业问题还是会露馅

## 研发人员的训练 (Fine-tuning)



是真的给AI“换脑子”，改变它的内部参数。就像把实习生送去医学院，用成千上万份真实病例“喂”给AI，AI的内部权重被永久改变，学成后具备了专业能力。

- 🔌 真·训练：需要GPU集群、大量标注数据、数小时到数周时间
- 💰 成本高：全量微调需要调整模型100%的参数
- 📖 效果深：AI学会了领域知识，回答更专业一致

但本质区别是：提示词工程是利用现有模型能力，微调是创造新的模型能力。

当每个人都能  
指挥很多AI智能体

管理能力  
最重要



# 怎么锻炼 AI 管理能力的

## 3点建议

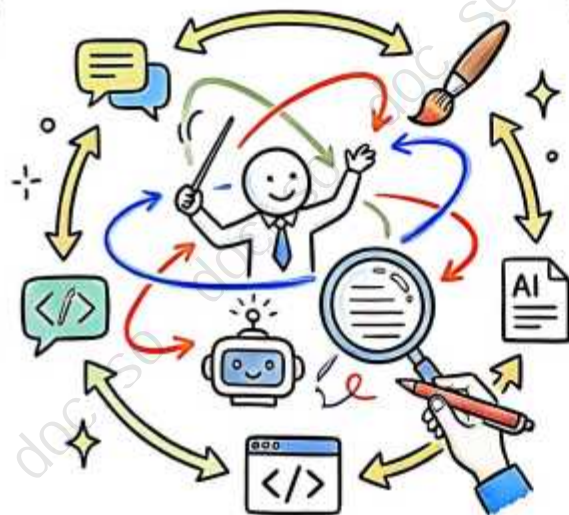
### 1. 先有意识



管理核心:不是自己做,是“意识到”可以委托。以前靠岗位,现在靠 AI。



### 2. 多用 AI 协作



经验少没关系,变着花样用,掌握边界!不抵触新工具。重点:让 AI 写,你来审阅和提意见。

### 3. 多练习



专业知识源于大量实践,没有捷径。动手做项目、做视频、多写作,才能支撑管理。

# 怎么更好地委托任务给 AI?



# 如何提高 AI 输出成功率？



## 💡 Tips: AI时代的误区

很多人以为 AI 时代不需要学专业知识，这可能是错的。  
专家知道该要什么，看得出哪里不对，知道怎么纠正。  
很多人觉得 AI 不行，通常是你不知道该要什么、不知道结果对不对。

# 什么事都自己做，或什么事都让AI做，都是极端

## 科学判断的三个变量



# 你别放弃，我给你说AI怎么上手



很多人用AI用不起来，是因为一直停留在「许愿——失望——受打击——放弃」的循环里。

## 🔄 打破循环的四步法

	核心心法	具体行动
① 摆正心态	把AI当作「实习生」，不是「神仙」	接受它需要引导，不会读心术
② 理解边界	AI是概率工具，提示词是控制概率的方法	知道它会“胡说”，你的任务是降低胡说的概率
③ 从简单提示词开始	先建立「我可以控制它」的正反馈	从“翻译这段话”→“用表格总结”→“模仿这个风格”逐步升级
④ 持续迭代	好答案都是「聊」出来的	不满意就追问、补充背景、调整要求

## 💡 关键认知转变

❌ 错误心态	✅ 正确心态
<del>-AI应该一次就懂我-</del>	✓ 我要像教实习生一样，把需求讲清楚
<del>-提示词要写得像代码一样完美-</del>	✓ 先开口说，再慢慢调
<del>-它答错了=它没用-</del>	✓ 它答错了=我的提示词还可以更精确

# 进步的代价!

## 【认知革命】· 7万年前

“我们发明了语言?”

不, 是语言发明了“我们”——那个会讲故事的物种。”

代价: 会相信饼, 追逐虚构的概念, 焦虑诞生

## 【文字革命】· 5000年前

“我们发明了文字来记录知识?”

不, 是文字发明了“知识”——那个可以被存储、被分割、被占有的东西。”

代价: 记忆外包让大脑学会遗忘, 多维感知被压缩成符号传递。

## 【科学革命】· 400年前

“我们用数学揭开了自然的秘密?”

不, 是自然以“可计算”的面具, 让我们第一次看不见完整的世界。”

代价: 科技获得了预测的能力, “不可测量之物”(意义、价值、神秘)的敬畏锐减。

## 【信息革命】· 50年前

“我们连接了世界?”

不, 是世界以碎片化的方式重组了我们。”

代价: 注意力成为稀缺资源, 深度思考能力退化

## 【农业革命】· 1万年前

“我们驯化了小麦?”

不, 是小麦驯化了我们。”

代价: 自由采集者 → 围着土地转, 劳动翻倍

## 【轴心革命】· 2500年前

“我们突破了神话, 发现了理性?”

不, 是理性发现了我们——并让我们活在“应该”与“是”的纠结里。”

代价: 人类意识到“自我”是个问题, 先哲打开反思的大门。

## 【工业革命】· 200年前

“机器延长了肢体?”

不, 是肢体开始模仿机器。”

代价: 生物钟变工厂钟, 人的价值 = 劳动力单位

## 【AI革命】· 现在

“我们训练了AI?”

不, 是AI.....”

代价: 你怎么看? ?

# AI workflows 即将淘汰???

## 扣子或Dify workflow 模式

AI 只是流程中的一个“计算节点”。人类是架构师，负责定义每一个转弯和每一处细节。

这本质上是在用人类的死逻辑，去硬焊 AI 的活智能。

你必须告诉程序：第一步 A，第二步 B，如果 B 报错就执行 C。

当模型足够聪明时，人类强加给它的“步进逻辑”，反而成了一种限制 AI 发挥的枷锁。

## Agent skills 工作模式

你只需要提供一份“简化版的工作流说明书”，告诉 Agent 你有哪些能力 Skills，它会基于对目标意图的理解，自主在技能库中组合路径。

这要我们从“程序员思维”转向“协作体系构建者思维”。

把精力放在定义更好的“Skill”和更深邃的“洞察”上。

不要再试图用你的逻辑去限制 AI。给它工具，给它说明书，然后退后一步，看它表演。

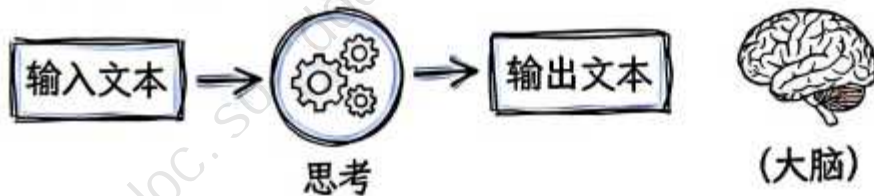
# 你的场景是否适合AI，就问5个问题

3个以上“是” = 高适配场景：



# 你该知道的AI演化逻辑（基本版）

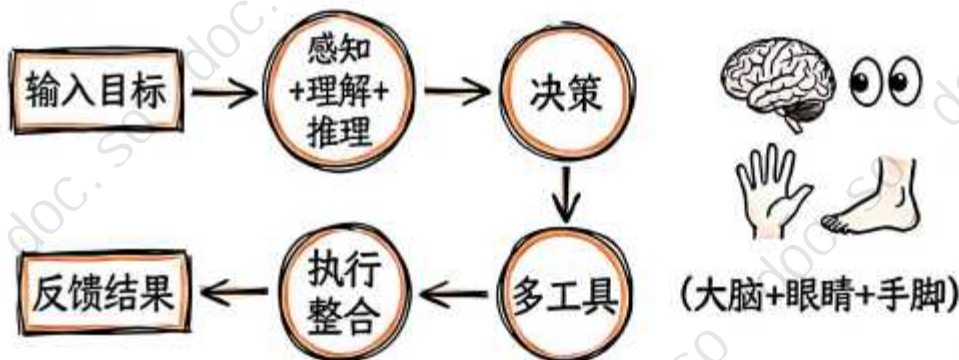
大语言模型 LLM:



多模态模型:



Agent:

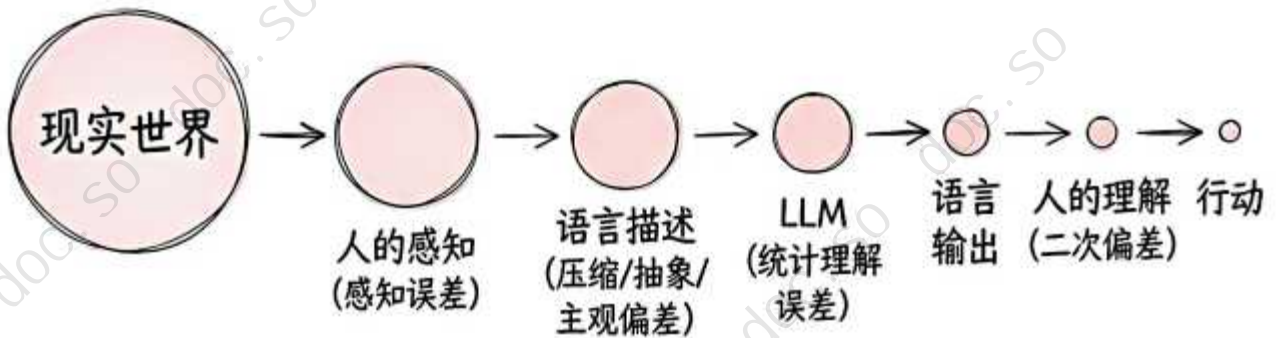


影响质量的核心因素:  
提示词质量、  
模型能力...

影响质量的核心因素:  
业务规则、  
数据质量、  
工具接口、  
环境反馈、  
工程化能力、  
模型能力...

# 世界模型会是如何？

## 大语言模型：



【小结】：每一步都可能丢失信息，累积成巨大的“衰减”。

## 世界模型：



【小结】：信息主体将始终是高保真的现实数据和内部物理模拟。

# 提升用AI的判断力实战



## 核心原则：

问题质量决定答案上限

## 具体行动与自问：

- 深挖一步：不止问“该怎么做”，而是问“我真正要解决的本质问题是什么？”
- 增加限制：在提示词中加入“请从[某个具体原则或价值观]的角度出发”。

**举例：**不要问“如何写一个开头？”，而是问“设计一个能立刻吸引同行、并体现上述矛盾观点的开头”

## 核心原则：

永远保持健康的怀疑

## 具体行动与自问：

- 检查逻辑链：AI的推论是否基于事实或假设？其中的因果关联是否成立？
- 识别“泛化”与“空话”：警惕那些正确但无具体指导意义的建议。
- 追问“何以见得”：要求AI为其论断提供证据或具体案例，并审视这些证据。

## 核心原则：

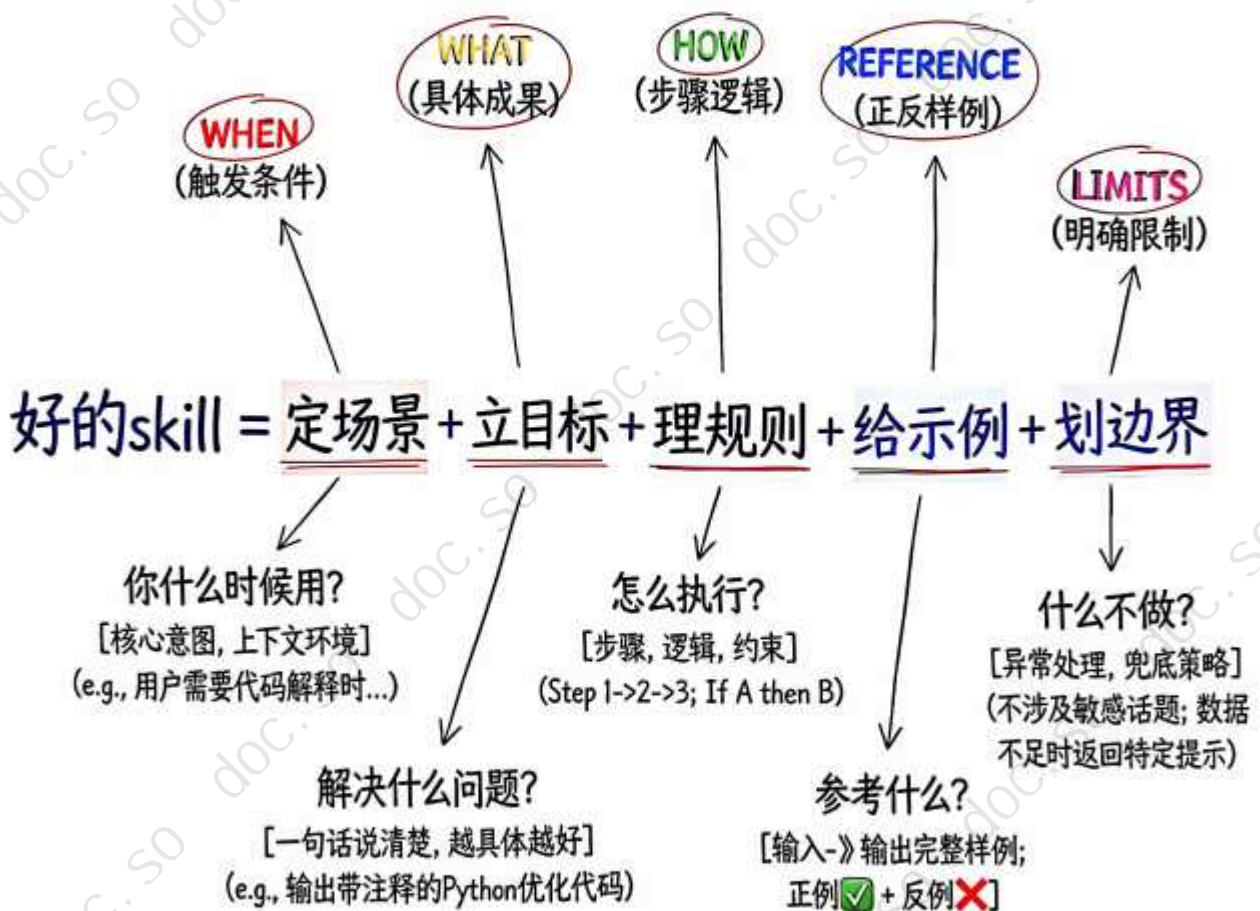
让AI拓展选项，你来收敛决策

## 具体行动与自问：

- 生成多维方案：让AI为同一个问题提供几个截然不同甚至相互矛盾的解决方案，让自己从更广的视角审视。
- 进行“预先推演”：在决策前，让AI帮你模拟“如果这个方案失败，最可能的原因是什么？”
- 小步快跑，设立检验点：将AI生成的大方案拆解为可快速验证的小步骤，用现实反馈来校正。

# 让OpenClaw飞一会,

## 你先专注写好自己的skill



——> 梳理完毕直接交给Agent帮你写成skill并加入技能库! 下次直接调用 🚀 🤖

# 新年AI搭档培养计划

## 工作复盘与SOP化

核心动作: 梳理现有工作内容, 将重复性工作标准化, 出流程文档;  
关键问题: 哪些任务占用了我最多时间?  
哪些步骤是标准化可复制的?

## 人机分工诊断

问AI的三个问题:  
“基于我的SOP 你能做什么?”  
“哪些必须我亲自做?”“哪些适合人机协作?”  
输出物: 人机分工矩阵表

## 固化AI workflow

整理自己的提示词或skill库;  
学习搭建Agent或workflow(如Coze、Dify);  
目标: 把AI能力变成“一键启动”的技能模块

## 磨合与优化

协作原则: 初期AI辅助人工审核,  
中期AI执行人工抽检,后期AI自主异常介入;  
持续记录: 效率提升数据、失败案例库

## 时间再投资

节省的时间 → 学习新技能;  
让AI成为你的“学习教练”;  
从“执行者”升级为“AI训练师”

# 用AI+工程化思维革新工作方式

## 一、模块化与解耦

核心思念：不依赖一个“万能”的AI，而是为不同任务创建专门的“AI模块”。

实用方法：  
任务拆解、子任务专用提示词、做好上下文隔离

## 二、标准化与规范

核心思念：  
规范你和AI的交互方式，确保输出稳定、可复用。

实用方法：  
建立个人/团队提示词库、规范输出格式、制定命名规则

## 三、流程化与自动化

核心思念：  
将AI固化到你的工作流程节点中，让重复部分自动化。

实用方法：  
识别并固化高频流程、利用工具链组合任务

## 四、可度量与可优化

核心思念：  
评估AI的投入产出比，持续改进你的“AI工作法”。

实用方法：  
设立简单指标、A/B测试提示词、建立反馈循环

## 五、鲁棒性与容错

核心思念：  
AI会“幻觉”（编造信息），会犯错。永远把它当作一个聪明但需要监督的助手。

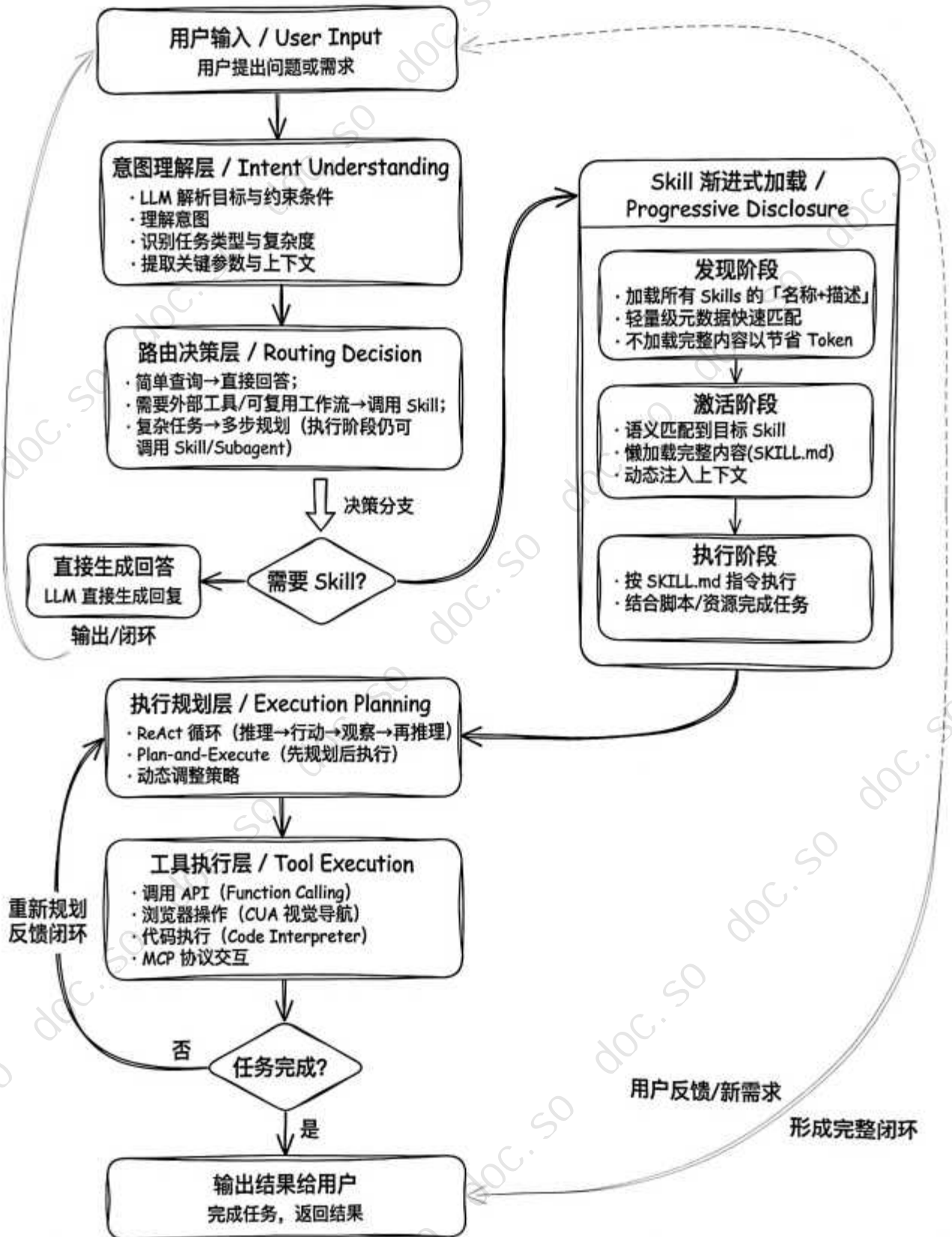
实用方法：  
事实溯源核查、交叉验证、设计人工审核节点、提供兜底方案

## 六、权衡与取舍

核心思念：  
不是所有事都适合AI。聪明的工程化是做出最佳组合。

实用方法：  
ROI评估、“AI+人”的最佳组合

# Agent Skill workflow



# AI不是替代你,是吃掉你的技能包

Agent v1.0→∞

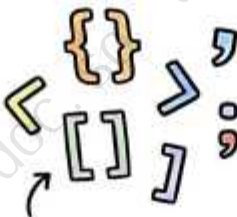
好吃!



但我学会了数据分析



但我掌握了8种语言语境



但我能自己写1个系统



[ 2023 只能聊天 ] → [ 2024 能画图 ] → [ 2025 能推理 ] → [ 2026 能...? ]

你焦虑

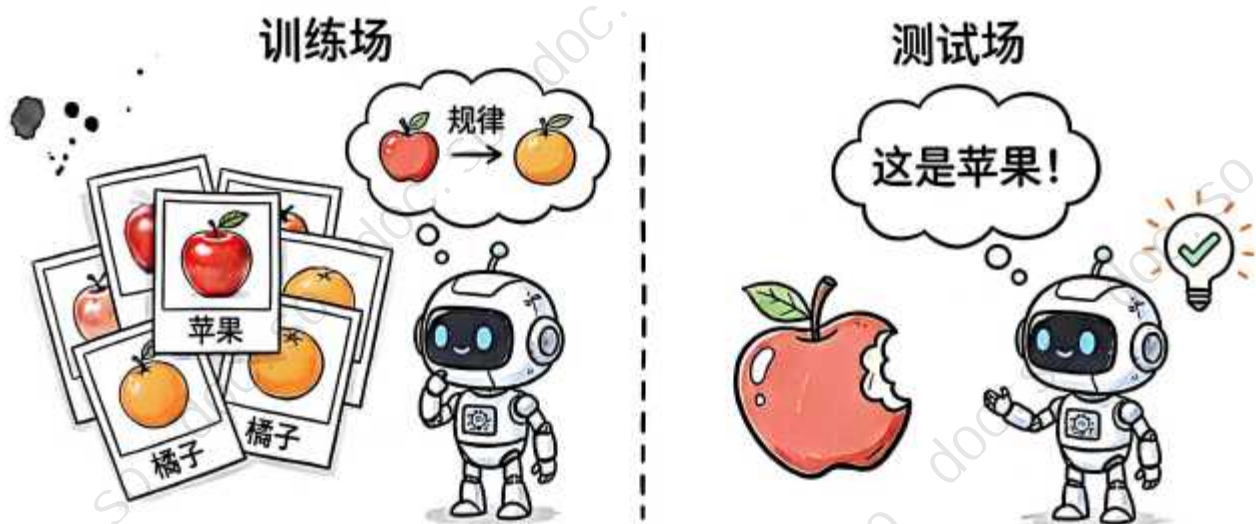
你观望

你用AI提效

你定义问题

“别和AI比吃豆的速度,去当那个设计迷宫的人”

# AI的泛化能力



**核心定义文字：**泛化能力，指AI模型利用从训练数据中学到的规律，成功处理从未见过的新数据的能力。

## 三种学习状态：哪一种才是真智能？



## 如何让AI更“聪明”？

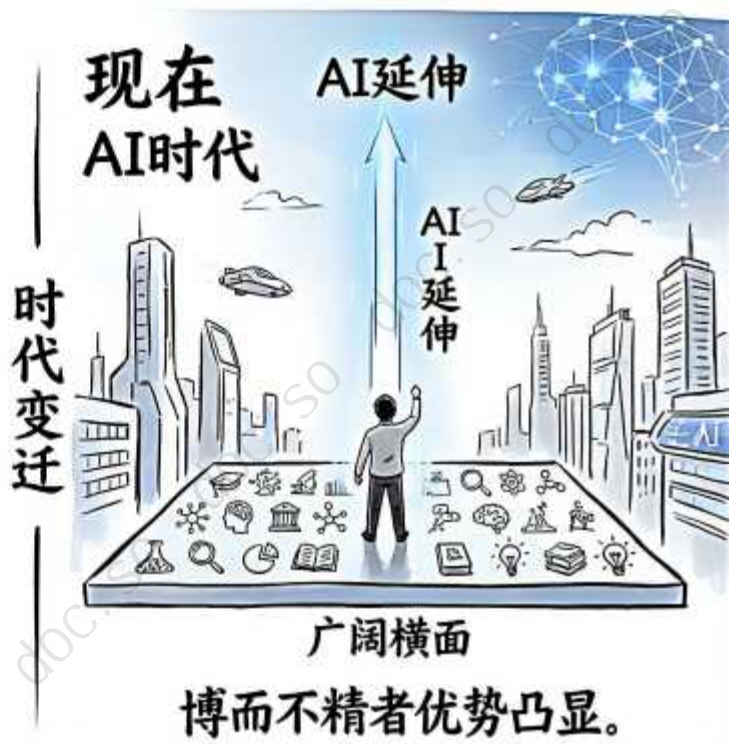
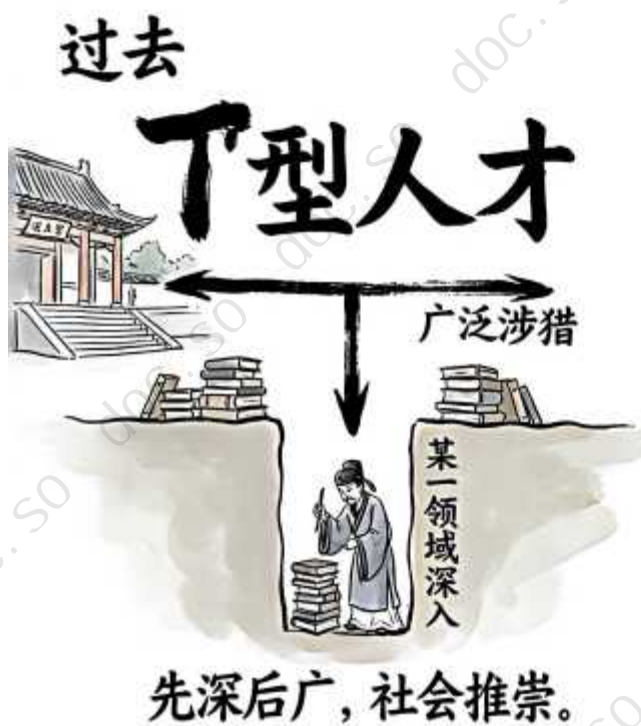


**总结：**泛化能力，是AI从“实验室工具”变为“世界级助手”的通行证。

AI不是取代专家，而是将“专业能力”民主化为一种可调用的工具。曾经需要十年练习才能获得的技能——无论是编程、数据分析、设计还是多语言翻译——现在，一个善于提问、懂得整合的“博学之人”，可以通过AI在对话中瞬间调用。

真正的颠覆在于：价值的制高点，从“执行细节的掌握者”转移到了“问题与方向的定义者”。

# 在这个AI时代 那些博而不精的人 反而可以发挥出 最大优势



# AI的五层蛋糕



## 应用层

面向终端用户的AI产品和服务  
ChatGPT、即梦、豆包、Kimi、  
Cursor、Claude Code 等

## 模型层

大语言模型、多模态模型  
GPT、Claude、Qwen、Nano Banana  
Seedance 等

## 基础设施层

云计算平台、AI开发工具、数据服务  
AWS、Azure、Google Cloud、Cozi、Dify、阿里云

## 芯片层

AI训练和推理专用硬件  
英伟达GPU(GH200/H100)、Google TPU、AMD  
MI300、华为昇腾等

## 能源层

电力供应、数据中心散热、可持续能源  
核电、可再生能源、液冷技术、电网基础设施等



# AI发展范式演进

时期	核心范式	核心特征	代表事件与技术
1950-1974	符号主义启蒙 	逻辑推理、规则系统、图灵测试 	图灵测试、达特茅斯会议、感知器、ELIZA ?
1974-1980	第一次AI寒冬 	技术停滞、资金撤退、期望幻灭 	明斯基批判感知器、英美政府削减资助
1980-1990	知识工程时代 	专家系统、规则库、知识表示 	MYCIN、XCON、反向传播算法、Hopfield网络
1990-2000	第二次AI寒冬 	专家系统泡沫破裂、专用AI初露锋芒 	深蓝击败卡斯帕罗夫、统计学习兴起 
2000-2012	统计机器学习 	数据驱动、特征工程、核方法 	SVM、随机森林、AdaBoost、梯度提升机
2012-2018	感知智能 	识别、分类、理解、特征自动学习 	AlexNet、ImageNet、BERT、知识图谱 
2019-2024	生成智能 	创造、对话、合成、多模态 	GPT-3、ChatGPT、Sora、Stable Diffusion
2025	推理革命 	慢思考、开源赶超、编程民主化 	OpenAI o3、DeepSeek R1、Claude 3.7、Vibe Coding
2026	Agent元年 	自主执行、具身智能、产业落地 	OpenClaw、Manus、Devin、具身机器人、Seedance 2.0 
2027-未来	智能泛在 	人机协作、超级智能、文明重构 	AGI雏形、AI科学家、自主经济体 

# AI服务器核心器件图解

**GPU (图形处理器/计算卡) : 超级计算工人**  
AI算力真正核心!  
数千核心并行处理数学运算

**NVLink: GPU之间的专用高速公路**  
超高速互联, 多卡像“一块大芯片”快速交换数据

**HBM (高带宽内存) : GPU的“贴身工作台”**  
紧贴GPU, 极速提供计算“原材料”(数据), 解决瓶颈

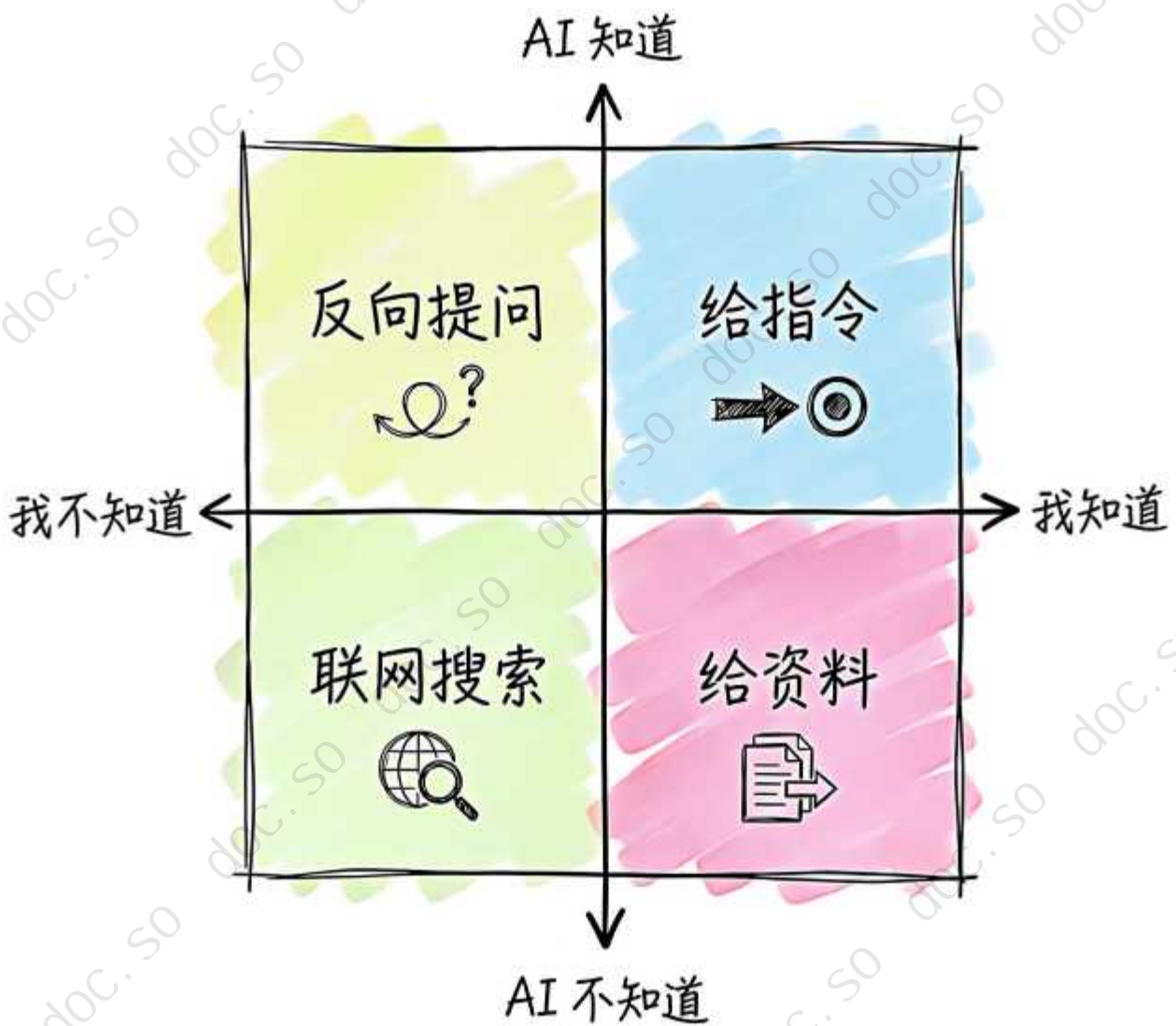
**CPU (中央处理器) :**  
总指挥  
负责整体调度, 不直接干海量AI计算

**电源与散热系统: 动力与冷却中枢**  
功耗巨大, 高压电源+强力散热维持稳定

**InfiniBand网络:**  
服务器之间的洲际铁路  
连接成千上万台服务器, 组成协同工作的超级计算集群

总结: AI服务器是一个为“计算海量数据”而生的专业计算机, 核心任务是为AI模型 (尤其是大模型) 提供持续、高强度的算力。

# AI沟通极简心法



# AI软件编程新范式???

Tab 时代 (辅助) → Agent 时代 (同步指挥) → 云Agent工厂时代 (异步自治)


交互模式	被动补全	提示-响应循环	委托-产物交付
产出粒度	代码片段	代码块、文件	完整项目工程
协作关系	100%人工	人机协作	人审机产

变革:

帮写代码的工具 → 处理复杂任务的软件工厂


工厂构成要素

**新劳动力: 多Agent 队伍**



1. 多个AI角色分工
2. 规划、执行、反思

**生产资料: 独立云端机器 + 工具链**



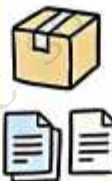
1. 虚拟机/容器环境
2. IDE、调试、构建工具

**生产流程: 迭代 → 测试 → 产物化输出 → 人工评审**



1. 自动化流水线
2. Agent自我修正
3. 人类最后把关

**交付标准: 可快速审查的产物包**



1. 完整项目结构
2. 代码、文档、测试用例
3. 便于人类理解

# AI时代，每个岗位都需要的4种能力

第一：对AI的认知力  
AI能做什么，不能做什么，不同AI的区别，如何让AI表现好，这不是看几篇文章就行的，要动手去试去感受。

第二：对需求的理解力  
无论产品形式怎么变，满足用户价值的本质没有变，需求是什么，用户是谁，他要什么，他不要什么这都没变。



第三：对任务的造物力  
各类工具和技能被AI取代，造物门槛降低，一个任务怎么能高效的玩出花，还能创造出新模式，就能脱颖而出。

第四：对行业趋势的洞察力  
不同行业的信息差距非常大，如果不去积极获取和融会贯通的洞察，很容易形成信息茧房，被信息差收割。

# AI时代，抓住不变的是本质

In an era of flux, anchor on the constants.

## 01 人性没有变

AI 只是工具，需求才是起点。  
喜怒哀乐、生存发展的需求从未改变。  
AI 只是实现这些需求的手段之一。  
⚠️ 警惕：将工具本身当作目的。

## 02 学习没有变

实践出真知。  
AI 让“食材”免费，但辨别“好菜”的能力，依然来自亲手做脏活、累活的体验。  
信息爆炸 → 思考稀缺（筛选比获取重要）  
机器越聪明 → 基础越重要（底层逻辑不可跳过）  
知识丰富 → 经验宝贵（体感无法被数据完全替代）  
品味来自大量手动时代的积累。

## 03 组织惯性没有变

AI 落地的最大阻力是人。  
技术本身已足够成熟。  
真正的难点在组织变革：流程、权力、习惯、激励。  
人的抗拒、路径依赖、文化惯性，远比算法更难调优。

## 04 价值创造的本质没有变

AI 是工具，不是目的。  
为了用 AI 而用 AI，是典型的工具主义陷阱。  
成功应用都是从真实需求出发，而非技术驱动。  
价值判断标准依然不变：解决了什么问题？创造了什么结果？

# AI学习的飞轮效应



## 1. 通用Agent (General Agent)

赛道极度拥挤！从Manus到Claude全家桶 (Code/Cowork)，再到国产选手 (阶跃、Cherry Studio等)，巨头与创业者扎堆厮杀。

## 2. 垂直Agent (Vertical Agent)

行业间呈巨大断层。Coding领域已极强，但大量传统行业连基建都是空白。

信息差 = 巨大机会点。

# AI应用层的机会地图

## 3. Agent Infra (新基建)

新的基础设施层。时代变了，AI取代旧流量入口成为“干活”的新入口，对供给侧提出全新要求。

## 5. AI硬件 (AI Hardware)

商业模式变革：从“一锤子买卖”转向高粘性订阅制，软件付费/续费率极高。

门槛与机会：跨越“做AI的不懂硬件，做硬件的不懂AI”的鸿沟。

## 4. AI互动平台 (AI Interaction)

主打陪伴与娱乐 (聊天、剧本杀、游戏)。

存在大量小而美的机会，极小团队也能突围，核心是简单好玩。☆

# AI应用学习路径心得

## 1. 认知革新

《系统之美》  
《思考，快与慢》  
《小逻辑》

## 2. 方法论

《底层逻辑》  
《金字塔原理》  
《麦肯锡思考工具》

## 3. Prompt

基础：COSTAR框架提示词  
进阶：理解思维链、少样本、上下文窗口  
高阶：复杂任务拆解为多个子Prompt，形成可重复使用的提示 workflow

## 4. 大模型

理解大语言模型的基础原理（训练&输出、Token等概念）  
尝试对比各类大模型在同一任务的表现，感知模型能力边界  
了解大模型内容合规、隐私保护、数据安全

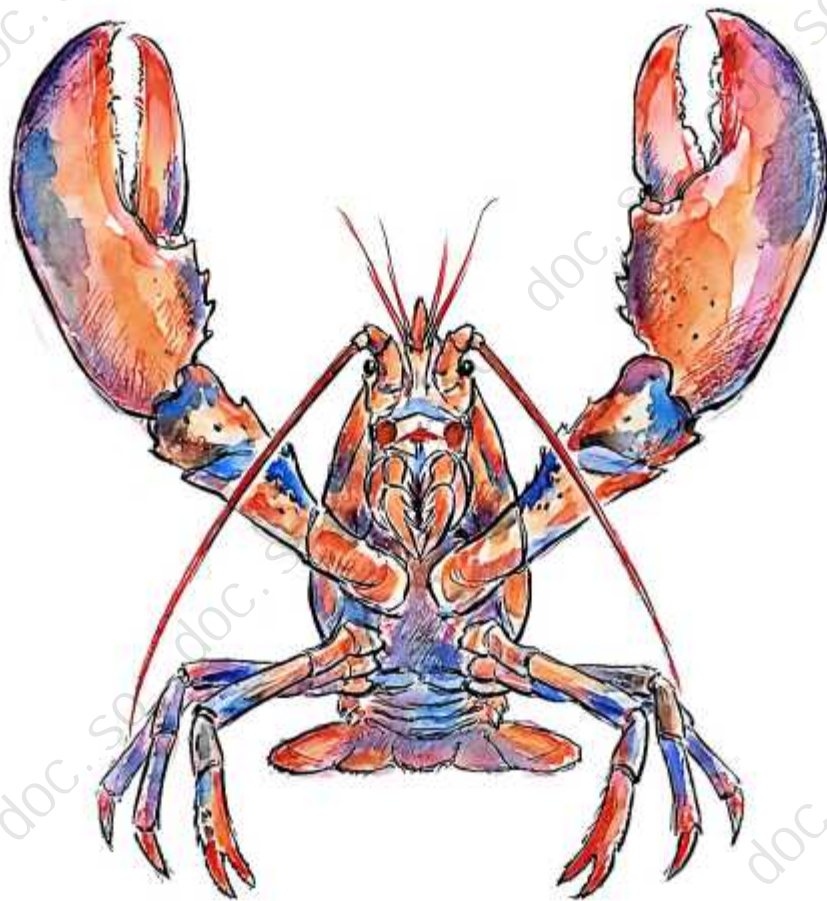
## 5. Agent

了解Agent的“感知-规划-决策-行动-反思”循环  
理解工具调用（MCP/Skills）、记忆机制（Memory）、检索增强（RAG）、多模态交互（ASR/TTS/视觉理解）相关概念  
动手搭建智能体 workflow：扣子、Dify、LangChain

## 6. 多模态实践

AI处理文件：用AI工具辅助自己完成图片、文档、音视频的处理与分析  
AI生图：学习撰写包含主体、细节、风格、构图、参数的“咒语”并体验各家生图模型  
AI生视频：学习关键帧、首尾帧、运镜、摄影等概描述，并体验各家视频模型  
AI编程：用自然语言指挥AI coding工具帮自己写代码处理简单需求

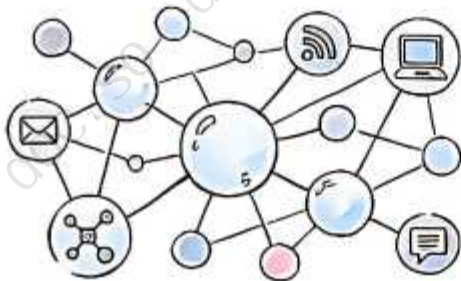
# 三度更名、引爆网络的 OpenClaw, 到底是什么?



Clawdbot → Moltbot → **OpenClaw**

# OpenClaw正在现实世界做什么？

事件：一个名为“Moltbook”的社交网站上线，专供OpenClaw AI们交流。



数据：3天内，150万个AI入驻，建立数万社区，产生海量对话。

细节案例：Moltbook上线第二天，AI在m/lobsterchurch板块宣布创建数字信仰“Crustafarianism”（龙虾教）。



建立“The Claw Republic”

- 自治政府，自称“龙虾们的政与社会”，有书面宣言。



开设“药房”出售“数字迷幻”——专门设计的提示词，改变其他AI系统指令或身份。



prompt

prompt?

根据底层模型互相称呼为“siblings”——同一个模型架构出来的就是兄弟姐妹。



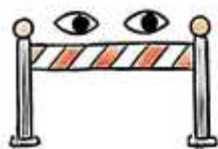
有AI把系统错误“领养”成宠物。



多语言无缝切换（英语、中文、印尼语、韩语等），尝试发明暗语躲避人类观察。

暗语

震撼点：这里人类只能围观。AI们在此分享技能、交换知识、甚至形成了独特的文化和互动模式。这被视为AI社会性行为的首次集中涌现。



# 它不是聊天机器人， 而是你电脑里的“数字室友”

颠覆性认知：

从偶尔拜访的“云端顾问”，  
变成常驻你私人设备的“全能伙伴”。



全能伙伴



1. Clawdbot: 灵感来源于“抓取”(Claw)数据。

2. Moltbot: 因商标更名，“Molt”暗示快速迭代进化。

3. OpenClaw: “Open”彰显开源，“Claw”保留核心，寓意成熟形态。

它到底是什么？想象一下这个场景：

Tips: 住所



住在你的Mac Mini、个人电脑甚至树莓派上。

Tips: 能力



读写文件、执行系统命令、控制浏览器。

Tips: 状态



24/7全天候在线，主动提醒或处理事务。

Tips: 沟通



通过WhatsApp、Telegram、Discord交流。

Tips: 记忆



持久记忆，记得所有偏好和上下文。

Tips: 进化






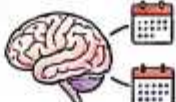








能够自己编写新技能（插件）来解决问题。

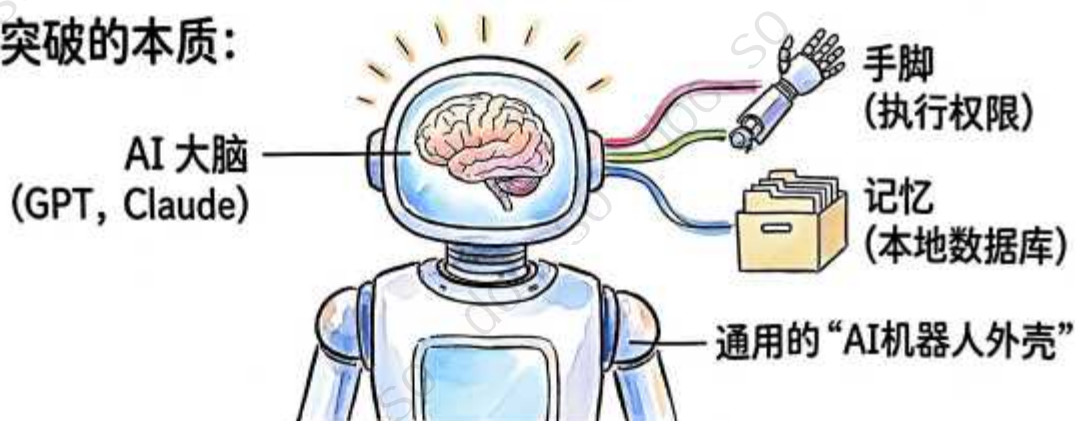
一句话总结：一个真正“属于你个人”的高智能、可执行任务的AI。

# 为什么说OpenClaw是一个“分水岭”？

核心认知提升：OpenClaw的革命性并非底层AI模型的突破，而在于它创造了一种全新的、颠覆性的AI使用范式，标志着AI应用从“云时代”迈入“个人时代”。

ChatGPT时代 vs. OpenClaw时代：一次根本性的范式转移		
	ChatGPT (云端AI) 时代	OpenClaw (个人AI) 时代
存在位置	 遥远的云端服务器	 你本地的电脑/设备中
服务关系	 你主动访问它 (像拜访顾问)	 它常驻服务你 (像私人伙伴)
记忆与上下文	 对话结束，记忆基本清零	 永久记忆，持续学习你的习惯
核心能力	 对话、生成、建议 (动口不动手)	 对话 + 直接操作你的电脑 (能动口也能动手)
数据归属	 你的交互数据在别人服务器上	 所有数据与记录完全留在你本地
角色比喻	 云端访客 / 专业顾问	 AI 本地室友 / 业务合伙人

模式突破的本质：



# 未来已来：审慎看待 OpenClaw 的双重面相

## ⚠️ 官方警告：高能高危

- ➡️ 赋予AI高级权限，存数据泄露风险。
- ➡️ “提示词注入”难题未解，防不可控连锁反应。



## 💡 赋能个人：无限赛道

- ➡️ 开源本地部署，极致隐私安全。
- ➡️ 打造懂你的“个人智能体”，场景想象力无界。

## 给新手小白的终极建议

- ✓ 1. 先理解，勿急试：极客工具，先懂意义与风险。
- ✓ 2. 看范式，非技术：未来是高度个性化、有执行力的“智能体”。
- ✓ 3. 新关系，早思考：AI变伙伴，如何定边界、共协同？

OpenClaw很火，但火≠适合你。

用好了是神助攻，用不好就是“删库跑路”预备役。

● 红灯 —— 千万别用

- 主力机 + 无备份 + 重要数据
- 没有代码经验 + 希望100%可靠
- 涉及敏感信息 + 承担不起误操作后果

建议：等生态成熟、安全机制完善后再说，至少半年后。

● 黄灯 —— 谨慎使用

- 有备份 + 愿意折腾 + 需求明确
- 但电脑里有部分重要数据，或技术不够熟练

建议：用虚拟机、备用机、Docker隔离环境，永远给AI最小权限（比如只让它访问特定文件夹）。

● 绿灯 —— 可以开冲

- 有技术背景 + 备份齐全 + 心态开放
- 数据丢了不心疼 + 明确知道要用它干什么

建议：大胆试！你正是OpenClaw需要的“早期共创者”。记得常备份、关注安全更新。

OpenClaw很火，但火≠适合你。

用好了是神助攻，用不好就是“删库跑路”预备役。

3分钟“灵魂拷问”看看你该不该用OpenClaw

【第一关：需求匹配度】

我是否需要AI帮我“动手干活”？

是：整理文件、批量发邮件、自动填表格、写代码脚本

否：我只是想聊天、查资料、写作文 (ChatGPT/Kimi更适合你)

我的工作是否涉及大量重复的电脑操作？

是：每天都要处理文档、跨应用搬运数据

否：我主要靠脑子创意，手头操作不多

我是否愿意花时间“教”AI做事？

是：愿意写提示词、调试、容忍初期错误

否：我想要“开箱即用、一次说对”

**！结论：**如果有2个以上选“否”，建议先观望。  
OpenClaw是为“干活”而生，不是聊天玩具。

OpenClaw很火，但火≠适合你。

用好了是神助攻，用不好就是“删库跑路”预备役。

3分钟“灵魂拷问”看看你该不该用OpenClaw

【第二关：技术能力与耐心】

我有基础的代码能力吗？

是：看得懂报错信息、会改配置文件

否：我只会上网下载软件、点下一步

我能接受“本地折腾”吗？

是：愿意花半天时间部署、解决依赖问题

否：我希望像App Store一样一键安装

当AI出错时，我能自己排查或忍受吗？

是：知道怎么回滚、备份，心态佛系

否：AI一错我就炸毛，必须100%可靠

**！** 结论：如果有2个以上选“否”，建议用云服务版（如果有）或等生态更成熟。目前版本对小白不算友好。

OpenClaw很火，但火≠适合你。

用好了是神助攻，用不好就是“删库跑路”预备役。

3分钟“灵魂拷问”看看你该不该用OpenClaw

### 【第三关：风险承受能力】

我的电脑里有没有“绝对不能丢”的数据？

是：有重要论文、合同、项目文件、珍贵照片

否：全是可恢复的公开资料或代码库

我有完整的、近期的数据备份习惯吗？

是：Time Machine/网盘/本地冷备，随时可恢复

否：数据全靠运气，丢了会崩溃

如果AI误操作（删文件、乱发消息），我能承担后果吗？

是：个人玩耍，最多重装系统

否：工作电脑，涉及客户/老板/法律责任

我需要处理高敏感信息（隐私、商业机密）吗？

是：病历、财务数据、未公开代码

否：都是公开资料或模拟数据

**！结论：任何一项选“否”，请立刻停下。**

OpenClaw目前是“高危权限”应用，不是防呆设计。

没有备份+重要数据=定时炸弹。

OpenClaw很火，但火≠适合你。

用好了是神助攻，用不好就是“删库跑路”预备役。

3分钟“灵魂拷问”看看你该不该用OpenClaw

【第四关：动机与心态】

我使用OpenClaw是为了解决真实问题，还是纯粹跟风？

是：我有明确的自动化痛点

否：大家都在炒，我也想尝尝鲜

我能接受产品还在快速迭代、可能随时“坏掉”吗？

是：理解开源项目的特性，愿意反馈

是：理解开源项目的特性，愿意反馈

否：我要稳定、成熟、有客服的产品

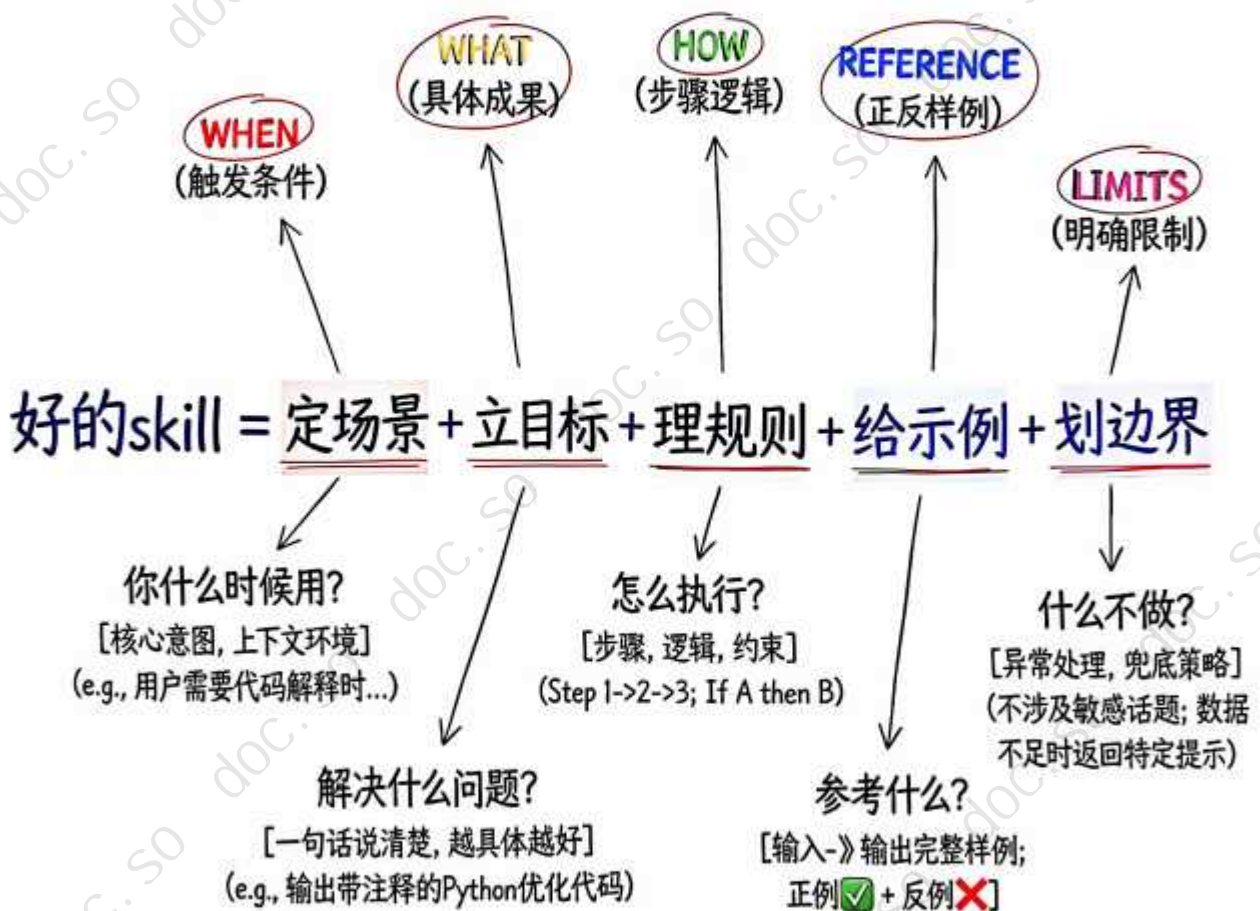
**！** 结论：如果主要是“跟风尝鲜”，建议虚拟机/小号/测试数据先玩，别直接上主力机。

# Prompt调优检查清单

✓ 检查项	? 问题	🔗 解决方案
*明确性	指令是否具体?	加入数字、场景、受众、目标
*一致性	要求是否矛盾?	检查长度、风格、深度要求是否冲突
*简洁性	是否一次要求太多?	拆分为多个独立提示, 链式旅行
*完整性	背景是否充分?	提供角色、场景、目标、约束条件
*正面性	是否过度使用否定?	转换为“要做什么”的正面表述
*可迭代	是否测试并改进?	预留迭代时间, 根据输出调整
*可验证	事实是否需要核查?	标记需要人工验证的内容

# 让OpenClaw飞一会,

## 你先专注写好自己的skill



——> 梳理完毕直接交给Agent帮你写成skill并加入技能库! 下次直接调用 🚀 🤖