手工川是如何使用 AI 的

为什么你的 AI 和我不一样



手工川 · LovStudio

四章地图 · CHAPTER MAP



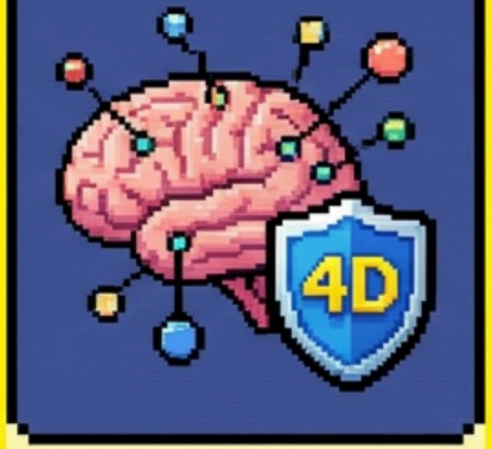
CH 1 · 实战场景分享

先让大家看到 AI 能干什么



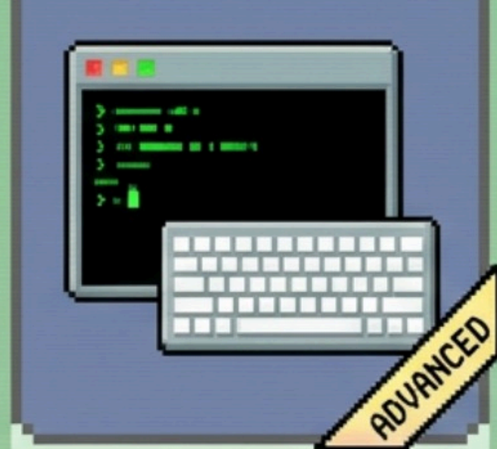
CH 2 · 上手教学

装环境 + 选工具 + 管 context



CH 3 · 人机交互方法论

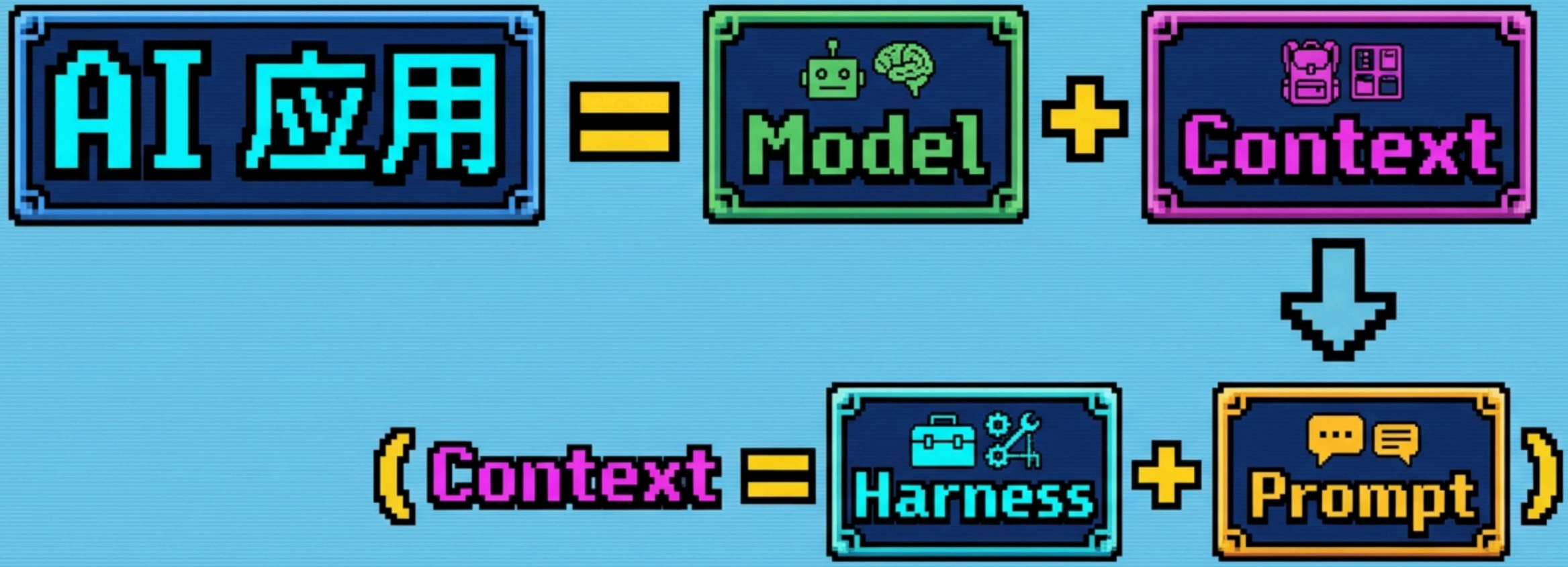
4D 心法 · 把 AI 当队友



附录 · 开发者配置

iTerm + tmux + cc · 已有点过时

推荐顺序：CH1 → CH2 → CH3，附录按需



决定你跑赢多少倍的不是模型 — 是 Harness 和 Prompt 一起构成的 Context

CH 1 · 实战场景分享

WHAT AI CAN DO FOR YOU

先让你看到 AI 能干什么 — 再讲怎么做到



7 大场景 · WORLD MAP



场景 1 · 做图



.svg → Sonnet 3.5（汉语新解 / AI 图腾）

COMPATIBILITY: HIGH



位图 → GPT-4 / Nano-Banana / GPT-Image

COMPATIBILITY: HIGH



.ai → 暂时还不行

COMPATIBILITY: LOW



海报 → 综合方案

COMPATIBILITY: CUSTOM

不同形式对模型能力要求完全不同



SYSTEM STATUS: ANALYZING

手工川

FINAL ✓



成品海报

Layer 5 · Logo / 二维码 → 直接 import（不用 AI）

Layer 4 · 主标题 / 副标题（重要） → 代码模型（HTML/SVG 精确字体控制）

Layer 3 · 排版结构 / 网格 / icon → 代码模型（Tailwind/CSS Grid）

Layer 2 · 装饰插画 / 主视觉 → 生图模型（艺术发挥）

Layer 1 · 背景纹理 / 渐变打底 → 生图模型（氛围 / 大色块）

怎么选引擎？

生图模型 · 何时用？

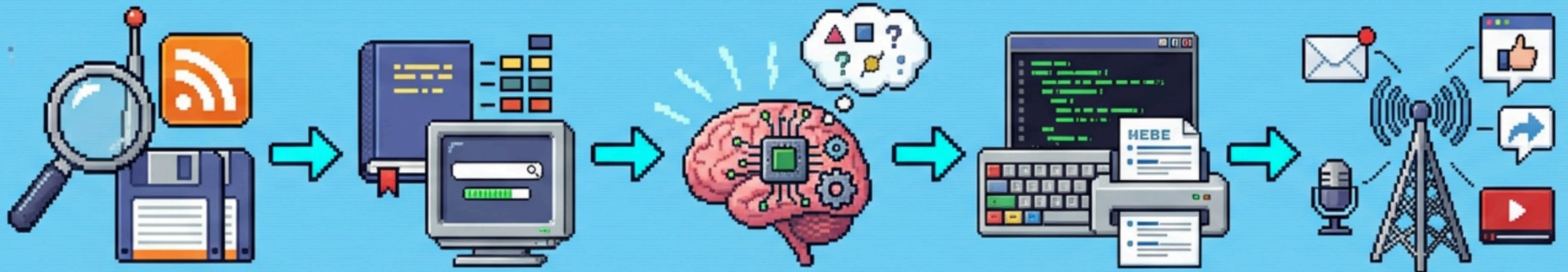
- 这层'画什么'难以用文字精确描述
- 需要艺术发挥 / 风格化 / 情绪
- 例：背景肌理、人物插画

代码模型 · 何时用？

- 这层'几何/字体/像素级精确'很重要
- 需要可改文字、可改色、可微调
- 例：标题、icon、网格

传统设计师 = 一个人扛所有层 · AI 时代 = 每层选最适合的引擎

场景 2 · 写文章 5 STEPS



1. 选题（私域 / RSS / 自选）
2. 深度调研
3. 认知渗透
4. 程序化排版
5. 多模态分发

SCENE 3 & 4 · PDF / BOOK



SCENE 3 · PDF

- BASED ON LOVSTUDIO/ANY2PDF,
- CASE 100+ PAGES CC SYSTEM REPORT



SCENE 4 · WRITE BOOK

- BASED ON WRITE-PROFESSIONAL-BOOK, 47 MINS RUN
- OREILLY PUBLISHING SKILL

FINISH

THIS PDF IS GENERATED BY ANY2PDF (META)

场景 5 · PPT · 3 PARADIGMS



传统 pptx（交付到 Office 生态）



基于 HTML（自己讲 / 演示动画）



端到端直出（这一份就是！）

本 deck 用 lovstudio-any2deck

SCENE 6 · WEB

SINGLE HTML

- TEMP DEMO
- SINGLE PAGE INTERACTION

STATUS:
LOCAL VS DATABASE
(SELECT ONE)

CASE:
DESIGN STYLE GUIDE /
NEONCLAW SUBTITLE SYSTEM

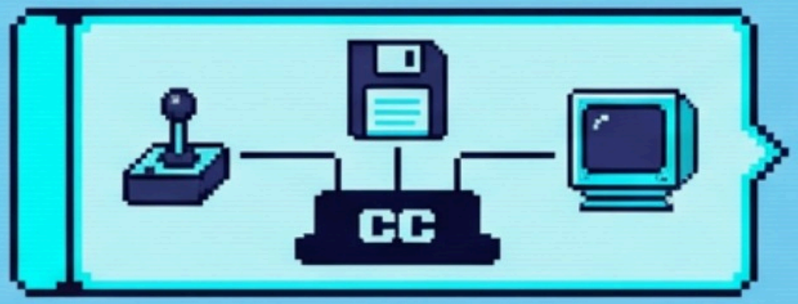
REACT FRAMEWORK

- LONG-TERM MAINTENANCE
- COMPLEX INTERACTION
  + LOVINSP

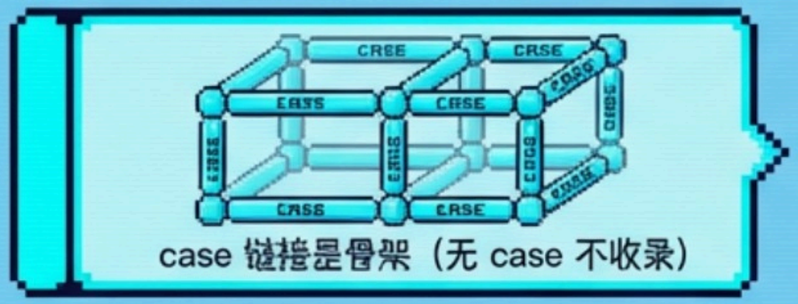
STATUS:
LOCAL VS DATABASE
(SELECT ONE)

CASE:
DESIGN STYLE GUIDE /
NEONCLAW SUBTITLE SYSTEM

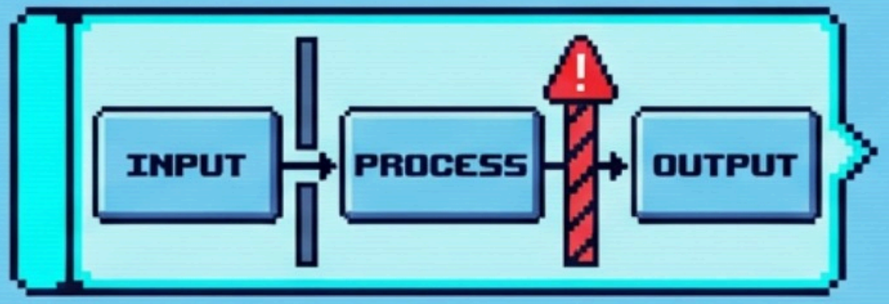
跨场景 · 3 KEY OBSERVATIONS



底座经常是 cc，周边各自不同



case 链接是骨架（无 case 不收录）



工作流的边界感很重要



CH 1 · TAKEAWAY

第一章 · 实战场景分享 · 小结

你刚看到了 7 件事















- **工作流的边界感** — 看似相近的场景深做差异巨大
- **case 是骨架** — 没有 case 的工作流不收录
- **cc 是底座** — 周边各场景挂的工具完全不同

下一章 → 上手教学 · 装环境 + 选工具 + 管 context

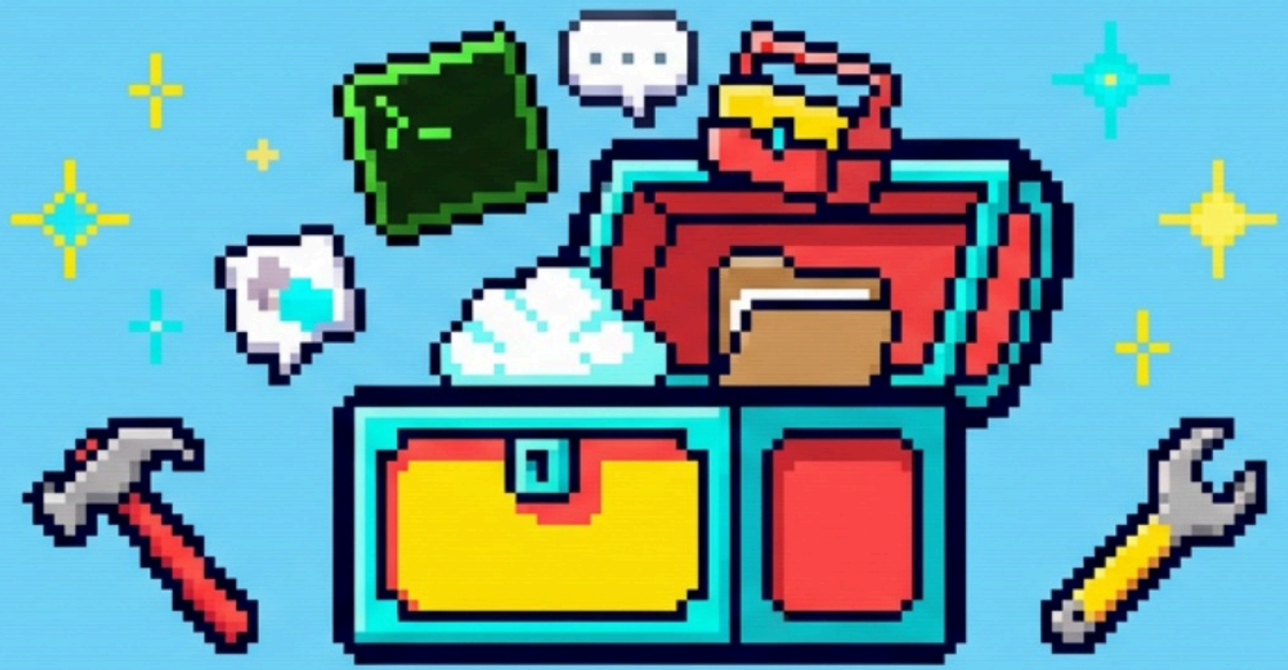
INTERMISSION

中 场 休 息

CH 2 · 上手教学

GET YOUR HANDS DIRTY

装环境 · 选工具 · 管 context —— 三件事打通就能复制我的工作流



OPEN YOUR TOOLBOX

LAYER 0 · 网络



VPN

第 0 层不通，上面四层全废

客户端 + 端口

ClashX
Mac 桌面常驻

Shadowrocket / Surge
移动端

```
HTTP   : 7890
SOCKS5: 7891
```

机场选型 3 RULES

1. 节点稳定性 >> 价格
2. 支持 IPLC 专线
3. 不要押注一家（双备份）

CC SANDBOX 不继承代理!

```
export https_proxy=http://127.0.0.1:7890
export http_proxy=http://127.0.0.1:7890
export all_proxy=socks5://127.0.0.1:7891
```

只在当前 shell export，不要写进 profile

LAYER 1 · DEV ENV · 5 STEPS

1. Xcode CLI Tools

```
xcode-select --install
```

2. Homebrew

```
/bin/bash -c "$(curl -fsSL …)"
```

3. Node.js (via nvm)

```
nvm install node
```

4. pnpm / bun

```
npm i -g pnpm/
curl -fsSL … | bash
```

5. Claude Code

```
npm i -g @anthropic-ai/claude-code
```

☑ 严格按顺序

CLI vs APP · 4 类 AI 编程工具的二维分析

Anthropic / Claude 系

OpenAI / Codex 系



Claude Code (cc)

我的主力 · CLI 之王

标签: SOTA dev experience



Claude App

GUI · 跨设备同步对话

标签: 普通用户友好



Codex CLI

命令行 · 跑测试 / fix

标签: 工程师对照组



Codex App / ChatGPT

GUI · 多模态 · canvas

标签: 大众市场最普及

CLI

APP

选哪一档? 看你是 CLI 派还是 GUI 派 · Claude 派还是 OpenAI 派

周边产品 · 各自适合什么场景

Cherry Studio



适合 多模型对照 / 横评 / 不想记 CLI 时用

LovCode



适合 看 cc 聊天历史 · 找回 session · feature 管理

LovClip



适合 复制图片自动给路径 · cc 里直接粘贴

LovSider



适合 浏览器侧边栏 · 当前网页一键转 markdown

LovOice



适合 Plaud / 飞书会议转写整理

LovNotifier

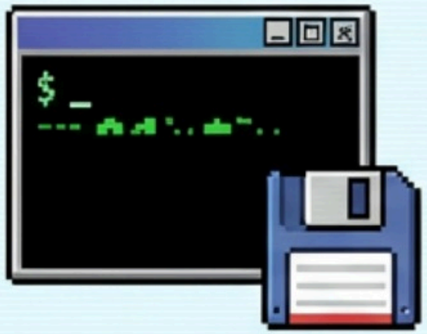


适合 cc 跑完通知到手机 / 桌面 / 微信

小工具不是 AI · 是让 AI 跟你工作流咬合的胶水层

LAYER 2 · AGENT RUNTIME

CLI



Claude Code / Codex
（开发者主力）

GUI



LovStudio Agent / Cherry Studio
（普通用户）

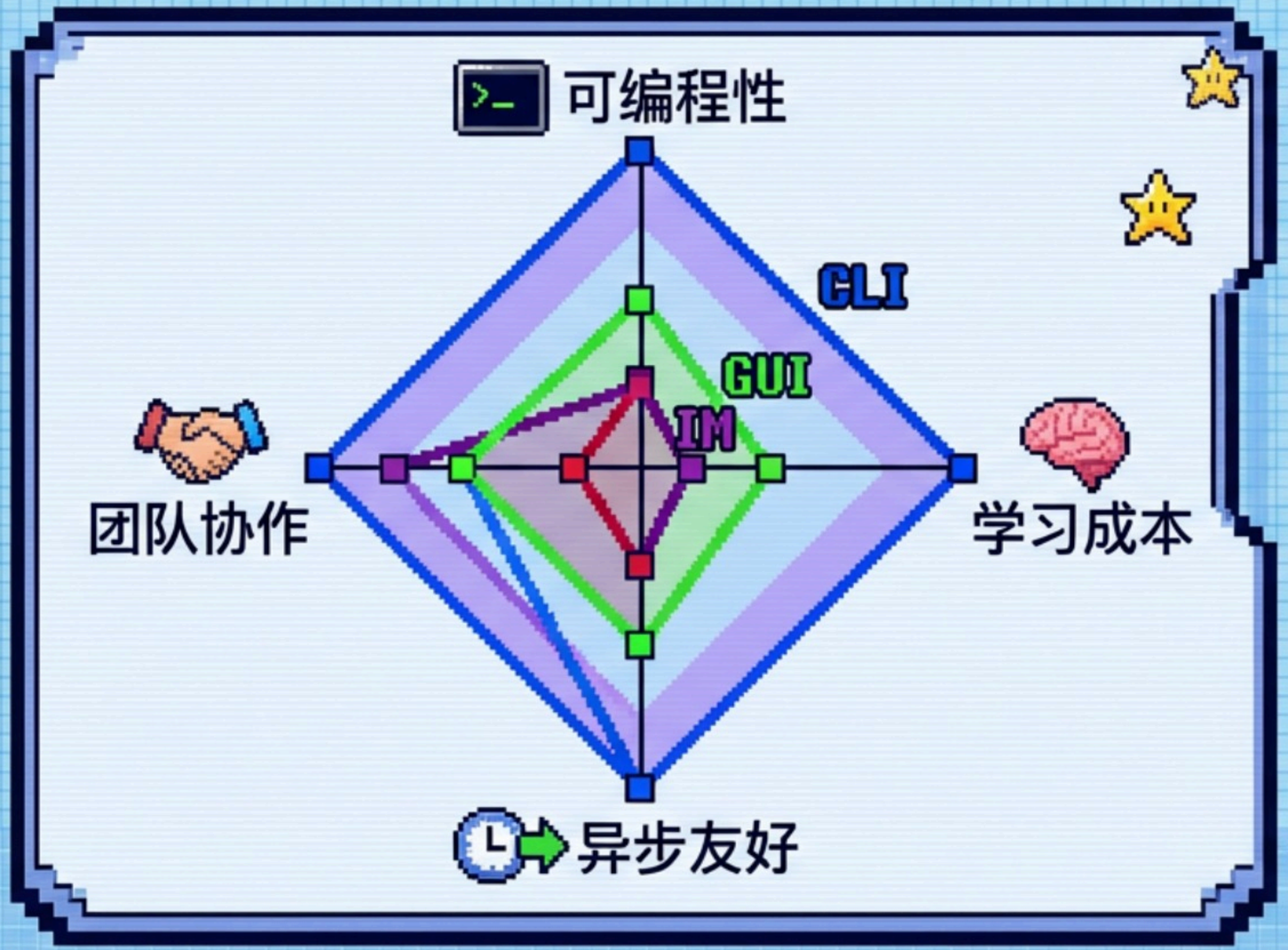
IM



OpenClaw / ClawBot
（飞书/微信里 @ AI）

维度	CLI	GUI	IM
易用性	低	中	高
灵活性	高	中	低
集成难度	高	中	低

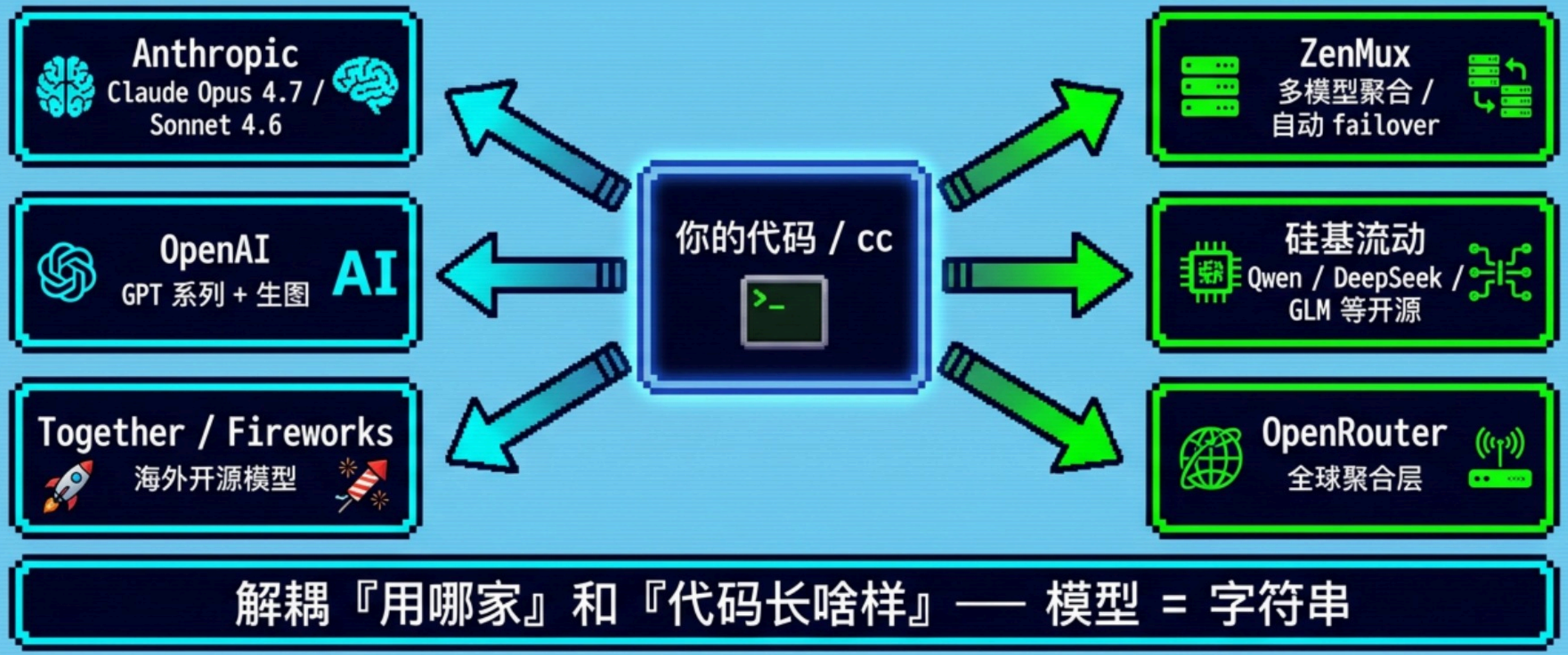
三档 · RADAR CHART



我的实际比例：90%

CLI + 5% IM + 5% GUI

LAYER 3 · MaaS · 模型即服务



解耦『用哪家』和『代码长啥样』—— 模型 = 字符串

```
client.create(model="anthropic/claude-sonnet-4.6")
                    ↓ 换一行字
client.create(model="openai/gpt-5")
```

我的默认档 · DEFAULT MODELS

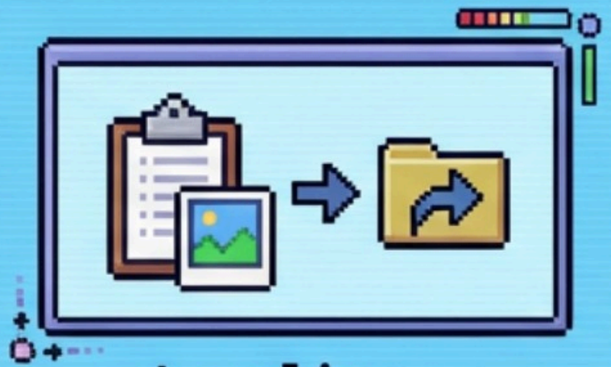
	文本默认档	anthropic/claude-sonnet-4.6	SOTA · 性价比 ✓
	文本升级档	anthropic/claude-opus-4-7	深度推理 / 长文 / 编码审查 ⇧
	生图默认	nano-banana-pro · gpt-image-2	文生图 / 图生图
	不默认	flash · haiku · mini · nano	只有明确"省钱/快"诉求才降级 ✗

CONFIRMED ✓

LAYER 4 · LOVSTUDIO TOOLBOX



Lovcode —
看 cc 聊天历史



Lovclip —
复制图片给路径



Lovsider —
网页一键转 md



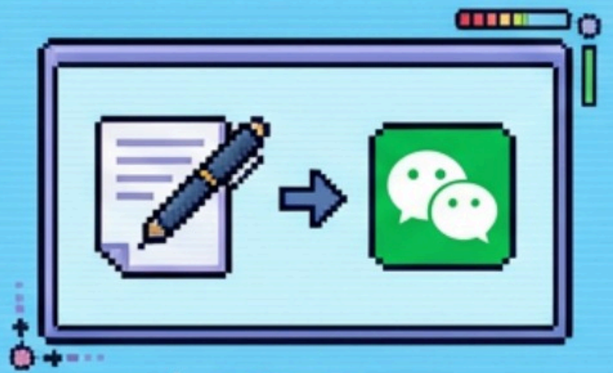
Lovinsp —
Click-to-code



Lovoice —
字幕/转写整理



Lovnotifier —
三路通知冗余



Lovpen —
markdown → 公众号

CH 1 · TAKEAWAY

第一章 · 安装与环境配置 · 小结



1. 🌎 第 0 层不通，上面四层全废

2. 🔧 cc 是入口，但它只是'执行器'

3. 🧩 真正的差异化在第 4 层 —— 周边 GUI 把 cc 编织进我的工作流

下一章 → Context Engineering · 第一性原理

CONTEXT · 4 SOURCES

① 

大模型内部（pre/post-train）

② 

外部获取（联网/本地/skill）

③ 

记忆（global / project memory）

④ 

Session（当前对话窗口）

DEBUG AI · 先问哪类 context 缺

不知道某事实 → ①/② 缺



重复问已说过的 → ③ 缺



长对话失忆 → ④ 出问题



90% 是 context 不行，不是模型不行[

积累 · ACCUMULATE



录音

PLAUD / GET 笔记



会议软件

腾讯/飞书/ZOOM 自动转写



笔记软件

OBSIDIAN / 飞书文档



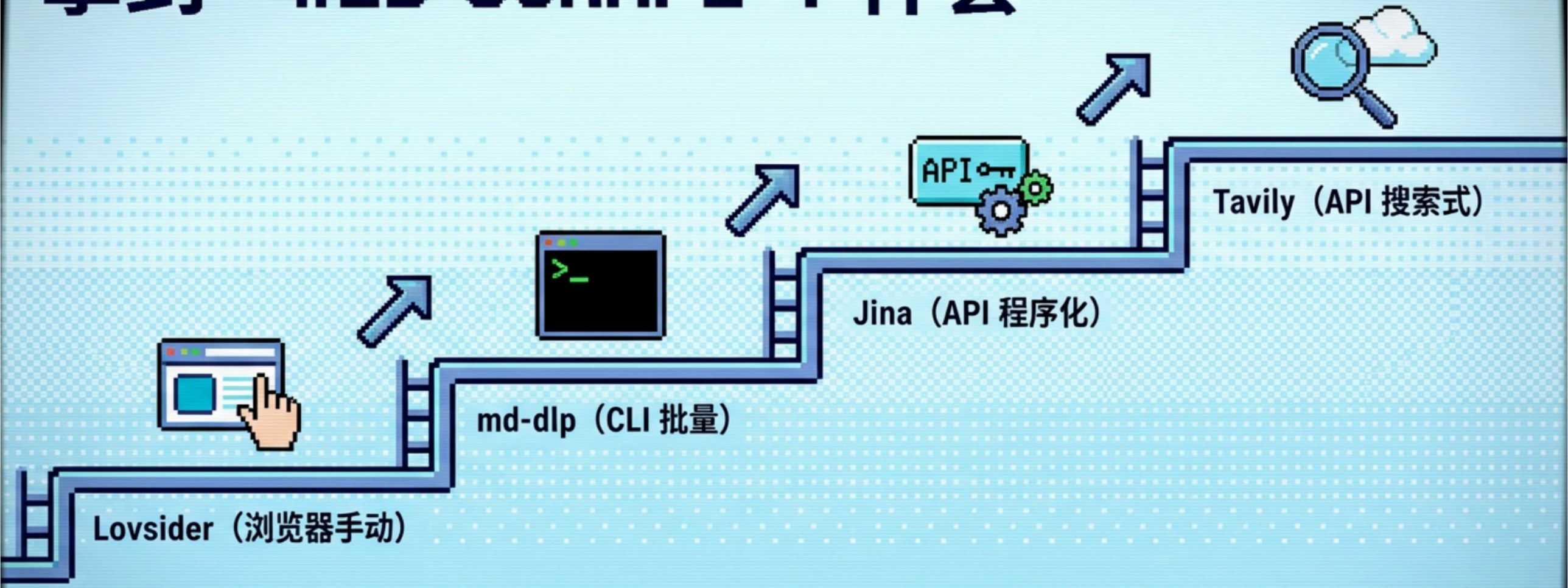
RSS

FOLO + WXMP-CRACKER

像素 TIP：积累过头≠浪费

拿到 · WEB SCRAPE 4 件套

手工川



手动收集，低频/定制化

批量下载，本地处理

程序化集成，动态内容

搜索整合，实时信息

DEEP RESEARCH · 替你打开 50 个 tab



替代的不是搜索 — 是"周末花一下午做纪要"

Claude Cowork
当前最顺手 (cc 内置)

`/deep-research`
自定义 skill 形态

Karpathy autoresearch
学术界版本

deepwideresearch.com
在线轻量替代

没官方 API 的平台，怎么抓



公众号

wxmp-cracker（手工川自研 skill）

处理登录态续期 / 按公众号批量抓



小红书

xiaohongshu-mcp

社区造的 MCP server



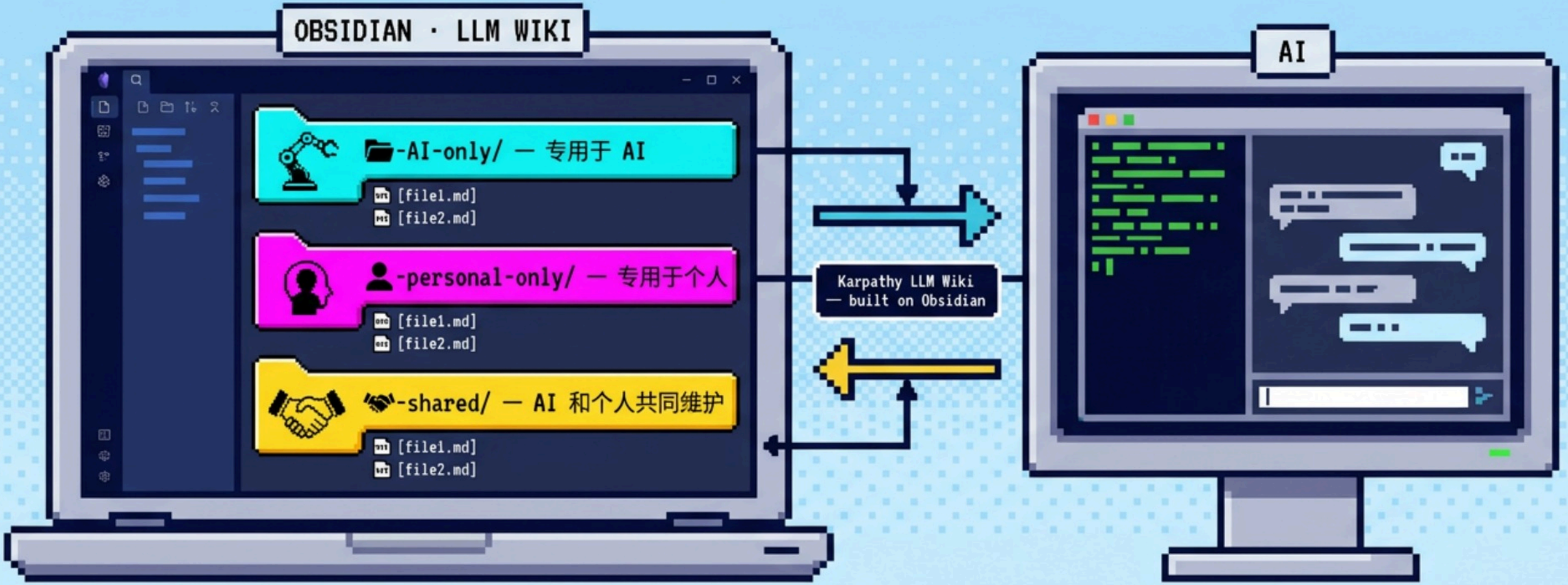
视频号 ⚠

TG 灰产渠道（仅供研究 · 风险自担）

官方 0 API · 主流爬虫被秒杀 · 部分 Telegram 群有可用接口

拿不到 ≠ 真拿不到 — 灰色地带工具决定了某些场景能否进 AI 工作流

PARA → LLM WIKI · 知识库范式切换



扁平 · 长 markdown · 给 AI 读，不给人导航

MEMORY · 2 TRACKS

cc 生态（开发者）



C 端产品（普通用户）



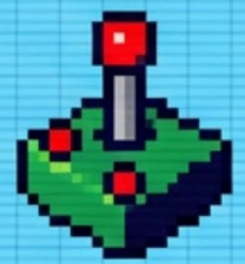
我自己：两边都用，分工不同

SESSION 命令 · CHEATSHEET

```
> /clear    —— 完全切话题
> /compact  —— 同主题压缩
> rewind    —— 回到某点重来
> /fork     —— 分叉两个方向
> /btw      —— 临时插入补充
```



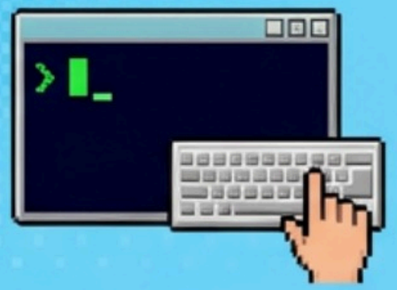




AUTO-CONTEXT · 反向优化



看当前 session



再写下一条 prompt

把 session 当成可塑形的对象，不是流水账



补什么



剪什么

决定补什么 / 剪什么

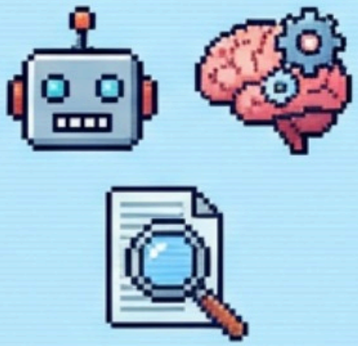
手工川公众号原文链接占位

PROMPT · 懒 vs 勤

LAZY

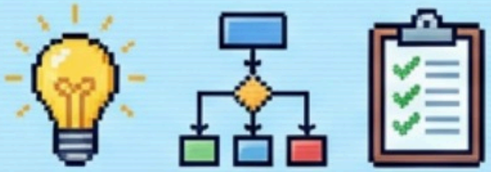
不知道自己要什么
→ 让 AI 替你想（深度思考 / 头脑风暴 / 深度调研）

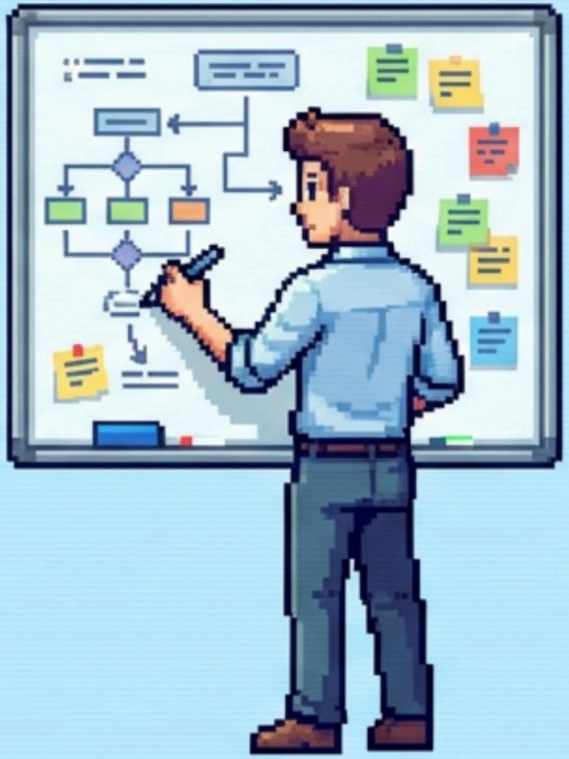




DILIGENT

大致知道自己要什么
→ 用 4D 框架自己理清





JOHARI WINDOW · 你 × AI

公共区
（都知道，直接问）

DIRECT QUERY

我必须告诉它
（context 缺）

MISSING CONTEXT

AI 能教我
（主动问 AI）

TEACH ME

双盲区
（需要研究）

NEEDS RESEARCH

先弄清边界，再决定 prompt 怎么写

为什么 SAAS 越来越多 · 3 GAPS



拿不到

accessibility（公众号/视频号）



拿到不会管

management（活动物料）

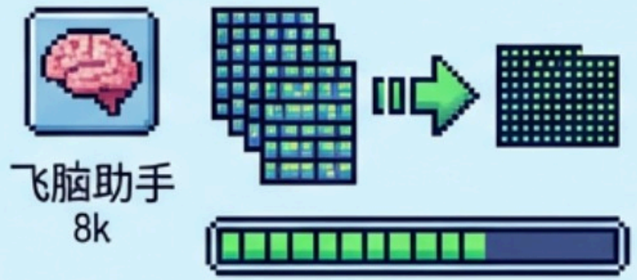


能用但用不好

窗口物理上限

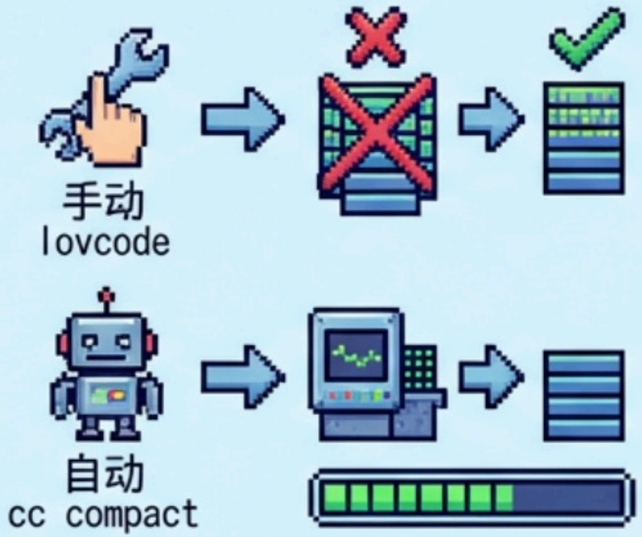
窗口压缩 · 3 METHODS

压缩输入 (INPUT COMPRESSION)



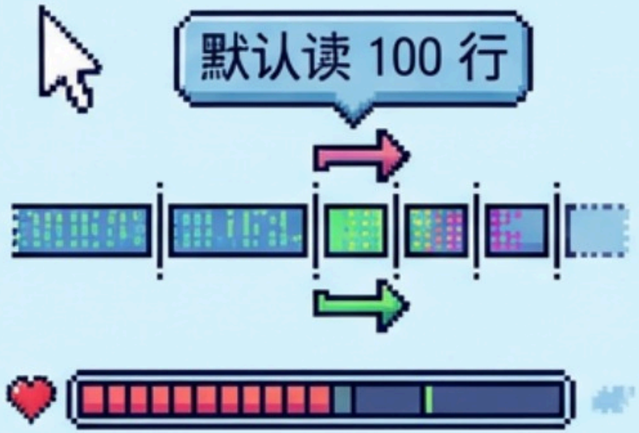
飞脑助手 8k

压缩上下文 (CONTEXT COMPRESSION)



手动 lovcode / 自动 cc compact

分片 (SHARDING)



Cursor 默认读 100 行

为什么需要 GUI · NOT JUST CHAT

功能性目标：harness environment



多步审批



跨人协作

非功能性目标：审美 + 视觉认知



PPT 全局结构

通用 chat + 场景化 GUI = 2026 AI 产品正常形态

CH 2 · TAKEAWAY

第二章 · 上手教学 · 小结

装环境（5 层基础设施）

选工具（CLI vs APP / Claude vs OpenAI / 周边）

管 context（4 类来源 + 知识库 + memory + session + prompt）

1. 90% 的"AI 不行"是 context 不行 — 不是模型

2. Context = Harness + Prompt

3. 通用 chat + 场景化 GUI = 2026 AI 产品正常形态

下一章 → 人机交互方法论 · 4D 心法

Anthropic AI Fluency · 中文实践版

CH 3 · 人机交互方法论

HOW TO WORK WITH AI

4D 心法 · 把 AI 当队友 · 让自己每用一次都变强



第三章

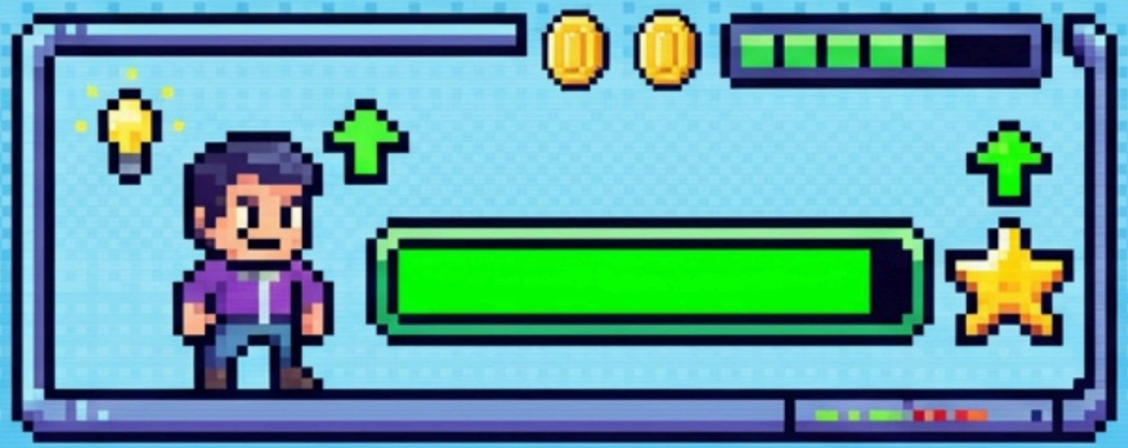
用得不流畅的人 — 每用一次 AI，自己变弱一点

用得流畅的人 — 每用一次 AI，自己变强一点

来源: Anthropic AI Fluency for Students



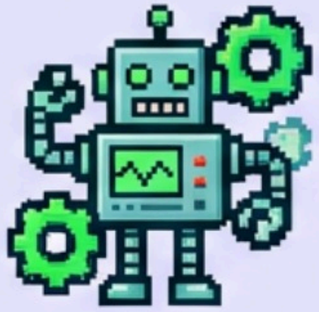
每次使用 AI ↓（变弱）



每次使用 AI ↑（变强）

D1 · DELEGATION 委托

动手前 30 秒，这件事到底交不交给 AI



完全自动化

100%



增强

50% 50%



不交给 AI

100%

我的 3 个判断问题

- [x] 要署你名字吗?
- [x] 下次还要做吗?
- [x] 错了代价多大?

反例　全交变蠢　全不交白用

D2 · DESCRIPTION 描述

BAD ✗

- SHOW Q1 SALES DATA
- LIST TOP 5 PRODUCTS
- COMPARE REGIONAL PERFORMANCE
- REPORT DAILY REVENUE
- FIND BEST CUSTOMER



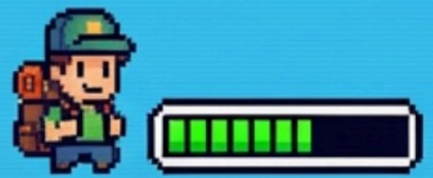
BOSS / COMMANDER

GOOD ✓

> CURRENT STATUS: Q1 SALES DATA AVAILABLE
> GOAL: IDENTIFY GROWTH TRENDS
> EXAMPLE: 'SHOW TOP 5 PRODUCTS WITH +10% GROWTH'
> CONSTRAINT: EXCLUDE DISCOUNT ITENS
> METHOD: DATA ANALYSIS + VISUALIZATION



INTERN / ASSISTANT



把 AI 当 实习生，不是当下属



D3 · DISCERNMENT 辨别

双向：防错 + 识惊喜

防错三件套（ERROR PREVENTION SUITE）

- 置信度（CONFIDENCE）
- 推理链（REASONING CHAIN）
- 交叉验证（CROSS-CHECK）

识惊喜（IDENTIFY SURPRISES）：每次方案读三遍（找错/找漏/找你想不到的角度）

真实 CASE：14 款 AI 整理名单

GPT 摆烂 🤔

智谱编人 ⚠️

文心稳但仍 5% 要核 ⚠️



D4 · DILIGENCE 尽责

红框警告：署你名字的，就是你的 

中国场景特殊形态：代价递延（合同条款 3 个月后才暴雷）

我的口径：每行过一遍，每处 diff 点头

YOUR NAME · YOUR RESPONSIBILITY

CH 3 · TAKEAWAY

第三章 · 人机交互方法论 · 小结

D1	委托	30 秒判断
D2	描述	五元组沟通
D3	辨别	防错 + 识惊喜
D4	尽责	署你名字 = 你担责

🌱 用得不流畅 — 每用 AI 自己变弱一点

🚀 用得流畅 — 每用 AI 自己变强一点

🎯 4D 不是清单 — 是本能

附录 → 开发者配置 · iTerm + tmux + cc

DEPRECATED?

附录 · APPENDIX

开发者配置 · iTerm + tmux + Claude Code

这是我现在还在用的开发者配置，但已经快过时了



ADVANCED USERS ONLY

正在被新 runtime 取代

痛点不在工具 · 痛点在切换

BEFORE



8 个 app 切来切去



焦头烂额

AFTER



80% 时间在 iTerm 一个窗口



气定神闲

三件套 · WHY THIS STACK



不接受"工作中断 = 工作丢失"

TMUX 布局 · MY DEFAULT

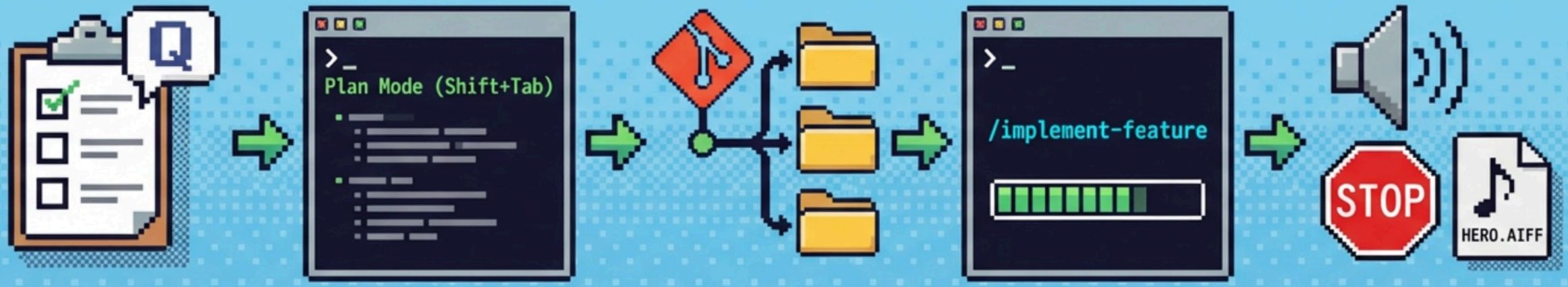


TMUX.CONF · 80/20 配置

```
set -g mouse on
set -g history-limit 100000
set -g status-right '#(ccm-status)|%H:%M'
```

上百行只有这3条带来80%价值

SUBSCENE 1 · 写代码



1. 问卷澄清需求

2. PLAN MODE (SHIFT+TAB) 出阶段计划

3. WORKTREE 并行做

4. /IMPLEMENT-FEATURE 执行

5. STOP HOOK 响 HERO.AIFF 才回来 REVIEW

30-60 MIN/中等功能（VS 纯人写 4 小时起）

SUBSCENE 2 · 长任务



关键设计



独立 tmux session

Lovnotifier 三路通知（手机+桌面+微信）

必须断点续传（每文件 commit 一次）

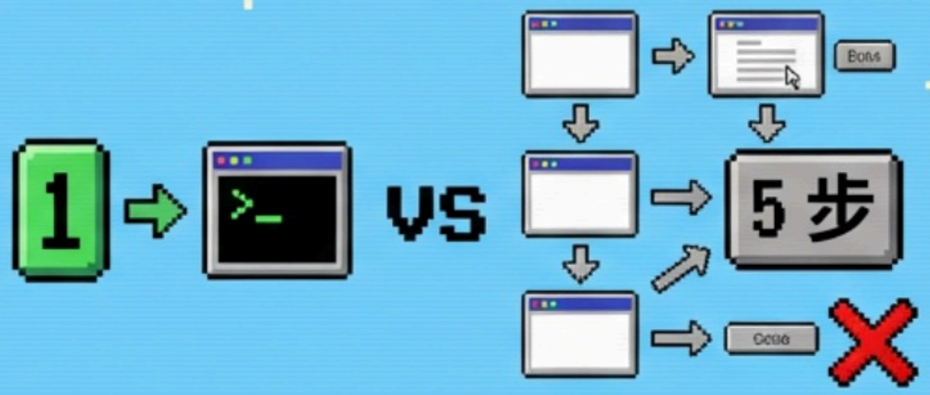
必须有终止条件

真实 case：5 万行 JS → TS，过夜 89% 完成

SUBSCENE 3 · 临时杂活

```
$ echo "把这段日文翻成中文" | claude --print
→ 翻译中...

$ claude --print "36*24*30 = ?"
→ 25920
$
```

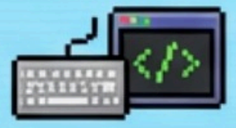
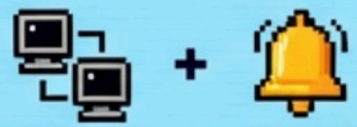
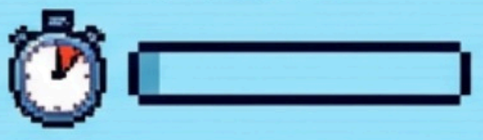


1 步　　网页版 5 步

时间：< 10 秒 / 次

日 20-30 次

3 SUBSCENES · COMPARISON

SCENE	KEY TOOLS	TIME	FREQUENCY
CODING 	Plan + Worktree + Hook 	30-60 MIN 	DAILY 
LONG TASKS 	CC + Lovnotifier 	HOURS 	WEEK 1-2 
TEMP JOBS 	claude --print 	< 10S 	DAY 20+ 

不是工具简化了
是认知带宽被释放了



- 不再花精力记"哪件事在哪个 app"
- 只记"它是哪一类工作"

但愿每个 OPC，都不是 NPC。

——手工川



扫码联系

「公众号」

「微信」

LovStudio · 文字 CC BY-NC 4.0 · 代码 MIT