



从教材到讲义

网页、小龙虾与 Code，各自管哪一层

一场面向老师的用法拆解：先看清门槛，再决定要不要打开终端

2026.03.18 分享：JKL-林老师

小龙虾把门槛拉低；真正决定质量的，是你有没有守住 Markdown 和人工检查。

这不是一份装机说明书

现场是一场聊天局。JKL-林老师从 2022 年用 GPT 讲起，一路走到 Gemini、Claude，再到今年因为小龙虾和更便宜的 token，把大量备课工作放进 Kimi。讲义把它收成一条可以复查的路径：先分层，再看案例，最后落到“教材 → 讲义”的固定流程。

整理原则

- 保留分层判断、工作步骤和可执行检查点。
- 删去寒暄、卡顿、现场找文件和与主题无关的闲聊。
- 保留分享者署名：JKL-林老师。听众与私下互动不点名。

阅读提醒

- 产品名、套餐和能力会随版本变化，文中价格只作当时经验。
- 本文是社群用法汇编，不是厂商说明或安全承诺。
- 加密文件、网盘和账号权限，只处理自己合法持有的资料。

“很多人觉得小龙虾很强，用过 Code 的人却 get 不到。因为它解决的不是更强，而是更低门槛。”

读完可以带走三件事

这份讲义给已经会用网页 AI、但还没想清楚要不要上 Agent 和 Code 的老师。它不替你选冠军，只帮你判断：这个任务该停在哪一层。

01

先分层

网页聊天、云端小龙虾、本地 Code，做的不是同一类工作。

02

再看案例

排课提醒、网盘整理、公众号早报、教材讲义，各自适合不同入口。

03

最后守质量

先出 Markdown，人看过再转 PDF。一次生成完，通常不好看，也不安全。

本节结论：工具热度会过，分层判断可以留下。

01

THE MAIN CLAIM

先别问谁最强

先问：这个任务要不要长期记住？要不要碰文件？我能不能接受它直接改我的电脑？

01

网页负责对话

生成初稿、分析材料、写提示词。聊完就走，记忆不稳定。

02

小龙虾负责贴身

定时提醒、抓取网页、把结果发到飞书或手机。门槛低，也更危险。

03

Code 负责项目

读一整本教材、解锁自己的PDF、连续出讲义。质量上限更高，也要会监工。

带走一句话：选层，比选品牌更重要。

同一件事，可以停在不同层

JKL-林老师自己的路径很清楚：网页版先养出“备课分身”，小龙虾做轻量助手，真正出讲义时再打开 VS Code 里的 Code。

01 网页对话

适合试提示词、看模型口味、生成一版能改的初稿。窗口一换，它常常忘记你刚说过的话。

02 云端小龙虾 / OpenClaw

适合定时任务、抓取、提醒、把文件发到手上。它把 Code 的能力做成更轻的接口，但本地安装风险高。

03 VS Code + Kimi Code / Claude Code

适合读本地大文件、连续生产讲义、把流程写成 Skill。199 套餐当时可同时跑两个 Kimi 窗口；Claude 更贵、更强，也更容易触顶。

Claude 像十分，Kimi 像五分；但对很多老师来说，五分已经能把国内网页版甩开。

WHY NOW / 为什么是现在

贵的不是能力，是用得起

现场反复出现的对比不是“谁更聪明”，而是谁能让老师持续用下去。国外模型质量高，但账号、封号和额度会把高频备课逼停。

高质量、高成本

Claude / Claude Code

适合难任务和关键稿件。网页里聊几句就可能触顶；JKL-林老师提到自己已有多个账号被封，因此不敢随使用。

够用、可包月

Kimi + Coding Plan

因为小龙虾出圈，也因为 token 相对便宜。包月/包周比按量烧 token 更容易估成本。她自己用得很勤，周用量仍可能很低。

2022

从 GPT 开始，学业和工作都离不开 AI。

2025

Gemini 2.5 带来一轮升级，图像和题目生成让她转向。

2026

为了小龙虾和更稳的价格，把大量流程迁到 Kimi。

本节结论：先选能持续付、持续用的层，再谈极致质量。

02

LIGHTWEIGHT AGENT

小龙虾真正擅长什么

它不是更强的聊天窗，而是一个能记住、能定时、能去网上抓东西的贴身助手。

01

教务提醒

开课时间、课后提醒、临时调课导致的课时错乱，尤其适合没有教务系统的独立老师。

02

预制反馈

把常用课后评语做成模板，改学生名、课次和时间就能发出。

03

抓取与整理

给一个网址或网盘，让它先出分析报告，再分批整理。不要一次吞下全部。

带走一句话：轻量任务交给龙虾，重文件交给 Code。

每天八点，先收集再写

JKL-林老师没有拿小龙虾去搜题，而是让它做 AI 教育资料的定时整理，并尝试写成公众号初稿。

场景	判断	做法	结果
信息太碎 每天都有新工具和新说法，人工追不过来。	先收集，后成文 不允许直接“感觉很好就写”，必须带上搜索链接。	定时发送 每天八点推送：核心要点 + 全部来源链接，方便核对幻觉。	能发，但未调教完 文章能写出来，文件却生在云电脑上，找起来很烦。

提醒

下一次先学会“做完把文件发到我手上”，不要让成品留在你进不去的云目录里。

本节结论：定时任务有价值，交付路径比文笔更先要打通。

先出报告，再允许它动手

老师们常见的不是“没有资料”，而是资料太多、重复收藏、自己懒得整理。现场演示的是连接网盘 API，让小龙虾先分析再整理。

场景	判断	做法	结果
资源湖 从早教到高年级的考试资料堆在网盘里，体积大到要分批处理。	云部署，不装本地 本地小龙虾能看见浏览器里的账号和卡片信息，风险高于整理收益。	分析报告 → 去重 先让它说明有什么、重复在哪，再开始移动和归类。	能用，但没做完 至少知道这些文件夹是干什么的；一次做不完就停，不要硬吞。

把它想成三岁小孩：你让它做什么，它就做什么。不是它坏，是它没有边界。

想试用，先走官方云部署

现场对“自己电脑上装一只龙虾”的态度非常明确：能云则云。本地只建议放在闲置、少有隐私资料的机器上。

01 Kimi Club / 官方云

199 套餐当时附带一键云部署，适合先试投入产出，不必自己配环境。

02 网页版其实够用时

如果只是对话和偶尔生成文件，不必为了热度强上小龙虾。

03 不要装在主力电脑

它可能被诱导读取页面上的账号、支付信息，再被第三方骗走。

04 旧电脑可以

如果一定要本地，用资料很少的闲置设备，降低泄露面。

本节结论：先降低安装风险，再谈它能帮你省多少时间。

03

WEB AGENT

网页端能做，别一次信完

Kimi 网页端已经能自己建专家、列待办、出进度条，甚至生成上百页教辅。看起来很完整，里面通常还要改。

01

它会自己拆工

听力、阅读、写作、口语、PDF 整合，可以分成多个子代理，并做一轮自检。

02

PPT 只是将就能用

当时观感弱于 NotebookLM；NotebookLM 更漂亮，且已支持单页修改。

03

人工不能缺席

缺页、卡住、格式错，都要人指回去。AI 很强，不等于说完就完。

带走一句话：进度条完成，不等于讲义完成。

04

FROM BOOK TO HANDOUT

打开 Code，是为了碰全书

网页上传常有体积上限。Code 读的是本地路径，所以能处理更大的教材，也能把讲义做成一个持续迭代的项目。

01

复制地址，不要拖文件

终端里通常不能直接扔文件，把路径贴给它即可。

02

先框架，后成稿

让它先出目录和设计，你认可了再生成 Markdown。

03

一边问，一边做

一次生成完往往不是你要的。它每问一次需求，产出通常更好。

带走一句话：你是发号施令的人，不是等一份完美成品的人。

教材到讲义，只记这一条链

现场演示的是历史教材章节讲义：先读解锁后的书，再按章节出中英双语学习指南。演示稿来不及细修，但路径已经完整。

01

阅读

把合法持有的教材路径交给它，说明做哪一章、给谁用、中英如何对照。

02

Markdown

人和 AI 都能改的中间稿。目录、目标、练习先写在这里。

03

人工检查

看中英比例、是否跑题、有没有乱码。问题在 MD 里改，不要等 PDF 再大改。

04

排版导出

再让它用 LaTeX 或其他排版工具转 PDF。缺字体就让它自己找适配字体。

Markdown 是内容，PDF 是衣服。衣服可以换，内容不能丢。

终端不可怕，怕的是同时塞太多

VS Code 对不会编程的老师也能用，因为它主要是在管理文件。JKL-林老师习惯开多个终端监工：一个窗口干完，再接下一个任务。

自己的锁文件

遇到“不能编辑、不能删广告页”的自有 PDF，可以让它写脚本产出可编辑副本。这只适用于你有权处理的文件。

多窗口监工

当时 199 套餐大约可同时跑两个 Kimi。四个窗口也可以，只是不要让同一个套餐同时超员。

乱码先查字体

中文或公式乱码，优先让它安装/匹配字体，而不是一上来重写全部内容。

做完要提醒

双屏也不一定看得到进度。让它在需要选择或完工时弹窗/发声，你才能去干别的事。

边界

公开版权教材不要要求它去找盗版。它有时会拒绝，有时会被绕过；讲义里不把绕过当作方法。

第一版能用，不等于能上课

同一条简单指令，现场走出了两种 PDF：一种没有封面、配色差；一种目标、关键词和练习更完整。差别不在模型突然变聪明，而在你有没有把它调回自己的教法。

常见翻车

- 没说是国际学校，它默认整页中文。
- 中英对照变成左右分栏，读起来别扭。
- 颜色很丑，但重点和自我检测已经有了。
- 它自检一轮，仍看不出你真正不满意的地方。

更好的改法

- 所有大改都在 Markdown 里完成。
- 先认可框架，再允许它继续写下一章。
- 把稳定流程写成 Skill，下次只说“按生成讲义那条”。
- 脚本不要在当前文件夹无限繁殖，让它放到专用目录。

本节结论：无人参与的成品，质量通常会很差。

05

HUMAN IN THE LOOP

你要当老板，也要当编辑

现场把 Code 说成：我不管过程怎样，你自己想办法交差。这句话只对了一半。过程可以放手，验收不能放手。

01	02	03	04
先给身份	再给范围	盯住中间稿	把成功写成 Skill
你的教法、学生、常用课型，越早写进记忆或文档越好。	做哪一章、学生版还是教师版、要不要音频和对照。	Markdown 比最终 PDF 更值得反复打开。	固定提示词不要每次重打。同事没辞职，只是变成了 Skill——这是玩笑，也是警告。

带走一句话：结果导向之前，先有检查点。

下次备课可以照着做

不必一次学会终端。从你这周真正要上的一章开始。

01

判断这一章停在哪一层

只需要初稿，就停在网页；需要定时和投送，用云端小龙虾；需要碰整本教材，再开 Code。

02

先出 Markdown

目录、学习目标、例题和练习先写成 MD，自己读一遍再转 PDF。

03

一次只交代一件事

读一章、改一处版式、装一套字体，做完再继续。

04

不在主力电脑装龙虾

试用走官方云部署。本地安装只考虑闲置设备。

最适合你的层，是这周能做完、能检查、能再上一次课的那一层。

门槛可以降低，判断不能外包

小龙虾让更多老师摸到 Code 的边。Kimi 让高频备课不必先过国外账号这一关。这些都值得用。真正不能交给工具的，是你对一节课的结构、对一份讲义能不能进课堂的判断。

分层

网页、小龙虾、Code，按任务停留，不按热度升级。

中间稿

Markdown 是人和 AI 的共同语言。

安全

云部署优先，主力电脑不养没有边界的助手。

验收

第一版用来看路，上课版必须经过人。

从一本书到一份讲义，并不神秘。神秘的是你愿不愿意改第二遍。

视频号 [JKL神思记](#) · 小红书 [JKL神思记](#)

来源与说明

主要来源：2026-03-18 AI 神思会线上分享录音文字稿《JKLL二次AI分享_带字幕_原文》。分享者：JKL-林老师。产品名、套餐和技术术语已按上下文做有限规范化；语音识别中的 OpenClaw / Kimi Code / NotebookLM / LaTeX 等已校正。

公开阅读版保留分享者署名，不点名现场听众。涉及版权材料、账号安全和破坏性操作的部分，只保留原则，不提供绕过方法。