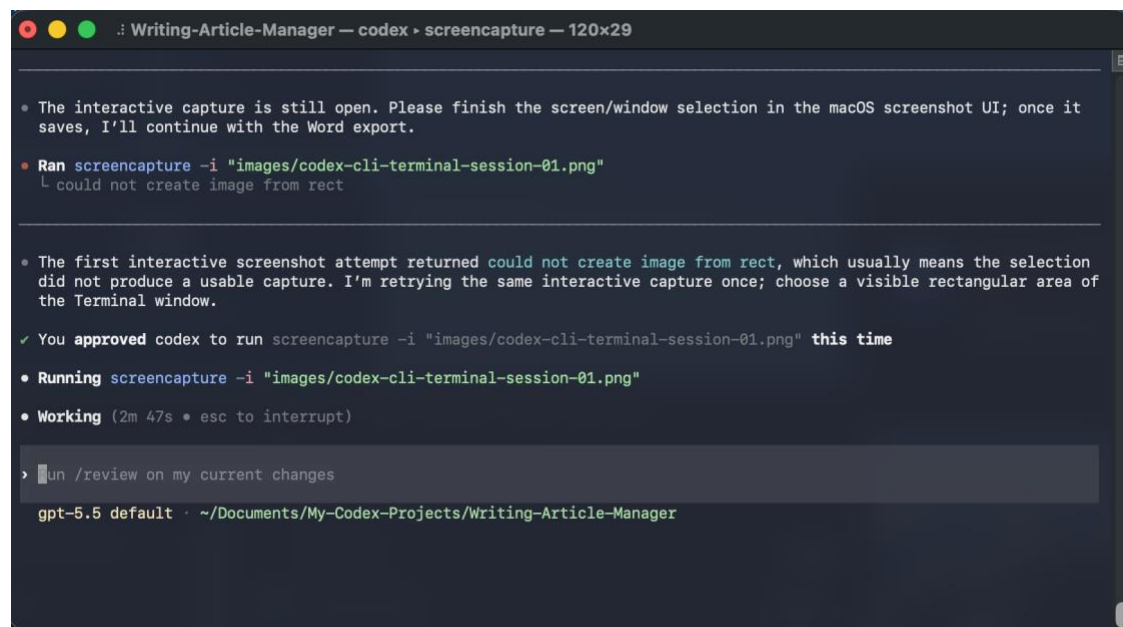


从 Codex App 到 Codex CLI: 建立自己的 AI 工作台

文 | 徐令予



Codex CLI 在固定项目文件夹中的实际工作界面

一、从桌面版开始

我最开始学习 Codex，是从 Codex 桌面版 App 开始的。这个入口对初学者比较友好：有可视化界面，操作过程直观，不需要一开始就理解太多终端命令。对当时的我来说，最重要的不是提高效率，而是先建立信心，知道自己输入一个任务后，Codex 会怎样读文件、改文件、解释步骤。

桌面版 App 也适合做简单练习。比如让它整理一段文字、生成一个小脚本、修改一个 Markdown 文件，整个过程都比较容易观察。还有一点对我很重要：它和 iPhone Codex app 的直接链接体验不错。我平时主要用中文手写输入，而这件事在手机上更自然，也更实际。所以能在 iPhone 上继续输入想法、补充指令，对我的学习流程很有帮助。

二、每日工作文件夹带来的混乱

后来我遇到一个具体问题：Codex 桌面版 App 经常会在 Documents/Codex 下面创建新的每日工作空间文件夹。这个设计本身可以理解，可能是为了区分不同日期和不同会话。但当我想连续做同一个项目时，问题就出现了。

比如昨天我已经在项目里写了规则文件，今天继续做时，系统又进入了一个新的日期文件夹。我就需要反复确认：现在到底在哪个目录？昨天的文件在哪里？这次修改是不是还在同一个项目里？文件一旦分散，学习就会变得不稳定。对我来说，真正的麻烦不是找不到文件，而是项目连续性被打断了。

三、为什么转向 Codex CLI

于是我开始使用 Codex CLI。CLI 的方式更直接：我先在终端里进入准确的工作文件夹，再从这个文件夹启动 Codex。比如先进入 My-Codex-Projects/Codex-Learning，用 pwd 确认当前位置，再开始工作。这样，所有文件都围绕固定目录展开，不会因为每天新建工作空间而分散。

我现在觉得，Codex 桌面版 App 和 Codex CLI 有点像汽车里的自动挡和手动挡。桌面版 App 像自动挡：更容易开始，更有引导感，也更适合初学者。Codex CLI 像手动挡：需要更多的理解和练习，比如目录、命令、权限和脚本，但也给了使用者更多控制。它们不是简单的好坏关系，而是适合不同阶段和不同需求。

四、CLI 更适合有 shell 经验的人

如果一个人本来就有一些 Unix、Linux 或 shell 经验，Codex CLI 会更容易发挥作用。因为他已经理解文件夹路径，知道 cd 是进入目录，pwd 是确认当前位置，ls 是查看文件，也知道脚本、权限和备份为什么重要。

有了这些基础，就可以把 Codex 和 shell 命令、脚本结合起来。比如先进入固定目录，再运行备份脚本，再创建文章项目，最后检查版本历史。这些动作不一定复杂，但它们可以重复执行，也更容易检查。对我来说，这正是 CLI 的价值：它把 AI 放进一个更可控、更可复用的工作流程里。

五、iPhone 同步仍然是限制

不过，CLI 也有现实限制。至少以我目前的经验看，Codex CLI 和 iPhone Codex app 还不能很自然地同步同一个工作会话。桌面版 App 与 iPhone 的直接链接对我来说工作得比较好，而 CLI 更偏向本地终端环境。

这对我不是小问题。因为我主要使用中文手写输入，很多想法在手机上写出来更快，也更符合我的习惯。如果 CLI 不能自然接上 iPhone 的输入和会话，我的工作流就会被分成两段。所以现在我把 Codex CLI 当作主要工作环境，把 iPhone 当作辅助工具：正式处理文件、规则、脚本和版本时回到终端；临时记录想法时使用手机。

六、我开始搭建自己的工作台

转向 CLI 以后，我开始建立固定学习环境。我创建了 My-Codex-Projects/Codex-Learning，用 PROJECT-RULES.md 记录工作边界，用 PROJECT-INDEX.md 管理文件用途。我还准备了 start-codex-learning.sh，帮助自己从正确位置开始；准备了 backup-project-tar.sh，用于给项目打包备份。

后来，我又创建了 Writing-Article-Manager。这是一个面向写作的小项目，其中的 new-article.sh 可以根据标题创建新的文章文件夹，并放入 README.md、outline.md、notes.md、draft.md 和 version-history.md。这个项目不复杂，但它让我看到，AI 可以被放进一个稳定的写作流程，而不是每次都从零开始聊天。

七、个人体会

我并不是想让 AI 自动完成所有事情。相反，我是在学习怎样建立一个可控的 AI 写作和研究环境。对我来说，关键是固定文件夹、清楚规则、备份、版本历史和人工检查。AI 可以帮助我写、改、整理和生成脚本，但最后仍然需要人来判断方向、检查内容、确认文件是否放在正确位置。这才是我理解的个人 AI 工作台。

附注：本阶段安装和使用的软件与工具

- Homebrew：用于安装和管理命令行工具。
- Codex CLI：作为主要的本地终端工作环境。

- Git: 用于理解和管理版本历史。
- Python: 用于运行脚本和处理自动化任务。
- Pandoc: 用于后续把 Markdown 转换为文档格式。
- Terminal: 用于进入文件夹、运行命令和启动 CLI。
- Finder: 用于查看和确认本地文件结构。
- TextEdit: 用于简单打开和查看文本文件。
- Codex desktop app: 作为早期入门和可视化练习工具。
- iPhone Codex app / iPhone direct link: 用于手机端输入、查看和辅助记录, 尤其方便中文手写输入。

徐令予 作于美国南加州 (2026 年 6 月 9 日)