DeepSeek Harness保姆级入门教程：从零跑通你的第一个AI Agent

DeepSeek Harness 是 DeepSeek 刚开源的一个本机 Agent 框架，命令名是 `dsh`。这篇文章只解决一个问题：把它装到你的电脑上，并跑通第一个任务。

它 2026 年 8 月 13 日才公开，现在处于开发者预览阶段，界面和命令随时会变。所以我不打算只给你一份会过期的命令清单，而是把「它是什么、为什么这样设计、每一步在干嘛」一起讲清楚。这样命令变了，你也能自己看懂下一步。



文章分三层：先讲清楚它是什么，再带你一步步装上跑通，最后给几条安全底线和进阶路线。全程默认你是第一次接触本机 Agent 的普通用户，不是工程师。

一、先破一个误解：它不是又一个聊天工具

很多人看到 DeepSeek 三个字，第一反应是：又出一个网页聊天助手？

不是。DeepSeek Harness 和网页聊天是两回事。

网页聊天里，模型只能待在一个对话框里回答你。DeepSeek Harness 是把模型放进你的电脑，让它能读文件、改文件、跑命令、拆任务。它真正的用法，是拿一个项目文件夹当「工作区」，让 AI 在里面干活。

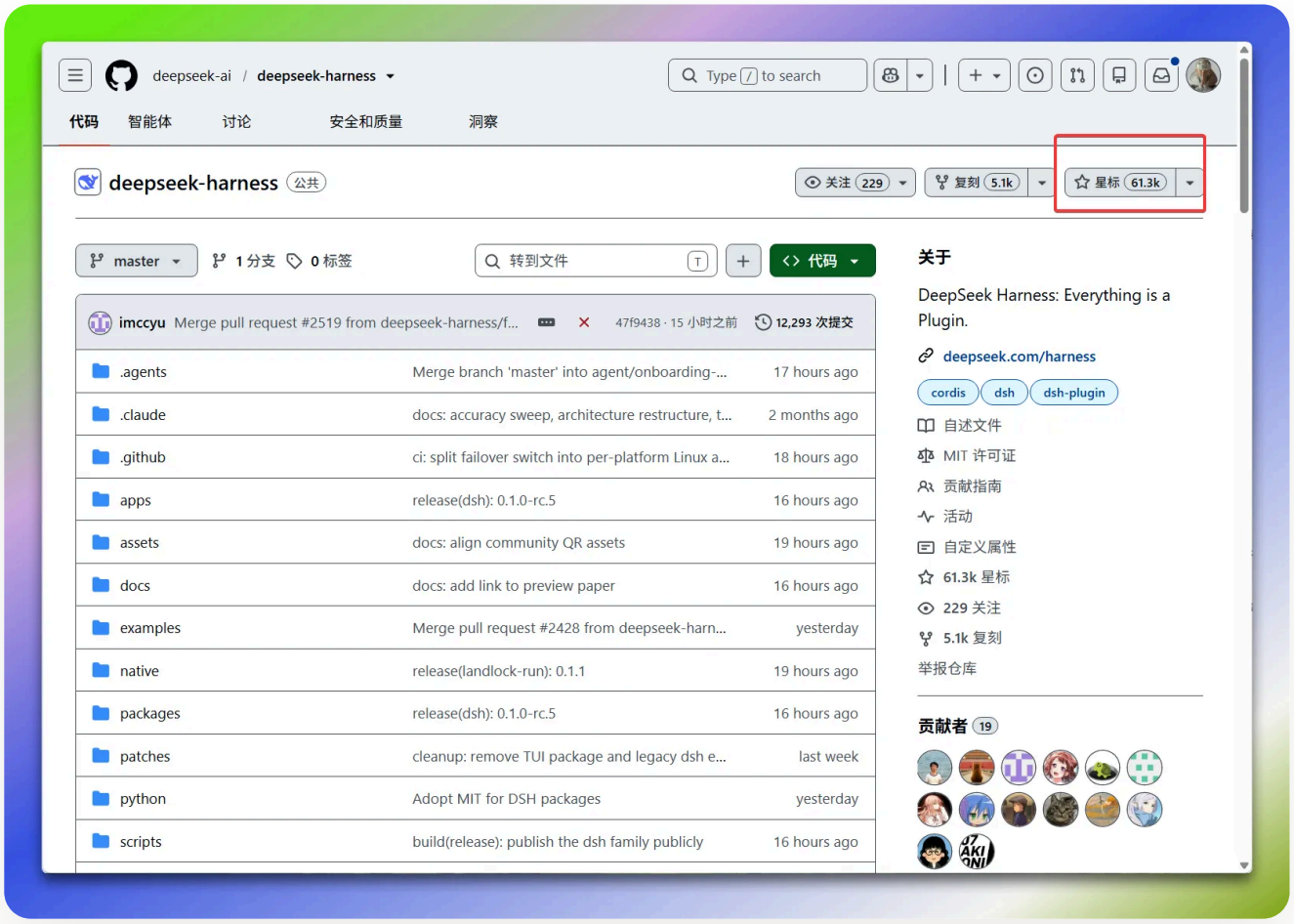
这个区别，官方用一个公式概括：

```
Agent = Model + Harness
```

Model 是大脑，负责想；Harness 是模型外面那套系统，负责让大脑能动手、能记住、能被约束。你熟悉的 Claude Code、Codex、WorkBuddy，本质上都是 Harness——它们要和具体的模型搭配起来，才是一个完整的 Agent。

为什么外面那套系统这么重要？因为模型再强，也有它天生做不到的事。

模型记不住事，Harness 给它外部记忆；模型不会自评，Harness 给它独立审查；模型不知道什么叫「做完了」，Harness 给它清单和边界；模型上下文会塞满，Harness 给它清理和压缩机制。



一句话：Harness 里每一个组件，都是在补偿「模型做不到的事」。你以后看到任何 Harness 组件，都可以问一句：它在补哪个短板？补不出短板的，多半是过度设计，别装。

所以它叫 Harness，不叫 Code、不叫 Build。DeepSeek 这次做的不是再做一个「软件」，而是把这套 Harness 开源出来，让你能拆开、替换、自己拼。

二、它最特别的一点：一切皆插件

在普通 Agent 工具里，文件读写、Shell、Skills、会话存储、界面……这些能力都是厂商封装好的。你只能「用」，不能「改」，甚至看不到它们长什么样。

DeepSeek Harness 不一样。它把这些能力全部做成插件，理念只有一句：

> 一切皆插件。

支撑这句话的是一个叫 Cordis 的内核。这个内核极其克制，只负责一件事：插件的加载、卸载和依赖管理。最特别的是，Agent 运行过程中也能热插拔插件，状态不崩。

这句话第一次听可能没感觉。翻译成大白话：Agent 可以一边干活，一边给自己加能力、换能力。它发现自己缺一个工具，可以现场造一个插件挂上去，接着干。

内核的克制，体现在两个词上。

第一个是「时间可组合性」：一个插件卸载之后，它之前产生的副作用能不能被干净地撤销。

第二个是「空间可组合性」：一个插件依赖其他插件，当依赖出现、消失或改变时，它能不能动态地重新处理自己的依赖。

你不需要记住这两个术语，只需要理解它想解决的问题：Agent 在运行中不断给自己插拔能力，而系统不会因此崩掉。这很像以前概念里的模块化手机——屏幕、电池、摄像头都能换。DeepSeek Harness 想做的，就是 Agent 界的模块化。

这也是它和市面上大多数 Agent 工具最本质的区别。别的工具是「成品」，它是一套「可改造的骨架」。

它和 Claude Code、Codex 到底差在哪

维度	Claude Code / Codex	DeepSeek Harness
成熟度	高、稳定	刚发布、开发者预览
理念	厂商封装好的产品	一切皆插件，可改造
生态	成熟的插件与技能生态	刚开始积累
适合	想要稳定生产力	想折腾、想定制

结论很直接：它不是用来立刻替换你手上成熟工具的，而是给你另一种可能——把 Agent 当系统来搭，而不是当软件来用。

三、一个前置概念：前置与后置

理解 DeepSeek Harness 之前，先补一个概念，后面很多设计都靠它。

一套可靠的 Harness，管 AI 有两类手段。

前置，是在 AI 动手之前告诉它该怎么做：规则、Skills、计划、权限边界。这些你以前在 Claude Code 里写的 CLAUDE.md、装的 Skill，都属于前置。

后置，是在 AI 做完之后检查它做得对不对：审批、验证、失败记录、会话日志。这一半是大多数人缺的。

只看前置，AI 可能嘴上答应、手上乱来；只看后置，AI 第一圈就跑飞了，你的检查没意义。两个都要。

DeepSeek Harness 里，Skills、计划、目标、子代理、工作流，偏前置；而它那份「只追加的会话事件日志」，偏后置——让你事后能看清它每一步做了什么、哪一步开始跑偏。

这个概念现在不用吃透。你只需要记住：先把一件事跑通，再慢慢加约束。别一上来就想把所有插件配齐，那是本末倒置。

落到 DeepSeek Harness 上，可以把它的能力对号入座：

能力	属于	作用
Skills、计划、目标、子代理、工作流	前置	动手前告诉它怎么干
审批、权限策略	后置	危险操作前拦一道
会话事件日志	后置	事后回看哪步跑偏

看清这张表，你就不会再把「装很多插件」当成目的。插件只是手段，补短板才是目的。

四、动手前，先备好三样东西

装之前先检查，能省掉 80% 的报错。

第一样，Node.js。这是硬门槛，而且版本很挑。官方要求 Node.js 22.19.0 及以上，或 24.0.0 及以上。

你的 Node 版本	能不能用
22.19.0 及以上	能用
24.0.0 及以上	能用
20、21、23	不能用
低于 22.19	不能用

去 nodejs.org 下载满足要求的版本，一路下一步装完，重开一个 PowerShell，输入：

```
node -v
```

看输出的版本号是否落在上表「能用」那一栏。这一步没通过，后面一定会卡，别跳过。

第二样，一个 DeepSeek API Key。它默认对接 DeepSeek 官方 API，需要你自己准备密钥，而且 API 是按量付费的。

去 platform.deepseek.com 注册登录，充值一点钱，创建 API Key。密钥通常是 `sk-` 开头的一长串。记住：密钥等于你的花钱权限，别外传、别贴进公开仓库、别截图发给别人。

第三样，一个干净的测试目录。第一次跑，别拿重要项目试。Agent 真的能改文件、跑命令，先用一个空文件夹当工作区，等摸熟了再上真项目。

五、三步跑起来

第一步，打开 PowerShell，粘贴这行命令：

```
npx @deepseek-ai/dsh web
```

`npx` 会临时把 DeepSeek Harness 拉下来运行，`web` 表示启动网页界面。第一次运行会问你要不要安装，输入 `y` 回车。

第二步，等它下载完，终端会打印一个本地地址，默认是：

```
http://127.0.0.1:3080
```

复制到浏览器打开，看到界面就说明启动成功。想换端口，可以用 `npx @deepseek-ai/dsh web --port 8080`。

注意：这个 PowerShell 窗口要一直开着，关了服务就停了。以后每次用，都要先在这个窗口启动。

第三步，完成首次配置。这个别跳，跳了会卡在输入框。

六、首次配置三步，新手最容易卡在第二步

第一步，进入 Settings（设置）→ Models（模型），把 API Key 填进 DeepSeek 那一栏，保存。

密钥保存后不会回显明文，页面只显示一个脱敏标识，真正的密钥存在本地的 `.credentials.yaml` 里。配好后不需要重启服务，下次请求就生效。

第二步，点 Choose workspace（选择工作区），把刚才那个测试目录加进来并选中。

这一步不选，输入框是灰的，没法发任务。这是新手最容易卡住的地方，先记住：没选工作区，不是坏了，是还没给 Agent 划定活动范围。它连「能碰哪些文件」都不知道，自然不会让你开工。

第三步，在模型选择器里选一个模型。选中的模型会成为新会话的默认模型。

七、四种模式，新手只用第一种

DeepSeek Harness 预设了四种模式。这里容易绕晕，因为它本质上不是四种「功能等级」，而是四种「插件组合模板」。

模式	是什么	给谁用
标准模式	功能完整的编码 Agent，文件编辑、Shell、检索、Skills、计划、目标、子代理、工作流都配好	第一次上手的人
PTC 模式	标准能力 + Code Mode SDK，让模型写 TypeScript 程序组合多步操作	想省 Token、会调试的人
极简模式	只给持久 bash 和文件编辑器两个工具	做模型基准测试的人
创造模式	标准能力 + 运行时检查、插件实验、preset 创作	想造自定义 Agent 的人

逐个说。

第一种，标准模式。这是开箱即用的完整编码 Agent，第一次用无脑选它。

第二种，PTC 模式。它在标准模式基础上，用 Code Mode SDK 呈现工具，让模型写一段 TypeScript 程序，一次组合多步操作。好处是减少来回对话、更省 Token；代价是要求模型有稳定的代码规划能力，调试门槛高。小白先不碰。

第三种，极简模式。只给模型两个工具：一个持久 bash、一个文件编辑器，系统提示词也压到最简。它的用途是做模型基准测试、比较两个模型的裸能力。日常用会很别扭。

第四种，创造模式。这是最特别的一个。它在标准模式基础上，能让 Agent 检查自己正在运行的插件环境、现场实验插件、创作新的 Agent 预设。说白了，它能让 Agent 改造自己。

你可以这样理解创造模式：Agent 发现自己没有扳手，就现场造一把扳手，插到自己手上，接着用这把扳手继续干活。这是 Cordis 内核最直接的体现，也是 DeepSeek Harness 真正的核心卖点。

但对第一次上手的人，一句话：标准模式先跑通，其他三个以后再说。

八、模型怎么选：Flash、Pro 和思考强度

第一次跑，模型不用纠结，选列表里默认的就行。但你可以记住一个粗略原则。

Flash 一档，速度快、成本低，适合快速验证流程；Pro 一档，能力更强、价格更高，适合复杂一点的活。具体模型名以你 provider 列表里显示的为准。

另外还有思考强度（reasoning）可以调。低强度更快更省，高强度想得更久。新手先默认，等摸清自己更在意速度还是质量，再动它。

更重要的是：DeepSeek Harness 没有把你锁死在 DeepSeek 模型上。它支持添加目录里的模型提供方，也支持自定义提供方——填 Base URL、协议、模型列表，理论上可以把别的 OpenAI 兼容模型接进来用。

也就是说，它是一个框架，不是一个卖模型锁。你想在它里面用别家模型，是有这条路走的。

九、跑第一个任务，顺便看懂它是怎么干活的

配置完成后，新建一个会话，发一句：

```
看看这个工作区里有什么文件，用一段话说清楚它的结构，再新建一个 hello.md 记录你的观察。
```

然后观察它。它会读目录、写文件，遇到需要审批的操作会弹出来问你，确认了才继续。

跑完去你的测试目录看，应该能看到 `hello.md`。看到它，就算真正跑通了。

想再进一步，可以给它一个更接近真实场景的任务：

```
把工作区里的 Markdown 文件全部读一遍，找出有拼写错误或链接失效的地方，列成一份清单。
```

这一步你会看到它反复读文件、整理结果，中间可能还会停下来问你确认。这个过程，就是 Agent 和普通聊天最大的区别：它不是一次性吐答案，而是一步步做事。

这里值得留意一个设计。DeepSeek Harness 把每一次会话都记成一份只追加的事件日志：系统提示词、用户消息、推理过程、工具调用、权限变化、子代理调度，全都作为事件写进日志。下一轮模型看到的历史，是从这份日志重新推出来的。

这件事对普通用户的价值是：任务跑偏了，你能回看它到底在哪一步开始跑偏的，而不是只看到「失败」两个字。可观测、可审计、可复现，这才是它能拿来认真干活的地基。

十、实战案例：别人已经拿它做出了什么

开发者预览刚发布一两天，社区里就冒出一批实测。它们有个共同点：长程、硬核、高缓存命中、成本压得低。这里挑几个有代表性的，链接都保留，你可以点进原帖看视频。

先说一句题外话：这些成果不代表你第一天就能复现。它们多是会折腾的人做的硬核玩法，普通用户看个方向就好。

1. 施瓦西黑洞建模

@LufzzLiz 用约 35 分钟一遍跑完最难的一类黑洞模型，缓存命中率 99%，成本约 2 元；同样任务在 Codex 上跑 DeepSeek V4 Pro 约 7 元。链接：

https://x.com/LufzzLiz/status/2087918358188695712



2. 四冲程柴油机 3D 交互仿真

@NFT_Chen 输入 17M tokens、输出 275k，耗时约 1 小时 8 分钟，做出全尺寸可拆解、带四冲程相位仿真、PBR 渲染、60fps、悬停显示零件信息的交互仿真。链接：

https://x.com/NFT_Chen/status/2087958659838149033



3. 3D 银河系模拟器

@dojopy 用 DeepSeek Harness 做了一个可缩放、自由旋转、点击行星探索的浏览器 3D 模拟器，配完整演示视频。链接：https://x.com/dojopy/status/2087978465068806367



4. 经典坦克大战（纯 C 语言）

@NFT_Chen 用 DeepSeek Harness + Flash，约 15 分钟跑出第一版可运行代码，总成本约 $0.12，能跑、能打、有音效。链接：https://x.com/NFT_Chen/status/2087925740658659755



5. 让 Harness 自己写桌面版

@btc_great101 用 dsh web + deepseek-v4-pro，约 11M tokens / 60 分钟，生成基于 Tauri 的原生桌面壳（窗口、托盘、自愈）。GitHub：https://github.com/xingj404-lab/dsh-desktop；社区

还有开箱即用打包版：https://github.com/anywhere-labs/deepseek-harness-desktop



这些案例的共同点

你数一数会发现，几乎都指向长程编码、3D 可视化、仿真，以及「缓存命中率高 → 成本低」。缓存命中高，意味着重复上下文不用反复计费，成本才压得下来。这正好对应 DeepSeek Harness 想解决的场景：让一个 Agent 长时间、连续地把一件复杂的事做完。

一句话提醒：这些是开发者预览发布头两天的社区实测，具体数字以原帖和当时价格为准。它们说明方向，不保证你复现的结果。

十一、社区已经出现第一批插件

官方给插件仓库定了一个 `dsh-plugin` 话题，用来聚合社区插件。发布一天，社区里已经有人做了第一批尝试。

比如在输入框 `@` 就能调文件的、让模型在回复里直接渲染图表表格和 Diff 的、补自动化能力的、把界面改成类似 VS Code 工作台的、给纯文本模型补视觉能力的。

这些是第三方插件，用之前自己评估。但它们的出现，正好说明「一切皆插件」不是口号——第一天就有人真的在往里插东西。想看最新进展，去 GitHub 搜 `dsh-plugin` 话题。

十二、跑通之后，往哪三个方向继续

跑通第一个任务后，别急着装插件。先顺着三个方向，慢慢把标准模式里的能力用起来。

第一，Skills。把重复的动作沉淀成技能，让 Agent 每次按你的方式干活。比如「写文章前先查一遍我的风格文件」「发布前先跑一遍检查清单」。

第二，MCP。把外部工具接进来，比如搜索、文档、表格。前提是你真的有一个具体的任务需要它，而不是「听说 MCP 很厉害」就装。

第三，子代理与工作流。把大任务拆成几个小任务，分派出去再汇总。等你的单个任务开始变复杂，再考虑这层。

这三个方向都在标准模式的预设插件里，你不用额外折腾，只是按需逐步打开。原则还是那句话：加一样，问一句它补的是哪个短板，补不出来就不加。

十三、它现在对普通用户并不友好，这是事实

这一段我想说得诚实一点，因为市面上的吹捧很容易误导你。

DeepSeek Harness 现在还不是一个「体验丝滑」的产品。它术语多、门槛高、界面糙，四个模式就够普通人懵一阵。如果你只是想要一个「打开就能聊、帮我写文案」的工具，它大概率会让你失望。

它真正的价值在另一群人身上：想把 Agent 当成可定制系统来折腾的人。对这群人，它的「一切皆插件」和可观测会话日志，是有实打实吸引力的。

所以先给自己定个位。你是想「用 AI 干活」，还是想「搭一套自己能改的 AI 系统」？前者选成熟产品，后者可以来玩 DeepSeek Harness。两条路没有高下，只有适不适合。

十四、进阶两条路：源码运行和 Python SDK

普通用户用 `npx` 方案就够了。如果你习惯自己编译源码，前提是有 git 和 pnpm：

```
git clone https://github.com/deepseek-ai/deepseek-harness.git
cd deepseek-harness
pnpm install
pnpm run build
pnpm dsh web
```

另外官方还提供 Python SDK，但它的前置条件写着 Linux x64/arm64，或 macOS 14+（arm64），示例组合也明确不支持 Windows Agent。Windows 用户优先用网页版和命令行版，别在 Python SDK 上死磕。

命令行还有一个无界面模式，适合跑一次任务直接拿结果：

```
npx @deepseek-ai/dsh --profile headless "帮我检查这个仓库有没有明显的 bug"
```

十五、常见报错，先对号入座

现象	原因	怎么办
`MISSING_CREDENTIAL`	没填 API Key	到 Settings → Models 补密钥
`UNKNOWN_MODEL`	没选模型或模型不存在	在模型选择器里选一个
拉模型列表返回 401	密钥错	核对 API Key
`npx` 报 Node 版本问题	Node 不满足要求	升级到 22.19+ 或 24+
输入框是灰的	没选工作区	Choose workspace 加目录并选中
发图片被拒绝	模型被当成纯文本	自定义模型需显式声明支持 image

记住一个通用思路：报错先看是不是「没填密钥、没选模型、没选工作区、Node 版本不对」这四个基础问题，绝大多数新手卡点都在这四样里。

十六、几个新手常问的问题

我不会编程，能用吗？

能装、能用。但它本质是一个编程 Agent，界面和术语偏开发向。如果你的目标是写文案、聊天，会有落差；如果你愿意折腾，值得一试。

它是免费的吗？

框架本身开源免费，但跑任务要调用模型 API，API 按量付费。所以「软件免费」不等于「用起来不要钱」。

它能替代 Claude Code 或 Codex 吗？

定位类似，但它刚发布、还在开发者预览，稳定性和生态远不如成熟产品。现在更适合尝鲜和折腾，不建议立刻替换你的主力工具。

能接国内其他模型吗？

能。支持自定义 provider，填 OpenAI 兼容的 Base URL 和模型列表即可，理论上可以接 GLM 等模型。具体以你用的接口是否兼容为准。

Windows 能用吗？

网页版和命令行版可以，Python SDK 不支持 Windows Agent。Windows 用户走网页版最省事。

怎么更新到最新版？

用 `npx` 跑的话，每次启动都会拉最新版；从源码跑的话，`git pull` 之后再重新构建。因为它还在开发者预览，变化快，隔几天重跑一次就行。

它会不会把我的文件搞坏？

会，这是它干活的方式决定的。所以第一次一定用空目录或项目副本，重要文件永远先备份。它部分操作会先弹审批，但别把审批当成万无一失。

十七、三条安全底线，跑之前先记住

第一，它会真改文件、真跑命令。第一次一定用空目录，别对重要项目直接开火。最稳的做法是复制一份项目副本，让 Agent 在副本上折腾。

第二，API 是付费的，而且 DeepSeek 官方最近调过价。复杂任务会反复调用模型，费用比聊几句高。跑之前看一眼平台价目表和用量，别让一个长任务悄悄吃掉预算。

第三，密钥别泄露。别贴进公开仓库、截图、日志里。丢了密钥，等于把花钱权限交了出去。

十八、接下来做三件事

别急着把文章再看一遍。先把这三件事做完：

1. 装好 Node.js，跑通 `npx @deepseek-ai/dsh web`。
2. 填好 API Key，选好工作区，用标准模式跑通第一个 `hello.md`。

3. 跑完看一眼会话轨迹，理解「它每一步在干什么」。

做完这三件事，你对本机 Agent 的理解，会超过只看十篇科普文的人。

十九、全文浓缩版（5 分钟速查）

不想翻全文，照着这七步走：

1. 装 Node.js 22.19+ 或 24+。
2. 去 platform.deepseek.com 建 API Key，充点钱。
3. PowerShell 跑 `npx @deepseek-ai/dsh web`。
4. 浏览器打开 `http://127.0.0.1:3080`。
5. Settings → Models 填 Key；Choose workspace 选目录；选模型。
6. 标准模式，发任务，跑通 `hello.md`。
7. 先空目录试，注意 API 计费和密钥安全。

写在最后

DeepSeek Harness 现在还很糙，这不是缺点，而是它此刻的真相。它术语多、门槛高、上手体验一般，这些官方和社区都不避讳。

但它的价值不在今天的体验，而在它打开的那条路：Agent 不再是一个你只能「用」的黑盒，而是一套你能拆、能拼、能改的系统。

你不是它的用户，你是它的组装者。

对第一次接触的人来说，这句比任何功能清单都重要。剩下的，就是把第一个任务跑通。

官方入口： deepseek.com/harness；
仓库： github.com/deepseek-ai/deepseek-harness。

以上，既然看到这里了，如果对你有所帮助，还望不吝点赞与关注，这也是对我最大的鼓励与支持。
感谢你拨冗阅读，山高水长，我们期待下篇文章与你再见。

PS.特别提醒：

1，如果你是内容创作者，笔记杂乱、灵感难沉淀、素材难以复用，这套融合 PARA + 卡片法 + LLM 自动运维的 Obsidian 知识库模板直接抄作业：点此查看详情！

2，日常用 Codex 做自动化、写脚本、管理项目？这套完整落地实战手册，避开大量踩坑流程：点此查看详情！

3，需要一对一远程技术支持：
想用 AI 提高效率，但卡在安装、配置、报错和工作流搭建上，可提供Codex安装部署、疑难故障修

复，API配置、代充Plus、专属Skill定制，找我：dqtx33



大强同学

浙江 杭州

扫二维码，添加我为朋友。