企业内训 · 半天 3 小时

WorkBuddy 入门实战营

从「会聊天」到「会干活」

4 次现场演练 + 1 个演练营，离开时手上至少有一套能替你跑腿的助理配置。

3 种工作模式　5 类扩展对象　4 张任务卡



课程地图：这半天怎么过

课程全景

180

分钟 · 半天

- 讲解 95 分钟
- 动手 75 分钟
- 复盘与互评 10 分钟

每一章结尾都有可交付物：

离开时你手上至少有一条跑起来的自动化。

01	**认识 WorkBuddy：它凭什么干活** 拆掉「它只是个聊天框」的误解，认识界面与模式	P03–P09 35 分钟
02	**把任务说清楚：模式、权限与验收** 四要素、上下文、工作空间与结果验收	P10–P16 35 分钟
03	**五类扩展对象** 专家 / 技能 / 连接器 / 自动化 / 记忆	P17–P25 35 分钟
04	**实战演练营** 四张任务卡抽签分组，现场交东西	P26–P32 60 分钟
05	**避坑与进阶** 七个坑、四道安全检查、90 天路线图	P33–P37 15 分钟

第 1 章 · P03–P09 · 35 分钟

认识 WorkBuddy 它凭什么干活

先拆掉一个误解：它不是「更聪明的搜索框」，
而是能把一件活从头做到交付的执行体。

界面四区　三种工作模式　两种权限



它不是聊天机器人：四行对照看差别

传统 AI 对话		WorkBuddy
只能对话，提供建议		**能够实际执行任务**
需要你手动操作文件		**自动读写本地文件**
只能处理单步骤简单任务		**能拆解多步骤复杂任务**
输出一段文字回复		**交付可验收的产物文件**

4

行差别，一条分水岭

差别不在「更聪明」，在「**能动手**」。

它能读你的文件、跑命令、查资料、生成文档与图表，把一个目标跑到交付。

所以：把它当实习生派活，
而不是当搜索框提问。

八大核心能力地图

01 研究与写作

联网搜索 + 网页抓取，产出报告、PRD 与提纲

02 数据与分析

处理 Excel / CSV，生成图表、透视与看板

03 编程与开发

读懂整个代码库，跨文件重构、写测试、修 Bug

04 浏览器自动化

打开网页、填表、抓数据、跑重复流程

05 桌面应用控制

跨 Windows / macOS 原生应用执行操作

06 多模态生成

文生图、文生视频、文生 3D 模型

07 连接器

打通飞书、腾讯文档、GitHub、邮箱等外部服务

08 自动化调度

定时与周期任务，后台按点自动执行并交付

八项能力，按角色各取所需

不需要八项都会，先认领你岗位上的那两三项。

业务 / 运营 / 行政

研究写作 + 数据分析，已能覆盖八成的日常琐碎活。

产品 / 设计 / 市场

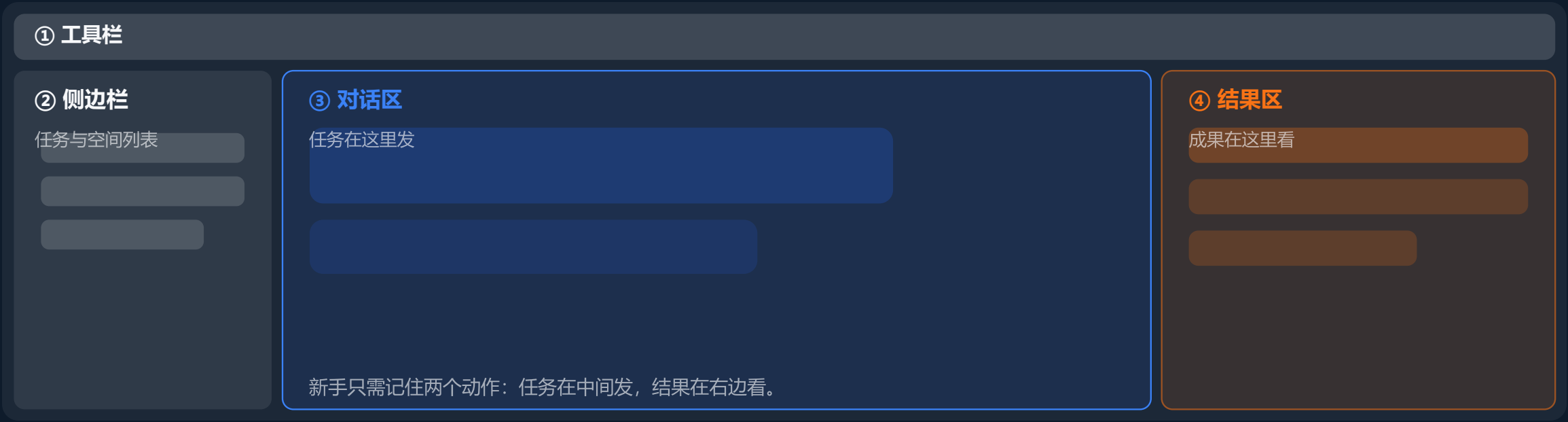
多模态生成出素材，浏览器自动化做竞品与舆情巡检。

技术 / IT

编程开发是主战场，桌面控制与自动化接管重复劳动。

真正的分水岭不是会几项，而是把「**能力 + 授权 + 定时**」串成一条流水线。

界面四区：先认清「谁在哪」



1 工具栏

顶部一排常用入口：切换模型、打开设置、查看快捷键与帮助反馈。第一次用，先在这里把默认模型定好，能省下后面不少等待。

2 侧边栏

所有任务按「空间」分组在这里排队。Ctrl / Cmd + K 可以全局搜索任务、空间和产物，忘了东西放哪就在这儿找。

3 对话区

你说需求、补上下文、随时打断和追加要求的地方。任务从这里发出去，执行过程也会实时显示在同一条消息流里。

4 结果区

右边看结果：概览里有工作空间文件、内置浏览器和本次变更；产物里是生成的文档、表格、PPT。先看变更再看产物。

演练 0 | 10 分钟跑通第一条指令

演练 0 · 全场动手

10

分钟 · 必须看到文件

本场目标

在你的磁盘上，
真的多出一个由它写出来的文件。

10 分钟没看到文件 = 没跑通。
别自己憋着，举手。

1 新建任务

模式保持默认「做一做」

点左侧「新建任务」，先确认输入框下方的模式是默认的「做一做」，别急着填内容——把工作空间选好更重要。

2 选择工作空间

别选桌面或项目仓库根目录

点输入框左下角的「选择工作空间」，挑一个空目录，例如 D:\wb-demo。这是它唯一被允许读写的范围。

3 贴入第一条指令

目标 / 输入 / 输出 / 约束都齐了

帮我在这目录新建 demo.md，写一份 300 字的本周工作计划提纲，分三条要点。

4 看结果区

看到文件才算跑通

右侧切到「产物」找到 demo.md；再切到「变更」，确认它确实写进了你选的那个目录。

三种工作模式：问一问 / 做一做 / 想一想

发出去之前先问自己：我需要它先给方案，还是先给答案？

按风险与确定性往下走

想一想 Plan

先出方案，你点头再动手

适合：需求模糊 / 多步骤 / 产出成本高

问一问 Ask

只回答、只分析，不改任何文件

适合：先摸思路 / 拿不准要不要动

做一做 Craft（默认）

边说边做，直接读写文件、跑命令

适合：目标明确 / 风险可控的日常活

3

种模式，是三档风险姿态

目标明确、单一任务 → **做一做**

目标模糊、怕改坏文件 → **想一想**

只想问清楚、不动文件 → **问一问**

选错的代价很具体：

要么重跑一遍，要么改错一批文件。

权限模式：默认权限 vs 完全访问



默认权限（推荐）

高风险动作会停下来，等你确认一次



完全访问权限

不再逐步确认，直接连续执行

它不是「更快」，是把闸门拆了。**用完即关，别长期开着。**

这道闸门到底管什么

遇到敏感路径写入、批量删除改名、执行脚本、访问网络，它会先停一下让你确认；完全访问会关掉这一步。

会被拦下来问你一次的操作

写入敏感路径、批量删除、批量改名、覆盖文件、执行脚本、访问网络。

它自带的保险

Windows 上修改已有文件前会自动备份；删除尽量走安全删除或回收站，不直接抹掉。

完全访问只在这类环境里短开

Docker 容器、虚拟机、一次性临时目录——炸了也不心疼的地方，用完即关。

第 2 章 · P10–P16 · 35 分钟

把任务说清楚
模式、权限与验收

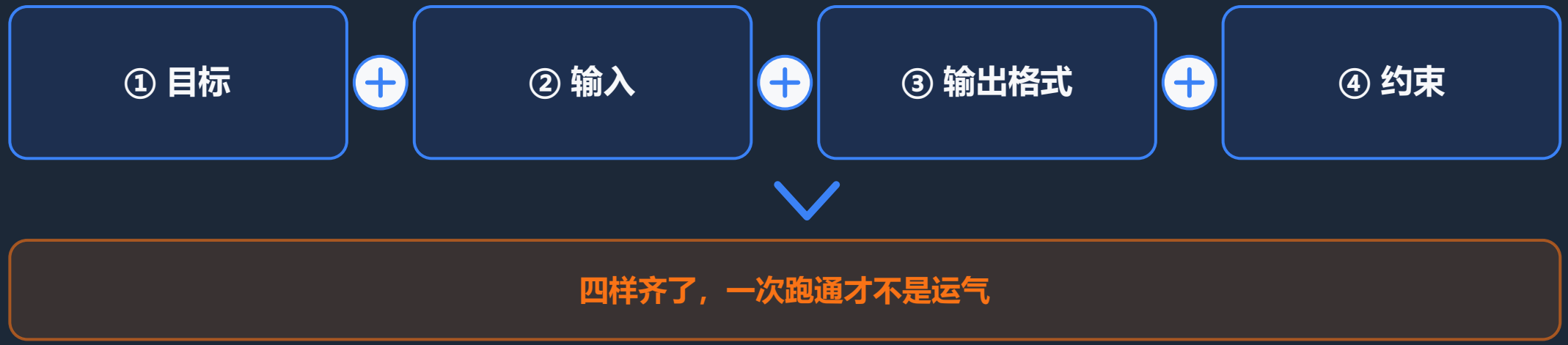


AI 干不好活，八成不是它笨，是需求里缺了东西。

这一章把那几样东西一次补齐。

四要素　上下文三件套　工作空间与验收

好任务的四要素



1 目标

要生成什么内容，或要处理哪批文件。讲清楚终点，不必讲过程。

反例：「帮我弄一下资料」

2 输入

要分析的数据、参考文档、会议录音或截图——能给原件就别给描述。

反例：「根据我的想法来」

3 输出格式

Word、Excel、PPT 还是网页；页数、字数、口径都写在这里。

反例：「随便给个结果」

4 约束

风格、字数上限、截止时间，以及哪些内容不许编、不许改。

反例：「你看着办」

上下文三件套：@ / 粘贴 / 上传

补进去的每一份材料，都在减少它猜的次数

三种最常见的补充方式，学会任意一种，都比「把需求描述得更详细」更有效。



@ 引用上下文

在输入框敲 @，直接引用文件、文档、规则或知识库条目。它会把这些内容当成「已知事实」来处理，而不是靠猜——引用得越准，返工越少。适合场景：你手上已经有明确的参考材料，不想再复述一遍，或者这份材料后面还要反复用到。

输入 @ → 选择文件 / 文档 / 规则



粘贴截图（Ctrl / Cmd + V）

界面截图、报错弹窗、手写白板照，直接截图后用 Ctrl / Cmd + V 粘贴进输入框。它能读懂图片里的内容，于是你省掉一大段「描述这个弹窗长什么样」的文字。适合场景：用文字说不清的问题、UI 走样、报错信息、纸质表格。

截图 → Ctrl / Cmd + V 粘贴



上传文件（拖拽也可以）

把 Excel、PDF、Word、图片直接拖进输入框，或者点上传按钮。文件会进入当前工作空间，之后所有追问都能继续用同一批材料，不用每次重传。适合场景：原始数据、长文档、需要反复引用的资料包；也适合把「一次性参考」和「长期素材」分开管理。

拖拽文件到输入框 / 点上传

演练 1 | 把真实任务改成四要素

演练 1 · 全员动手

15

分钟 · 用你手上真实的活

本场目标

挑一件你下周本来
就要完成的工作，
把它写成一句可执行的话。

别现编题目。
用真的活，才知道哪里说不清。

改前（90% 的人第一次都这么写）

「帮我整理一下资料。」

缺输入、缺输出格式、缺约束——它只能靠猜，猜错一次就要重来。

改后（四要素齐全）

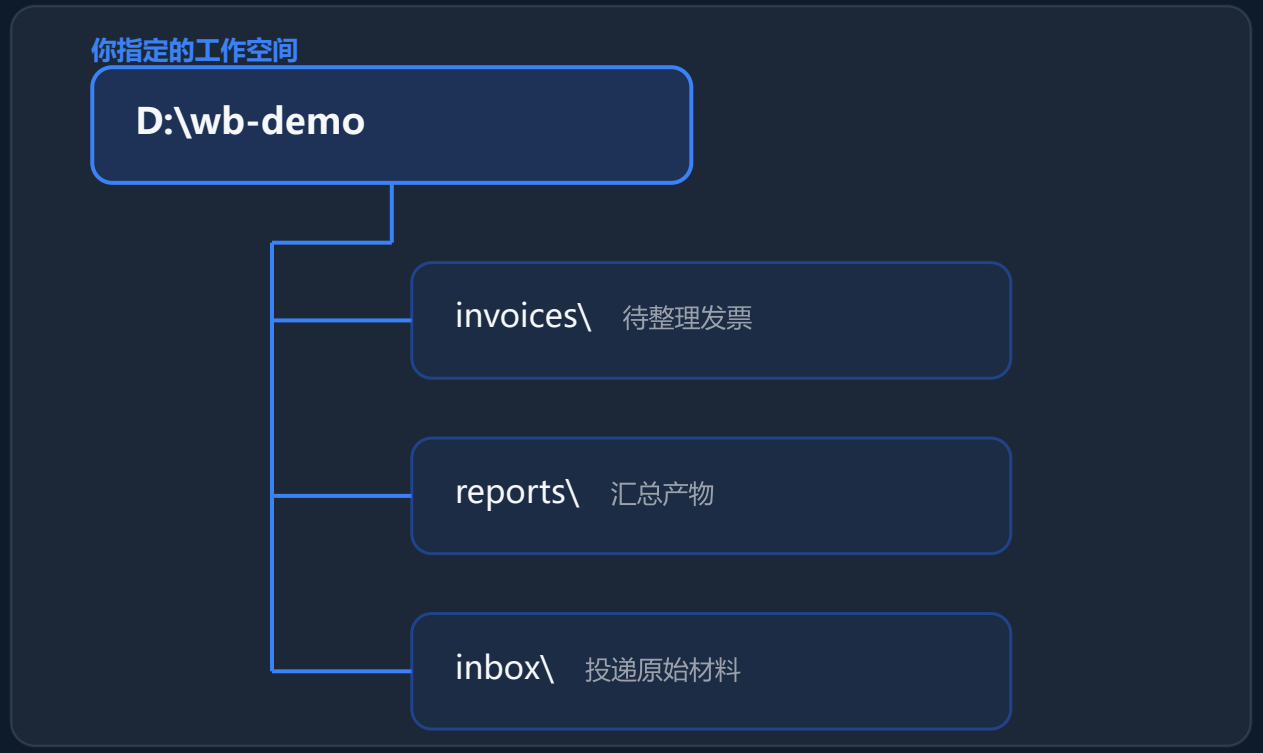
把 D:\wb-demo\inbox 里的 6 份门店月报 Excel 合并成一张汇总表，按门店和月份汇总销售额与退货率，输出一份 Excel（含明细页和汇总页，数字保留两位小数）。金额按含税口径，缺失数据统一标注「待确认」，不要自己编数。

输入：6 份 Excel　输出：Excel　约束：不许编数

互评规则：同桌交换，只挑毛病

- 四要素是否齐全：目标 / 输入 / 输出格式 / 约束，缺一项就标出来。
- 输出格式是否写到可验收的程度：文件类型、页数或字数、口径。
- 约束里有没有写「不许做什么」——这一项最容易漏，也最容易出事。

工作空间：给 AI 划出的操作区



系统会重点保护的区域

桌面 · 下载 · 个人文档根目录 · 用户主目录 · 项目仓库根目录。往这些地方写入、批量删除或改名，它会先停下来问你。

它只被允许在你划定的范围里动手

新建任务时点一下「选择工作空间」，它就拿到了明确的读写边界。边界之外的文件，它不会主动去碰。

1 **按任务类型分空间**

发票整理、周报、待处理照片，各自一个目录。范围越窄，误伤的可能越小。

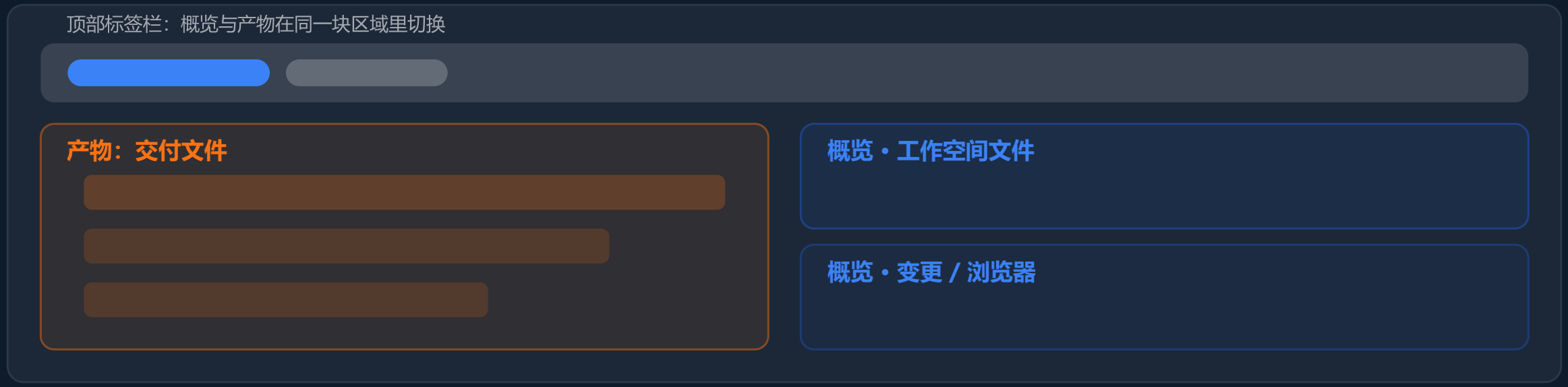
2 **别拿这些目录当工作空间**

桌面、下载、个人文档根目录、项目仓库根目录——系统会重点保护这些区域。

3 **不确定就新建一个空目录**

先小范围试一次，跑通了再逐步把更多材料放进去。

结果区怎么用：概览 + 产物



验收顺序建议：**先看「变更」确认没改错，再看「产物」确认交付对不对。**顺序反了，容易返工两遍。

产物

交付物在哪

生成的文件都在这儿：文档、表格、演示、报告。可分享，也能上传到腾讯文档 / ima / 乐享。

工作空间文件

写到哪了

当前工作空间的文件树。用来确认它到底把文件写进了哪个目录，有没有写错地方。

浏览器

效果对不对

网页、页面原型、本地启动的 Web 应用直接在这里预览，支持同时开多个页签对照。

变更

改没改错

本次任务改了哪些文件、每处改了什么。代码、脚本、配置类任务必须先看这一栏。

验收与返工：不满意怎么改

✗

重开一个任务，把需求再说一遍

上一轮试过的错、已经对齐过的口径、已经排掉的歧义，全部归零。大概率会在同一个地方再错一次——这才是最贵的浪费。

✓

在原任务里继续追问，只改那一处

它带着这一轮的上下文改：「第 3 页数字不对」就只改第 3 页。改到对为止，而不是从头再来一遍。

返工的三句标准话术

1. 「第 3 页的数字和原始纪要不一致，请按原文修正，不要自行推算。」
2. 「表格里留空的行请标注『待确认』，不要填 0。」
3. 「这次输出压到 8 页以内，每页标题就是那一页的结论。」

「不满意不是重新问一遍，
而是在原任务里，把话说得更准。」



这三类活，先「想一想」再执行

批量删除 · 批量改名 · 覆盖原文件。先让它出一份方案给你过目，确认范围无误再动手——这几秒钟能省掉后面几小时的欲哭无泪。

第 3 章 · P17–P25 · 35 分钟

五类扩展对象
缺哪一层补哪层

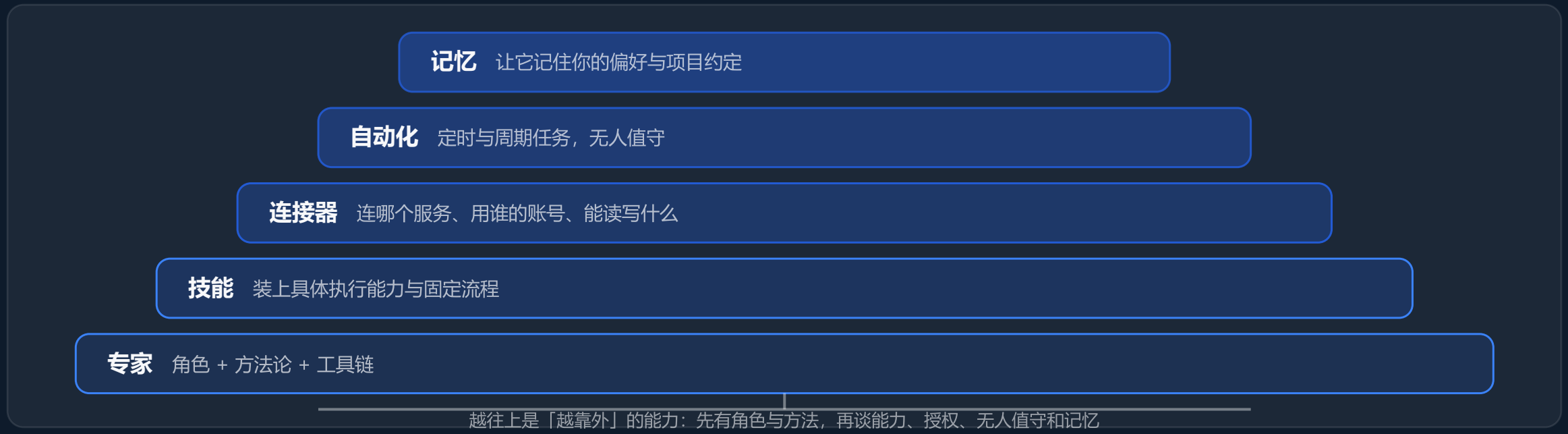
想让助理变强，靠的是这五样。

它们不在同一层——选错了，就是白折腾。

专家　技能　连接器　自动化　记忆



五类对象各站哪一层



它给的视角太外行，不懂行业行话	**换个专家**
同一套流程每次都要重新描述一遍	**沉淀成技能**
数据在别的系统里，它根本读不到	**配一个连接器**
每周固定要做的事，还靠人盯着	**建一条自动化**

记忆是个例外

它不补某一项能力，而是让它记住你的偏好与项目约定——跨会话保留，下次开新任务不用再交代一遍。

想让它记住，就明说「记住这条」。

专家：100+ 垂直角色与专家团

① 单个专家：一个视角，一条直线

6 份客户邮件 → **客户成功专家** → 风险周报

② 专家团：多角色并行，一个负责人兜底

一批材料 + 一个目标

↓

负责人：拆解任务、分派、汇总

↓

客户成功 | **法务** | **项目管理**

↓

一份汇总产出 + 待决策清单

专家给的是视角，不是权限

机制就三样：人设 + 方法论 + 工具链。材料不变，换个专家，关注点和风险点全不一样。

同一批客户邮件，三个视角

客户成功视角

先看续费风险与客户情绪，谁在变冷淡一眼看出。

法务视角

先看承诺、责任边界与合同条款，哪句话不能答应。

项目管理视角

先看延期、依赖和负责人，哪条链路要断。

选专家**不会自动打开**你的邮箱、网盘或本地文件——那要靠技能与连接器的授权。

另外：专家团的消耗，大约是单专家的 3–5 倍。

技能：把你的流程装进去

没有技能：每次从零描述一遍

第 1 次	第 2 次	第 3 次
从头描述 → 输出	再描述一遍 → 输出	还得描述 → 输出

装了技能：沉淀一次，之后直接调用



技能 = 一段可调用的指令集：把一套做法固定下来，同类任务不再靠即兴发挥。

官方推荐

1 内置精选技能，覆盖文档处理、数据报表、海报设计、文件整理等高频场景。

SkillHub 社区

2 社区作品聚合，也有打包好的套件。装之前看清来源与权限申请。

自己建

3 点「创建技能」，用大白话描述你的任务，它会帮你生成一套可复用的流程。

装第三方技能前，看三样东西

来源是谁 · 申请了什么权限 · 脚本会不会把内容发到外部。非官方技能可能存在提示词注入、越权访问等隐患——优先用官方推荐。

只启用当前任务需要的技能。**关掉不等于卸载**，技能文件还在，随时可以再打开。

演练 2 | 装一个技能并跑通

演练 2 · 各自动手

20

分钟 · 装一个真有用的

本场目标

你手上从「每次都要重新描述」，变成「一条可重复调用的流程」。

验收标准：

换一天、换一份材料，还能跑出同样的结果。

1. **找一个你真正要用的技能**
   在技能市场里搜你下周一就要用的那件事，别挑看不懂的。
2. **装之前，先看它申请了什么权限**
   来源是谁、要读什么、会不会把内容发到外部；非官方技能尤其要看清。
3. **在对话里用它跑一次真任务**
   直接说「用这个技能处理这份文件」，看它是不是真的按流程走。
4. **把这次用不上的技能先关掉**
   只留当前任务需要的。关掉不等于卸载，随时能再打开。

连接器：授权边界先看清

连接器打通的是「外部系统 + 授权」

用到哪个连哪个

协作办公	飞书 · 钉钉 · 企业微信 · 腾讯文档 · 金山文档 · Notion
开发协作	GitHub · GitLab · Gitee · CNB · 代码托管与流水线
通讯与会议	QQ 邮箱 · 网易邮箱 · 腾讯会议 · 腾讯问卷
数据与知识	ima 知识库 · 腾讯乐享 · 企业知识库 · 内部 wiki
商业情报	天眼查 · 企查查 · 客户与销售系统
存储与部署	腾讯微云 · 百度网盘 · 云托管与静态站点部署

连之前，先答四道题

任何一道答不上来，就先别连。

1. **连哪个服务**
   只连你任务真正要用到的那个，不必一次全连上。
2. **用谁的账号**
   以你本人的身份执行——所以你看到的范围，就是它能动的范围。
3. **能读什么、写什么**
   按服务逐个看授权页，别只看「已连接」三个字。
4. **怎么撤权**
   不用了在哪里禁用、断开、撤销授权，先找到这个入口再连。

发邮件、发群消息、提交 PR 这类**对外动作**，它会先要你确认一次才执行。

自动化：让任务自己跑起来

无人值守

24/7

小时 × 天，它不会下班

设一次，之后到点自动执行，结果直接交到你手上。

你只负责决策和审核。

一次性任务

指定未来某个时间点，执行一次就结束。

「明天下午三点提醒我开会」

周期性任务

按天 / 周 / 月的固定节奏重复执行。

「每个工作日早上九点生成一份行业简报」

建之前必须知道的一件事

它执行的时候看不到你今天说过的话。公司名、文件路径、指标口径、格式偏好——全部要写进那段描述里，而且要写得像给一个刚接手的人看。

记忆：越用越懂你的三层机制

云端画像 跨项目 · 只读

平台自动学习的你的工作背景与习惯，下次会话自动带上——你改不了它。

用户级记忆 所有项目 · 可写

你明确要求长期记住的规则与偏好，例如「以后都用中文回答」「涨用红色、跌用绿色」。

工作区记忆 当前项目 · 可写

项目约定、技术选型，以及每天的工作日志——只有这个项目的任务会读到它。

想让它长期记住

直接说「记住这条」，会写进用户级记忆，之后所有项目都生效。

写进用户级记忆

只对某个项目生效

说「记到项目记忆里」，只有这个项目的任务会带上它。

写进工作区记忆

没说过要记的，不会自动留

随口提过的偏好不会被自动保留，开新任务时请重新交代。

下不为例：显式说清楚

演练 3 | 给自己建一条每日自动化

演练 3 · 各自动手

15

分钟 · 建一条真的用得上的

本场目标

带走一条会在你
不看屏幕的时候，
自己跑完的自动化。

别拿「每天签到」凑数。
挑一件你真的想省掉的事。

建之前，先想清三件事

1. **跑什么：**一句话说清任务与产物：汇总哪些文件、输出什么格式、发到哪里。
2. **几点跑：**工作日的早上九点，还是每周五下班前——按你真正会看的时间设。
3. **结果发到哪：**发到你自己的单聊、某个群，还是只存在产物区等你自己看。

直接改这句话就能用

每周一早上 9 点，汇总我上周在 D:\wb-demo\reports 里的周报文件，生成一份 Markdown 摘要（含本周完成、下周计划、风险三项），并发送到飞书「我」的单聊。

替换掉路径、时间与推送对象，就是你自己的一条。

写完先读一遍再保存

三个月后的它，只能看到这段文字。凡是「你应该知道」的信息，都要写进去。

第 4 章 · P26–P32 · 60 分钟

实战演练营
不是演示，是交东西



60 分钟，四张任务卡，按组抽签。

每张卡都写死了验收标准——不达标就是没做完。

业务 2 张　技术 1 张　混合 1 张

四张任务卡总览

任务卡 A 业务

会议纪要 → 汇报 PPT

输入：一份 40 分钟会议的录音转写，约 6000 字。

交付物：一份 8 页的汇报 PPT，含结论页与行动项。

验收：每页标题即结论；数字与原文一致；不编造没出现过的人。

任务卡 B 业务

多份 Excel → 汇总看板

输入：12 家门店的月度经营表，字段格式并不完全一致。

交付物：一张汇总看板 + 一份异常门店清单。

验收：统计口径写在页脚；异常判定规则写清楚；数字可回算。

任务卡 C 技术

陌生代码库 → 定位并修复

输入：一个能稳定复现的缺陷描述，加上完整代码库。

交付物：定位结论 + 修复 diff + 验证脚本的运行结果。

验收：先复现、再修复、最后跑通验证；没跑通不算修好。

任务卡 D 混合

调研 → 报告 → 发飞书

输入：一个调研主题，例如「同行最近三个月的产品更新」。

交付物：调研报告 + 已发送的飞书消息 + 归档记录。

验收：每条结论都能追到来源链接，不接受「综合来看」。

怎么用这四张卡

不是看老师演示，是你们自己交东西。

- **分组**

  4 人一组，抽签或自选，同组尽量混岗位。

- **节奏**

  20 分钟做，3 分钟讲，剩 2 分钟接受提问。

- **交付**

  必须真的产出一个文件，口述不算完成。

- **评分**

  交付物完整度 / 指令清晰度 / 安全动作 / 可复用性。

卡上的**验收标准就是评分线**，一条不达标就扣一项。

任务卡 A ｜ 会议纪要 → 汇报 PPT

1　喂进原始材料

会议录音转写，约 6000 字，不做任何预处理

2　让它先出大纲

按议题分页，每页标题写成一句结论

3　按大纲生成 PPT

指定页数、页面结构、表格字段——不指定它自己发挥

4　核对数字后导出

逐页比数字与原文，对不上就返工

这张卡在练什么

把一份乱糟糟的原始材料，变成一份能直接站上去讲的汇报。

指令可以直接抄这句

把这份会议转写整理成一份 8 页的汇报 PPT：第 1 页写会议结论，第 2–6 页按议题分页、每页标题就是那一页的结论，第 7 页是行动项表格（负责人 / 截止时间），第 8 页写遗留问题。数字必须与转写原文一致，不要编造没出现过的人名。

最容易翻车的三处

· 没写输出格式 → 拿回来一堆散文，还得自己排版

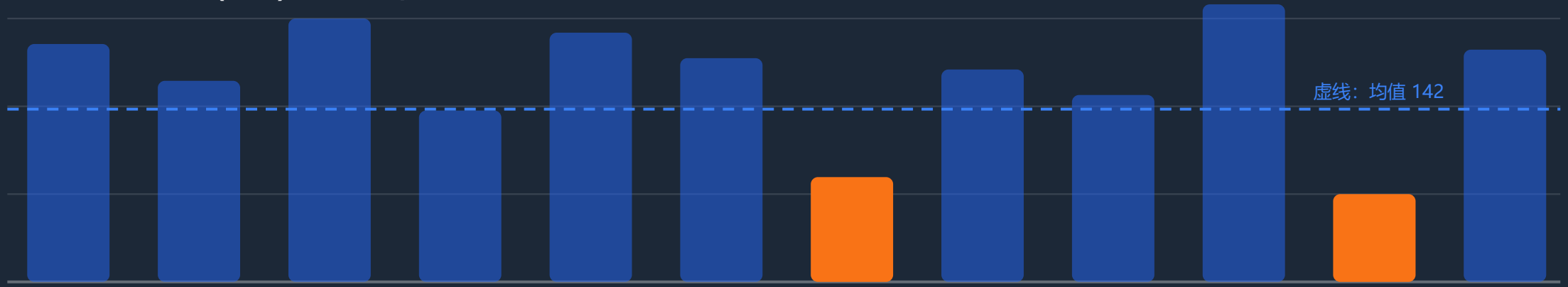
· 没写「数字与原文一致」 → 它会开始合理推算

· 没写页数 → 它可能给你 30 页，也可能给你 3 页

验收：每页标题即结论，数字与原文一致，无编造参会人。

任务卡 B ｜ 多份 Excel → 汇总看板

12 家门店销售额（万元）· 汇总看板示意



口径：含税销售额，退货已扣减，统计周期 1 月 1 日 – 1 月 31 日 ｜ 橙色两根 = 低于均值 60% 的异常门店（G、K）

1 先要字段映射

12 份表字段名不一样。让它先输出一张字段对照表，你确认后再合并。

省一次返工

2 口径写在页脚

含税还是不含税、退货是否扣减、统计周期到哪一天——写死，别让它替你定。

数据可信

3 异常规则要你给

「低于均值 60% 标为异常」比「找出异常门店」靠谱得多。

可复算

4 随手回算两家

挑两家门店手工算一遍，对不上就带着具体数字让它重做。

验收动作

任务卡 C ｜ 陌生代码库 → 定位并修复

STEP 1
复现
先把问题稳定跑出来。不能稳定复现的问题，后面全是猜。

别跳过：直接凭报错猜原因

STEP 2
定位
让它读整个代码库、跨文件搜索调用点，结论要落到文件名和行号。

别只给一句「大概是这里」

STEP 3
修复
出 diff，说清改了什么、为什么改、动了哪几个文件。

别让它顺手重构无关代码

STEP 4
验证
跑验证脚本或测试用例。没跑通验证的修复，不算修好。

别只看它说「已修复」就收工

回环：验证没过，就回到 ② **定位**，不要直接改第三次——第三次改的一定不是根因。

这张卡在练什么

在一个你从没读过的代码库里，走完一次完整的排障闭环。

指令可以直接抄这句

在这个仓库里复现这个问题：用户上传的 CSV 超过 2000 行时，页面会卡死。先复现，再定位到具体文件和行号，然后给出最小改动量的修复 diff，最后跑一遍现有测试。把 diff 和测试输出一起给我，不要顺手重构无关代码。

三条安全底线

· 别在项目仓库根目录下开「完全访问权限」

· 动手前先确认工作目录，先备份再让它改

· 没跑通验证脚本的修复，一律不算修好

任务卡 D ｜ 调研 → 报告 → 发飞书

STEP 1

联网调研

搜 5–8 个来源，交叉验证后再下结论

能力：联网搜索

STEP 2

出报告

每条结论都挂在来源上，附链接

能力：文档生成

STEP 3

发飞书群

发送前它会把内容给你确认一次

授权：连接器

STEP 4

归档到知识库

存进资料库，下次可以直接引用

授权：资料库

一句话交代四个环节 —— 这就是「一条指令跑完一条流水线」的样子。

先授权，再演示

连接器必须先连好并授权；涉及对外发送的动作，它会先要你确认一次才执行。

授权边界

让它自己挂来源

在指令里写死「每条结论附来源链接」，你就省掉逐条核实的时间。

可核查

整条链路一次说清

一句话交代四个环节。拆成四次对话，上下文就断了，效率反而更低。

别拆散

成果展示与互评

展示环节

3

分钟讲清，2 分钟答问

只讲四件事

① 你要解决什么

② 你写的那句指令

③ 交出来的文件

④ 中间卡在哪一步

第 ④ 条最值钱。

别人踩过的坑，才是这次演练的产出。

四个评分维度，每项 1–5 分

同桌互评 + 讲师抽查

01 交付物完整度

该交的文件都交了吗？验收标准逐条对过了吗？

02 指令清晰度

四要素齐不齐？换个人拿去能不能跑出差不多的结果？

03 安全动作

有没有先看权限、先备份？高风险动作有没有先出方案？

04 可复用性

这套做法能不能变成一条技能或自动化，下次直接用？

第 5 章 · P33–P37 · 15 分钟

避坑与进阶
怎么才能不翻车

前面五章讲的是怎么把活做成，

这一章只讲一件事：别把不该动的动了。

七个高频坑　安全四道检查　90 天路线图



七个高频坑

事故记录

7

条高频坑 · 每一条都有代价

前面讲的都是
怎么把活做成，
这一页讲怎么不翻车。

这不是提醒清单，
是别人已经付过学费的记录。

1 **拿模糊指令去清理个人目录**
它会先只出报告；但若开着完全访问，可能直接动手。

2 **拿桌面、下载或项目仓库根目录当工作空间**
误伤成本最高，改错一次很难恢复。

3 **一条任务里塞进太多不相关的事**
上下文变长之后，它会忘掉前面说过的约束。

4 **不写输出格式**
必然返工，拿回来的东西还得自己重排一遍。

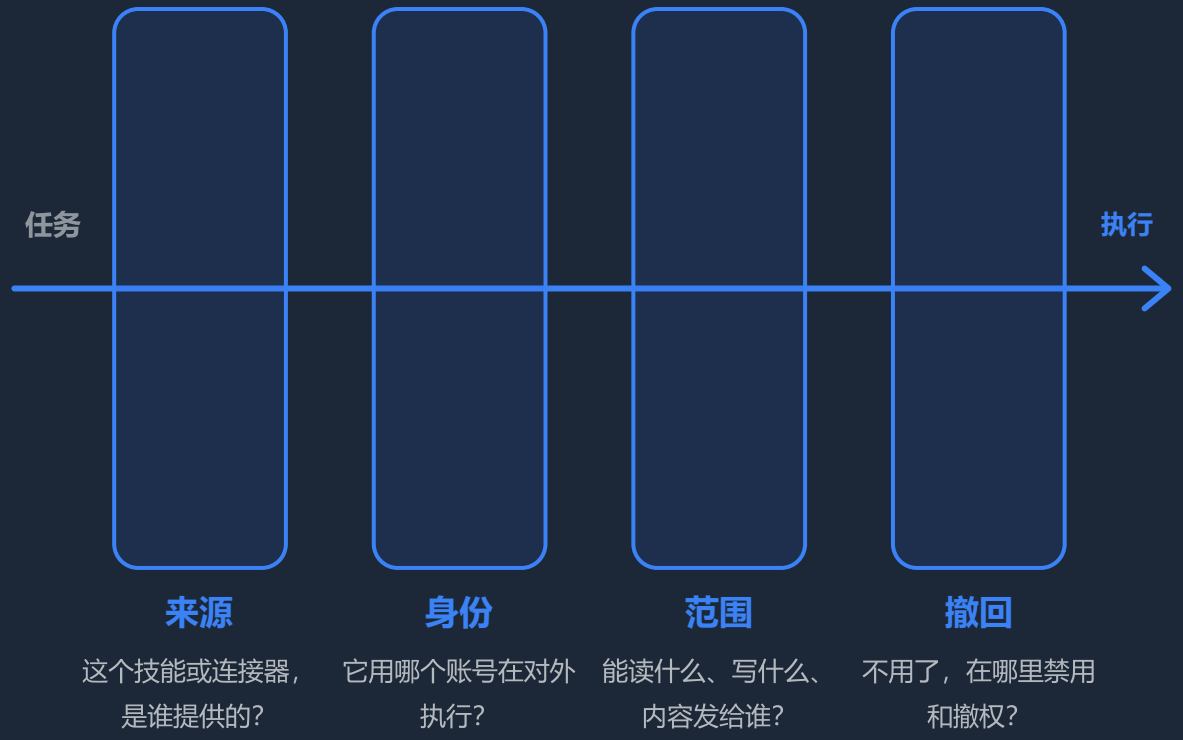
5 **验收只看「任务已完成」**
文件可能没落到你要的目录——先看变更，再看产物。

6 **自动化的描述写得太简略**
三个月后的它只看到那段文字，路径和口径全丢了。

7 **长期开着完全访问权限**
一次误操作就够后悔。只在临时环境短开，用完即关。

动手前的安全四道检查

一条任务要穿过这四道闸门，才算真正动手



动手前 60 秒，照着念一遍

四道全部答得上来，再开始。

1. **来源可信吗**
   官方推荐优先。社区作品要先看它的权限申请和脚本内容。
2. **身份正确吗**
   它用的是你本人的账号——你的权限边界，就是它的边界。
3. **范围清楚吗**
   你知道它能碰哪些目录、哪些系统，以及会把什么传出去。
4. **退路明确吗**
   先找到「在哪关掉、在哪断开」这个入口，再决定要不要连。

这些场景别开完全访问

生产数据 · 客户资料 · 财务资料 · 唯一副本 · 没有备份的重要文件。

90 天养成路线图

第 1–2 周
把话说清楚

每次派活都写满四要素；每次验收都走一遍「变更 → 产物」。

第 3–8 周
沉淀可复用的东西

挑 3 个每月都做的活沉成技能；再建 2 条自动化：日报 + 周报。

第 9–12 周
往团队推

把跑顺的流程整理成可分享的技能包；统一工作空间与权限约定。

只练两件事

· 每次派活都写满四要素

· 每次验收都走「变更 → 产物」

目标：一周内不再出现「重新问一遍」

开始沉淀

· 挑 3 个每月都做的活，沉成技能

· 建 2 条自动化：日报 + 周报

目标：手上至少有 5 件不靠人盯的活

往团队推

· 把跑顺的流程整理成技能包

· 统一工作空间划分与权限约定

目标：新人照着能跑出同样的结果

1

一个任务 · 一个工作空间 · 一条自动化

今天就挑一件真活

「能不能帮你把活干完，

比任何宣传都更有说服力。」

挑一件下周本来就要完成的工作

把它真正交给它跑一次