2026 • AI PRACTICAL GUIDE

# Antigravity 保姆级 实战攻略

从安装、安全配置，到 AI 编程、
真实项目、GitHub Pages 上线。

---

15 章 • 50 项验收操作 • 20 条完整提示词
3 个项目实战 • 7 天执行路线 • 官方来源

**AI 酋长 Andy**

独立原创 • 非 Google 官方文档 • 更新核查 2026-11-23

个人官网 https://metaver.vip/ • 微信 AIPMAndy

# 阅读路线

先完成第 1～4 章的安全与基础；接着完成实战 A 的网站开发、测试和上线；再按需求学习旧站修复、企业业务页及进阶扩展。

> **重要安全说明**
> Antigravity 可能运行命令、修改文件或连接外部工具。初学者应启用审查型权限，严禁未经确认的删除、上传密钥、推送代码和高权限执行。免费计划不代表无限额度。


| | | |
|---|---|---|
| 01 | 先弄明白：Antigravity 到底是什么 | 8 min |
| 02 | 下载安装：Mac / Windows 从零开始 | 12 min |
| 03 | 安全第一：Agent 可以改文件，必须有边界 | 12 min |
| 04 | 会提需求，比会敲指令更重要 | 10 min |
| 05 | 实战 A：0 到 1 做个人品牌官网（定义） | 15 min |
| 06 | 实战 A：生成页面并在本地跑通 | 20 min |
| 07 | 实战 A：测试，不合格就不能交付 | 20 min |
| 08 | 实战 A：部署 GitHub Pages 并查线上 | 20 min |
| 09 | 实战 B：用 Antigravity 维护已有网站 | 25 min |
| 10 | 实战 C：做能接住客户的业务落地页 | 30 min |

11 Agent App、IDE、CLI：如何分工不打架 14 min

12 Rules、Skills、MCP，扩展能力不是乱装插件 15 min

13 配额、常见报错与风险排查 12 min

14 交付一个别人能接手的项目 12 min

15 照着做：7 天实战计划 7 min

本书所有操作都需要读者在自己的电脑上实际执行；本 PDF 不代表已运行用户的 Antigravity 客户端。

CHAPTER 01 / 15 · 认知

# 先弄明白：Antigravity 到底是什么

本章目标：选对 App / IDE / CLI，不在工具选择上浪费一天 · 参考 8 分钟

## 1. 认识四种入口

Antigravity App 是管理 Agent 任务的桌面入口；Antigravity IDE 是带代码编辑器的开发环境；CLI 在终端用 agy；SDK 给开发者编排 Python 代理。初学者从 IDE 开始，想多任务管理再学 App。

**验收：**能用一句话说出 IDE 与 App 的区别。

相关指令：让输出更适合小白（见提示词库）

## 2. 看清能做与不能做

可以让 Agent 分析代码、编辑文件、运行测试、产出计划与验证证据。但它不是无需审核的自动赚钱机器，也不能替代本人核验代码、授权、上线和真实客户结果。

**验收：**写下一个具体目标、一个不希望它碰的文件夹。

相关指令：需求拆解成最小可验证任务（见提示词库）

## 3. 理解费用和限额

官方提供个人免费层，但请求额度、模型可用性、付费权益会变动；在官网 Pricing 和产品设置中看当前账户条件。避免根据旧截图判断“永久无限免费”。

**验收：**已打开官方定价页并确认个人计划。

# 下载安装：Mac / Windows 从零开始

本章目标：使用官方渠道完成首次登录与工作区准备 · 参考 12 分钟

## 1. 选择正确安装包

打开 antigravity.google/download，Mac 按 Apple Silicon 或 Intel 选版本；Windows 选 x64 或 ARM64；Linux 选对应平台。安装后首次启动，不从第三方网盘下载修改版。

**验收：**安装包来源为 Google 官方，系统架构匹配。

## 2. 登录与初始选项

按界面提示用 Google 账户登录。若安装 Antigravity IDE，先选择审查驱动/需要批准的代理工作方式；不同版本菜单会变，遇到不一致以官方当前文档为准。

**验收：**已登录，能看到 Agent 输入框，未开启无限制权限。

相关指令：安全配置与变更边界（见提示词库）

## 3. 创建干净的实验目录

在本地创建 antigravity-lab 文件夹并用 IDE 打开；建议单独建立一个空实验项目，勿把整个用户目录或保存客户密钥的文件夹作为工作区。

**验收：**编辑器中明确显示实验文件夹，且没有私人资料。

CHAPTER 03 / 15 · 必修

# 安全第一：Agent 可以改文件，必须有边界

本章目标：建立最小授权、人工审核与回滚机制 · 参考 12 分钟

## 1. 选择审核型设置

优先选择 Request Review 或 Default 等具有审批/沙箱边界的模式；不同系统与版本名称可能不同。不要为“省点击”直接启用 Turbo、Always Proceed 等高权限模式。

**验收：**确认高风险终端命令需要你审核，记录设置入口。

相关指令：安全配置与变更边界（见提示词库）

## 2. 先检查，再执行

第一次发指令只允许分析目录与输出计划；对安装脚本、文件删除、权限提升、浏览器账号、Git push 和外部 API 授权逐项审查。所有密钥放到 .env 并加入 .gitignore；不把真实密钥发送给 Agent。

**验收：**完成一次只读项目扫描，无文件被改动。

相关指令：先读项目，不准直接改（见提示词库）

## 3. 建立 Git 回滚点

开始改动前执行 git status，清点未提交内容。若是新项目先初始化仓库；若是已有项目，用独立分支并确保关键工作已备份。撤销前先看 diff，避免直接执行破坏性恢复。

> **验收：**能看懂工作区是否干净，知道最近一次安全提交。

相关指令：提交前代码审查（见提示词库）

---

CHAPTER 04 / 15 · 方法

# 会提需求，比会敲指令更重要

本章目标：写出含约束和验收的任务书 · 参考 10 分钟

## 1. 先写用户问题而不是喊做个网站

确定谁会访问网站、访问后要完成什么、不能做什么、用什么证据证明已完成。最小版本只做 1 个首要目标，需求范围不要无限扩大。

**验收：**有目标用户、核心任务、非需求各一条。

相关指令：需求拆解成最小可验证任务（见提示词库）

## 2. 要求 Agent 先给方案

要求列文件清单、步骤、依赖、风险和测试标准；没看懂方案前不要同意修改整个项目。每一步执行后看文件差异和测试输出。

**验收：**拿到可以逐条打钩的验收清单。

相关指令：需求拆解成最小可验证任务（见提示词库）

## 3. 把“很好看”变成可测条件

例如：360px 无横向滚动；首屏 3 秒内知道服务内容；主 CTA 能点击；无假数据；所有交互可用键盘操作。不能证实的性能分数不要写死。

> **验收：**至少写 5 条可以实测而非主观感受的标准。

相关指令：对照验收清单自测（见提示词库）

---

# 实战 A：0 到 1 做个人品牌官网（定义）

本章目标：输出网站原型任务书并批准最小方案 · 参考 15 分钟

## 1. 准备五项真实资料

你的姓名/品牌、一句话定位、三项真实服务、作品链接、联系方式。没有可公开的客户成果就写“尚未公开案例”，不要编造成交额或评价。

**验收：**五项信息齐全，敏感数据已剔除。

## 2. 复制第一版开发指令

在空目录里打开 Agent 面板，复制“第一版个人品牌网站”提示词。要求静态优先，不装包；先看方案，确认后再执行。

**验收：**Agent 能清晰解释要生成哪几个文件。

相关指令：第一版个人品牌网站（见提示词库）

## 3. 确认上线范围

只做能被访问的首页，不做登录、支付、用户数据库；这些功能涉及后端、权限和隐私合规，应等真实需求验证后再单独设计。

**验收：**需求文档标明这次不处理的范围。

相关指令：需求拆解成最小可验证任务（见提示词库）

# 实战 A：生成页面并在本地跑通

本章目标：浏览器中看到真实的响应式页面 · 参考 20 分钟

## 1. 生成 HTML/CSS/JS

先让 Agent 写静态文件：index.html、style.css、script.js 和 README；如果技术栈不需要框架，就不要因为“潮流”引入 React 和一堆依赖。

**验收：**文件可打开，结构清楚，无未解释的第三方请求。

相关指令：第一版个人品牌网站（见提示词库）

## 2. 启动本地服务器

Mac 或 Windows 安装 Python 时，可在项目目录运行 python3 -m http.server 8000（Windows 可能用 py -m http.server 8000），浏览器打开 http://localhost:8000。不要将 0.0.0.0 服务暴露公网。

**验收：**浏览器打开 localhost 后确实看到你的页面。

相关指令：真正跑起来并给我地址（见提示词库）

## 3. 看差异，而不只看结果截图

检查 Agent 提供的 changed files / diff：改了什么？是否多装依赖？是否增加隐蔽追踪脚本？是否删除旧文件？有疑问就撤回该任务，让它解释。

> **验收：**能说出最重要的三个改动。

相关指令：提交前代码审查（见提示词库）

---

# 实战 A：测试，不合格就不能交付

本章目标：形成移动端、链接、键盘和控制台的验收记录 · 参考 20 分钟

## 1. 做三种宽度检查

用浏览器开发者工具分别看 360px、390px 和 1280px；不要只看电脑大屏。页面不应横向溢出，文字不应被卡片遮挡。

**验收：**三种宽度均可阅读和点击。

相关指令：对照验收清单自测（见提示词库）

## 2. 真正点一次每个按钮

测试导航、官网链接、邮箱/微信指引、复制、折叠菜单。不能运行浏览器 Agent 时手动点击并记录事实，不把“理论可用”当作“已验证”。

**验收：**记录至少 5 个实际点击结果。

相关指令：对照验收清单自测（见提示词库）

## 3. 回归与安全检查

看控制台报错、失败网络请求、密钥是否进仓库、是否有危险 shell 命令。发现错误后要求 Agent 用最小补丁修复，再重新执行验收。

> **验收：**有测试清单、实际结果和剩余问题。

相关指令：只改必要的代码（见提示词库）

---

# 实战 A：部署 GitHub Pages 并查线上

本章目标：得到一个真实可访问的 HTTPS 页面 · 参考 20 分钟

## 1. 建立仓库并提交源代码

在自己的 GitHub 创建公开仓库，确认没有密钥后执行 git add、git commit，再按仓库给出的 remote 命令 push。仓库命名、默认分支与组织权限以本人实际账户为准。

**验收：**GitHub 仓库网页能看到 index.html 和提交记录。

相关指令：提交前代码审查（见提示词库）

## 2. 配置 Pages 发布源

进入仓库 Settings → Pages；纯静态项目可以选择 Deploy from a branch，按现有仓库布局选择 main/root；复杂构建项目可改用 GitHub Actions。是否可选取决于权限与仓库设置。

**验收：**Pages 配置页面显示发布源和正式域名。

相关指令：部署 GitHub Pages（见提示词库）

## 3. 线上实地验收

按 GitHub 提供的 Pages 地址访问网站。个人仓库一般为 https://用户名.github.io/仓库名/，但若配置自定义域名或用户主页仓库，URL 会不同。实际检查首页、CTA 与移动端，而非只看 Actions 绿色。

> **验收：**公开地址可直接访问，核心链接真实有效。

相关指令：上线后的客观验收（见提示词库）

# 实战 B：用 Antigravity 维护已有网站

本章目标：安全修复一个真实问题，不误伤其他功能 · 参考 25 分钟

## 1. 读代码，找真实阻塞

先读 README、运行命令、构建配置、Git diff，再找最影响用户的问题。避免一上来就“全站重写”，因为迁移成本和回归风险很高。

**验收：**输出具体问题、复现方式与风险排序。

相关指令：旧站安全诊断（见提示词库）

## 2. 先复现再修复

例如移动端按钮无法点击，写下复现步骤；有条件先增加失败的 UI 测试，再让 Agent 修一处代码并跑原有测试。

**验收：**问题可复现、可测试、修复后同条件通过。

相关指令：制作可复现实验（见提示词库）

## 3. 逐个合并而不是一口气改完

推荐单问题、单分支、单验收，尤其多人协作时避免两个 Agent 同时写同一个文件。每次推送前人工看 diff。

**验收：**改动范围与测试结果均已留下记录。

相关指令：只改必要的代码（见提示词库）

# 实战 C：做能接住客户的业务落地页

本章目标：做出有真实服务闭环的静态业务页面 · 参考 30 分钟

## 1. 先验证业务，不做花哨动画

写清用户痛点、提供的服务、交付范围、工作方法、真实联系方式。先用最简单的页面承接咨询，而不是投入大量时间做无转化的特效。

**验收：**陌生人能在 10 秒内知道你为谁解决什么问题。

相关指令：企业获客落地页（见提示词库）

## 2. 联系入口做低风险闭环

早期可放官网、企业邮箱、微信号复制入口。涉及姓名、手机号、表单收集时，应先明确合法目的、隐私告知、存储位置和权限，未经验证不应把数据直接推送陌生第三方。

**验收：**联系渠道真实、可用，未默认采集个人资料。

相关指令：企业获客落地页（见提示词库）

## 3. 用实际咨询验证页面

记录实际访问来源、咨询问题、咨询转化的真实数字；没有数据时只称为实验，不宣称转化率或收入。下一次只改一个假设。

**验收：**写下一条真实用户反馈或“尚无数据”。

相关指令：7 天复盘（见提示词库）

CHAPTER 11 / 15 • 进阶

# Agent App、IDE、CLI：如何分工不打架

本章目标：多任务有边界，谁改什么、谁验收清楚 • 参考 14 分钟

## 1. IDE 负责看代码和逐行审查

适合新手写第一版、看 diff、调试；利用文件、终端和 Agent 面板将准确上下文交给代理。当前版本快捷键以官方文档和设置为准。

**验收：**能找到 Agent 面板、文件改动与终端日志。

相关指令：先读项目，不准直接改（见提示词库）

## 2. App 负责多个项目的任务管理

Antigravity App 侧重调度与监督多个 Agent；开始时宁可只跑一个重要任务。多任务并行前写清文件边界，不共享同一待修改文件。

**验收：**每个任务有独立输出和停止条件。

相关指令：Agent任务拆分（见提示词库）

## 3. CLI 负责终端内的可复现流程

进阶时可按官网安装 Antigravity CLI，启动命令为 agy。复制 curl | bash 或 irm | iex 前务必核验官方域名、审查脚本与企业安全规范，不应从不明博客复制命令。

**验收：**知道何时用 CLI，也知道如何中止授权。

相关指令：Agent任务拆分（见提示词库）

CHAPTER 12 / 15 · 进阶

# Rules、Skills、MCP，扩展能力不是乱装插件

本章目标：项目规则稳定，第三方能力走最小授权 · 参考 15 分钟

## 1. Rules 写稳定的项目约束

例如静态优先、必须跑测试、不自动 push、不接触 .env。规则是降低重复沟通，不是安全沙箱的替代品。先由你审核规则文件。

**验收：**项目有 5 条具体且可检查的规则。

相关指令：项目规则初稿（见提示词库）

## 2. Skills 封装可复用的 SOP

把一次成功的工作流程整理为包含 SKILL.md 的技能目录，例如“开发前读 README → 先列验收 → 执行最小改动 → 测试 → 汇报证据”。任何第三方 Skill 都要先检查来源和行为。

**验收：**能区分技能说明与真正执行权限。

相关指令：项目规则初稿（见提示词库）

## 3. MCP 只连接可信工具

MCP 可能连接外部服务并暴露数据/写权限。逐项审查发布者、授权范围、日志和撤销方式；先只读，禁止一键给予 GitHub、邮箱、数据库全权限。

> **验收：**准备好 MCP 的权限清单和撤销方案。

相关指令：MCP接入风险核查（见提示词库）

---

# 配额、常见报错与风险排查

本章目标：遇到登录、限额、环境与权限问题能定位 · 参考 12 分钟

## 1. 额度不足不是电脑坏了

先看账户计划、模型额度、当前版本公告；同一任务缩小范围、减少重复试错，避免通过未知中转站购买“无限额度”。不同产品界面的配额不是同一数字。

**验收：**知道官方额度说明入口，不写虚假的无限免费承诺。

相关指令：从报错到修复（见提示词库）

## 2. 代理不执行命令怎么办

检查审查弹窗、工作区信任、沙箱状态和命令权限；确认被阻止的原因，不要为了修一个按钮就永久开启最高权限。

**验收：**能说清被阻止的是哪条具体操作。

相关指令：从报错到修复（见提示词库）

## 3. 站点本地能开线上 404 怎么办

依次检查远程分支是否 push、Pages 是否启用、发布目录、入口 index.html 路径、大小写、部署日志和传播缓存。不要一看到 404 就删仓库重建。

> **验收：**定位到“未部署/路径错误/缓存”中的一个证据。

相关指令：上线后的客观验收（见提示词库）

---

## 4. 浏览器代理不可用怎么办

功能随版本、账户、工作区类型滚动开放。先查看官方 Browser use 文档；如果当前账户不可用，用浏览器开发者工具手动验证或 Playwright 自动化，禁止伪报浏览器测试已通过。

> **验收：**当前环境的测试方式被如实记录。

相关指令：对照验收清单自测（见提示词库）

---

CHAPTER 14 / 15 · 交付

# 交付一个别人能接手的项目

本章目标：源码、说明、测试、回滚和真实链接齐全 · 参考 12 分钟

## 1. 交付清单完整

最少包含 README、源码、如何运行、如何部署、已做的测试、已知限制、来源、开源授权；客户项目另需明确数据所有权与维护责任。

**验收：**陌生人只看 README 能在本机跑起来。

相关指令：输出离线交付包（见提示词库）

## 2. 注意版权与原创素材

不擅自复制别人的页面文字、商标、付费素材；使用开源依赖要核对许可证。商业展示的事实、案例和图片必须可追溯。

**验收：**所有外部素材有来源与可用许可。

相关指令：提交前代码审查（见提示词库）

## 3. 发布前最后一眼

网站线上 URL、GitHub commit SHA、PDF 版次、测试证据分别写清楚。“代码已提交”不等于“网站已上线”。

> **验收：**最终交付链接真实可访问。

相关指令：上线后的客观验收（见提示词库）

---

CHAPTER 15 / 15 · 行动

# 照着做：7 天实战计划

本章目标：每天一件事，最后拿到公开作品 · 参考 7 分钟

## 1. Day 1｜安装、安全、只读扫描

官方下载安装；确认权限不是最高级；打开空工作区；复制“先读项目”提示词。

**验收：**有安全配置截图和空项目说明。

相关指令：先读项目，不准直接改（见提示词库）

## 2. Day 2｜需求、结构与第一版

写 5 条验收标准，生成静态站点第一版，不做额外功能。

**验收：**本地页面已经可以打开。

相关指令：第一版个人品牌网站（见提示词库）

## 3. Day 3｜移动端和链接

用 360/390/1280px 验收，实际点击每个主要按钮。

> **验收：**至少 5 条具体检查有通过或失败记录。

相关指令：对照验收清单自测（见提示词库）

---

## 4. Day 4｜修一个真实问题

从测试中的失败项里选一个，先复现再最小修复。

> **验收：**保留改动 diff 和复测结果。

相关指令：只改必要的代码（见提示词库）

---

## 5. Day 5｜提交并上线

检查密钥、提交 GitHub、配置 Pages、访问线上链接。

> **验收：**公开页面真正可访问。

相关指令：部署 GitHub Pages（见提示词库）

---

## 6. Day 6｜业务闭环

加一个真实可用的联系渠道、服务范围或项目案例，不伪造数字。

> **验收：**可验证的联系路径已测试。

相关指令：企业获客落地页（见提示词库）

---

## 7. Day 7｜复盘与下一轮

记录真实访问与反馈，找出下一周最值得优化的 1-3 件事。

> **验收：**有一份基于事实的简短复盘。

相关指令：7 天复盘（见提示词库）

---

COPY & RUN

# 20 条可直接复制的 Agent 指令

提示词是项目协作的起点，不等于授权无限制执行；涉及终端、密钥、Git push、部署和外部工具时，应先检查计划和权限。

## 01. 先读项目，不准直接改 / 安全与分析

你现在是我的 AI 编程助手。先只读检查当前工作区：目录结构、README、依赖、入口文件、Git 状态、最近一次运行方法。输出：①项目在做什么 ②文件地图 ③潜在风险 ④建议的最小任务和验收标准。此轮不得编辑文件、运行安装脚本、删除文件或提交代码。

## 02. 安全配置与变更边界 / 安全与分析

请为这个项目制定安全边界：仅修改当前工作区；不读取工作区外私人文件；不访问浏览器已登录账户；不打印或提交密钥；安装依赖和访问外网前说明用途；禁止破坏性命令。先给出你预计需要的权限清单，等待我明确同意再执行高风险动作。

## 03. 需求拆解成最小可验证任务 / 规划

目标：[填入你想交付的真实结果]。用户：[填写目标用户]。约束：零预算起步、静态优先、手机优先、不收集用户个人数据。请先用表格输出需求、非需求、失败条件、里程碑、可验证的验收清单，并列出最小版本的文件清单。不要开始写代码。

## 04. 第一版个人品牌网站 / 网站实战

在当前空目录实现一个可在浏览器直接打开的个人品牌展示网站。只用 HTML/CSS/少量原生 JavaScript，不引入构建工具或第三方脚本。板块：一句话定位、三个真实服务、代表作品、常见问题、联系方式。色彩黑白灰、留白克制、手机 360px 无横向滚动。先给简要方案，再实现。不得虚构客户、收入和评价。

---

## 05. 真正跑起来并给我地址 / 网站实战

请启动最简单的本地静态服务（例如 python3 -m http.server 8000），说明准确访问地址。列出你实际执行的命令、退出码和可复现步骤。不要声称已运行或已测试，除非你确实拿到了验证证据。

---

## 06. 对照验收清单自测 / 网站实战

请按之前的验收清单逐项检查：360/390/1280px 布局、导航可点、所有 CTA 指向有效目标、表单行为真实、键盘聚焦、对比度、浏览器控制台报错。可以运行浏览器测试时请给截图和测试结果；无法运行时必须明确标为未验证。先修复可复现的问题，再汇报。

---

## 07. 只改必要的代码 / 开发规范

现在修复这个问题：[粘贴错误和复现步骤]。请先定位根因并说明证据；只修改解决该根因所必需的文件和代码，不做无关重构；提供改动前后差异、回归测试步骤和剩余风险。

---

## 08. 提交前代码审查 / 开发规范

在提交前自查：Git diff 是否含密钥、客户数据或本地路径？是否有与任务无关的改动？是否有删除和破坏性命令？链接是否失效？依赖是否异常？汇总改动文件、测试结果和无法验证项。先展示 diff 摘要，不要自动 git push。

## 09. 部署 GitHub Pages / 发布

请先识别仓库是否为纯静态网站以及 Pages 目前采用 branch 还是 GitHub Actions 发布源。输出最小变更部署方案、准确的 GitHub Pages URL 规则、必要配置和回滚方式。等待确认后才执行 push / 远程发布；部署后要实际访问线上首页和至少 3 个关键链接，不得仅凭 commit 宣称上线。

## 10. 旧站安全诊断 / 已有项目

这是一个已有业务网站。先只读检查代码、依赖、版本、CI、Pages、SEO、移动端、真实入口和错误日志。只提出影响用户的 Top 5 问题，按用户影响/修复风险/工作量排序。任何数据库迁移、删除、批量改写、权限变更都须单独审批。

## 11. 制作可复现实验 / 已有项目

请把我反馈的异常转换为一个最小复现：操作步骤、预期结果、实际结果、环境条件、截图或日志证据。能写自动化测试时先写失败测试，再做最小修复，最后重新跑测试并给出结果。

## 12. 企业获客落地页 / 业务实战

为真实的小型服务公司设计并实现一个静态获客落地页，包含：清晰的服务边界、用户问题、工作流程、真实案例占位说明、常见问题、可点击的官方联系渠道。禁止伪造客户见证、虚构数字和承诺收益。不搭建采集个人数据的后端；任何表单仅在明确设计隐私告知和处理方式后再上线。

---

## 13. Agent任务拆分 / 进阶

将目标拆成 3 个可独立验收的任务：需求与信息架构、UI 与可用性、测试与发布。为每个任务标记独立文件边界、输入、输出、依赖和停止条件。高风险操作必须经我审核，禁止并行 Agent 同时改同一文件。

---

## 14. 项目规则初稿 / 进阶

根据当前仓库生成一份简洁的项目级规则草稿：优先阅读 README；修改前先看 Git 状态；按现有编码规范；不引入不必要依赖；不读取 .env 和密钥；不自动推送或删除；每次改动都运行可复现的测试。先让我审核，不要擅自安装第三方 Skill 或 MCP。

---

## 15. MCP接入风险核查 / 进阶

我准备接入这个 MCP：[名称/来源链接]。请先审查发布者、仓库维护、权限范围、网络端点、日志与数据流、是否写入外部服务、是否有不可逆操作。按最小权限给出只读试运行流程和撤销方案，不要直接安装或授权。

---

## 16. 输出离线交付包 / 交付

请整理交付：README（安装/启动/部署）、变更日志、可复现测试命令、可截图的成品、许可证与素材来源、已知限制、回滚步骤和下一轮迭代建议。不要把环境变量、认证信息、用户隐私数据打入压缩包。

---

## 17. 7 天复盘 / 复盘

基于真实 Git 提交、实际跑过的测试、线上访问记录与用户反馈，复盘本周：哪 3 个功能产生可验证价值？哪 3 类错误反复发生？有哪些提示词需要优化？提出下周最多 3 项改动，按收益/风险/工作量排序，明确没有数据的推断。

---

## 18. 让输出更适合小白 / 通用

用 12 岁也能理解的中文解释你接下来要做的事。每一步只写：要点哪里/输入什么、为什么要做、怎么知道做对了。先提供一条可以复制执行的最短命令；遇到我的环境和你预期不一致时停止猜测。

---

## 19. 从报错到修复 / 排错

下面是完整报错：[粘贴报错]。我的系统和版本：[填写]。请给出最多 3 个由高概率到低概率的原因，并为每个原因设计不破坏现有环境的验证步骤。先找根因，再给修复，最后检查是否引入新问题。

---

## 20. 上线后的客观验收 / 发布

请验证实际公开地址：[填入 URL]。检查 HTTP 状态、首页标题、移动布局、导航、关键 CTA、robots/sitemap、浏览器错误和缓存问题。分别汇报：已经实际验证的证据、无法验证的项目、阻塞问题。切勿把本地通过当线上通过。

Q & A

# 常见问题

### Antigravity 和 Gemini 有什么不同?

Gemini 是 Google 的模型/产品品牌之一；Antigravity 面向代理式开发工作流，组织 Agent 在本地项目中计划、编辑、执行与验证。具体可选模型和功能按账户与版本变化。

### 完全不会编程能学吗?

可以从静态网页开始。新手仍需要学习文件、路径、Git 和测试的基础概念，否则很容易把“它生成了代码”误认为“产品可以交付”。

### Antigravity 永久免费吗?

官方有个人免费计划，但有使用限额，模型和额度可调整。任何声称“完全无限免费”的教程都需要核对当日官方价格页。

### 我已经用 VS Code，还要换 IDE 吗?

不一定。官方提供多种 IDE 集成，也提供独立 Antigravity IDE 和 CLI。优先选你能稳定审查代码、运行测试的入口。

### Agent 能自动操作浏览器吗?

官方支持浏览器相关能力，但版本、账户与本地/远端工作区存在限制。若当前不可用，用手动 DevTools 或本地 Playwright 验收，不应声称全部用户都能用。

### 能直接部署并收费吗?

可以辅助生成和部署产品，但收款、个人数据、隐私协议、税务及安全要求需要独立设计和人工验收，不能以一条提示词代替。

### 需要买 Gemini API Key 才能学吗?

本教程从 Antigravity 的个人账户登录和 IDE 操作入门，不要求为第一版静态网站购买 API Key；SDK 或额外 Google Cloud 服务可能涉及单独认证和费用。

### 为什么页面本地正常、上线后空白？

常见原因是资源路径写成根路径、部署目录或 base path 不正确、脚本报错和缓存。先在浏览器控制台与 Network 中定位，而不是直接重写全部代码。

### 教程有真实项目源码吗？

网站提供可下载的原生 HTML 练习模板和提示词；完整网站源码与 PDF 在本教程配套包中。实战中的结果由学习者在本机完成并验收，不伪装为已帮你在 Antigravity 执行。

### 如何判断 Agent 是否真的完成？

要求提交链接、实际退出码、截图/自动化测试、文件 diff、线上 HTTP 验证与尚未完成项。没有证据的“已完成”不能作为验收。

SOURCES

# 引用与核验

本书参考的公开官方资料与事实边界如下；最新界面、政策、计划和服务可用性以官方页面为准。所有外部链接可点击。

**Google Antigravity — Get started**
https://antigravity.google/docs/getting-started
安装路径、产品形态、首次启动

**Google Antigravity — Features**
https://antigravity.google/docs/features
Skills、Rules、MCP、任务编排

**Google Antigravity — Agent settings**
https://antigravity.google/docs/agent-settings
安全权限、沙箱、审核模式

**Google Antigravity — Browser use**
https://antigravity.google/docs/browser
浏览器代理、页面验证与可用性边界

**Google Antigravity — Pricing**
https://antigravity.google/pricing
免费层与付费层的范围、限额

**Google Codelab — Getting Started with Antigravity IDE**
https://codelabs.developers.google.com/getting-started-agy-ide
IDE 首次设置、Agent 面板、审查配置

**Google Codelab — Building with Google Antigravity**
https://codelabs.developers.google.com/building-with-google-antigravity
完整项目开发、测试与产物审查

**Google Antigravity — Download**
https://antigravity.google/download
正版安装渠道、系统要求

**GitHub Docs — Configuring a publishing source for your GitHub Pages site**
https://docs.github.com/en/pages/getting-started-with-github-pages/configuring-a-publishing-source-for-your-github-pages-site
Pages 发布流程和源配置

---

### 教程免费，合作自愿

作者：AI 酋长 Andy。免费阅读无需添加微信。需要具体项目诊断、AI 产品实战陪跑或企业 AI 获客合作，可自愿联系。

个人微信：**AIPMAndy**
个人官网：https://metaver.vip/
公司官网：https://www.a6no1.cn/
在线网站：https://aipmandy.github.io/antigravity/

非 Google 官方出品。文字 CC BY 4.0，网站代码 MIT。第三方商标归其权利人。更新核查 2026-11-23。