

CHAPTER 1

ANTIGRAVITY 介紹

HORAZON

ANTIGRAVITY

什麼是 ANTIGRAVITY?

- Google DeepMind 開發的高階 AI 程式代理 (Agent)。
- 它不只是一個聊天機器人，而是一個**能行動**的虛擬工程師。
- 具備 Agentic Mode (**代理模式**)：能自主規劃、執行複雜任務。
| (類似概念如 Claude Code，但其僅限於 CLI 命令列介面)

關鍵技術發布時程 (TIMELINE)

技術項目	發布時間	說明
Antigravity (Preview)	2025/11	首次私人預覽
Gemini 3 Pro	2025/11	邏輯推理能力大幅提升
Antigravity 1.0	2026/01	正式公開版本
Agent Skills	2025/12	Anthropic推出Skills協定
Antigravity Skills	2026/01	Antigravity可使用Skills協定

核心能力

1. 工具使用 (Tool Use)：

- 讀寫檔案、執行終端機指令、建立資料夾。
- 甚至能操控瀏覽器 (Browser Use) 進行測試。

2. 自我修正 (Self-Correction)：

- 遇到編譯錯誤或執行失敗，會閱讀錯誤訊息並嘗試修復。
- 不會寫完程式就跑掉，會負責到底。

3. 專案視角 (Project Awareness)：

- 不只看單一檔案，能理解整個專案的結構與依賴關係。

如何與 ANTIGRAVITY 協作？

1. Pair Programming (結對編程)：

- 你是架構師 (Architect)，它是實作者 (Implementer)。
- 給予清楚的指令與目標，讓它負責細節。

2. Artifacts (產出物)：

- 它會建立 `path/to/artifacts` 來管理文件。
- 包含：`task.md` (任務清單)、`implementation_plan.md` (實作計畫)。

3. Task Boundary (任務邊界)：

- 透過明確的任務分割，讓工作更有效率。

ANTIGRAVITY VS 傳統 LLM

特性	傳統 LLM (ChatGPT/Claude)	Antigravity
互動方式	一問一答 (Chat)	任務導向 (Task Based)
程式碼	只提供文字片段	直接修改檔案
錯誤處理	需要人貼錯誤訊息	主動偵測並修復
上下文	視窗大小限制	整合 IDE 資訊與專案索引

安裝方式

1. 請前往官方網站：antigravity.google
2. 點擊左上角的 Download 按鈕。
3. 依照指示完成安裝即可。

安裝完成後，請依指示登入帳號。

必備擴充功能

為了獲得最佳體驗，請一併安裝以下擴充功能：

1. Traditional Chinese (Taiwan)

- 將 VS Code 介面中文化，降低學習門檻。
- 搜尋 `Chinese (Traditional)` 安裝繁體中文套件。

2. Quota Monitor

- 監控 AI 使用量與剩餘額度。
- 避免不小心用超量導致暫停服務。

3. 其他 VS Code 擴充功能

- Antigravity 建構於 VS Code 之上。
- 所有 VS Code Marketplace 的套件 (Python, C#, Prettier 等) **皆可安裝！**

設定個人規則 (USER RULES)

讓 Antigravity 更懂你的風格：

1. 點擊介面上的 **三個點 (...)** 圖示。
2. 選擇 **Customizations**。
3. 在 **Rule** 欄位新增以下片段：

```
# Antigravity Settings
```

```
## Global Language / 全局語言
```

- All conversations, plans, task, and outcomes must be in ****Traditional Chinese (Taiwan)****.
- 所有對話、計畫、任務、產出結果請全部使用****繁體中文****。

```
## Code Comments / 程式註解
```

- Use English for variable and function names (standard coding conventions).
- Use ****Traditional Chinese**** for all comments.
- 程式使用英文變數，但註解都使用****繁體中文****。

代理模式 (AGENTIC MODE)

這是 Antigravity 最強大的功能！

- **開啟方式**：在輸入框點擊 **Robot Icon** 切換至 Agentic Mode。(會變成紫色邊框)
- **運作邏輯**：
 - i. 你給出一個 **高層次目標** (例如：幫我做完這個功能)。
 - ii. 它會**暫停對話**，開始**思考計畫**。
 - iii. 它會**連續執行工具** (讀檔、寫檔、測試) 直到任務完成。
 - iv. 最後回報結果給你。

適合用在：重構程式、撰寫單元測試、實作完整功能模組。

專案實作建議

Playground (實驗場)

- 初次使用建議先開一個空資料夾或 Playground 專案。
- 盡情測試指令，熟悉 AI 的思考模式，不用擔心弄壞程式。

正式專案 (Production)

- 熟悉後再導入正式專案。
- 適合：**重構程式碼、補完單元測試、新增獨立功能。**
- **重要：**在讓 Agent 動手前，記得先 **Git Commit** 保護你的進度！

提問技巧 (AGENT MODE)

Agent 模式有兩種思考方式，請依需求選擇：

1. Fast Mode (快速模式)

- 適合簡單問題、修 Bug、解釋程式碼。
- 反應快，但推理深度較淺。

2. Plan Mode (計畫模式)

- 適合複雜任務、重構、新增功能。
- 會先寫 **Plan** 再執行，步驟嚴謹。

模型選擇：建議使用 Gemini 3 pro(High) 以獲得最佳推理能力。

同意授權 (APPROVAL)

當 Agent 嘗試執行「有風險」的操作時 (如寫入檔案、刪除檔案、執行指令)，會跳出詢問視窗。

- **Approve (同意)**：允許這次操作。
- **Reject (拒絕)**：阻止操作，並說明原因。
- **Always Allow (總是同意)**：開啟 Yolo Mode。
 - Agent 會自動執行後續所有操作，速度最快。
 - **風險**：可能會不小心刪錯檔案，請確保有 Git 版控再使用！

ARTIFACTS 與 註解溝通

Agent 會產生兩種主要文件，放在Artifacts資料夾中：

1. Task.md：任務清單與進度條。
2. Implementation_Plan.md：詳細技術規格書。

如何給予回饋？

- 直接在這些 Markdown 檔案中寫下 **註解 (Comments)**。
- 或使用 HTML 註解格式 `<!-- 這裡有錯，請修正 -->`。
- Agent 會讀取這些變更，並依據你的回饋調整計畫。

測試與持續溝通

開發並非終點，而是協作的開始：

1. 功能測試 (Verification)：

- 匯出產品後，請務必進行實際測試 (如 PDF 排版驗證)。
- Agent 會提供 `walkthrough.md` 說明其完成的測試內容。

2. 雙向反饋：

- 若結果不符預期，隨時提出修正要求。
- 溝通越明確，Agent 的修正就越精準。

3. 迭代優化：

- 透過不斷的小規模調整，達到最終理想的品質。

持續溝通是 AI 協作成功的關鍵！