



AI Coding 前 置作業

楊健宇

【AI共創教練X專案管理】

2026/2/10

議程

- | | |
|----|-------------------------|
| 00 | AI Coding前置作業 |
| 01 | 如何用AI建置商業網站(前台) |
| 02 | 如何用AI建置商業網站(前台+資料庫) |
| 03 | 如何設計商業網站(前台+控制項+資料庫) |
| 04 | 如何設計商業網站(後台+Auth/OAuth) |
| 05 | 如何建立AI小助手(Agent Skills) |

AI Coding前置作業大綱

1. 安裝AI IDE (透過Agent進行Vibe Coding)
2. 安裝npm和Node.js (Node管理器及JS網站套件)
3. 安裝gh (版本控制)
4. 安裝uv (Python工具及專案管理)

安裝AI IDE(擇一即可)

- 方式一：下載**Cursor**-搭配Github帳號
- 方式二：下載**VS Code(+GitHub Copilot)**-搭配Github帳號
- 方式三：下載**Antigravity**-搭配Google或Github帳號

The image displays three side-by-side screenshots of software download pages:

- CURSOR**: A dark-themed page for downloading the Cursor AI IDE. It features a large "Download Cursor" button and sections for "macOS", "Windows", and "Linux".
- Visual Studio Code**: A dark-themed page for downloading Visual Studio Code. It shows download links for Windows, Linux (.deb, .rpm), and Mac.
- Google Antigravity**: A light-themed page for the Antigravity AI IDE. It features a large "Experience liftoff with the next-generation IDE" headline and a "Download for Windows" button.

【AI共創教練X專案管理】

2026/2/10

winget install(Win10/11)

- **winget install** 是 Windows 封裝管理員 (WinGet) 的一個命令，用於在 Windows 10/11 系統上透過指令列快速尋找、下載並安裝軟體應用程式，省去手動去網站下載和執行安裝程式的繁瑣步驟
- 安裝**npm**和**Node.js** (Node管理器及JS網站套件)
`winget install OpenJS.NodeJS.LTS`

- 安裝**gh** (版本控制)

```
winget install GitHub.cli
```

- 安裝**uv** (Python工具及專案管理)

```
winget install astral-sh.uv
```

感謝聆聽 Thank You

共創與合作

感謝您的時間與支持。我們以AI與團隊協作，啟動更好的可能。

保持連結

期待與您持續交流，共同推動創新與影響力。



安裝Node.js和npm

- 用Powershell安裝Node.js (包括安裝npm)

winget install OpenJS.NodeJS (安裝最新版)

winget install OpenJS.NodeJS.LTS (安裝Long-Term Support版，推薦)

```
PS C:\> winget install OpenJS.NodeJS
找到 Node.js [OpenJS.NodeJS] 版本 25.4.0
此應用程式已由其擁有者授權給您。
Microsoft 不負任何責任，也不會授與協力廠商封裝的任何授權。
正在下載 https://nodejs.org/dist/v25.4.0/node-v25.4.0-x64.msi 30.7 MB / 30.7 MB
已成功驗證安裝程式雜湊
正在啟動套件安裝...
安裝程式會要求以系統管理員身分執行。預期提供提示。
已成功安裝
PS C:\>
```

(安裝Node.js也會安裝npm)

- 檢查Node和npm安裝版本

```
PS C:\> node -v
v25.4.0
PS C:\> npm -v
11.7.0
```

- 在powershell安裝npm失敗，設定PS執行原則

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
```

安裝gh

- 用Powershell安裝gh

```
winget install GitHub.cli
```

```
PS C:\> winget install GitHub.cli
找到 GitHub CLI [GitHub.cli] 版本 2.83.2
此應用程式已由其擁有者授權給您。
Microsoft 不負任何責任，也不會授與協力廠商封裝的任何授權。
正在下載 https://github.com/cli/cli/releases/download/v2.83.2/gh\_2.83.2\_windows\_amd64.msi
[██████████] 17.7 MB / 17.7 MB
已成功驗證安裝程式雜湊
正在啟動套件安裝...
已成功安裝
PS C:\> |
```

- 檢查gh安裝版本

```
gh --version
```

```
PS C:\> gh --version
gh version 2.83.2 (2025-12-10)
https://github.com/cli/cli/releases/tag/v2.83.2
PS C:\> |
```

安裝uv

- 用Powershell安裝uv

```
winget install astral-sh.uv
```

```
PS C:\> winget install astral-sh.uv
'msstore' 來源要求您必須先檢視下列合約，再使用。
Terms of Transaction: https://aka.ms/microsoft-store-terms-of-transaction
來源需要將目前電腦的 2 個字母地理區域傳輸到後端服務，才能正確(例如 "US")。

是否同意所有來源合約條款？
[Y] 是 [N] 否：y
找到 uv [astral-sh.uv] 版本 0.9.26
此應用程式已由其擁有者授權給您。
Microsoft 不負任何責任，也不會授與協力廠商封裝的任何授權。
此套件需要下列相依性：
- 套件
  Microsoft.VCRedist.2015+.x64
正在下載 https://github.com/astral-sh/uv/releases/download/0.9.26/uv-x86\_64-pc-windows-msvc.zip
20.9 MB / 20.9 MB
已成功驗證安裝程式雜湊
正在擷取封存...
已成功擷取封存
正在啟動套件安裝...
已修改路徑環境變數；重新啟動命令介面以使用新值。
新增的命令列別名："uvx"
新增的命令列別名："uv"
新增的命令列別名："uvw"
已成功安裝
PS C:\> |
```

winget解決安裝套件

- 用powerhell列出和Node.js或npm有關套件

範例：`winget list nodejs`

- 以<Package_ID>解除套件

`winget uninstall --id <Package_ID>`

範例：`winget uninstall --id OpenJS.NodeJS`

- 以"name"解除套件

範例：`winget uninstall --name "Node.js"`

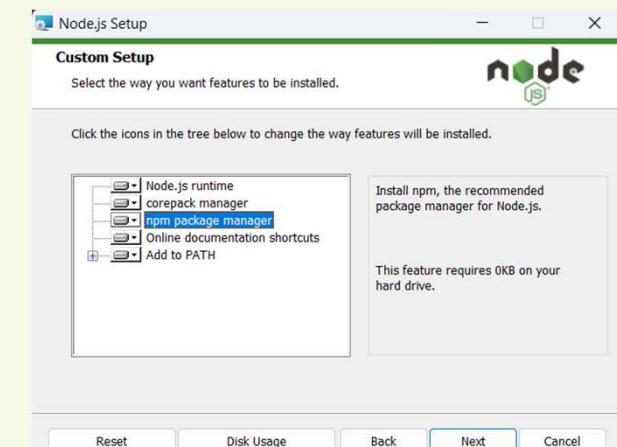
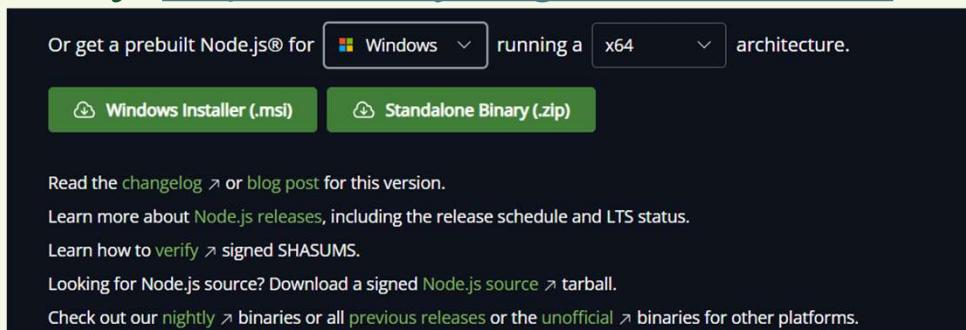
- 重新安裝套件強制清除之前殘餘檔案(假設安裝套件有問題)

範例：`npm cache clean --force`

```
PS C:\> winget list nodejs
名稱 識別碼 版本 來源
-----
Node.js OpenJS.NodeJS 25.4.0 winget
PS C:\> winget list npm
名稱 識別碼 版本 來源
-----
Node.js OpenJS.NodeJS 25.4.0 winget
PS C:\> winget uninstall --id OpenJS.NodeJS
找到 Node.js [OpenJS.NodeJS]
正在啟動套件解除安裝...
已成功解除安裝
PS C:\>
```

安裝Node.js和npm

- Node.js 是一個開源、跨平台的 JavaScript 執行環境，讓 JavaScript 不再只能在瀏覽器執行，而是能在伺服器端（後端）運行，用於**建立高效能的 Web 應用程式**、命令列工具與腳本
- 安裝Node.js <https://nodejs.org/en/download/>



- 安裝Node.js通常也會安裝npm(套件管理工具)
- 確認node版本和npm版本

```
node -v
```

```
npm -v
```



更新npm

- Node Package Manager (npm) 這個工具**管理 JavaScript 套件**，它是 Node.js 預設的套件管理系統，讓開發者能輕鬆安裝、分享和管理專案所需的第三方函式庫、框架與工具

如何使用 npm 安裝：通常隨 Node.js 附帶，可輸入 `npm -v` 檢查版本

1. **安裝套件**：在終端機執行 `npm install <套件名稱>`，例如 `npm install express`
2. **依賴管理**：在 `package.json` 中設定後，只需執行 `npm install` (不帶套件名)，即可安裝所有定義的套件

(範例)

- 更新npm至最新版本
`npm install -g npm`
- 取消funding訊息
`npm config set fund false --location=global`

安裝uv

- <https://github.com/astral-sh/uv/releases> 下載本機OS版本
- 需設置PATH環境變數

Download uv 0.9.25		
File	Platform	Checksum
uv-aarch64-apple-darwin.tar.gz	Apple Silicon macOS	checksum
uv-x86_64-apple-darwin.tar.gz	Intel macOS	checksum
uv-aarch64-pc-windows-msvc.zip	ARM64 Windows	checksum
uv-i686-pc-windows-msvc.zip	x86 Windows	checksum
uv-x86_64-pc-windows-msvc.zip	x64 Windows	checksum
uv-aarch64-unknown-linux-gnu.tar.gz	ARM64 Linux	checksum
uv-i686-unknown-linux-gnu.tar.gz	x86 Linux	checksum
uv-powerpc64-unknown-linux-gnu.tar.gz	PPC64 Linux	checksum
uv-powerpc64le-unknown-linux-gnu.tar.gz	PPC64LE Linux	checksum
uv-riscv64gc-unknown-linux-gnu.tar.gz	RISCV Linux	checksum
uv-s390x-unknown-linux-gnu.tar.gz	S390x Linux	checksum
uv-x86_64-unknown-linux-gnu.tar.gz	x64 Linux	checksum
uv-armv7-unknown-linux-gnueabihf.tar.gz	ARMv7 Linux	checksum
uv-aarch64-unknown-linux-musl.tar.gz	ARM64 MUSL Linux	checksum
uv-i686-unknown-linux-musl.tar.gz	x86 MUSL Linux	checksum
uv-x86_64-unknown-linux-musl.tar.gz	x64 MUSL Linux	checksum
uv-arm-unknown-linux-musleabihf.tar.gz	ARMv6 MUSL Linux (Hardfloat)	checksum
uv-armv7-unknown-linux-musleabihf.tar.gz	ARMv7 MUSL Linux	checksum

如何使用uv

- 建立一個 my-project 專案，

```
mkdir my-project
```

```
cd my-project
```

```
uv init
```

```
my-project/
├── .python-version # 指定 Python 版本
├── main.py # 主程式進入點
└── pyproject.toml # 專案設定檔（核心）
    └── README.md # 專案說明
```

- 當程式寫好之後，uv 會幫你在專案目錄下建立 .venv，

```
uv run python main.py
```

- 移植至新目標主機安裝 uv，uv 幫你安裝需要套件並且建立 .venv

```
uv run python main.py 或 uv sync
```

在VS Code安裝Gemini 3 Pro(1/2)

• 需安裝Gemini CLI

使用 Gemini CLI 時，只要將「預先發布功能」設定變更為 **true**，即可在使用者或工作區層級啟用 Gemini 3。

如要為特定工作區套用「搶先體驗功能」設定，請務必從該工作區的目錄執行 Gemini CLI。

1. 關閉 VS Code。

2. 如果尚未安裝，請[安裝最新穩定版 Gemini CLI](#)。

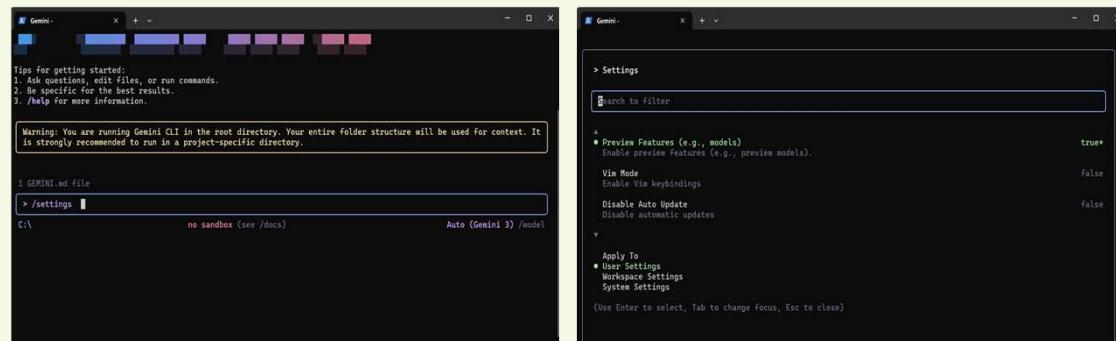
`npm install -g @google/gemini-cli`

3. 安裝完成後，請執行下列指令來執行 Gemini CLI：

Gemini

4. 在 Gemini CLI 中，使用下列指令開啟設定：

`/settings`



5.前往「預先發布版功能」設定，然後在工作區或使用者層級啟用這項功能。

6. 重新開啟 VS Code。

在VS Code安裝Gemini 3 Pro(2/2)

- 在使用者或工作區層級啟用 Gemini CLI 搶先版功能設定

從特定專案目錄或其子目錄執行 VS Code 代理模式或 Gemini CLI 時，才會套用這個檔案。這項設定會覆寫使用者設定。

1.如果工作區根目錄中沒有下列目錄和檔案，請建立：

.gemini/settings.json

2.在檔案中新增下列內容：

```
{  
  "general": {  
    "previewFeatures": true  
  }  
}
```

3.儲存檔案並重新開啟 VS Code。

這些設定會套用至目前使用者的所有專案。。

1.如果檔案系統中尚未建立下列目錄和檔案，請加以建立：

Linux/macOS :

~/.gemini/settings.json

Windows :

C:\Users\<YourUsername>\.gemini\settings.json

2.在檔案中新增下列內容：

```
{  
  "general": {  
    "previewFeatures": true  
  }  
}
```

3.儲存檔案並重新開啟 VS Code。