

# Windows 從零開始：Git 安裝、GitHub 設定、專案上傳與跨機器使用實務手冊

最新版：2025-09-02 | 作者：ChatGPT 助教 | 平台：Windows (PowerShell 指令為主)

Windows 從零開始：Git 安裝、GitHub 設定、專案上傳與跨機器使用實務手冊

最新版：2025-09-02 | 作者：ChatGPT 助教 | 平台：Windows (PowerShell 指令為主)

本手冊帶你從 0 開始：安裝 Git、設定 GitHub (SSH 金鑰)、在本機初始化專案、推上 GitHub、以及在新電腦複製 (clone) 並重建環境與再次上傳的完整流程。所有步驟皆以 Windows 平台為主，指令使用 PowerShell 語法 (有提到 CMD 時會特別註明)。

小抄 (整體順序)

- 1) 安裝 Git    2) 設定 Git 身分    3) 建 SSH 金鑰並加到 GitHub    4) 測試 ssh -T
- 5) GitHub 建空 repo    6) 本機建立專案、初始化、首次 commit    7) 綁定 origin    8) push
- 9) (可選) 建 Conda/venv 環境與匯出規格檔    10) 新電腦 clone + 重建環境 + 執行 + push

## # 1. 安裝 Git (Windows)

### 1.1 先檢查是否已安裝：

```
PS> git --version
```

```
PS> where git
```

若顯示版本號 (如 git version 2.xx.x.windows.x) 且路徑存在，表示已安裝。

### 1.2 若未安裝，建議用 winget (Windows 10/11 通常已內建)：

```
PS> winget install --id Git.Git -e --source winget
```

安裝完成後，關閉再開 PowerShell，重新檢查：

```
PS> git --version
```

```
PS> where git
```

## # 2. Git 一次性基本設定 (全域)

設定你的身份與常用預設：

```
PS> git config --global user.name "你的名字"
```

```
PS> git config --global user.email "你的Email (建議用 GitHub 的 Email)"
```

```
PS> git config --global core.autocrlf true
```

```
PS> git config --global init.defaultBranch main
```

檢查設定：

```
PS> git config --global --list
```

## # 3. 建立 SSH 金鑰並連接 GitHub (推薦)

### 3.1 產生一把「給 GitHub 用」的金鑰 (PowerShell)：

```
PS> ssh-keygen -t ed25519 -C "你的GitHub Email" -f
```

```
"$env:USERPROFILE\.ssh\id_ed25519_github"
```

(在 CMD 用完整路徑版本)

```
CMD> ssh-keygen -t ed25519 -C "你的GitHub Email" -f C:\Users\<你>\.ssh\id_ed25519_github
```

# Windows 從零開始：Git 安裝、GitHub 設定、專案上傳與跨機器使用實務手冊

最新版：2025-09-02 | 作者：ChatGPT 助教 | 平台：Windows (PowerShell 指令為主)

提示 Passphrase 可直接按 Enter (留空) 以免每次輸入。

## 3.2 複製公鑰內容到剪貼簿：

```
PS> Get-Content "$env:USERPROFILE\.ssh\id_ed25519_github.pub" | Set-Clipboard
```

## 3.3 到 GitHub Settings SSH and GPG keys New SSH key 貼上剛複製的內容並儲存。

## 3.4 啟動 ssh-agent 並加入金鑰 (讓系統記住這把鑰)：

```
PS> Set-Service ssh-agent -StartupType Automatic
PS> Start-Service ssh-agent
PS> ssh-add "$env:USERPROFILE\.ssh\id_ed25519_github"
```

## 3.5 測試連線：

```
PS> ssh -T git@github.com
```

若看到：

```
Hi <你的GitHub帳號>! You've successfully authenticated, but GitHub does not provide shell
access.
```

表示成功。若 Permission denied (publickey)，請建立 ~/.ssh/config：

```
PS> notepad "$env:USERPROFILE\.ssh\config"
```

內容：

```
Host github.com
  HostName github.com
  User git
  IdentityFile C:/Users/<你>/.ssh/id_ed25519_github
  IdentitiesOnly yes
  AddKeysToAgent yes
```

## # 4. 在 GitHub 建立「空的」Repository

### 4.1 登入 GitHub 右上角「+」 New repository

### 4.2 輸入 Repository name (例：my-first-project)，Visibility 任選。

### 4.3 \*\*不要勾選\*\* Initialize (README/.gitignore/license 先不要)。

### 4.4 建立後，複製 SSH 位址 (形如 git@github.com:<帳號>:<repo>.git)。

## # 5. 本機建立專案、初始化、首次提交與推上 GitHub

### 5.1 建資料夾並初始化：

```
PS> mkdir $env:USERPROFILE\Documents\myproj
PS> cd $env:USERPROFILE\Documents\myproj
PS> git init
PS> git branch -M main
```

### 5.2 建立基本檔案與 .gitignore：

# Windows 從零開始：Git 安裝、GitHub 設定、專案上傳與跨機器使用實務手冊

最新版：2025-09-02 | 作者：ChatGPT 助教 | 平台：Windows ( PowerShell 指令為主 )

```
PS> @'
__pycache__/
*.py[cod]
*.ipynb_checkpoints/
.env
.venv/
venv/
ENV/
data/*
!data/.gitkeep
.DS_Store
Thumbs.db
.vscode/
'@ | Set-Content -Path .gitignore

PS> New-Item -ItemType Directory -Path .\src.\data -Force | Out-Null
PS> New-Item -ItemType File -Path .\data\.gitkeep -Force | Out-Null
PS> Set-Content -Path .\README.md -Value "# myproj"
PS> @'
from dotenv import load_dotenv
import os

def main():
    load_dotenv()
    print("Hello, myproj! API_KEY =", os.getenv("API_KEY", "NO_KEY"))

if __name__ == "__main__":
    main()
'@ | Set-Content -Path .\src\main.py
PS> Set-Content -Path .\env.example -Value "API_KEY=請填入你的key"
```

## 5.3 ( 可選 ) 建立 Conda 環境並匯出規格：

```
PS> conda create -n myproj python=3.11 pip ipykernel jupyterlab numpy pandas scikit-learn matplotlib python-dotenv -y
PS> conda activate myproj
PS> python -m ipykernel install --user --name myproj
PS> conda env export --name myproj --from-history > environment.yml
```

## 5.4 首次提交：

```
PS> git add .
PS> git commit -m "init: project skeleton"
```

## 5.5 綁定遠端並推上去 ( 把 <帳號>/<repo> 換成你的 )：

```
PS> git remote add origin git@github.com:<你的帳號>/<你的repo>.git
PS> git push -u origin main
```

# Windows 從零開始：Git 安裝、GitHub 設定、專案上傳與跨機器使用實務手冊

最新版：2025-09-02 | 作者：ChatGPT 助教 | 平台：Windows ( PowerShell 指令為主 )

若你在 GitHub 建 repo 時不小心勾了 README，第一次推會被拒，解法：

```
PS> git pull --rebase origin main
```

```
PS> git push
```

## # 6. 日常工作流（之後每次開工）

### 6.1 先拉最新（避免日後衝突）：

```
PS> git pull --rebase
```

### 6.2 修改檔案後提交與推送：

```
PS> git status
```

```
PS> git add -A
```

```
PS> git commit -m "feat: 新功能說明 / fix: 修正某問題"
```

```
PS> git push
```

### 6.3 檢查遠端位址是否為 SSH：

```
PS> git remote -v
```

（應看到 git@github.com:... 才是 SSH；若是 HTTPS 可用 set-url 改回來）

```
PS> git remote set-url origin git@github.com:<你的帳號>/<你的repo>.git
```

## # 7. 案例（Case）：新電腦從 GitHub 取回、重建環境、執行、修改後上傳

### 步驟 A：新電腦準備

#### 1) 安裝 Git：

```
PS> winget install --id Git.Git -e --source winget
```

#### 2) 設定 Git 身分：

```
PS> git config --global user.name "你的名字"
```

```
PS> git config --global user.email "你的Email"
```

```
PS> git config --global core.autocrlf true
```

```
PS> git config --global init.defaultBranch main
```

#### 3) 產生這台電腦專用的 GitHub SSH 金鑰並加入 GitHub：

```
PS> ssh-keygen -t ed25519 -C "你的GitHub Email" -f
```

```
"$env:USERPROFILE\.ssh\id_ed25519_github"
```

```
PS> Get-Content "$env:USERPROFILE\.ssh\id_ed25519_github.pub" | Set-Clipboard
```

（到 GitHub Settings SSH and GPG keys New SSH key 貼上）

啟動 ssh-agent 並加入金鑰：

```
PS> Set-Service ssh-agent -StartupType Automatic
```

```
PS> Start-Service ssh-agent
```

```
PS> ssh-add "$env:USERPROFILE\.ssh\id_ed25519_github"
```

測試：

```
PS> ssh -T git@github.com
```

# Windows 從零開始：Git 安裝、GitHub 設定、專案上傳與跨機器使用實務手冊

最新版：2025-09-02 | 作者：ChatGPT 助教 | 平台：Windows (PowerShell 指令為主)

## 步驟 B：抓下現有專案

```
PS> cd $env:USERPROFILE\Documents
PS> git clone git@github.com:<你的帳號>/<你的repo>.git
PS> cd <你的repo>
```

## 步驟 C：重建並啟用環境（兩種擇一）

【Conda 路線】（若 repo 有 environment.yml）

```
PS> conda env create -f environment.yml
PS> conda activate <環境名稱>
PS> python -m ipykernel install --user --name <環境名稱>
PS> python -m src.main
（或 jupyter lab）
```

【venv 路線】（若 repo 有 requirements.txt）

```
PS> python -m venv .venv
PS> .\.venv\Scripts\Activate.ps1
PS> python -m pip install --upgrade pip
PS> pip install -r requirements.txt
PS> python -m src.main
（或 jupyter lab）
```

## 步驟 D：修改 提交 上傳

```
PS> git status
PS> git add -A
PS> git commit -m "docs: 更新說明檔 / chore: 改環境規格"
PS> git push
```

## # 8. 常見問題與排錯（FAQ）

Q1：ssh -T 出現 Permission denied (publickey)

- 確認你上傳到 GitHub 的公鑰內容與本機 ~/.ssh/id\_ed25519\_github.pub 完全一致：

```
PS> Get-Content "$env:USERPROFILE\.ssh\id_ed25519_github.pub"
```

- 確認 ~/.ssh/config 指定 IdentityFile 正確，並已有 ssh-add：

```
PS> notepad "$env:USERPROFILE\.ssh\config"
```

```
PS> ssh-add -l
```

- 查看偵錯輸出（會列出實際嘗試的金鑰與路徑）：

```
PS> ssh -vT git@github.com
```

Q2：push 時提示 remote contains work that you do not have locally

- 代表遠端已有提交（常見於你勾了 README）：

```
PS> git pull --rebase origin main
```

```
PS> git push
```

Q3：remote origin already exists

- 你先前已加過 origin，可改 URL 或移除後重加：

```
PS> git remote -v
```

# Windows 從零開始：Git 安裝、GitHub 設定、專案上傳與跨機器使用實務手冊

最新版：2025-09-02 | 作者：ChatGPT 助教 | 平台：Windows ( PowerShell 指令為主 )

```
PS> git remote set-url origin git@github.com:<你>/<repo>.git
```

或：

```
PS> git remote remove origin
```

```
PS> git remote add origin git@github.com:<你>/<repo>.git
```

Q4：fatal: not a git repository

- 目前路徑不是 Git 專案，請先 cd 到專案資料夾或先 git init。