

Conda 常用指令小抄 (Windows / PowerShell 版)

更新日期：2025-09-02 | 作者：ChatGPT 助教

這份小抄整理了在 Windows (PowerShell) 下最常用的 Conda 指令與實務建議，包含環境建立、套件管理、環境規格 (environment.yml)、channels 設定、清理、Jupyter Kernel、與 pip 協作等。

0) 先讓 Conda 在 PowerShell 可用

```
PS> where conda
```

```
PS> conda --version
```

```
PS> conda init powershell # 若找不到 conda/activate，執行後關掉重開 PowerShell
```

1) 環境 (env) 管理

```
PS> conda create -n myenv python=3.11 numpy pandas -y # 建環境
```

```
PS> conda activate myenv # 啟用
```

```
PS> conda deactivate # 離開
```

```
PS> conda env list # 列出所有環境
```

```
PS> conda create -n newenv --clone myenv # 複製環境
```

```
PS> conda env remove -n myenv # 刪除環境
```

```
PS> conda run -n myenv python script.py # 不切環境直接執行指令
```

2) 套件 (package) 管理

```
PS> conda search pandas
```

```
PS> conda install -n myenv scikit-learn
```

```
PS> conda update -n myenv scikit-learn
```

```
PS> conda remove -n myenv scikit-learn
```

```
PS> conda update -n myenv --all
```

```
PS> conda list -n myenv
```

```
PS> conda list -n myenv pandas
```

建議：能用 conda 安裝的先用 conda；必要時再 pip install，完後記得更新規格檔。

3) environment.yml (可重現的關鍵)

```
PS> conda env export -n myenv --from-history > environment.yml # 匯出 ( 建議 from-history )
```

```
PS> conda env create -f environment.yml # 用規格建立新環境
```

```
PS> conda env update -n myenv -f environment.yml --prune # 用規格更新 ( 同步刪多餘套件 )
```

備註：--from-history 只記錄你當初明確安裝的包，較精簡且可移植；

若想嚴格鎖版本可不加，但不同平台可能較難解相依。

4) Channels 設定 (conda-forge 推薦)

```
PS> conda config --show channels
```

```
PS> conda config --show-sources
```

```
PS> conda config --add channels conda-forge
```

```
PS> conda config --set channel_priority strict
```

```
PS> conda config --remove channels conda-forge # 需要時再移除
```

說明：conda-forge 更新快、相容性好；strict 有助避免混源相依衝突。

5) 清理與資訊

```
PS> conda clean -a -y # 清快取、索引、未用包
```

```
PS> conda info # 顯示 conda 與環境資訊
```

```
PS> where conda
```

```
PS> where python # 查可能的多版本衝突
```

6) Jupyter Kernel (讓 Notebook 能選到你的環境)

```
PS> conda activate myenv
```

```
PS> python -m ipykernel install --user --name myenv --display-name "Python (myenv)"
```

之後在 Jupyter / VS Code 內即可選擇 Python (myenv) 核心。

7) 與 pip 協作 (必要時)

Conda 常用指令小抄 (Windows / PowerShell 版)

更新日期：2025-09-02 | 作者：ChatGPT 助教

```
PS> conda activate myenv
```

```
PS> pip install package-only-on-pip
```

```
PS> conda env export -n myenv --from-history > environment.yml # 記得更新規格檔
```

若遇到 ResolvePackageNotFound , 多半是 yml 中的 pip 套件或找不到來源 ,

讓 pip 套件列在 yml 的 `pip:` 區塊即可 (conda env export 會幫你處理) 。

8) 常見錯誤速修

- conda/activate 找不到

```
PS> conda init powershell # 然後關掉重開 PowerShell
```

- UnsatisfiableError (解不出相依)

1) 加 conda-forge ; 2) 開 strict ; 3) 放寬版本鎖定

- CondaValueError: prefix already exists 你要建的環境資料夾已存在 , 換名字或先移除

- 混到 Windows Store 的 Python (WindowsApps\python.exe)

設定 App Execution Aliases 關掉 python.exe/python3.exe 或調整 PATH 順序

9) mamba (選配：加速版 conda)

```
PS> conda install -n base -c conda-forge mamba
```

```
PS> mamba create -n myenv python=3.11 -y
```

```
PS> mamba install -n myenv numpy pandas
```

```
PS> mamba env create -f environment.yml
```

說明：mamba 與 conda 語法幾乎相同，但解析速度快很多。

10) 一日工作流 (典型流程)

```
PS> cd $env:USERPROFILE\Documents\myproj
```

```
PS> conda activate myproj
```

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

```
PS> conda install -n myproj <pkg> # 或 pip install <pkg>
```

```
PS> python -m src.main
```

```
PS> jupyter lab
```

```
PS> conda env export -n myproj --from-history > environment.yml # 有加新套件就更新規格
```