

Python虛擬環境與套件安裝介紹

Mar 2024, ANSYS Taiwan

Python強大功能大多來自於豐富的第三方套件。這些套件可以通過 `pip` 這樣的包管理工具安裝。

將所有套件安裝在Python的主安裝目錄中是可行的，但這樣做會有幾個缺點：

1. **版本衝突**：不同的項目可能會需要不同版本的相同套件，安裝在同一目錄下容易造成版本衝突。
2. **管理困難**：當安裝了大量套件後，很難管理每個項目所依賴的套件及其版本。
3. **清理困難**：當不再需要某個項目時，移除相關依賴可能會影響到其他項目。

為了解決這些問題，我們通常會使用**虛擬環境**。虛擬環境是一種技術，可以在不同的目錄中創建隔離的Python運行環境。每個虛擬環境都有自己的Python執行檔和一套獨立的套件，這樣可以：

- **隔離項目依賴**：每個項目使用自己的虛擬環境，避免了套件版本衝突的問題。
- **便於管理**：清晰地管理每個項目所需的依賴，便於維護和更新。
- **易於清理**：項目結束時，直接刪除虛擬環境的目錄即可清除所有依賴，不影響其他項目。

建立虛擬環境的方法很簡單，可以使用Python內建的 `venv` 套件來完成。以下是使用 `venv` 創建虛擬環境的基本步驟：

1. 打開終端或命令提示符。
2. 選擇一個項目目錄，並導航到該目錄。
3. 執行命令創建虛擬環境，例如：`python3 -m venv myenv`（其中 `myenv` 是虛擬環境的名稱）。
4. 啟動虛擬環境：
 - 在Windows上，執行 `myenv\Scripts\activate`。
 - 在Unix或macOS上，執行 `source myenv/bin/activate`。
5. 現在，任何使用 `pip` 安裝的套件都將僅安裝在這個虛擬環境中。

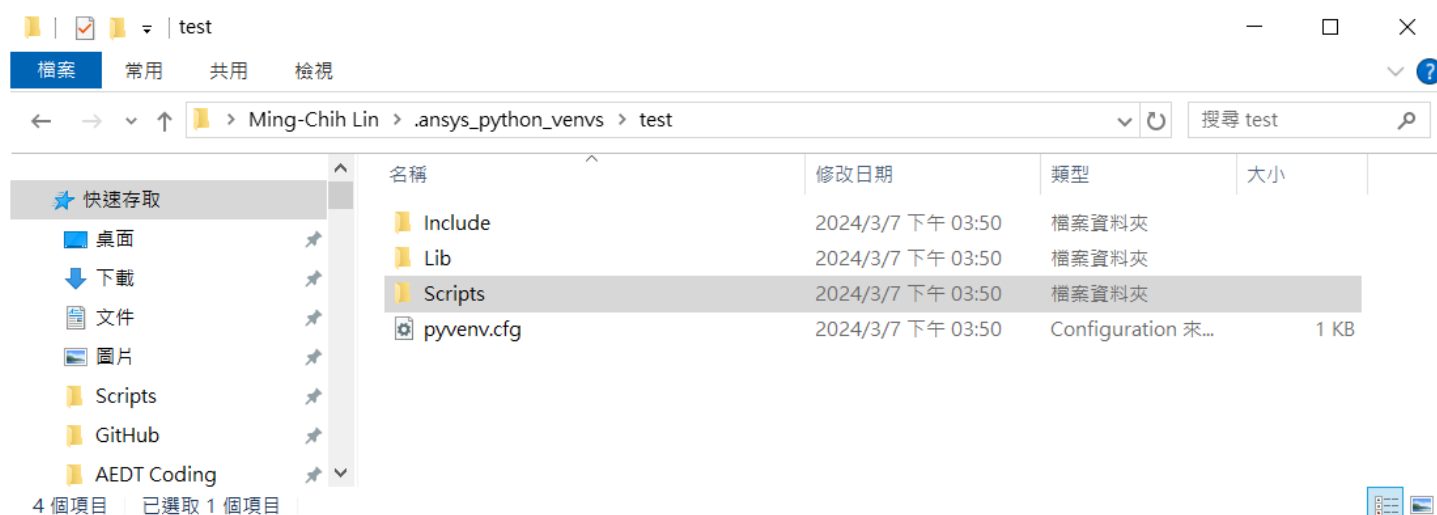
透過這種方式，您可以為每個Python項目創建一個乾淨、隔離的工作環境，有效地管理項目依賴並避免潛在的衝突問題。

虛擬環境目錄

當您使用 `venv`（或 `virtualenv`）創建一個虛擬環境時，會在指定目錄下生成一系列的目錄和檔案。這些組件共同構成了一個隔離的Python運行環境，使其能夠獨立於系統全局環境運行。以下是虛擬環境中常見的目錄和檔案：

- bin（在Windows中為Scripts）目錄**：包含虛擬環境的執行檔，例如 `python`（或 `python.exe` 在Windows上）和 `pip`。此外，這個目錄還包含一個 `activate` 腳本，用於啟動虛擬環境。
- include 目錄**：存放Python標頭檔（header files），這些檔案主要用於編譯安裝某些依賴C語言擴展的Python套件。
- lib 目錄**：這是安裝Python套件的地方。每個虛擬環境都會有自己的 `lib` 目錄，在這個目錄下，您會找到一個Python版本命名的子目錄（例如 `python3.8`），其中包含虛擬環境安裝的所有Python包。
- pyvenv.cfg 檔案**：這是一個配置檔案，記錄了虛擬環境的一些基本設定，如使用的Python解釋器位置等。

這些構件一起工作，使虛擬環境能夠運行獨立於系統Python安裝的應用。啟動虛擬環境（使用 `activate` 腳本）後，系統的 `PATH` 會臨時修改，以優先使用虛擬環境中的Python解釋器和相關工具。這樣，當您在虛擬環境中運行Python或安裝新的套件時，所有操作都將限定在這個隔離的環境內，不會影響到系統中的其他Python環境。



在虛擬環境當中安裝套件

當您啟動（`activate`）虛擬環境之後，實際上是在您的命令行界面（`console`）中修改了一系列環境變數，這樣做的主要目的是為了將當前的Python解釋器切換到虛擬環境中的解釋器。這包括修改 `PATH` 環境變數，使得在這個命令行會話中調用的 `python` 和 `pip` 命令都是指向虛擬環境中的版本，而不是全局安裝的版本。

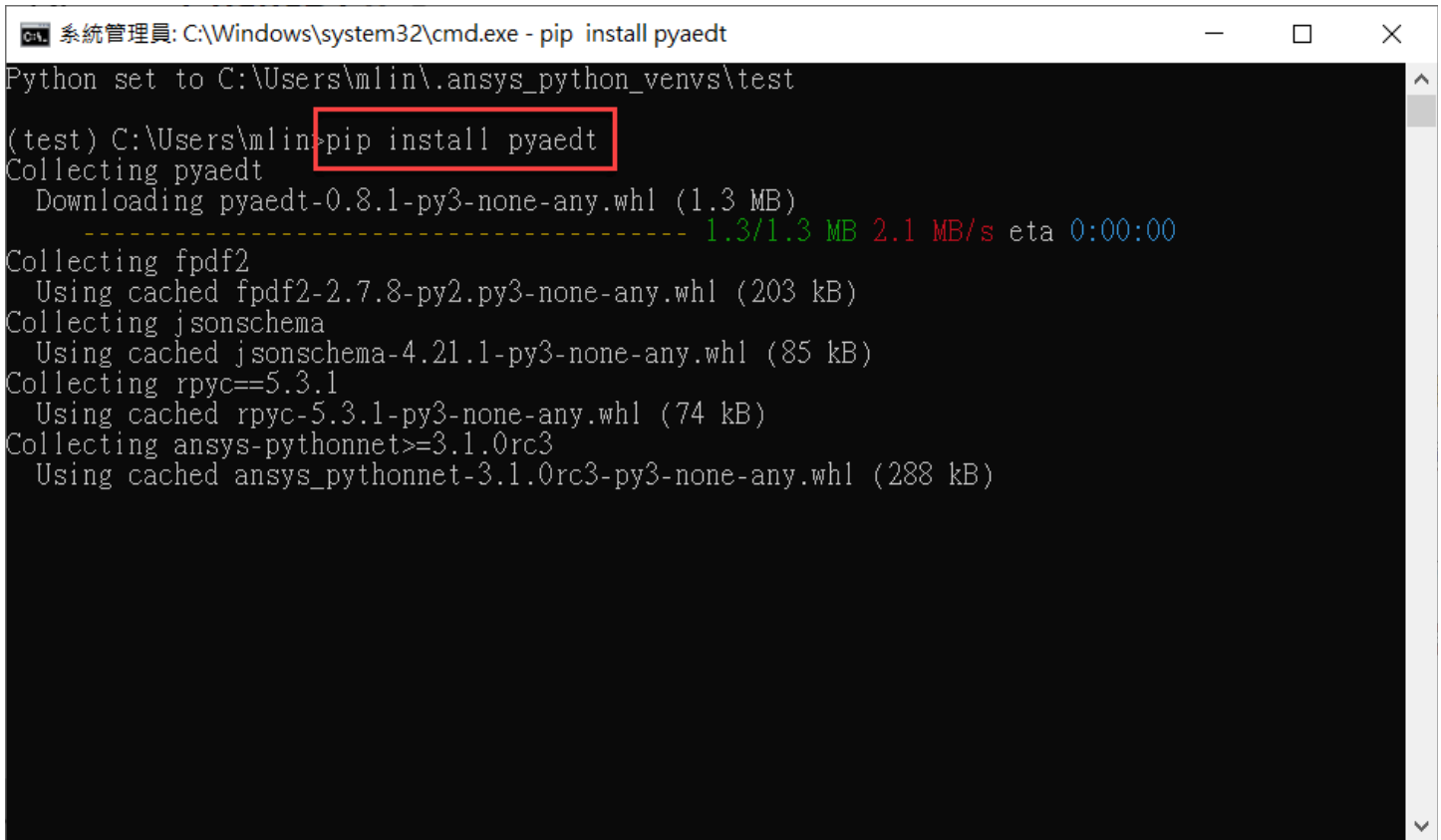
一旦虛擬環境被啟動，您確實可以使用 `pip` 來安裝、升級或卸載該虛擬環境中的Python包。這裡的 `pip` 將僅對當前啟動的虛擬環境有效，任何通過 `pip` 安裝的包都將被安裝在該虛擬環境的 `lib` 目錄中，而不會影響到系統的Python安裝或其他虛擬環境。

這樣的設計允許開發者在不同的項目之間進行隔離管理，每個項目都可以有自己的依賴和環境設置，而不會互相干擾。操作流程大致如下：

1. 開啟命令行界面（`console`）。
2. 導航到您的項目目錄。
3. 啟動虛擬環境：
 - 對於Windows，使用命令 `myenv\Scripts\activate`。
 - 對於Unix或macOS，使用命令 `source myenv/bin/activate`。
4. 使用 `pip` 安裝所需的包：`pip install package_name`。

這樣，您就可以在虛擬環境中自由地管理Python包，而不用擔心影響到系統中的其他Python環境。

以安裝pyaedt為例

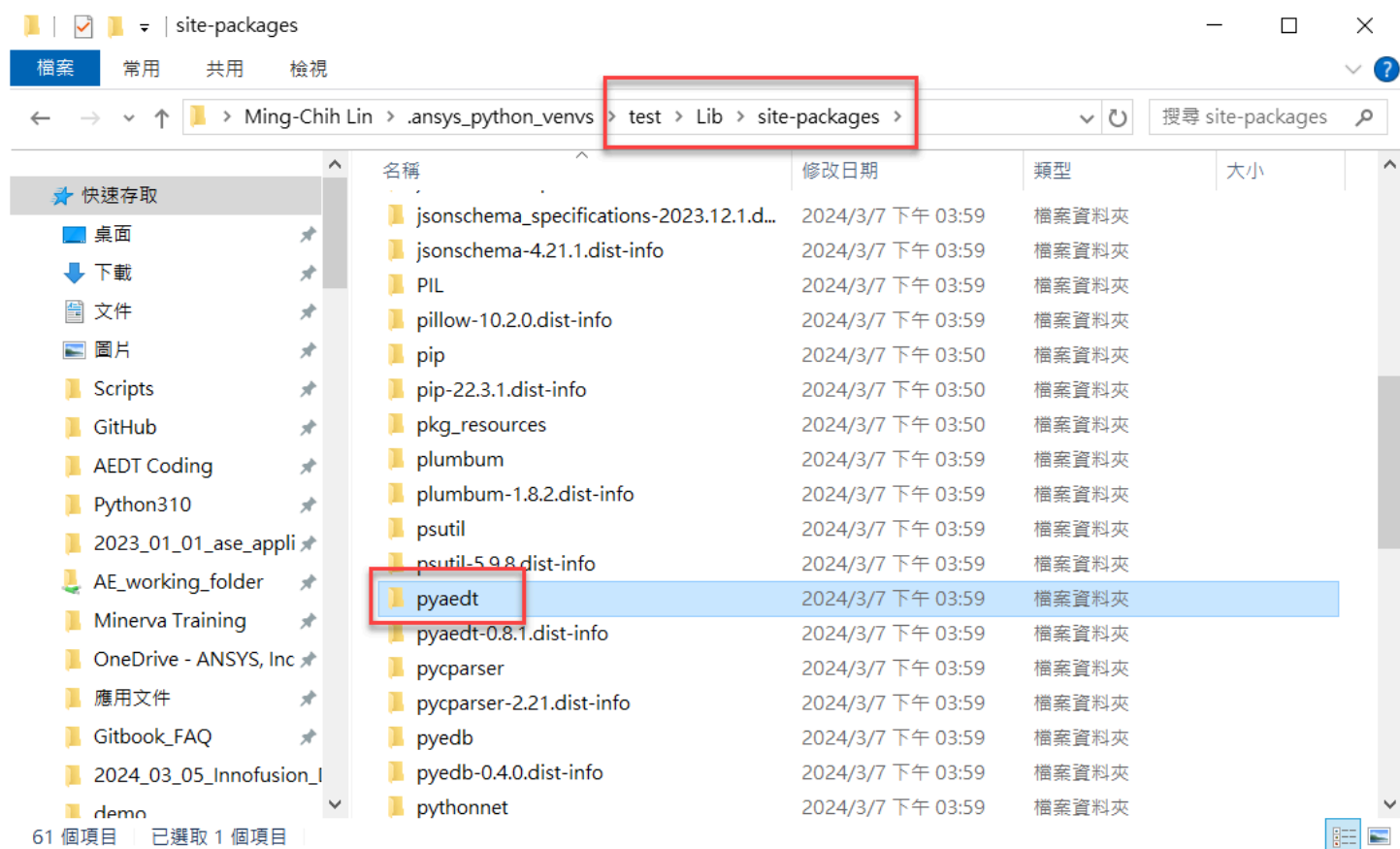


```
系統管理員: C:\Windows\system32\cmd.exe - pip install pyaedt
Python set to C:\Users\mlin\.ansys_python_venvs\test
(test) C:\Users\mlin>pip install pyaedt
Collecting pyaedt
  Downloading pyaedt-0.8.1-py3-none-any.whl (1.3 MB)
----- 1.3/1.3 MB 2.1 MB/s eta 0:00:00
Collecting fpdf2
  Using cached fpdf2-2.7.8-py2.py3-none-any.whl (203 kB)
Collecting jsonschema
  Using cached jsonschema-4.21.1-py3-none-any.whl (85 kB)
Collecting rpyc==5.3.1
  Using cached rpyc-5.3.1-py3-none-any.whl (74 kB)
Collecting ansys-pythonnet>=3.1.0rc3
  Using cached ansys_pythonnet-3.1.0rc3-py3-none-any.whl (288 kB)
```

當您在虛擬環境中使用`pip install pyaedt`命令安裝`pyaedt`後，`pip`會將`pyaedt`及其所有依賴的包安裝到虛擬環境的`lib`目錄下。這是因為大多數Python包都有依賴其他包才能正常工作，`pip`會自動解析這些依賴關係並安裝需要的所有包。

例如，假設您創建了一個名為`myenv`的虛擬環境並啟動它。然後，您執行`pip install pyaedt`。這個動作會在`myenv/lib/pythonX.Y/site-packages/`目錄下（`X.Y`表示Python的版本，例如`3.8`）創建一個`pyaedt`目錄和其他依賴包的目錄。這些目錄中包含了相應包的所有文件，如Python套件、擴展套件、靜態資料文件等。

`site-packages`目錄是Python用來存放第三方包的地方。當您在虛擬環境中導入一個包（例如，使用`import pyaedt`）時，Python解釋器會在`site-packages`目錄下搜索該包。



列出該環境已安裝套件

啟動虛擬環境後，您可以使用 `pip list` 命令來列出當前虛擬環境中安裝的所有套件及其版本。這個命令會生成一個列表，展示了所有通過 `pip` 安裝在該虛擬環境中的包和相應的版本號。

```
系統管理員: C:\Windows\system32\cmd.exe
Python set to C:\Users\mlin\.ansys_python_venvs\test
(test) C:\Users\mlin> pip list
Package            Version
-----
ansys-pythonnet    3.1.0rc3
attrs              23.2.0
cffi               1.16.0
clr-loader         0.2.6
defusedxml         0.7.1
fonttools          4.49.0
fpdf2              2.7.8
jsonschema         4.21.1
jsonschema-specifications 2023.12.1
pillow             10.2.0
pip                22.3.1
plumbum            1.8.2
psutil             5.9.8
pyaedt            0.8.1
pycparser          2.21
pyedb              0.4.0
pytomlpp           1.0.13
pywin32            306
referencing        0.33.0
rpds-py            0.18.0
rpyc               5.3.1
setuptools         65.5.0

[notice] A new release of pip available: 22.3.1 -> 24.0
[notice] To update, run: python.exe -m pip install --upgrade pip
```

ANSYS Python Manager





對於剛開始接觸 Python 開發的工程師來說，透過以上所介紹的命令行界面建立開發環境可能會遇到一些挑戰。ANSYS Python Manager 的出現，正是為了簡化這個過程。以下是 ANSYS Python Manager 如何幫助簡化建立開發環境步驟的一些方式：

1. **簡化 Python 安裝**：用戶可以透過幾個點擊直接在界面上選擇和安裝 Python 的不同版本，而不需要手動下載安裝檔案或設定環境變數。
2. **管理虛擬環境**：ANSYS Python Manager 提供了一個直觀的界面來創建和管理虛擬環境，使用者無需記住和輸入命令行指令。
3. **套件管理**：安裝和管理第三方套件可以通過用戶界面輕鬆完成，這避免了使用命令行工具如 `pip` 的需要。
4. **啟動開發工具**：ANSYS Python Manager 允許從其界面直接啟動多種開發和交互式工具，如 Jupyter Notebook 和 Spyder IDE，這對於習慣使用圖形界面的用戶來說更為直接和方便。

這樣的工具降低了初學者的入門門檻，使得他們能夠更快地開始進行實際的開發工作。

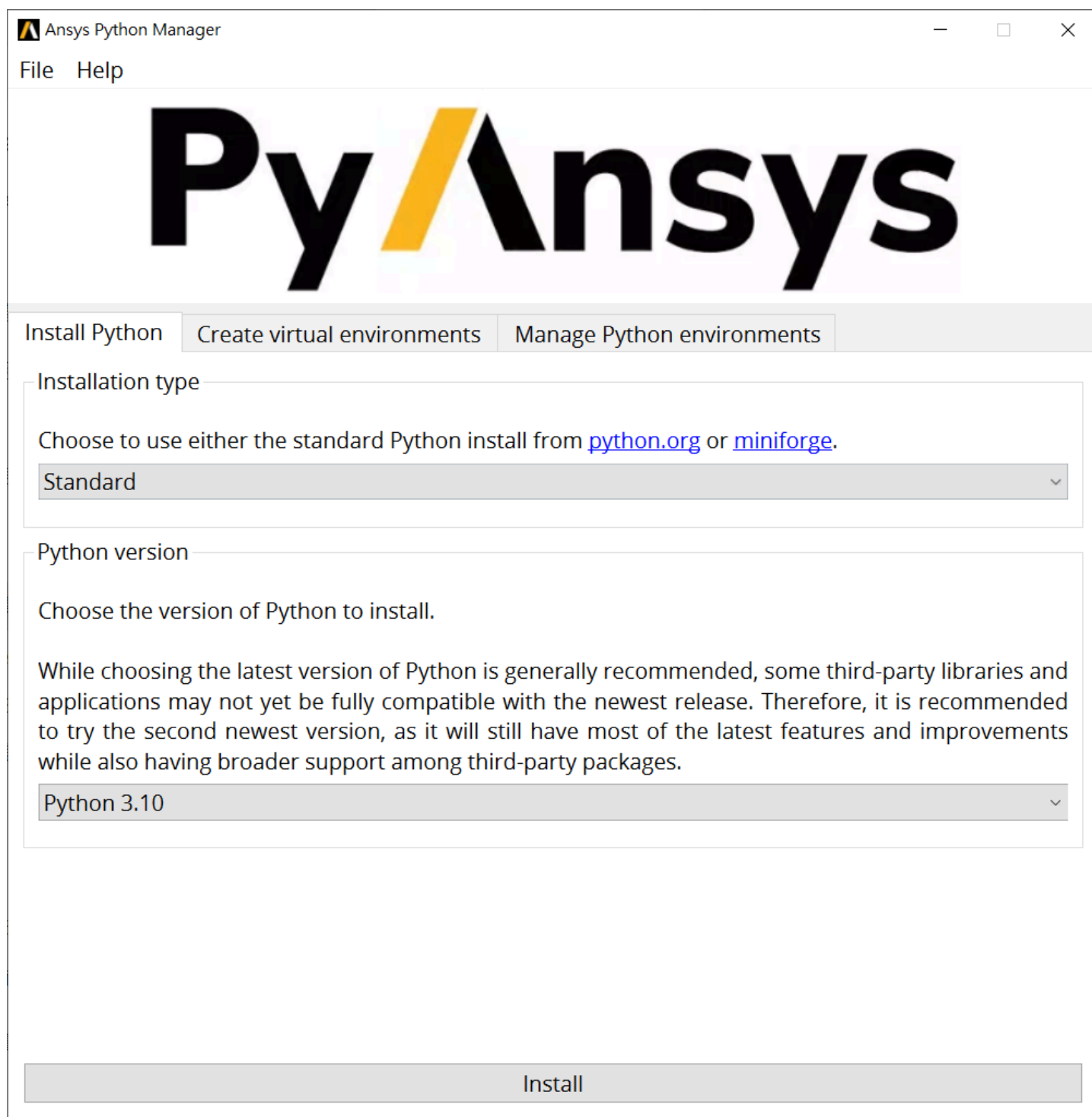
附註

ANSYS Python Manager 下載連結

▼ Assets 4		
 Ansys-Python-Manager-Setup-v0.2.13.exe	49.8 MB	last month
 Unsigned_Ansys-Python-Manager-Setup-v0.2.13.exe	49.8 MB	last month
 Source code (zip)		last month
 Source code (tar.gz)		last month

1. Python 下載及安裝頁面

選擇欲安裝之 Python 版本，按下 Install 鍵便可以自動下載並完成安裝。如果電腦已有安裝所需版本，可以跳過此步驟，選擇虛擬環境建立頁面建立虛擬環境。



2. 虛擬環境建立頁面

選擇Python版本，輸入虛擬環境名稱，按下Create鍵便可以建立一個新的虛擬環境(約10秒鐘)。

File

Help

Py

Ans

sys

Install Python

Create virtual environments

Manage Python environments

Select Python version

Choose the version of Python to use for your virtual environment.

Please select the Python version from the table below to create its respective virtual environment.

Version	Admin	Path
Python 3.10.10	False	C:\Users\mlin\AppData\Local\Programs\Python\Python310\
Python 3.8.7	False	C:\Users\mlin\AppData\Local\Programs\Python\Python38\
Python 3.9.12	False	D:\demo\Python39\

Virtual environment name

Provide the name for your virtual environment.

Virtual environments are created under user directory `.ansys_python_venvs`. If the name provided already exists for another virtual environment, it will not be created. Users will receive a warning informing of the situation.

myvenv

Create

3. 套件安裝頁面

首先選擇欲安裝套件的虛擬環境。裡面有幾個安裝套件的按鍵可以協助安裝套件。

- **Launch console**：啟動一個命令行界面，你可以在這個命令行界面中使用 `pip` 這個指令來手動安裝、升級或移除 Python 套件。
- **Install Python default packages**: 從 PyPI 或其他配置的源自動下載和安裝一系列基礎的科學計算和數據分析套件。如 `scipy`、`pandas` 等等
- **PyAnsys package management**：這個界面允許用戶選擇特定版本的 PyANSYS 套件，然後通過點擊 "Install" 按鈕來安裝。這個過程背後就是執行類

似於 `pip install` 的命令，但是用戶無需手動打開命令行界面來進行安裝，這進一步簡化了套件安裝流程。

