

【Python】venv vs pyenv 比較

venv 與 pyenv 都是 Python 環境管理工具，但功能不同、用途也不同。以下是兩者的詳細比較：

VS venv vs pyenv 比較表

項目	venv	pyenv
□ 功能	建立虛擬環境 (isolated packages)	管理多個 Python 版本 (例如 3.6, 3.8, 3.11)
□ 主要用途	為每個專案建立獨立依賴環境	切換/安裝不同版本的 Python
□ 安裝方式	Python 內建 (3.3+)	需額外安裝 (使用 Git clone 或安裝腳本)
□ 使用方式	<code>python -m venv myenv</code>	<code>pyenv install 3.11.3</code> + <code>pyenv global/local</code>
□ 環境位置	專案內產生 <code>myenv/</code> 資料夾	全局安裝於 <code>~/.pyenv/versions/</code>
□ 是否支援套件分離	□ 支援 (每個虛擬環境獨立)	□ 不直接支援 (但可配合 <code>pyenv-virtualenv</code>)
□ 是否支援版本共存	□ 不支援 (依系統 Python)	□ 支援多版本並快速切換
□ 可搭配使用	N/A	□ 可搭配 <code>pyenv-virtualenv</code> 使用 <code>venv</code> 功能

□ 何時該用哪個？

使用 venv：

- 你只需在同一台機器上建立**隔離的 Python 套件環境**
- 你已安裝好適合的 Python 版本，不需要版本切換

使用 pyenv：

- 你需要安裝或切換不同版本的 Python (例如某專案需 3.6，另一個用 3.11)
- 系統預設 Python 無法修改 (尤其在 macOS/Linux)
- 想搭配 `pyenv-virtualenv` 做進階管理

□ 進階：兩者可以一起用

你可以使用 `venv` 結合安裝多個 Python 版本的方式，讓不同專案使用不同版本的 Python。以下是完整的步驟：

□ 步驟 1：安裝多個 Python 版本

macOS/Linux 建議使用 `pyenv`

```
brew install pyenv      # macOS 安裝 pyenv
pyenv install 3.8.18    # 安裝 Python 3.8.18
pyenv install 3.11.9    # 安裝 Python 3.11.9
```

Windows 使用 [Python 官網](#) 安裝多版本，並記住安裝路徑。

□ 步驟 2：為專案建立虛擬環境並指定 Python 版本

使用 `pyenv` 時

```
pyenv shell 3.8.18
python -m venv venv38      # 使用 3.8.18 建立虛擬環境
```

使用絕對路徑（不使用 `pyenv`）

```
# 假設你已安裝 Python 3.11 在 /usr/local/bin/python3.11
/usr/local/bin/python3.11 -m venv venv311
```

□ 步驟 3：啟用虛擬環境

```
source venv38/bin/activate  # macOS/Linux
venv38\Scripts\activate.bat # Windows CMD
```

啟用後，你執行 `python` 將會使用虛擬環境中的 Python 版本。

□ 步驟 4：確認 Python 版本

```
python --version
```

以下是將 **多 Python 版本虛擬環境與開發工具整合** 的常見場景與操作：

□ 1. 在 VS Code 中使用不同版本虛擬環境

□ 步驟：

1. 先建立虛擬環境（如前述）
2. 開啟 VS Code 該專案資料夾
3. 左下角點選 Python 解譯器版本
4. 選擇你剛建立的虛擬環境（例如 `.venv/bin/python` 或 `venv/Scripts/python.exe`）

□ 小技巧：

你也可以在專案內加一個 `.vscode/settings.json`：

```
{
  "python.pythonPath": ".venv/bin/python"
}
```

□ 2. `pipx` 搭配多 Python 版本

`pipx` 可以讓你用不同版本 Python 安裝 CLI 工具，保持環境乾淨。

```
pipx install some-tool --python python3.11
```

□ 3. `poetry` 管理虛擬環境與依賴（推薦進階用戶）

```
poetry env use /usr/local/bin/python3.11
poetry install
```

它會自動為專案建立 `.venv` 並鎖定依賴版本。

□ 4. 為不同目錄自動切換 Python 版本（pyenv local）

```
cd my-project38
pyenv local 3.8.18

cd ../my-project311
pyenv local 3.11.9
```

`pyenv` 會在目錄中寫入 `.python-version`，自動切換。

如果你告訴我你的作業系統（macOS/Linux/Windows）與偏好（如 VS Code/CLI），我可以給你量身打造的設定方式。要嗎？

🕒 修訂版本 #4

★ 由 treeman 建立於 24     2025 22:32:08

✎ 由 treeman 更新於 24     2025 23:38:53