Python 的 50+ 練習：資料科學學習手冊

> *使用 conda 管理環境*

數據交點 | 郭耀仁 yaojenkuo@datainpoint.com

關於環境管理

關於 Python 程式語言的描述

- 直譯式。
- 動態類別。
- 物件導向。
- **泛用**。
- ...等。

Python 是廣泛用途的程式語言

廣泛用途（General-purposed）是指作者 Guido van Rossum 在開發 Python 的過程中並沒有針對特定的應用場景去設計語言特性，而是交由程式語言的「使用者」利用第三方模組自行發展。而因為 Python 有大量來自不同領域的使用者，也使得其應用場景非常多元而且發展蓬勃。

- 資料科學。
- 網頁應用程式開發。
- 桌面軟體使用者介面開發。
- 遊戲開發。
- ...等。

來源：https://www.python.org/about/apps/

什麼是環境管理

當程式語言的應用場景多元，伴隨而來的就是要依據專案需求安裝設定不同的 Python 版本以及不同的模組版本。在一台電腦管理軟體版本，確保各個專案的開發環境不會受到彼此影響的機制，就稱為環境管理。

- 專案一需要 Python 3.9 與 NumPy/Pandas 模組進行資料科學。
- 專案二需要 Python 3.8 與 Flask/Django 模組進行網頁應用程式開發。
- 專案三需要 Python 3.7 與 PyQt/Kivy 模組進行桌面軟體使用者介面開發。
- ...等。

環境管理可以理解為管理軟體版本

- Python 版本。
- 模組版本。

掌握環境管理是從 Python 初學者成長為進階者的重要里程碑。

標準的 Python 版本與模組管理工具

- Python 版本管理：標準模組 `venv`：https://docs.python.org/3/tutorial/venv.html
- 模組版本管理：第三方模組 `pip`：https://pip.pypa.io/en/stable，雖然 `pip` 是第三方模組，但因為太受歡迎，自從 Python 3.4 以後已經變得像標準模組一樣會伴隨 Python 直譯器而安裝。

conda 可以同時管理 Python 版本與模組版本

> *conda 是一個開源的跨平台、跨程式語言軟體，能夠在 Windows、macOS 與 Linux 上安裝運行，它被設計作為 Python、R、Scala 與 Java 等任何程式語言的模組、依賴性以及工作環境管理軟體。*

來源：https://conda.io/projects/conda/en/latest/index.html

conda 入門

conda 以終端機作為執行介面

- 課程的 JupyterLab 已經安裝好 conda。
- 點選上方選單的 File -> New -> Terminal 開啟終端機。
- 檢視 conda 的版本來確定已經安裝妥當。

```
conda --version
#conda 4.9.2
```

檢視環境清單

- 顯示所有可以使用的環境名稱與安裝路徑。
- 以 `*` 標註目前所在的環境。

```
conda env list
# conda environments:
#
#base                     /srv/conda
#notebook              *  /srv/conda/envs/notebook
```

檢視目前所在環境的 Python 版本與模組版本

- 環境名稱：`notebook`
- Python 版本：3.9.7

```
python --version
conda list
#Python 3.9.7
# packages in environment at /srv/conda/envs/notebook:
#
# Name                    Version                   Build  Channel
#...
```

建立環境

- 環境名稱：`python3812`
- Python 版本：3.8.12

```
conda create --name python3812 python=3.8.12
# ...
#Proceed ([y]/n)? y
# ...
```

啟動環境

- 環境名稱：`python3812`
- 啟動之後會終端機會出現環境名稱提示 `(python3812)`

```
conda activate python3812
```

檢視目前所在環境的 Python 版本

- 環境名稱：`python3812`
- Python 版本：3.8.12

```
(python3812) python --version
#Python 3.8.12
```

在目前環境安裝模組

- 環境名稱：`python3812`
- 模組名稱：`ipykernel` 用來管理 Jupyter Notebook 的運算核心（Kernels）。

```
(python3812) conda install ipykernel
# ...
#Proceed ([y]/n)? y
# ...
```

新增 Jupyter Notebook 的運算核心

- 環境名稱：`python3812`
- 運算核心顯示名稱：`Python 3.8.12`

```
(python3812) python -m ipykernel install --user --name python3812 --display-name "Python 3.8.12"
#Installed kernelspec python3812 in /home/jovyan/.local/share/jupyter/kernels/python3812
```

檢視 Jupyter Notebook 的運算核心清單

```
(python3812) jupyter kernelspec list
#Available kernels:
#  python3812    /home/jovyan/.local/share/jupyter/kernels/python3812
#  python3       /srv/conda/envs/python3812/share/jupyter/kernels/python3
```

新增一個運算核心為 `Python 3.8.12` 的筆記本

- 稍等一下再重新開啟 Launcher
- 點選 Notebook Python 3.8.12
- 透過標準模組 `sys` 在 Jupyter Notebook 中檢視 Python 版本。

```
# Run in Jupyter Notebook instead of Terminal
import sys

print(sys.version)
#3.8.12 | packaged by conda-forge | (default, Oct 12 2021, 21:57:06)
#[GCC 9.4.0]
```

卸載環境

回到 `notebook` 環境。

```
(python3812) conda deactivate
(notebook) conda env list
# conda environments:
#
#base                     /srv/conda
#notebook              *  /srv/conda/envs/notebook
#python3812               /srv/conda/envs/python3812
```

卸載環境（續）

回到 `base` 環境。

```
(notebook) conda deactivate
conda env list
# conda environments:
#
#base                  *  /srv/conda
#notebook                 /srv/conda/envs/notebook
#python3812               /srv/conda/envs/python3812
```

移除 Jupyter Notebook 運算核心

- 移除 Jupyter Notebook 運算核心：`python3812`
- 檢視 Jupyter Notebook 的運算核心清單。

```
jupyter kernelspec remove python3812
jupyter kernelspec list
#Kernel specs to remove:
#  python3812            /home/jovyan/.local/share/jupyter/kernels/python3812
#Remove 1 kernel specs [y/N]: y
#[RemoveKernelSpec] Removed /home/jovyan/.local/share/jupyter/kernels/python3812
#Available kernels:
#  python3    /srv/conda/envs/notebook/share/jupyter/kernels/python3
```

移除環境

- 移除環境 `python3812`
- 檢視環境清單。

```
conda remove --name python3812 --all
conda env list
#Remove all packages in environment /srv/conda/envs/python3812:
#
#...
#
#Proceed ([y]/n)? y
#
#...
# conda environments:
#
#base                  *  /srv/conda
#notebook                 /srv/conda/envs/notebook
```

安裝 Miniconda

- 安裝 Miniconda
- Miniconda 是 Anaconda 輕量的版本，基本只會在電腦中安裝特定 Python 直譯器版本與 conda
- Python 初學者會建議安裝 Anaconda，但現在我們已經是 Python 進階者，所以建議安裝 Miniconda

在 Windows 作業系統使用 conda 建立 Python 資料科學環境

Windows 作業系統下載與安裝 Miniconda

點此下載安裝 .exe 安裝檔

conda 以終端機作為執行介面

- **Windows 作業系統：Anaconda Prompt**
- macOS：Terminal（終端機）。

檢視目前所在環境的 Python 版本

環境名稱：`base`

```
(base) python --version
```

在 base 環境安裝 jupyterlab

環境名稱：`base`

```
(base) conda install jupyterlab
# ...
#Proceed ([y]/n)? y
# ...
```

建立環境

- 環境名稱：`pythonfiftyplus`
- Python 版本：3.9.7

```
(base) conda create --name pythonfiftyplus python=3.9.7
# ...
#Proceed ([y]/n)? y
# ...
```

啟動環境

- 環境名稱：`pythonfiftyplus`
- 啟動之後會終端機會出現環境名稱提示 `(pythonfiftyplus)`

```
(base) conda activate pythonfiftyplus
```

在目前環境安裝資料科學模組

環境名稱：`pythonfiftyplus`

```
(pythonfiftyplus) pip install ipykernel numpy==1.21 pandas==1.3 xlrd==2.0 openpyxl==3.0
matplotlib==3.5 scikit-learn==1.0
# ...
#Proceed ([y]/n)? y
# ...
```

新增 Jupyter Notebook 的運算核心

- 環境名稱：`pythonfiftyplus`
- 運算核心顯示名稱：`Python Fifty Plus`

```
(pythonfiftyplus) python -m ipykernel install --user --name pythonfiftyplus --display-name "Python Fifty Plus"
```

新增一個運算核心為 `Python Fifty Plus` 的筆記本

- 在 Anaconda Prompt 輸入 `conda deactivate` 回到 base 環境。
- 在 Anaconda Prompt 輸入 `jupyter lab` 後按下 Enter
- 點選 Notebook Python Fifty Plus
- 透過標準模組 `sys` 在 Jupyter Notebook 中檢視 Python 版本。

```
# Run in Jupyter Notebook instead of Terminal
import sys

print(sys.version)
#3.9.7 | packaged by conda-forge | (default, Sep 29 2021, 19:20:46)
#[GCC 9.4.0]
```

在 macOS 使用 conda 建立 Python 資料科學環境

macOS 下載與安裝 Miniconda

點此下載安裝 .pkg 安裝檔

conda 以終端機作為執行介面

- Windows 作業系統：Anaconda Prompt
- **macOS**：**Terminal（終端機）**。

檢視目前所在環境的 Python 版本

環境名稱：`base`

```
(base) python --version
```

在 base 環境安裝 jupyterlab

環境名稱：`base`

```
(base) conda install jupyterlab
# ...
#Proceed ([y]/n)? y
# ...
```

建立環境

- 環境名稱：`pythonfiftyplus`
- Python 版本：3.9.7

```
(base) conda create --name pythonfiftyplus python=3.9.7
# ...
#Proceed ([y]/n)? y
# ...
```

啟動環境

- 環境名稱：`pythonfiftyplus`
- 啟動之後會終端機會出現環境名稱提示 `(pythonfiftyplus)`

```
(base) conda activate pythonfiftyplus
```

在目前環境安裝資料科學模組

環境名稱：`pythonfiftyplus`

```
(pythonfiftyplus) pip install ipykernel numpy==1.21 pandas==1.3 xlrd==2.0 openpyxl==3.0
matplotlib==3.5 scikit-learn==1.0
# ...
#Proceed ([y]/n)? y
# ...
```

新增 Jupyter Notebook 的運算核心

- 環境名稱：`pythonfiftyplus`
- 運算核心顯示名稱：`Python Fifty Plus`

```
(pythonfiftyplus) python -m ipykernel install --user --name pythonfiftyplus --display-name "Python Fifty Plus"
```

新增一個運算核心為 `Python Fifty Plus` 的筆記本

- 在終端機輸入 `conda deactivate` 回到 base 環境。
- 在終端機輸入 `jupyter lab` 後按下 Enter
- 點選 Notebook Python Fifty Plus
- 透過標準模組 `sys` 在 Jupyter Notebook 中檢視 Python 版本。

```
# Run in Jupyter Notebook instead of Terminal
import sys

print(sys.version)
#3.9.7 | packaged by conda-forge | (default, Sep 29 2021, 19:20:46)
#[GCC 9.4.0]
```

重點統整

- 掌握環境管理是從 Python 初學者成長為進階者的重要里程碑。
- conda 可以同時管理 Python 版本與模組版本。
- conda 以終端機作為執行介面。

重點統整（續）

- 以 `conda env list` 檢視環境清單。
- 以 `conda create --name ENVNAME python=MAJOR.MINOR.PATCH` 建立環境。
- 以 `conda install MODULENAME` 安裝模組。

重點統整（續）

- 以 `conda activate ENVNAME` 啟動環境。
- 以 `conda deactivate` 卸載環境。
- 卸載環境後以 `conda remove --name ENVNAME --all` 移除環境。