

使用 conda 建立虛擬環境

# 建立虛擬環境

- 為什麼需要虛擬環境
  - 為了避免 Python 套件相衝
  - 安裝套件不需要透過 SA ( 避免 `sudo pip install ...` )
- 可使用的套件管理工具
  - conda (安裝 Anaconda/Miniconda) (推薦)
  - pipenv (推薦)
  - venv
  - virtualenv
  - pyenv
  - ...

## 在 Terminal 下建立虛擬環境 ( 以 Miniconda 為例 )

### A. 將環境放置在預設路徑 `~/.conda/envs`

- 語法：

```
$ conda create -n <環境名稱> python=<版本號>
```

- Example:

```
$ conda create -n py37keras python=3
```

### B. 將環境放置在指定路徑

- 語法：

```
$ conda create -p <路徑> python=<版本號>
```

- Example:

```
$ conda create -p ~/Environments/py37keras python=3.7
```

# 啟動/退出虛擬環境 (Linux)

- 啟動虛擬環境

- 語法：

- ```
$ conda activate <環境名稱/路徑>
```

- Example:

- ```
$ conda activate py37keras
```

- ```
$ conda activate ~/Environments/py37keras
```

- 退出虛擬環境

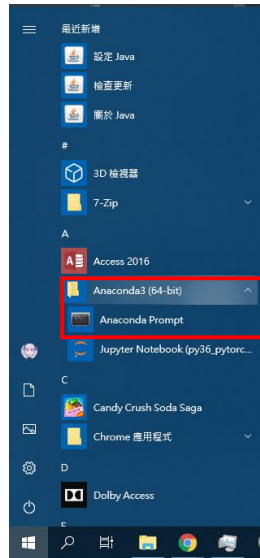
- ```
$ conda deactivate
```

# 啟動/退出虛擬環境 (Windows)

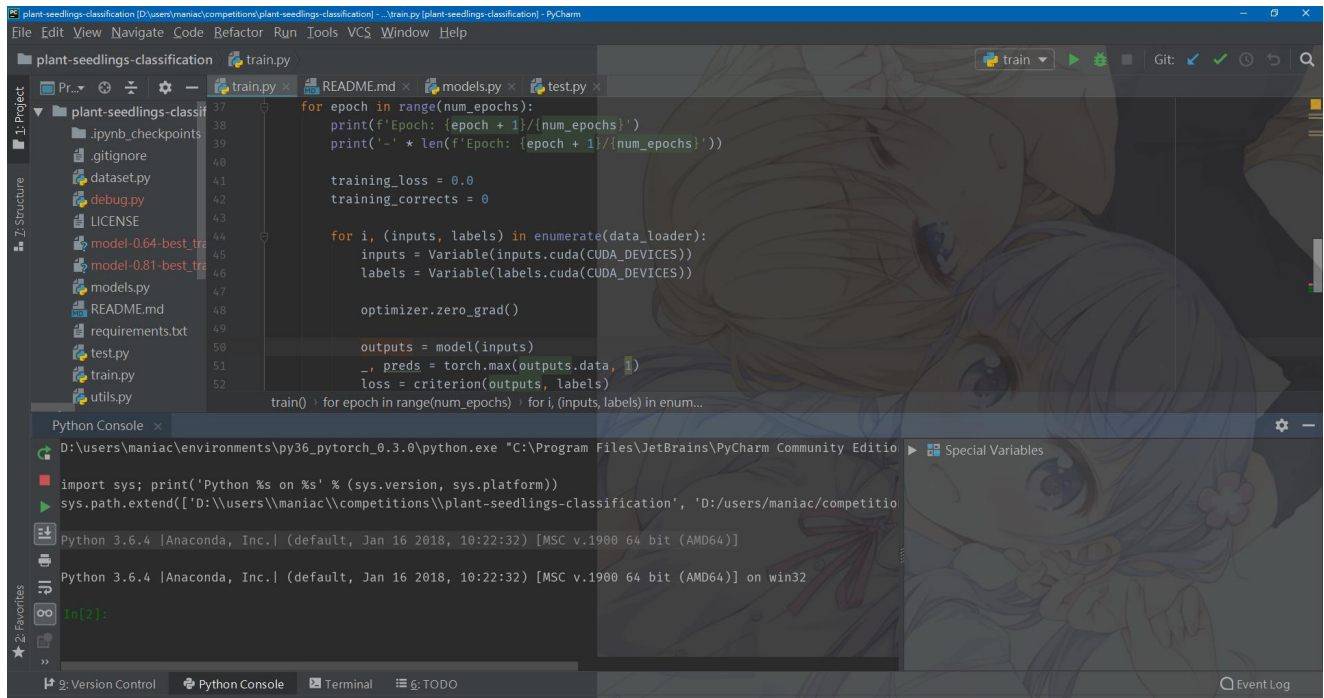
## 1. 先找出 Anaconda Prompt

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Anaconda3

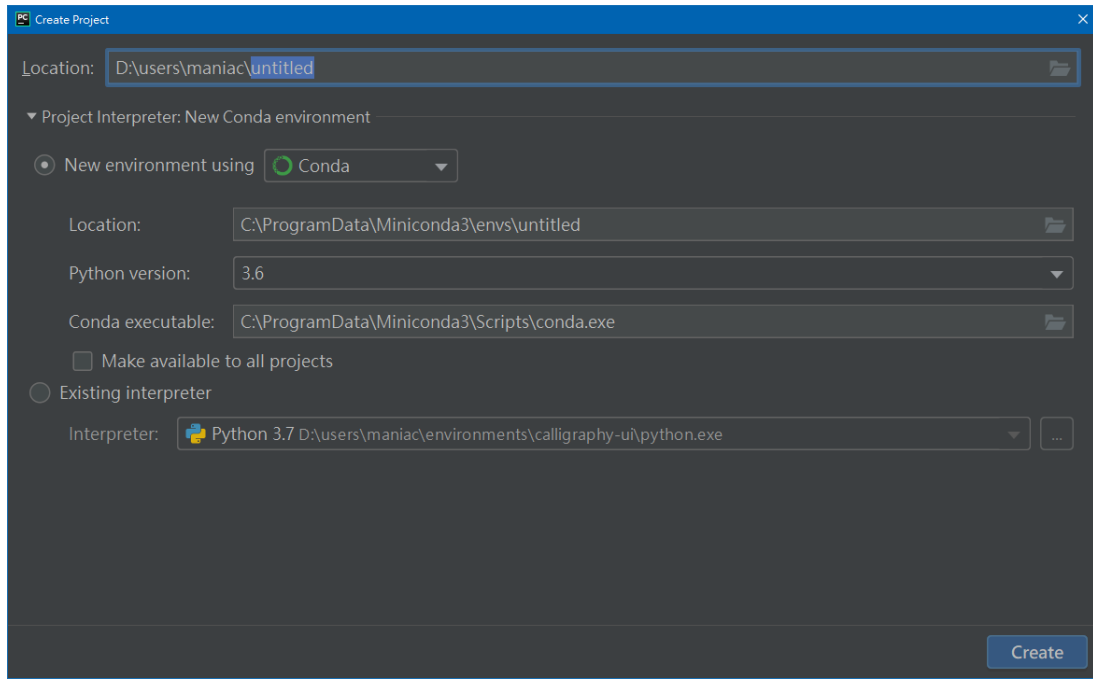
## 2. 語法和 Linux 幾乎相同



# PyCharm 介面

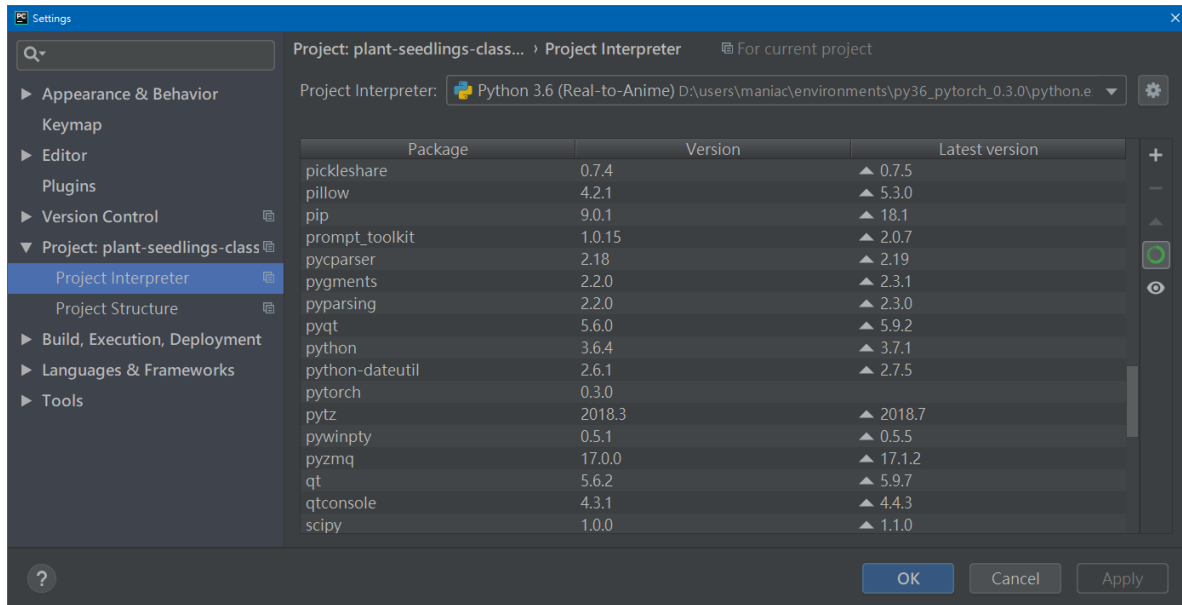


# 在 PyCharm 下建立/啟動虛擬環境



# 在 PyCharm 下修改虛擬環境

- File->Settings->Project->Project Interpreter





# 深度學習伺服器使用教學

# 深度學習伺服器使用教學

- 伺服器概況
- 設定環境變數
- 協助
- 申請帳號
- 限制 GPU 使用量的設定方法

# 伺服器概況

- 總共三台
  - IP address:
    - 140.123.97.173:2222 (Ubuntu 16.04)
    - 140.123.97.173:2223 (Ubuntu 16.04)
    - 140.123.97.173:2224 (Ubuntu 14.04)
- 功能：深度學習實驗
- 已安裝：
  - Miniconda (conda version: 4.4.10)
    - 路徑：/opt/anaconda/miniconda3
  - CUDA 8.0 (default)
  - CUDA 9.0

# 設定環境變數

- 預設使用 Miniconda

- 開啟 ~/.bashrc 加入以下：

- ```
export PATH=/opt/anaconda/miniconda3/bin:$PATH
```

- 使用 CUDA 9.0

- 開啟 ~/.bashrc 加入以下：

- ```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda-9.0/lib64
```

- 使用 CUDA 8.0

- 開啟 ~/.bashrc 加入以下：

- ```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda-8.0/lib64
```

# 協助

- 如果需要安裝軟體
  - 作業系統相關套件 ( e.g. 更新 CUDA/cuDNN 版本 )
    - 請來信洽談
  - Python 套件
    - 請自己用虛擬環境安裝，SA 不負責安裝任何 Python 套件
    - 若是 Python 套件相依某個軟體 ( e.g. OpenCV <- ffmpeg )，請來信討論
- 如果發生問題
  - 把問題清楚描述後寄信給 SA
  - 聯絡方式
    - SA 聯絡信箱：[remorse@csie.io](mailto:remorse@csie.io)

# 申請帳號

- 申請方式
  - Google 表單 <https://goo.gl/forms/O0SLOW4Q7juJ8KLF3>
    - 請填寫申請人 email、姓名、學號、指導教授、帳號名稱
  - 審核通過後寄出帳號密碼
- 帳號規定
  - 研究所
    - 每間實驗室可申請三個永久帳號
  - 大學部工學院
    - 每位同學可申請一個帳號，帳號期限為一年
  - 非工學院
    - 接受因課程需要的申請原因（需在表單內註明是哪一堂課）
    - 每位同學可申請一個帳號，帳號期限至上課結束

# 如何避免過度使用 GPU

- 查看 GPU 使用率
- 指定使用的顯示卡
  - 因為 Tensorflow 預設使用伺服器上的所有 GPU

# 查看 GPU 使用率

輸入指令：

```
$ nvidia-smi
```

輸出結果：

|                                                  |           |               |               |                        |        |          |             |
|--------------------------------------------------|-----------|---------------|---------------|------------------------|--------|----------|-------------|
| NVIDIA-SMI 384.81                                |           |               |               | Driver Version: 384.81 |        |          |             |
|                                                  |           |               |               |                        |        |          |             |
| GPU                                              | Name      | Persistence-M |               | Bus-Id                 | Disp.A | Volatile | Uncorr. ECC |
| Fan                                              | Temp      | Perf          | Pwr:Usage/Cap | Memory-Usage           |        | GPU-Util | Compute M.  |
| =====+=====+=====                                |           |               |               |                        |        |          |             |
| 0                                                | Tesla K80 | On            |               | 00000000:05:00.0       | Off    | 0        |             |
| N/A                                              | 73C       | P0            | 62W / 100W    | 116MiB / 11439MiB      |        | 0%       | Default     |
| +-----+-----+-----+-----+-----+-----+-----+----- |           |               |               |                        |        |          |             |
| 1                                                | Tesla K80 | On            |               | 00000000:06:00.0       | Off    | 0        |             |
| N/A                                              | 52C       | P0            | 98W / 100W    | 10961MiB / 11439MiB    |        | 82%      | Default     |
| +-----+-----+-----+-----+-----+-----+-----+----- |           |               |               |                        |        |          |             |



# GPU 概況

- 顯示卡型號：NVIDIA Tesla K80
- 每台機器插兩張顯示卡（ 每台 4 個 GPU，總共 12 個 GPU ）
- 每個 GPU 有 ~12 GB Memory
- 如果你直接執行程式，很可能會一次吃光所有 **GPU Memory**
- 因此規定
  1. 每人限用 1 GPU
  2. 超出使用量會被刪除 process

# 指定使用的顯示卡

1. 先用 `nvidia-smi` 看 GPU 使用狀況
  2. 挑一張還有 GPU memory 的顯示卡
  3. 設定環境變數 `CUDA_VISIBLE_DEVICES` ( 0, 1, 2, 3 )
- 如果要用第 4 張，設定方式如下 ( 選一種方式 ) :
    1. `$ CUDA_VISIBLE_DEVICES=3 python my_script.py`
    2. Python 程式碼內加入

```
import os
os.environ["CUDA_VISIBLE_DEVICES"] = "3"
```

# 使用建議

- 在自己的電腦安裝 CPU 版本的 Tensorflow+Keras
- 確認可以執行以後上傳到伺服器執行
- 在伺服器安裝 tensorflow-gpu 務必確認可以使用 GPU
  - 如有問題可以寄信問 SA
- 如果使用 GPU 時出現 out of memory
  - 調整 batch size 設小一點