

Python安裝與環境建置

郭忠義

jykuo@ntut.edu.tw

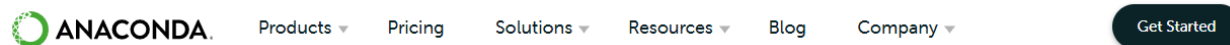
臺北科技大學資訊工程系

安裝 Anaconda

❑ 安裝 Anaconda 套件包

<https://www.anaconda.com/products/individual/>

- 包含眾多科學、數學、工程、資料分析的 Python 套件
- 內建 spyder 編譯器



Individual Edition

Windows 

Python 3.8

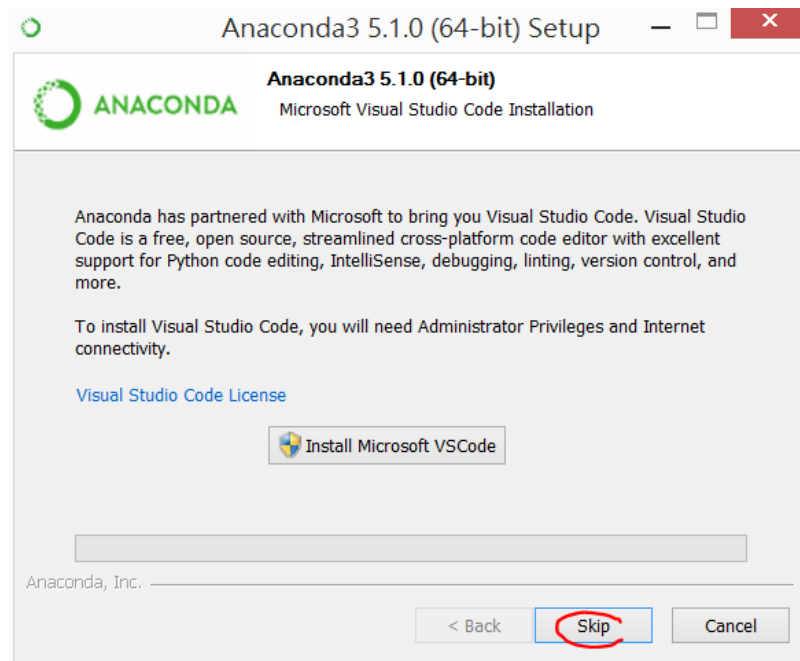
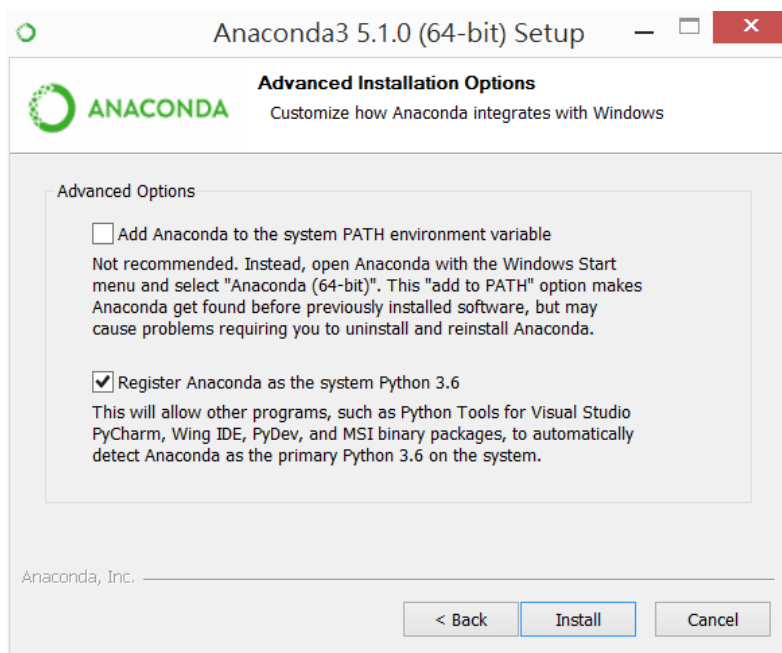
64-Bit Graphical Installer (466 MB)

32-Bit Graphical Installer (397 MB)

安裝 Anaconda

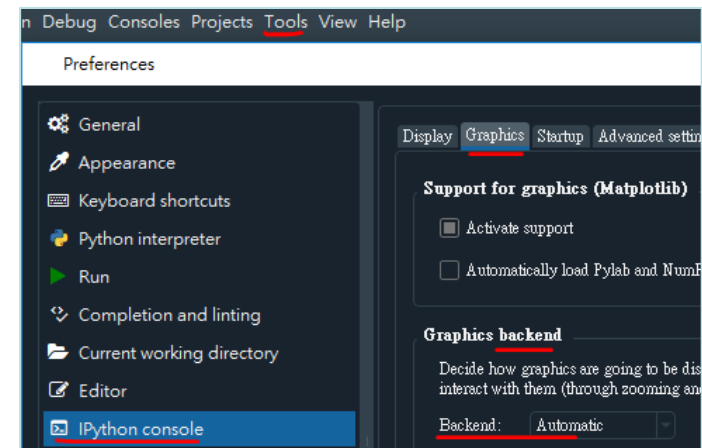
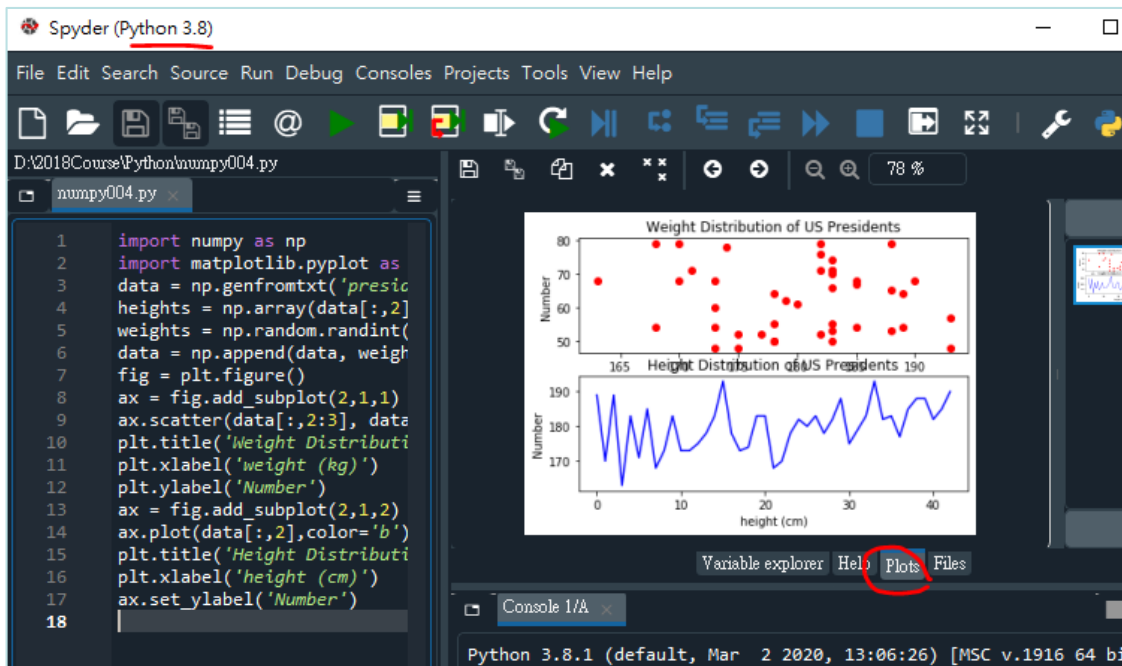
❑ 執行 Anaconda3-xxxx_64.exe

○ 設定路徑， C:\ProgramData\Anaconda3



使用Spyder撰寫

- ❑ Jupyter Notebook: 編輯執行器
- ❑ Spyder: IDE整合編輯環境
- ❑ matplotlib.pyplot要能畫圖，選 Plots
 - Tools > Preferences > iPython console > Graphics > Graphics backend > Automatic



安裝 tensorflow

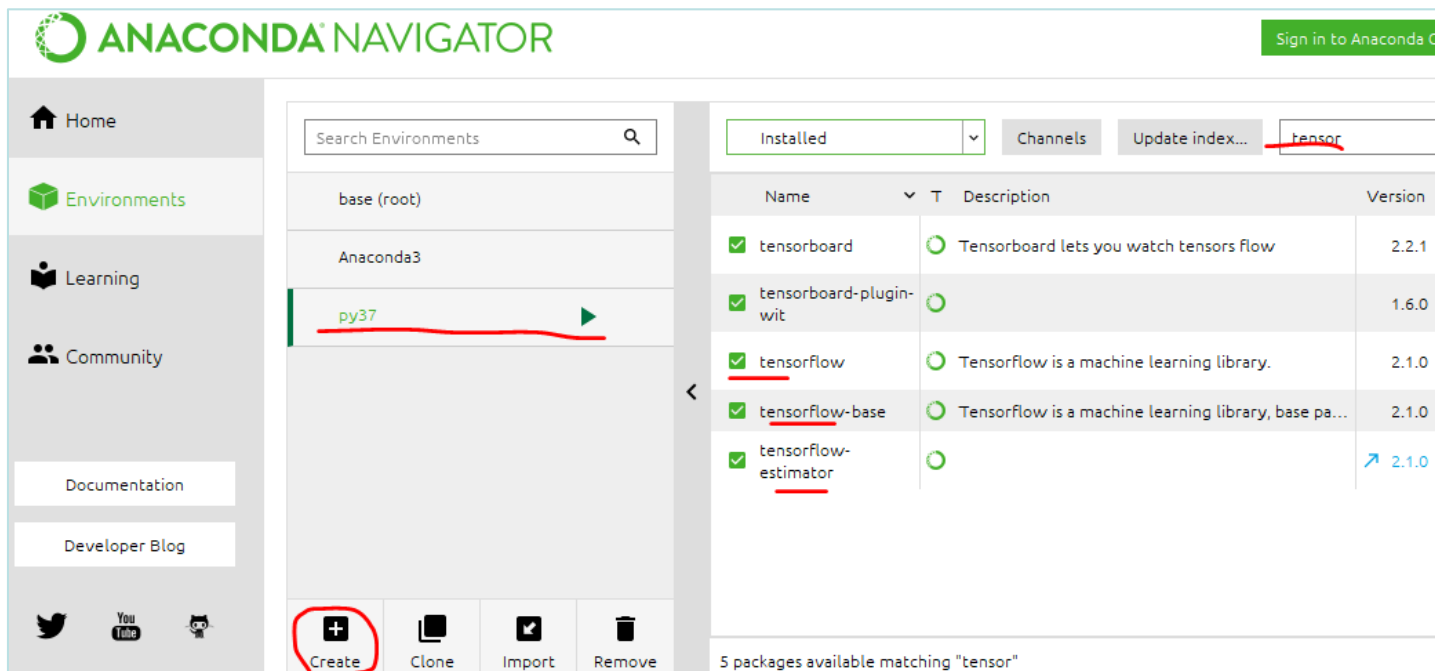
❑ 開始: Anaconda Prompt 執行 (安裝 python 3.7環境)

- `conda create -n py37 anaconda=2020.02 python=3.7`

❑ UI介面安裝套件 Anaconda Navigator

- Create 3.7版 (建議使用Prompt會產生捷徑)

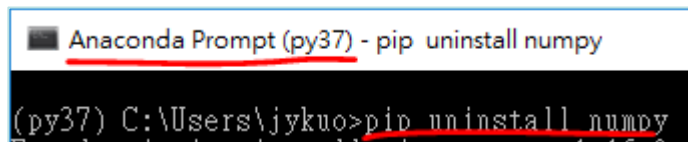
- (Environment – Not installed): tensor - Apply



安裝 tensorflow

❑ 先移除 numpy :

- pip uninstall numpy
- (pip3 uninstall numpy) (只有 python 3 , pip pip3沒分別)



```
Anaconda Prompt (py37) - pip uninstall numpy  
(py37) C:\Users\jykuo>pip uninstall numpy
```

❑ 再安裝 1.17 以下

- pip install numpy==1.16.0
- (pip3 install numpy==1.16.0)

```
#測試是否安裝 tensorflow  
import tensorflow as tf  
import numpy as np  
print(tf.__version__)  
print(np.__version__)  
hello = tf.constant('hello tensorflow!')  
sess = tf.compat.v1.Session()  
print(sess.run(hello))  
print(sess.run(hello).decode()) #解成 unicode 字串
```

```
1.14.0  
1.16.0  
b'hello tensorflow!'  
hello tensorflow!
```

Prompt 安裝套件

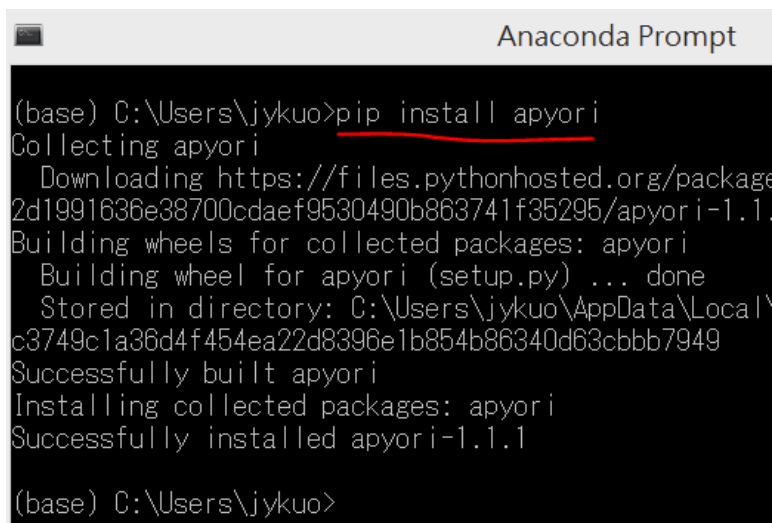
□ Anaconda Prompt

○ 指令執行安裝套件

- conda install "套件名稱"
- conda update "套件名稱"
- conda update -all

○ 若為非anaconda repository package

- 安裝 association rule 套件
- pip install apyori



```
Anaconda Prompt
(base) C:\Users\jykuo>pip install apyori
Collecting apyori
  Downloading https://files.pythonhosted.org/packages/2d/19/16/36e38700cdaef9530490b863741f35295/apyori-1.1.1-py2.py3-none-any.whl
Building wheels for collected packages: apyori
  Building wheel for apyori (setup.py) ... done
  Stored in directory: C:\Users\jykuo\AppData\Local\Temp\pip-build-env-3749c1a36d4f454ea22d8396e1b854b86340d63cbbb7949
Successfully built apyori
Installing collected packages: apyori
Successfully installed apyori-1.1.1

(base) C:\Users\jykuo>
```

Prompt 安裝 PyGame 套件

□ 安裝 pygame

- 1. 開啟 anaconda prompt，檢視 python 版本。
- 2. 下載相對應版本 pygame 安裝包，儲存到 prompt 目錄。網址：<https://www.lfd.uci.edu/~gohlke/pythonlibs/#pygame>
- 3. 安裝指令：`python -m pip install --user pygame-1.9.4-cp37-cp37m-win_amd64.whl`

使用Notepad++撰寫

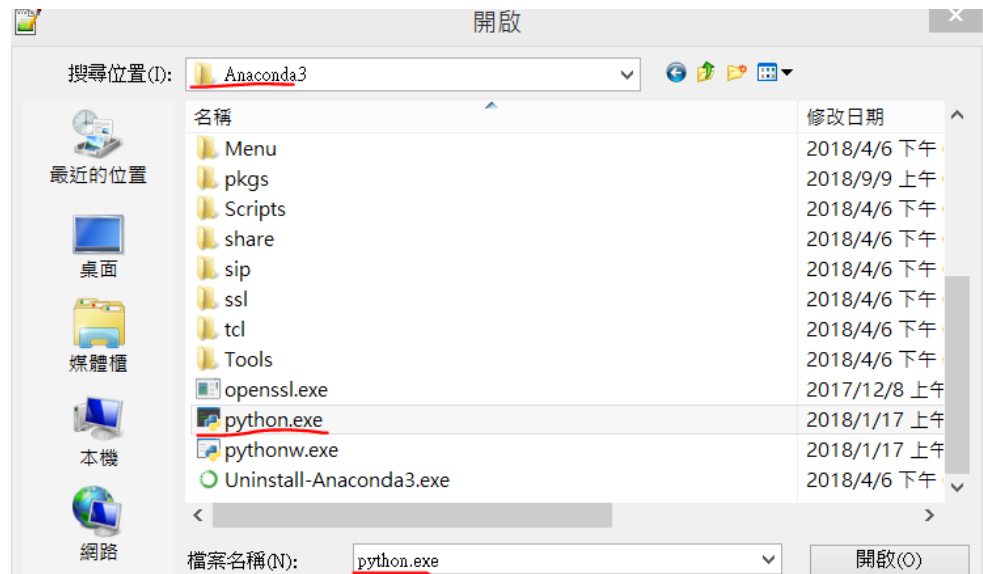
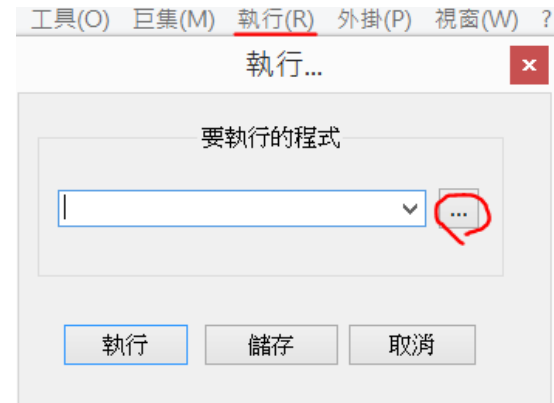
❑ Notepad++ 下載安裝

❑ 選單 - 執行

❑ 選擇 Anaconda3 目錄

○ 在C:\ProgramData\Anaconda3

○ 選 python.exe



使用NotePad++撰寫

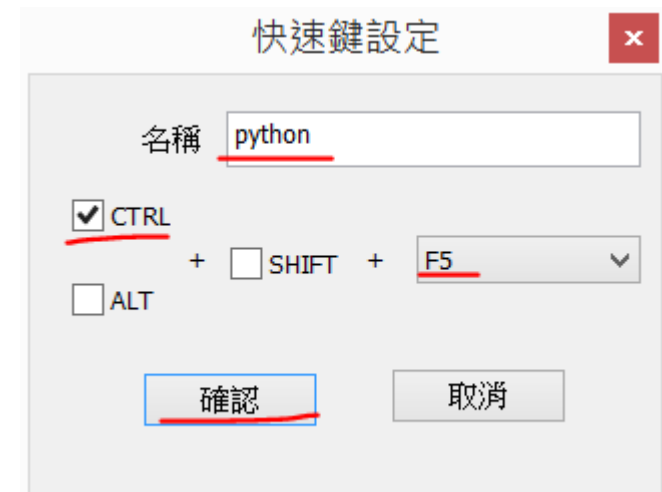
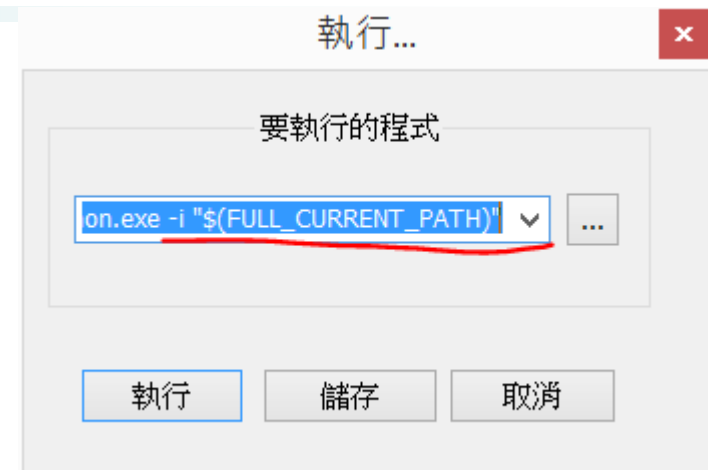
- ❑ 選擇 Anaconda3 目錄，
- ❑ 在python.exe 空一格填入
 - -i "\$(FULL_CURRENT_PATH)"

❑ 變成

- C:\ProgramData\Anaconda3\python.exe -i "\$(FULL_CURRENT_PATH"

❑ 儲存－快速鍵設定

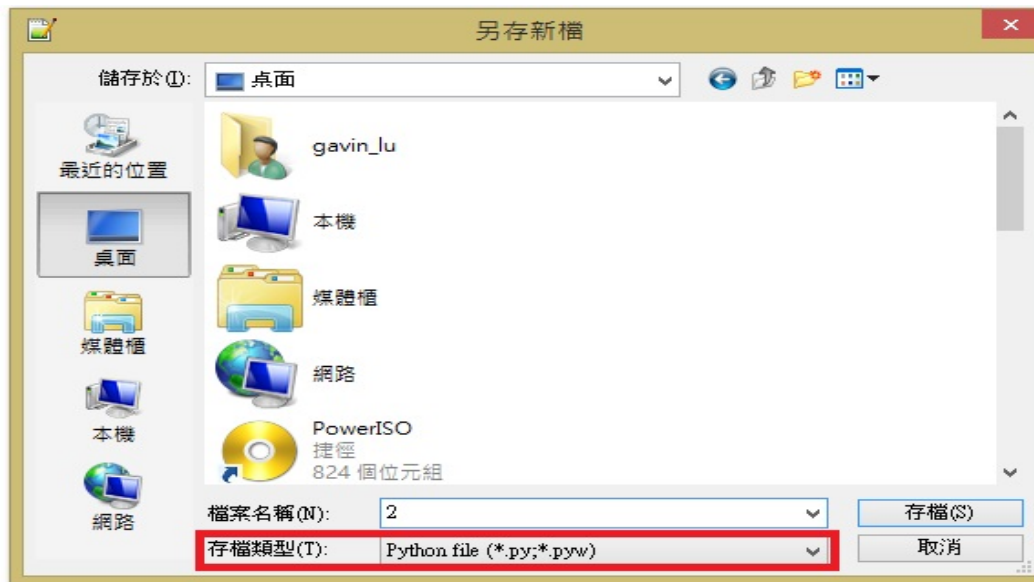
- python, CTRL-F5



使用NotePad++撰寫

- ❑ 輸入以下程式，儲存檔案，存檔類型->Python file

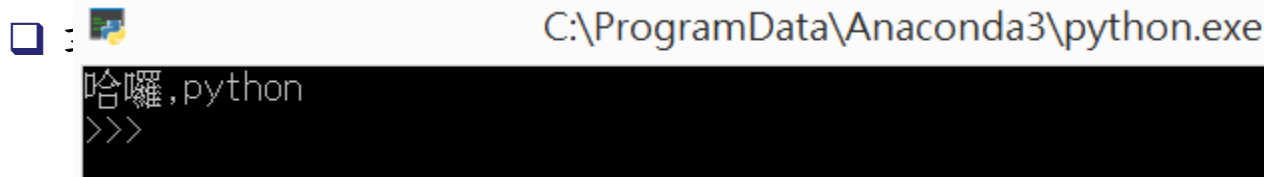
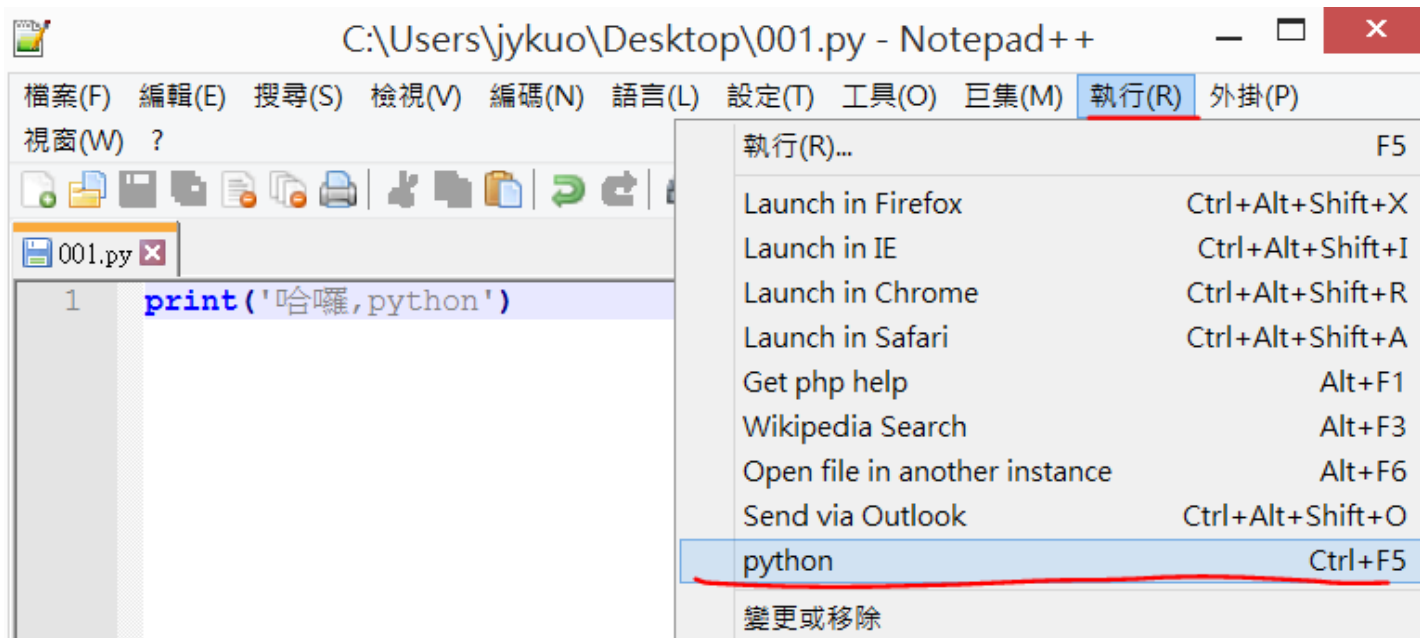
```
print('哈囉,python')
```



使用Notepad++撰寫

❑ 執行程式碼

○ 選單-執行 – python 或 CTRL+F5



使用 VS Code 撰寫

❑ 安裝 VS Code 環境

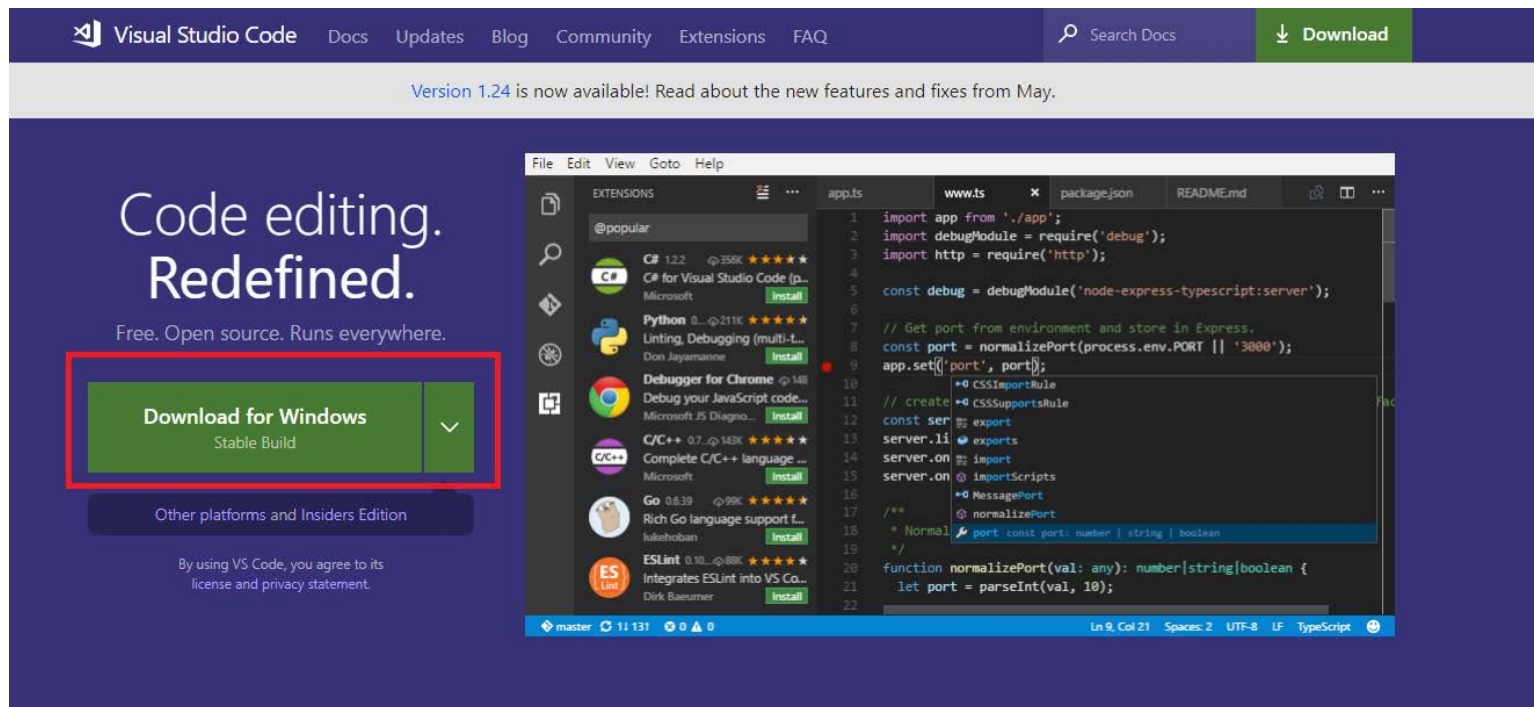
- 檢視- 擴充功能

❑ 安裝 pylint

- PyLint 是一種廣泛使用的工具，可檢查 Python 程式碼中的錯誤，有助於撰寫良好的 Python 程式碼模式，因此 Visual Studio 已針對 Python 專案整合這項工具。
- 打開 command line > `pip install pylint`

使用 VS Code 撰寫

□ 前往 VS Code 官網



使用 VS Code 撰寫

□ 打開檢視 > 擴充功能



□ 搜尋python > 點擊安裝



安裝pyLint

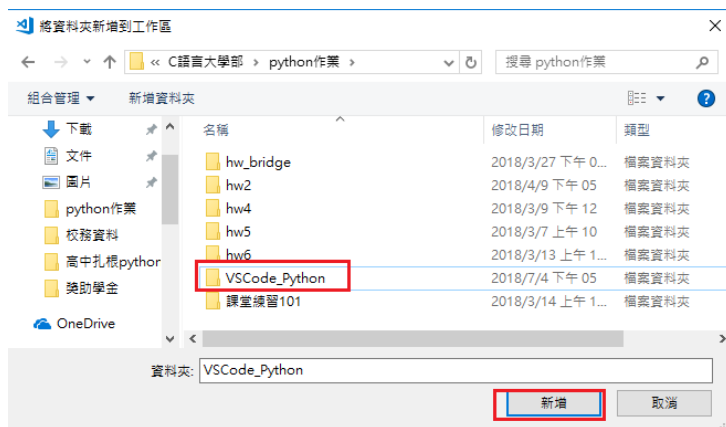
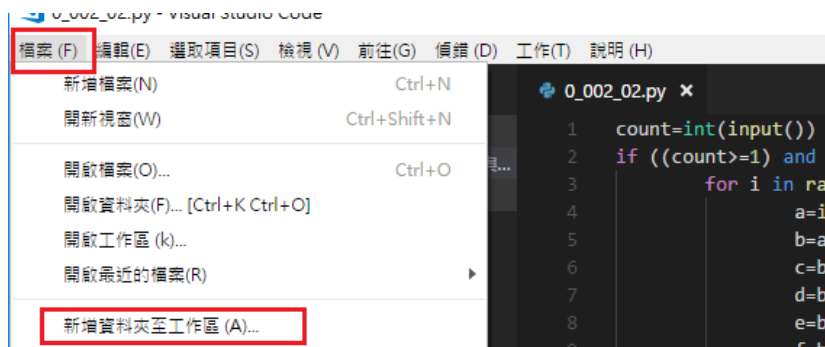
- ❑ PyLint 是一種廣泛使用的工具，可檢查 Python 程式碼中的錯誤，有助於撰寫良好的 Python 程式碼模式，因此 Visual Studio 已針對 Python 專案整合這項工具。
- ❑ 打開command line > pip install pylint

```
C:\Users\user>pip install pylint
Collecting pylint
  Downloading https://files.pythonhosted.org/packages/f2/95/0ca03c818ba3cd14f2dd4e95df5b7fa232424b7fc6
/pylint-1.9.2-py2.py3-none-any.whl (690kB)
100% |#####| 696kB 3.0MB/s
Collecting six (from pylint)
  Downloading https://files.pythonhosted.org/packages/67/4b/141a581104b1f6397bfa78ac9d43d8ad29a7ca43ea
/six-1.11.0-py2.py3-none-any.whl
Collecting astroid<2.0,>=1.6 (from pylint)
  Downloading https://files.pythonhosted.org/packages/0e/9b/18b08991c8c6aaa827faf394f4468b8fee41db1f73
/astroid-1.6.5-py2.py3-none-any.whl (293kB)
100% |#####| 296kB 3.2MB/s
Collecting colorama; sys.platform == "win32" (from pylint)
```


安裝pyLint

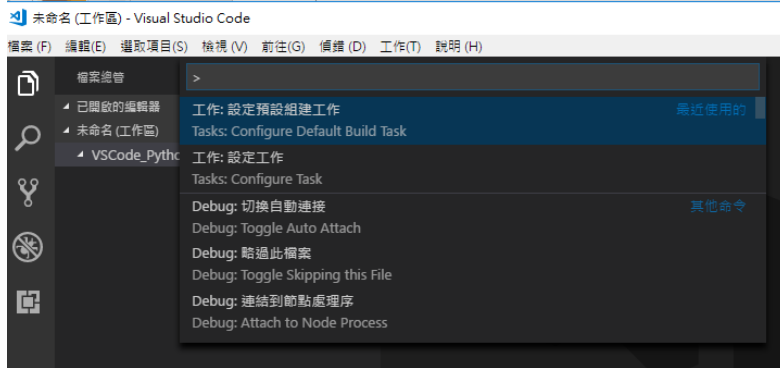
❑ 將資料夾新增到工作區

- Step 1. 點擊 檔案 > 新增資料夾至工作區
- Step 2. 新建一個資料夾，將其選擇為工作區



設定Compiler

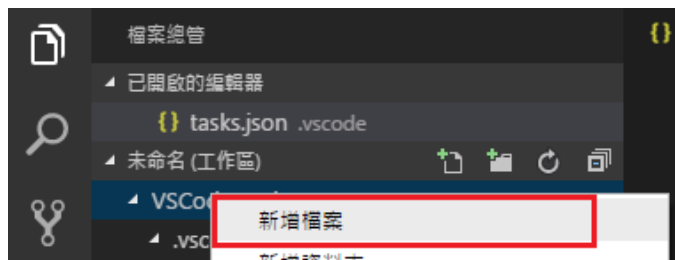
- ❑ Step 1. 按 F1
- ❑ Step 2. 點選上方Task: Configure Default Build Task(如下圖)
- ❑ Step 3. 從範例建立tasks.json檔案
- ❑ Step 4. 點選Others，VS Code會產出一個tasks.json
- ❑ Step 5. 將tasks.json內容改為右下圖



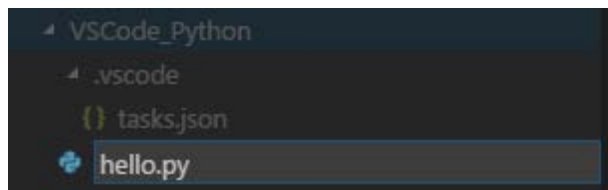
```
tasks.json x
1 {
2   // See https://go.microsoft.com/fwlink/?LinkId=
3   // for the documentation about the tasks.json
4   "version": "2.0.0",
5   "tasks": [
6     {
7       "label": "python Compiler",
8       "type": "shell",
9       "command": "python",
10      "args": ["${file}"]
11    }
12  ]
13 }
```

新增python檔

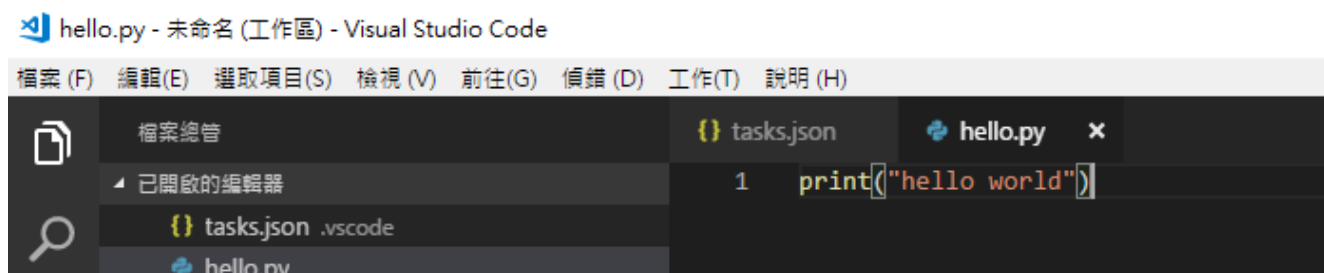
- Step 1. 工作區點選右鍵，新增專案



- Step 2. 建立python檔(取名為hello，副檔名.py)



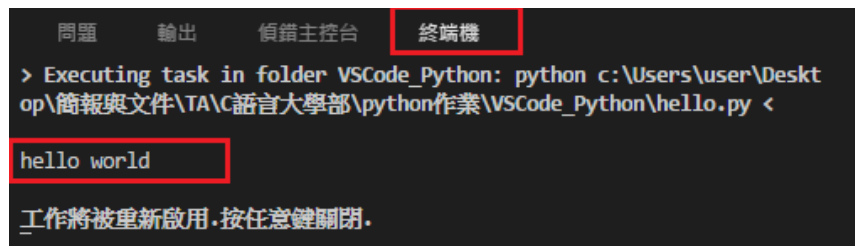
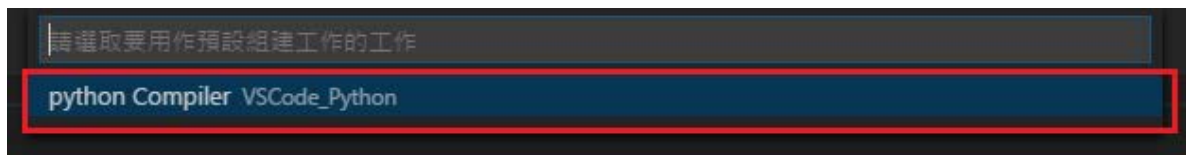
- Step 3. 編寫內容，並點選儲存(ctrl+S)



執行python

❑ Step 1. 點選Ctrl+ Shift + B (OS X：CMD + SHIFT + B)

○ 如果沒有選擇Compiler則會彈出下面圖示，點選紅框部分設定即可



Linux 安裝Python

❑ 環境

- 作業系統 : Ubuntu 18.04
- Python版本 : Python 3.8
- Tensorflow : Tensorflow 1.10.1
- OpenCV : OpenCV 3.4.3

❑ `sudo apt-get update`

- 若剛安裝ubuntu請輸入以上指令更新。

❑ `sudo apt-get install python3-pip`

- 若pip3沒有安裝，請輸入以上指令，若有請略過。

❑ `pip3 install --upgrade pip`

- 若pip3版本不是10.0以上版本，請輸入此指令，否則安裝opencv會有問題，若是請略過。

Linux 安裝Python

❑ 更新後使用pip3指令 若出現 ImportError: cannot import name 'main' 問題

- 前往路徑cd /usr/bin

- 輸入指令sudo vi pip3 或 sudo gedit pip3

- 進行修改，將程式碼改為

- ```
from pip import __main__
```

- ```
if __name__ == '__main__':
```

- ```
 sys.exit(__main__._main())
```

- 儲存後即可正常使用pip3

# Linux 安裝Python

## ❑ 安裝python套件

- `sudo pip3 install opencv-python`
- `sudo pip3 install keras`
- `sudo pip3 install pandas`
- `sudo pip3 install tensorflow`
- `sudo pip3 install matplotlib`
- `sudo apt install python3-tk` (此為ub才需要安裝)
- `sudo apt install tk-dev` (此為ub才需要安裝)

# Python指令行執行

郭忠義

[jykuo@ntut.edu.tw](mailto:jykuo@ntut.edu.tw)

臺北科技大學資訊工程系

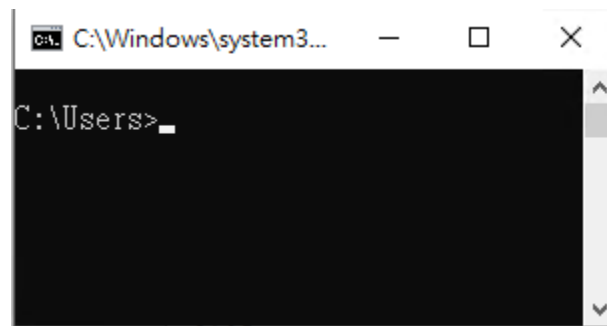
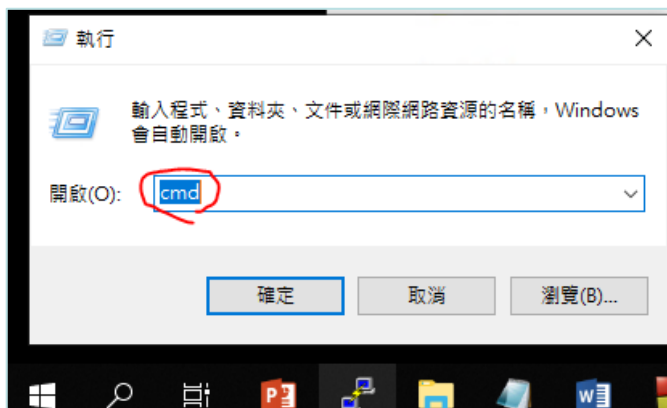


# 開啟指令行

- ❑ Windows左下角圖示點選，右鍵，執行



- ❑ 輸入cmd按確定，開啟指令行執行視窗

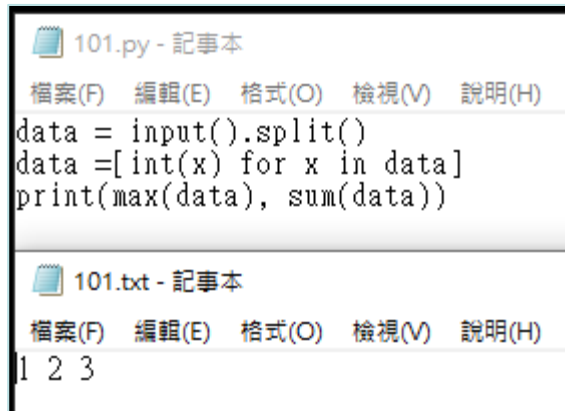


# 編輯程式與測試資料檔案

❑ 將.py程式、測試資料文字檔.txt放置於某一目錄，例如，

○ 編輯 101.py

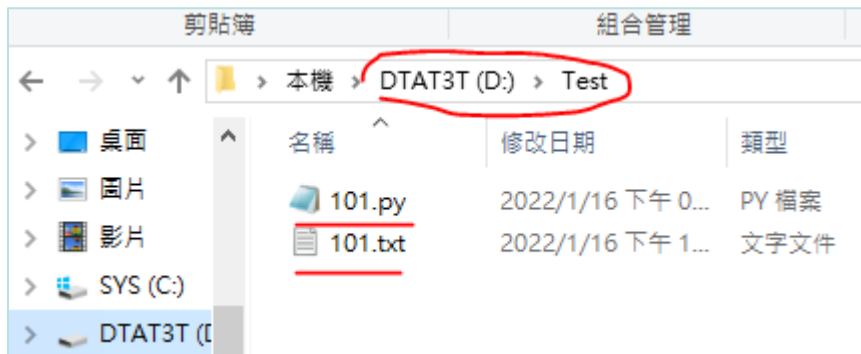
○ 編輯測試資料檔案101.txt



```
101.py - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
data = input().split()
data = [int(x) for x in data]
print(max(data), sum(data))

101.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
1 2 3
```

○ 將 101.py, 101.txt，放置於 d:\Test



# 安裝 Anaconda

## ❑ 在cmd視窗下指令切換到d:\Test

○ d:

○ cd Test

```
C:\Windows\system32\cmd.exe
C:\Users>d:
D:\>cd Test
D:\Test>
```

## ❑ 確認101.py, 101.txt在該目錄

○ dir

○ type 101.py

○ type 101.txt

```
C:\Windows\system32\cmd.exe
D:\Test>dir
磁碟區 D 中的磁碟是 DTAT3T
磁碟區序號: 82EF-AFE4

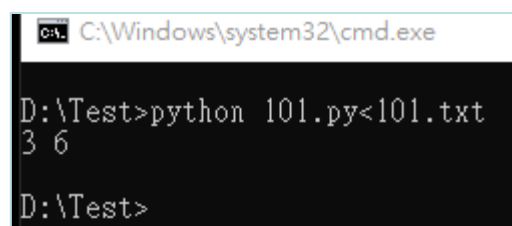
D:\Test 的目錄
2022/01/16 下午 01:00 <DIR> .
2022/01/16 下午 01:00 <DIR> ..
2022/01/16 下午 01:01 81 101.py
2022/01/16 下午 12:59 5 101.txt
 2 個檔案 86 位元組
 2 個目錄 1,873,009,532,928 位元組可用
```

```
C:\Windows\system32\cmd.exe
D:\Test>type 101.py
data = input().split()
data = [int(x) for x in data]
print(max(data), sum(data))
D:\Test>type 101.txt
1 2 3
D:\Test>
```

# 安裝 Anaconda

- ❑ 執行101.py，使用101.txt當輸入

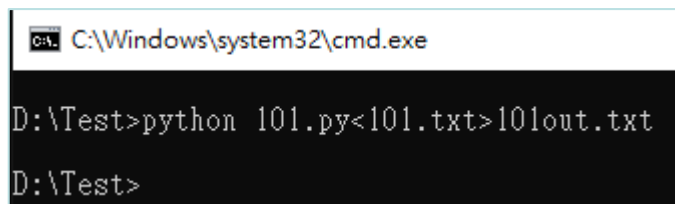
- python 101.py < 101.txt



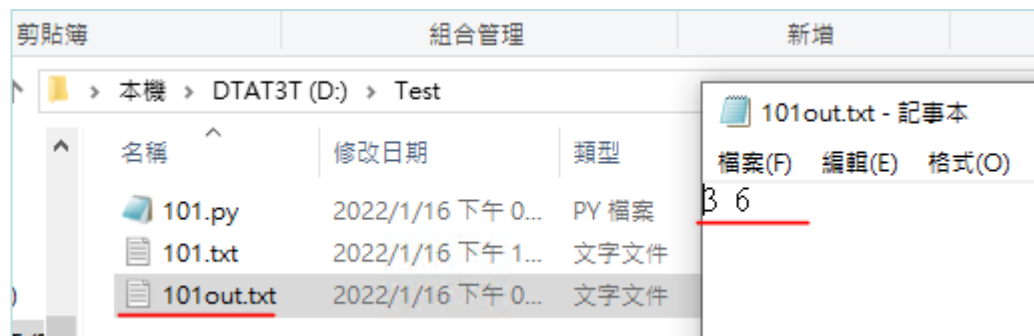
```
C:\Windows\system32\cmd.exe
D:\Test>python 101.py<101.txt
3 6
D:\Test>
```

- ❑ 執行101.py，使用101.txt當輸入，輸出到101out.txt

- python 101.py < 101.txt >101out.txt



```
C:\Windows\system32\cmd.exe
D:\Test>python 101.py<101.txt>101out.txt
D:\Test>
```



# Python

# 計算機程式設計

郭忠義

[jykuo@ntut.edu.tw](mailto:jykuo@ntut.edu.tw)

臺北科技大學資訊工程系

# 計算機程式設計

## □ 資訊系統開發流程

- 需求/問題的取得/理解 – 需求規格書
  - 文字描述、圖形介面呈現
- 系統分析 - 系統規格書
  - 測試案例設計、測試劇本/使用情境設計
- 系統設計
  - 資料結構設計
  - 程式流程設計/問題處理邏輯設計
- 程式碼撰寫
  - 程式語法正確性 (筆試/上機測驗)
  - 程式邏輯正確性 (筆試/上機測驗)
- 系統驗證
  - 靜態分析 – 程式碼檢視 – 人工追蹤程式運作 (筆試)
  - 動態測試 – 測試案例、測試程式執行 (上機測驗)

# 軟體品質問題 1

- ❑ 2020：南山人壽境界成就計畫問題。
- ❑ 2014：臺灣新戶政系統初上線因相容問題，系統無法製作身分證明。
- ❑ 2014：臺灣高速公路收費系統扣款有問題。
- ❑ 2007：臺灣高鐵售票系統剛上線，壓力測試不足，系統無法應付龐大湧現購票人潮。
- ❑ 2007：臺灣彩卷系統因十億獎金引爆人潮，造成系統頻繁當機。
- ❑ 2000：巴拿馬國家癌症中心，病人接受過量放射線照射，5位喪生，15位嚴重併發症。
- ❑ 2000：美國海軍飛機因控制軟體缺陷而墜落，4人喪生。
- ❑ 1997：韓國航空因雷達控制軟體缺陷，225人喪生。
- ❑ 1995：美國航空因導航軟體缺陷，在哥倫比亞撞山造成159人喪生。
- ❑ 1994：華航飛機自動駕駛軟體重飛模式和手動控制升降舵相互對抗，名古屋空難264人喪生。
- ❑ 1991：波灣戰爭，愛國者反導彈校時系統缺陷，攔截飛毛腿飛彈失誤，28名士兵喪生。

# 軟體品質問題2

- ❑ 1963年美國太空總署，一個FORTRAN 程式迴圈敘述
  - DO 5 I=1,3 -----(正確)
- ❑ 被人為錯誤打成
  - DO 5 I=1.3 -----(錯誤)
  - FORTRAN編譯器視為
  - DO5I=1.3 -----(缺陷)
- ❑ 導致飛往火星的火箭爆炸，造成一千萬美元的損失。