

附錄 A

安裝與執行 Python

A-1 : Windows 作業系統的安裝 Python 版

A-2 : 啟動 Python 可執行檔案

A-3 : 找尋 Python 可執行檔案

A-4 : 在 Python Shell 編輯環境

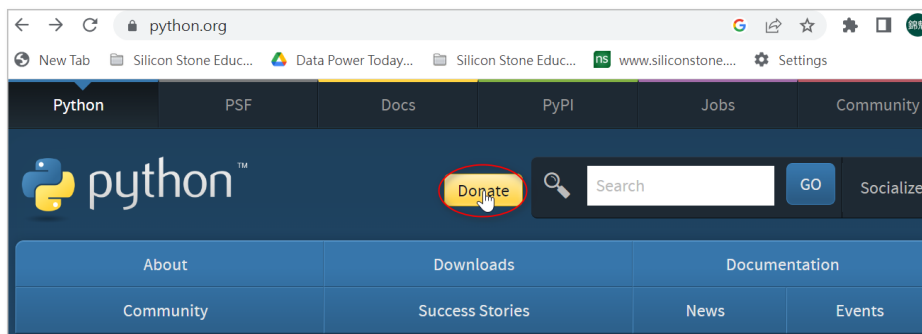
A-5 : 進入編輯 Python 程式環境

A-6 : matplotlib 建立含中文的圖表

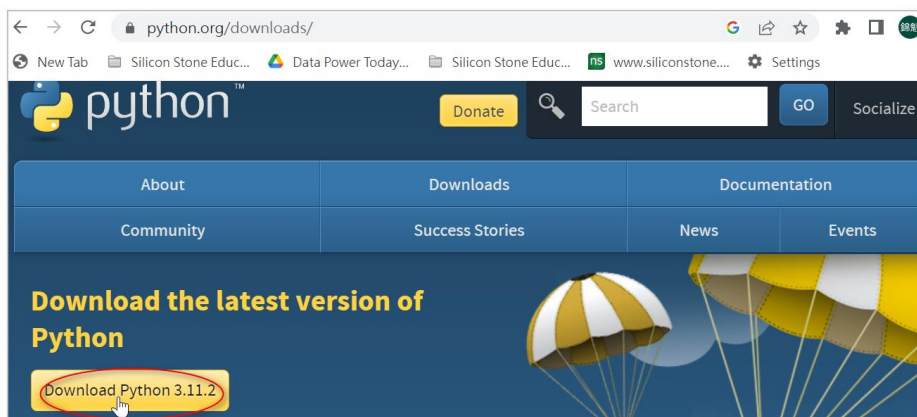
Python 安裝程式在安裝前會先偵測你的電腦使用環境，然後自動協助選擇安裝程式。請先進入下列網頁：

www.python.org

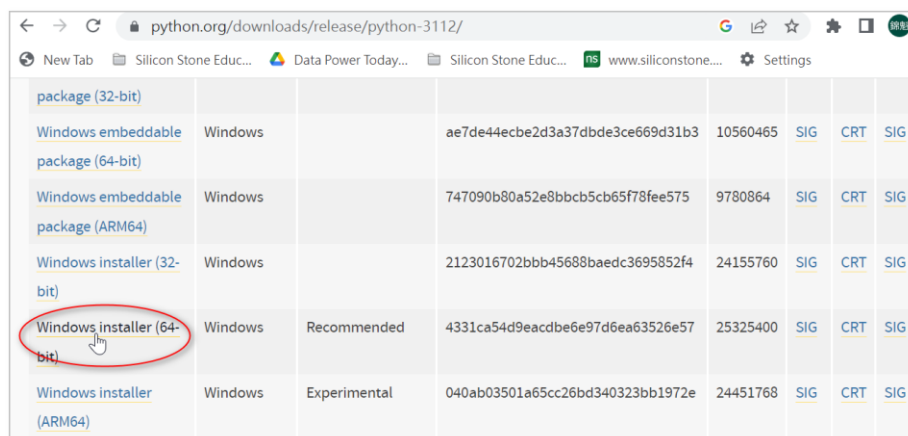
在螢幕可以看到 Downloads，請按一下 Downloads。



可以看到下列畫面。



請按一下 Download Python 3.11.2 鈕。



上述是依照使用的電腦選擇 Python 版本，筆者選擇 Windows installer(64-bit)，讀者可依照自己的電腦做選擇，點選後就可以下載 Python 的安裝程式，成功後

可以在瀏覽器視窗下方看到所下載的程式。

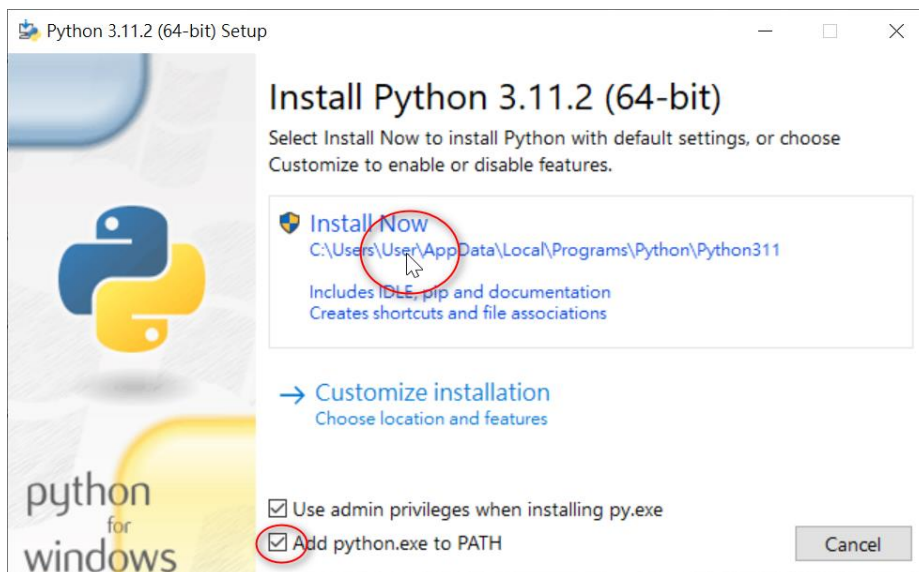


上述請點選開啟。

~~點選上述就可以安裝，雖然下列是 3.7 版本的安裝畫面，但是意義與方式觀念相同。~~

A-1：Windows 作業系統的安裝 Python 版

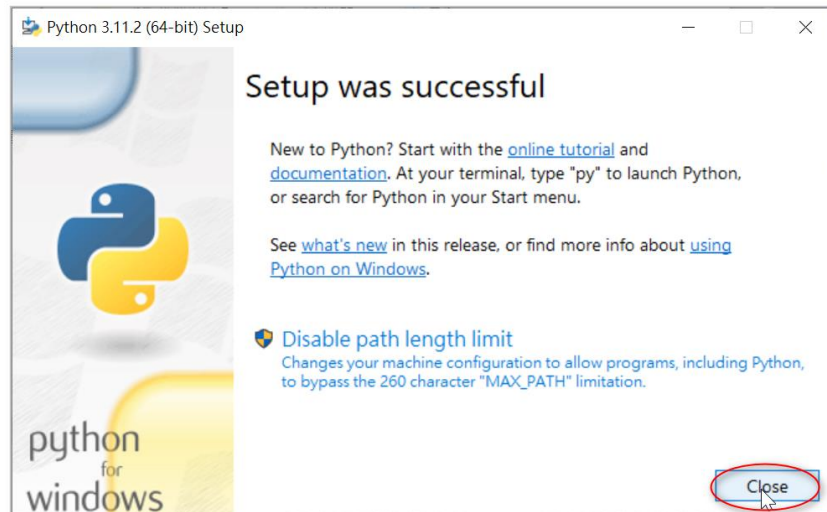
此時讀者可以選擇下載那一個版本，此例筆者選擇下載 3.7 版，筆者使用 Internet Explorer 流覽器然後請按執行鈕，電腦將直接執行位於下載區的 python-3.7.exe 檔案，進行安裝，然後將看到下列安裝畫面：



註 1：如果點選 Add python.exe to PATH，不論是在那一個資料夾均可以執行 python 可執行檔，非常方便。預設畫面是未勾選狀態，建議勾選。

註 2：上述預設安裝路徑是在比較深層的 C:\資料夾路徑，如果想安裝在比較淺層，建議可以點選 Customize installation，然後再選擇路徑，例如：選擇 C:\即可。

下列是筆者採用預設安裝路徑的畫面，上述如果點選 Install Now 選項可以進行安裝，下方可以看到，未來安裝 Python 的所在的資料夾。安裝完成後將看到下列畫面。



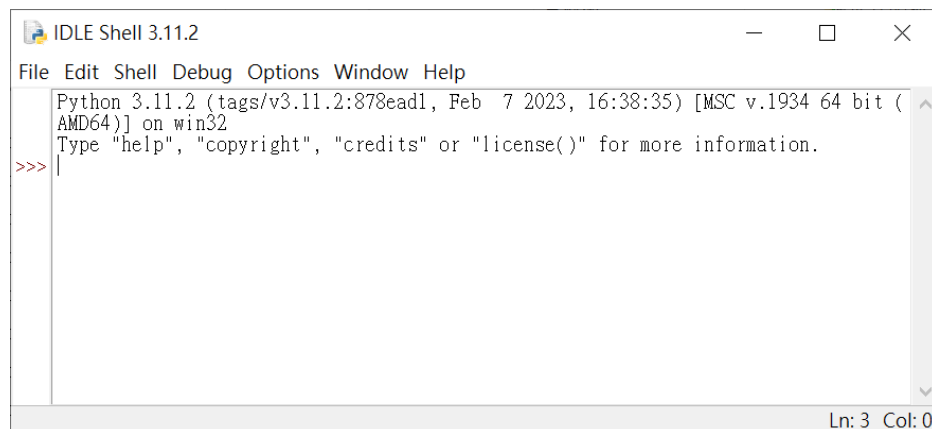
安裝完成後，可以按 Close 鈕。

A-2：啟動 Python 可執行檔案

點選 Windows 螢幕左下角的視窗鈕，然後選擇 Python 3.11，同時點選 IDLE，如下所示：

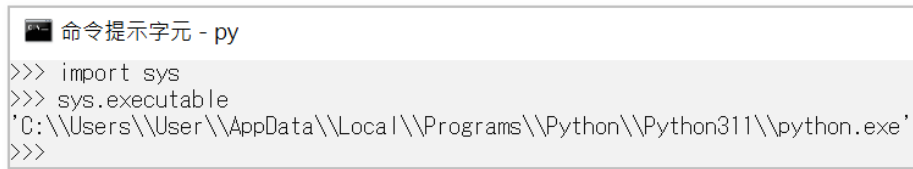


就可以啟動 Python，下列是啟動的畫面，此視窗又稱 Python Shell 視窗或是 IDLE 環境視窗。



A-3 : 找尋 Python 可執行檔的路徑

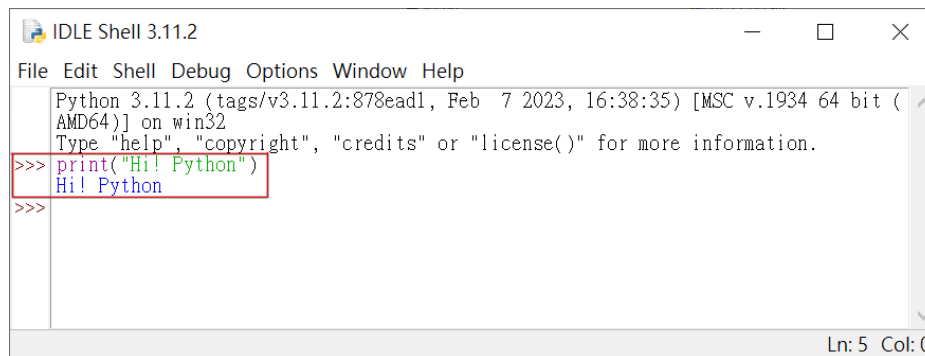
請在 Windows 左下角搜尋框輸入 `cmd`，可以開啟命令提示字元視窗，然後參考下列輸入，就可以找到 Python 3.11 版本程式所在的資料夾路徑。



```
>>> import sys
>>> sys.executable
'C:\\Users\\User\\AppData\\Local\\Programs\\Python\\Python311\\python.exe'
>>>
```

A-4 : 在 Python Shell 編輯環境

在 IDLE 的 Python Shell 視窗可以輸入指令，同時獲得執行結果。

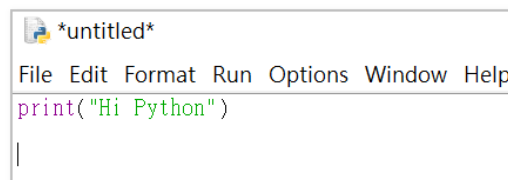


```
Python 3.11.2 (tags/v3.11.2:878ead1, Feb 7 2023, 16:38:35) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print('Hi! Python')
Hi! Python
>>>
```

A-5 : 進入編輯 Python 程式環境

A-5-1 : 進入編輯環境

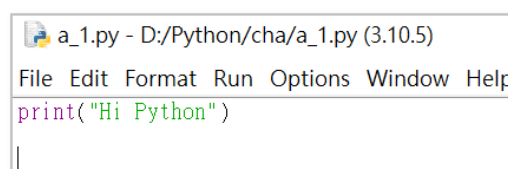
在 Python Shell 視窗執行 **File/New File** 指令，可以開啟視窗編輯程式，下列是示範。



```
*untitled*
File Edit Format Run Options Window Help
print('Hi Python')
|
```

A-5-2 : 儲存檔案

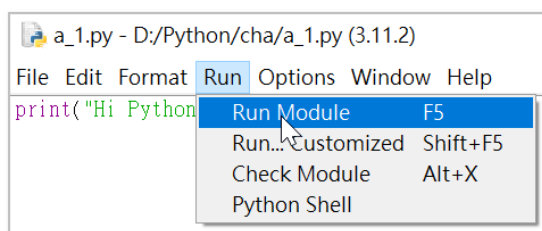
執行 **File/Save As** 指令可以儲存檔案，下列是將檔案儲存至 D:\Python\cha 資料夾的 `a_1.py` 的結果。



```
a_1.py - D:/Python/cha/a_1.py (3.10.5)
File Edit Format Run Options Window Help
print('Hi Python')
|
```

A-5-3：執行程式

執行 **Run/Run Module** 可以執行程式，如下所示：



執行後，可以在 **Python Shell** 視窗得到執行結果。

```
===== RESTART: D:/Python/cha/a_1.py =====  
Hi Python
```

A-6：Pillow 模組 - 在影像內填寫文字

ImageDraw 模組也可以用於在影像內填寫英文或中文，所使用的函數是 **text()**，語法如下：

text((x,y), text, fill, font) # **text** 是想要寫入的文字

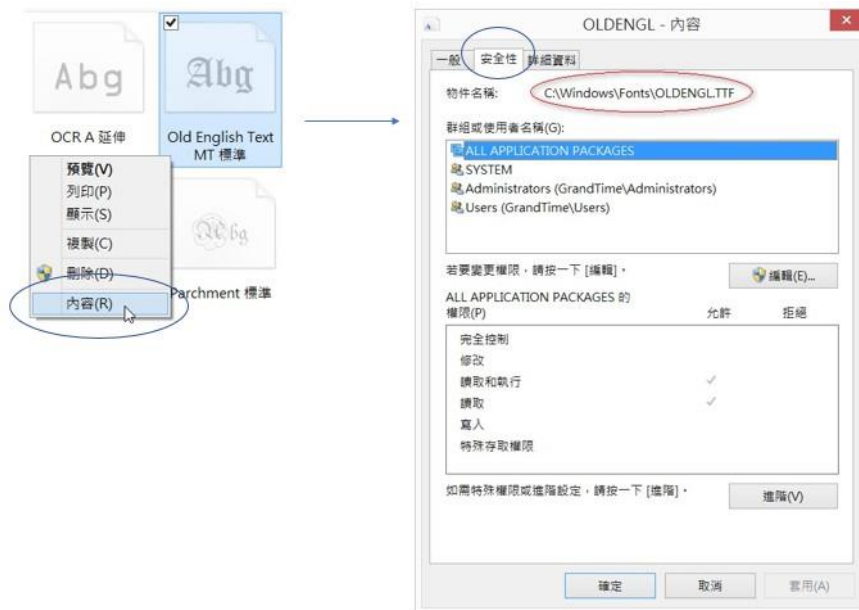
如果要使用預設方式填寫文字，可以省略 **font** 參數，可以參考 **a_2.py** 第 8 列。如果想要使用其它字型填寫文字，需呼叫 **ImageFont.truetype()** 方法選用字型，同時設定字型大小。在使用 **ImageFont.truetype()** 方法前需在程式前方導入 **ImageFont** 模組，可參考 **a_2.py** 第 2 列，這個方法的語法如下：

text(字型路徑, 字型大小)

在 **Windows** 系統字型是放在 **C:\Windows\Fonts** 資料夾內，在此你可以選擇想要的字型。



點選**字型**，按**滑鼠右鍵**，執行**內容**，再選**安全性**標籤可以看到此字型的檔案名稱。下列是點選 **Old English Text** 的示範輸出。



讀者可以用複製方式獲得字型的路徑，有了字型路徑後，就可以輕鬆在影像內輸出各種字型了。

程式實例 [a_2.py](#)：在影像內填寫文字，第 7-8 列是使用預設字型，執行英文字串”Ming-Chi Institute of Technology”的輸出。第 10-11 列是設定字型為 Old English Text，字型大小是 36，輸出相同的字串。第 13-15 列是設定字型為新細明體，字型大小是 48，輸出中文字串”明志科技大學”。

```

1 # a_2.py
2 from PIL import Image, ImageDraw, ImageFont
3
4 newImage = Image.new('RGBA', (600, 300), 'Yellow') # 建立300*300黃色底的影像
5 drawObj = ImageDraw.Draw(newImage)
6
7 strText = 'Ming-Chi Institute of Technology' # 設定欲列印英文字串
8 drawObj.text((50,50), strText, fill='Blue') # 使用預設字型與字型大小
9 # 使用古老英文字型，字型大小是36
10 fontInfo = ImageFont.truetype('C:\Windows\Fonts\OLDENGL.TTF', 36)
11 drawObj.text((50,100), strText, fill='Blue', font=fontInfo)
12 # 使用Microsoft所提供的新細明體中文字型處理中文字體
13 strCtext = '明志科技大學' # 設定欲列印中文字串
14 fontInfo = ImageFont.truetype('C:\Windows\Fonts\mingliu.ttc', 48)
15 drawObj.text((50,180), strCtext, fill='Blue', font=fontInfo)
16 newImage.save("outa_2.png")

```

執行結果



A-7 : matplotlib 建立含中文的圖表

一個圖表無法顯示中文，坦白說讀者內心一定感覺有缺憾，至少筆者感覺如此。matplotlib 無法顯示中文主要是安裝此模組時所配置的檔案：

~Python38\Lib\site-packages\matplotlib\mpl-data\matplotlibrc

註：上述 38 是 Python 版本編號。

在此檔案內的 `font_sans-serif` 沒有配置中文字型，我們可以使用在此欄位增加中文字型，但是筆者不鼓勵更改系統內建檔案。筆者將使用動態配置方式處理，讓圖表顯示中文字型。其實可以在程式內增加下列程式碼，`rcParams()` 方法可以配置中文字型參數，就可以顯示中文了。

```
plt.rcParams['font.family'] = ['Microsoft JhengHei']    # 設定中文字體
plt.rcParams['axes.unicode_minus'] = False              # 可以顯示負號
```

這時所有圖表文字皆會改成上述微軟正黑體(Microsoft JhengHei)，讀者可以任選 C:\Windows\Fonts 內的字型名稱。

程式實例 a_3.py：含中文圖表的實例。

```
1 # a_3.py
2 import matplotlib.pyplot as plt
3
4 plt.rcParams["font.family"] = ["Microsoft JhengHei"]    # 可以顯示中文
5 plt.rcParams["axes.unicode_minus"] = False              # 可以顯示負號
6 squares = [0, 1, 4, 9, 16, 25, 36, 49, 64]
7 plt.plot(squares, lw=10)    # 串列squares數據是y軸的值，線條寬度是10
8 plt.title('圖表')
9 plt.xlabel('X 軸值')
10 plt.ylabel('平方')
11 plt.show()
```

執行結果

