

UNIT 1



pythonTM

Python Introduction, Setup Environment

張傑帆 Chang, Jie-Fan



OUTLINE

- Python 簡介
- 安裝環境 + 安裝 Python
- Python 使用方式



PYTHON簡史

- 創始人為吉多·范羅蘇姆
(Guido van Rossum)
- 在荷蘭出生、成長，1982年在阿姆斯特丹大學獲得數學和計算機科學碩士學位
- 在1989年12月，為打發聖誕節的無趣，便開發一個新的指令碼解釋程式來時間
- 以BBC喜劇Monty Python's Flying Circus命名



BRIEF HISTORY OF PYTHON

- Python是完全物件導向的語言
 - 函式、模組、數字、字串都是物件
- 完全支援繼承、重載、衍生、多繼承
 - 有益於增強原始碼的複用性
- Python被分類為「腳本語言」（script language）
 - 高階動態程式語言？
- Python 2.0於2000年10月16日發布，主要是實作了完整的垃圾回收（Garbage Collection），並且支援Unicode
- 2008年12月3日發布Python 3.0。它不完全相容之前的Python代碼。不過，很多新特性後來也被移植到舊的Python 2.6/2.7版本



PYTHON 簡介

- Python 官方對於自己所下的註解：

Python 語言是一種**簡單易學**、**功能強大**的程式語言。它具有高效率的**高階資料結構**，以及簡單且有效的**物件導向程式設計**方式。Python 語言具有**簡潔的語法**、**動態的型別**和**直譯式語言**的本質。由於擁有這些特質，使得Python 語言在處理程式腳本與**快速開發**應用程式方面，成為多數作業系統平台與眾多應用領域的理想程式語言。

- 可以從其中看到Python 的特性：
簡潔的語法 和 **直譯式語言**的本質



PYTHON 簡介

- Python 的特色簡述如下：
 - 自然性：
 - 以接近自然語言的方式來描述程式邏輯。
 - 易讀性：
 - 使用縮排區分程式碼區塊，讓程式碼方便閱讀。
 - 完整性：
 - 內建許多方便的函數，同時具有強大的標準函數庫及大量的第三方函數庫，讓您站在巨人的肩膀上可以看得更遠、走得更快。



PYTHON 簡介

- 開放性：

- 開放原始碼讓Python 快速發展與流通

- 易安裝性：

- 大部分的作業系統都內建有Python 的直譯器
 - Windows 的安裝過程簡單，可輕易體驗Python 的好處

- 相容與嵌入性：

- Python本身非常容易被擴充及大量的第三方函式庫
 - 很容易與C/C++ 等C 系列語言互通
 - 異種語言的程式碼互相嵌入Python 中一起使用
 - Python也可以嵌在其它程式裡面，這樣的特性讓Python非常有彈性



PYTHON介紹

```
def get_final_answer(filename):
```

```
    """說明文字"""
```

縮排

```
    line1
```

```
    line2
```

```
    return total_counter
```

■ 軟體品質

- 可讀性
- 強制縮排
- 物件導向

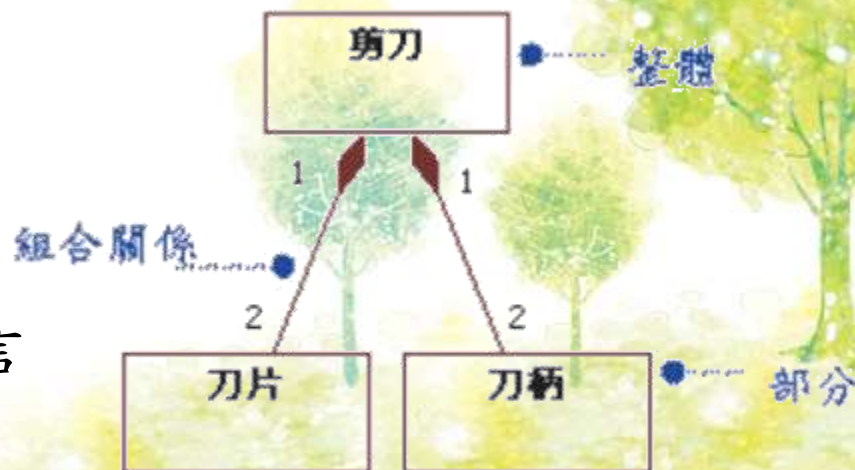
■ 動態語言

- 直譯式的語言
- 增加了使用上的彈性
- 節省重新編譯的時間

■ 強類型定義語言

- 強制數據類型定義的語言

■ 豐富的標準函式庫



簡介

Python的使用

- Google網站的搜尋系統
- Youtube視訊共享服務
- BitTorrent點對點檔案共享系統
- NSA的加密和智能分析
- iRobot開發商業機器人吸塵器
- NASA、Los Alamos、Fermilab、JPL的科學程式設計任務
- Industrial Light & Magic、Pixar製作電影動畫



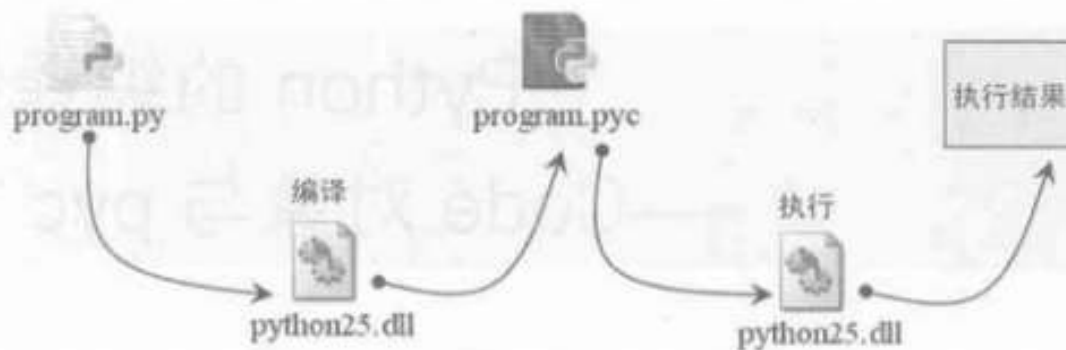
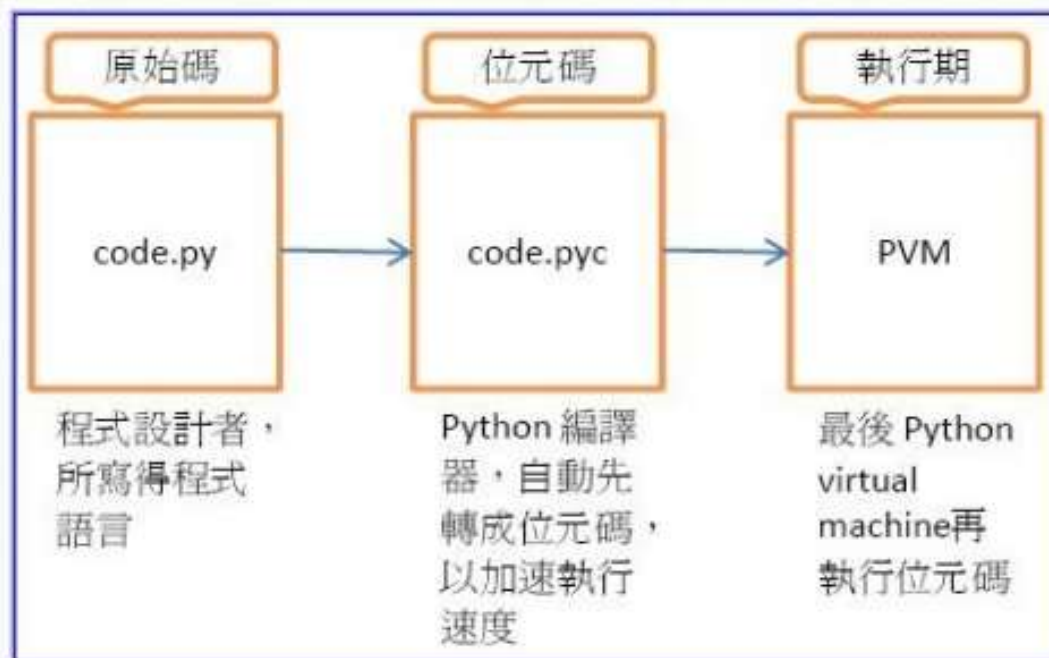
PYTHON 簡介

Aug 2014	Aug 2013	Change	Programming Language	Ratings	Change
1	2	▲	C	16.401%	+0.43%
2	1	▼	Java	14.984%	-0.99%
3	4	▲	Objective-C	9.552%	+1.47%
4	3	▼	C++	4.695%	-4.68%
5	7	▲	Basic	3.635%	-0.24%
6	6		C#	3.409%	-2.71%
7	8	▲	Python	3.121%	-0.48%
8	5	▼	PHP	2.864%	-3.83%
9	11	▲	Perl	2.218%	+0.18%
10	9	▼	JavaScript	2.172%	+0.08%
11	-	▲	Visual Basic	2.014%	+2.01%
12	13	▲	Visual Basic .NET	1.310%	-0.01%
13	10	▼	Ruby	1.242%	-0.83%
14	23	▲	F#	1.096%	+0.56%
15	18	▲	Pascal	1.044%	+0.42%
16	12	▼	Transact-SQL	1.043%	-0.35%
17	38	▲	ActionScript	1.001%	+0.69%
18	14	▼	Delphi/Object Pascal	0.915%	-0.00%
19	16	▼	Lisp	0.828%	+0.08%
20	17	▼	PL/SQL	0.786%	+0.03%



簡介

- Python如何執行



INTERPRETED LANGUAGES

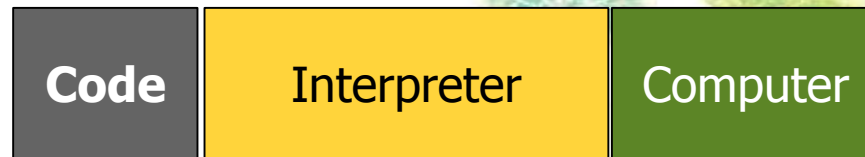
■ 直譯器

- 不需像Java或是C/C++那樣需要編譯Not(compiled)
- 程式碼一寫下後立即被「直譯器」直接翻譯
- 將命令輸入直譯器後便會立即得到結果

Java:



Python:



簡介

- 安裝

- Window

- Python-3.x.msi

- Linux

- 系統本身

- 進入Python互動直譯器 (for Linux)

- Python 2.7 以下版本

- #python

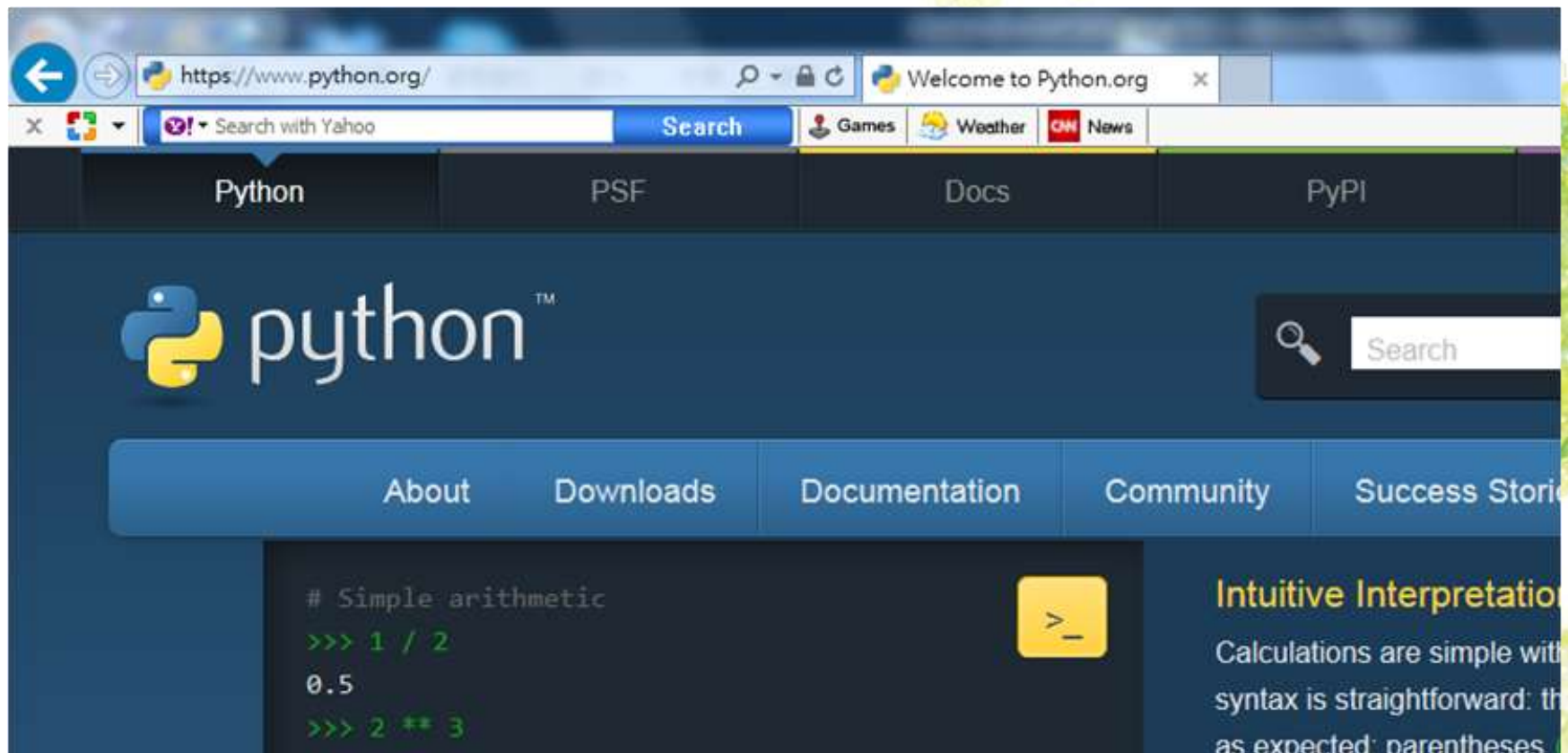
- Python 3.4

- #python3



安裝PYTHON

- Step 1：打開瀏覽器，在網址列輸入 <http://www.python.org> 以進入Python 的官網。



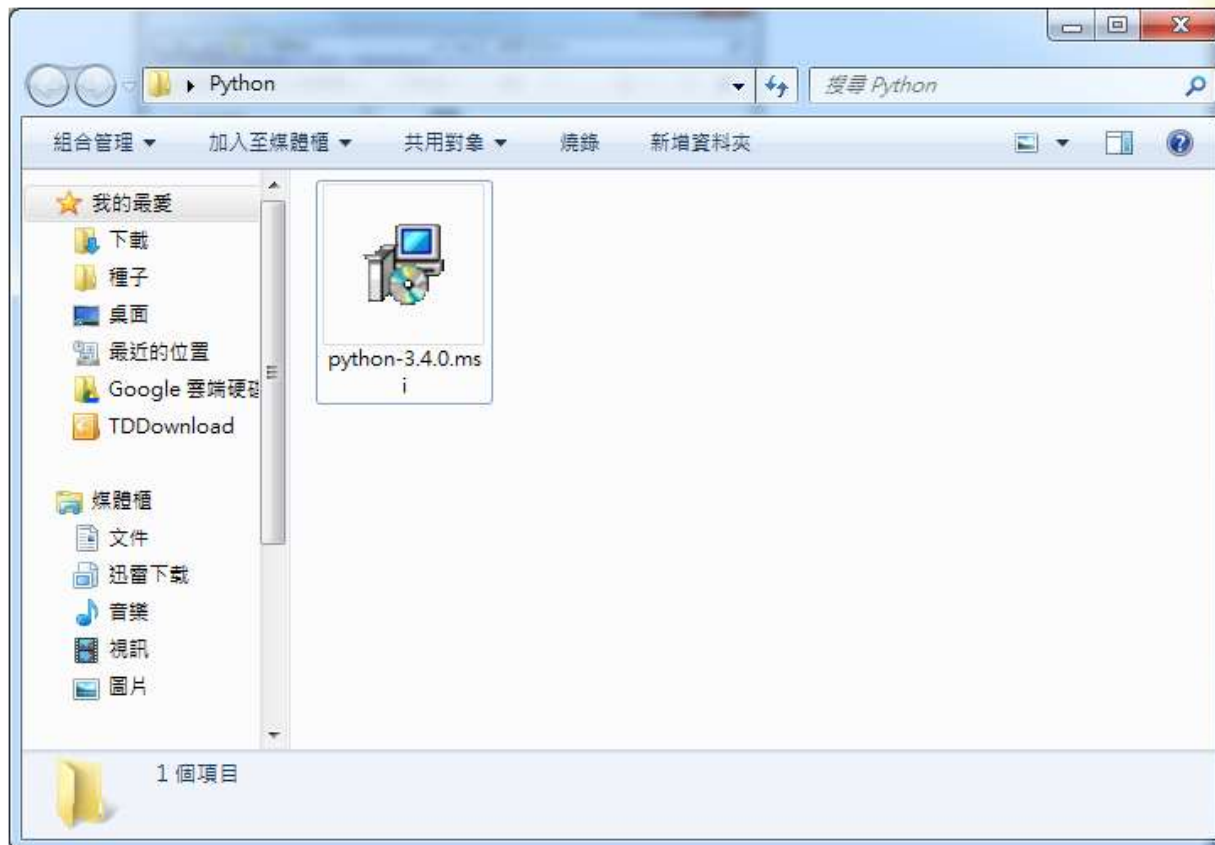
安裝PYTHON

- Step 2：點擊【下載（Download）】欄位，並選擇你所對應的開發環境。此處以Windows 64 位元下的Python 3.4.x 版本作為本課程之開發環境。



安裝PYTHON

- **Step 3**：下載完成後，在您所指定的下載路徑資料夾中會出現【python-3.4.0.msi】這個檔案，請點取並雙擊滑鼠左鍵後執行該安裝檔。



安裝PYTHON

- Step 4：如果出現安全性警告視窗（如下圖所示），請直接點選【執行】按鈕。



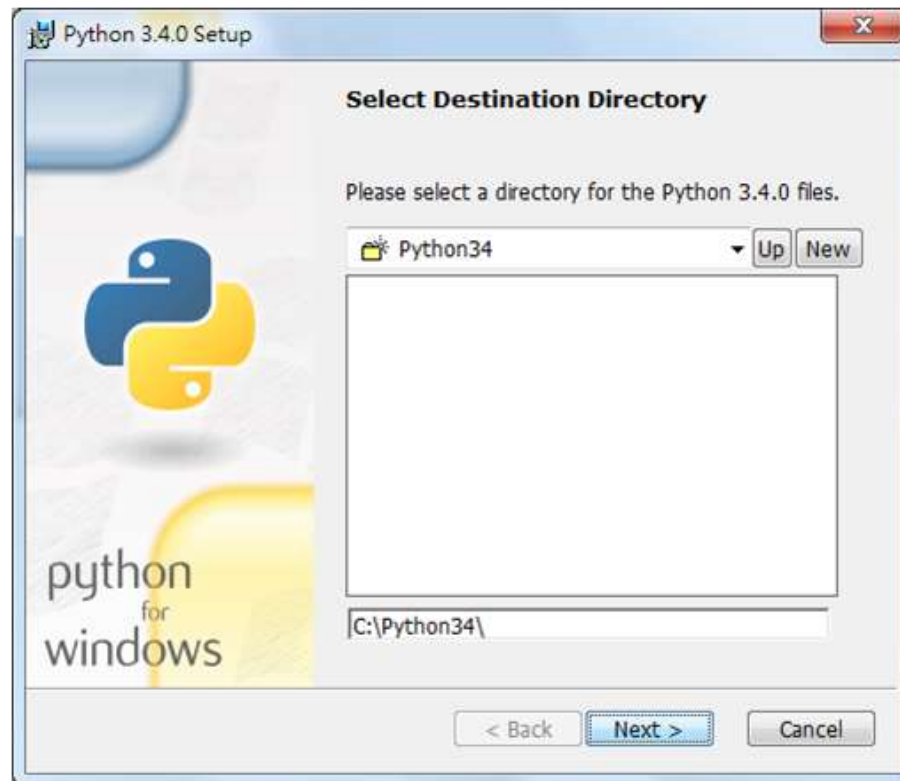
安裝PYTHON

- Step 5：選擇Python 環境要安裝給哪些使用者使用，此處以【安裝適用所有使用者（Install for all users）】為例。



安裝PYTHON

- Step 6：選擇安裝路徑，此處以預設【C:\Python34\】為例。



安裝PYTHON

- **Step 7**：請決定Python 哪些細項功能需要被安裝，此處僅更改最後一項【增加python.exe 到路徑（Add python.exe to Path）】，並點選【Next>】按鈕進行安裝，此時會出現等待安裝完成畫面。



Install Python 3.6.4 (32-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.



Install Now

C:\Users\user\AppData\Local\Programs\Python\Python36-32

Includes IDLE, pip and documentation
Creates shortcuts and file associations



Customize installation

Choose location and features

☒ Install launcher for all users (recommended)

☒ Add Python 3.6 to PATH

Cancel



python
for
windows



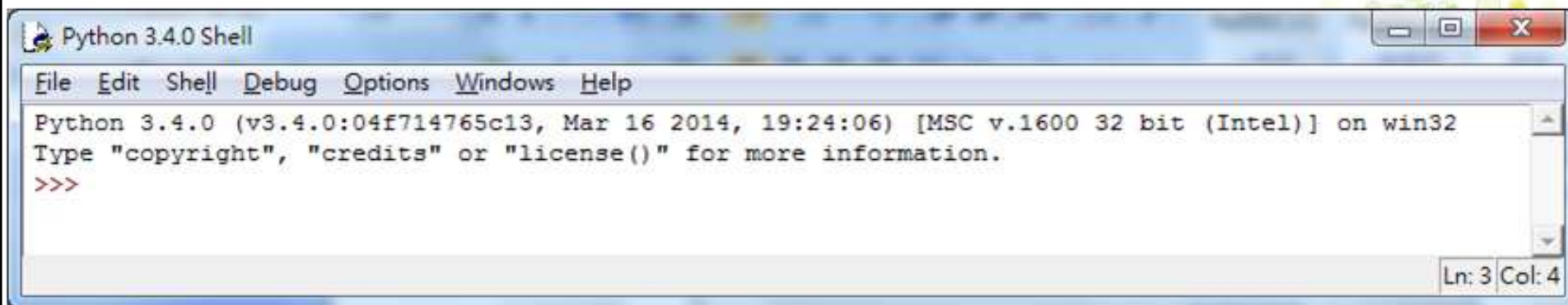
安裝PYTHON

- Step 8：點選【Finish】按鈕以完成此次安裝。



安裝PYTHON

- Step 9：點選Windows 的【開始】工作列，點選【所有程式】，找到【Python3.4】，點選【IDLE (Python GUI)】便可以開始藉由IDLE 體驗Python 的奇妙世界。



```
Python 3.4.0 Shell

File Edit Shell Debug Options Windows Help

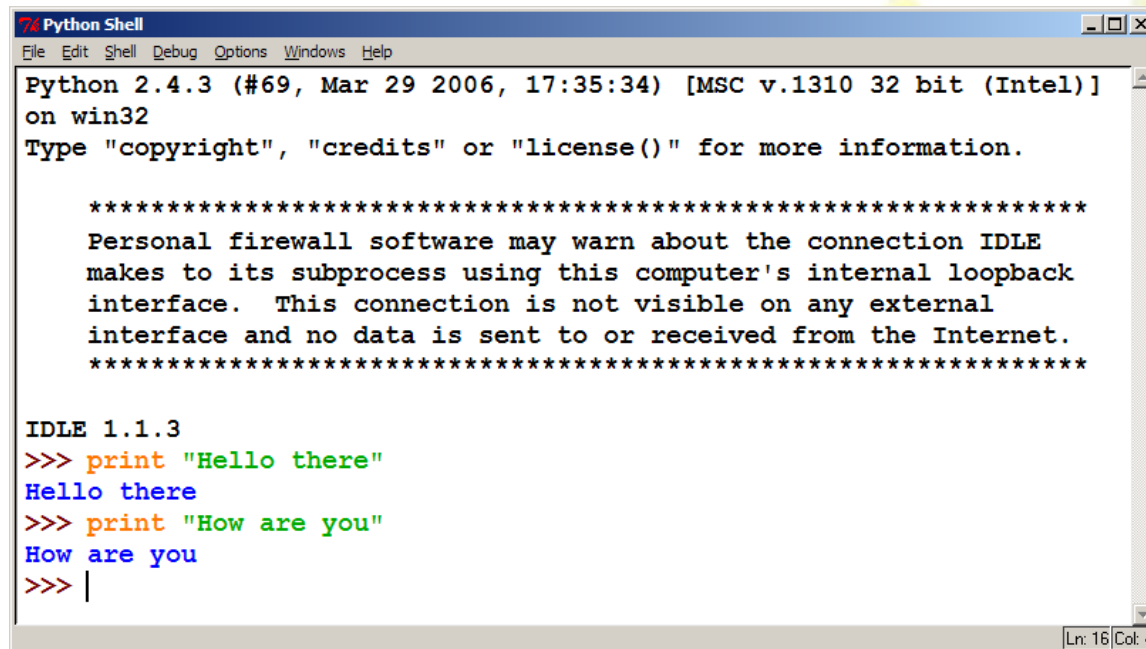
Python 3.4.0 (v3.4.0:04f714765c13, Mar 16 2014, 19:24:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
```

- ◆ File：開檔，關檔，存檔，開新視窗
- ◆ Edit：複製，貼上，取代，尋找，復原
- ◆ Shell：重啟shell
- ◆ Debug：除錯的功能與工具
- ◆ Option：可以設定IDLE
- ◆ Help：IDLE幫助與Python文件



PYTHON 直譯器

- 能讓你一次輸入一個指令並看到結果
- 很方便能讓你探索python指令與語法的方法
 - 重覆上一個指令: **Alt + p** (Ctrl + p on MAC)
 - 重覆下一個指令: **Alt + n** (Ctrl + n on MAC)
 - 指令自動完成: **TAB**



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 2.4.3 (#69, Mar 29 2006, 17:35:34) [MSC v.1310 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.

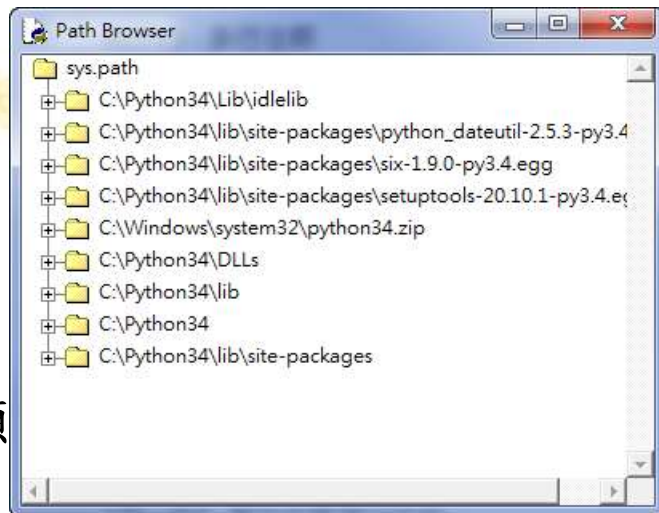
*****
Personal firewall software may warn about the connection IDLE
makes to its subprocess using this computer's internal loopback
interface. This connection is not visible on any external
interface and no data is sent to or received from the Internet.
*****

IDLE 1.1.3
>>> print "Hello there"
Hello there
>>> print "How are you"
How are you
>>> |
```



IDLE常用快捷

- **ALT+3** : 多行注釋
- **ALT+4** : 取消多行注釋
- **ALT+P** : 翻出上一條命令, 類似於向上的箭頭
- **ALT+N** : 翻出下一條命令, 類似於向下的箭頭
- **CTRL+[**、**CTRL+]** 多行的代碼的縮排
- **CTRL+F**: 查找指定的字符串
- **ALT+F4**: 關閉Windows窗口
- **ATL+DD**: 開啟Code debug功能
- **ALT+M** : 打開模塊代碼, 先選中模塊, 就可以查看該模塊的源碼
- **ALT+X** : 進入Python Shell模式
- **ALT+F+P**: 打開路徑瀏覽器, 方便選擇導入包進行查看、瀏覽
- **ALT+C** : 打開類瀏覽器, 方便在模塊方法體之間的切換
- **F5** : 進入Python Shell 執行
- **TAB** : Python Shell模式下的自動補齊功能



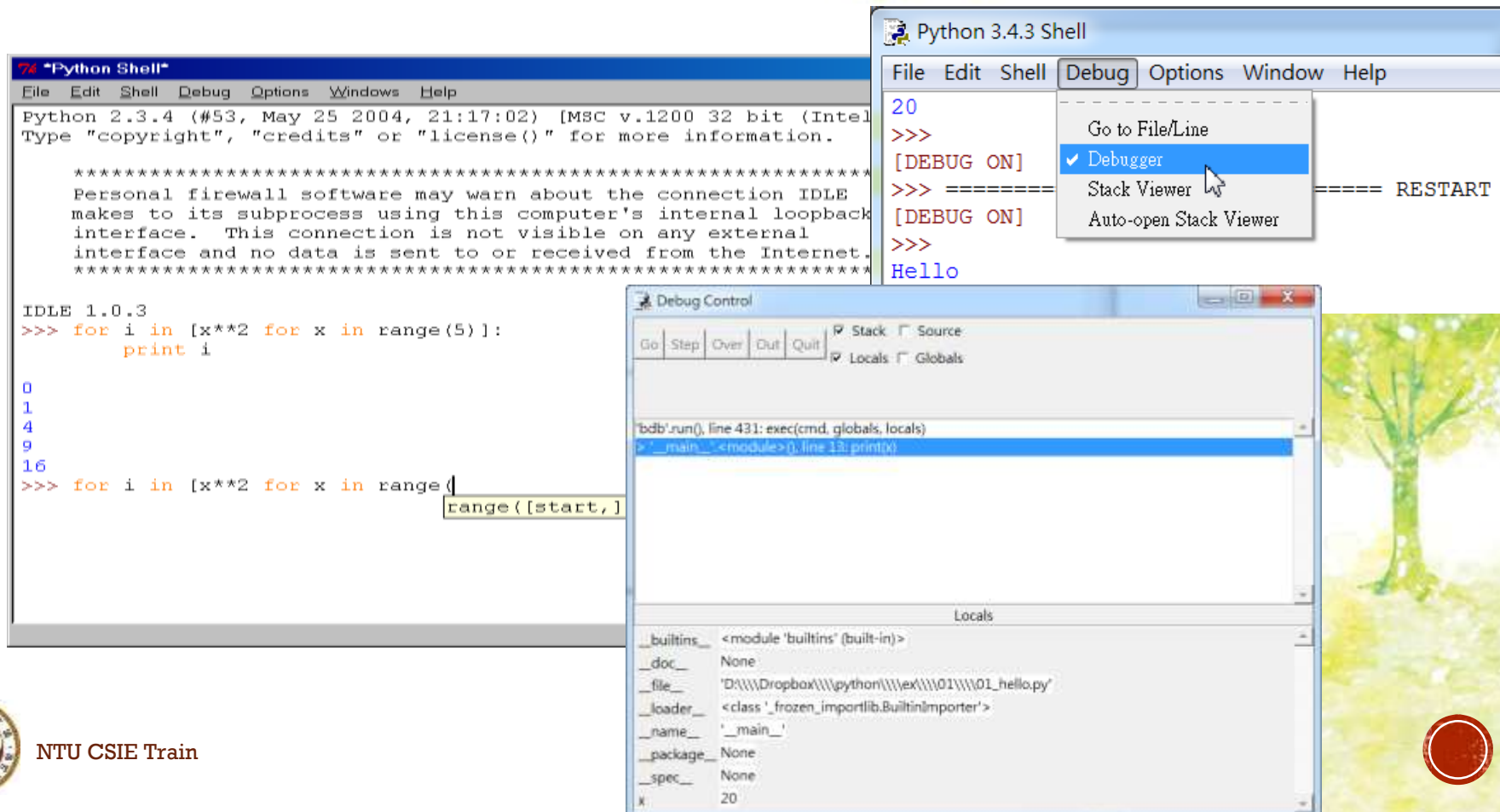
IDLE常用快捷

- F1 : Python document 說明
- Alt+5 Alt+6 切換縮排方式 空格<=>Tab
- Alt+/ 單詞完成，只要文中出現過，就可以幫你自動補齊。多按選擇
- Alt+DG 先定位到錯誤行，按此快速鍵可以快速定位到出錯位置
- Alt+DS 直接顯示出錯歷史，找到根源，方便啊
- Alt+DA 如果每次都要按，還不夠方便,按這個，以後出錯都出歷史
- Alt+DD 打開調試視窗，進入單步調試，方便。
- Ctrl+F6 為了清空前面的導入記錄等，重新開機shell
- Ctrl+方向鍵 能夠得到增強功能



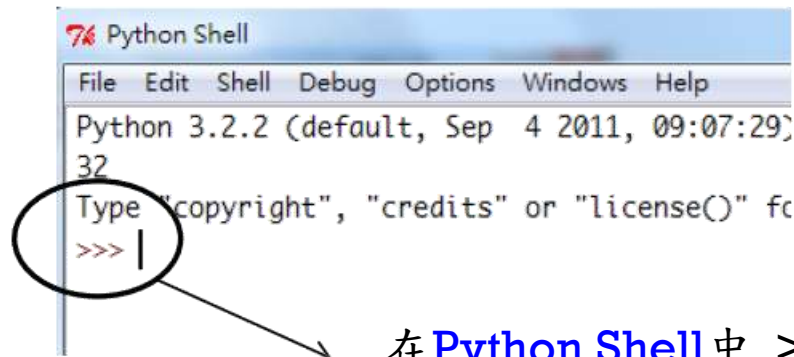
IDLE 開發環境

- IDLE 是官方的“官方的” IDE 環境
- 整合了 除錯器(debugger)與syntax highlighting



PYTHON SHELL

- 當開啟IDLE 之後，可以直接編輯並儲存為一個新的.py 檔，然而我比較偏好使用【New File】開啟一個新的視窗來編輯，因為如果直接在IDLE上編輯，每次按下【Enter】鍵便會執行該行命令。



在Python Shell中,>>> 是一個輸入的提示
所有要下給直譯器的命令都要打在>>>之後

- 而這種每按一次【Enter】就會執行該行命令的方式叫「指令互動環境」，通常較適合拿來測試新學習的指令效果，但若已經熟悉該指令者，反而會容易中斷程式撰寫的思緒。



PYTHON使用方式

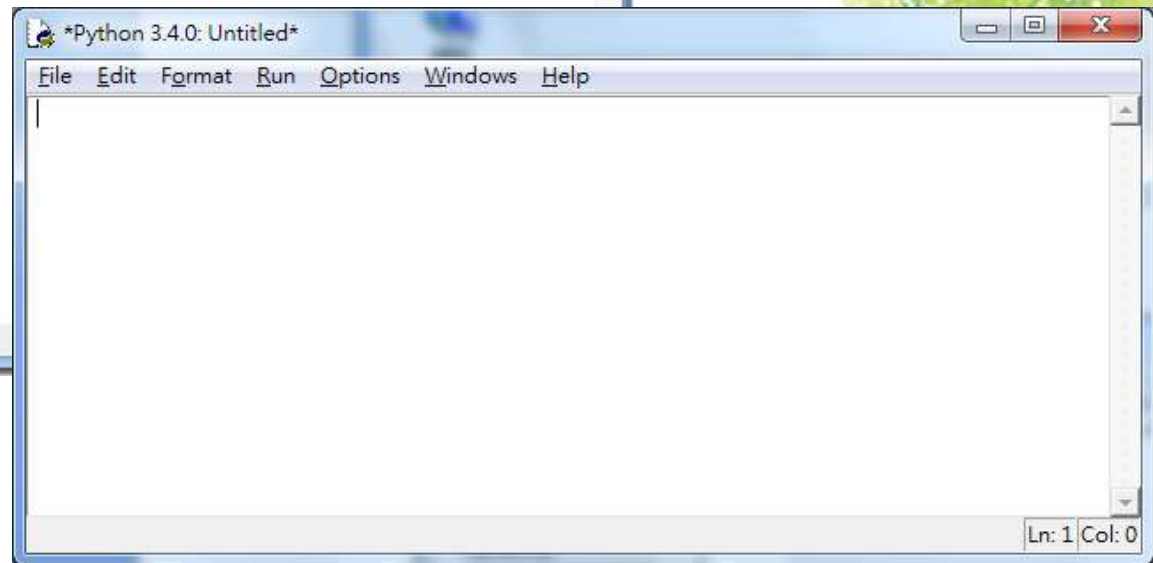
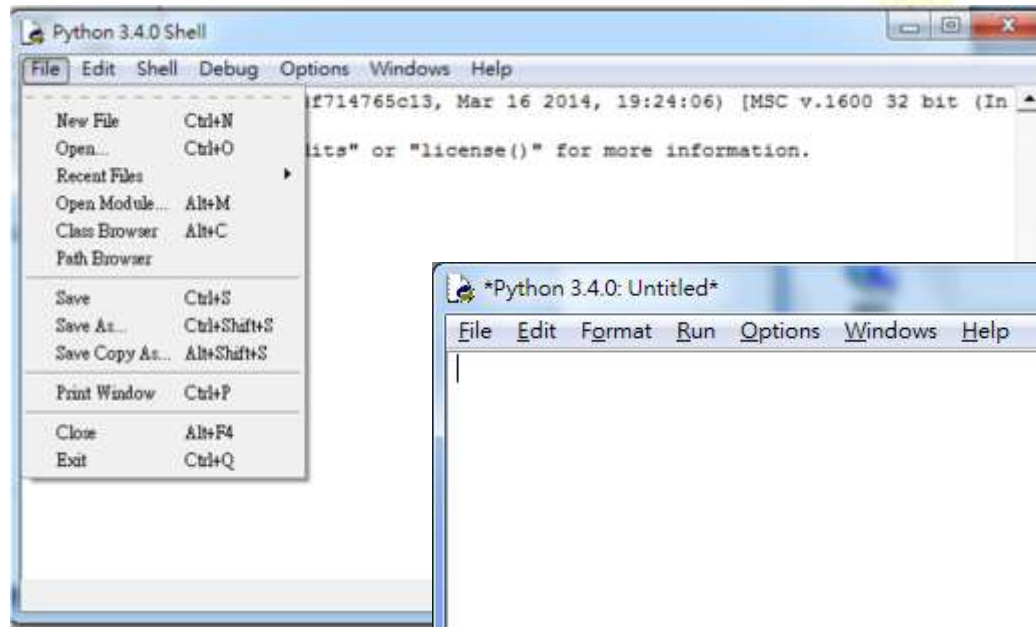
- 在開啟新檔案之後，便會出現全白的視窗，這與「記事本」相當相似。
- 在編輯完程式碼後，要執行並觀看結果，可以先儲存為.py檔之後，到剛儲存檔案的地方開啟或依照上面所教的方式，使用cmd 命令視窗來執行；
- 不過，如果沒有輸入暫停指令，又沒有透過cmd 命令視窗執行.py 檔，在一般的視窗環境下，只要執行完就會直接關閉，便沒有辦法看到執行結果，而且每次都透過cmd 命令視窗執行Python 也相當麻煩。



PYTHON使用方式

■ 實際操作IDLE：

- ◆ 自動縮排
- ◆ 語法高亮
- ◆ 提供互動介面
- ◆ 容易學習與測試



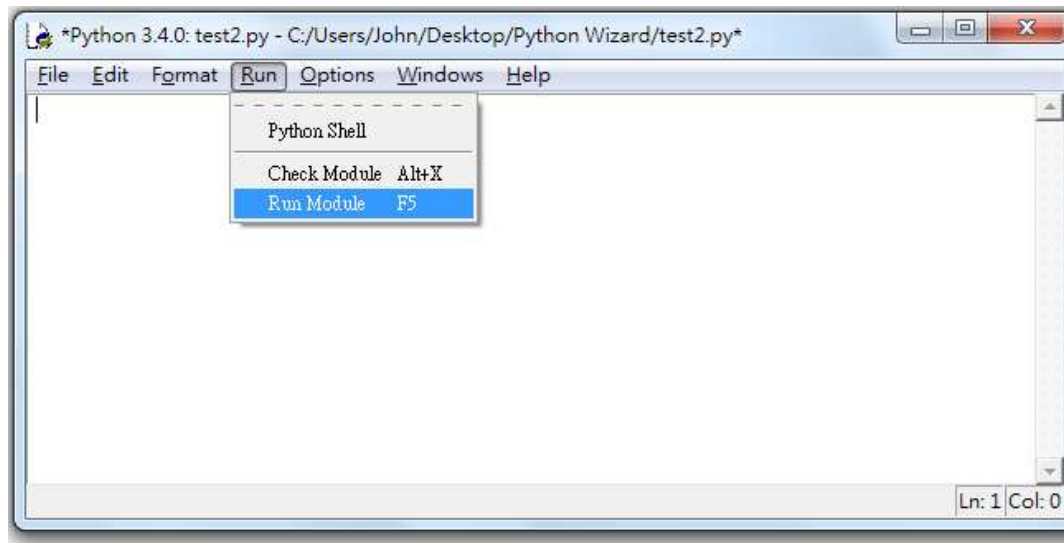
PYTHON使用方式

- 比起使用一般的文字編輯器來撰寫Python 的程式碼，**IDLE** 使用起來更為方便，功能也完整許多。
- 本課程的範例一律使用Python 3.4.0 版本的IDLE 撰寫。
 - 使用IDLE 的方式就跟記事本的方式類似，不過要注意的是，如果使用的是2.x 版的Python，在儲存時需要手動選擇檔案格式為.py，如此Python 才能辨別出這個檔案是否可以被執行。
 - 如果跟本書一樣是使用3.x 的版本，那麼在儲存時就不用特別選擇.py 的檔案格式，因為預設就是如此了。



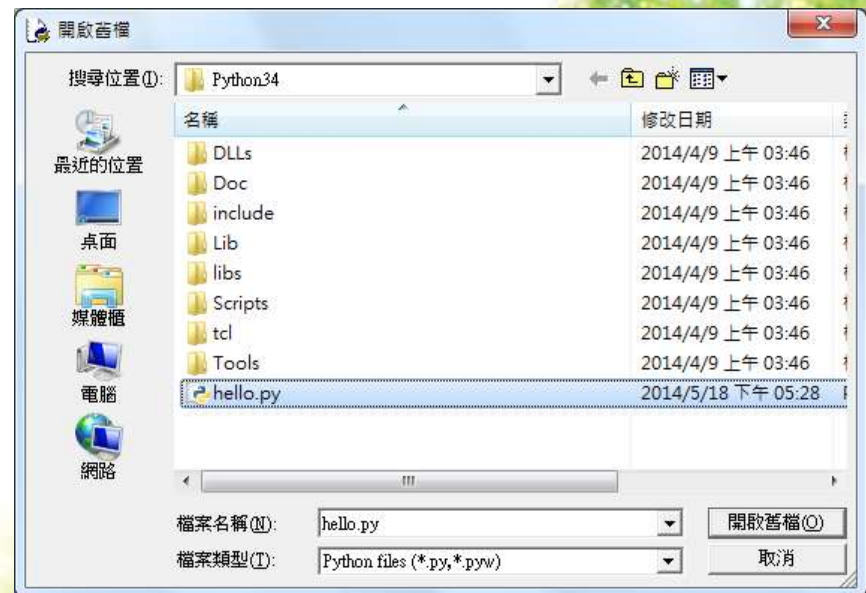
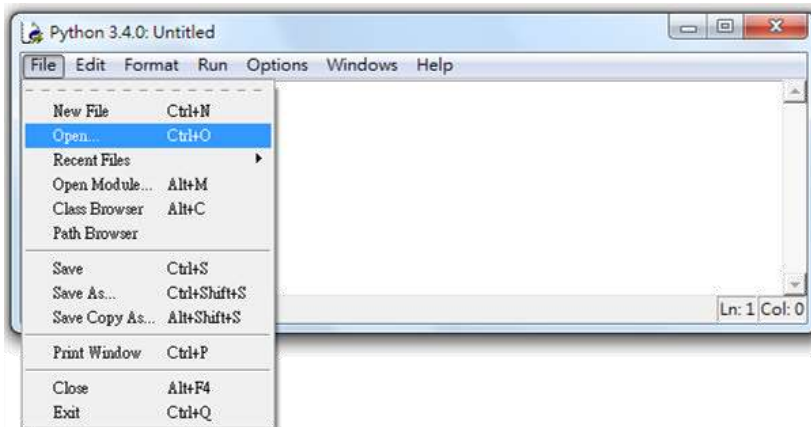
PYTHON使用方式

- 這裡建議使用【Run】之【Run Module】方式，也就是回到IDLE的主頁面Shell命令視窗來執行。若想要節省時間，也可以使用【F5】快捷鍵來執行。

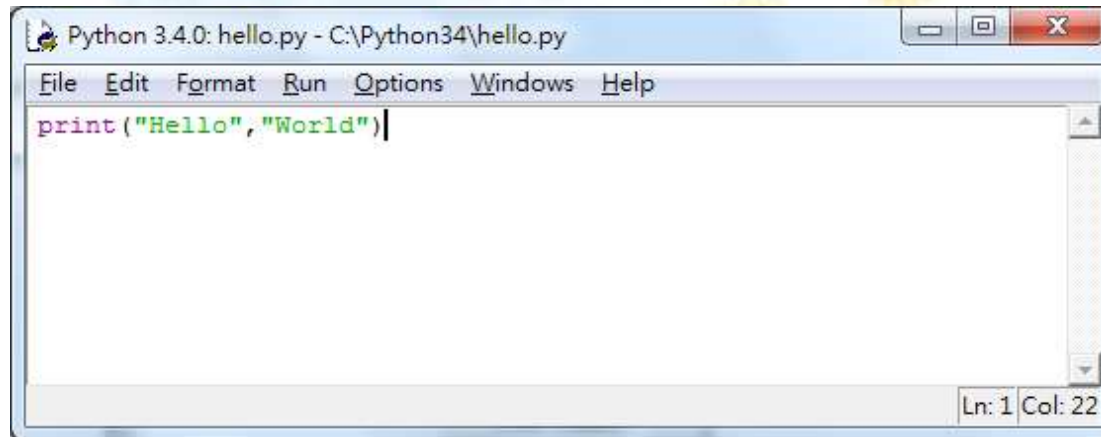


PYTHON使用方式

- 如果你想繼續編輯之前已經編輯且存檔過的檔案，可以使用【File】之【Open...】的指令來開啟並繼續編輯。
- 在這裡再次開啟剛剛存檔的hello.py，而路徑是剛剛系統安裝時所設定的「C:\Python34」，並且使用上面教的方式在IDLE執行。

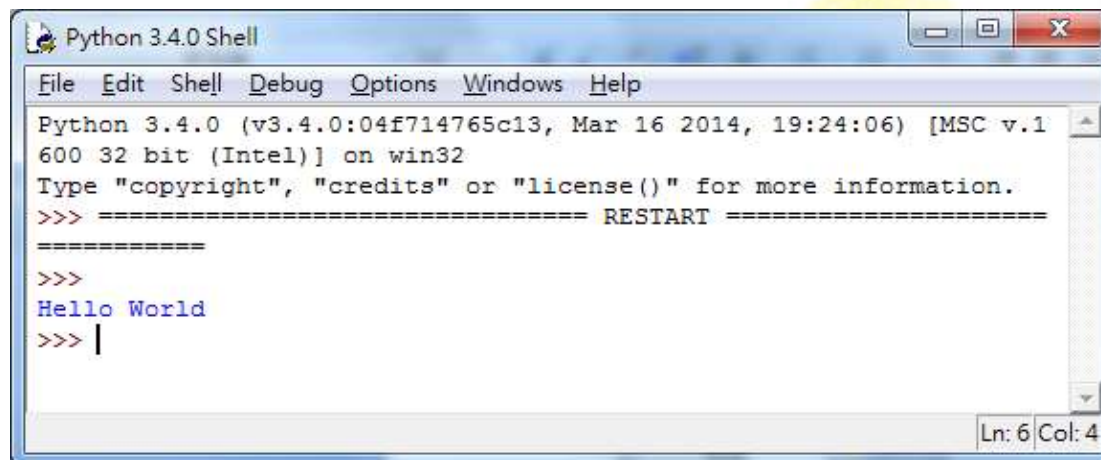


PYTHON使用方式



A screenshot of a Python 3.4.0 IDE window titled "Python 3.4.0: hello.py - C:\Python34\hello.py". The window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Windows", and "Help". The main text area contains the code `print("Hello", "World")`. The status bar at the bottom right shows "Ln: 1 Col: 22".

```
Python 3.4.0: hello.py - C:\Python34\hello.py
File Edit Format Run Options Windows Help
print("Hello", "World")
Ln: 1 Col: 22
```



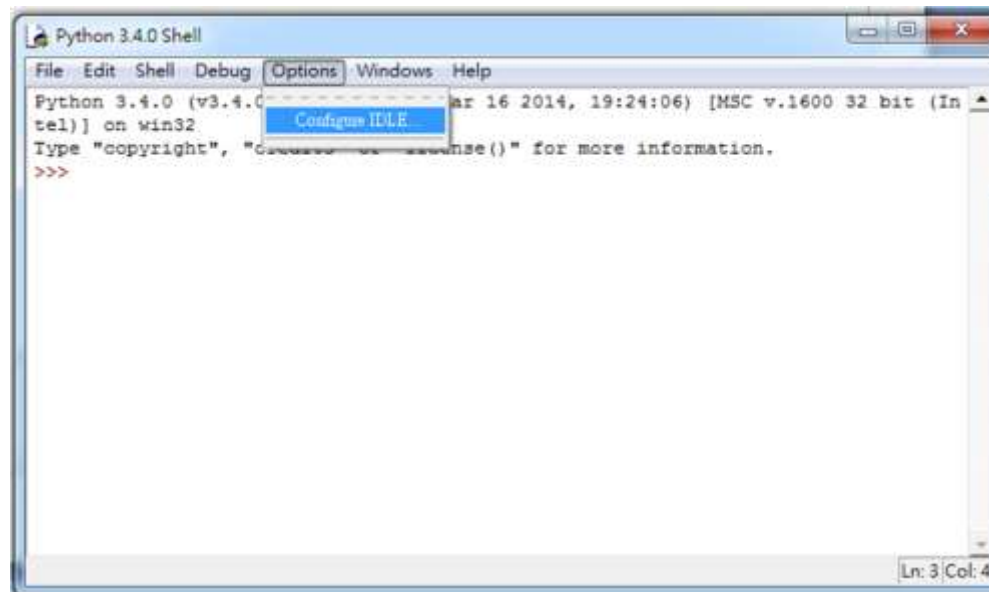
A screenshot of a Python 3.4.0 Shell window titled "Python 3.4.0 Shell". The window has a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main text area shows the execution output: "Python 3.4.0 (v3.4.0:04f714765c13, Mar 16 2014, 19:24:06) [MSC v.1600 32 bit (Intel)] on win32", "Type 'copyright', 'credits' or 'license()' for more information.", a separator line, and the output "Hello World". The prompt is ">>>". The status bar at the bottom right shows "Ln: 6 Col: 4".

```
Python 3.4.0 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.0 (v3.4.0:04f714765c13, Mar 16 2014, 19:24:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Hello World
>>> |
Ln: 6 Col: 4
```

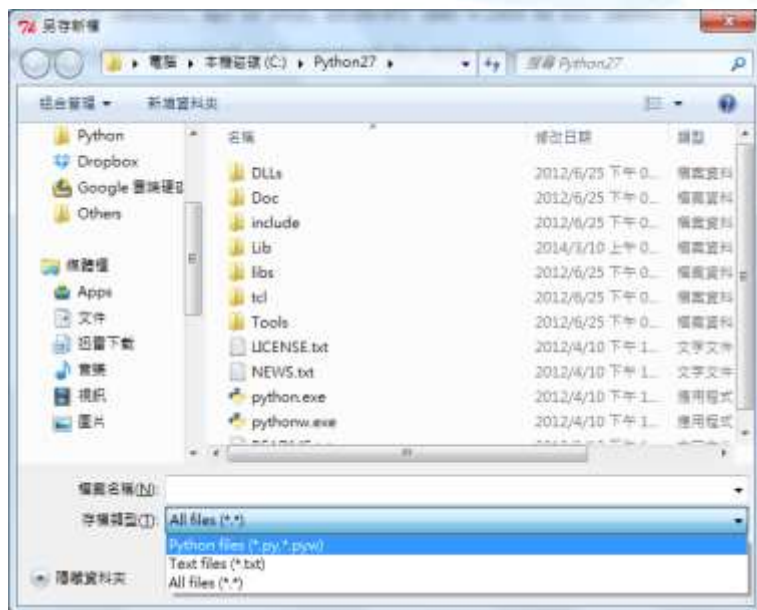


PYTHON使用方式

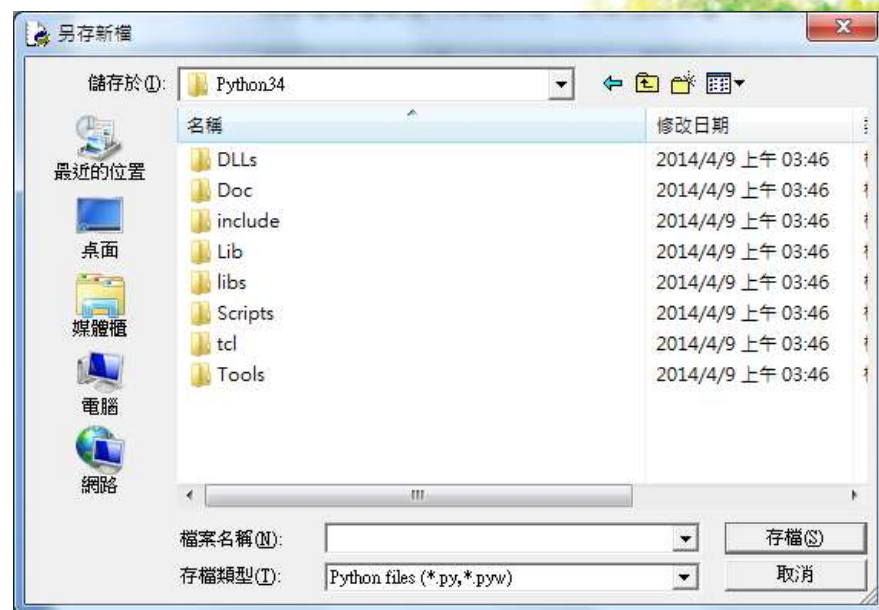
- 如果在編寫Python的過程中，不滿意於IDLE所預設的文字大小、顏色、縮排長度等，可以透過【Option】之【Configure IDLE...】功能來更改設定。



PYTHON使用方式



← Python2.7 儲存時須選擇.py 檔案格式



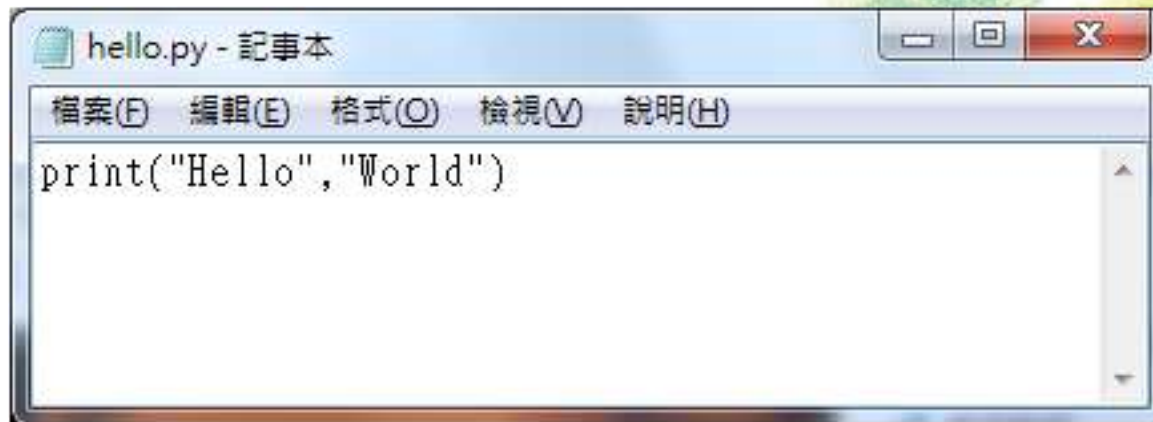
Python 3.x 的版本預設就是.py 的格式 ↑



PYTHON使用方式

- 從Windows 的操控台（又被稱為命令提示字元或cmd）來啟動Python
- 首先，打開【記事本】應用程式，在【記事本】中鍵入以下文字：

```
print("Hello", "World")
```
- 這行Python 程式碼所代表的功能是将圓括號裡面的東西印在螢幕上



PYTHON使用方式

- 然後，將這個檔案存檔，檔名取為**hello.py**，並放在一個容易找尋的位置。
- 在這裡為求統一方便說明，以剛剛安裝Python 的檔案路徑（預設為**C:\Python34**）為預設儲存的目錄，這樣準備的工作就完成了。



第一個小練習

- 請撰寫一個可以獨立執行的**hello.py**檔
- 使其可以在螢幕上列印出下列指定文字
- ※注意文字內容要一模一樣(包含大小寫與空白都需相同)

Hello World!

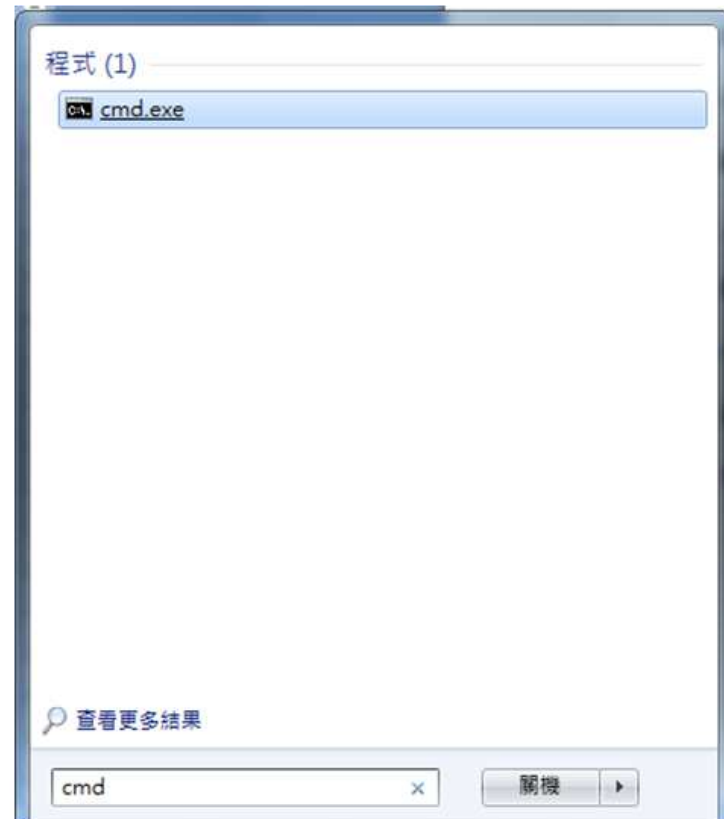
This is my first program.

- 並練習提交到批改系統



PYTHON使用方式

- 如果您是Windows 7 的用戶，請從【開始】按鈕下之【搜尋程式及檔案】欄位中鍵入cmd，則會出現【cmd.exe】，最後再點選兩下以開啟畫面。



PYTHON使用方式

- 開啟cmd 之後應該會出現這樣的畫面：



PYTHON使用方式

- 假設這台的電腦帳戶名稱是**user**，所以一開始的路徑會是「**C:\Users\使用者名稱（在此為user）**」。此時請輸入以下指令：
- **cd** 是更改目前目錄位置（**Change Directory**）的意思，也就是讓電腦把目標從目前的資料夾移到**c:\Python34**這個資料夾，如果磁碟槽不同須先切換磁碟槽。

```
cd c:\Python34
```

```
C:\Users\user> D:
```

```
C:\Users\user> cd D:\exc\
```

```
D:\exc\>
```



PYTHON使用方式

- 然後再輸入：
- 前面的python 是跟電腦說要執行python.exe 這支程式，後面的hello.py是剛剛存檔的，也就是準備給python.exe 這支程式開啟的檔案。

```
python hello.py
```

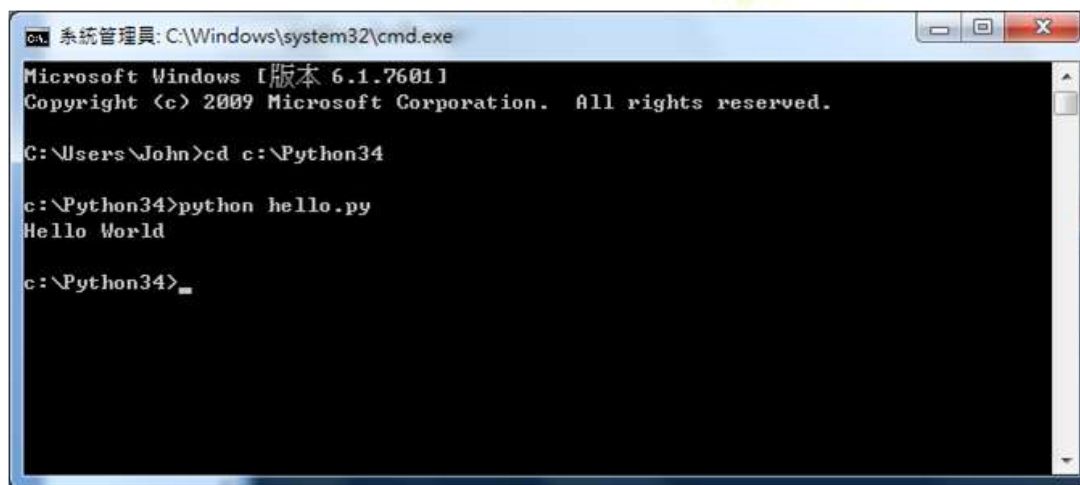
- Linux / MAC OS X :

```
python3 hello.py
```



PYTHON使用方式

- 執行後的結果會像：



```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\John>cd c:\Python34

c:\Python34>python hello.py
Hello World

c:\Python34>
```

- cmd 剛剛輸入的文字「`print("Hello", "World")`」在螢幕上印出了Hello World，以上就是用cmd 命令視窗執行Python 程式的方法。



簡介

- 結束Python(in command line)
 - Linux
 - Ctrl+D
 - Windows
 - Ctrl+Z
 - 然後按 Enter
 - 通用
 - `import sys; sys.exit()`



```
*Python Shell*
File Edit Shell Debug Options Windows Help
Python 2.6.4 (r264:75708, Oct 26 2009, 08:23:19) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.

*****
Personal firewall software may warn about the connection IDLE
makes to its subprocess using this computer's internal loopback
interface. This connection is not visible on any external
interface and no data is sent to or received from the Internet.
*****

IDLE 2.6.4
>>> from __future__ import print_function
>>> print
```

```
Python (command line)
Python 2.6.4 (r264:75708, Oct 26 2009, 08:23:19) [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print "ABC"
ABC
>>> ^Z_
```



PYTHON使用方式

- 在編輯、學習Python的過程中，如果有不了解任何一個函數或者物件（例如前面所提的`print()`、或者資料型態6的`int`、`str`等等），可以在IDLE的指令互動環境裡面鍵入「`help()`」，然後再輸入想查詢的東西。
- Python有內建英文說明，相信對學習Python會有所幫助。底下以`print()`為查詢範例：



PYTHON使用方式

```
*Python 3.4.0 Shell*
File Edit Shell Debug Options Windows Help
Python 3.4.0 (v3.4.0:04f714765c13, Mar 16 2014, 19:24:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> help()

Welcome to Python 3.4! This is the interactive help utility.

If this is your first time using Python, you should definitely check out
the tutorial on the Internet at http://docs.python.org/3.4/tutorial/.
```

Enter the name of any module, Python programs and using Python return to the interpreter, just

To get a list of available modules, keywords, symbols, or topics, type "modules", "keywords", "symbols", or "topics". Each module also comes with a one-line summary of what it does; to list the modules whose name or summary contain a given string

help> |

```
*Python 3.4.0 Shell*
File Edit Shell Debug Options Windows Help
To get a list of available modules, keywords, symbols, or topics, type
"modules", "keywords", "symbols", or "topics". Each module also comes
with a one-line summary of what it does; to list the modules whose name
or summary contain a given string such as "spam", type "modules spam".

help> print
Help on built-in function print in module builtins:

print(...)
    print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

    Prints the values to a stream, or to sys.stdout by default.
    Optional keyword arguments:
    file: a file-like object (stream); defaults to the current sys.stdout.
    sep: string inserted between values, default a space.
    end: string appended after the last value, default a newline.
    flush: whether to forcibly flush the stream.

help> |
```

Ln: 32 Col: 6



PYTHON 2.X與PYTHON 3.X的差異

Python2.X

- ① 穩定版本
- ② 較多的第三方函式庫
- ③ 仍有較多的老派使用者

Python3.X

- ① 官方推薦也是未來趨勢
- ② 火熱發展中
- ③ 對初學者上手較容易
- ④ 老舊第三方函式庫不支援

■ **print**從陳述式(statement)改為函數(function)

■ **ex:**

- `print "123?", 10**2`
 - 123? 100
- `print("123?", 10**2)`
 - 123? 100

■ 若想让2.x版可以使用**print()**函數
則加入以下指令：

```
from __future__ import print_function
```



您的第一個PYTHON程式 – HELLO WORLD

- 使用互動式命令列 (**shell**)

```
>>> print ("Hello World")
```

```
Hello World
```

```
>>>
```

- 放在**檔案**裡

python 路徑與檔名.py

- 使用**python**編輯器
- 也可使用**notepad++**或其它**IDE**

hello.py

```
1 print("Hello, world!")
```



其它

- 解決 mac 無法輸入中文的問題
- <https://www.youtube.com/watch?v=jo9XAs6Vsuw>

