

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。

架設網站的方式有許多種，最簡單的方式就是把檔案放在網路主機上，當有網友透過瀏覽器連線進來的時候，網路伺服器把這個檔案提供給瀏覽器加以解析再呈現給瀏覽者，如圖 1-1 所示。

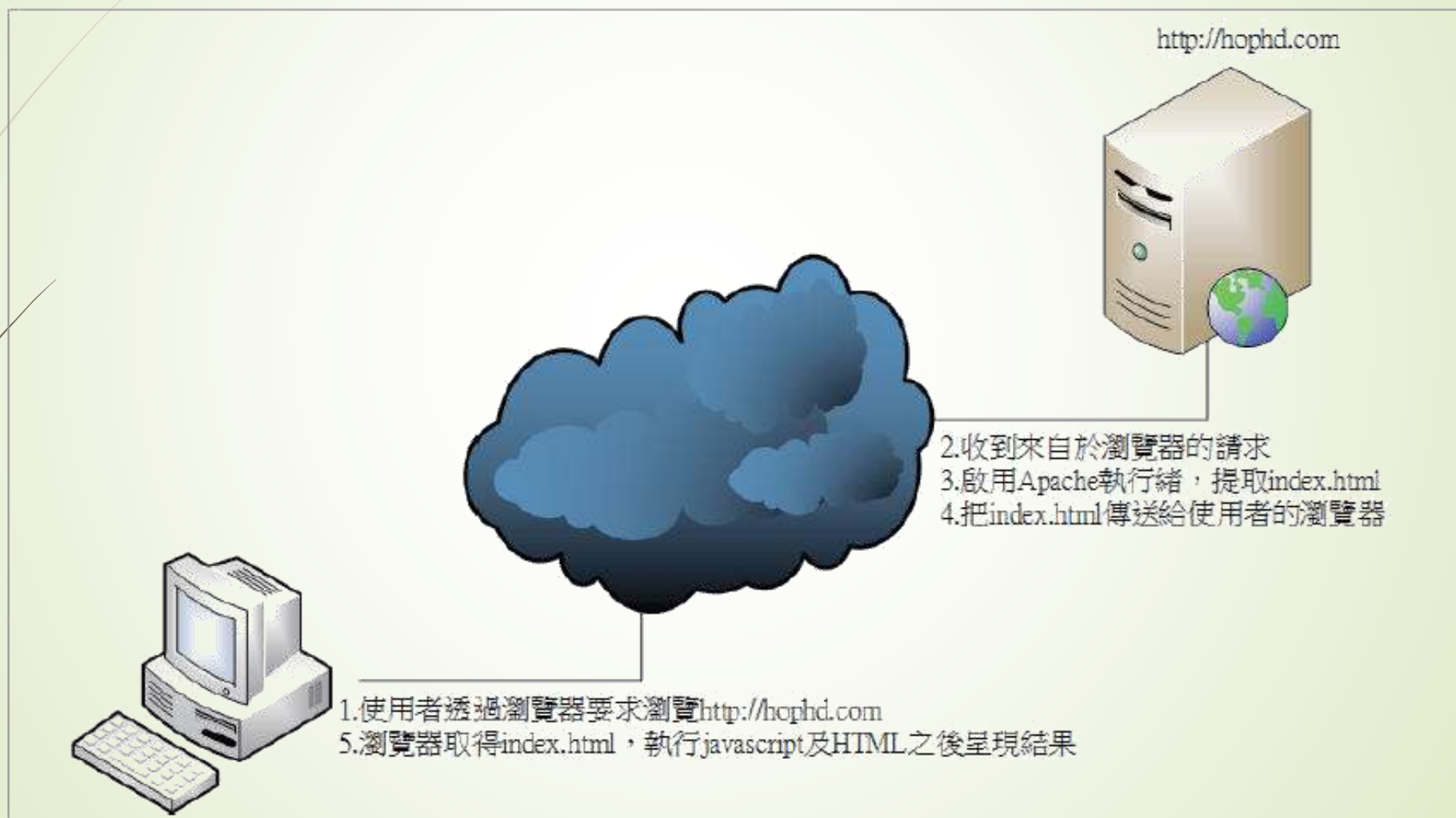


圖 1-1：使用者瀏覽網站的基本流程


前者所說明的網站，稱為靜態網站。所有的檔案以及資料都是不會因應使用者的需求而臨時產生，而是事先已準備好的。如果需要因應不同的存取而產生不同的資料，甚至在呈現資料之前還要從資料庫或是其他的資料來源取得資料加以整合、計算、分析之後再呈現給瀏覽者，則有一些工作必需要在網站主機上先執行過，這就是所謂動態網站的概念，如圖 1-2 所示。



1.1.2 Python/Django 扮演的角色

從前一節的介紹可以瞭解，當瀏覽者輸入一個網址的時候，網站主機的軟體伺服器（以下都以 Apache 為例）在收到此請求之後，會先依照設定檔的內容，決定要把此網址請求交給那哪一個檔案處理。如果沒有特別的設定，一般都會以執行 PHP 為主。也就是說，如果想要以別的程式語言來執行網頁的請求，就必須做好 Apache 的設定，在設定完畢之後，理論上所有可以在伺服器上執行的程式，都可以當作是後端的程式語言。只是在開始處理瀏覽器的請求時，這些被呼叫到的程式也要自行處理 HTTP（Hyper Text Transfer Protocol，WWW 網站的專用通訊協定）的相關細節，對於沒有相關模組可用的程式語言來說，其實是非常麻煩的。

所幸，這些常被用來處理網站的語言如 Python、Ruby、Perl 等，已有許多的相關模組可以使用，在設定這些網站的時候就容易多了。而更進一步地，因應許多現代網站的必備功能（如使用者帳號管理、表單輸入、網頁輸出樣式、網站和資料庫連結等等），甚至出現了所謂的 Web Framework（網站框架），讓使用這些程式語言的網站開發者，可以使用一些簡單的命令即可產出一個網站的基本架構，然後依循其框架的設計，有系統地結構化設計出正式的、可商用的多功能網站。其中，Ruby 語言最有名氣的當屬 Ruby on Rails，而 Python 語言則以 Flask 和 Django 為大宗。



Django 是為了網站開發人員所設計的以 Python 語言所編寫網站框架，簡單地說，它就是可以協助程式設計師迅速建立全功能網站的一組 Python 程式，透過 MVC（Model View Controller）觀念把視圖和控制邏輯分割開來，讓程式設計師可以儘量不用擔心網站通訊協定的瑣碎細節，而專心在想要建立網站的功能上。此組程式放在主機的某一個特定的資料夾之下，透過對於 Apache 的設定檔指定此組程式所在的位置，當有網頁存取時執行並傳回執行後的結果給 Apache，再傳送到使用者的瀏覽器中。

也就是說，每當主機接收到來自瀏覽器的連結請求時，Django 中的某一個程式檔案就可以得到被執行的機會，我們就可以在這個程式檔案中以 Python 語言來編寫需要處理、運算、存取資料庫等等程式碼，讓網頁的請求可以更加地客製化，即時回應使用者的需求，提供更多網站的服務。

因此，對於想要利用 Python 來建立網站服務的程式設計師來說，只要學會了 Django 的架構內容以及運作原理，就可以充份運用 Python 在處理字串、資料庫、影像繪圖、商業統計、科學運算等等拿手的功能，處理網站相關細節，並提供更多網站服務功能。

1.1.3 使用 Python/Django 建立網站的優勢

如前一小節所述，因為 Django 的框架可以省去處理通訊協定的相關細節，把精力都放在網站相關的服務設計上，而且 Django 本身就提供了一個網站所需要的程式碼，只要依照這些流程編寫程式，可以輕易地完成許多原本非常複雜的事情。

另外，由於 Django 在設計的時候均有遵循模組化的設計概念，又把資料庫和 Python 的連結做了抽象化的設計，以使用者資料庫為主的模型化技巧，讓一些外部的第三方網站功能模組也可以輕易地加入我們的網站，無形中讓擴充網站能力的方式更加地容易。而由於資料庫的抽象化，所以在網站的設計中基本上不需使用到 SQL 查詢語言，而是使用 Python 的方式來處理資料庫中的資料，日後如果需要更換資料庫種類，也不至於要大量修改網站的程式內容。

因此，對於使用 Python 來架網站的初學者來說，一旦熟悉的 Django 的運作邏輯，可以在非常短的時間內建構出一個出色的專業網站。

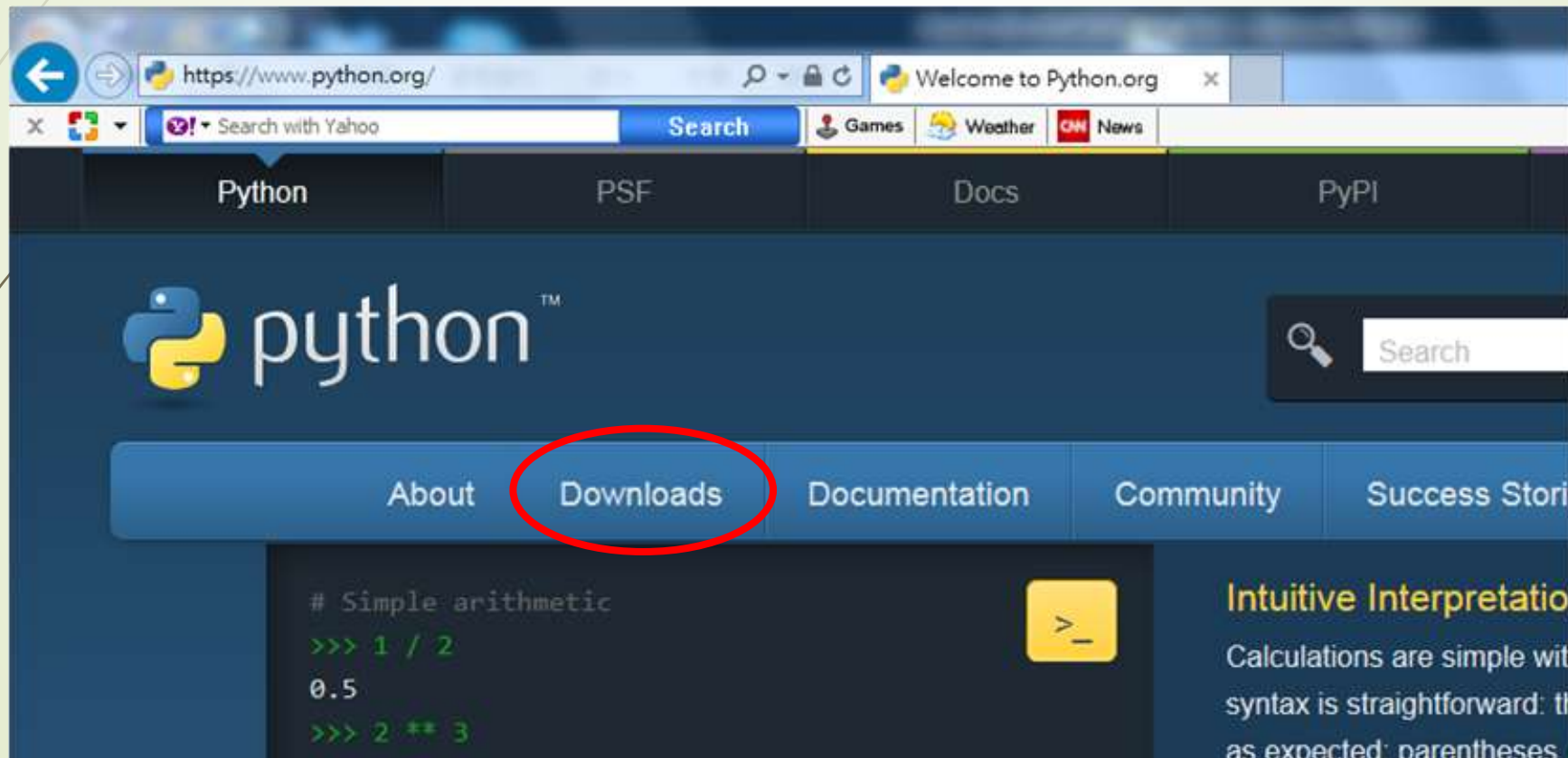


為了避免在不同的電腦間開發網站的問題，作者建議的開發環境建議如下：

- 在每一台電腦中新增 Linux 虛擬機，並建立相同的開發環境
- 以 virtualenv 來設置 Python 的虛擬環境
- 建立一個遠端倉庫
- 使用 Git 分散式版本控制在本地端建立版本控管的倉庫
- 隨時保持本地和遠端倉庫的同步

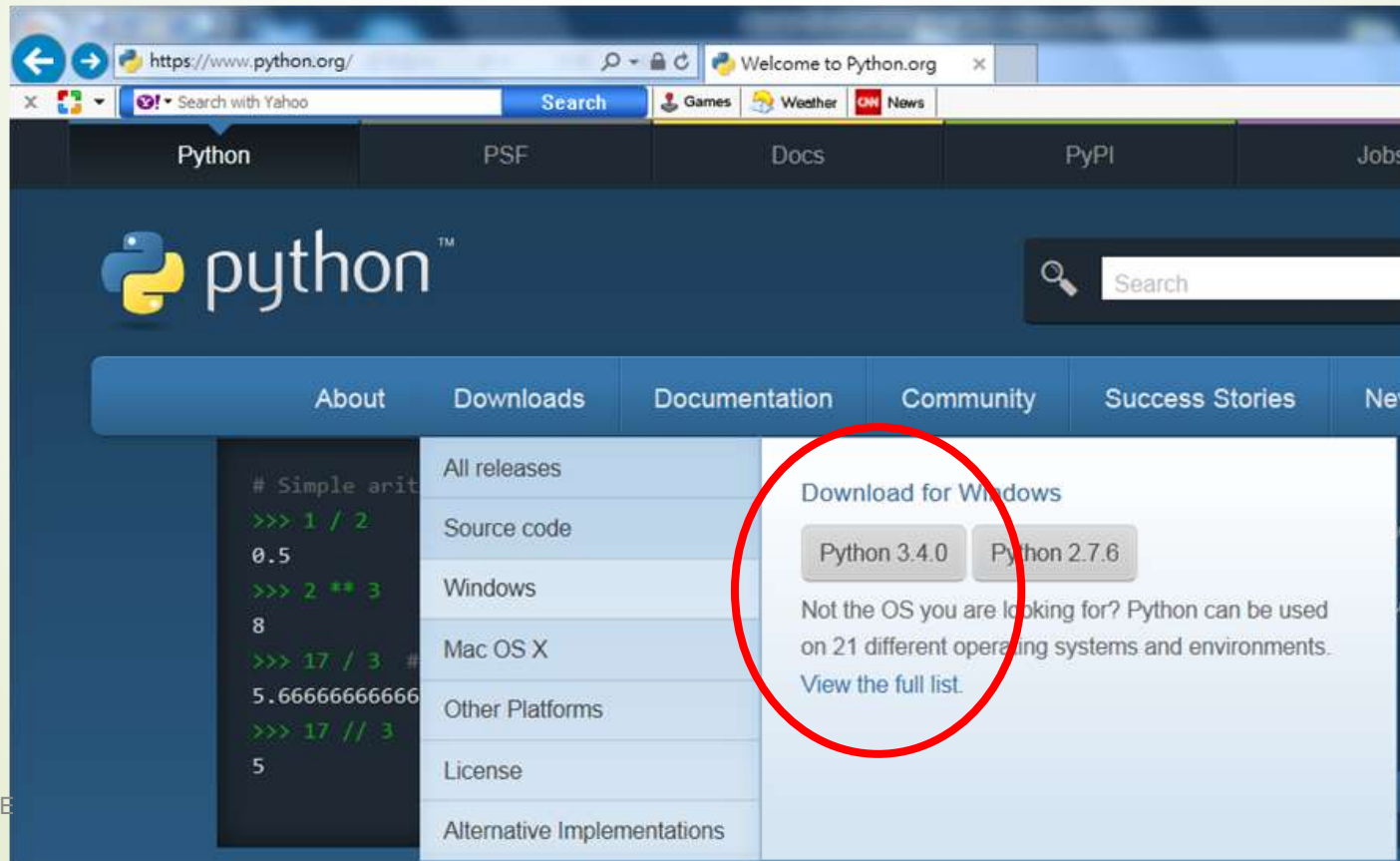
安裝Python

- 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。



Python Software Foundation [US] | https://www.python.org

python

Search GO Socialize

About Downloads Documentation Community Success Stories News Events

```
# Python 3:
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while n > 0:
>>>         yield a
>>>         a, b = b, a + b
>>>     print(list(fib(n)))
>>> fib(1000)
0 1 1 2 3 5
987
```

All releases
Source code
Windows
Mac OS X
Other Platforms
License
Alternative Implementations

Download for Windows

Python 3.7.2

Note that Python 3.5+ cannot be used on Windows XP or earlier.

Not the OS you are looking for? Python can be used on many operating systems and environments.

[View the full list of downloads.](#)

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)

Get Started
Whether you're new to programming or an

Download
Python source code and installers are available for

Docs
Documentation for Python's standard library, along with

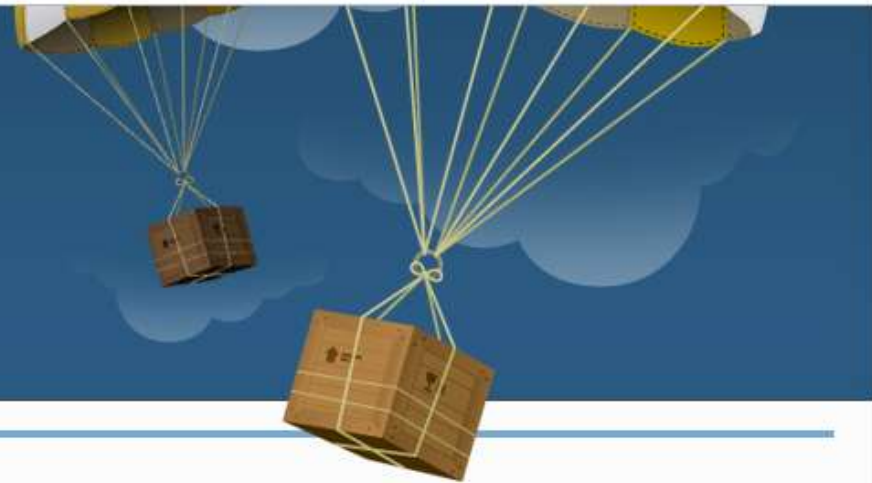
Jobs
Looking for work or have a Python related position that

Download Python 3.7.2

Looking for Python with a different OS? Python for [Windows](#),
[Linux/UNIX](#), [Mac OS X](#), [Other](#)

Want to help test development versions of Python? [Pre-releases](#)

Looking for Python 2.7? See below for specific releases



Looking for a specific release?

Python releases by version number:

Release version	Release date	Click for more	
Python 3.7.2	2018-12-24	Download	Release Notes
Python 3.6.8	2018-12-24	Download	Release Notes
Python 3.7.1	2018-10-20	Download	Release Notes
Python 3.6.7	2018-10-20	Download	Release Notes
Python 3.5.6	2018-08-02	Download	Release Notes
Python 3.4.9	2018-08-02	Download	Release Notes
Python 3.7.0	2018-06-27	Download	Release Notes
Python 3.6.6	2018-06-27	Download	Release Notes

[View older releases](#)

INSTALLATION FOR INFORMATION ABOUT SSL/TLS CERTIFICATE VALIDATION AND THE `install --certificates` command.

[Full Changelog](#)

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		48f393a04c2e66c77bfc114e589ec630	23010188	SIG
XZ compressed source tarball	Source release		51aac91bdf8be95ec0a62d174890821a	17212420	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	eb1a23d762946329c2aa3448d256d421	33258809	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	786c4d9183c754f58751d52f509bc971	27073838	SIG
Windows help file	Windows		0b04278f5bdb8ee85ae5ae66af0430b2	7868305	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	73df7cb2f1500ff36d7dbeac3968711	7276004	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	72f37686b7ab240ef70fdb931bdf3cb5	31830944	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	39dde5f535c16d642e84fc7a69f43e05	1331744	SIG
Windows x86 embeddable zip file	Windows		60470b4cceba52094121d43cd3f6ce3a	6560373	SIG
Windows x86 executable installer	Windows		9c7b1ebdd3a8df0eebfda2f107f1742c	30807656	SIG
Windows x86 web-based installer	Windows		80de96338691698e10a935ecd0bdaacb	1296064	SIG

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。



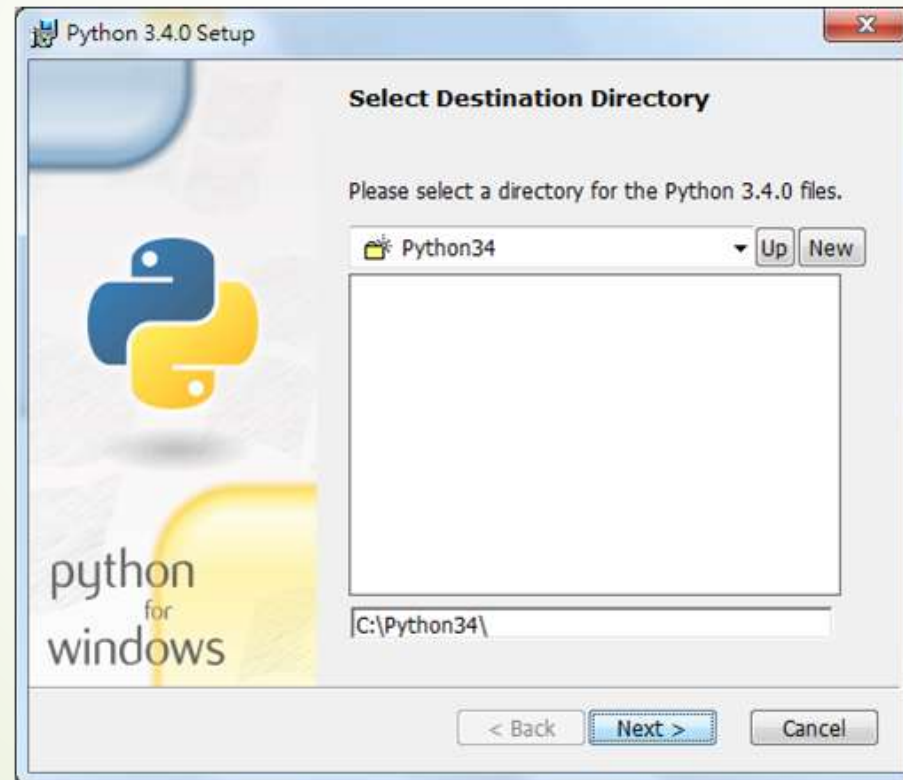
安裝Python

- 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

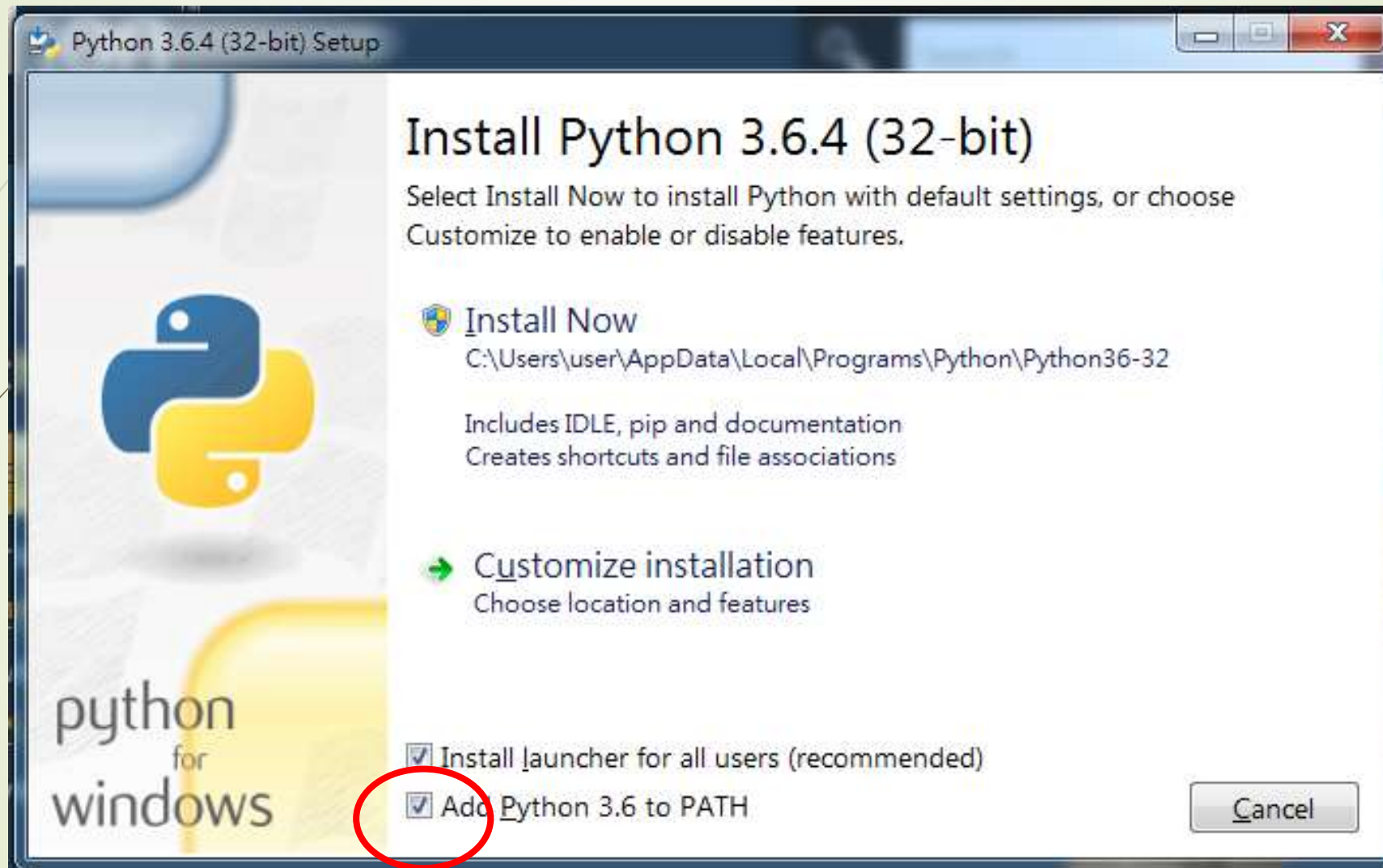
- Step 6：選擇安裝路徑，此處以預設【C:\Python34\】為例。



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。





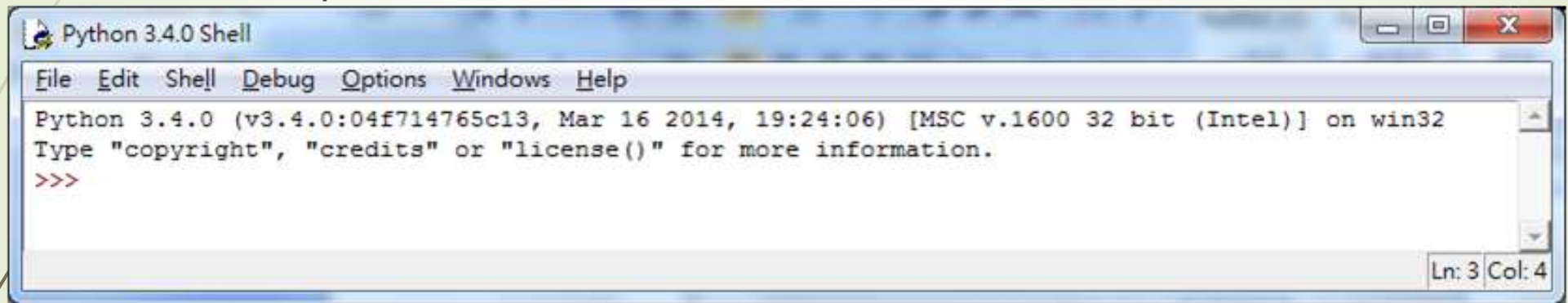
安裝Python

- 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。

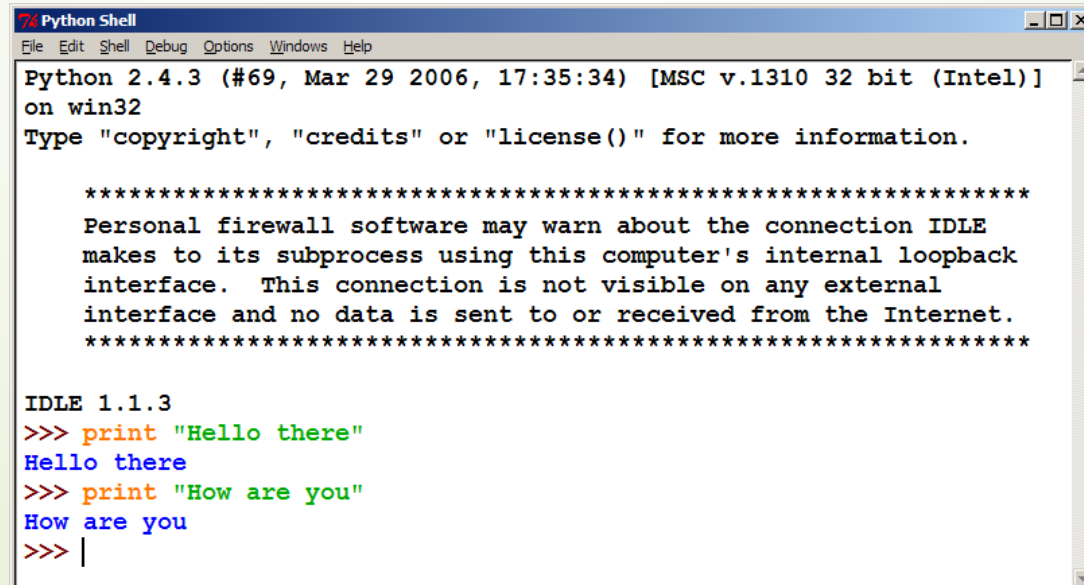


```
Python 3.4.0 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.0 (v3.4.0:04f714765c13, Mar 16 2014, 19:24:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
```

- ◆ 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**



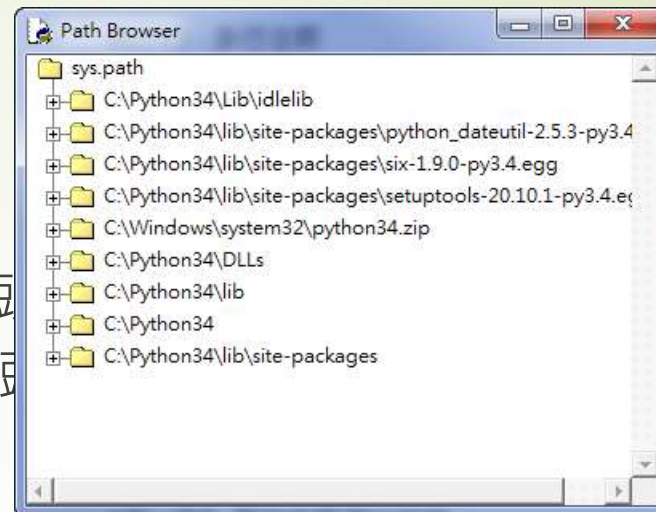
```
Python Shell
File Edit Shell Debug Options Windows Help
Python 2.4.3 (#69, Mar 29 2006, 17:35:34) [MSC v.1310 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.

*****
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

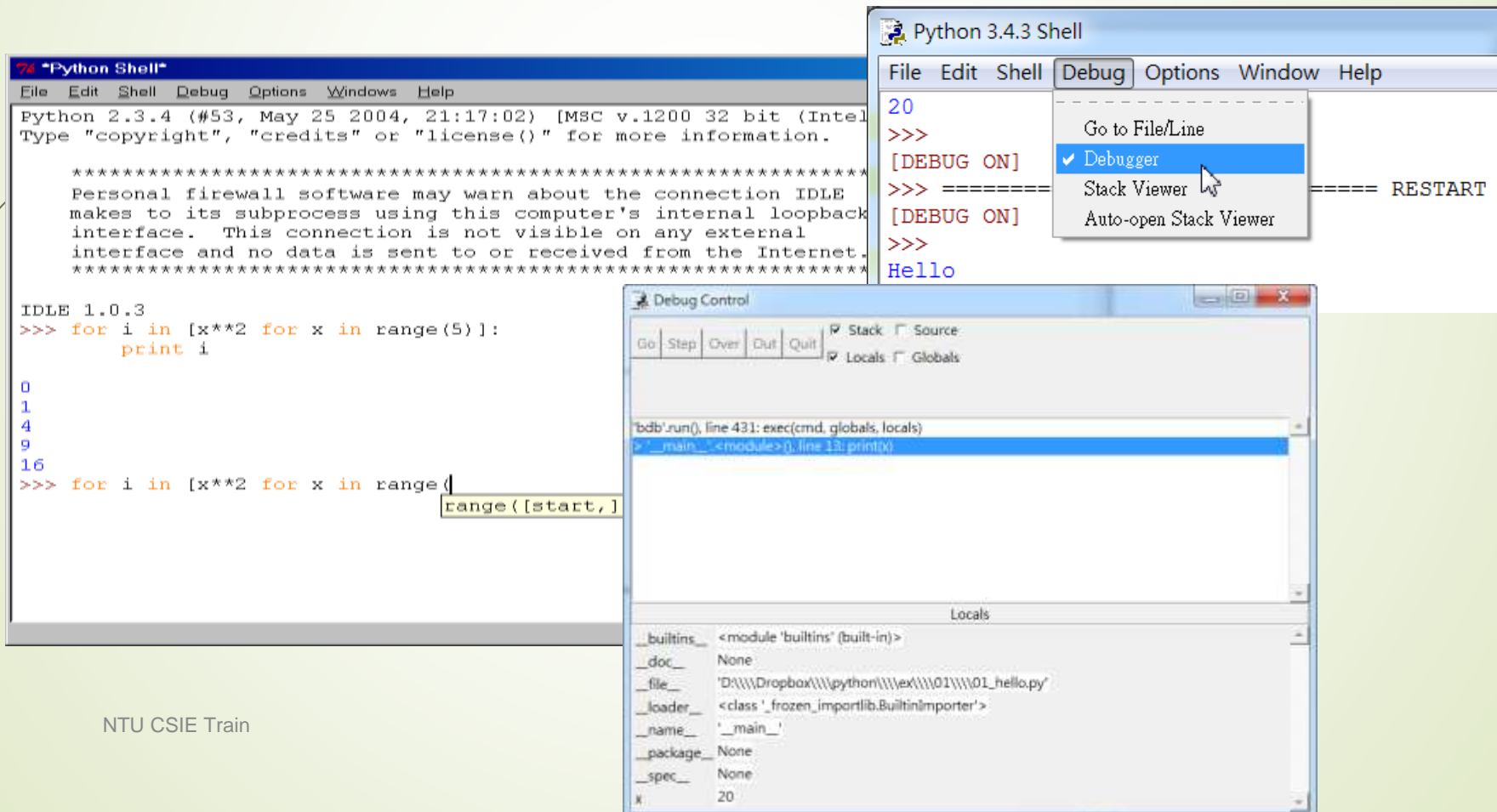


IDLE常用快捷

- 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

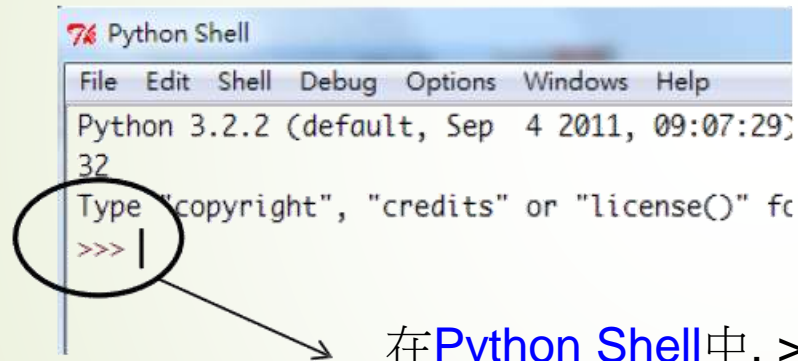
IDLE 開發環境

- 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。



The screenshot shows a window titled "Python Shell" with a menu bar (File, Edit, Shell, Debug, Options, Windows, Help). The text inside the window reads: "Python 3.2.2 (default, Sep 4 2011, 09:07:29)", "32", "Type 'copyright', 'credits' or 'license()' for", and a prompt ">>> |". A black circle highlights the ">>> |" prompt, with an arrow pointing to the explanatory text below.

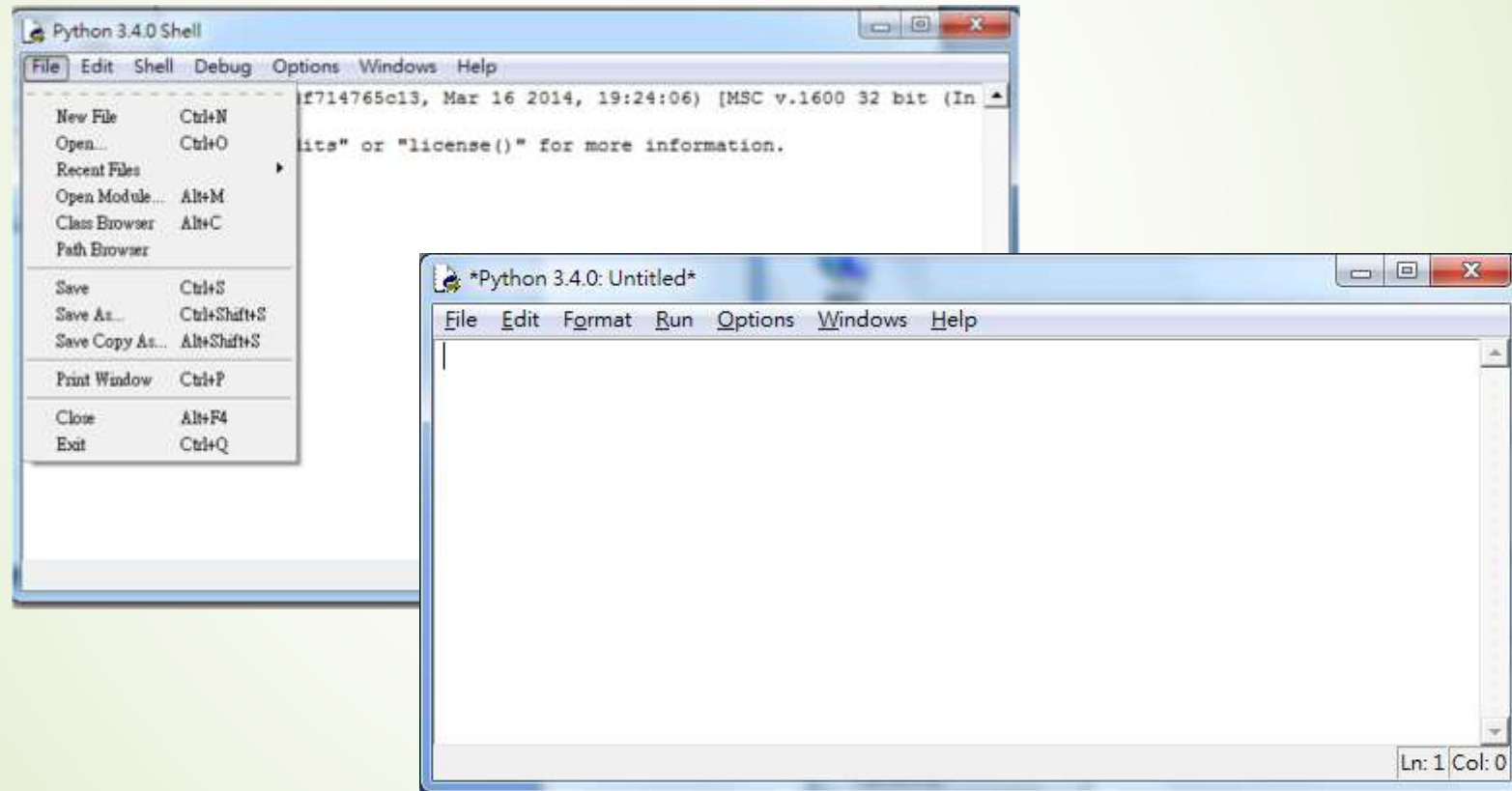
在Python Shell中，>>> 是一個輸入的提示
所有要下給直譯器的命令都要打在>>>之後

- 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

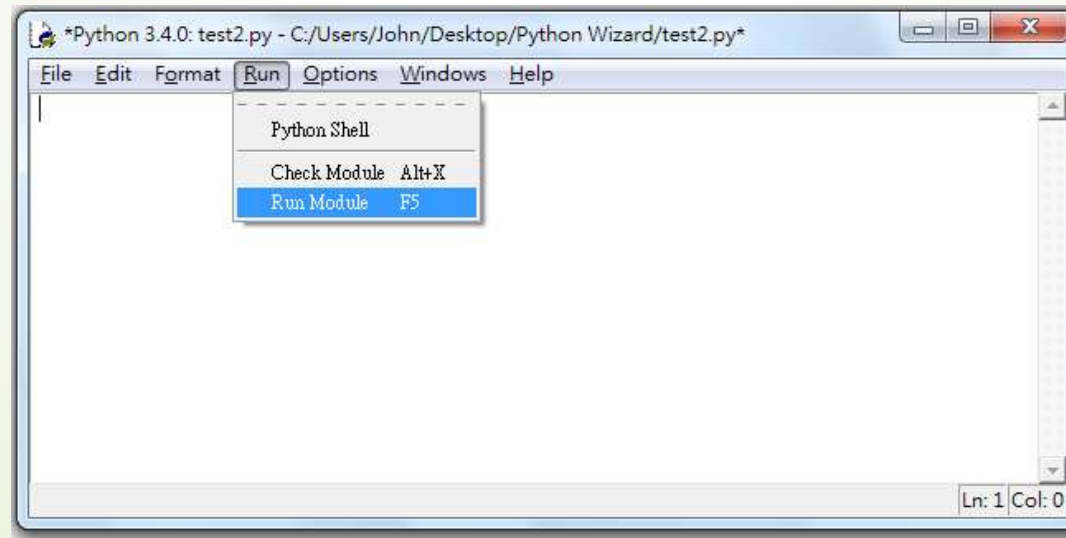
➤ 實際操作IDLE：

- ◆ 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

- 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。



Visual Studio Code

Visual Studio Code Docs Updates Blog API Extensions FAQ Search Docs Download

Version 1.34 is now available! Read about the new features and fixes from April.

Code editing. Redefined.

Free. Built on open source. Runs everywhere.

[Download for Windows](#)
Stable Build

Other platforms and Insiders Edition

By using VS Code, you agree to its license and privacy statement.

EXTENSIONS

- CF 1.2.2 350K ⭐⭐⭐⭐⭐
CF for Visual Studio Code (p...
Microsoft [Install](#)
- Python 0.16.0 21K ⭐⭐⭐⭐⭐
Linting, Debugging (multi-L...
Dot Jeyarajaseelan [Install](#)
- Debugger for Chrome 1.14.0
Debug your JavaScript code...
Microsoft JS Diagnos... [Install](#)
- C/C++ 0.7.0 143K ⭐⭐⭐⭐⭐
Complete C/C++ language ...
Microsoft [Install](#)
- Go 0.8.38 99K ⭐⭐⭐⭐⭐
Rich Go language support f...
lukehobart [Install](#)
- ESLint 0.10.0 88K ⭐⭐⭐⭐⭐
Integrates ESLint into VS Co...
Dirk Baumer [Install](#)

```
1 import app from './app';
2 import debugModule = require('debug');
3 import http = require('http');
4
5 const debug = debugModule('node-express-typescript:server');
6
7 // Get port from environment and store in Express.
8 const port = normalizePort(process.env.PORT || '3000');
9 app.set('port', port);
10
11 // create
12 const server = http.createServer(
13   server.on('request', (req, res) => {
14     server.on('message', (msg) => {
15       // normalizePort
16     })
17   })
18 );
19
20 function normalizePort(val: any): number|string|boolean {
21   let port = parseInt(val, 10);
22 }
```

IntelliSense Debugging Built-in Git Extensions

