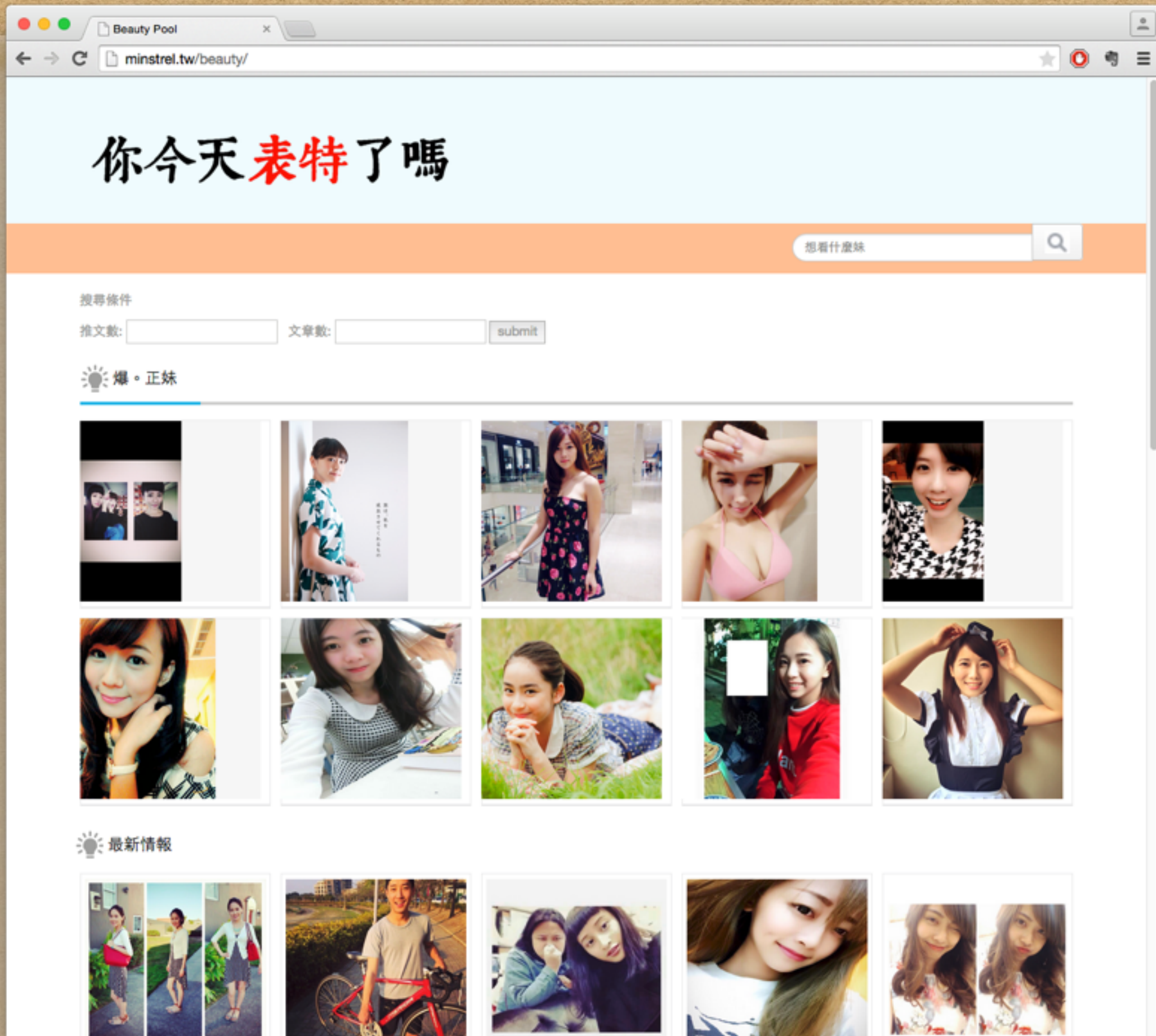


你今天表特了嗎

part2 - python & crawler

NC



使用到的技術

- ◆ 網站前端：html, css, jQuery, bootstrap
- ◆ 網站後端：Django
- ◆ 爬蟲程式：python (requests, pyquery)
- ◆ 資料庫：mongoDB (pymongo)
- ◆ 伺服器：apache

今天分享的部分

- ◆ 爬蟲程式：python (requests, pyquery)
- ◆ 資料庫：mongoDB (pymongo)

Python

Outline

- Why Python?
- Install
- hello world
- Property
- Type
- Control Flow
- Function
- File I/O
- Object and Class
- Module
- Built-in function
- Reference

Why Python?

- ♦ PEP 20

```
>>> import this
```

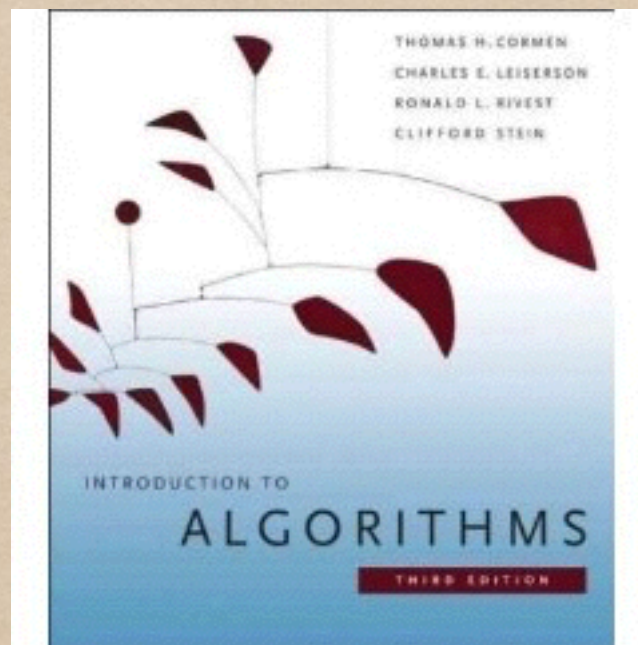
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.

Why Python?

- 好寫、好讀、好學
- 不用加分號「;」（想想看你被它陰過幾次）
- 利用縮行取代括號
- 好用的內建型別與內建函式

“I was translating pseudocode into Python.
It got smaller and more readable.”



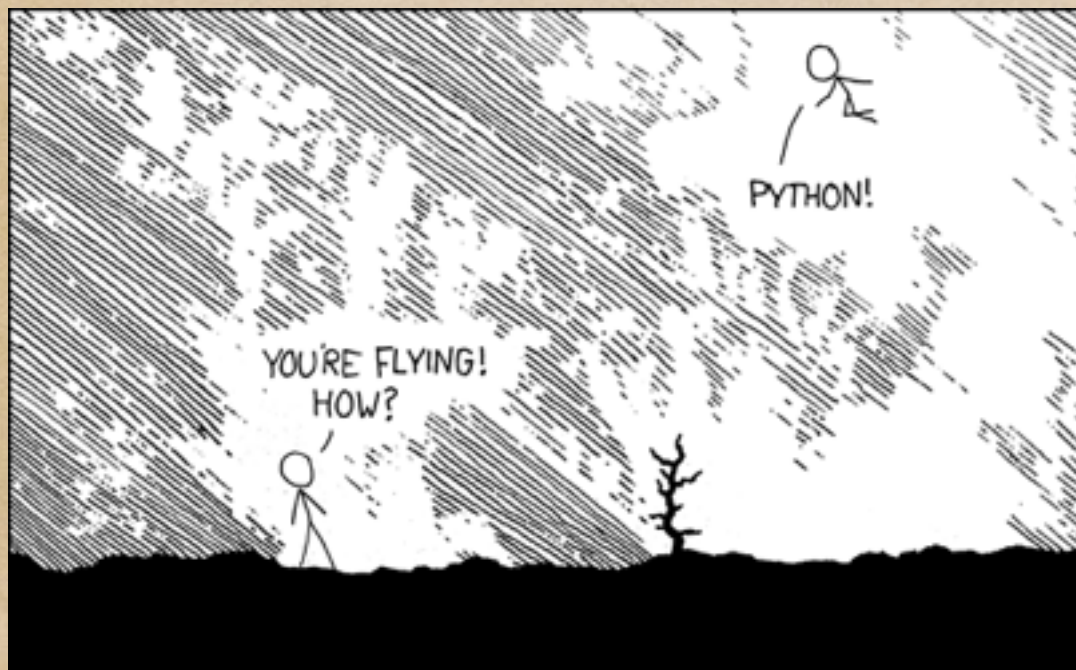
– Allen B. Downey (author of Think Python)

Why Python?

```
$ python -c "import antigravity"
```


Why Python?

```
$ python -c "import antigravity"
```



Only Python 2 today...

and something important
related to Crawler / Django

Install

Install

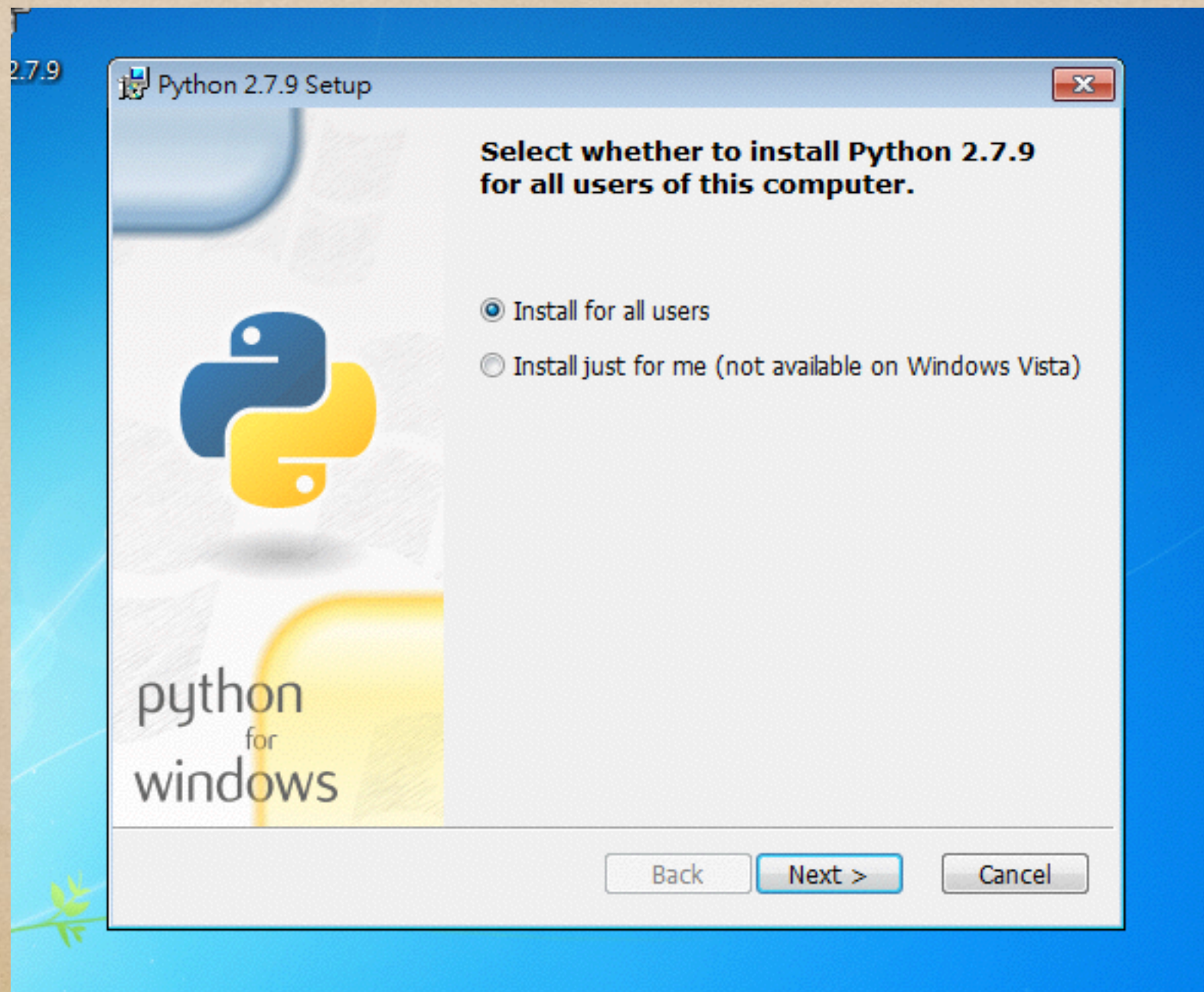
- ◆ Windows (win7)
 - ◆ <https://www.python.org/downloads/>
 - ◆ 下載2.7.9

Files

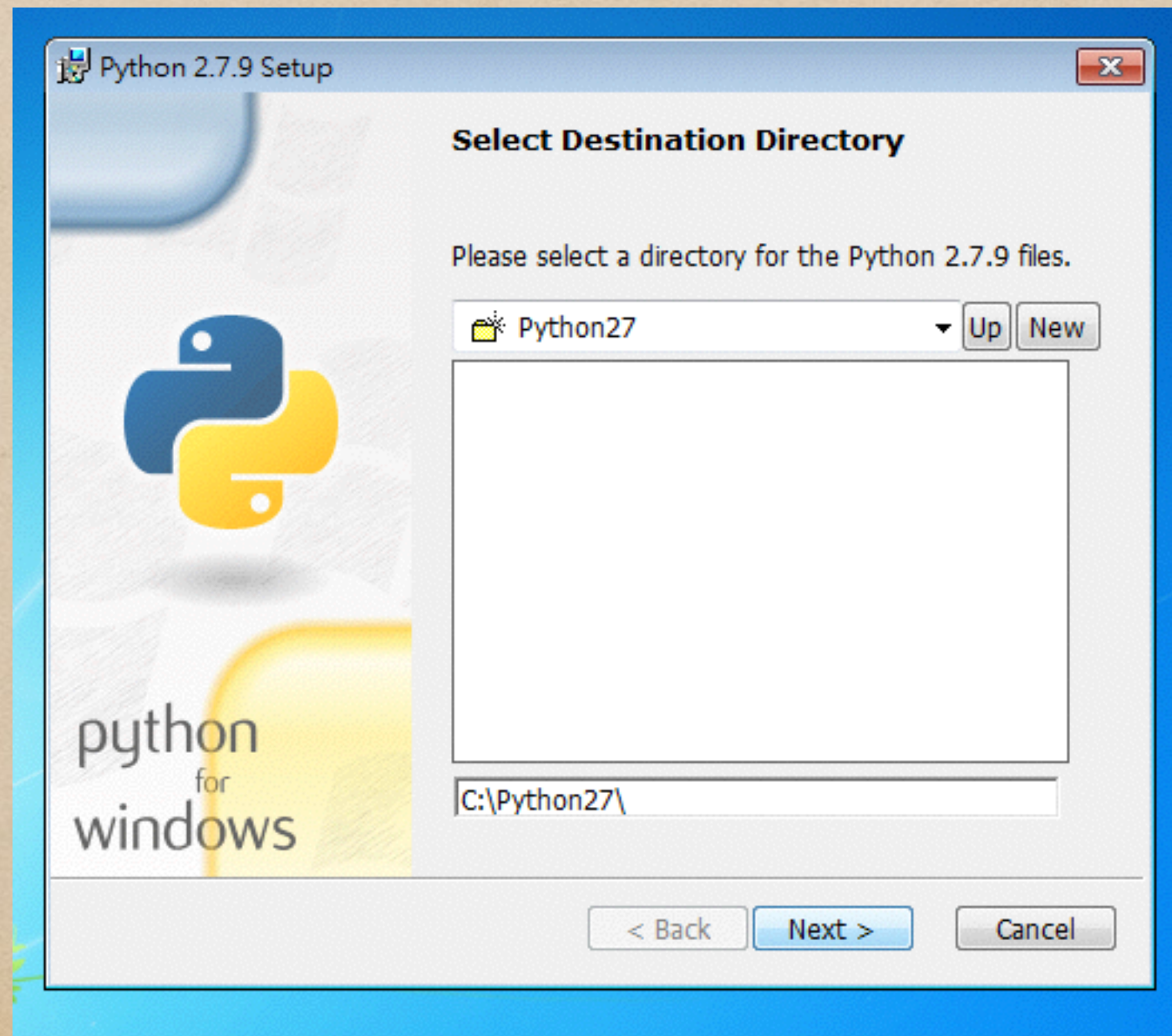
Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		5eebcaa0030dc4061156d3429657fb83	16657930	SIG
XZ compressed source tarball	Source release		38d530f7efc373d64a8fb1637e3baaa7	12164712	SIG
Mac OS X 32-bit i386/PPC installer	Mac OS X	for Mac OS X 10.5 and later	8d8a26fed767302ff38bc5056612c73a	23759976	SIG
Mac OS X 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	307c2b99a212204e7a1182a354328e94	22006891	SIG
Windows debug information files	Windows		c5838ec1cdd529a7065902c7573d1607	25969730	
Windows debug information files for 64-bit binaries	Windows		544e1137e8ecdce4f4cd2954ea520fa7	23979074	
Windows help file	Windows		dd438e999824c48001e54a2138c4f455	6120616	
Windows x86-64 MSI installer	Windows	for AMD64/EM64T/x64, not Itanium processors	21ee51a9f44b7160cb6fc68e29a1ddd0	18833408	
Windows x86 MSI installer	Windows		3ed20d8b06dcd339f814b38861f88fc9	18309120	

下載這個

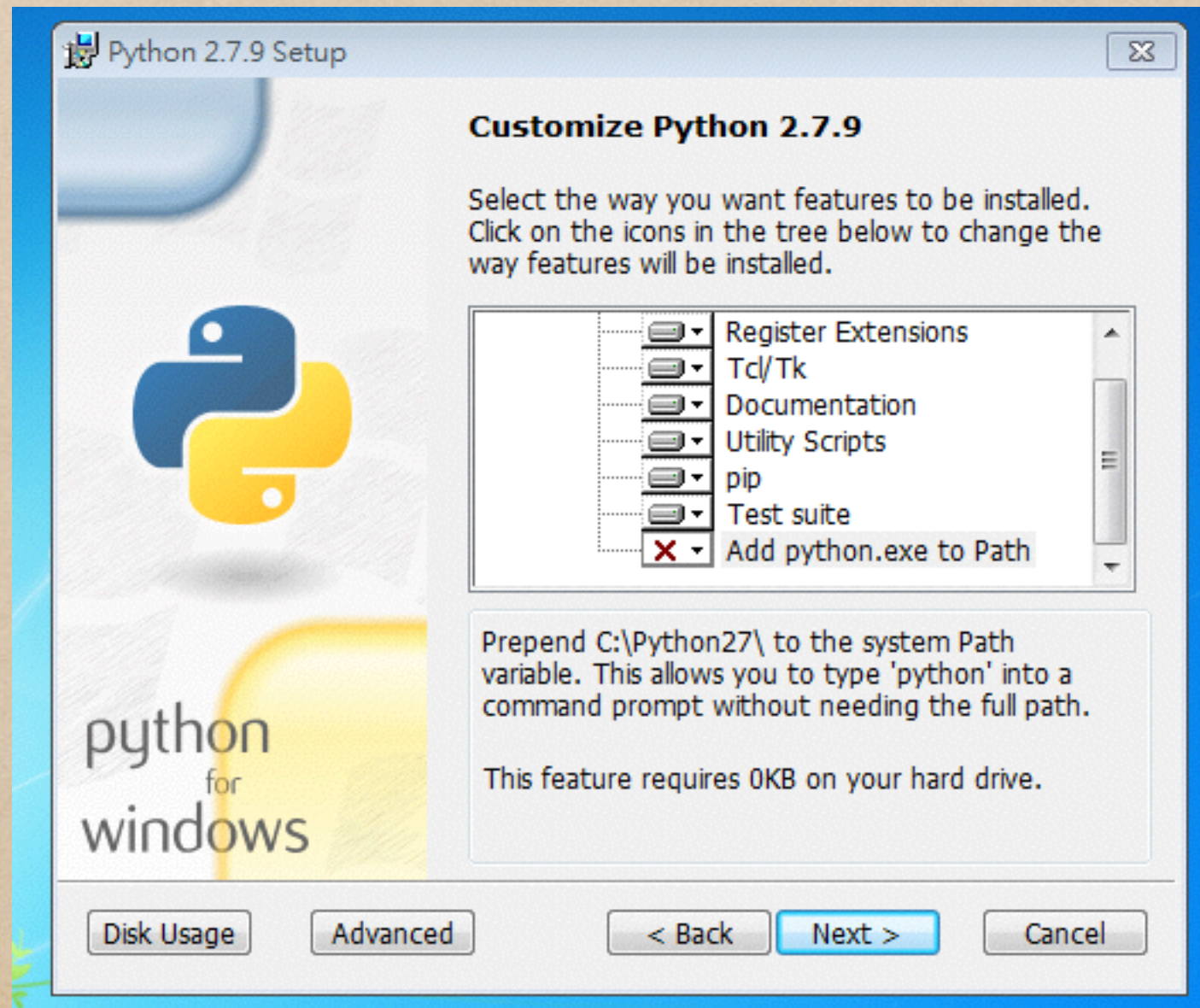
設定使用者 按「next」



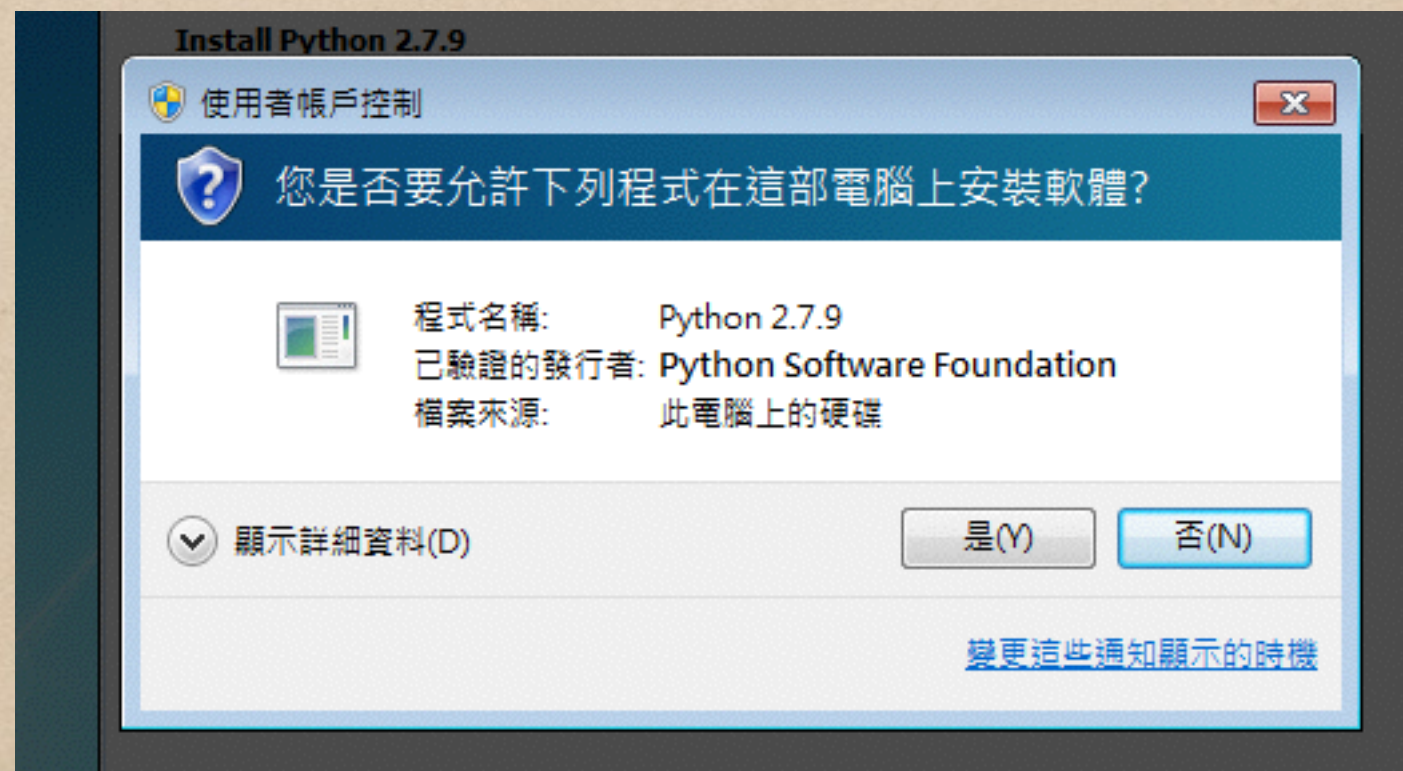
設定安裝路徑 再按「next」



設定安裝項目 再按「next」



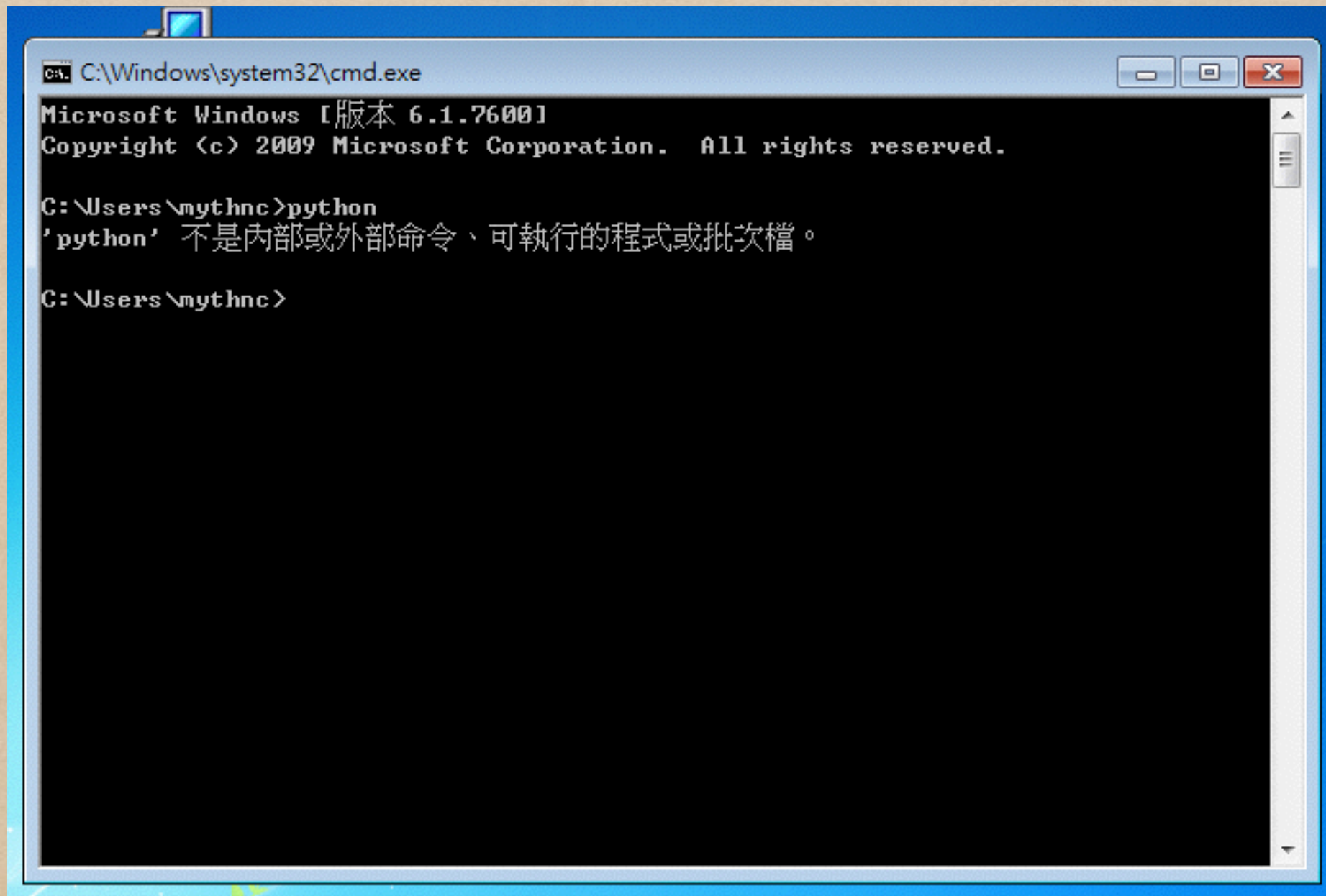
按「是」



安裝完成



沒設定PATH，錯誤示範

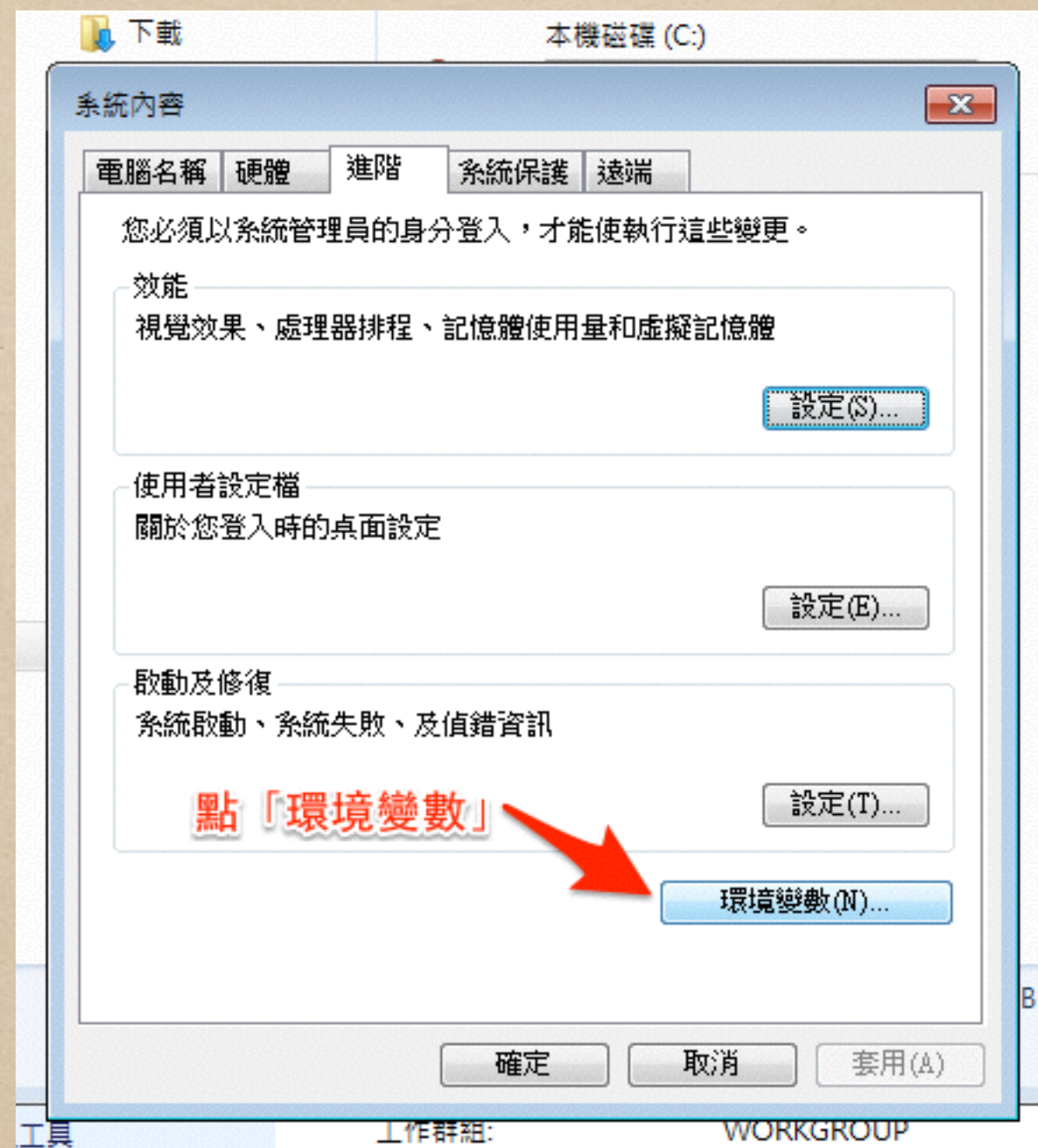


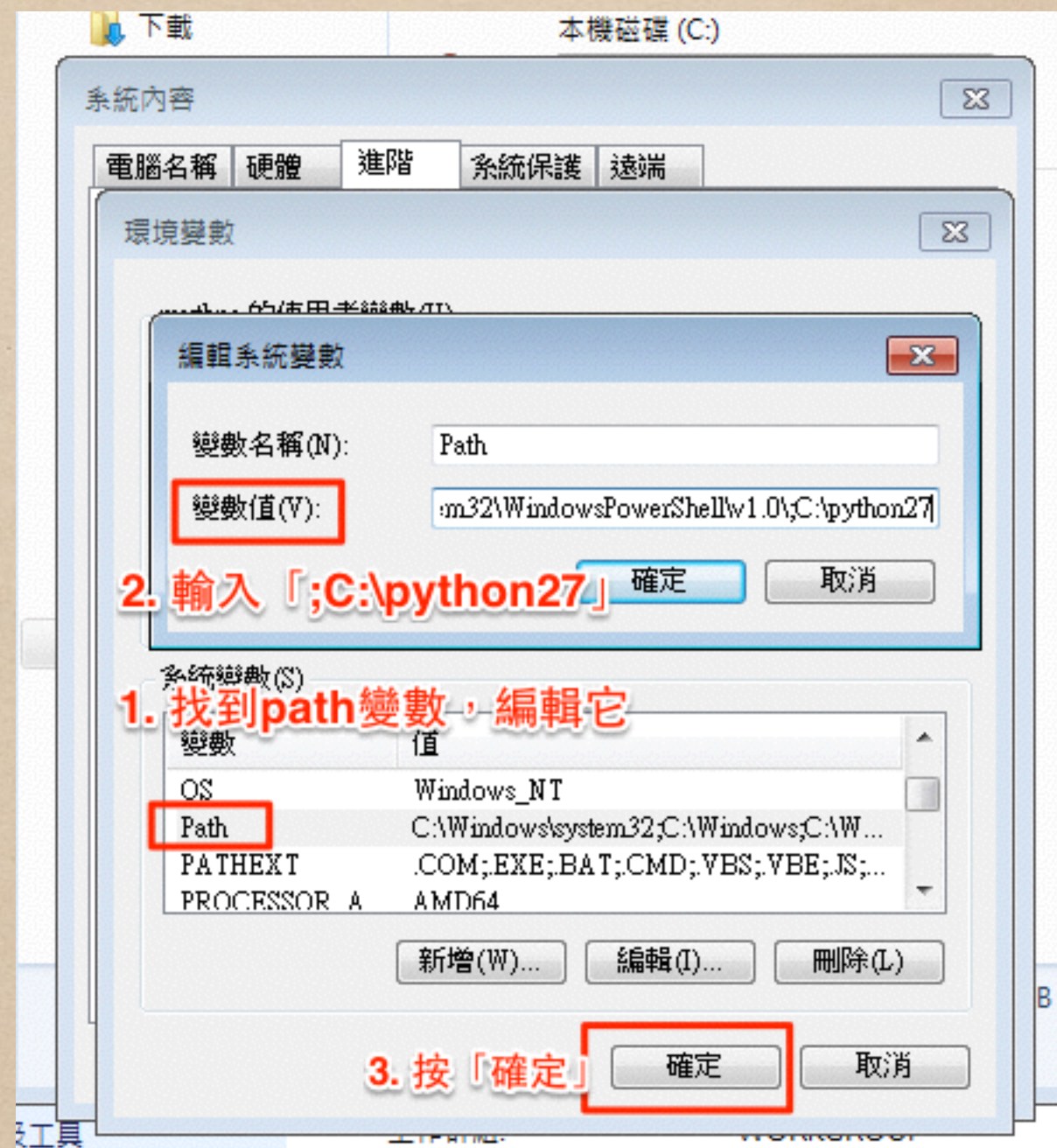
A screenshot of a Windows command prompt window. The title bar shows the path `C:\Windows\system32\cmd.exe`. The window contains the following text:

```
Microsoft Windows [版本 6.1.7600]  
Copyright (c) 2009 Microsoft Corporation. All rights reserved.  
  
C:\Users\mythnc>python  
'python' 不是內部或外部命令、可執行的程式或批次檔。  
  
C:\Users\mythnc>
```

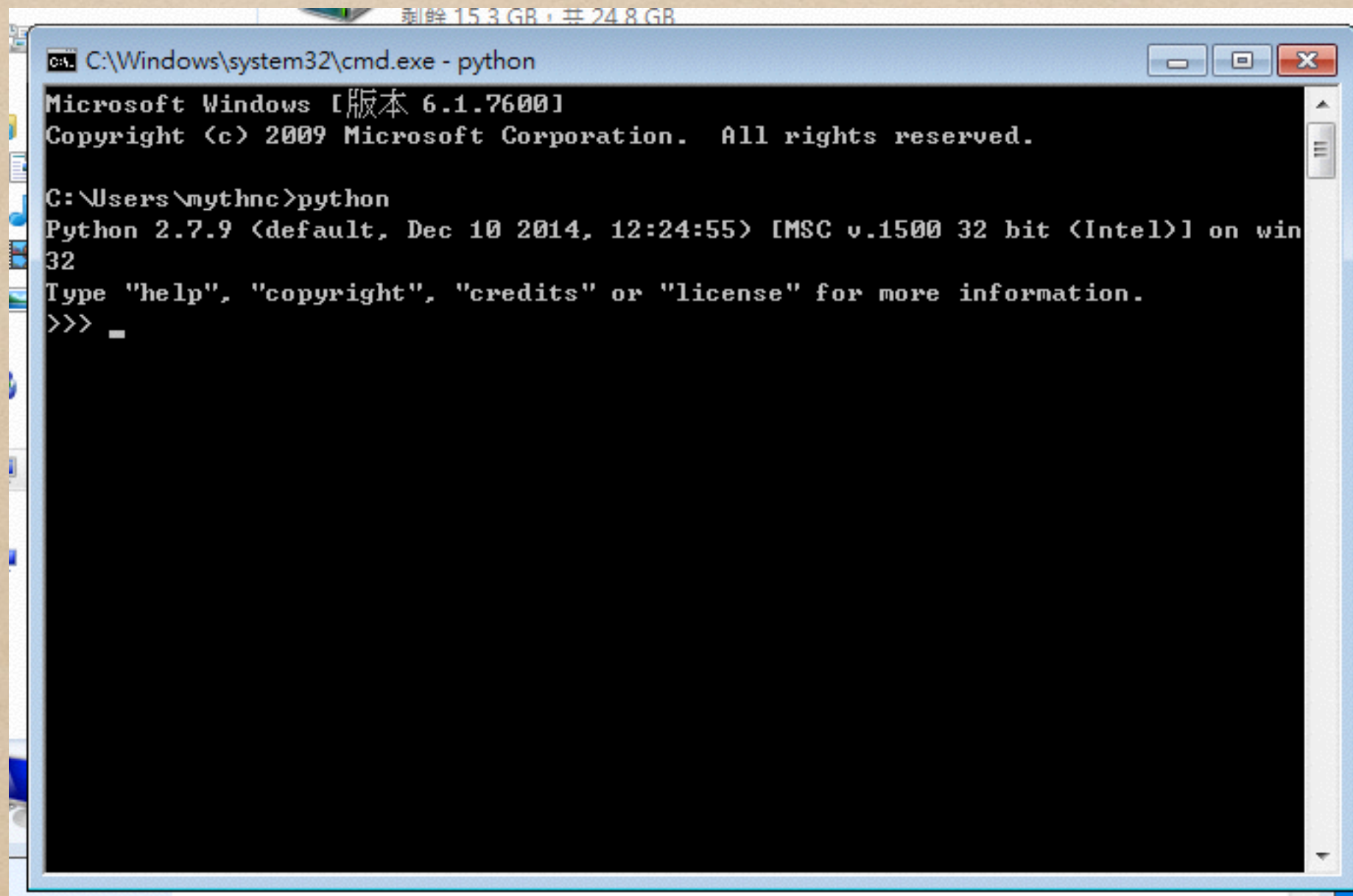






成功！



A screenshot of a Windows command prompt window. The title bar reads "C:\Windows\system32\cmd.exe - python". The window content shows the following text: "Microsoft Windows [版本 6.1.7600] Copyright (c) 2009 Microsoft Corporation. All rights reserved. C:\Users\mythnc>python Python 2.7.9 (default, Dec 10 2014, 12:24:55) [MSC v.1500 32 bit (Intel)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>> _". The cursor is positioned after the underscore on the last line.

```
C:\Windows\system32\cmd.exe - python
Microsoft Windows [版本 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\mythnc>python
Python 2.7.9 (default, Dec 10 2014, 12:24:55) [MSC v.1500 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```


Install

- ◆ Mac
 - ◆ <http://brew.sh>
 - ◆ 需下載「命令列開發者工具」
(系統會提示)
 - ◆ `$ brew install python`
- ◆ 可裝可不裝

Install

- ◆ Linux (Debian / Ubuntu)
 - ◆ `$ sudo apt-get install python`
- ◆ 可裝可不裝

Install

- How to check?
- type “python” in terminal / command line

```
Python 2.7.9 (default, Jan 7 2015, 11:49:12)  
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.56)]  
on darwin  
Type "help", "copyright", "credits" or "license" for  
more information.  
>>>
```

- Interactive mode (very useful!!)

Install

- ◆ 如果安裝遇到問題，試著自行解決。
(善用google)
- ◆ 因為你會一直遇到安裝的問題
(安裝ruby...安裝java...安裝eclipse...安裝...)

pip path setting

- ◆ A tool for installing python packages
- ◆ windows練習題：試著加入pip的路徑於path中
 - ◆ 一樣的做法
 - ◆ 在path中加入「;C:\Python27\Scripts」
 - ◆ 如此，即可在cmd中使用pip

IPython

- ◆ 神兵利器
- ◆ 易查詢，提供補齊
- ◆ `$ pip install ipython`
- ◆ windows使用者請再裝pyreadline才有補齊功能
- ◆ `$ pip install pyreadline # windows`
- ◆ `$ ipython`

IPython

- ◆ 練習題

1. 試著在ipython中查詢函式，並閱讀說明
輸入"`help(str.find)`"與"`str.find?`"，
並比較之間的差異
2. 試著查詢module
輸入"`import csv`"，接著輸入"`csv?`"與
"`csv??`"，"`csv.`"按<tab>，並比較之間的差異

IPython

- ◆ 練習題

3. 試著隨便輸入變數(`a = 3`, `b = 's'`)
接著輸入 `'who'`, `'whos'`，你有什麼發現？

4. 試著交替使用 `python` 與 `ipython`，
你覺得哪個比較好用？

Editor / IDE

- ◆ Pick whatever you like
 - ◆ VIM, Emacs, Sublime, Notepad++...etc
 - ◆ 如果可以，選個可以寫任何程式語言的編輯器
- ◆ <https://wiki.python.org/moin/PythonEditors>

Hello World

hello world!

1. in Interactive mode

```
In [1]: print 'hello world!'
hello world!
```

2. in py file (hello.py) (在windows上免寫第一行)

```
#!/usr/bin/env python
print 'hello world!'
```

```
$ python hello.py
hello world!
```

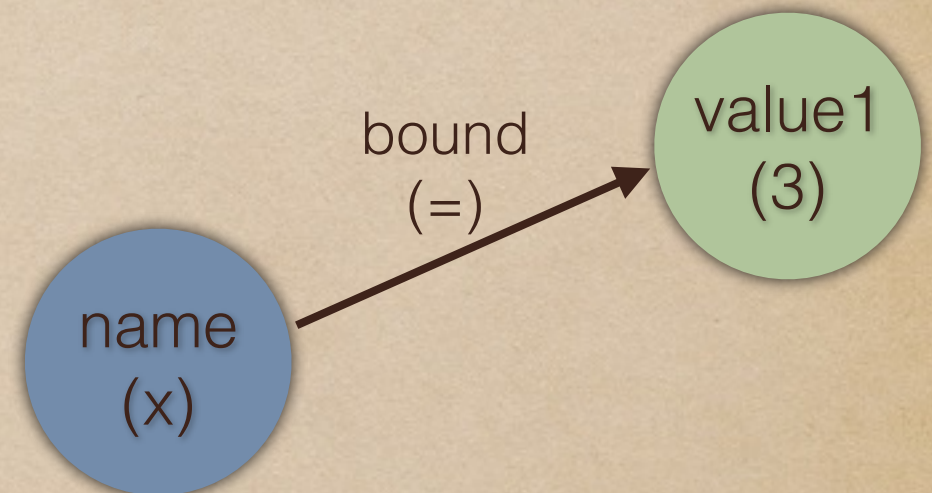

Property

- ◆ Dynamic Typing
 - ◆ checking types in run time
 - ◆ 變數不需要事先宣告
 - ◆ a name bound to a value

Property

- Dynamic Typing
 - checking types in run time
 - 變數不需要事先宣告
 - a name bound to a value

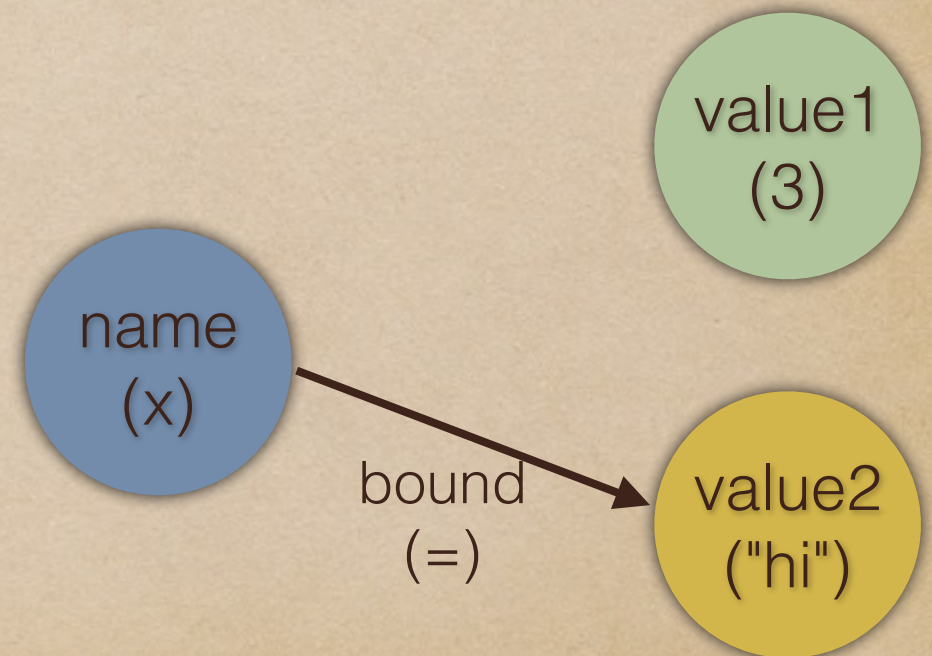
```
>>> x = 3
```



Property

- Dynamic Typing
 - checking types in run time
 - 變數不需要事先宣告
 - a name bound to a value

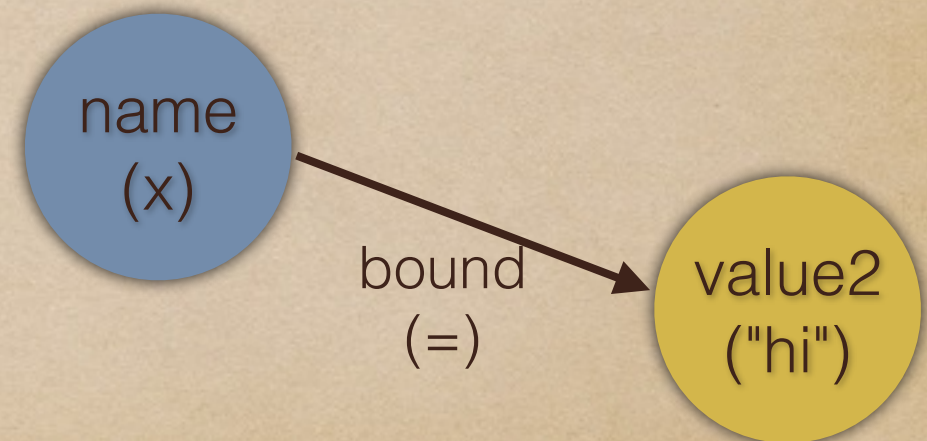
```
>>> x = 3  
>>> x = "hi"
```



Property

- Dynamic Typing
 - checking types in run time
 - 變數不需要事先宣告
 - a name bound to a value

```
>>> x = 3
>>> x = "hi"
```



Property

- Strong Typing
 - 不同型別無法相互運算

```
>>> 'hello' + 3 # TypeError
```

```
>>> 'hello' + str(3) # work  
'hello3'
```


Property

- ♦ mutable
 - ♦ value can be modified after created.
 - ♦ dict, list, set
- ♦ immutable
 - ♦ value can't be modified after created.
 - ♦ number, str, tuple

Type

- ♦ None
- ♦ Numbers: int, long, float, complex, bool
- ♦ Sequences: str, unicode, list, tuple, xrange
- ♦ Mapping: dict
- ♦ Sets: set, frozenset

Type - 今天會講到的

- ◆
- ◆ Numbers: int, , float, , bool
- ◆ Sequences: str, , list, tuple,
- ◆ Mapping: dict
- ◆ Sets: set,

Please check the `python_tutorial.html` file

Built-in Function

Built-in Functions				
<code>abs()</code>	<code>divmod()</code>	<code>input()</code>	<code>open()</code>	<code>staticmethod()</code>
<code>all()</code>	<code>enumerate()</code>	<code>int()</code>	<code>ord()</code>	<code>str()</code>
<code>any()</code>	<code>eval()</code>	<code>isinstance()</code>	<code>pow()</code>	<code>sum()</code>
<code>basestring()</code>	<code>execfile()</code>	<code>issubclass()</code>	<code>print()</code>	<code>super()</code>
<code>bin()</code>	<code>file()</code>	<code>iter()</code>	<code>property()</code>	<code>tuple()</code>
<code>bool()</code>	<code>filter()</code>	<code>len()</code>	<code>range()</code>	<code>type()</code>
<code>bytearray()</code>	<code>float()</code>	<code>list()</code>	<code>raw_input()</code>	<code>unichr()</code>
<code>callable()</code>	<code>format()</code>	<code>locals()</code>	<code>reduce()</code>	<code>unicode()</code>
<code>chr()</code>	<code>frozenset()</code>	<code>long()</code>	<code>reload()</code>	<code>vars()</code>
<code>classmethod()</code>	<code>getattr()</code>	<code>map()</code>	<code>repr()</code>	<code>xrange()</code>
<code>cmp()</code>	<code>globals()</code>	<code>max()</code>	<code>reversed()</code>	<code>zip()</code>
<code>compile()</code>	<code>hasattr()</code>	<code>memoryview()</code>	<code>round()</code>	<code>__import__()</code>
<code>complex()</code>	<code>hash()</code>	<code>min()</code>	<code>set()</code>	<code>apply()</code>
<code>delattr()</code>	<code>help()</code>	<code>next()</code>	<code>setattr()</code>	<code>buffer()</code>
<code>dict()</code>	<code>hex()</code>	<code>object()</code>	<code>slice()</code>	<code>coerce()</code>
<code>dir()</code>	<code>id()</code>	<code>oct()</code>	<code>sorted()</code>	<code>intern()</code>

don't reinvent the wheel!!

<https://docs.python.org/2.7/library/functions.html>

3rd party library

- ◆ 主要還是看你想做什麼(寫網頁, 寫系統, 文字處理, 資料分析, 寫database...etc)
- ◆ 幾乎都有相對應的第三方函式庫跟教學文件
 - PyPI
- ◆ 善用pip管理python套件

Reference

Resources

- ◆ Python / 第一次用就上手
 - ◆ <http://wiki.python.org.tw/Python/第一次用就上手>
- ◆ ptt@python板
- ◆ [官方文件](#)很完整
- ◆ 電子週報 - [PythonWeekly](#)

E-books

1. Dive Into Python (free)

- <http://www.diveintopython.net>



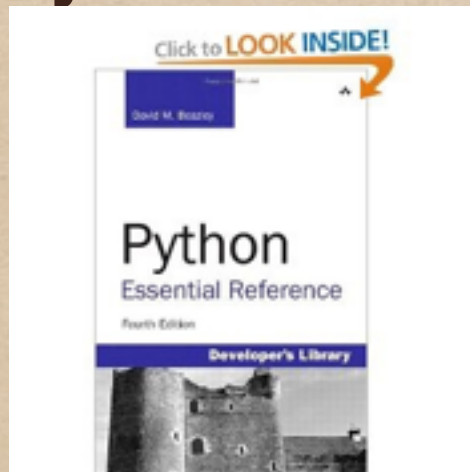
2. Think Python (free)

- <http://www.greenteapress.com/thinkpython/thinkpython.html>

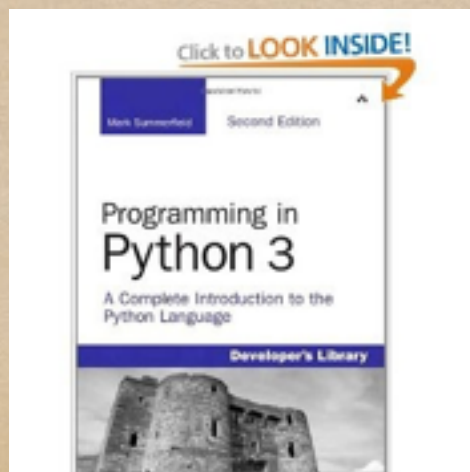


Books

3. Python Essential Reference



4. Programming in Python 3



Community

- ◆ Taipei.py
- ◆ PyHUG (新竹)
- ◆ Taichung.py
- ◆ Tainan.py
- ◆ PyLadies @ Taiwan (女生專屬)
- ◆ 花蓮.py
- ◆

Conference

- ◆ 2012首次舉辦
- ◆ PyCon APAC 2015
 - ◆ 6/5 ~ 6/7 @中研院
 - ◆ 學生票很便宜！
 - ◆ 研究生沒有的福利
- ◆ 好吃又好玩
- ◆ 錄影檔會釋出放在youtube

Question?

Crawler

預先安裝套件


- ◆ 依序裝入下列套件
- ◆ \$ pip install requests
- ◆ \$ pip install pyquery
- ◆ \$ 安裝mongodb
 - ◆ Install MongoDB on Windows
- ◆ \$ pip install pymongo

下載對應的安裝檔

Production Release (3.0.2)

4/9/2015 [Release Notes](#) [Changelog](#)

Download Source: [tgz](#) | [zip](#)

 Windows

 Linux

 Mac OS X

Solaris

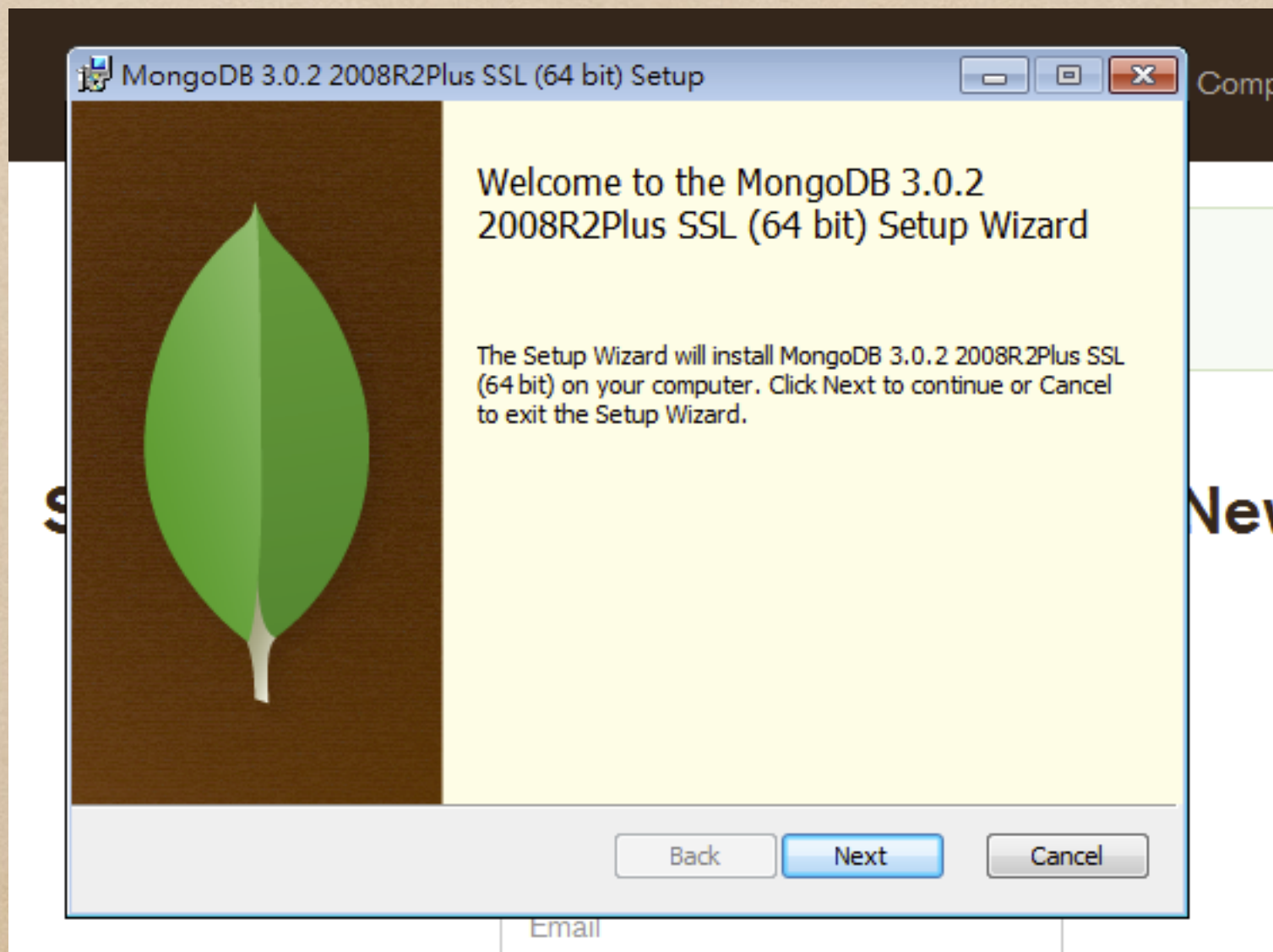
VERSION:

Windows 64-bit 2008 R2+

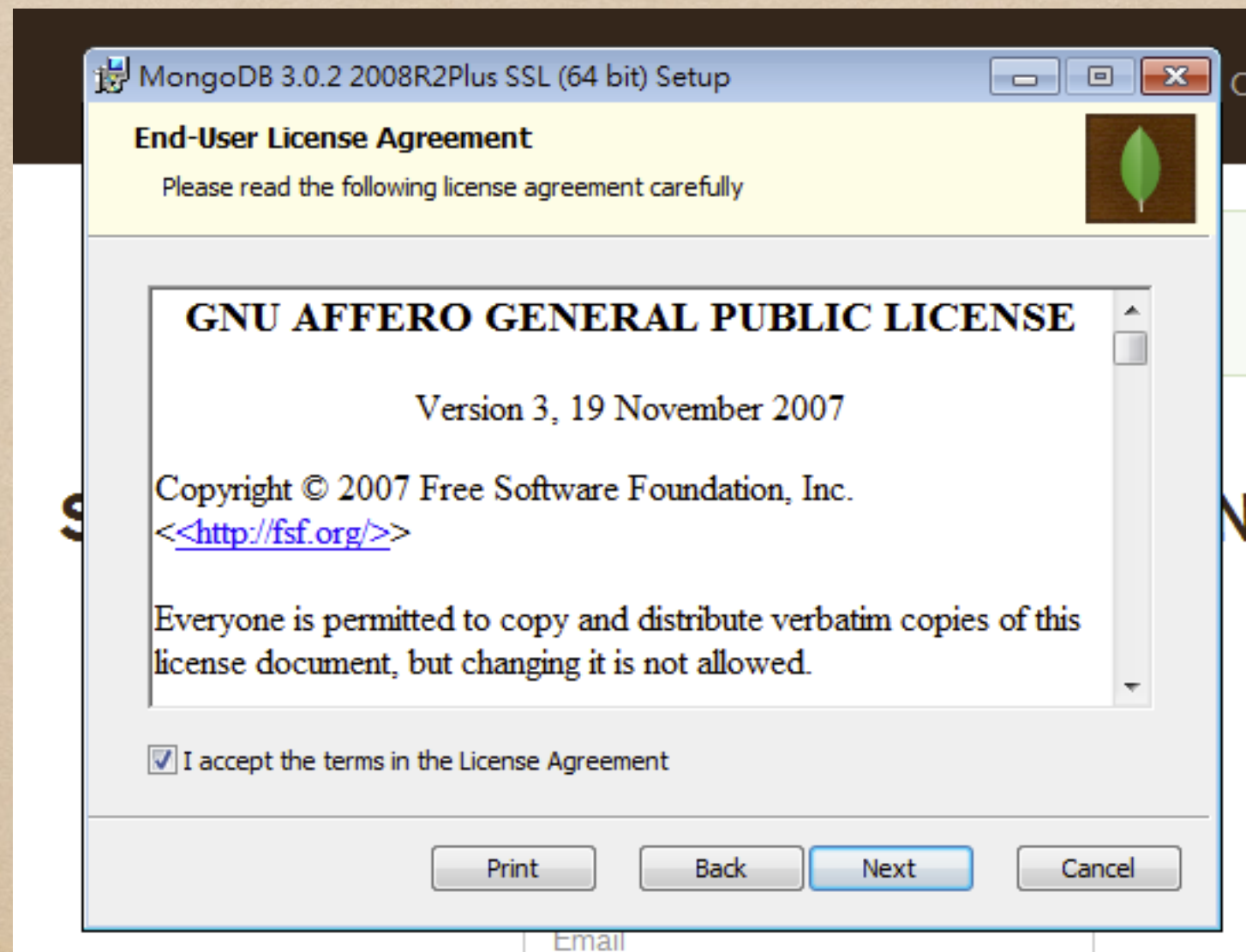
INSTALLATION PACKAGE: [Installation Instructions](#) [View Build Archive](#)

 **DOWNLOAD (MSI)**

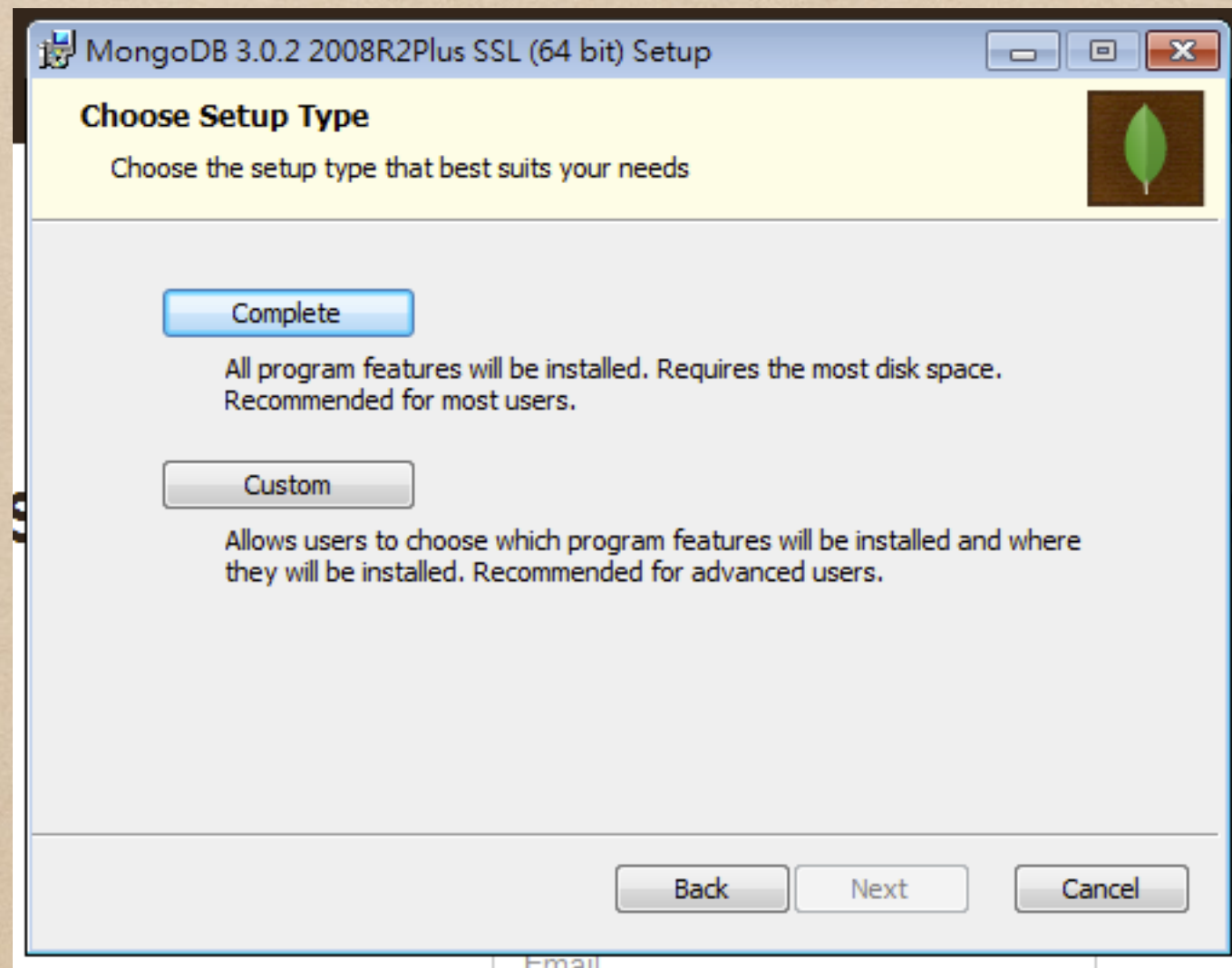
按next



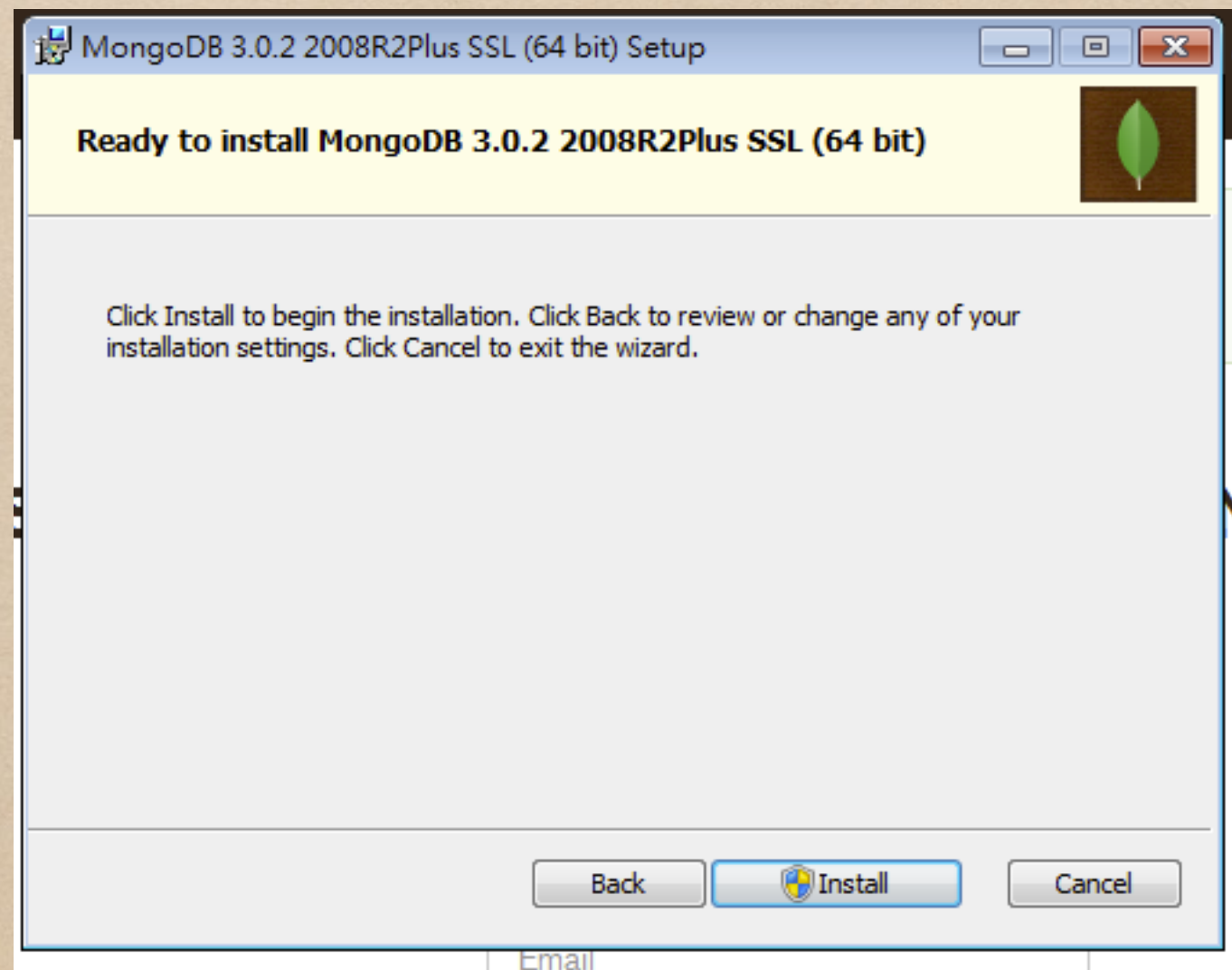
打勾，按next



選complete，按next



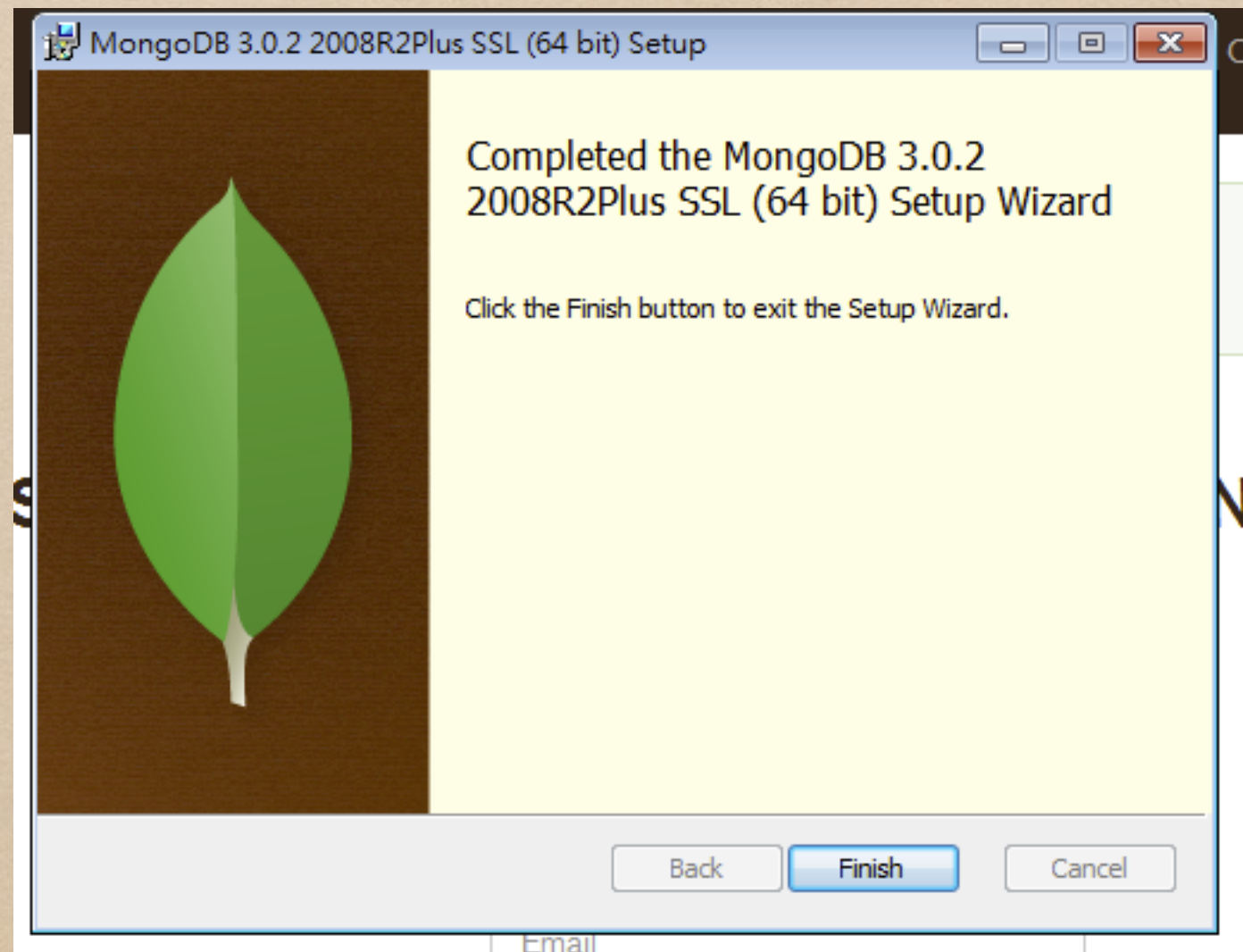
按install



選「是」



安裝完成



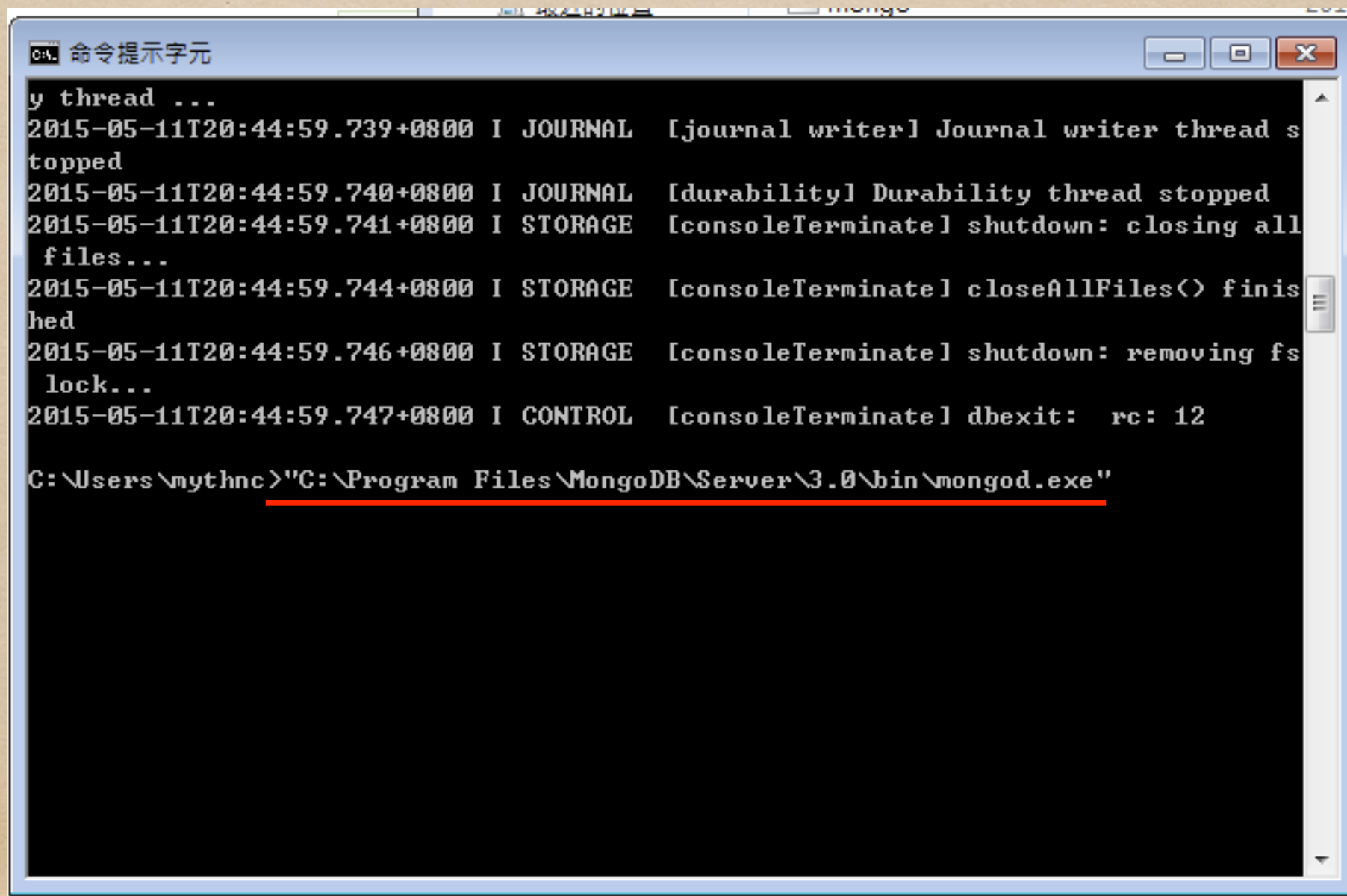
接著是測試

設置資料庫

- ◆ \$ md \data\db

啟動mongod(一定要先開)

- 把mongod拖曳到cmd中，按enter

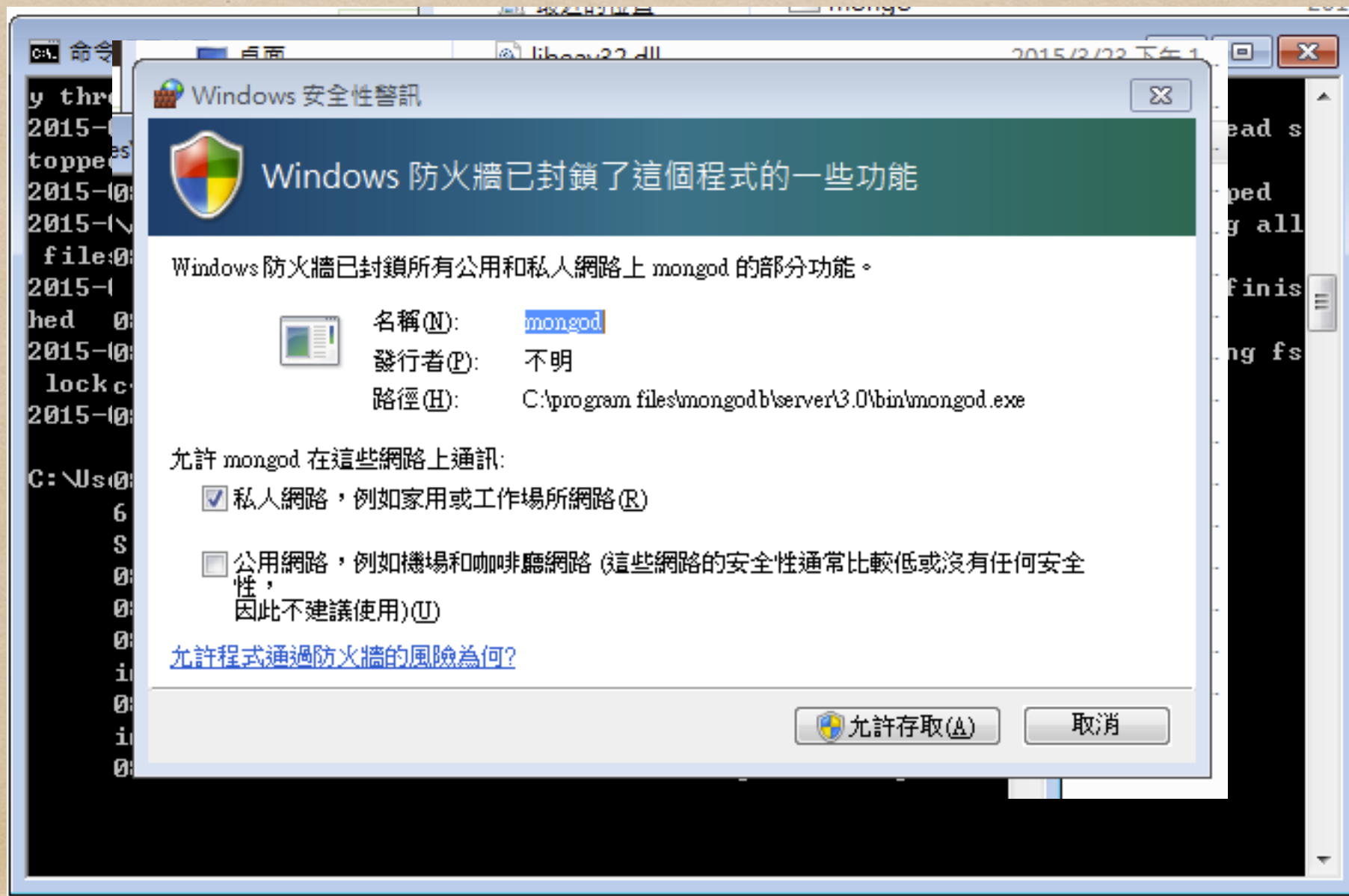


```
命令提示字元
y thread ...
2015-05-11T20:44:59.739+0800 I JOURNAL [journal writer] Journal writer thread s
topped
2015-05-11T20:44:59.740+0800 I JOURNAL [durability] Durability thread stopped
2015-05-11T20:44:59.741+0800 I STORAGE [consoleTerminate] shutdown: closing all
files...
2015-05-11T20:44:59.744+0800 I STORAGE [consoleTerminate] closeAllFiles() finis
hed
2015-05-11T20:44:59.746+0800 I STORAGE [consoleTerminate] shutdown: removing fs
lock...
2015-05-11T20:44:59.747+0800 I CONTROL [consoleTerminate] dbexit: rc: 12

C:\Users\mythnc>"C:\Program Files\MongoDB\Server\3.0\bin\mongod.exe"
```


點「允許存取」

- 把mongod拖曳到cmd中，按enter



寫入資料至mongodb

- ◆ 參考教學 ([連結](#))
- ◆ mongod記得要先開

```
from pymongo import MongoClient
import datetime

client = MongoClient()
db = client.test_database
collection = db.test_collection

post = {"author": "Mike",
        "text": "My first blog post!",
        "tags": ["mongodb", "python", "pymongo"],
        "date": datetime.datetime.utcnow()}

posts = db.posts
post_id = posts.insert_one(post).inserted_id
print post_id
```


寫入資訊

```
命令提示字元 - "C:\Program Files\MongoDB\Server\3.0\bin\mongod.exe"
2015-05-11T20:54:18.564+0800 I NETWORK [initandlisten] connection accepted from
127.0.0.1:49356 #2 (2 connections now open)
2015-05-11T20:54:18.568+0800 I INDEX [conn2] allocating new ns file C:\data\db\test_database.ns, filling with zeroes...
2015-05-11T20:54:18.999+0800 I STORAGE [FileAllocator] allocating new datafile
C:\data\db\test_database.0, filling with zeroes...
2015-05-11T20:54:19.000+0800 I STORAGE [FileAllocator] creating directory C:\data\db\_tmp
2015-05-11T20:54:19.151+0800 I STORAGE [FileAllocator] done allocating datafile
C:\data\db\test_database.0, size: 64MB, took 0.149 secs
2015-05-11T20:54:19.160+0800 I WRITE [conn2] insert test_database.posts query
: { _id: ObjectId('5550a67a3564af0878d9e4bc'), date: new Date(1431348816323), te
xt: "my firest post", tags: [ "mongo", "pymongo" ], author: "mike" } ninserted:1
keyUpdates:0 writeConflicts:0 numYields:0 locks:< Global: { acquireCount: { w:
2 } }, MMAPV1Journal: { acquireCount: { w: 8 } }, Database: { acquireCount: { w:
1, W: 1 } }, Collection: { acquireCount: { W: 1 } }, Metadata: { acquireCount:
{ W: 4 } } } 593ms
2015-05-11T20:54:19.161+0800 I COMMAND [conn2] command test_database.$cmd comma
nd: insert { insert: "posts", ordered: true, documents: [ { _id: ObjectId('5550a
67a3564af0878d9e4bc'), date: new Date(1431348816323), text: "my firest post", ta
gs: [ "mongo", "pymongo" ], author: "mike" } ] } keyUpdates:0 writeConflicts:0 n
umYields:0 reslen:40 locks:< Global: { acquireCount: { w: 2 } }, MMAPV1Journal:
{ acquireCount: { w: 8 } }, Database: { acquireCount: { w: 1, W: 1 } }, Collecti
on: { acquireCount: { W: 1 } }, Metadata: { acquireCount: { W: 4 } } } 594ms
```


簡易解説

- ◆ MongoDB is No SQL Database
- ◆ collections / databases in MongoDB is that they are created lazily
- ◆ Database - Collection - Document (NoSQL)
Database - Table - Row/Record (RDBMS)

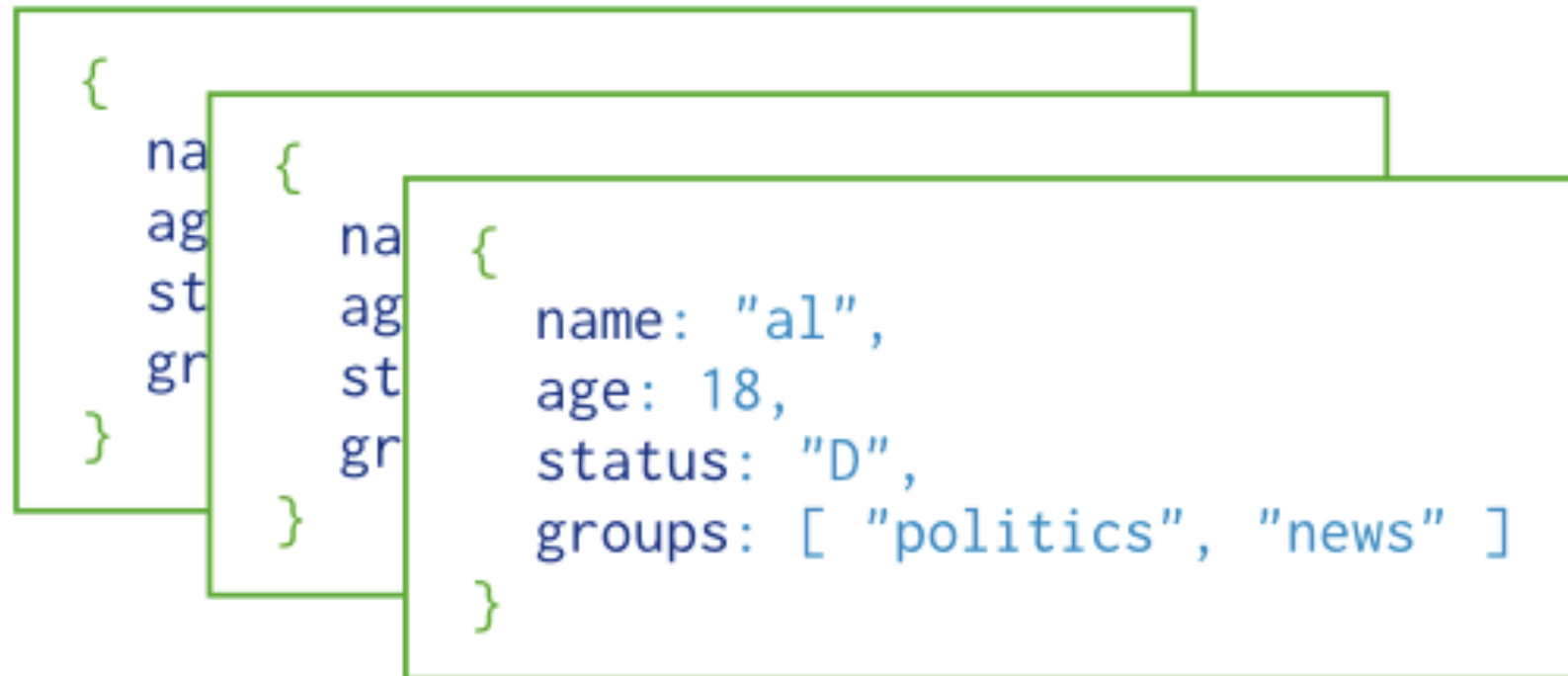
Document

- ♦ documents are similar to JSON objects

```
{  
  name: "sue",  
  age: 26,  
  status: "A",  
  groups: [ "news", "sports" ]  
}
```



Collection



```
{  
  name: "al",  
  age: 18,  
  status: "D",  
  groups: [ "politics", "news" ]  
}
```

Collection

Query

- ◆ `for post in posts.find():`
 `post`
- ◆ 結果會以dict形式顯示，每個post是一筆結果
- ◆ 搜尋條件放在`find()`中
- ◆ `for post in posts.find({'author': 'Mike'}):`
 `post`

Index

- ♦ To make query faster

```
from pymongo import MongoClient
import datetime

client = MongoClient()
db = client.test_database
collection = db.test_collection

post = {"author": "Mike",
        "text": "My first blog post!",
        "tags": ["mongodb", "python", "pymongo"],
        "date": datetime.datetime.utcnow()}

posts = db.posts
post_id = posts.insert_one(post).inserted_id
print post_id
```

```
from pymongo import ASCENDING, DESCENDING
posts.create_index([("date", DESCENDING), ("author", ASCENDING)])
```


Please check the crawler_tutorial.html file

API

- ◆ 爬蟲之餘的選擇
- ◆ facebook API, twitter API, plurk API, flickr API

Reference

- ◊ 爬蟲相關
 - ◊ Web Crawler教學 (c3h3)
 - ◊ Web Cralwer工具箱 (joe)
 - ◊ kimono (turn websites into APIs)
- ◊ MongoDB相關
 - ◊ MongoDB Tutorial (CodeData)
 - ◊ mongodb doc
- ◊ Library相關
 - ◊ PyMongo Tutorial
 - ◊ Requests
 - ◊ pyquery

Question?