

大同大學資訊創意研究社

# Python 程式語言教學 (I)

社長 黃一晉

Source by Mosky 2011/04/07 自由軟體校園工作坊 -  
Python 介紹：程式設計？就這麼簡單！



# Programming In Fedora

## Basic Python Intro

Today's Topic

# Mosky

未來三週講師介紹

# Ubuntu-TW

活動長



# COSCUP 2011

場務組副工



# Plurk

## 實習生



今天是來

~~洗腦~~

# 介紹

Python 這個語言



程式語言那麼多

Why Python ?

因為

Python 用途廣泛

從

算離期中考還有幾天

到

發射火箭

都

用的到 Python



恭喜您獲得虛寶序號一組！序號：密碼：

恭喜您獲得虛寶序號一組！序號：**1188848771**密碼：

CSA869C-00000000

**恭喜您獲得虛寶序號一組！序號：110872密碼：**

71CBF024-00000000-00

恭喜您獲得虛寶序號一組！序號：116 273密碼：

BCC7C7 F242

恭喜您獲得虛寶序號一組！序號：110 374密碼：

C102DE7 [REDACTED] C72

恭喜你獲得虛寶序號一組！序號：118818375密碼：

4612E7 8131

恭喜您獲得虛寶序號一組！序號：110021376密碼：

REF ID: A68079 REF ID: A68079

恭喜您獲得虛寶序號一組！序號：11802 77密碼：

B2E8481-16931 F

**恭喜您獲得虛寶序號一組！序號：11000078密碼：**

D9F55[REDACTED]41E6

恭喜您獲得虛實序號一組！序號：11-379密碼：

CE854B8034B5F0A5

恭喜您獲得虛寶序號一組！序號：11 380密碼：

12B596 [REDACTED] F8D8E

恭喜您獲得虛寶序號一組！序號：116-81密碼：

51FBE[REDACTED]D7484

```
=====
: 用途：快速填寫 資訊工程系 工程認證問卷系統 :
: 作者：Tiger :
: 填寫問卷：課程問卷 :
: 填寫方式：全部填入中間選項 :
: 問卷網址：http://eecs.csie.nfu.edu.tw/eqis :
:
=====
```

UserName: [REDACTED]

Password:

登入成功！開始取得問卷列表

99學年度上學期 課程：視窗程式設計  
九十九學年度第一學期期中教學反應問卷<已填>  
九十九學年度第一學期期末課程問卷<未填>  
<未填>  
填寫中....

填寫完成!

填寫中....

填寫完成!

99學年度上學期 課程：網路程式設計  
九十九學年度第一學期期中教學反應問卷<已填>  
九十九學年度第一學期期末課程問卷<未填>  
<未填>  
填寫中....

填寫完成!

填寫中....

填寫完成!

99學年度上學期 課程：資料結構  
九十九學年度第一學期期中教學反應問卷<已填>  
九十九學年度第一學期期末課程問卷<未填>  
<未填>  
填寫中....

填寫完成!

填寫中....

填寫完成!

99學年度上學期 課程：電子學<一>  
九十九學年度第一學期期中教學反應問卷<已填>  
九十九學年度第一學期期末課程問卷<未填>  
<未填>  
填寫中....

填寫完成!

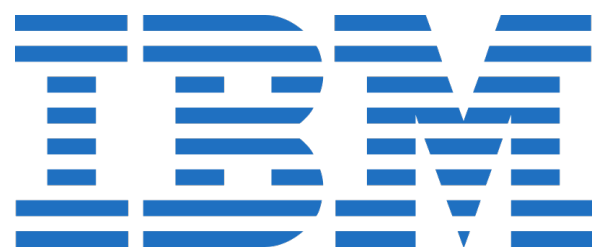
除了算期中考還有幾天和發射火箭外

也有這些公司在用 Python



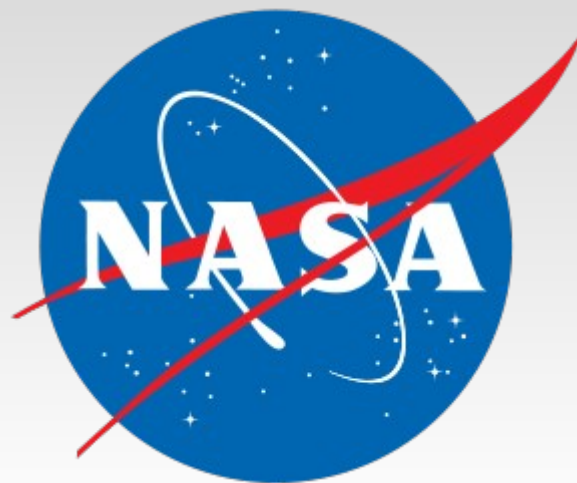
# Google





**Seagate**





UBS

PIXAR

ANIMATION STUDIOS

也因為

可以快速驗證想法

也能夠

以其他程式語言擴充



Jython

IronPython

也因為

容易寫出高維護性的程式

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
    printf("Hello, world!\n");
```

```
    for(int i = 0; i < 10; i++)
```

```
        printf("%d\n", i);
```

```
    return 0;
```

```
}
```

比方說，  
這是一段 C++ 的程式碼，  
印出 Hello, world! 及 0~9

C



```
# include<stdio.h>
```

另一種大括號的 style

```
int main() {  
    printf("Hello, world!\n");  
    for(int i = 0; i < 10; i++)  
        printf("%d\n", i);  
    return 0;  
}
```

C

```
# include<stdio.h>
```

希望在 debug 的時候不會誤認 ...

```
int main() {
```

```
    printf("Hello, world!\n");
```

```
    for(int i = 0; i < 10; i++)
```

```
        printf("%d\n", i);
```

```
    return 0;
```

```
}
```

C

```
#include<stdio.h>
```

再改一個縮排 ...

```
int main() {
```

```
printf("Hello, world!\n");
```

```
    for(int i = 0; i < 10; i++)
```

```
        printf("%d\n", i);
```

```
    return 0;
```

```
}
```

C

根本是在整人！

```
#include<stdio.h>
```

```
int main() {printf("Hello,  
world!\n"); for(int i = 0; i < 10;  
i++) printf("%d\n", i); return 0; }
```

C

唯一的寫法

```
print 'Hello, world!'  
for i in range(10):  
    print i
```



更有

豐富的 STL 與 Third Lib

最重要的！

Python 是會寫的很開心的

那我們來

安裝 Python 吧！

( 以 Windows 7 為例 )



Download Python x

www.python.org/download/

python™

» Download

# python.org/download

ABOUT »

NEWS »

DOCUMENTATION »

**DOWNLOAD »**

License

Releases

Windows

Macintosh

Other

Source

下載 »

COMMUNITY »

FOUNDATION »

CORE DEVELOPMENT »

Python Wiki

Python Insider Blog

Python 2 or 3?

Help Maintain Website

Help Fund Python

PayPal DONATE or VISA MasterCard

Non-English Resources

Release Schedule

顯示 3/29 之後的活動

Alternative Implementations

This site hosts the "traditional" implementation of Python (nicknamed CPython). A number of alternative implementations are available as well, namely

- [IronPython](#) (Python running on .NET)
- [Jython](#) (Python running on the Java Virtual Machine)
- [PyPy](#) (A fast python implementation with a JIT compiler)
- [Stackless Python](#) (Branch of CPython supporting microthreads)

## Download Python

The current production versions are [Python 2.7.1](#) and [Python 3.2](#).

Start with one of these versions for learning Python or if you want the most stability; they're both considered stable production releases.

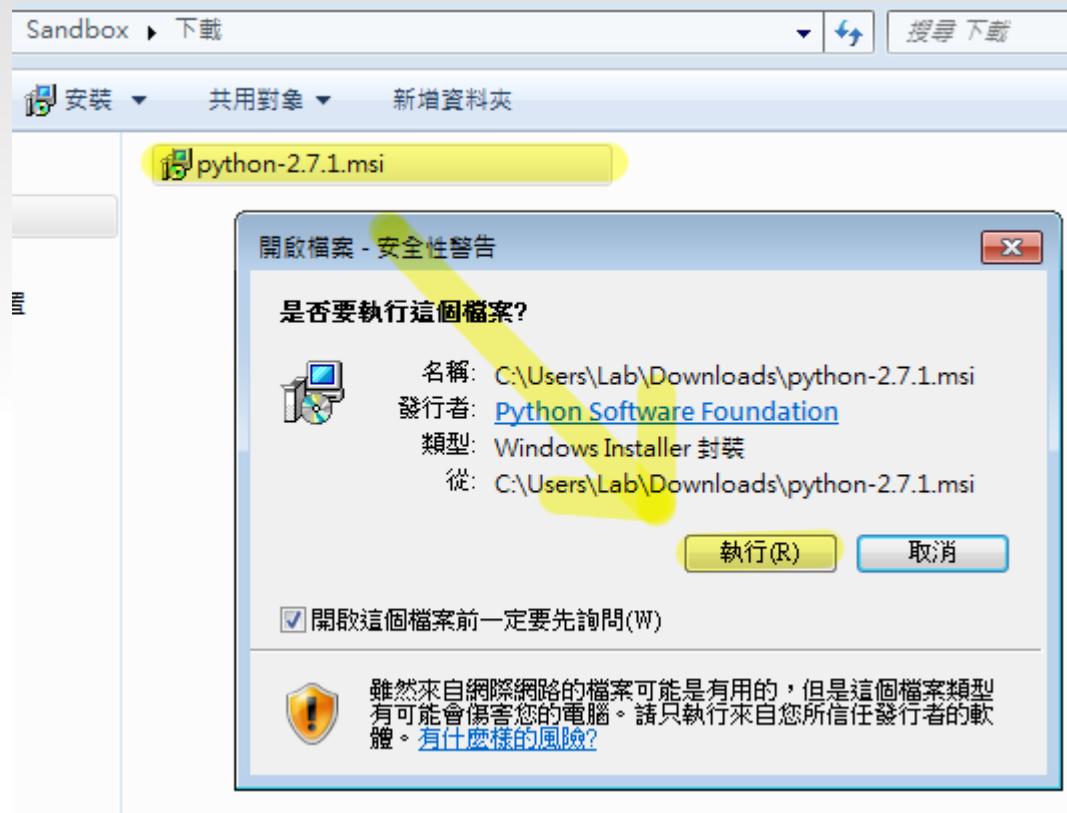
If you don't know which version to use, start with Python 2.7; more existing third party software is compatible with Python 2 than Python 3 right now.

For the MD5 checksums and OpenPGP signatures, look at the [detailed Python 2.7.1](#) page:

- [Python 2.7.1 Windows Installer](#) (Windows binary -- does not include source)
- [Python 2.7.1 Windows X86-64 Installer](#) (Windows AMD64 / Intel 64 / X86-64 binary [1] -- does not include source)
- [Python 2.7.1 Mac OS X 32-bit i386/PPC Installer](#) (for Mac OS X 10.3 through 10.6 [2])
- [Python 2.7.1 Mac OS X 64-bit/32-bit x86-64/i386 Installer](#) (for Mac OS X 10.6 [2])
- [Python 2.7.1 compressed source tarball](#) (for Linux, Unix or Mac OS X)
- [Python 2.7.1 bzipipped source tarball](#) (for Linux, Unix or Mac OS X, more compressed)

Also look at the [detailed Python 3.2](#) page:

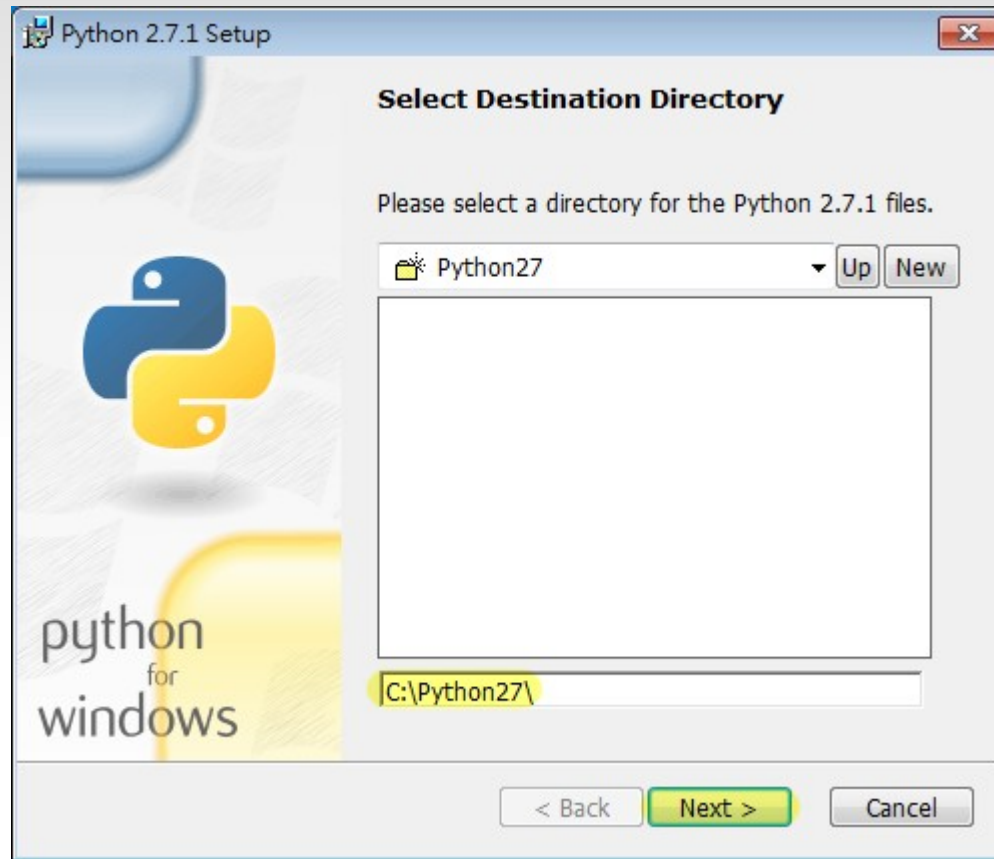
- [Python 3.2 Windows x86 MSI Installer](#) (Windows binary -- does not include source)



# 執行



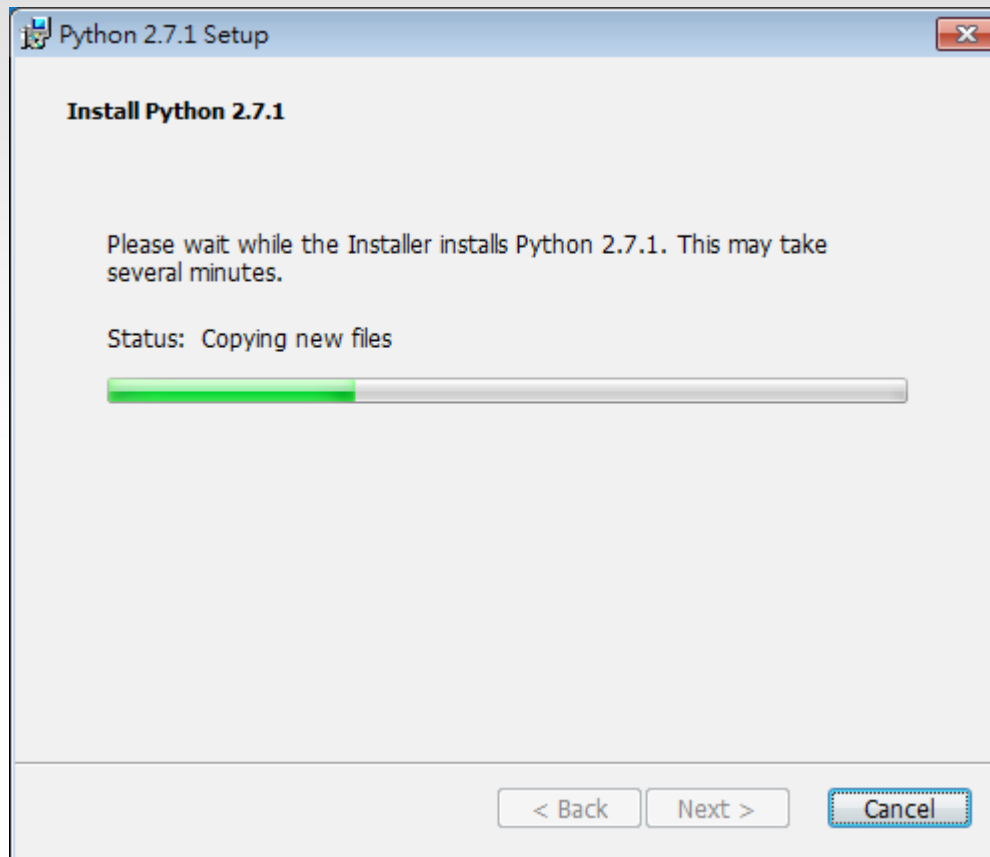
Next



Next



Next





# Finish



電腦

右鍵

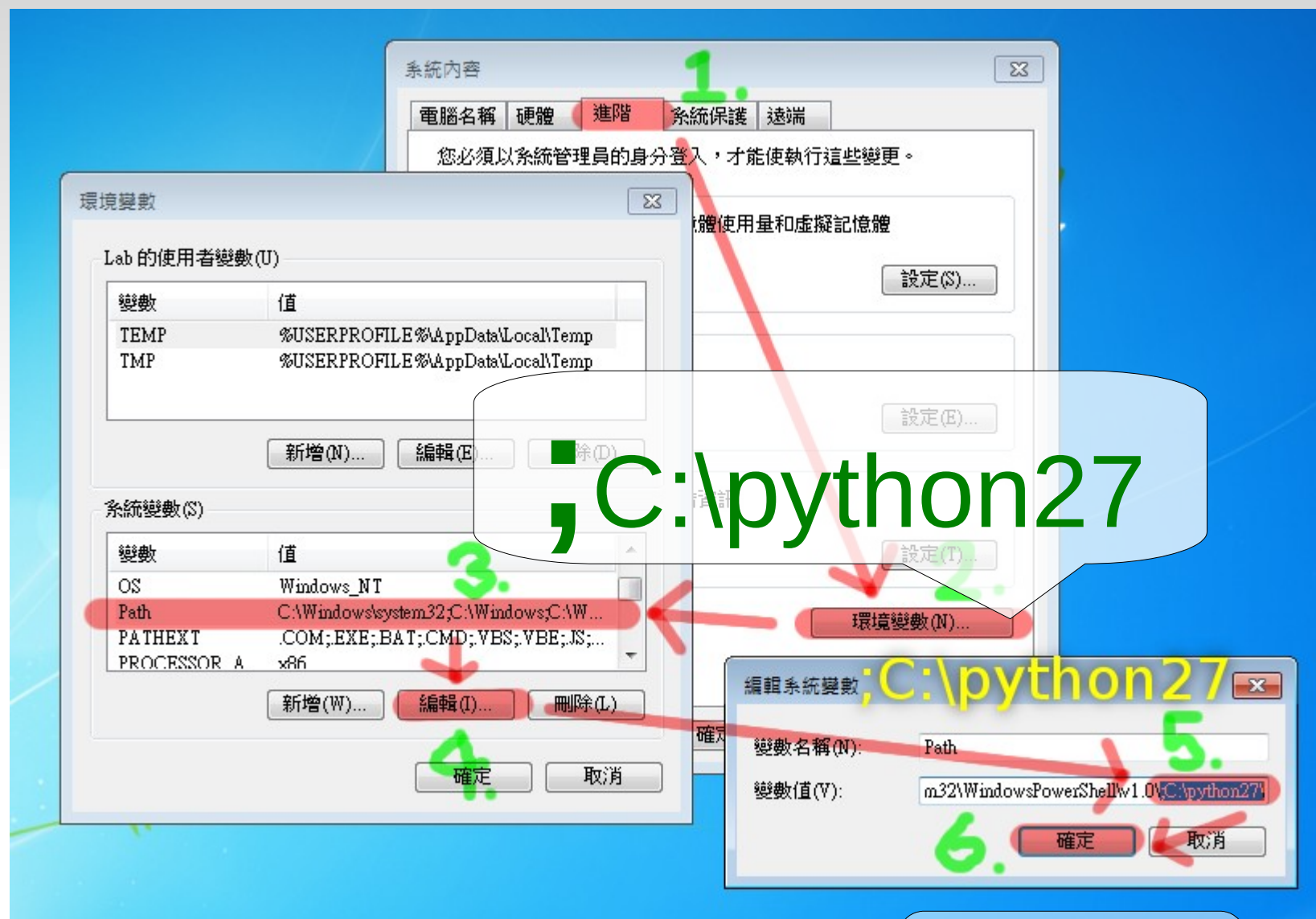
內容

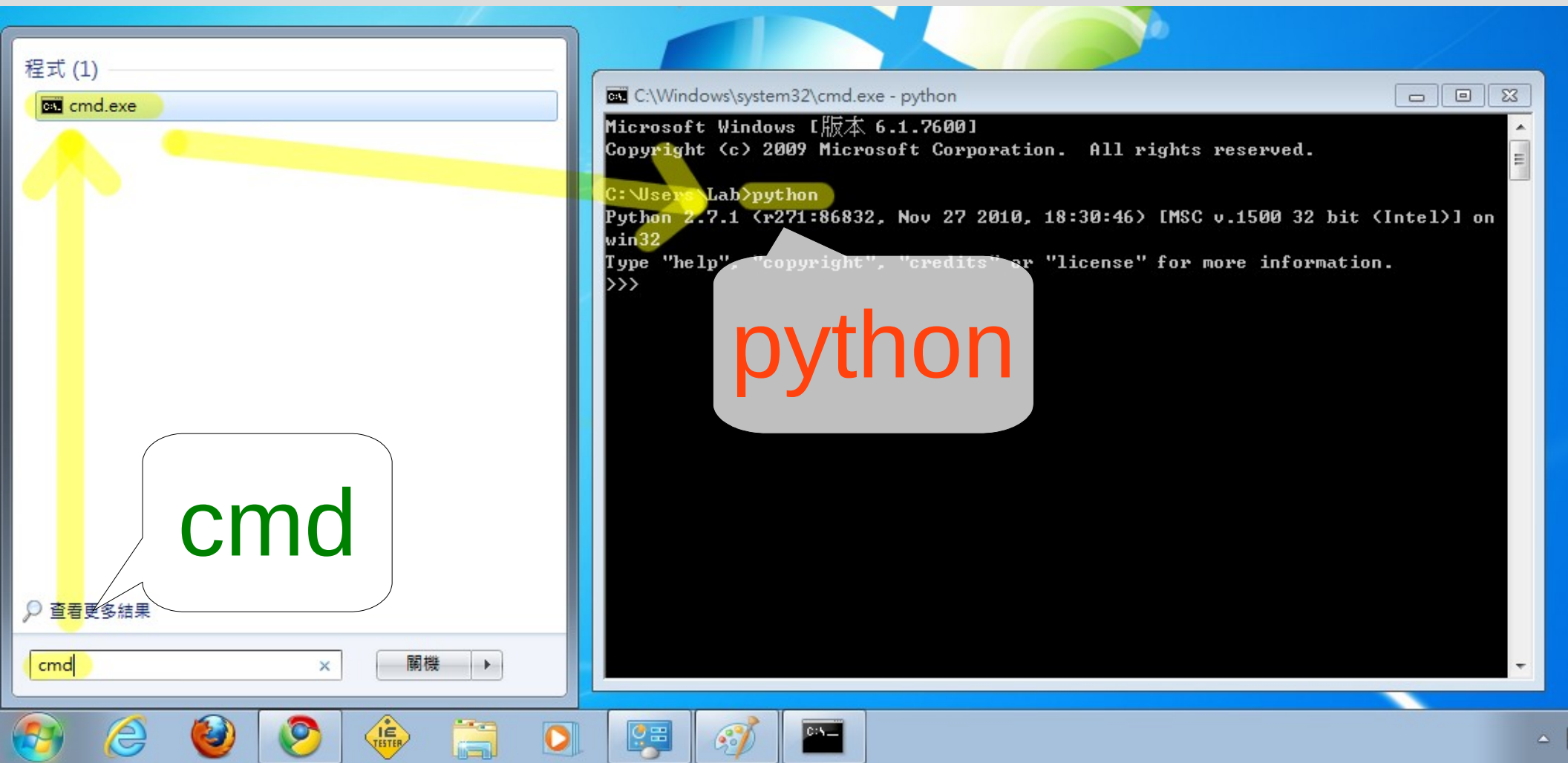




# 進階系統設定







裝好了

# 玩玩 Python Shell

請先下載今天的範例檔案：[j.mp/pyintro](http://j.mp/pyintro)

再使用 `cd` 切換到範例資料夾內



```
mosky@mosky-desktop ~/samples/photos $ ls
IMAG0074.jpg IMAG0076.jpg IMAG0078.jpg IMAG0080.jpg IMAG0082.jpg IMAG0084.jpg
IMAG0075.jpg IMAG0077.jpg IMAG0079.jpg IMAG0081.jpg IMAG0083.jpg
mosky@mosky-desktop ~/samples/photos $ python
Python 2.6.6 (r266:84292, Sep 15 2010, 15:52:39)
[GCC 4.4.5] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import os
>>> photos = os.listdir('.')
>>> photos
['IMAG0083.jpg', 'IMAG0079.jpg', 'IMAG0081.jpg', 'IMAG0075.jpg', 'IMAG0078.jpg', 'IMAG0082.jpg', 'IMAG0077.jpg', 'IMAG0080.jpg', 'IMAG0084.jpg', 'IMAG0076.jpg', 'IMAG0074.jpg']
>>> photos.sort()
>>> photos
['IMAG0074.jpg', 'IMAG0075.jpg', 'IMAG0076.jpg', 'IMAG0077.jpg', 'IMAG0078.jpg', 'IMAG0079.jpg', 'IMAG0080.jpg', 'IMAG0081.jpg', 'IMAG0082.jpg', 'IMAG0083.jpg', 'IMAG0084.jpg']
>>> for i, name in enumerate(photos):
...     os.rename(name, 'scenery-%03d.jpg' % i)
...
>>>
mosky@mosky-desktop ~/samples/photos $ ls
scenery-000.jpg scenery-003.jpg scenery-006.jpg scenery-009.jpg
scenery-001.jpg scenery-004.jpg scenery-007.jpg scenery-010.jpg
scenery-002.jpg scenery-005.jpg scenery-008.jpg
mosky@mosky-desktop ~/samples/photos $
```

按 Ctrl+Z 或 Ctrl+D 離開

```
>>> 2 + 2 * 2 - 1
```

```
>>> (2 + 2) * 2 - 1
```

```
>>> 2 ** 10
```

```
>>> - 24
```

```
>>> _ ** 1000
```



```
>>> 5 / 2
```

```
>>> 5.0 / 2
```

```
>>> 5.0 // 2
```

```
>>> 5 % 2
```

```
>>> income = 100
```

```
>>> expense = 120
```

```
>>> income - expense
```





```
>>> s = 'string'
```

```
>>> s[::-1]
```

```
>>> s[-3:]
```

```
>>> s[:2]
```

```
>>> s[-1]
```



```
>>> a = ['hey', 123]; a
>>> b = ("hey"); b
>>> c = ('hey', ); c
>>> d = {
...     u"小明": 100,
...     u'大明': 90,
... }
...
>>> d; d[u'小明']
```



接著來比較 Python 與 C++ ，

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

```
1 #include<stdio.h>
2
3 int main(){
4
5     int i, j;
6
7     for( i = 1; i < 20; i+=2){
8         for( j = 0; j < (20-i)/2; j++)
9             printf(" ");
10        for( j = 0; j < i; j++)
11            printf("*");
12        printf("\n");
13    }
14
15    return 0;
16 }
```

C

```
for i in range(1, 20, 2):  
    print ('*' * i).center(20)
```



顯示檔案內容

```
1 #include<stdio.h>
2
3 int main()
4 {
5     FILE *fp;
6     char c;
7
8     fp = fopen( "fin.txt", "r");
9
10    for( ; ; ){
11        c = fgetc(fp);
12        if( c == EOF) break;
13        printf("%c", c);
14    }
15 }
```

C



```
print open( 'fin.txt' ).read()
```



# HTTP Server

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
    // . . . . .
```

```
}
```

C

```
$python -m 'SimpleHTTPServer'
```



OK ,

進階到檔案！

完整度比較高的 IDE

Eclipse + PyDev

但今天我們用

Notepad++ (Windows)

<http://notepad-plus-plus.org/release/>

Gedit (Linux)

就足夠了！

打開

sample/helloworld.py



第 4~6 行

對於 module 的 docstring

第 8~9 行

dict 的 key 只要 immutable

第 11 行

def 定義一個函數

多行敘述記得都要加上：

第 12 行

函數的 docstring

第 26 行

raise 一個自訂的錯誤

第 28 行

短 if 的用法

第 29 行

for item in iterable

第 32 行

回傳結果



第 42 行

while 的用法

第 57 行

迴圈的 else 子句

第 63 行

當不做為模組引入時執行

第 64 行

引入 `sys` 這個模組

第 65 行

以 len 取得參數的長度

第 75 行

將所有字串轉成數字

最後一行

將序列拆成參數傳入

## 實作 1/2

請寫一個能夠以 1. 身高 (公分) 和 2. 體重 (公斤) 計算 BMI 值的函數，在以傳參數的方式執行時會對計算結果提出建議。



## 實作 2/2

請寫一個 script，接收兩個參數，第 1 是標題，第 2 是路徑，將路徑中的所有檔案，按照原始檔名順序，重新命名成 [ 標題 ]-[1 ... n].[ 原始副檔名 ]。

hint : `os`, `split`, [docs.python.org](https://docs.python.org)

以上是今天的部份，如果還有興趣

歡迎報名後續的課程！

Python 還有 ...

Built-in Function, Comprehension /  
String, Unicode, Regular Expression

有 ...

Function & Module, File IO & Open  
URL / Closure, Iterator & Generator,  
Functional Programming

還有 ...

Object-Oriented, Exception

Q & A

謝謝！