吳老師教學中心

吳老師教學中心

加入吳老師教學中心@Line:https://line.me/R/ti/p/%40justinwu

- 吳老師教學中心粉絲團

- www.facebook.com/chaiyenwu/
- 吳老師教學會員中心
- http://www.justinwu.com.tw/

- 智慧手機App應用程式,iOS,Android.人工智慧,Fintech,大數據，Python,討論區

- https://www.facebook.com/groups/2125450901013468/

吳老師教學中心

- 吳老師教學中心
- http://www.justinwu.com.tw
- Facebook粉絲專區
- https://www.facebook.com/chaiyenwu/
- Facebook會員專區
- https://www.facebook.com/groups/2125450901013468/

python

Python 程式設計教學
---從零開始徹底精通Python,
親手實作網站擷取與資料分析,
精通Selenium

Python

網站擷取與資料分析

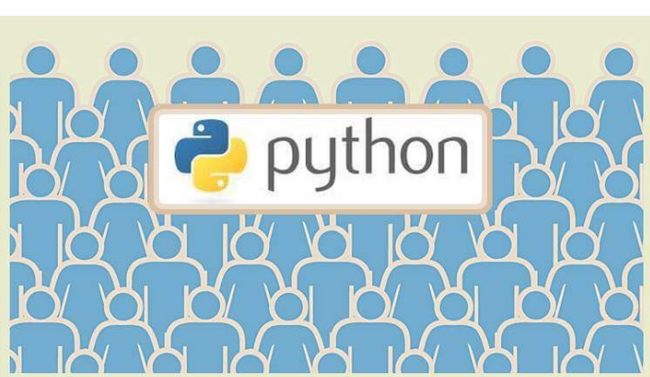
吳老師 - 為 udemy上 python
學生最多的老師, 已著有電腦
書籍１０餘本

python

吳老師教學中心

www.facebook.com/chaiyenwu/

Python程式語言,大數據,人工智慧,機器學習,深度學習



Python一天學會

吳佳諺 老師

python

1.下載及安裝Python軟體
2.Python直譯器與計算機
3.資料結構
4.控制結構
5.函數
6.類別
7.繼承
8.異常或錯誤處理
9.使用matplotlib畫圖
10.網站擷取使用Python
11.行程和執行緒
12.資料及檔案處理
13.SQLite資料庫
14.人機界面及影像處理
15.Python遊戲設計

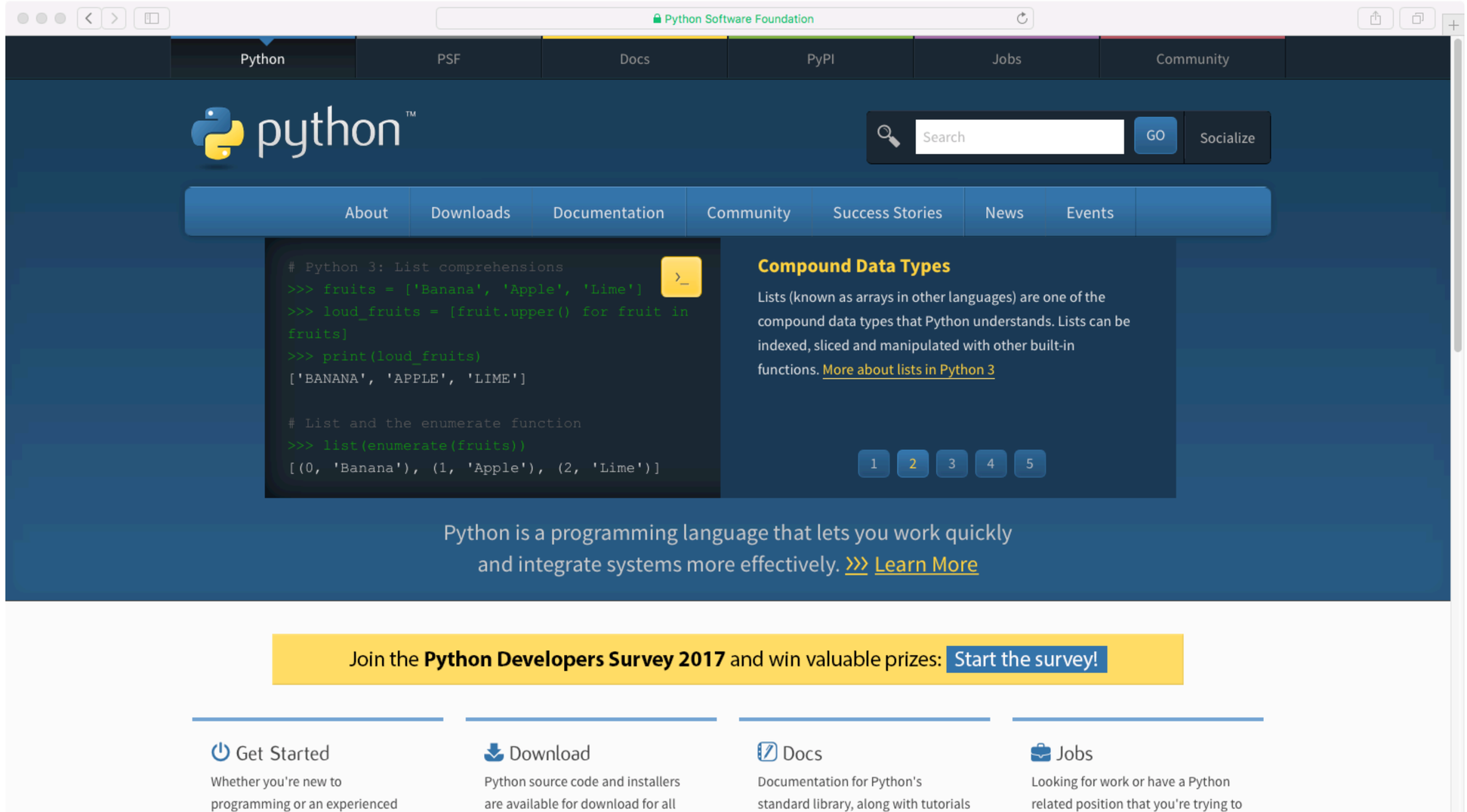
python

1.下載及安裝Python軟體

到python.org下載軟體

到Aanconda下載Python組合

到python.org下載軟體



Python安裝

- 這是在Mac上安裝完的Python Launcher

這是安裝完Python的直譯環境

Python 3.6.2 Shell

```
Python 3.6.2 (v3.6.2:5fd33b5926, Jul 16 2017, 20:11:06)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> WARNING: The version of Tcl/Tk (8.5.9) in use may be unstable.
Visit http://www.python.org/download/mac/tcltk/ for current information.

>>>
```

使用easy_install安裝pip套件

- easy_install 套件名稱
- 安裝pip
- easy_install install pip

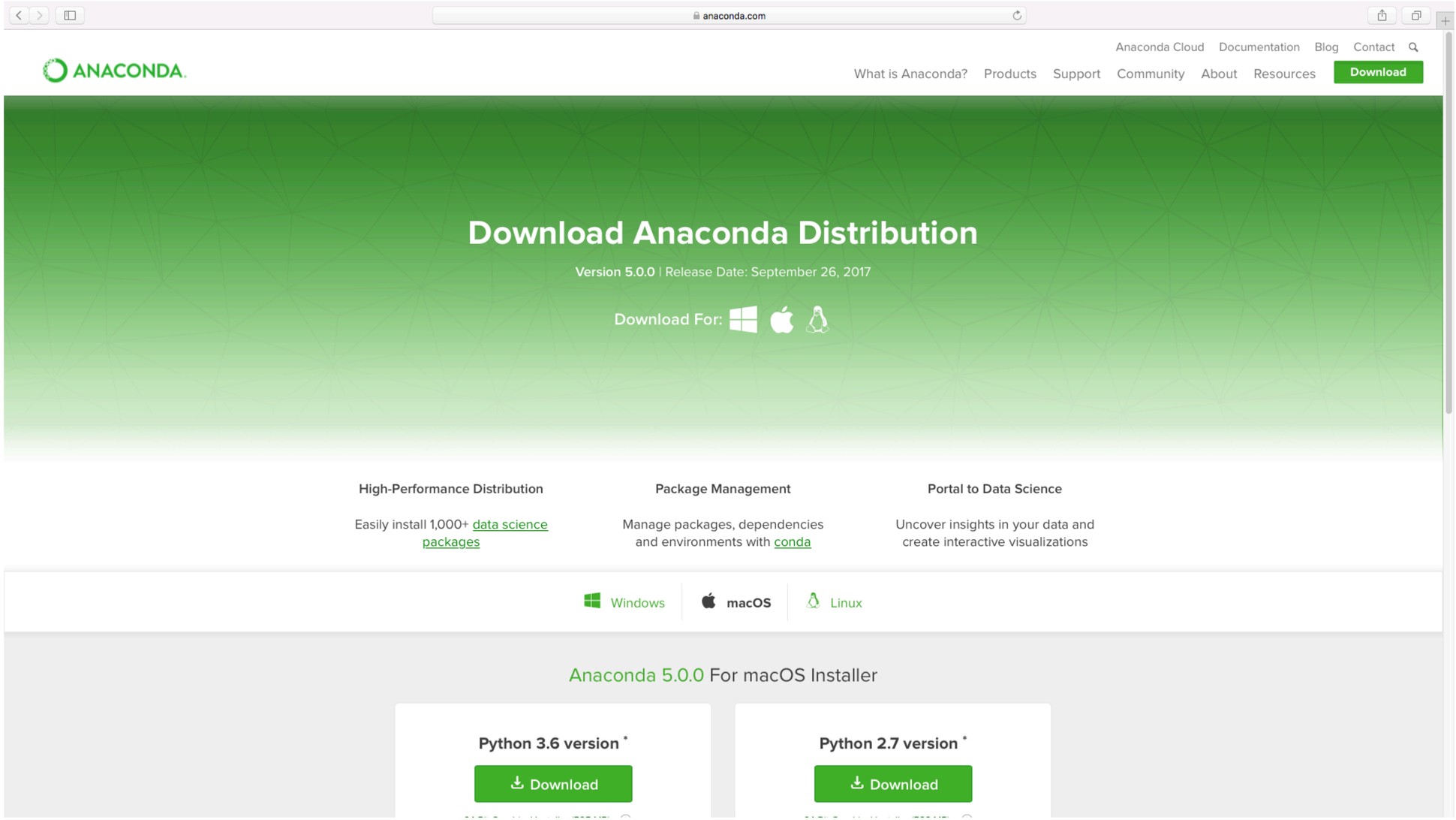
使用pip install 套件名稱

- 安裝套件
- pip install 套件名稱
- 解除安裝套件
- pip uninstall 套件名稱
- 檢視目前系統已經安裝的套件
- pip list

安裝
Anaconda,Numpy,Matplotilib

- 安裝Anaconda組合包
- 此包含Python,spyder,和科學大數據計算軟體
- 安裝Numpy,Matplotilib,Scipy

- 到Anconda下載Python組合包https://www.anaconda.com/

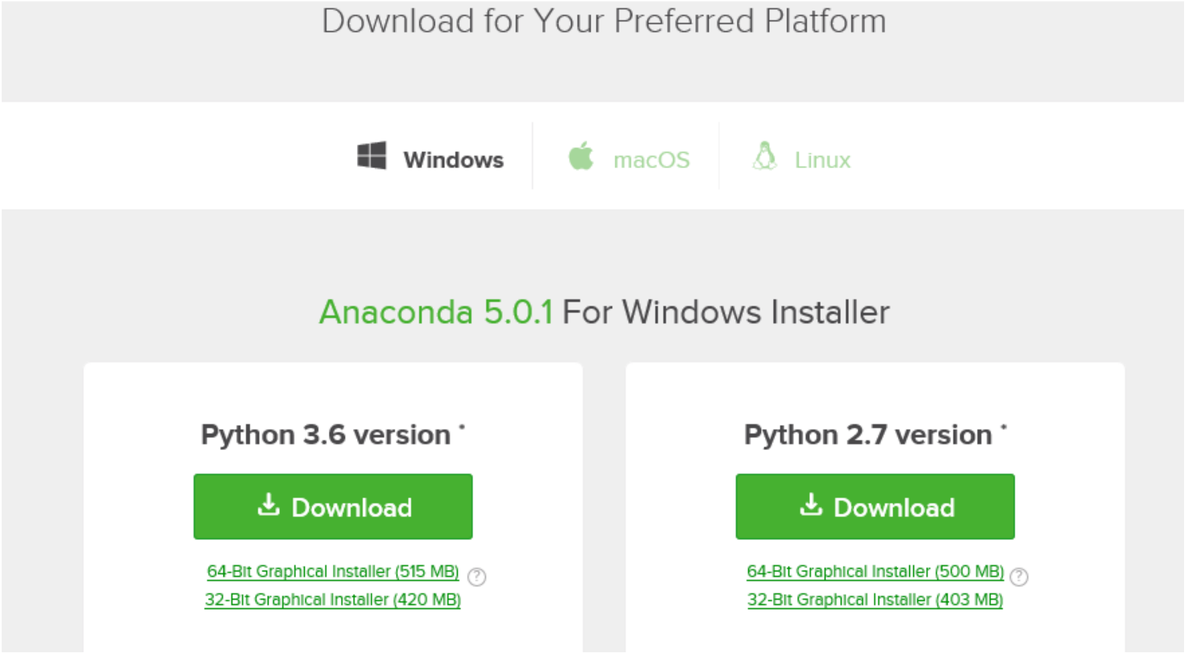


使用conda升級套件

justinwu — -bash — 80×24

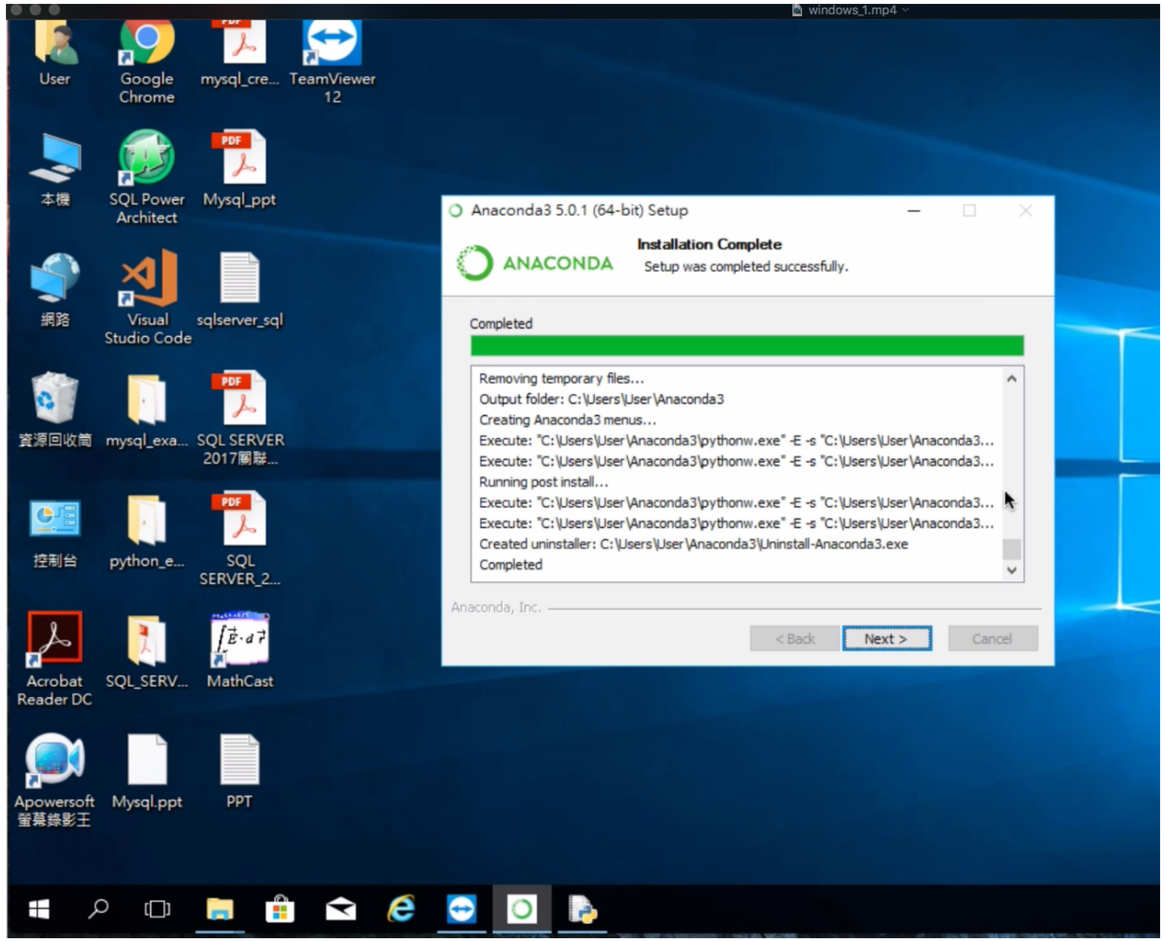
```
Last login: Mon Oct 16 07:12:00 on console
60-250-191-81:~ justinwu$ pip install scipy
Requirement already satisfied: scipy in ./anaconda3/lib/python3.6/site-packages
60-250-191-81:~ justinwu$ pip install -U scipy
Requirement already up-to-date: scipy in ./anaconda3/lib/python3.6/site-packages
60-250-191-81:~ justinwu$ conda install numpy
Fetching package metadata ...........
Solving package specifications: .


Package plan for installation in environment /Users/justinwu/anaconda3:

The following packages will be UPDATED:

    anaconda: 5.0.0-py36hd9bc8a5_0  --> custom-py36_0
    numpy:    1.13.1-py36h93d791d_2 --> 1.13.3-py36h2cdce51_0

Proceed ([y]/n)?
anaconda-custo 100% |################################| Time: 0:00:00 238.23 kB/s
numpy-1.13.3-p 100% |################################| Time: 0:00:00  11.34 MB/s
60-250-191-81:~ justinwu$
```

在Windows上安裝



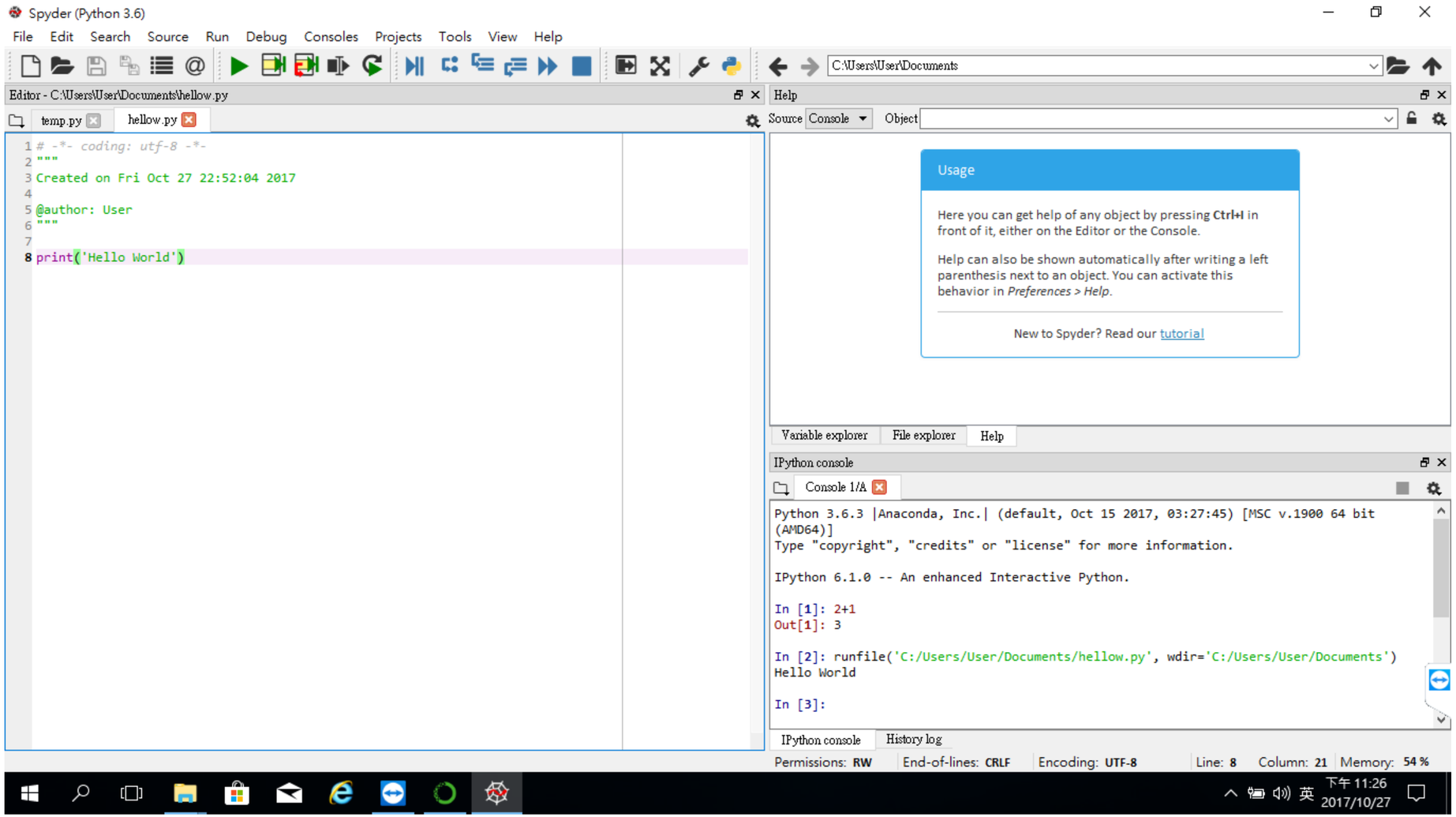
下載並且安裝Anaconda

安裝軟體Anaconda



選取Anaconda Navigator和Spider

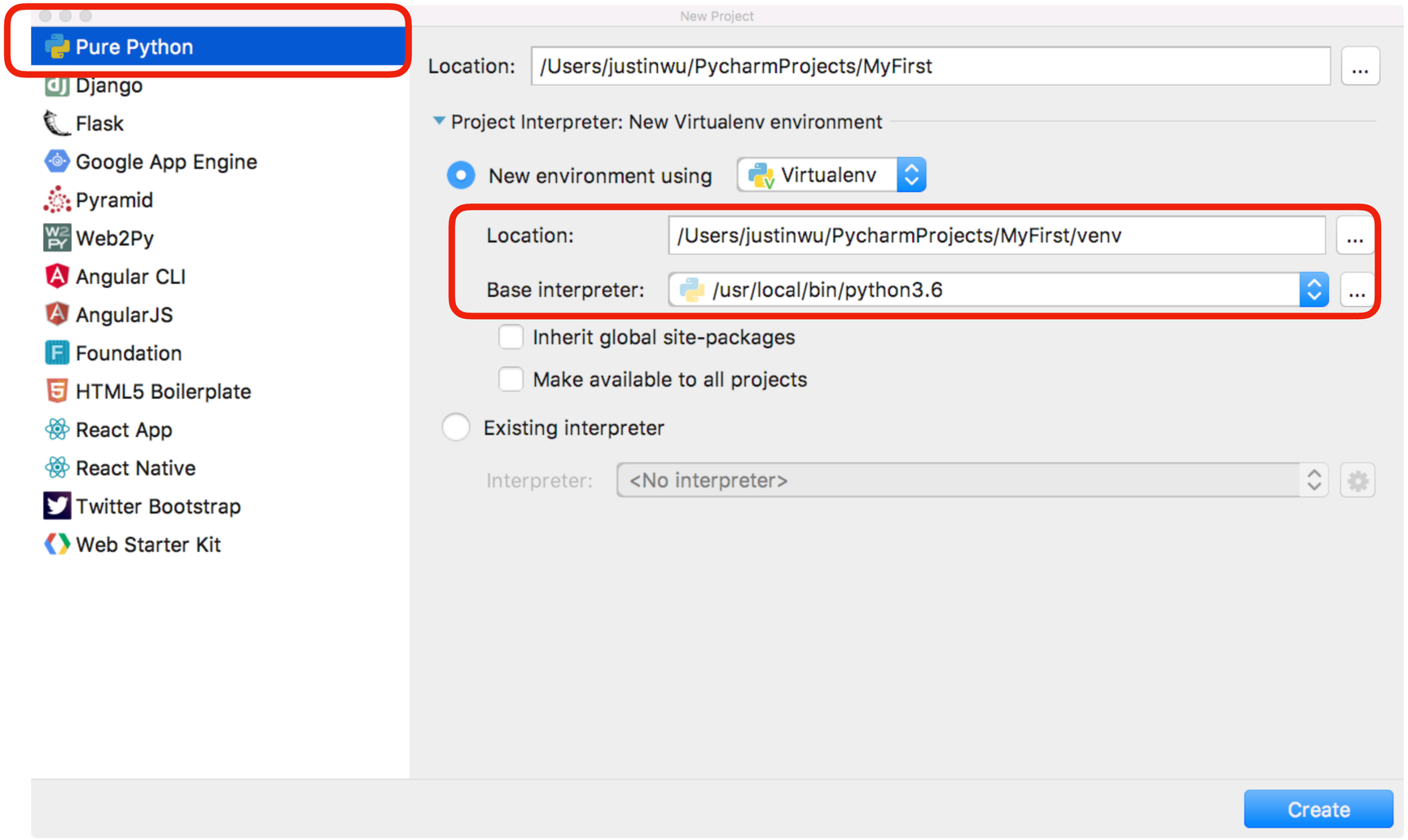
執行Spider


Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\User\Documents
Editor - C:\Users\User\Documents\hellow.py
temp.py
hellow.py
-*- coding: utf-8 -*-
"""
Created on Fri Oct 27 22:52:04 2017
@author: User
"""
print('Hello World')
Help
Source Console
Object
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial
Variable explorer
File explorer
Help
IPython console
Console 1/A
Python 3.6.3 |Anaconda, Inc.| (default, Oct 15 2017, 03:27:45) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.
IPython 6.1.0 -- An enhanced Interactive Python.
In [1]: 2+1
Out[1]: 3
In [2]: runfile('C:/Users/User/Documents/hellow.py', wdir='C:/Users/User/Documents')
Hello World
In [3]:
IPython console
History log
Permissions: RW
End-of-lines: CRLF
Encoding: UTF-8
Line: 8
Column: 21
Memory: 54 %
下午 11:26
2017/10/27

1-1下載及安裝PyCharm

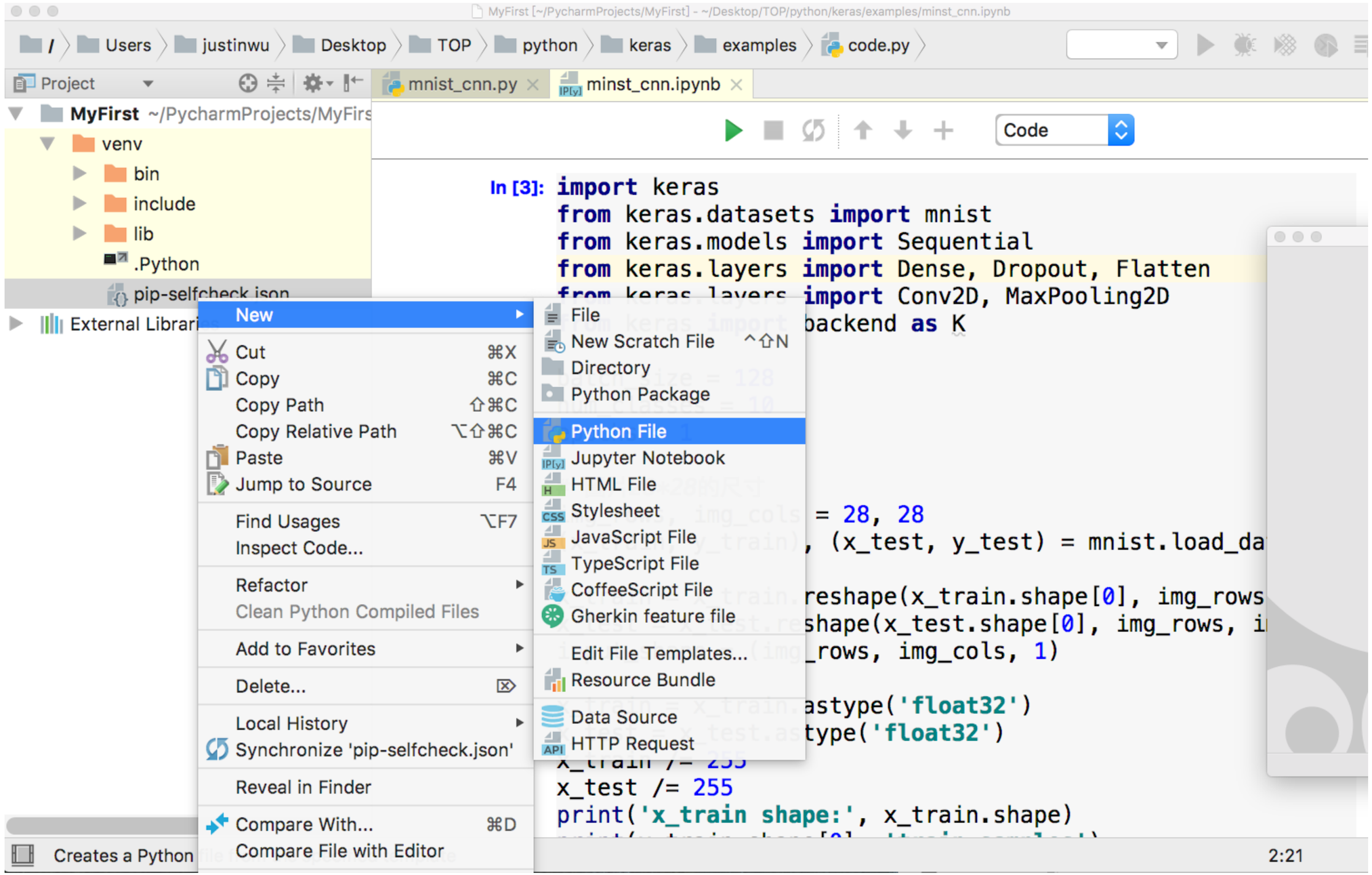
- https://www.jetbrains.com/pycharm/

安裝完Python後,設定直譯器路徑



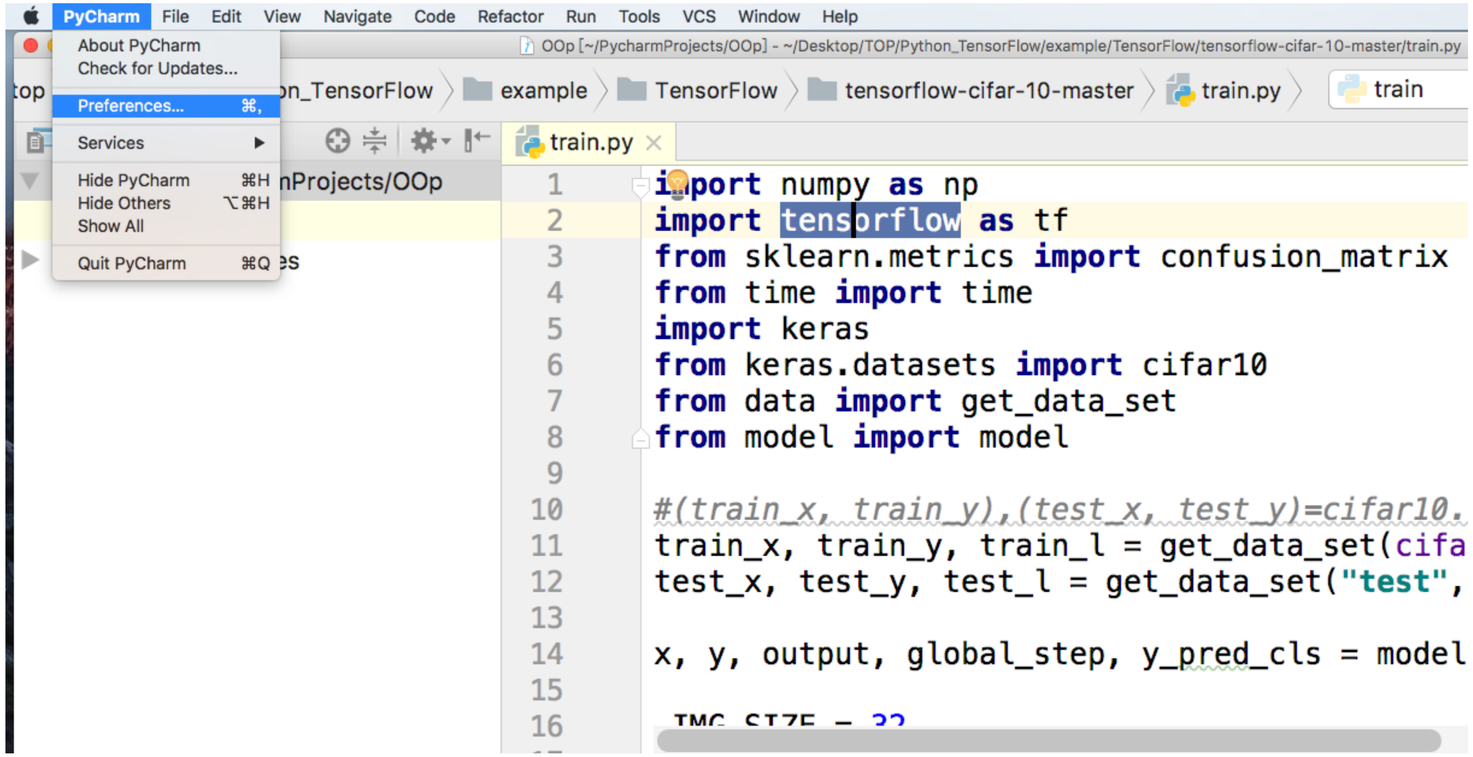
Pycharm可以使用Python和Jupyter notebook的檔案

- 我們可以新增Python的檔案
- 選取File->NEW->Python File

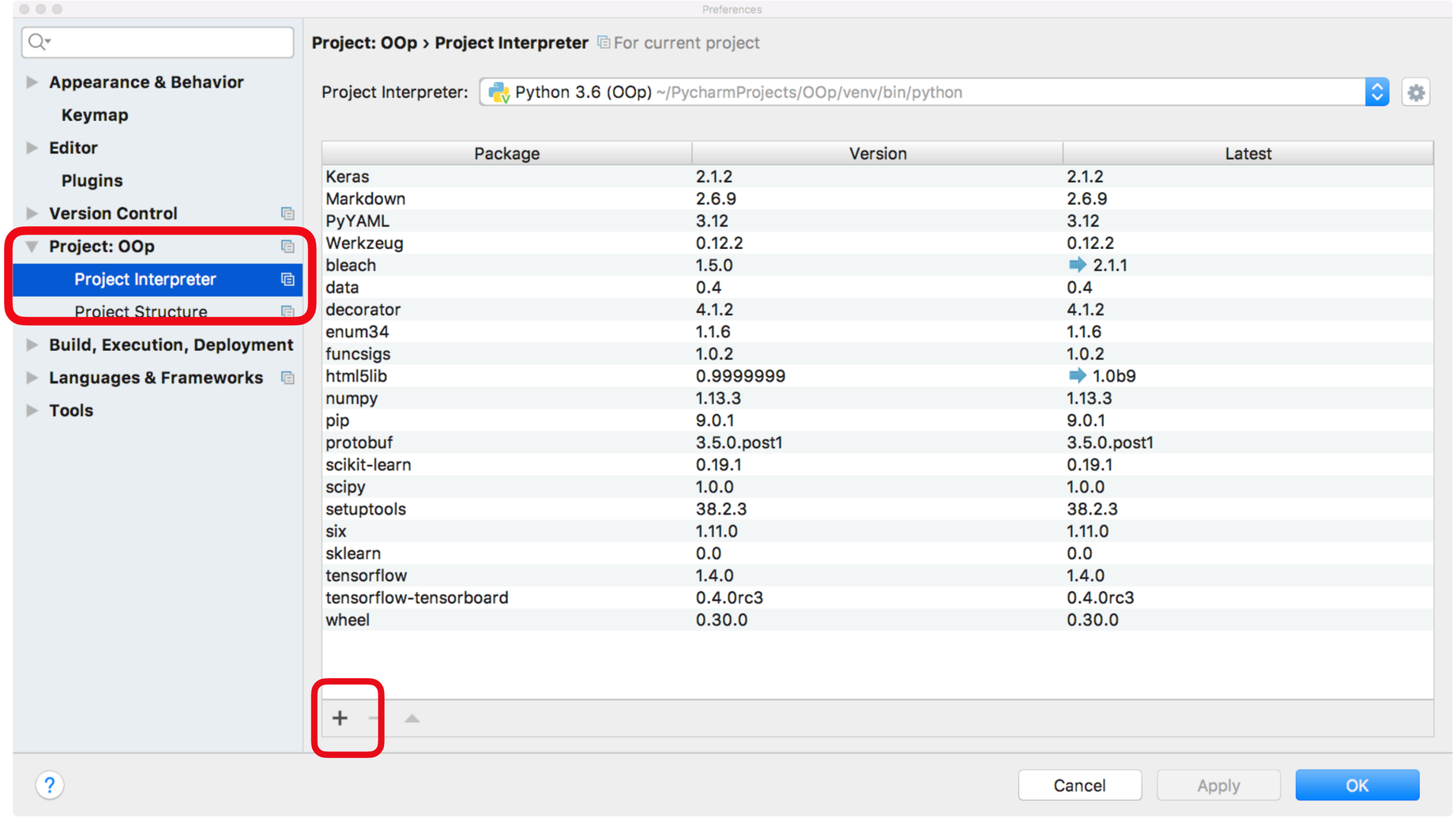

MyFirst [~/PycharmProjects/MyFirst] - ~/Desktop/TOP/python/keras/examples/minst_cnn.ipynb
/ Users justinwu Desktop TOP python keras examples code.py
Project
mnist_cnn.py
minst_cnn.ipynb
MyFirst ~/PycharmProjects/MyFirs
venv
bin
include
lib
.Python
pip-selfcheck.json
External Librari
Code
In [3]: import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D, MaxPooling2D
backend as K
= 28, 28
(x_test, y_test) = mnist.load_da
reshape(x_train.shape[0], img_rows
shape(x_test.shape[0], img_rows, i
rows, img_cols, 1)
astype('float32')
type('float32')
x_test /= 255
print('x_train shape:', x_train.shape)
New
Cut ⌘X
Copy ⌘C
Copy Path ⇧⌘C
Copy Relative Path ⌥⇧⌘C
Paste ⌘V
Jump to Source F4
Find Usages ⌥F7
Inspect Code...
Refactor
Clean Python Compiled Files
Add to Favorites
Delete...
Local History
Synchronize 'pip-selfcheck.json'
Reveal in Finder
Compare With... ⌘D
Compare File with Editor
File
New Scratch File ^⇧N
Directory
Python Package
Python File
Jupyter Notebook
HTML File
Stylesheet
JavaScript File
TypeScript File
CoffeeScript File
Gherkin feature file
Edit File Templates...
Resource Bundle
Data Source
HTTP Request
Creates a Python
2:21

1-2在Pycharm新增函式庫

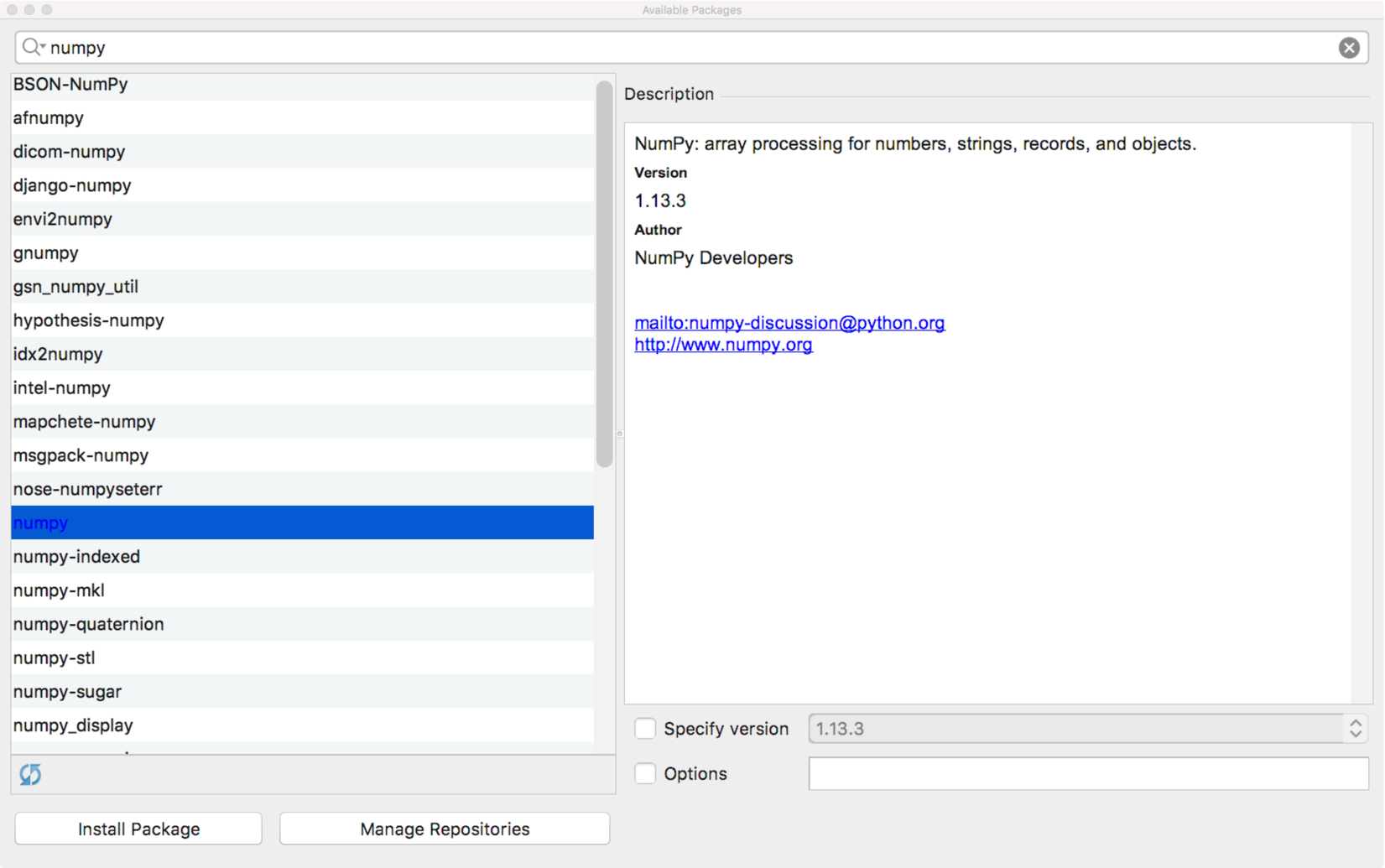
- 選取PyCharm->Preference



選取直譯器，選取＋


Preferences
Appearance & Behavior
Keymap
Editor
Plugins
Version Control
Project: OOp
Project Interpreter
Project Structure
Build, Execution, Deployment
Languages & Frameworks
Tools
Project: OOp › Project Interpreter For current project
Project Interpreter: Python 3.6 (OOp) ~/PycharmProjects/OOp/venv/bin/python
Package Version Latest
Keras 2.1.2 2.1.2
Markdown 2.6.9 2.6.9
PyYAML 3.12 3.12
Werkzeug 0.12.2 0.12.2
bleach 1.5.0 2.1.1
data 0.4 0.4
decorator 4.1.2 4.1.2
enum34 1.1.6 1.1.6
funcsigs 1.0.2 1.0.2
html5lib 0.9999999 1.0b9
numpy 1.13.3 1.13.3
pip 9.0.1 9.0.1
protobuf 3.5.0.post1 3.5.0.post1
scikit-learn 0.19.1 0.19.1
scipy 1.0.0 1.0.0
setuptools 38.2.3 38.2.3
six 1.11.0 1.11.0
sklearn 0.0 0.0
tensorflow 1.4.0 1.4.0
tensorflow-tensorboard 0.4.0rc3 0.4.0rc3
wheel 0.30.0 0.30.0
Cancel
Apply
OK

選取函式庫，再install Package



2.Python直譯器與計算機

- Mac電腦/usr/local/bin
- Windows電腦C:\python36
- set path=%path%;C:\python36

輸入python執行

```
$ python
Python 3.6.2 |Anaconda, Inc.| (default,
Sep 21 2017, 18:29:43)
[GCC 4.2.1 Compatible Clang 4.0.1 (tags/
RELEASE_401/final)] on darwin
Type "help", "copyright", "credits" or
"license" for more information.
>>>
```

UTF-8編碼

-*- coding: encoding -*-

這是設定utf-8-*-編碼

#-*- coding: utf-8 -*-

註解

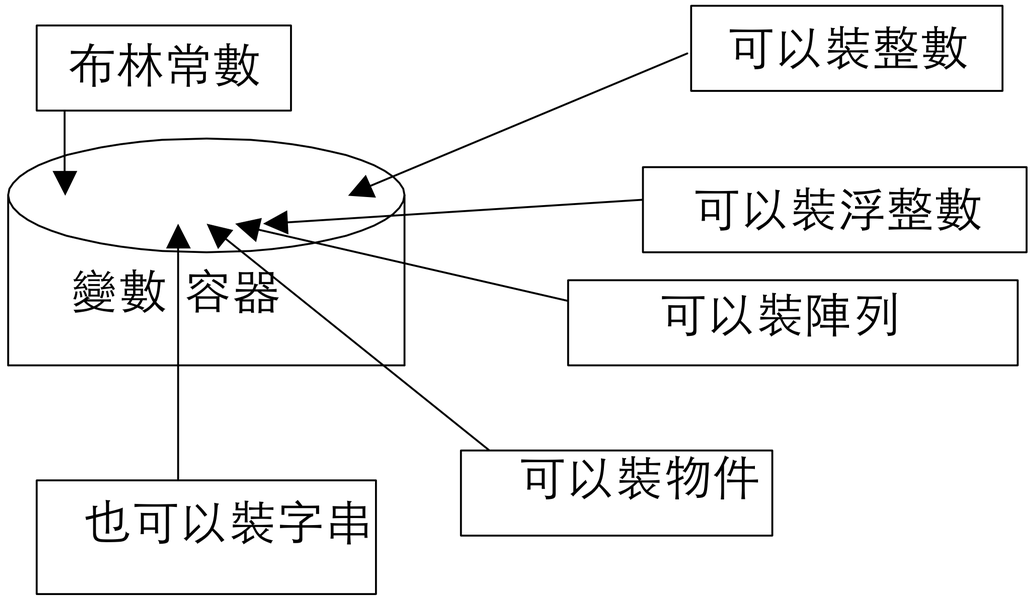
- #是註解符號

python計算機

```
>>> 1+2
3
>>> 2-1
1
>>> 3*2
6
>>> 3/2
1.5
>>> 3%2
1
>>> 50-5/6
49.166666666666664
>>> (50-5*6)/4
5.0
>>> 5**2
25
>>> 2**5
32
>>>
```

2-1簡單的程式

- Python支援的資料型態:
- 浮整數(floating point)
- 字元(char)
- 整數(integer)
- 物件(object)
- 布林常數(Boolean)
- 空值(null)
- 字串(string)。



- 範例：Circle.py
- 第一行宣告了pi的資料型態為雙精度浮點數，並且給予初始值3.14159。
- 第二行宣告了radius半徑的資料型態為雙精度浮點數。
- 第三行宣告了area面積的資料型態為雙精度浮點數。

Circle.py

```
1  pi=3.14159
2  radius=10.0
3  area=radius*radius*pi
4  print(area)
5
6
7
```

```
ircle
/Users/justinwu/PycharmProjects/Pyt
314.159

Process finished with exit code 0
```

2-2識別名稱

- 每一個變數都有識別名稱。
- 我們變數宣告時，就是給該變數一個識別名稱。
- 所有的識別名稱是由字元、數字、下底線(_)所組成。
- Python是有大小寫的區隔，a和A是不一樣的變數。

2-3數值資料型態與運算子

- 變數a=1+1,c，c=5，變數c的值為5在這個中
- a=1+1是一個運算式
- c=5是一個運算式
- +和=是運算子，變數a和數值5，和數值1是運算元。
- 運算式就是由運算子和運算元所組成。

運算子優先順序,先乘除後加減

Arithmatic.py

```python
a = 1+1
b = 2-1
div=6/2
pi=3.14159
radius=10
area=radius*radius*pi
c=5
x=2*3+4/2+1-3%2+5
print(a)
print(b)
print(div)
print(area)
print(c)
print(x)
```

thmatic

```
/Users/justinwu/PycharmProjects/Pytho
2
1
3.0
314.159
5
13.0
```

2-4運算子結合優先順序

Arithmatic.py × operator_preference.py ×

```python
x=2**3+5*2-5/5+5%2
y=1<<2
print(x)
print(y)
b=3
c=3
d=b&c
print(d)
e=b^c
print(e)
f=b|c
print(f)
g=1<5
print(g)
h=1>5
print(h)
```

```
18.0
4
3
0
3
True
False
```

3.資料結構

- 變數
- 運算式與運算子
- 串列
- 堆疊
- 佇列

變數

- 資料型態
  - 整數
  - 浮點數
  - 字串

```python
#!/usr/bin/env python3
#_*_coding:utf-8_*_
"""
Created on Thu Oct 26 07:40:42 2017

@author: justinwu
"""
#這是註解

#這是註解
1+2
#print(1+2)
x=2-1
print(x)
y=3+2
print(y)
z=3.2*2
print(z)
str='大家好'
print(str)
str2='Python'
mychar=str[0]
print(mychar)
```

Name	Type	Size	
mychar	str	1	大
str	str	1	大家好
str2	str	1	Python
x	int	1	1
y	int	1	5
z	float	1	6.4

Console 1/A

```
In [28]: runfile('/Users/ju
wdir='/Users/justinwu/Deskt
1
5
6.4
大家好
大

In [29]:
```

運算式與運算子

- 運算式是由運算子與運算元組成
- +加-減*乘/除是運算子,先乘除後加減的結合優先順序
- 運算元是變數,數字,字串和資料結構
- =是分配符號,將右邊的值分配給左邊變數

```python
#!/usr/bin/env python3
#_*_coding:utf-8_*_
"""
Created on Thu Oct 26 07:40:42 2017

@author: justinwu
"""
#這是註解

#這是註解
1+2
#print(1+2)
x=2-1
print(x)
y=3+2
print(y)
z=3.2*2
print(z)
str='大家好'
print(str)

```

Name	Type	Size	
str	str	1	大家好
x	int	1	1
y	int	1	5
z	float	1	6.4

Console 1/A

```
In [22]: runfile('/Users/j
wdir='/Users/justinwu/Desk
1
5
6.4
大家好

In [23]:
```

串列

```
>>> fruits = ['orange','apple','pear','banana','kiwi','apple','banana']
>>> fruits.count('apple')
2
>>> fruits.index('banana')
3
>>> fruits.index('banana',4)#Finding next banana starting a position 4
6
>>> fruits.reverse()
>>> fruits
['banana', 'apple', 'kiwi', 'banana', 'pear', 'apple', 'orange']
>>> fruits.append('grape')
>>> fruits
['banana', 'apple', 'kiwi', 'banana', 'pear', 'apple', 'orange', 'grape']
>>> fruits.sort()
>>> fruits
['apple', 'apple', 'banana', 'banana', 'grape', 'kiwi', 'orange', 'pear']
>>> fruits.pop()
'pear'
>>>
```

堆疊

Python 3.6.2 Shell

```
Python 3.6.2 (v3.6.2:5fd33b5926, Jul 16 2017, 20:11:06)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> WARNING: The version of Tcl/Tk (8.5.9) in use may be unstable.
Visit http://www.python.org/download/mac/tcltk/ for current information.

>>> stack=[3,4,5]
>>> stack.append(6)
>>> stack.append(7)
>>> stack
[3, 4, 5, 6, 7]
>>> stack.pop()
7
>>> stack.pop()
6
>>> stack
[3, 4, 5]
>>>
```

佇列

```python

from collections import deque
queue = deque(['阿呆','Eric','John','Michael','小寶','小文'])
queue.append("Terry")
queue.append("Graham")
print(queue.popleft())
print(queue.popleft())
print(queue)

```

```
In [1]: runfile('/Users/justinwu/Desktop/queue.py', wdir='/Us
阿呆
Eric
deque(['John', 'Michael', '小寶', '小文', 'Terry', 'Graham'])

In [2]:
```

數組tuple,集合set和字典

- 可以用數組tuple來儲存固定的元素,使用小括號()來建立一數組tuple
- 集合的元素放置沒有按照順序,可以使用{}大括號來建立一集合Set
- 集合加上索引就是字典{索引:值}

Tuple數組

- 也可以從字串中建立數組
- tp5 = tuple(‘Ivy Lin’)
- 從數組得到串列
- list1 = list(tp5)

```python

tp1=()
print(tp1)
tp2=(1,2,3,4,5,6,7,8)
print(tp2)
print(sum(tp2))
print('--------------------')
print(tp2[2:5])#切割運算子
print(tp2[-1])
tp3 =tuple([2*x for x in range(1,8)])
print(tp3)
print('----------------')
tp4=tuple('Ivy Lin')
print(tp4)
tp5=("John",'小寶','小文')
print(tp5)
print(len(tp5))
print(tp4+tp5)
print('--------------')
tp6=tuple([1,2,3,4,5,6,7,8,9])
print(tp6)
print(max(tp6))
print(min(tp6))

```

```
In [6]: runfile('/Users/justinwu/Desktop/tuple.py', wdir='
Desktop')
()
(1, 2, 3, 4, 5, 6, 7, 8)
36
--------------------
(3, 4, 5)
8
(2, 4, 6, 8, 10, 12, 14)
----------------
('I', 'v', 'y', ' ', 'L', 'i', 'n')
('John', '小寶', '小文')
3
('I', 'v', 'y', ' ', 'L', 'i', 'n', 'John', '小寶', '小文')
--------------
(1, 2, 3, 4, 5, 6, 7, 8, 9)
9
1

In [7]:
```

```python

tp6=tuple([66,22,3,46,5,65,7,83,19])
print(tp6)
list1 = list(tp6)
list1.sort()#排序串列
print(list1)
print('-----------')
tp8=tuple(list1)
tp9=tuple(list1)
print(tp8)
print(tp8 == tp9)#比較兩個數組tuple


```

```
In [15]: runfile('/Users/justinwu/
Desktop')
(66, 22, 3, 46, 5, 65, 7, 83, 19)
[3, 5, 7, 19, 22, 46, 65, 66, 83]
-----------
(3, 5, 7, 19, 22, 46, 65, 66, 83)
True

In [16]:
```

Set集合

- 集合(set)用來儲存沒有重複的元素.
- 集合的元素是不可以複製的,元素放置也沒有按照順序
- 可以使用{}大括號來建立一集合Set

Set集合

```python

st1=set()#建立一個空集合
st2=set([1,2,3,4,5])
print(st2)
st3={'a','b','c','d','e'}
print(st3)
print('-------------------')
st3.add('f')
print(st3)
st3.remove('d')
print(st3)
print('----------')
print(st3.union(st2))
st5={'a','b','c','d','e'}
print(st3.intersection(st5))
print(st3.difference(st5))
```

```
In [30]: runfile('/Users/justinwu/Deskto
Desktop')
{1, 2, 3, 4, 5}
{'d', 'a', 'e', 'c', 'b'}
-------------------
{'d', 'f', 'a', 'e', 'c', 'b'}
{'f', 'a', 'e', 'c', 'b'}
----------
{1, 2, 3, 4, 5, 'f', 'a', 'e', 'c', 'b'}
{'b', 'a', 'c', 'e'}
{'f'}

In [31]:
```

字典

- 集合加上索引就是字典{索引:值}

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Oct 27 19:40:39 2017

@author: justinwu
"""
tel={'Justin':'0920909872','Ivy':'0922876895'}
tel['Johny']='0920885356'
print(tel)
print(tel['Johny'])
del tel['Johny']
tel['Mary']='0922865255'
print(tel)
print(list(tel.keys()))
print(sorted(tel.keys()))
print('Ivy' in tel)
print('Johny' in tel)
```

```
In [5]: runfile('/Users/justinwu/Desktop/TOP/python/
example/dictionary_1.py', wdir='/Users/justinwu/Desktop/
TOP/python/example')
{'Justin': '0920909872', 'Ivy': '0922876895', 'Johny':
'0920885356'}
0920885356
{'Justin': '0920909872', 'Ivy': '0922876895', 'Mary':
'0922865255'}
['Justin', 'Ivy', 'Mary']
['Ivy', 'Justin', 'Mary']
True
False

In [6]:
```

4.控制結構

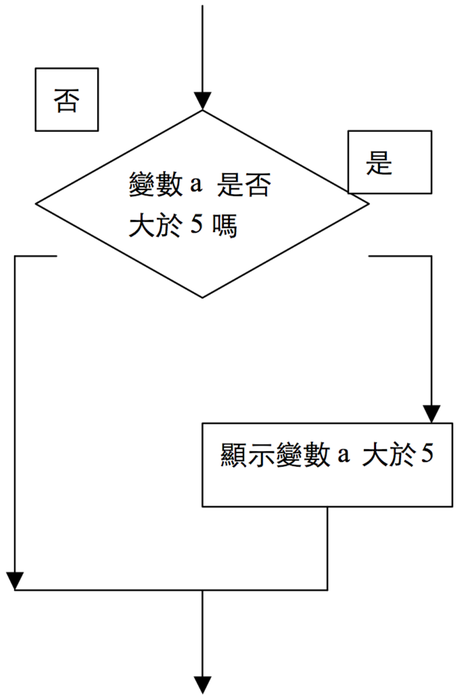
- 選取結構if
- 迴圈結構while,for

選取結構if

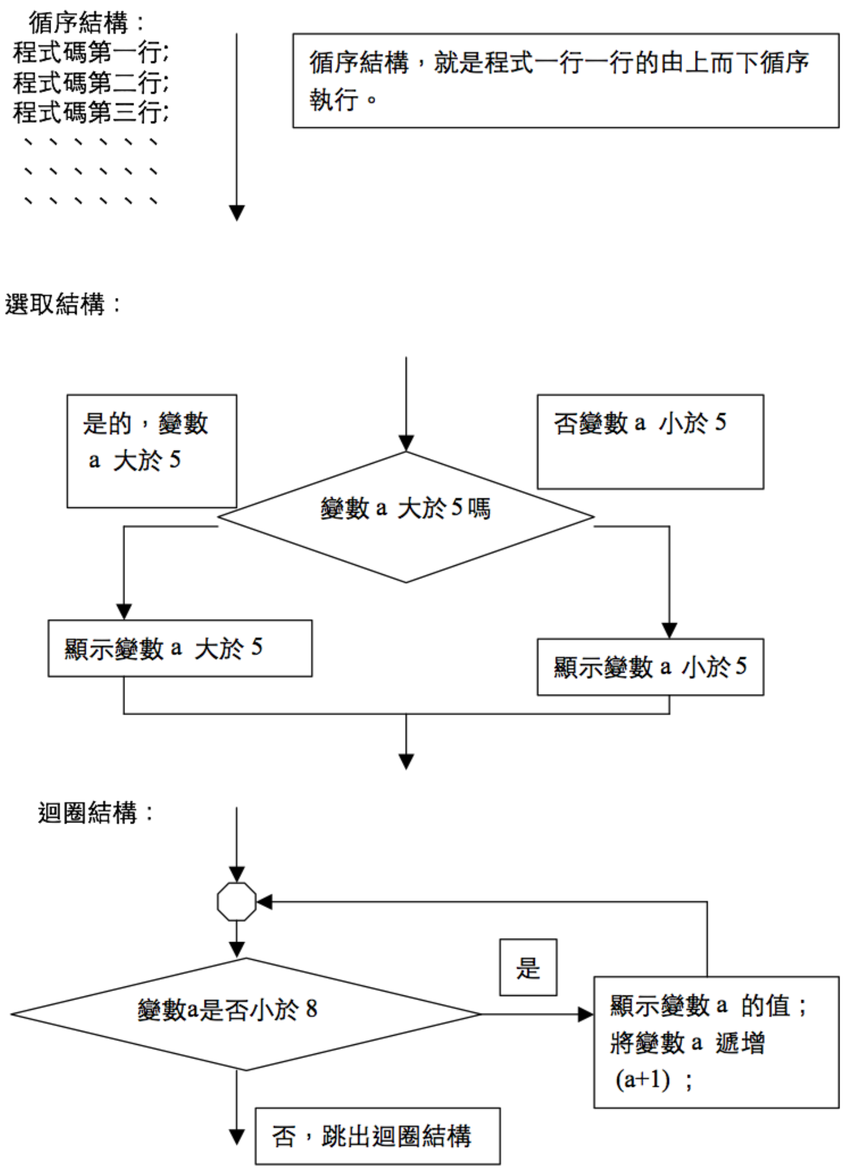
- 語法if:
- if 條件運算式:
- 　程式敘述1
- else:
- 　程式敘述2

布林運算式

- 如何來選擇流程前進的方向，我們必須經過測試條件，例如，當條件成立時往左方，當條件不成立時往右方。我們使用布林表示式來測試工作。
- 布林Boolean代數定義在一個二元素的集合上，即B={true,false}，true為真，false為假。我們可以使用這個值的結果來決定我們行進的方向。
- 當下列菱形四邊形成立true時會執行右方的流程，當下列菱形四邊行的條件不成立false時會執行左方的流程。true和false就是屬於布林代數，這是用在if判別式。



- 當下列菱形四邊形成立true時會執行右方的流程，當下列菱形四邊行的條件不成立false時會執行下方或右方的流程。True和false就是屬於布林代數，這是運用在迴圈結構。



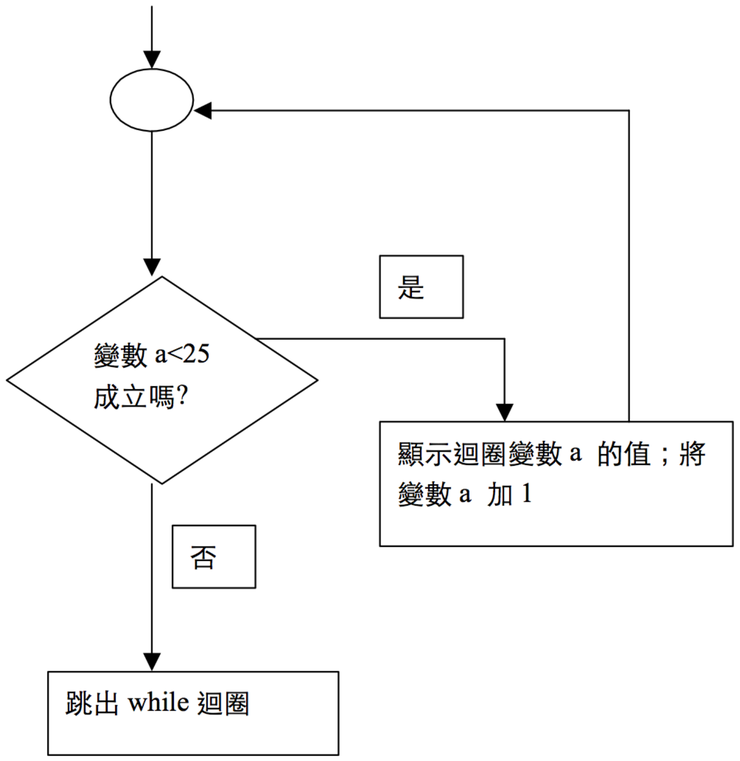
選取結構if

```
8
9  a = int(input("請輸入薪水:"))
10 if a < 50000:
11     print("薪水小於50000")
12 else:
13     print("薪水大於50000")
```

```
Console 4/A

Python 3.6.2 |Anaconda, Inc.|
Type "copyright", "credits" or

IPython 6.1.0 -- An enhanced I
薪水小於50000請輸入薪水:38000


In [2]:

In [2]: runfile('/Users/justin

請輸入薪水:85000
薪水大於50000

In [3]:
```

while迴圈

- 在if敘述中，條件後的敘述只執行一次，而在while敘述中，則可執行一次以上。While敘述的程序圖形中.選取結構和循序結構，都只執行程式敘述一次，如果我們要讓同一行程式重複執行好幾遍則要用迴圈敘述。迴圈敘述可以重複執行某一段程式好幾遍，直到條件的不成立才跳出這個迴圈。迴圈敘述：while、do……while。



迴圈結構for,while

迴圈結構for
迴圈結構while

迴圈結構for

- 語法:
- for 計數變數 in range(起始值,終始值):
  - 程式敘述

range(8,19)為8到18的數值

```
8
9 for i in range(8,19):
10     print("i的值:",i)
```

```
Console 5/A
Python 3.6.2 |An
Type "copyright"

IPython 6.1.0 --

In [1]: runfile(
i的值: 8
i的值: 9
i的值: 10
i的值: 11
i的值: 12
i的值: 13
i的值: 14
i的值: 15
i的值: 16
i的值: 17
i的值: 18
```

迴圈結構while

```
8
9 i=5
10 while i<=10:
11     print("i:",i)
12     i=i+1
```

```
Console 6/A
Python 3.6.2 |Ana
Type "copyright",

IPython 6.1.0 --

In [1]: runfile('
i: 5
i: 6
i: 7
i: 8
i: 9
i: 10

In [2]:
```

布林運算式

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Oct 26 14:30:21 2017

@author: justinwu
"""

x=True
y=False
print(x&y)
print(x|y)
if (x&y):
    print(x&y)
else:
    print(x&y)
z=True
print(z&y)
print(z|y)
```

```
In [30]: runfil
Users/justinwu/
False
True
False
False
True

In [31]:
```

continue繼續執行迴圈

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Oct 26 14:07:44 2017

@author: justinwu
"""

for i in range(8,19):
    if(i%2==1):
        continue
    print("i的值:",i)
```

```
In [19]: ru
Users/justi
i: 5
i: 6
i: 7
i: 8
i: 9

In [20]:
```

break跳出while迴圈

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Oct 26 14:09:11 2017

@author: justinwu
"""

i=5
x =True
while x:
    print("i:",i)
    i=i+1
    if(i>=10):
        break;

```

```
In [19]: runfile('
Users/justinwu/Des
i: 5
i: 6
i: 7
i: 8
i: 9

In [20]:
```

4-1邏輯運算子

- 邏輯運算子可以結合條件，以一個表達式判斷許多條件，而這些條件的結果不是真True就是假False。
- and稱為”與邏輯運算子”，只有當所有條件都成立時才會回傳真True，否則回傳假False。
- or稱為或邏輯運算子，只要運算式中一個條件成立就會回傳真True，只有當所有的條件都為假Fasle時，才會回傳假False。
- not為相反邏輯運算子，真True的條件加上not相反邏輯運算子時，就會變成假False；當假False的條件加上not相反邏輯運算子時，就會變成真True。

邏輯運算子

```
1   x=True
2   y=False
3   z=x and y
4   print(z)
5   z1=x or y
6   print(z1)
7   z2=not x
8   z3=not y
9   print(z2)
10  print(z3)
```

```
/Users/justinwu/PycharmProjects/Pyth
False
True
False
True

Process finished with exit code 0
```

4-2一個選擇的if敘述

- if 條件:

  敘述

一個選擇的if敘述

```
1   x=8
2   if x>5 :
3       print("x is bigger than 5")
```

trol_if

```
/Users/justinwu/PycharmProjects/Pytho
x is bigger than 5

Process finished with exit code 0
```

5.函數

```
i = 10
def f():
    print(i)

i = 42
f()
```

函數與參數

```
14
15 def addf(x,y):
16     print(x+y)
17
18 i1=23
19 i2=12
20 addf(23,12)
21
22
```

return回傳

```python
i = 10
def f():
    print(i)

i = 42
f()

def addf(x,y):
    print(x+y)

i1=23
i2=12
addf(23,12)

def myreturn(x,y):
    return x*y

i3=2
i5=5
i8=myreturn(i3,i5)
print(i8)

def myreturn(a,x=2,y=3):
    return a*x*y

i3=2
i5=5
i8=myreturn(8,i3,i5)
print(i8)
```

```
Console 2/A
Python 3.6.2 |Ana
Type "copyright"

IPython 6.1.0 --

In [1]: runfile(
42
35
10
80

In [2]:
```

遞迴函數

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Oct 26 16:25:09 2017

@author: justinwu
"""

def factorial(n):
    if(n==0):
        return 0
    if(n==1):
        return 1
    else:
        return n*factorial(n-1)

print(factorial(1))
print(factorial(2))
print(factorial(3))
```

```
In [2]: runfile('/Us
wdir='/Users/justinw
1
2
6

In [3]:
```

費式係數

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Oct 26 16:28:50 2017

@author: justinwu
"""

def fibonaci(n):
    if(n==0):
        return 0
    if(n==1):
        return 1
    else:
        return fibonaci(n-1)+fibonaci(n-2)

print(fibonaci(1))
print(fibonaci(2))
print(fibonaci(3))
print(fibonaci(10))
print(fibonaci(20))
```

```
In [2]: runfile
wdir='/Users/ju
1
2
6

In [3]:
```

python

6.類別

```
8  class MyClass:
9      #範例屬性參考
10     i=12345
11
12
13 print(MyClass.i)
14
15
16 class Complex:
17     #實體建構
18     def __init__(self,realpart,imagpart):
19         self.r = realpart
20         self.i = imagpart
21
22 x=Complex(3.0,-4.5)
23 print(x.r,x.i)
24
25
```

成員屬性與成員方法

```
8
9 class MyClass2:
10     #範例屬性參考
11     i=12345
12     def f(self):
13         return 'hello world'
14
15 x=MyClass2()
16 print(x.i)
17 print(x.f())
18
19
```

類別和實體變數

- __init__(self,..)為建構函數,實體化物件時會呼叫它
- self為自己這個物件

```
>>> class Dog:
        kind = 'small dog'
        def __init__(self,name):
            self.name = name


>>> d =Dog('small dog')
>>> e = Dog('very small dog')
>>> print(d.kind)
small dog
>>> print(e.kind)
small dog
>>> print(d.name)
small dog
>>> print(e.name)
very small dog
>>>
```

__init__(self)建構物件,
__del__(self)解構物件

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Oct 27 04:52:38 2017

@author: justinwu
"""

class MyClass:
    i=12345

print(MyClass.i)

class Complex:
    def __init__(self,realpart,imagepart):
        self.r=realpart
        self.i=imagepart
    def __del__(self):
        print('delete object')
x=Complex(3.0,-4.5)
print(x.r,x.i)
x=None
```

```
In [4]: runfile('/
wdir='/Users/justi
12345
3.0 -4.5
delete object

In [5]:
```

7.繼承

- class 子類別(父類別):
- 　　敘述1
- 　　敘述2

__為私有存取控制修飾,只有該類別方法才能存取

```python

class Vehicle:
    def __init__(self,name,engine):
        self.__name = name
        self.__engine = engine

    def getName(self):
        return self.__name

    def getEngine(self):
        return self.__engine

    def setEngine(self,engine):
        self.__engine = engine


class Car(Vehicle):
    def __init__(self,name,engine,electric):
        super().__init__(name,engine)
        self.__electric = electric

    def getCarName(self):
        print("名子"+self.getName())
        print("引擎"+self.getEngine())
        print("電動車"+self.__electric)

    def getAuto(self):
        print("自動駕駛車")

myCar = Car("特斯拉","磁電Engine","電力")
myCar.getCarName()
print(myCar.getAuto())
```

Console 2/A

```
Python 3.6.2 |Anacon
Type "copyright", "c

IPython 6.1.0 -- An
名子特斯拉
引擎磁電Engine
電動車電力
自動駕駛車
None

In [2]:

In [2]:
```

多重繼承

- class 子類別(父類別1,父類別2,父類別3,..):
- 　　敘述1
- 　　敘述2
- 當子類別繼承 (inheritance) 超過一個來源的時候，會以寫在最左邊的父類別優先繼承，多個父類別如果有相同名稱的屬性 (attribute) 與方法 (method) ，就會以最左邊的父類別優先。

```python
class Vehicle:
    def __init__(self,name,engine):
        self.__name = name
        self.__engine = engine

    def getName(self):
        return self.__name

    def getEngine(self):
        return self.__engine

    def setEngine(self,engine):
        self.__engine = engine

class Electric:
    def __init_(self,PowerElectric):
        self.__PowerElectric = PowerElectric

    def getPower(self):
        return self.__PowerElectric

    def setPower(self,PowerElectric):
        self.__PowerElectric = PowerElectric
```

Console

```
名子:特斯拉
引擎:磁電Engine
電動車:電力
自動駕駛車

In [13]:
```

```python
class Car(Vehicle,Electric):
    def __init__(self,name,engine,PowerElectric,auto):
        super().__init__(name,engine)
        self.setPower(PowerElectric)
        self.__Auto = auto

    def getCarName(self):
        print("名子:"+self.getName())
        print("引擎:"+self.getEngine())
        print("電動車:"+self.getPower())

    def getAuto(self):
        return self.__Auto


myCar = Car("特斯拉","磁電Engine","電力","自動駕駛車")
myCar.getCarName()
print(myCar.getAuto())
```

多型

子類別和父類別
有同名的
getEngine()名稱

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

class Vehicle:
    def __init__(self,name,engine):
        self.__name=name
        self.__engine=engine

    def getName(self):
        return self.__name
    def getEngine(self):
        return self.__engine

class Car(Vehicle):
    def __init__(self,name,engine,electric):
        super().__init__(name,engine)
        self.__electric=electric

    def getEngine(self):
        return ("超級")

    def getAuto(self):
        print("自動駕駛車")

myCar=Car("特斯拉","磁電Engine","電力")
myCar.getAuto()
print(myCar.getEngine())
```

```
In [17]: r
car_in.py'
自動駕駛車
超級

In [18]:
```

8.異常或錯誤處理

```
>>> while True:
        try:
                x = int(input("Please enter a number: "))
                break
        except ValueError:
                print("input error")
```

```
Please enter a number: f
input error
Please enter a number: f
input error
Please enter a number: ffffff
input error
Please enter a number:
```

異常或錯誤處理

```python
import sys
try:
    #f = open('myle.txt')
    f = open('mysql.py')
    s = f.readline()
    i = int(s.strip())
except OSError as err:
    print("OS error: {0}".format(err))
except ValueError:
    print("Could not convert data to an integer.")
except:
    print("Unexpected error:", sys.exc_info()[0])
```

Name	Type	Size	
s	str	1	#!/usr/bin/python

Variable explorer | File ex

IPython con

Console 1/A

```
In [5]: runfile('/Users/justinwu/Desktop/exception.py', wdi
OS error: [Errno 2] No such file or directory: 'myle.txt'

In [6]: runfile('/Users/justinwu/Desktop/exception.py', wdi
Could not convert data to an integer.

In [7]:
```

使用raise關鍵字丟出例外

```
8
9  import sys
10
11 def displaySalary(salary):
12     if salary<0:
13         raise ValueError("薪水為正")
14     print("薪水="+str(salary))
15
16 try:
17     #f = open('myle.txt')
18     Salary = eval(input("請輸入薪水:"))
19     displaySalary(Salary)
20 except OSError as err:
21     print("OS error: {0}".format(err))
22 except ValueError:
23     print("錯誤:輸入薪水值為正")
24 except:
25     print("Unexpected error:", sys.exc_info()[0])
```

Console 2/A

```
Unexpected error: <class 'SyntaxError'>

In [14]: runfile('/Users/justinwu/Deskto

請輸入薪水:80000
薪水=80000

In [15]: runfile('/Users/justinwu/Deskto

請輸入薪水:x
Unexpected error: <class 'NameError'>

In [16]: runfile('/Users/justinwu/Deskto

請輸入薪水:-10000
錯誤:

In [17]: runfile('/Users/justinwu/Deskto

請輸入薪水:-10000
錯誤:輸入薪水值為正
```

檔案處理

- fp=open(‘檔案名稱’,’檔案開啟模式’)

模式字串	當開啟檔案已存在	當開啟檔案不存在
r	開啟唯獨的檔案	產生異常錯誤
w	清除檔案內容後寫入	建立寫入檔案
a	開啟檔案從檔尾後開始寫入	建立寫入檔案
r+	開啟讀寫的檔案	產生錯誤
w+	清除檔案內容後讀寫內容	建立讀寫檔案
a+	從檔案尾巴開使讀寫	建立讀寫檔案

開啟,關閉及寫入檔案

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Oct 28 06:39:38 2017

@author: justinwu
"""
fp=open('file.txt','w')
if fp !=None:
    print('檔案開啟成功')
fp.close()

fp=open('file.txt','w')
if fp !=None:
    fp.write("小白")
fp.close()

fp=open('file.txt','w')
if fp !=None:
    fp.write("宇哲")
fp.close()
```

```
In [8]: runfile
Users/justinwu/
檔案開啟成功

In [9]:
```

讀取檔案

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Oct 28 06:46:03 2017

@author: justinwu
"""

fp=open('file.txt','r')
if fp !=None:
    str=fp.read()
    print(str)
fp.close()

fp=open('file.txt','r')
if fp !=None:
    strList=fp.readlines()
    print(strList)
fp.close()
```

```
In [11]: runfil
Users/justinwu/
宇哲
['宇哲']

In [12]:
```

8-1讀寫資料

- 檔案輸入輸出
- mode第一個字母代表操作
- mode第二個字母是檔案的類型
- t或無代表文字，b代表二進位

開啟寫入文字檔模式wt

write()寫入文字檔

```
7  poem='''呱呱呱呱呱
8  醜小鴨呀醜小鴨
9  腿兒短短腳掌大
10 長長脖子扁嘴巴
11 走起路來搖呀搖
12 愛到河邊去玩耍
13 喉嚨雖小聲音大
14 可是只會呱呱呱'''
15 print(len(poem))
16
17 fout=open('song.txt','wt')
18 fout.write(poem)
19 fout.close()
20
```

Console 1/A

```
In [21]: ru
python/exam
chaiyenwu/D
61

In [22]:
```

開啟寫入二進位檔模式wb

```
8  binaryData=bytes(range(0,128))
9  print(len(binaryData))
10 fout=open('BinaryFile','wb')
11 fout.write(binaryData)
12 fout.close()
13
14 fileIn=open('BinaryFile','rb')
15 bdata=fileIn.read()
16 len(bdata)
17 for i in bdata:
18     print(bdata[i])
```

```
Help can also be shown
Variable explorer   File explorer   Help
IPython console
Console 1/A
AAA/python/example/data/BinaryFile.py',
wdir='/Users/chaiyenwu/Desktop/AAA/python/
example/data')
128
0
1
2
```

用with來關閉檔案

- Python可清除已開啟的檔案環境管理器context managers.
- 語法:　運算式 as 變數

```
 9 binaryData=bytes(range(0,128))
10 print(len(binaryData))
11 fout=open('BinaryFile','wb')
12 fout.write(binaryData)
13 fout.close()
14
15 with open('BinaryFile','rb') as fileIn:
16     bdata=fileIn.read()
17     len(bdata)
18     for i in bdata:
19         print(bdata[i])
```

```
Console 1/A

In [21]: ru
Desktop/AAA
write.py',
Desktop/AAA
61

In [22]:
```

8-2文字檔案結構JSON

- JavaScript Object Notation(JSON)是網路的資料傳輸交換物件.
- JSON為文字檔案結構,常用在網路網頁傳輸
- 使用json.dumps(Employee)將資料結構Employee轉成JSON字串
- 使用json.loads(company_employee)來將JSON物件company_employee轉成Python資料結構字典company2
- company2.keys()顯示資料字典的鍵

文字檔案結構JSON

```python
import json

Employee={
"employee_1":{
"姓名":   "林小寶",
"年齡":   25,
"地址": "台北市復興南路199號2F",
"電話":
{
"公司":   "0922828835",
"住家":   "02-23567890"
}},
"employee_2":{
"姓名":   "江美麗",
"年齡":   35,
"地址": "台北市敦化南路19號2F",
"電話":
{
"公司":   "02-25879099",
"住家":   "02-56890989"
}}}
```

Employee	dict	2	{'empl…
company2	dict	2	{'empl…

Variable explorer | File explorer | Help

IPython console

Console 1/A

```
{
  "employee_1": {
    "姓名": "林小寶",
    "年齡": 25,
    "地址": "台北市復興南路199號2F",
    "電話": {
      "公司": "0922828835",
      "住家": "02-23567890"
    }
  },
  "employee_2": {
    "姓名": "江美麗",
    "年齡": 35,
    "地址": "台北市敦化南路19號2F",
    "電話": {
      "公司": "02-25879099",
      "住家": "02-56890989"
    }
```

ensure_ascii=False,indent=2設定dump為中文字

```
14 "地址": "台北市復興南路199號2F",
15 "電話":
16 {
17 "公司":   "0922828835",
18 "住家":   "02-23567890"
19 }},
20 "employee_2":{
21 "姓名":   "江美麗",
22 "年齡":   35,
23 "地址": "台北市敦化南路19號2F",
24 "電話":
25 {
26 "公司":   "02-25879099",
27 "住家":   "02-56890989"
28 }}}
29 #,ensure_ascii=False,indent=2設定dump為中文字
30 company_employee=json.dumps(Employee,
31                 ensure_ascii=False,indent=2)
32 print(company_employee)
33 company2=json.loads(company_employee)
34 print(company2)
35 print(company2.keys())
36
```

Employee	dict	2	{'empl…
company2	dict	2	{'empl…
company_employee	str	1	{

Variable explorer | File explorer | Help

IPython console

Console 1/A

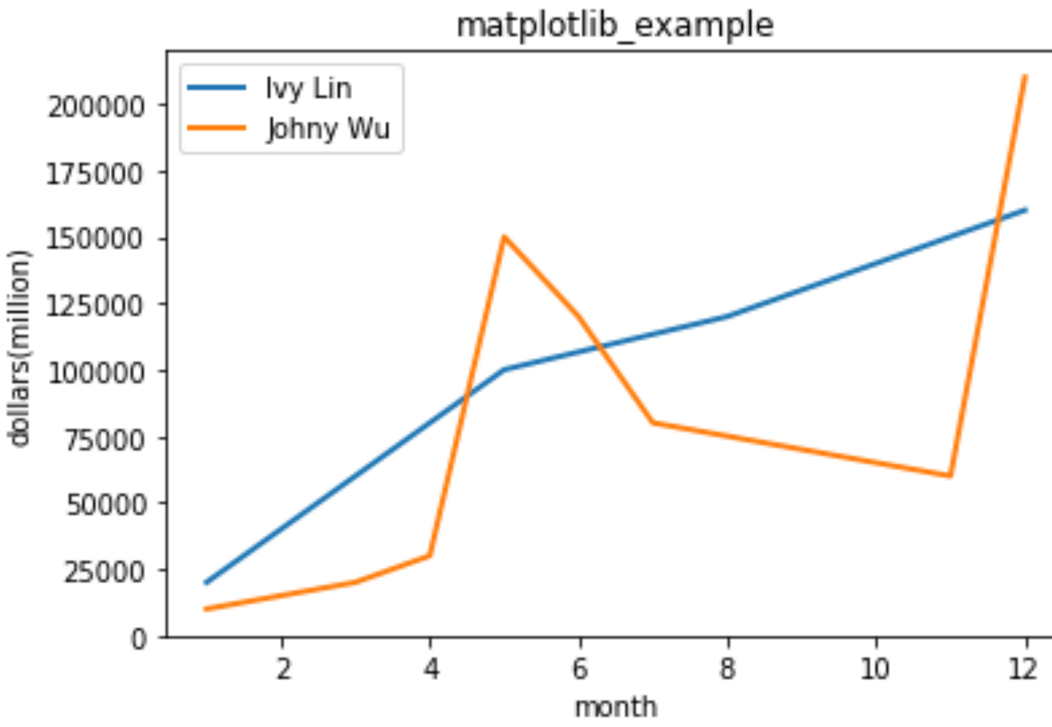
```
      "公司": "02-25879099",
      "住家": "02-56890989"
    }
  }
}
{'employee_1': {'姓名': '林小寶',
'年齡': 25, '地址': '台北市復興南路199
號2F', '電話': {'公司':
'0922828835', '住家':
'02-23567890'}}, 'employee_2':
{'姓名': '江美麗', '年齡': 35, '地
址': '台北市敦化南路19號2F', '電話':
{'公司': '02-25879099', '住家':
'02-56890989'}}}
dict_keys(['employee_1',
'employee_2'])

In [31]:
```

9.使用matplotlib畫圖

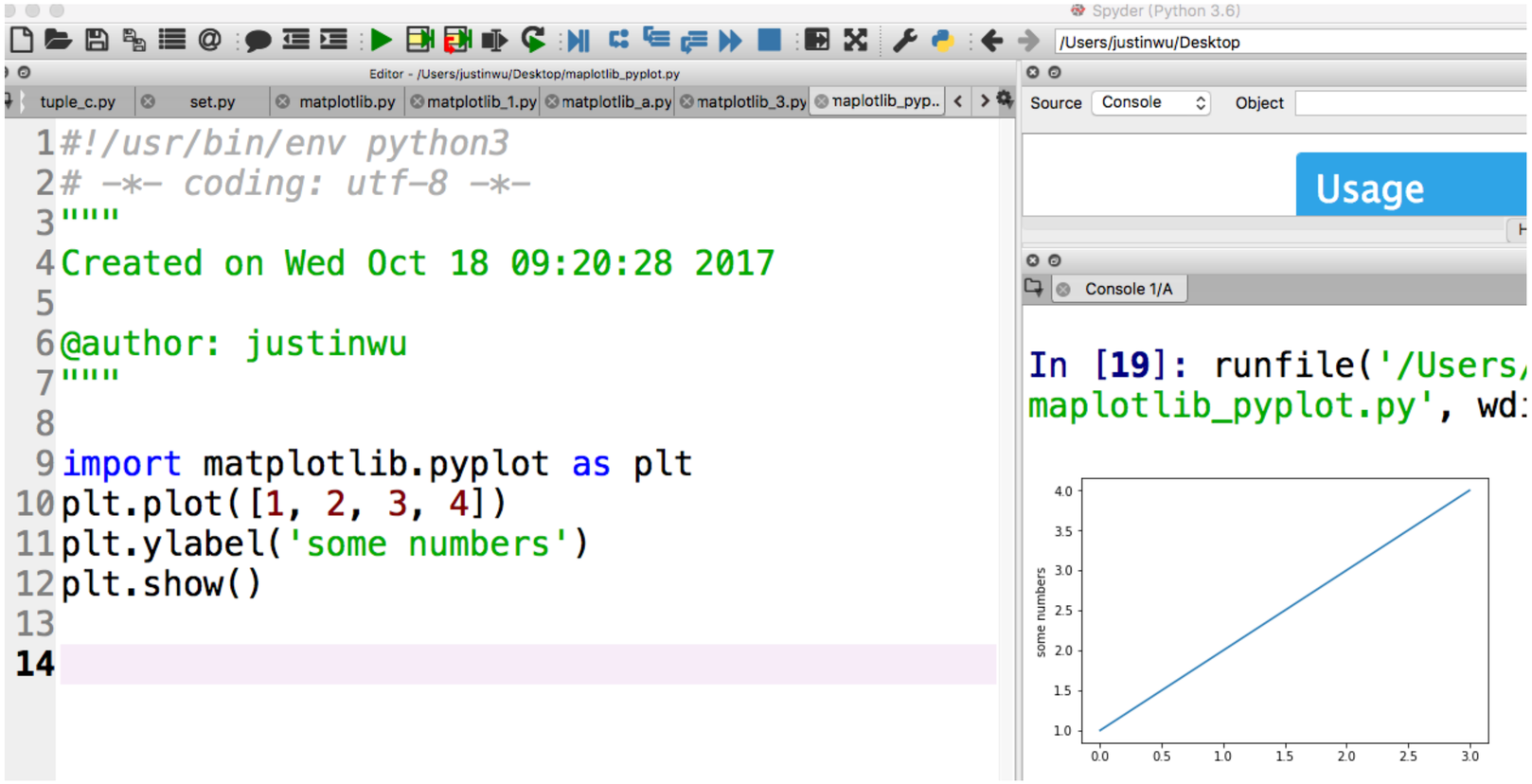
- Matplotlib.pyplot是畫圖的命令集合函數.每一個pyplot函數可以建立或修改圖形

使用matplotlib畫圖

```
6  @author: justinwu
7  """
8  import matplotlib.pyplot as mpt
9
10 month1 = [1,2,3,4,5,8,10,12]
11 month2 = [1,3,4,5,6,7,11,12]
12 sales1 = [20000,40000,60000,80000,100000,120000,140000,160000]
13 sales2 = [10000,20000,30000,150000,120000,80000,60000,210000]
14
15 mpt.plot(month1,sales1,lw=2,label='Ivy Lin')
16 mpt.plot(month2,sales2,lw=2,label='Johny Wu')
17 mpt.xlabel('month')
18 mpt.ylabel('dollars(million)')
19 mpt.legend()
20 mpt.title('matplotlib_example')
21 mpt.show()
```



plt.plot([1,2,3,4])預設是X軸,而Y軸是我們輸入的資料串列[1,2,3,4].



第一個[1,2,3,4]參數是X軸,第二個參數是Y軸

Spyder (Python 3.6)

/Users/justinwu/Desktop

Editor - /Users/justinwu/Desktop/matplotlib_a1.py

tuple.py | tuple_c.py | set.py | matplotlib.py | matplotlib_1.py | matplotlib_a1.py | matplotlib_3.py

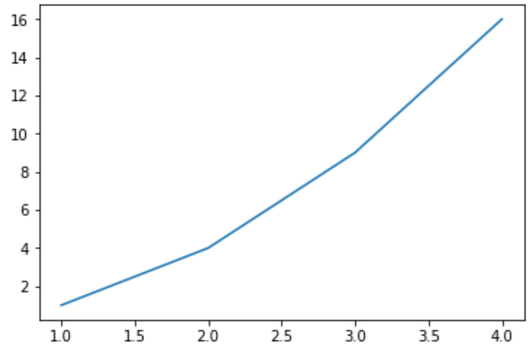
```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Oct 18 08:51:23 2017

@author: justinwu
"""

import matplotlib.pyplot as plt


plt.plot([1, 2, 3, 4], [1, 4, 9, 16])
plt.show()

```

Source | Console | Object

Usage

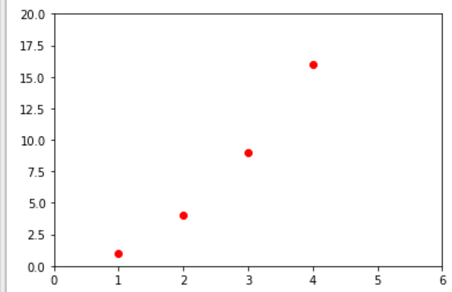
Console 1/A

```
In [24]: runfile('/Use
wdir='/Users/justinwu/
```



plot()第三個參數是格式字串點plot,'ro'為顯示紅色圓圈

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Oct 18 09:34:44 2017

@author: justinwu
"""
import matplotlib.pyplot as plt
#第三個參數是格式字串點plot,'ro'為顯示紅色圓圈
plt.plot([1, 2, 3, 4], [1, 4, 9, 16], 'ro')
plt.axis([0, 6, 0, 20])
plt.show()
```



'r--'紅色虛線,'bs'藍色矩形,'g^'綠色三角形

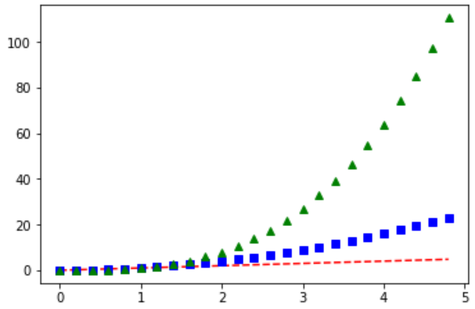
Editor - /Users/justinwu/Desktop/python/example/plot_s.py

plot_ro.py | plot_s.py

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Oct 18 09:42:54 2017

@author: justinwu
"""

import numpy as np

# 0到5每步0.2
t = np.arange(0., 5., 0.2)

# red dashes, blue squares and green triangles
# 'r--'紅色虛線,'bs'藍色矩形,'g^'綠色三角形
plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3,
plt.show()
```

Console 1/A

```
In [33]: runfile('/Users/ju
plot_s.py', wdir='/Users/ju
```



python

- Thanks.

```python
#!/usr/bin/env python3
#_*_coding:utf-8_*_
"""
Created on Thu Oct 26 07:40:42 2017

@author: justinwu
"""
#這是註解

#這是註解
1+2
#print(1+2)
x=2-1
print(x)
y=3+2
print(y)
z=3.2*2
print(z)
str='大家好'
print(str)
str2='Python'
mychar=str[0]
print(mychar)
```

Name	Type	Size	
mychar	str	1	大
str	str	1	大家好
str2	str	1	Python
x	int	1	1
y	int	1	5
z	float	1	6.4

Console 1/A

```
In [28]: runfile('/Users/ju
wdir='/Users/justinwu/Deskt
1
5
6.4
大家好
大

In [29]:
```

運算式與運算子

- 運算式是由運算子與運算元組成
- +加-減*乘/除是運算子,先乘除後加減的結合優先順序
- 運算元是變數,數字,字串和資料結構
- =是分配符號,將右邊的值分配給左邊變數

```python
#!/usr/bin/env python3
#_*_coding:utf-8_*_
"""
Created on Thu Oct 26 07:40:42 2017

@author: justinwu
"""
#這是註解

#這是註解
1+2
#print(1+2)
x=2-1
print(x)
y=3+2
print(y)
z=3.2*2
print(z)
str='大家好'
print(str)

```

Name	Type	Size	
str	str	1	大家好
x	int	1	1
y	int	1	5
z	float	1	6.4

Console 1/A

```
In [22]: runfile('/Users/j
wdir='/Users/justinwu/Desk
1
5
6.4
大家好

In [23]:
```

串列

```
>>> fruits = ['orange','apple','pear','banana','kiwi','apple','banana']
>>> fruits.count('apple')
2
>>> fruits.index('banana')
3
>>> fruits.index('banana',4)#Finding next banana starting a position 4
6
>>> fruits.reverse()
>>> fruits
['banana', 'apple', 'kiwi', 'banana', 'pear', 'apple', 'orange']
>>> fruits.append('grape')
>>> fruits
['banana', 'apple', 'kiwi', 'banana', 'pear', 'apple', 'orange', 'grape']
>>> fruits.sort()
>>> fruits
['apple', 'apple', 'banana', 'banana', 'grape', 'kiwi', 'orange', 'pear']
>>> fruits.pop()
'pear'
>>>
```

堆疊

Python 3.6.2 Shell

```
Python 3.6.2 (v3.6.2:5fd33b5926, Jul 16 2017, 20:11:06)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> WARNING: The version of Tcl/Tk (8.5.9) in use may be unstable.
Visit http://www.python.org/download/mac/tcltk/ for current information.

>>> stack=[3,4,5]
>>> stack.append(6)
>>> stack.append(7)
>>> stack
[3, 4, 5, 6, 7]
>>> stack.pop()
7
>>> stack.pop()
6
>>> stack
[3, 4, 5]
>>>
```

佇列

```python

from collections import deque
queue = deque(['阿呆','Eric','John','Michael','小寶','小文'])
queue.append("Terry")
queue.append("Graham")
print(queue.popleft())
print(queue.popleft())
print(queue)

```

```
In [1]: runfile('/Users/justinwu/Desktop/queue.py', wdir='/Us
阿呆
Eric
deque(['John', 'Michael', '小寶', '小文', 'Terry', 'Graham'])

In [2]:
```

數組tuple,集合set和字典

- 可以用數組tuple來儲存固定的元素,使用小括號()來建立一數組tuple
- 集合的元素放置沒有按照順序,可以使用{}大括號來建立一集合Set
- 集合加上索引就是字典{索引:值}

Tuple數組

- 也可以從字串中建立數組
- tp5 = tuple(‘Ivy Lin’)
- 從數組得到串列
- list1 = list(tp5)

```python

tp1=()
print(tp1)
tp2=(1,2,3,4,5,6,7,8)
print(tp2)
print(sum(tp2))
print('--------------------')
print(tp2[2:5])#切割運算子
print(tp2[-1])
tp3 =tuple([2*x for x in range(1,8)])
print(tp3)
print('----------------')
tp4=tuple('Ivy Lin')
print(tp4)
tp5=("John",'小寶','小文')
print(tp5)
print(len(tp5))
print(tp4+tp5)
print('--------------')
tp6=tuple([1,2,3,4,5,6,7,8,9])
print(tp6)
print(max(tp6))
print(min(tp6))

```

```
In [6]: runfile('/Users/justinwu/Desktop/tuple.py', wdir='
Desktop')
()
(1, 2, 3, 4, 5, 6, 7, 8)
36
--------------------
(3, 4, 5)
8
(2, 4, 6, 8, 10, 12, 14)
----------------
('I', 'v', 'y', ' ', 'L', 'i', 'n')
('John', '小寶', '小文')
3
('I', 'v', 'y', ' ', 'L', 'i', 'n', 'John', '小寶', '小文')
--------------
(1, 2, 3, 4, 5, 6, 7, 8, 9)
9
1

In [7]:
```

```python

tp6=tuple([66,22,3,46,5,65,7,83,19])
print(tp6)
list1 = list(tp6)
list1.sort()#排序串列
print(list1)
print('-----------')
tp8=tuple(list1)
tp9=tuple(list1)
print(tp8)
print(tp8 == tp9)#比較兩個數組tuple


```

```
In [15]: runfile('/Users/justinwu/
Desktop')
(66, 22, 3, 46, 5, 65, 7, 83, 19)
[3, 5, 7, 19, 22, 46, 65, 66, 83]
-----------
(3, 5, 7, 19, 22, 46, 65, 66, 83)
True

In [16]:
```

Set集合

- 集合(set)用來儲存沒有重複的元素.
- 集合的元素是不可以複製的,元素放置也沒有按照順序
- 可以使用{}大括號來建立一集合Set

Set集合

```python

st1=set()#建立一個空集合
st2=set([1,2,3,4,5])
print(st2)
st3={'a','b','c','d','e'}
print(st3)
print('-------------------')
st3.add('f')
print(st3)
st3.remove('d')
print(st3)
print('----------')
print(st3.union(st2))
st5={'a','b','c','d','e'}
print(st3.intersection(st5))
print(st3.difference(st5))
```

```
In [30]: runfile('/Users/justinwu/Deskto
Desktop')
{1, 2, 3, 4, 5}
{'d', 'a', 'e', 'c', 'b'}
-------------------
{'d', 'f', 'a', 'e', 'c', 'b'}
{'f', 'a', 'e', 'c', 'b'}
----------
{1, 2, 3, 4, 5, 'f', 'a', 'e', 'c', 'b'}
{'b', 'a', 'c', 'e'}
{'f'}

In [31]:
```

字典

- 集合加上索引就是字典{索引:值}

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Oct 27 19:40:39 2017

@author: justinwu
"""
tel={'Justin':'0920909872','Ivy':'0922876895'}
tel['Johny']='0920885356'
print(tel)
print(tel['Johny'])
del tel['Johny']
tel['Mary']='0922865255'
print(tel)
print(list(tel.keys()))
print(sorted(tel.keys()))
print('Ivy' in tel)
print('Johny' in tel)
```

```
In [5]: runfile('/Users/justinwu/Desktop/TOP/python/
example/dictionary_1.py', wdir='/Users/justinwu/Desktop/
TOP/python/example')
{'Justin': '0920909872', 'Ivy': '0922876895', 'Johny':
'0920885356'}
0920885356
{'Justin': '0920909872', 'Ivy': '0922876895', 'Mary':
'0922865255'}
['Justin', 'Ivy', 'Mary']
['Ivy', 'Justin', 'Mary']
True
False

In [6]:
```

4.控制結構

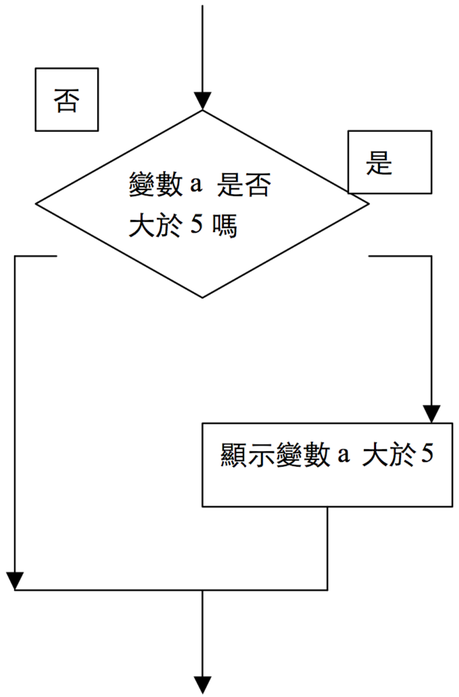
- 選取結構if
- 迴圈結構while,for

選取結構if

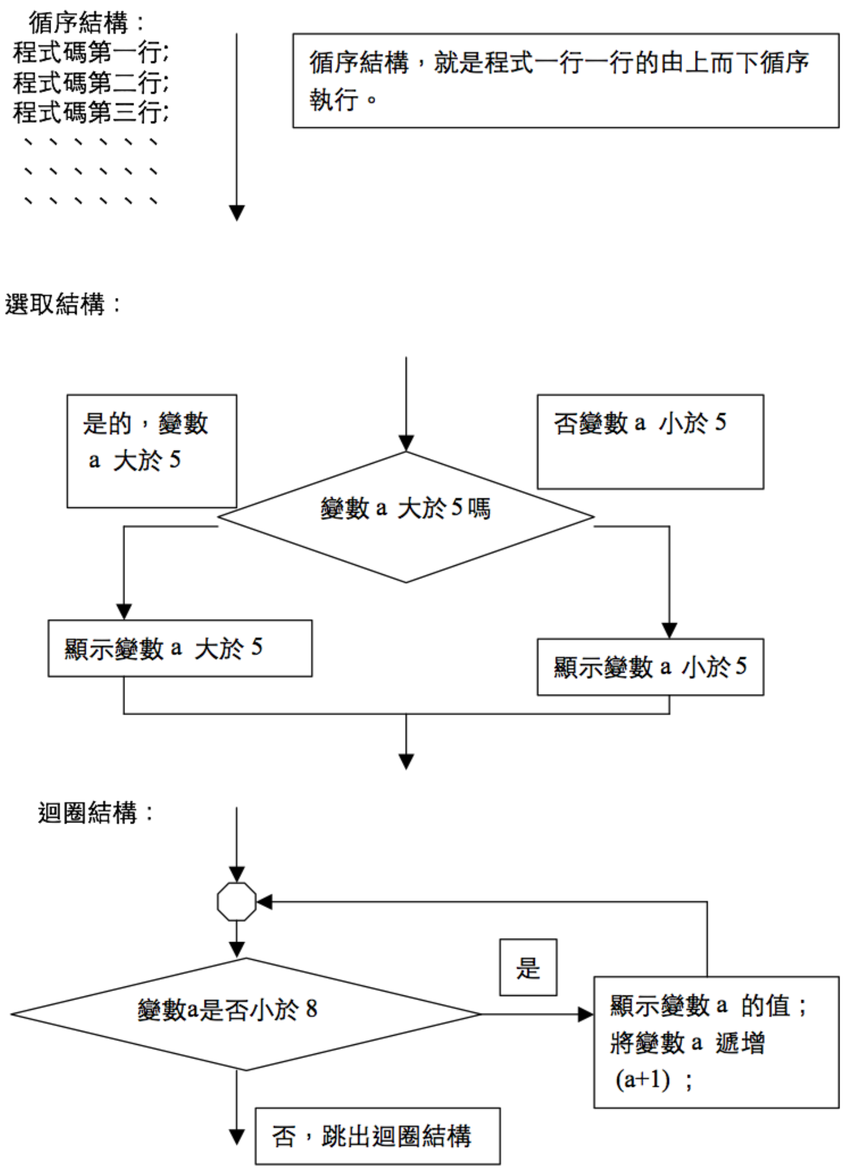
- 語法if:
- if 條件運算式:
- 　程式敘述1
- else:
- 　程式敘述2

布林運算式

- 如何來選擇流程前進的方向，我們必須經過測試條件，例如，當條件成立時往左方，當條件不成立時往右方。我們使用布林表示式來測試工作。
- 布林Boolean代數定義在一個二元素的集合上，即B={true,false}，true為真，false為假。我們可以使用這個值的結果來決定我們行進的方向。
- 當下列菱形四邊形成立true時會執行右方的流程，當下列菱形四邊行的條件不成立false時會執行左方的流程。true和false就是屬於布林代數，這是用在if判別式。



- 當下列菱形四邊形成立true時會執行右方的流程，當下列菱形四邊行的條件不成立false時會執行下方或右方的流程。True和false就是屬於布林代數，這是運用在迴圈結構。



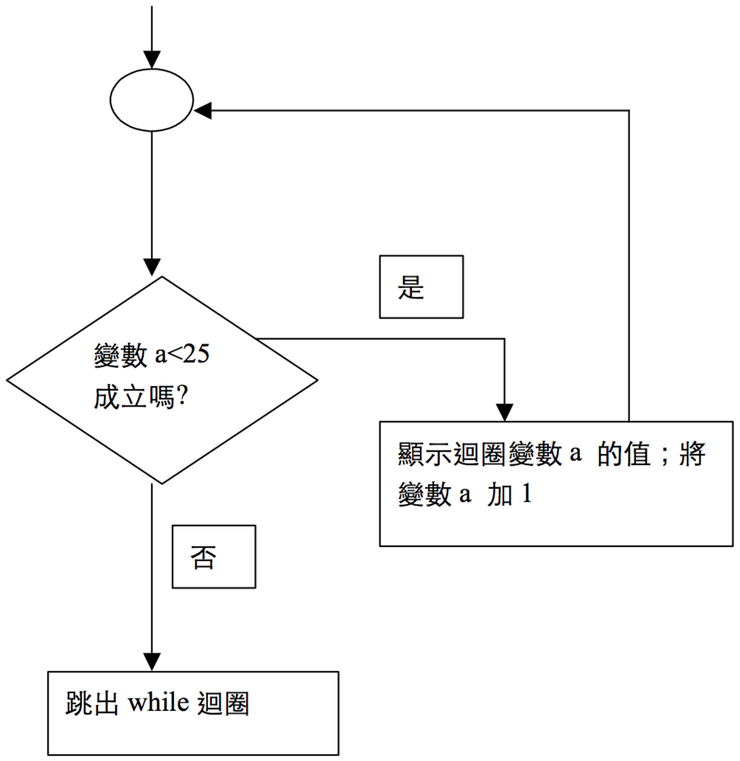
選取結構if

```
8
9  a = int(input("請輸入薪水:"))
10 if a < 50000:
11     print("薪水小於50000")
12 else:
13     print("薪水大於50000")
```

```
Console 4/A

Python 3.6.2 |Anaconda, Inc.|
Type "copyright", "credits" or

IPython 6.1.0 -- An enhanced I
薪水小於50000請輸入薪水:38000


In [2]:

In [2]: runfile('/Users/justin

請輸入薪水:85000
薪水大於50000

In [3]:
```

while迴圈

- 在if敘述中，條件後的敘述只執行一次，而在while敘述中，則可執行一次以上。While敘述的程序圖形中.選取結構和循序結構，都只執行程式敘述一次，如果我們要讓同一行程式重複執行好幾遍則要用迴圈敘述。迴圈敘述可以重複執行某一段程式好幾遍，直到條件的不成立才跳出這個迴圈。迴圈敘述：while、do……while。



迴圈結構for,while

迴圈結構for
迴圈結構while

迴圈結構for

- 語法:
- for 計數變數 in range(起始值,終始值):
  - 程式敘述

range(8,19)為8到18的數值

```
8
9 for i in range(8,19):
10     print("i的值:",i)
```

```
Console 5/A
Python 3.6.2 |An
Type "copyright"

IPython 6.1.0 --

In [1]: runfile(
i的值: 8
i的值: 9
i的值: 10
i的值: 11
i的值: 12
i的值: 13
i的值: 14
i的值: 15
i的值: 16
i的值: 17
i的值: 18
```

迴圈結構while

```
8
9 i=5
10 while i<=10:
11     print("i:",i)
12     i=i+1
```

```
Console 6/A
Python 3.6.2 |Ana
Type "copyright",

IPython 6.1.0 --

In [1]: runfile('
i: 5
i: 6
i: 7
i: 8
i: 9
i: 10

In [2]:
```

布林運算式

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Oct 26 14:30:21 2017

@author: justinwu
"""

x=True
y=False
print(x&y)
print(x|y)
if (x&y):
    print(x&y)
else:
    print(x&y)
z=True
print(z&y)
print(z|y)
```

```
In [30]: runfil
Users/justinwu/
False
True
False
False
True

In [31]:
```

continue繼續執行迴圈

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Oct 26 14:07:44 2017

@author: justinwu
"""

for i in range(8,19):
    if(i%2==1):
        continue
    print("i的值:",i)
```

```
In [19]: ru
Users/justi
i: 5
i: 6
i: 7
i: 8
i: 9

In [20]:
```

break跳出while迴圈

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Oct 26 14:09:11 2017

@author: justinwu
"""

i=5
x =True
while x:
    print("i:",i)
    i=i+1
    if(i>=10):
        break;

```

```
In [19]: runfile('
Users/justinwu/Des
i: 5
i: 6
i: 7
i: 8
i: 9

In [20]:
```

4-1邏輯運算子

- 邏輯運算子可以結合條件，以一個表達式判斷許多條件，而這些條件的結果不是真True就是假False。
- and稱為”與邏輯運算子”，只有當所有條件都成立時才會回傳真True，否則回傳假False。
- or稱為或邏輯運算子，只要運算式中一個條件成立就會回傳真True，只有當所有的條件都為假Fasle時，才會回傳假False。
- not為相反邏輯運算子，真True的條件加上not相反邏輯運算子時，就會變成假False；當假False的條件加上not相反邏輯運算子時，就會變成真True。

邏輯運算子

```
1   x=True
2   y=False
3   z=x and y
4   print(z)
5   z1=x or y
6   print(z1)
7   z2=not x
8   z3=not y
9   print(z2)
10  print(z3)
```

```
/Users/justinwu/PycharmProjects/Pyth
False
True
False
True

Process finished with exit code 0
```

4-2一個選擇的if敘述

- if 條件:

  敘述

一個選擇的if敘述

```
1   x=8
2   if x>5 :
3       print("x is bigger than 5")
```

trol_if

```
/Users/justinwu/PycharmProjects/Pytho
x is bigger than 5

Process finished with exit code 0
```

5.函數

```
i = 10
def f():
    print(i)

i = 42
f()
```

函數與參數

```
14
15 def addf(x,y):
16     print(x+y)
17
18 i1=23
19 i2=12
20 addf(23,12)
21
22
```

return回傳

```python
i = 10
def f():
    print(i)

i = 42
f()

def addf(x,y):
    print(x+y)

i1=23
i2=12
addf(23,12)

def myreturn(x,y):
    return x*y

i3=2
i5=5
i8=myreturn(i3,i5)
print(i8)

def myreturn(a,x=2,y=3):
    return a*x*y

i3=2
i5=5
i8=myreturn(8,i3,i5)
print(i8)
```

```
Console 2/A
Python 3.6.2 |Ana
Type "copyright"

IPython 6.1.0 --

In [1]: runfile(
42
35
10
80

In [2]:
```

遞迴函數

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Oct 26 16:25:09 2017

@author: justinwu
"""

def factorial(n):
    if(n==0):
        return 0
    if(n==1):
        return 1
    else:
        return n*factorial(n-1)

print(factorial(1))
print(factorial(2))
print(factorial(3))
```

```
In [2]: runfile('/Us
wdir='/Users/justinw
1
2
6

In [3]:
```

費式係數

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Oct 26 16:28:50 2017

@author: justinwu
"""

def fibonaci(n):
    if(n==0):
        return 0
    if(n==1):
        return 1
    else:
        return fibonaci(n-1)+fibonaci(n-2)

print(fibonaci(1))
print(fibonaci(2))
print(fibonaci(3))
print(fibonaci(10))
print(fibonaci(20))
```

```
In [2]: runfile
wdir='/Users/ju
1
2
6

In [3]:
```

python

6.類別

```
8  class MyClass:
9      #範例屬性參考
10     i=12345
11
12
13 print(MyClass.i)
14
15
16 class Complex:
17     #實體建構
18     def __init__(self,realpart,imagpart):
19         self.r = realpart
20         self.i = imagpart
21
22 x=Complex(3.0,-4.5)
23 print(x.r,x.i)
24
25
```

成員屬性與成員方法

```
8
9 class MyClass2:
10     #範例屬性參考
11     i=12345
12     def f(self):
13         return 'hello world'
14
15 x=MyClass2()
16 print(x.i)
17 print(x.f())
18
19
```

類別和實體變數

- __init__(self,..)為建構函數,實體化物件時會呼叫它
- self為自己這個物件

```
>>> class Dog:
        kind = 'small dog'
        def __init__(self,name):
            self.name = name


>>> d =Dog('small dog')
>>> e = Dog('very small dog')
>>> print(d.kind)
small dog
>>> print(e.kind)
small dog
>>> print(d.name)
small dog
>>> print(e.name)
very small dog
>>>
```

__init__(self)建構物件,
__del__(self)解構物件

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Oct 27 04:52:38 2017

@author: justinwu
"""

class MyClass:
    i=12345

print(MyClass.i)

class Complex:
    def __init__(self,realpart,imagepart):
        self.r=realpart
        self.i=imagepart
    def __del__(self):
        print('delete object')
x=Complex(3.0,-4.5)
print(x.r,x.i)
x=None
```

```
In [4]: runfile('/
wdir='/Users/justi
12345
3.0 -4.5
delete object

In [5]:
```

7.繼承

- class 子類別(父類別):
- 　　敘述1
- 　　敘述2

__為私有存取控制修飾,只有該類別方法才能存取

```python

class Vehicle:
    def __init__(self,name,engine):
        self.__name = name
        self.__engine = engine

    def getName(self):
        return self.__name

    def getEngine(self):
        return self.__engine

    def setEngine(self,engine):
        self.__engine = engine


class Car(Vehicle):
    def __init__(self,name,engine,electric):
        super().__init__(name,engine)
        self.__electric = electric

    def getCarName(self):
        print("名子"+self.getName())
        print("引擎"+self.getEngine())
        print("電動車"+self.__electric)

    def getAuto(self):
        print("自動駕駛車")

myCar = Car("特斯拉","磁電Engine","電力")
myCar.getCarName()
print(myCar.getAuto())
```

Console 2/A

```
Python 3.6.2 |Anacon
Type "copyright", "c

IPython 6.1.0 -- An
名子特斯拉
引擎磁電Engine
電動車電力
自動駕駛車
None

In [2]:

In [2]:
```

多重繼承

- class 子類別(父類別1,父類別2,父類別3,..):
- 　　敘述1
- 　　敘述2
- 當子類別繼承 (inheritance) 超過一個來源的時候，會以寫在最左邊的父類別優先繼承，多個父類別如果有相同名稱的屬性 (attribute) 與方法 (method) ，就會以最左邊的父類別優先。

```python
class Vehicle:
    def __init__(self,name,engine):
        self.__name = name
        self.__engine = engine

    def getName(self):
        return self.__name

    def getEngine(self):
        return self.__engine

    def setEngine(self,engine):
        self.__engine = engine

class Electric:
    def __init_(self,PowerElectric):
        self.__PowerElectric = PowerElectric

    def getPower(self):
        return self.__PowerElectric

    def setPower(self,PowerElectric):
        self.__PowerElectric = PowerElectric
```

Console

```
名子:特斯拉
引擎:磁電Engine
電動車:電力
自動駕駛車

In [13]:
```

```python
class Car(Vehicle,Electric):
    def __init__(self,name,engine,PowerElectric,auto):
        super().__init__(name,engine)
        self.setPower(PowerElectric)
        self.__Auto = auto

    def getCarName(self):
        print("名子:"+self.getName())
        print("引擎:"+self.getEngine())
        print("電動車:"+self.getPower())

    def getAuto(self):
        return self.__Auto


myCar = Car("特斯拉","磁電Engine","電力","自動駕駛車")
myCar.getCarName()
print(myCar.getAuto())
```

多型

子類別和父類別
有同名的
getEngine()名稱

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

class Vehicle:
    def __init__(self,name,engine):
        self.__name=name
        self.__engine=engine

    def getName(self):
        return self.__name
    def getEngine(self):
        return self.__engine

class Car(Vehicle):
    def __init__(self,name,engine,electric):
        super().__init__(name,engine)
        self.__electric=electric

    def getEngine(self):
        return ("超級")

    def getAuto(self):
        print("自動駕駛車")

myCar=Car("特斯拉","磁電Engine","電力")
myCar.getAuto()
print(myCar.getEngine())
```

```
In [17]: r
car_in.py'
自動駕駛車
超級

In [18]:
```

8.異常或錯誤處理

```
>>> while True:
        try:
                x = int(input("Please enter a number: "))
                break
        except ValueError:
                print("input error")
```

```
Please enter a number: f
input error
Please enter a number: f
input error
Please enter a number: ffffff
input error
Please enter a number:
```

異常或錯誤處理

```python
import sys
try:
    #f = open('myle.txt')
    f = open('mysql.py')
    s = f.readline()
    i = int(s.strip())
except OSError as err:
    print("OS error: {0}".format(err))
except ValueError:
    print("Could not convert data to an integer.")
except:
    print("Unexpected error:", sys.exc_info()[0])
```

Name	Type	Size	
s	str	1	#!/usr/bin/python

Variable explorer | File ex

IPython con

Console 1/A

```
In [5]: runfile('/Users/justinwu/Desktop/exception.py', wdi
OS error: [Errno 2] No such file or directory: 'myle.txt'

In [6]: runfile('/Users/justinwu/Desktop/exception.py', wdi
Could not convert data to an integer.

In [7]:
```

使用raise關鍵字丟出例外

```
8
9  import sys
10
11 def displaySalary(salary):
12     if salary<0:
13         raise ValueError("薪水為正")
14     print("薪水="+str(salary))
15
16 try:
17     #f = open('myle.txt')
18     Salary = eval(input("請輸入薪水:"))
19     displaySalary(Salary)
20 except OSError as err:
21     print("OS error: {0}".format(err))
22 except ValueError:
23     print("錯誤:輸入薪水值為正")
24 except:
25     print("Unexpected error:", sys.exc_info()[0])
```

Console 2/A

```
Unexpected error: <class 'SyntaxError'>

In [14]: runfile('/Users/justinwu/Deskto

請輸入薪水:80000
薪水=80000

In [15]: runfile('/Users/justinwu/Deskto

請輸入薪水:x
Unexpected error: <class 'NameError'>

In [16]: runfile('/Users/justinwu/Deskto

請輸入薪水:-10000
錯誤:

In [17]: runfile('/Users/justinwu/Deskto

請輸入薪水:-10000
錯誤:輸入薪水值為正
```

檔案處理

- fp=open(‘檔案名稱’,’檔案開啟模式’)

模式字串	當開啟檔案已存在	當開啟檔案不存在
r	開啟唯獨的檔案	產生異常錯誤
w	清除檔案內容後寫入	建立寫入檔案
a	開啟檔案從檔尾後開始寫入	建立寫入檔案
r+	開啟讀寫的檔案	產生錯誤
w+	清除檔案內容後讀寫內容	建立讀寫檔案
a+	從檔案尾巴開使讀寫	建立讀寫檔案

開啟,關閉及寫入檔案

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Oct 28 06:39:38 2017

@author: justinwu
"""
fp=open('file.txt','w')
if fp !=None:
    print('檔案開啟成功')
fp.close()

fp=open('file.txt','w')
if fp !=None:
    fp.write("小白")
fp.close()

fp=open('file.txt','w')
if fp !=None:
    fp.write("宇哲")
fp.close()
```

```
In [8]: runfile
Users/justinwu/
檔案開啟成功

In [9]:
```

讀取檔案

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Oct 28 06:46:03 2017

@author: justinwu
"""

fp=open('file.txt','r')
if fp !=None:
    str=fp.read()
    print(str)
fp.close()

fp=open('file.txt','r')
if fp !=None:
    strList=fp.readlines()
    print(strList)
fp.close()
```

```
In [11]: runfil
Users/justinwu/
宇哲
['宇哲']

In [12]:
```

8-1讀寫資料

- 檔案輸入輸出
- mode第一個字母代表操作
- mode第二個字母是檔案的類型
- t或無代表文字，b代表二進位

開啟寫入文字檔模式wt

write()寫入文字檔

```
7  poem='''呱呱呱呱呱
8  醜小鴨呀醜小鴨
9  腿兒短短腳掌大
10 長長脖子扁嘴巴
11 走起路來搖呀搖
12 愛到河邊去玩耍
13 喉嚨雖小聲音大
14 可是只會呱呱呱'''
15 print(len(poem))
16
17 fout=open('song.txt','wt')
18 fout.write(poem)
19 fout.close()
20
```

Console 1/A

```
In [21]: ru
python/exam
chaiyenwu/D
61

In [22]:
```

開啟寫入二進位檔模式wb

```
8  binaryData=bytes(range(0,128))
9  print(len(binaryData))
10 fout=open('BinaryFile','wb')
11 fout.write(binaryData)
12 fout.close()
13
14 fileIn=open('BinaryFile','rb')
15 bdata=fileIn.read()
16 len(bdata)
17 for i in bdata:
18     print(bdata[i])
```

```
Help can also be shown
Variable explorer   File explorer   Help
IPython console
Console 1/A
AAA/python/example/data/BinaryFile.py',
wdir='/Users/chaiyenwu/Desktop/AAA/python/
example/data')
128
0
1
2
```

用with來關閉檔案

- Python可清除已開啟的檔案環境管理器context managers.
- 語法:　運算式 as 變數

```
 9 binaryData=bytes(range(0,128))
10 print(len(binaryData))
11 fout=open('BinaryFile','wb')
12 fout.write(binaryData)
13 fout.close()
14
15 with open('BinaryFile','rb') as fileIn:
16     bdata=fileIn.read()
17     len(bdata)
18     for i in bdata:
19         print(bdata[i])
```

```
Console 1/A

In [21]: ru
Desktop/AAA
write.py',
Desktop/AAA
61

In [22]:
```

8-2文字檔案結構JSON

- JavaScript Object Notation(JSON)是網路的資料傳輸交換物件.
- JSON為文字檔案結構,常用在網路網頁傳輸
- 使用json.dumps(Employee)將資料結構Employee轉成JSON字串
- 使用json.loads(company_employee)來將JSON物件company_employee轉成Python資料結構字典company2
- company2.keys()顯示資料字典的鍵

文字檔案結構JSON

```python
import json

Employee={
"employee_1":{
"姓名":   "林小寶",
"年齡":   25,
"地址": "台北市復興南路199號2F",
"電話":
{
"公司":   "0922828835",
"住家":   "02-23567890"
}},
"employee_2":{
"姓名":   "江美麗",
"年齡":   35,
"地址": "台北市敦化南路19號2F",
"電話":
{
"公司":   "02-25879099",
"住家":   "02-56890989"
}}}
```

Employee	dict	2	{'empl…
company2	dict	2	{'empl…

Variable explorer | File explorer | Help

IPython console

Console 1/A

```
{
  "employee_1": {
    "姓名": "林小寶",
    "年齡": 25,
    "地址": "台北市復興南路199號2F",
    "電話": {
      "公司": "0922828835",
      "住家": "02-23567890"
    }
  },
  "employee_2": {
    "姓名": "江美麗",
    "年齡": 35,
    "地址": "台北市敦化南路19號2F",
    "電話": {
      "公司": "02-25879099",
      "住家": "02-56890989"
    }
```

ensure_ascii=False,indent=2設定dump為中文字

```
14 "地址": "台北市復興南路199號2F",
15 "電話":
16 {
17 "公司":   "0922828835",
18 "住家":   "02-23567890"
19 }},
20 "employee_2":{
21 "姓名":   "江美麗",
22 "年齡":   35,
23 "地址": "台北市敦化南路19號2F",
24 "電話":
25 {
26 "公司":   "02-25879099",
27 "住家":   "02-56890989"
28 }}}
29 #,ensure_ascii=False,indent=2設定dump為中文字
30 company_employee=json.dumps(Employee,
31                 ensure_ascii=False,indent=2)
32 print(company_employee)
33 company2=json.loads(company_employee)
34 print(company2)
35 print(company2.keys())
36
```

Employee	dict	2	{'empl…
company2	dict	2	{'empl…
company_employee	str	1	{

Variable explorer | File explorer | Help

IPython console

Console 1/A

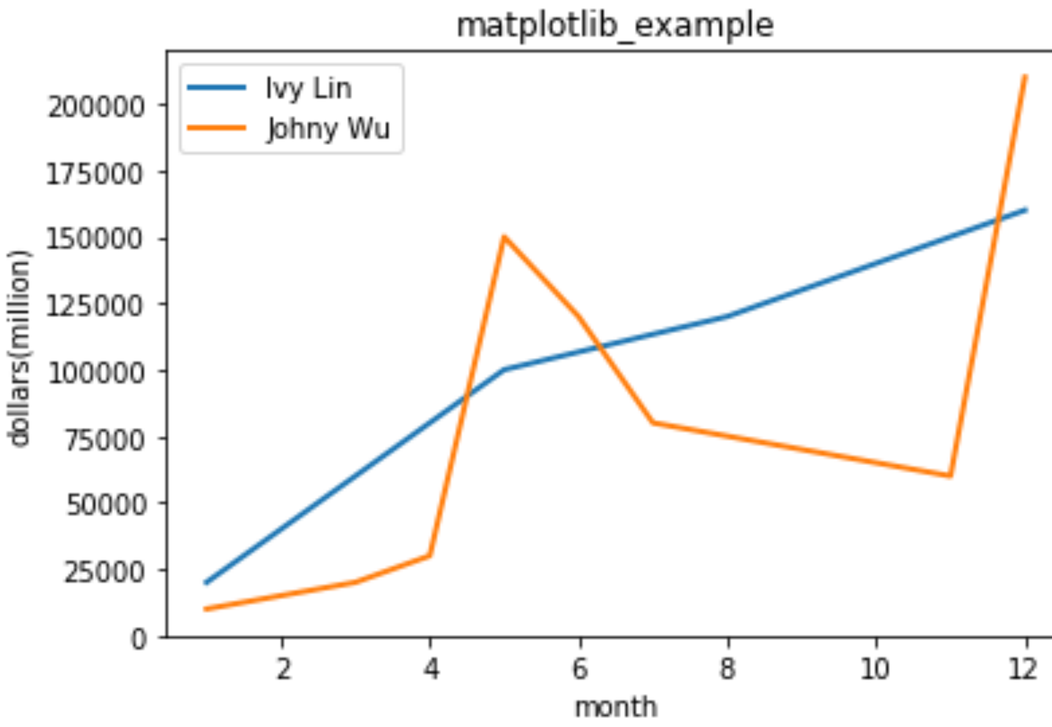
```
      "公司": "02-25879099",
      "住家": "02-56890989"
    }
  }
}
{'employee_1': {'姓名': '林小寶',
'年齡': 25, '地址': '台北市復興南路199
號2F', '電話': {'公司':
'0922828835', '住家':
'02-23567890'}}, 'employee_2':
{'姓名': '江美麗', '年齡': 35, '地
址': '台北市敦化南路19號2F', '電話':
{'公司': '02-25879099', '住家':
'02-56890989'}}}
dict_keys(['employee_1',
'employee_2'])

In [31]:
```

9.使用matplotlib畫圖

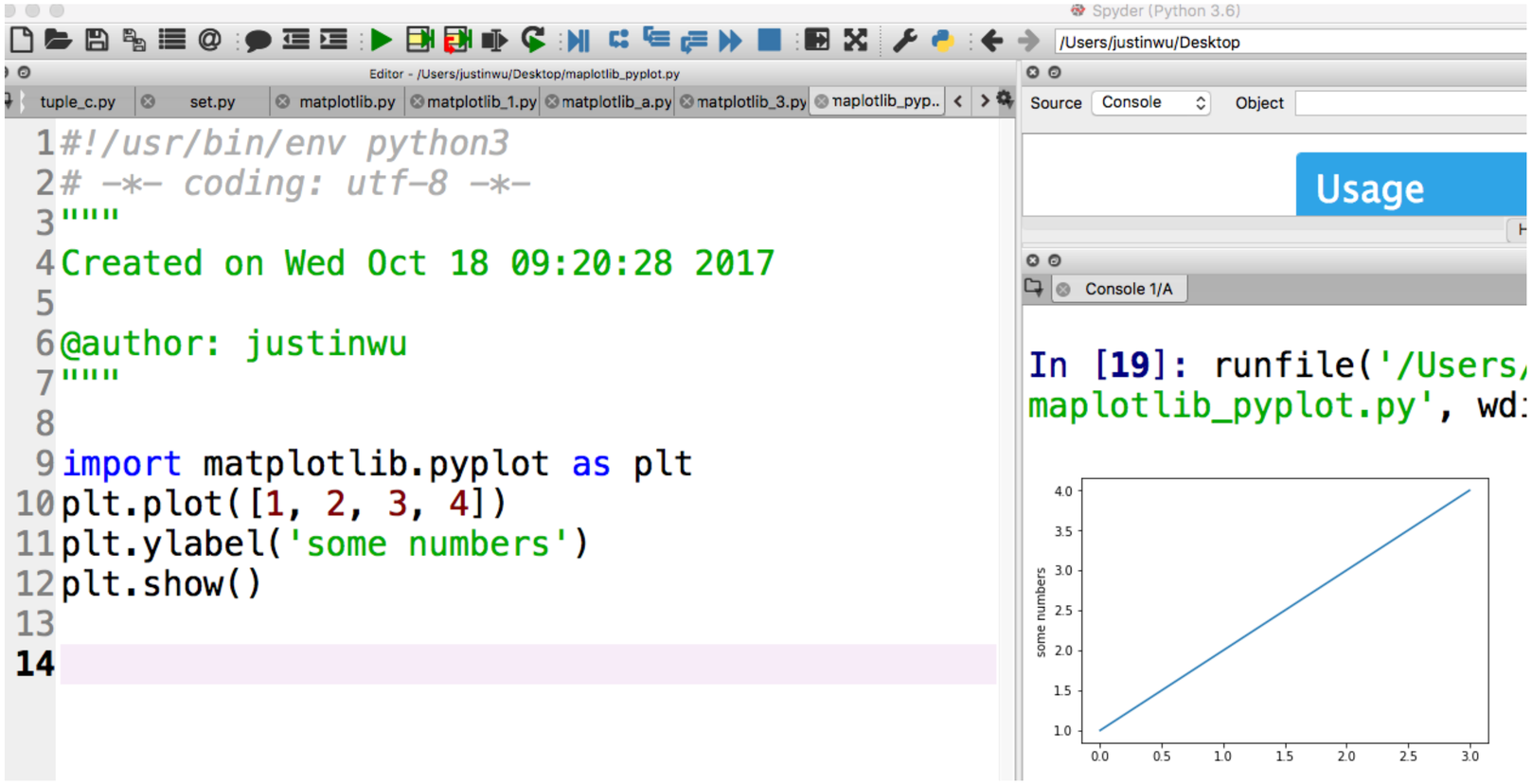
- Matplotlib.pyplot是畫圖的命令集合函數.每一個pyplot函數可以建立或修改圖形

使用matplotlib畫圖

```
6  @author: justinwu
7  """
8  import matplotlib.pyplot as mpt
9
10 month1 = [1,2,3,4,5,8,10,12]
11 month2 = [1,3,4,5,6,7,11,12]
12 sales1 = [20000,40000,60000,80000,100000,120000,140000,160000]
13 sales2 = [10000,20000,30000,150000,120000,80000,60000,210000]
14
15 mpt.plot(month1,sales1,lw=2,label='Ivy Lin')
16 mpt.plot(month2,sales2,lw=2,label='Johny Wu')
17 mpt.xlabel('month')
18 mpt.ylabel('dollars(million)')
19 mpt.legend()
20 mpt.title('matplotlib_example')
21 mpt.show()
```



plt.plot([1,2,3,4])預設是X軸,而Y軸是我們輸入的資料串列[1,2,3,4].



第一個[1,2,3,4]參數是X軸,第二個參數是Y軸

Spyder (Python 3.6)

/Users/justinwu/Desktop

Editor - /Users/justinwu/Desktop/matplotlib_a1.py

tuple.py | tuple_c.py | set.py | matplotlib.py | matplotlib_1.py | matplotlib_a1.py | matplotlib_3.py

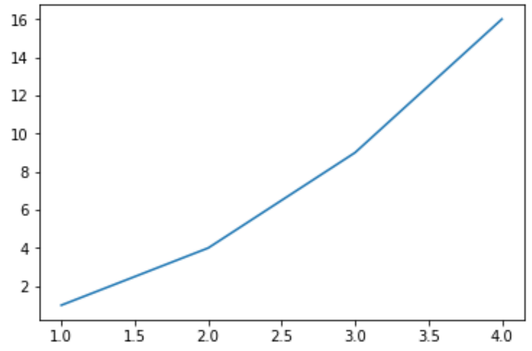
```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Oct 18 08:51:23 2017

@author: justinwu
"""

import matplotlib.pyplot as plt


plt.plot([1, 2, 3, 4], [1, 4, 9, 16])
plt.show()

```

Source | Console | Object

Usage

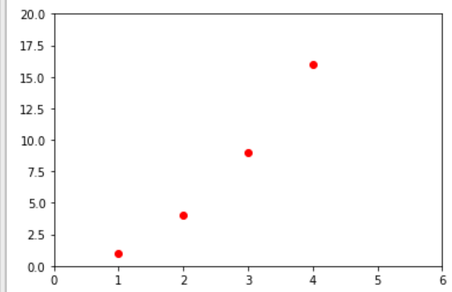
Console 1/A

```
In [24]: runfile('/Use
wdir='/Users/justinwu/
```



plot()第三個參數是格式字串點plot,'ro'為顯示紅色圓圈

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Oct 18 09:34:44 2017

@author: justinwu
"""
import matplotlib.pyplot as plt
#第三個參數是格式字串點plot,'ro'為顯示紅色圓圈
plt.plot([1, 2, 3, 4], [1, 4, 9, 16], 'ro')
plt.axis([0, 6, 0, 20])
plt.show()
```



'r--'紅色虛線,'bs'藍色矩形,'g^'綠色三角形

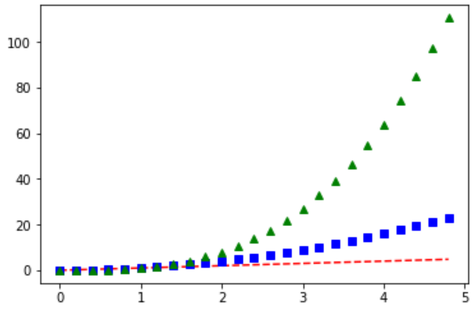
Editor - /Users/justinwu/Desktop/python/example/plot_s.py

plot_ro.py | plot_s.py

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Oct 18 09:42:54 2017

@author: justinwu
"""

import numpy as np

# 0到5每步0.2
t = np.arange(0., 5., 0.2)

# red dashes, blue squares and green triangles
# 'r--'紅色虛線,'bs'藍色矩形,'g^'綠色三角形
plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3,
plt.show()
```

Console 1/A

```
In [33]: runfile('/Users/ju
plot_s.py', wdir='/Users/ju
```



python

- Thanks.

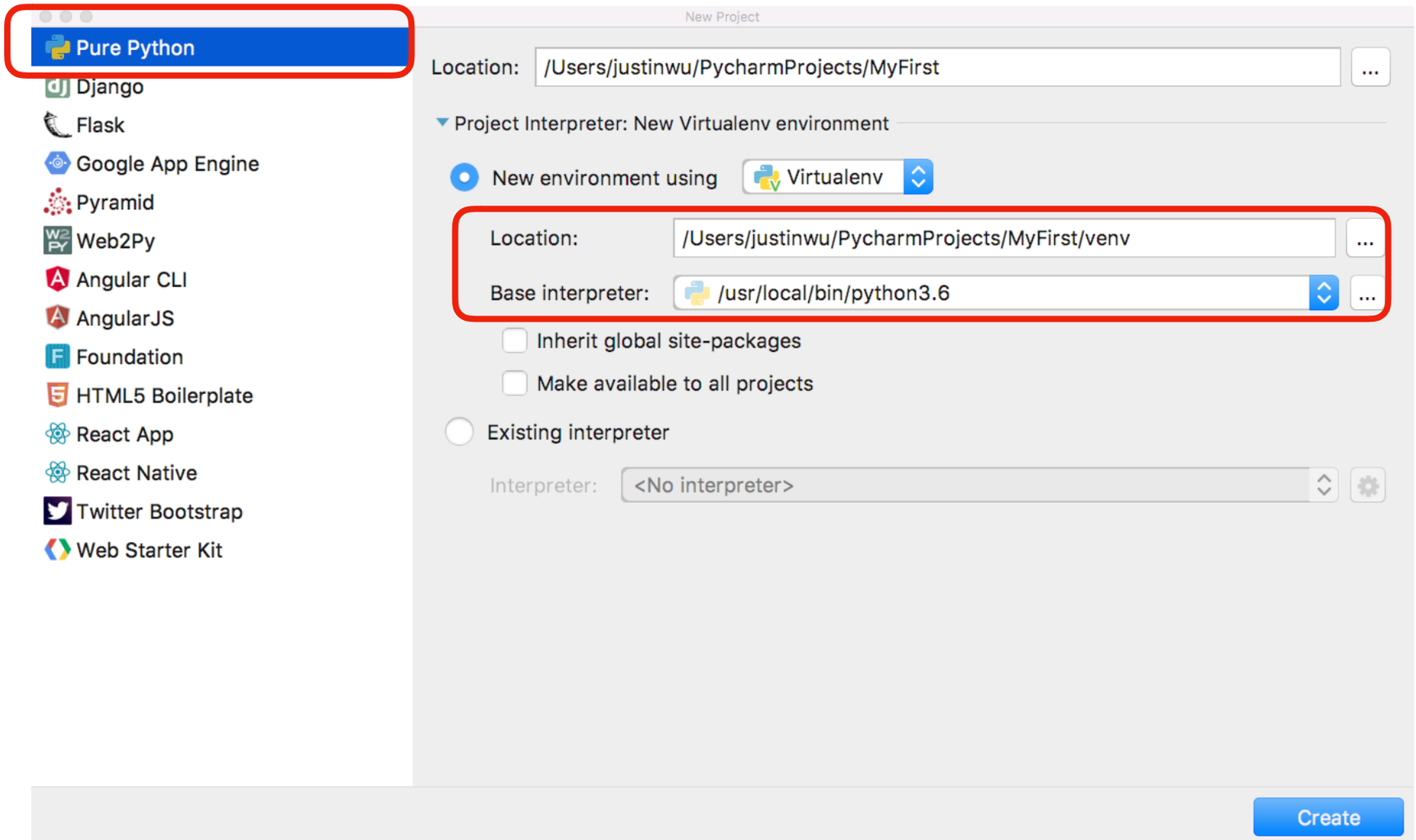
Python一天學會

吳佳諺 老師

1-1下載及安裝PyCharm

- https://www.jetbrains.com/pycharm/

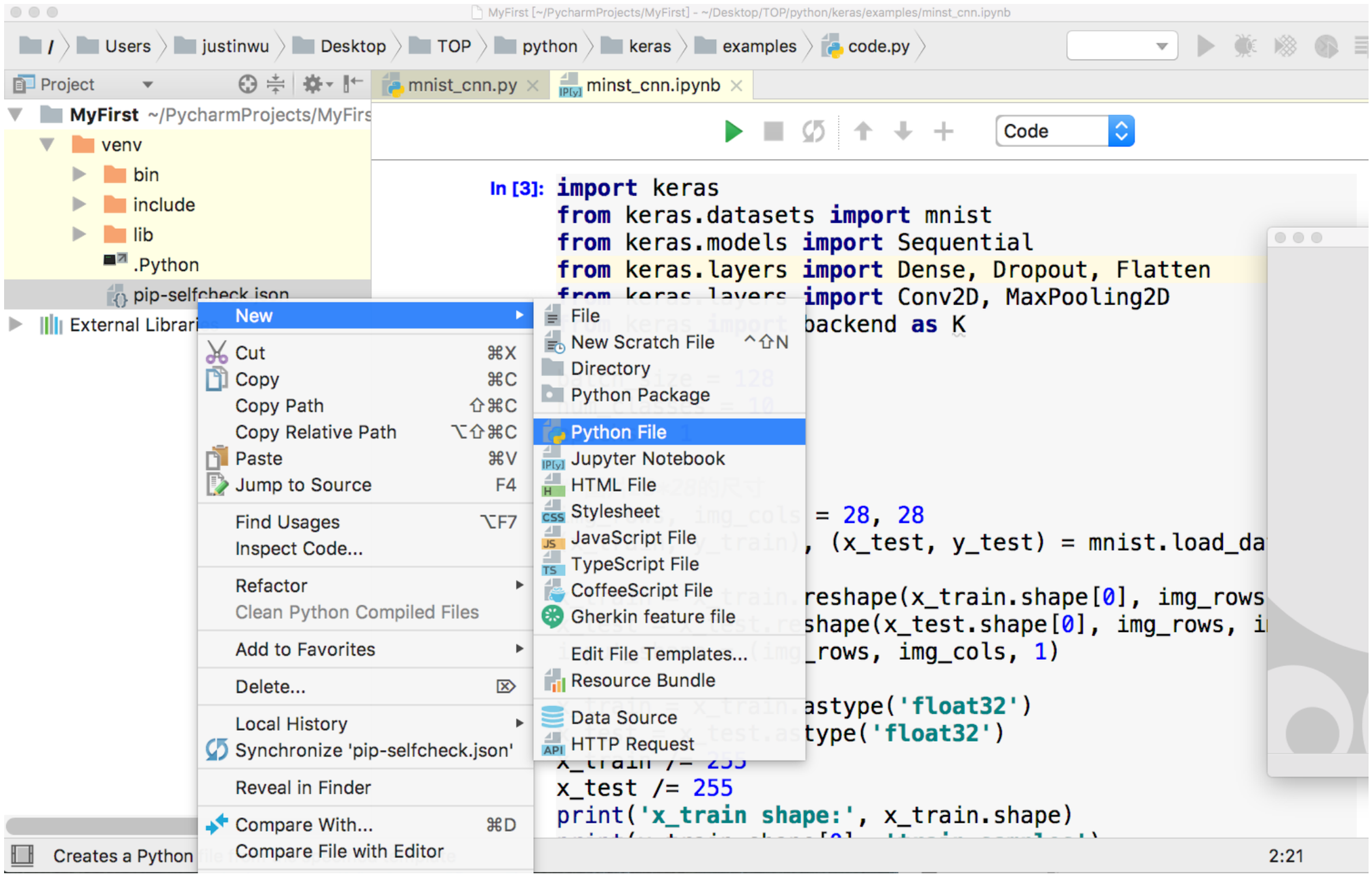
安裝完Python後,設定直譯器路徑



Pycharm可以使用Python和Jupyter notebook的檔案

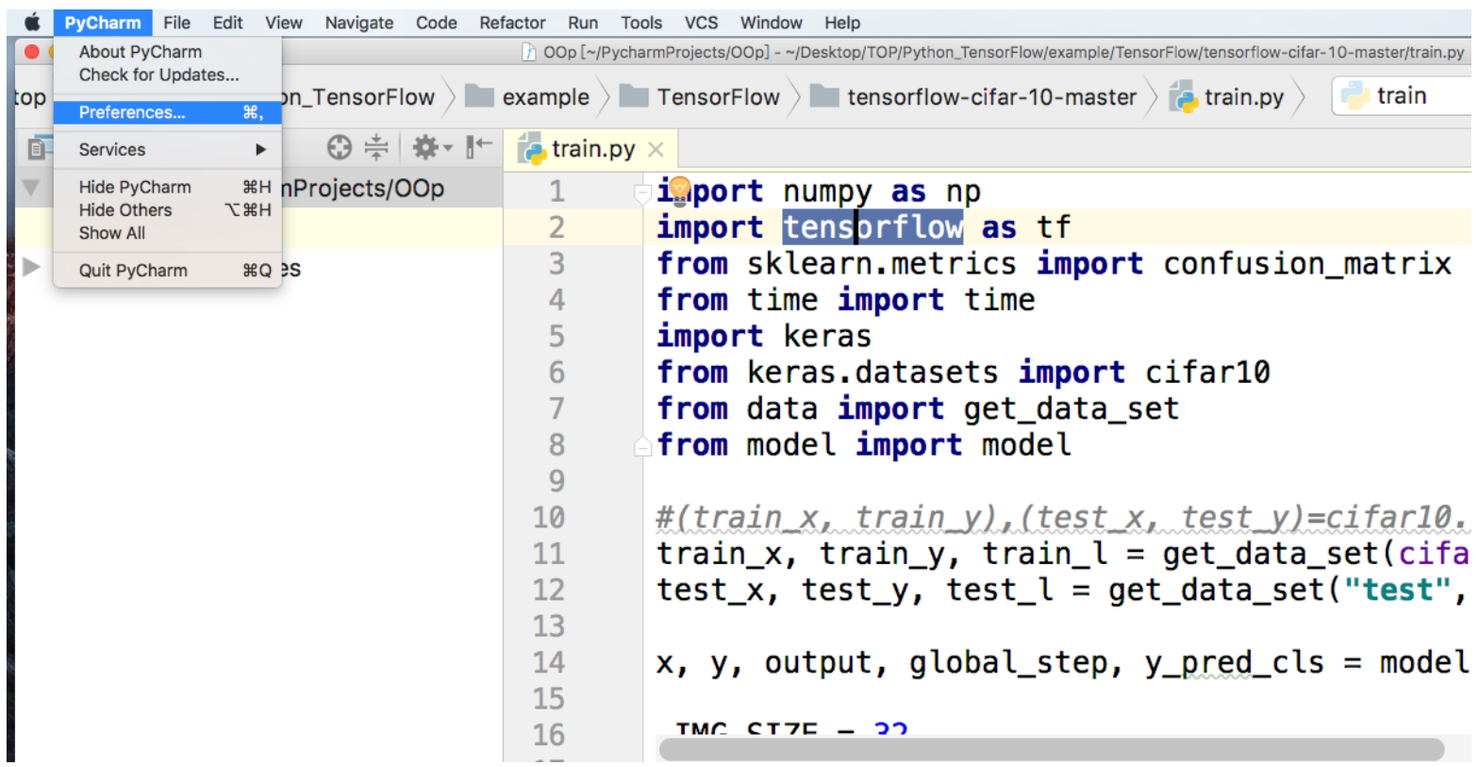
- 我們可以新增Python的檔案

- 選取File->NEW->Python File

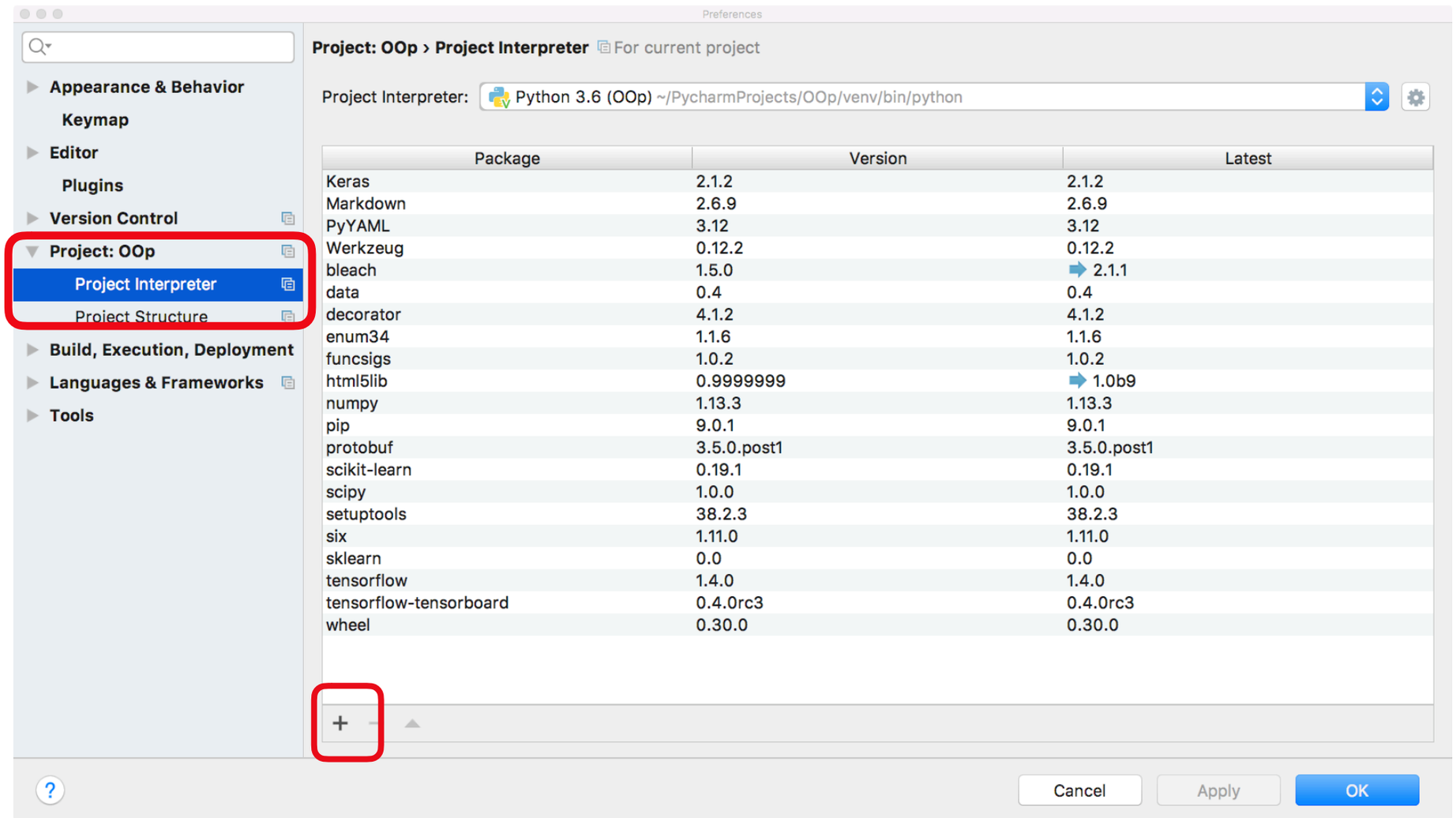

MyFirst [~/PycharmProjects/MyFirst] - ~/Desktop/TOP/python/keras/examples/minst_cnn.ipynb
/ Users justinwu Desktop TOP python keras examples code.py
Project
mnist_cnn.py
minst_cnn.ipynb
MyFirst ~/PycharmProjects/MyFirs
venv
bin
include
lib
.Python
pip-selfcheck.json
External Librari
Code
In [3]: import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D, MaxPooling2D
backend as K
= 28, 28
(x_test, y_test) = mnist.load_da
reshape(x_train.shape[0], img_rows
shape(x_test.shape[0], img_rows, i
rows, img_cols, 1)
astype('float32')
type('float32')
x_test /= 255
print('x_train shape:', x_train.shape)
New
Cut ⌘X
Copy ⌘C
Copy Path ⇧⌘C
Copy Relative Path ⌥⇧⌘C
Paste ⌘V
Jump to Source F4
Find Usages ⌥F7
Inspect Code...
Refactor
Clean Python Compiled Files
Add to Favorites
Delete...
Local History
Synchronize 'pip-selfcheck.json'
Reveal in Finder
Compare With... ⌘D
Compare File with Editor
File
New Scratch File ^⇧N
Directory
Python Package
Python File
Jupyter Notebook
HTML File
Stylesheet
JavaScript File
TypeScript File
CoffeeScript File
Gherkin feature file
Edit File Templates...
Resource Bundle
Data Source
HTTP Request
Creates a Python
2:21

1-2在Pycharm新增函式庫

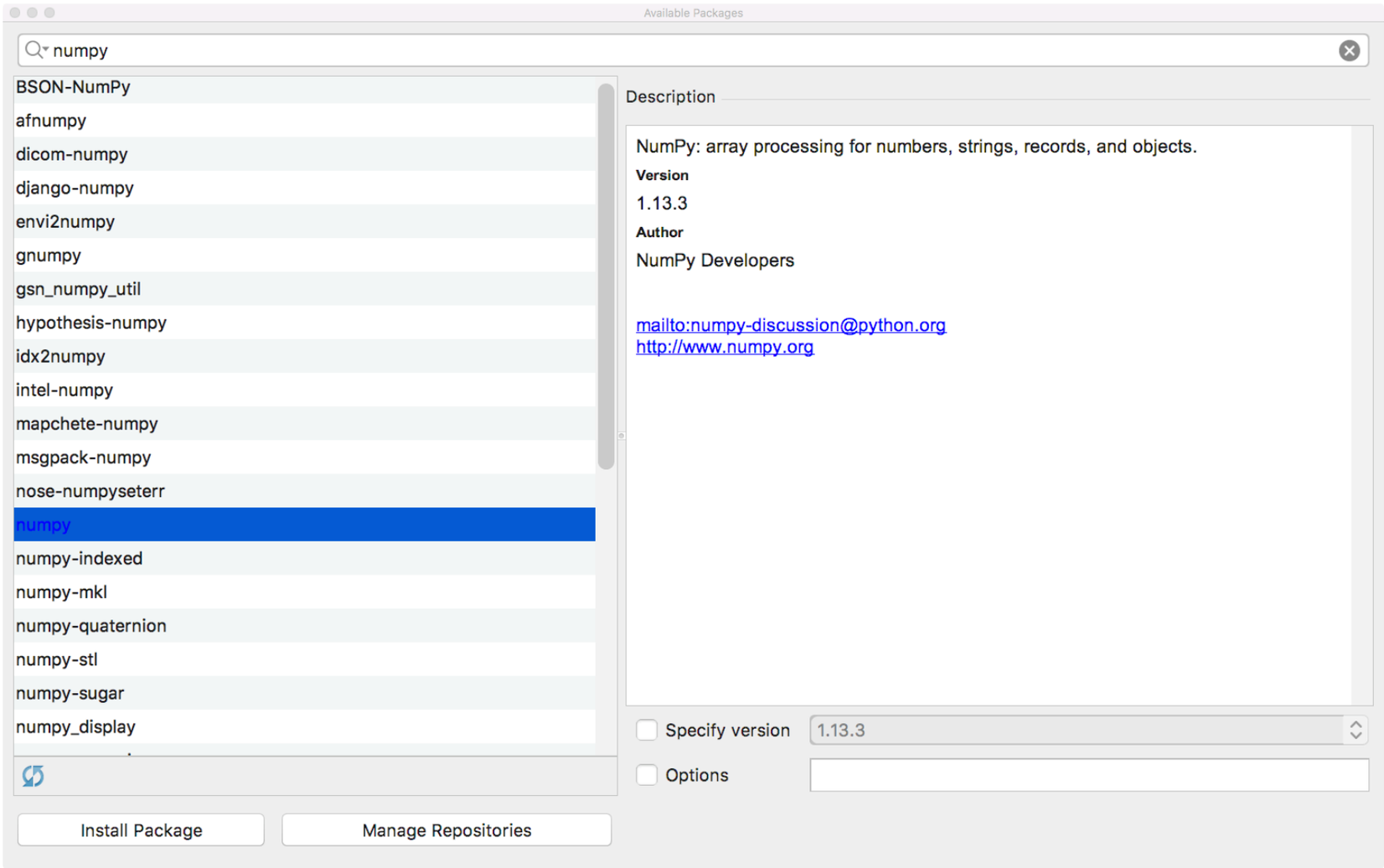
- 選取PyCharm->Preference



選取直譯器，選取＋


Preferences
Appearance & Behavior
Keymap
Editor
Plugins
Version Control
Project: OOp
Project Interpreter
Project Structure
Build, Execution, Deployment
Languages & Frameworks
Tools
Project: OOp › Project Interpreter For current project
Project Interpreter: Python 3.6 (OOp) ~/PycharmProjects/OOp/venv/bin/python
Package Version Latest
Keras 2.1.2 2.1.2
Markdown 2.6.9 2.6.9
PyYAML 3.12 3.12
Werkzeug 0.12.2 0.12.2
bleach 1.5.0 2.1.1
data 0.4 0.4
decorator 4.1.2 4.1.2
enum34 1.1.6 1.1.6
funcsigs 1.0.2 1.0.2
html5lib 0.9999999 1.0b9
numpy 1.13.3 1.13.3
pip 9.0.1 9.0.1
protobuf 3.5.0.post1 3.5.0.post1
scikit-learn 0.19.1 0.19.1
scipy 1.0.0 1.0.0
setuptools 38.2.3 38.2.3
six 1.11.0 1.11.0
sklearn 0.0 0.0
tensorflow 1.4.0 1.4.0
tensorflow-tensorboard 0.4.0rc3 0.4.0rc3
wheel 0.30.0 0.30.0
Cancel
Apply
OK

選取函式庫，再install Package



Python 精通程式語言

Python直譯器與計算機

吳佳諺 老師

Python直譯器與計算機

- 2-1資料型態
- 2-2識別名稱
- 2-3運算式與運算子
- 2-4運算子結合優先順序
- 2-5數字，字串與變數

2.Python直譯器與計算機

- Mac電腦/usr/local/bin
- Windows電腦C:\python36
- set path=%path%;C:\python36

輸入python執行

```
$ python
Python 3.6.2 |Anaconda, Inc.| (default,
Sep 21 2017, 18:29:43)
[GCC 4.2.1 Compatible Clang 4.0.1 (tags/
RELEASE_401/final)] on darwin
Type "help", "copyright", "credits" or
"license" for more information.
>>>
```

UTF-8編碼

-*- coding: encoding -*-

這是設定中文utf-8-*-編碼

#-*- coding: utf-8 -*-

註解

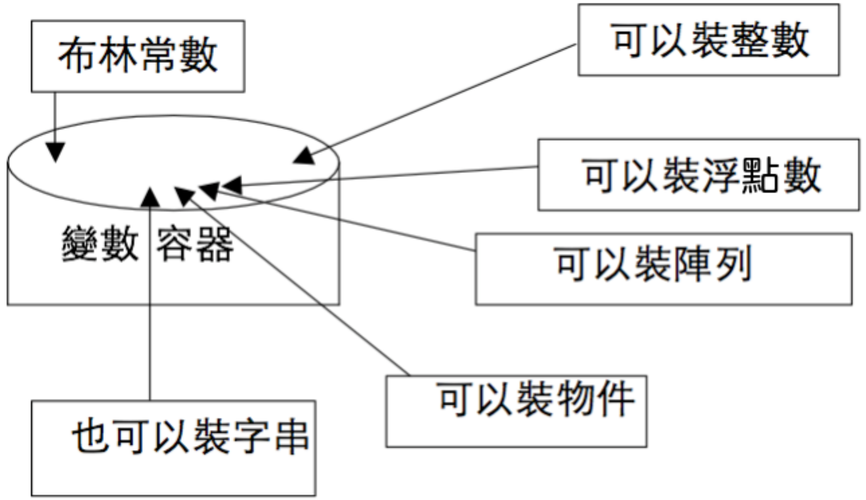
- #是註解符號

python計算機

```
>>> 1+2
3
>>> 2-1
1
>>> 3*2
6
>>> 3/2
1.5
>>> 3%2
1
>>> 50-5/6
49.166666666666664
>>> (50-5*6)/4
5.0
>>> 5**2
25
>>> 2**5
32
>>>
```

2-1資料型態

- Python支援的資料型態:
- 浮點數(floating point)
- 字元(char)
- 整數(integer)
- 物件(object)
- 布林常數(Boolean)
- 空值(null)
- 字串(string)。



- 範例：Circle.py
- 第一行宣告了pi的資料型態為雙精度浮點數，並且給予初始值3.14159。
- 第二行宣告了radius半徑的資料型態為雙精度浮點數。
- 第三行宣告了area面積的資料型態為雙精度浮點數。

Circle.py

```
pi=3.14159
radius=10.0
area=radius*radius*pi
print(area)
```

```
/Users/justinwu/PycharmProjects/Pyt
314.159

Process finished with exit code 0
```

2-1-1整數

- 十進位
- >>> 88888
- 88888
- >>> 99999
- 99999
- >>> 1000
- 1000
- >>> 10
- 10
- >>> 15
- 15

二進位

- 以0b為開頭的代表二進位
- >>> 0b0001
- 1
- >>> 0b0010
- 2
- >>> 0b1000
- 8
- >>> 0b0011
- 3
- >>> 0b0101
- 5

八進位

- 以0o為開頭的代表八進位
- >>> 0o001
- 1
- >>> 0o010
- 8
- >>> 0o011
- 9
- >>> 0o100
- 64
- >>> 0o101
- 65
- >>> 0o110
- 72

十六進位

- 以0x為開頭的代表16進位
- >>> 0x0011
- 17
- >>> 0x0010
- 16
- >>> 0x0001
- 1
- >>> 0x000A
- 10
- >>> 0x000D
- 13

整數位元運算子

x|y	**整數x和整數y作位元的或or運算**
x^y	**整數x和整數y作位元的互斥或xor運算**
x&y	**整數x和整數y作位元的and運算**
x<<y	整數x被左位移y個位元
x>>y	整數x被右位移y個位元
-x	將x位元值反向,正的變負,負的變正

資料型態轉換成整數使用int()函數

- >>> int(3.14159)
- 3
- >>> int(1.0e5)
- 100000
- >>> int(1.0e-2)
- 0
- >>> int(True)
- 1
- >>> int(False)
- 0
- >>> int('8')
- 8
- >>> int('-8')
- -8

整數int的數值大小,可以無窮大無窮小

- >>> number=10**100
- >>> number
- 1000000000

2-1-2布林資料型態

- 有True和False兩種布林資料型態
- and真值表

A	B	A and B
True	True	True
True	False	False
False	True	False
False	False	False

or真值表

A	B	A or B
True	True	True
True	False	True
False	True	True
False	False	False

not真值表

A	not A
True	False
False	True

python

- >>> x=True
- >>> y=False
- >>> x and y
- False
- >>> x or y
- True
- >>> not y
- True
- >>> not x
- False

2-1-3浮點數型態

- Python提供三種浮點數值,float,complex和標準函式庫的decimal.Decimal三種型態
- 3.14159
- 1.0e9=1000000000
- 1.0e-3=0.001

float浮點數

- 輸入math數學模組
- >>> import math
- >>> math.pi
- 3.141592653589793
- >>> int(math.pi)
- 3
- >>> round(math.pi)
- 3
- >>> round(3.6)
- 4
- math.floor()是向下轉成最接近整數
- >>> math.floor(3.6)
- 3
- math.ceil()是向上轉成最接近整數
- >>> math.ceil(5.5)
- 6

將其它資料類型轉成浮點數使用float()

- >>> float(True)
- 1.0
- >>> float(False)
- 0.0
- >>> float(1e5)
- 100000.0
- >>> float(8)
- 8.0
- >>> float(90)
- 90.0

Complex複數

- 複數資料型態包含一對float浮點數,表現實數和虛數j.
- >>> z=58-3.14159j
- >>> z.real
- 58.0
- >>> z.imag
- -3.14159
- conjugate()函數可以變更虛數部分的正負號
- >>> z.conjugate()
- (58+3.14159j)
- >>>

Decimal數字

- Decimal數字,它的精確度是由我們來指定.
- Decimal的引數可以是整數,或是字串.但不可以是浮點數,因為浮點數IEEE754的表達方式可能會有誤差.
- 使用前先輸入decimal模組
- >>> import decimal
- >>> 46/1.1
- 41.81818181818181
- >>> decimal.Decimal(46)/decimal.Decimal('1.1')
- Decimal('41.818181818181818181818182')
- >>> print(decimal.Decimal(46)/decimal.Decimal('1.1'))
- 41.818181818181818181818182

2-1-4字串

- 可以使用雙引號”或單引號’包起來的字串
- >>> "林小寶"
- ‘林小寶'
- >>> '林曉雯'
- ‘林曉雯'
- >>> 'Alex Chen'
- 'Alex Chen'
- >>> 'Justin Wi'
- 'Justin Wi'
- >>> 'Peter Huang'
- 'Peter Huang'

- 字串符號,雙引號,裡面可以包單引號的字串

- >>> "請問'林小寶'是'林曉雯'的兒子嗎?"
- "請問'林小寶'是'林曉雯'的兒子嗎?"

多行文字包三重引號在裡面

- #!/usr/bin/env python3
- # -*- coding: utf-8 -*-
- “”” 中文編碼#-*-coding:utf-8-*-
- #”””為雙引號字串符號”””
- Created on Thu Jan 4 11:12:38 2018

python

- @author: justinwu
- “”’”’”’為單引號字串符號'''
- poem='''呱呱呱呱呱
- 醜小鴨呀醜小鴨
- 腿兒短短腳掌大
- 長長脖子扁嘴巴
- 走起路來搖呀搖
- 愛到河邊去玩耍
- 喉嚨雖小聲音大
- 可是只會呱呱呱'''
- print(len(poem))

- >>> """#為雙引號字串符號"""
- '#為雙引號字串符號'
- >>> '''為單引號字串符號'''

- '為單引號字串符號'
- >>> '''呱呱呱呱呱
- 醜小鴨呀醜小鴨
- 腿兒短短腳掌大
- 長長脖子扁嘴巴
- 走起路來搖呀搖
- 愛到河邊去玩耍
- 喉嚨雖小聲音大
- 可是只會呱呱呱'''
- '呱呱呱呱呱\n醜小鴨呀醜小鴨\n腿兒短短腳掌大\n長長脖子扁嘴巴\n走起路來搖呀搖\n愛到河邊去玩耍\n喉嚨雖小聲音大\n可是只會呱呱呱'
- >>>

print()會自動將字串引號去除

- >>> print("""#為雙引號字串符號""")
- #為雙引號字串符號
- >>> print('''為單引號字串符號''')
- 為單引號字串符號
- >>> print('為單引號字串符號')
- 為單引號字串符號

空白字元

- >>> ''
- ''
- >>> ""
- ''
- >>> ''' '''
- ' '
- >>> """"""""
- ''
- >>> """" """"
- ' '
- >>> x=''
- >>> print(x)

- >>>

將其他資料型態轉成字串使用str()函數

- 字串符號包起來的為字串
- >>> str(3.14159)
- '3.14159'
- >>> str(1.0e5)
- '100000.0'
- >>> str(1.0e-2)
- '0.01'
- >>> str(True)
- 'True'
- >>> str(88999)
- '88999'

用\來代表特殊含義

- \n為換行
- \tab 為空tab鍵
- \’代表單引號
- \”代表雙引號
- \\代表反斜線

\n為換行符號

- >>> x='醜小鴨呀醜小鴨\n腿兒短短腳掌大\n長長脖子扁嘴巴\n'
- >>> print(x)
- 醜小鴨呀醜小鴨
- 腿兒短短腳掌大
- 長長脖子扁嘴巴

\tab 為空tab鍵

- >>> x='\t醜小鴨呀醜小鴨\n腿兒短短腳掌大\n長長脖子扁嘴巴\n'
- >>> print(x)
- 醜小鴨呀醜小鴨
- 腿兒短短腳掌大
- 長長脖子扁嘴巴

\’代表單引號

\”代表雙引號

- >>> x=""\'為單引號字串符號\""
- >>> print(x)
- ‘為單引號字串符號'
- >>> x="""\"#為雙引號字串符號\"""
- >>> print(x)
- "#為雙引號字串符號"

\\代表反斜線

- >>> x='\\代表反斜線'
- >>> print(x)
- \代表反斜線

用[]符號來擷取字元

- >>> words='abcdefghijklmnopqrstuvwxyz'
- >>> words[0]
- 'a'
- >>> words[1]
- 'b'
- >>> words[-1]
- 'z'
- >>> words[-2]
- 'y'
- >>> words[25]
- 'z'

使用replace()來更改某個字元

- >>> name='Peter'
- >>> name.replace('P','p')
- 'peter'
- >>> name
- 'Peter'

切片運算子[開始]

- [開始:結束]
- 用[開始:結束:間隔]來擷取元素

```
>>> words='abcdefghijklmnopqrstuvwxyz'
>>> print(words[:])
abcdefghijklmnopqrstuvwxyz
>>> print(words[0:2])
ab
>>> print(words[24:25])
y
>>> print(words[0:26])
abcdefghijklmnopqrstuvwxyz
>>> print(words[0:26:2])
acegikmoqsuwy
```

2-1-5物件類別

- 類別是抽象的，而物件就是類別的實體。
- 物件也是Python的資料型態。
- 類別class的實體就是物件型態
- >>> a='justin'
- >>> type(a)
- <class 'str'>
- >>> x=5
- >>> type(x)
- <class 'int'>
- >>> y=8.9
- >>> type(y)
- <class ‘float’>

python

- >>> z=(1,2,3,4,5)
- >>> type(z)
- <class 'tuple'>
- >>> data=[1,2,3,4,5]
- >>> type(data)
- <class 'list'>
- >>> dic1={'a':1,'b':2,'c':3}
- >>> type(dic1)
- <class 'dict'>

2-2識別名稱

- 每一個變數都有識別名稱。
- 我們變數宣告時，就是給該變數一個識別名稱。
- 所有的識別名稱是由字元、數字、下底線(_)所組成。
- Python是有大小寫的區隔，a和A是不一樣的變數。

Python關鍵保留字

- Python關鍵字有大小寫之分

and	elif	finally	lambda	while
as	else	for	none cal	with
assert	yield	from	not	
break	TRUE	global	or	
class	yield	if	pass	
continue	with	import	raise	
def	except	in	return	
del	False	is	try	

關鍵字模組keyword

```
>>> import keyword
>>> print(keyword.kwlist)
['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', 'continue'
, 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global',
'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise
', 'return', 'try', 'while', 'with', 'yield']
```

2-3運算式與運算子

- 2-3-1邏輯運算子
- 2-3-2關係運算子
- 2-3-3算術運算子

運算式與運算子

- 運算式是由運算子與運算元組成
- +加-減*乘/除是運算子,先乘除後加減的結合優先順序
- 運算元是變數,數字,字串,物件類別和資料結構
- =是分配符號,將右邊的值分配給左邊變數

數值資料型態與運算子

- 變數a=1+1,c，c=5，變數c的值為5在這個中
- a=1+1是一個運算式
- c=5是一個運算式
- +和=是運算子，變數a和數值5，和數值1是運算元。
- 運算式就是由運算子和運算元所組成。

2-3-1邏輯運算子

- 邏輯運算子and,or,not
- 身份運算子is
- 成員運算子in

邏輯運算子and,or,not

- Python提供了三個邏輯運算子and,or,not.可以使用短路邏輯short-circuit邏輯
- 運算式出現在布林中,求值結果就是布林值
- >>> x=True
- >>> y=False
- >>> x and y
- False
- >>> x or y
- True
- >>> not y
- True

身份運算子is

- >>> x='justin'
- >>> x='Johny'
- >>> y='Ivy'
- >>> x is y
- False
- >>> y=x
- >>> x is y
- True
- >>> print(y)
- Johny
- >>> print(x)
- Johny

隸屬運算子in

- 對於群集資料數組,串列和陣列,我們可以使用in運算子來測試隸屬關係
- >>> x=(1,2,3,4,5,6,7,8)
- >>> 5 in x
- True
- >>> 9 in x
- False
- >>> 1 in x
- True

2-3-2關係運算子

- 比較運算子和邏輯運算子所得到的結果就是一個布林值(True或False)

比較運算子	功能
<	小於
>	大於
>=	大於等於
!=	不相等
<=	小於等於
==	相等

python

- >>> x=3
- >>> y=5
- >>> x<y
- True
- >>> x>y
- False
- >>> x>=y
- False
- >>> x==y
- False
- >>> z=3
- >>> x==z
- True
- >>> x!=y
- True
- >>> x<=y
- True

2-3-3算術運算子

運算子	名稱
+	加法
-	減法
*	乘法
/	除法
**	指數
%	取餘數
//	整數除法

運算子	名稱
+=	加法指定
-=	減法指定
*=	乘法指定
/=	除法指定
**=	指數指定
%=	取餘數指定

python

- >>> 8/2
- 4.0
- >>> 2/8
- 0.25
- >>> 2**3
- 8
- >>> 3**2
- 9
- >>> 5%2
- 1
- >>> 5%3
- 2
- >>> 9//2
- 4

python

2-4運算子結合優先順序

Arithmatic.py × operator_preference.py ×

```python
x=2**3+5*2-5/5+5%2
y=1<<2
print(x)
print(y)
b=3
c=3
d=b&c
print(d)
e=b^c
print(e)
f=b|c
print(f)
g=1<5
print(g)
h=1>5
print(h)
```

```
18.0
4
3
0
3
True
False
```

運算子優先順序,先乘除後加減

Arithmatic.py

```python
a = 1+1
b = 2-1
div=6/2
pi=3.14159
radius=10
area=radius*radius*pi
c=5
x=2*3+4/2+1-3%2+5
print(a)
print(b)
print(div)
print(area)
print(c)
print(x)
```

thmatic

```
/Users/justinwu/PycharmProjects/Pytho
2
1
3.0
314.159
5
13.0
```

2-5數字，字串與變數

- 變數名稱只能使用小寫字母(a到z),大寫字母(A到Z),數字(0到9),底線(_)
- 名稱的第一個字元不能用數字,Python會將底線開頭的名稱視為特殊用途(例如__main__)

變數

- 變數可以儲存下列資料型態的資料
- 資料型態
  - 整數
  - 浮點數
  - 字串
  - 物件參考
  - 布林值

python

```python
#!/usr/bin/env python3
#_*_coding:utf-8_*_
"""
Created on Thu Oct 26 07:40:42 2017

@author: justinwu
"""
#這是註解

#這是註解
1+2
#print(1+2)
x=2-1
print(x)
y=3+2
print(y)
z=3.2*2
print(z)
str='大家好'
print(str)
str2='Python'
mychar=str[0]
print(mychar)
```

Name	Type	Size	
mychar	str	1	大
str	str	1	大家好
str2	str	1	Python
x	int	1	1
y	int	1	5
z	float	1	6.4

Console 1/A

```
In [28]: runfile('/Users/ju
wdir='/Users/justinwu/Deskt
1
5
6.4
大家好
大

In [29]:
```

python

```python
#!/usr/bin/env python3
#_*_coding:utf-8_*_
"""
Created on Thu Oct 26 07:40:42 2017

@author: justinwu
"""
#這是註解

#這是註解
1+2
#print(1+2)
x=2-1
print(x)
y=3+2
print(y)
z=3.2*2
print(z)
str='大家好'
print(str)

```

Name	Type	Size	
str	str	1	大家好
x	int	1	1
y	int	1	5
z	float	1	6.4

Console 1/A

```
In [22]: runfile('/Users/j
wdir='/Users/justinwu/Desk
1
5
6.4
大家好

In [23]:
```

python

- Thanks.

Python 精通程式語言
資料結構

吳佳諺 老師

資料結構

資料結構
4.控制結構
5.函數
6.類別
7.繼承
8.異常或錯誤處理
9.使用matplotlib畫圖

資料結構

- 變數
- 運算式與運算子
- 串列
- 堆疊
- 佇列

變數

- 資料型態
  - 整數
  - 浮點數
  - 字串

```python
#!/usr/bin/env python3
#_*_coding:utf-8_*_
"""
Created on Thu Oct 26 07:40:42 2017

@author: justinwu
"""
#這是註解

#這是註解
1+2
#print(1+2)
x=2-1
print(x)
y=3+2
print(y)
z=3.2*2
print(z)
str='大家好'
print(str)
str2='Python'
mychar=str[0]
print(mychar)
```

Name	Type	Size	
mychar	str	1	大
str	str	1	大家好
str2	str	1	Python
x	int	1	1
y	int	1	5
z	float	1	6.4

Console 1/A

```
In [28]: runfile('/Users/ju
wdir='/Users/justinwu/Deskt
1
5
6.4
大家好
大

In [29]:
```

運算式與運算子

- 運算式是由運算子與運算元組成
- +加-減*乘/除是運算子,先乘除後加減的結合優先順序
- 運算元是變數,數字,字串和資料結構
- =是分配符號,將右邊的值分配給左邊變數

```python
#!/usr/bin/env python3
#_*_coding:utf-8_*_
"""
Created on Thu Oct 26 07:40:42 2017

@author: justinwu
"""
#這是註解

#這是註解
1+2
#print(1+2)
x=2-1
print(x)
y=3+2
print(y)
z=3.2*2
print(z)
str='大家好'
print(str)

```

Name	Type	Size	
str	str	1	大家好
x	int	1	1
y	int	1	5
z	float	1	6.4

Console 1/A

```
In [22]: runfile('/Users/j
wdir='/Users/justinwu/Desk
1
5
6.4
大家好

In [23]:
```

串列

```
>>> fruits = ['orange','apple','pear','banana','kiwi','apple','banana']
>>> fruits.count('apple')
2
>>> fruits.index('banana')
3
>>> fruits.index('banana',4)#Finding next banana starting a position 4
6
>>> fruits.reverse()
>>> fruits
['banana', 'apple', 'kiwi', 'banana', 'pear', 'apple', 'orange']
>>> fruits.append('grape')
>>> fruits
['banana', 'apple', 'kiwi', 'banana', 'pear', 'apple', 'orange', 'grape']
>>> fruits.sort()
>>> fruits
['apple', 'apple', 'banana', 'banana', 'grape', 'kiwi', 'orange', 'pear']
>>> fruits.pop()
'pear'
>>>
```

堆疊

Python 3.6.2 Shell

```
Python 3.6.2 (v3.6.2:5fd33b5926, Jul 16 2017, 20:11:06)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> WARNING: The version of Tcl/Tk (8.5.9) in use may be unstable.
Visit http://www.python.org/download/mac/tcltk/ for current information.

>>> stack=[3,4,5]
>>> stack.append(6)
>>> stack.append(7)
>>> stack
[3, 4, 5, 6, 7]
>>> stack.pop()
7
>>> stack.pop()
6
>>> stack
[3, 4, 5]
>>>
```

佇列

```python

from collections import deque
queue = deque(['阿呆','Eric','John','Michael','小寶','小文'])
queue.append("Terry")
queue.append("Graham")
print(queue.popleft())
print(queue.popleft())
print(queue)

```

```
In [1]: runfile('/Users/justinwu/Desktop/queue.py', wdir='/Us
阿呆
Eric
deque(['John', 'Michael', '小寶', '小文', 'Terry', 'Graham'])

In [2]:
```

數組tuple,集合set和字典

- 可以用數組tuple來儲存固定的元素,使用小括號()來建立一數組tuple
- 集合的元素放置沒有按照順序,可以使用{}大括號來建立一集合Set
- 集合加上索引就是字典{索引:值}

Tuple數組

- 也可以從字串中建立數組
- tp5 = tuple(‘Ivy Lin’)
- 從數組得到串列
- list1 = list(tp5)

```python

tp1=()
print(tp1)
tp2=(1,2,3,4,5,6,7,8)
print(tp2)
print(sum(tp2))
print('--------------------')
print(tp2[2:5])#切割運算子
print(tp2[-1])
tp3 =tuple([2*x for x in range(1,8)])
print(tp3)
print('----------------')
tp4=tuple('Ivy Lin')
print(tp4)
tp5=("John",'小寶','小文')
print(tp5)
print(len(tp5))
print(tp4+tp5)
print('--------------')
tp6=tuple([1,2,3,4,5,6,7,8,9])
print(tp6)
print(max(tp6))
print(min(tp6))

```

```
In [6]: runfile('/Users/justinwu/Desktop/tuple.py', wdir='
Desktop')
()
(1, 2, 3, 4, 5, 6, 7, 8)
36
--------------------
(3, 4, 5)
8
(2, 4, 6, 8, 10, 12, 14)
----------------
('I', 'v', 'y', ' ', 'L', 'i', 'n')
('John', '小寶', '小文')
3
('I', 'v', 'y', ' ', 'L', 'i', 'n', 'John', '小寶', '小文')
--------------
(1, 2, 3, 4, 5, 6, 7, 8, 9)
9
1

In [7]:
```

```python

tp6=tuple([66,22,3,46,5,65,7,83,19])
print(tp6)
list1 = list(tp6)
list1.sort()#排序串列
print(list1)
print('-----------')
tp8=tuple(list1)
tp9=tuple(list1)
print(tp8)
print(tp8 == tp9)#比較兩個數組tuple


```

```
In [15]: runfile('/Users/justinwu/
Desktop')
(66, 22, 3, 46, 5, 65, 7, 83, 19)
[3, 5, 7, 19, 22, 46, 65, 66, 83]
-----------
(3, 5, 7, 19, 22, 46, 65, 66, 83)
True

In [16]:
```

Set集合

- 集合(set)用來儲存沒有重複的元素.
- 集合的元素是不可以複製的,元素放置也沒有按照順序
- 可以使用{}大括號來建立一集合Set

Set集合

```python

st1=set()#建立一個空集合
st2=set([1,2,3,4,5])
print(st2)
st3={'a','b','c','d','e'}
print(st3)
print('-------------------')
st3.add('f')
print(st3)
st3.remove('d')
print(st3)
print('----------')
print(st3.union(st2))
st5={'a','b','c','d','e'}
print(st3.intersection(st5))
print(st3.difference(st5))
```

```
In [30]: runfile('/Users/justinwu/Deskto
Desktop')
{1, 2, 3, 4, 5}
{'d', 'a', 'e', 'c', 'b'}
-------------------
{'d', 'f', 'a', 'e', 'c', 'b'}
{'f', 'a', 'e', 'c', 'b'}
----------
{1, 2, 3, 4, 5, 'f', 'a', 'e', 'c', 'b'}
{'b', 'a', 'c', 'e'}
{'f'}

In [31]:
```

字典

- 集合加上索引就是字典{索引:值}

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Oct 27 19:40:39 2017

@author: justinwu
"""
tel={'Justin':'0920909872','Ivy':'0922876895'}
tel['Johny']='0920885356'
print(tel)
print(tel['Johny'])
del tel['Johny']
tel['Mary']='0922865255'
print(tel)
print(list(tel.keys()))
print(sorted(tel.keys()))
print('Ivy' in tel)
print('Johny' in tel)
```

```
In [5]: runfile('/Users/justinwu/Desktop/TOP/python/
example/dictionary_1.py', wdir='/Users/justinwu/Desktop/
TOP/python/example')
{'Justin': '0920909872', 'Ivy': '0922876895', 'Johny':
'0920885356'}
0920885356
{'Justin': '0920909872', 'Ivy': '0922876895', 'Mary':
'0922865255'}
['Justin', 'Ivy', 'Mary']
['Ivy', 'Justin', 'Mary']
True
False

In [6]:
```

python

- Thanks.

Python 精通程式語言
控制結構

吳佳諺 老師

4.控制結構

4-1邏輯運算子

4-2一個選擇的if敘述

4-3巢狀的if敘述與多重選擇決策

4-4迴圈結構for

4-5迴圈結構while

控制結構

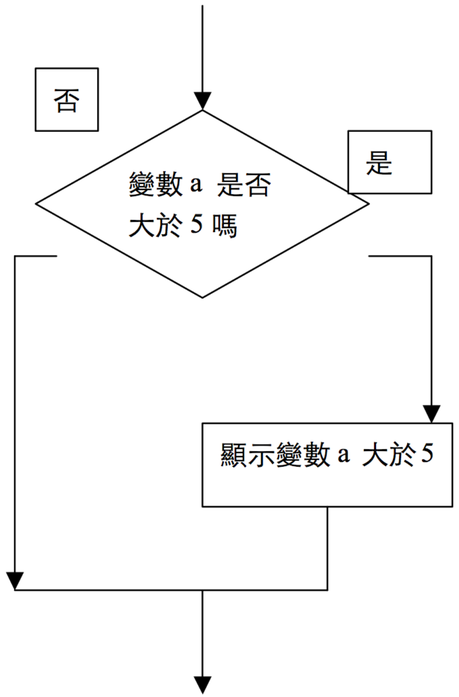
- 選取結構if
- 迴圈結構while,for

選取結構if

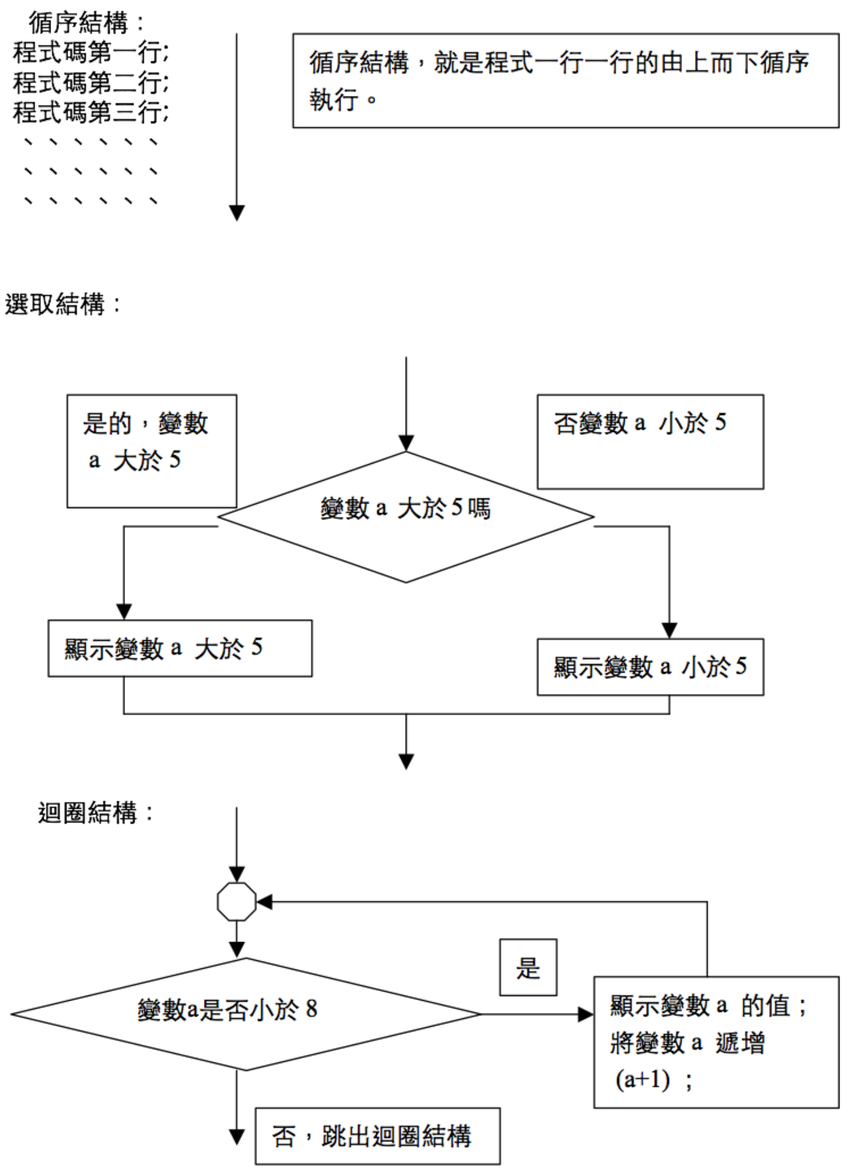
- 語法if:
- if 條件運算式:
- 　程式敘述1
- else:
- 　程式敘述2

布林運算式

- 如何來選擇流程前進的方向，我們必須經過測試條件，例如，當條件成立時往左方，當條件不成立時往右方。我們使用布林表示式來測試工作。
- 布林Boolean代數定義在一個二元素的集合上，即B={true,false}，true為真，false為假。我們可以使用這個值的結果來決定我們行進的方向。
- 當下列菱形四邊形成立true時會執行右方的流程，當下列菱形四邊行的條件不成立false時會執行左方的流程。true和false就是屬於布林代數，這是用在if判別式。



- 當下列菱形四邊形成立true時會執行右方的流程，當下列菱形四邊行的條件不成立false時會執行下方或右方的流程。True和false就是屬於布林代數，這是運用在迴圈結構。



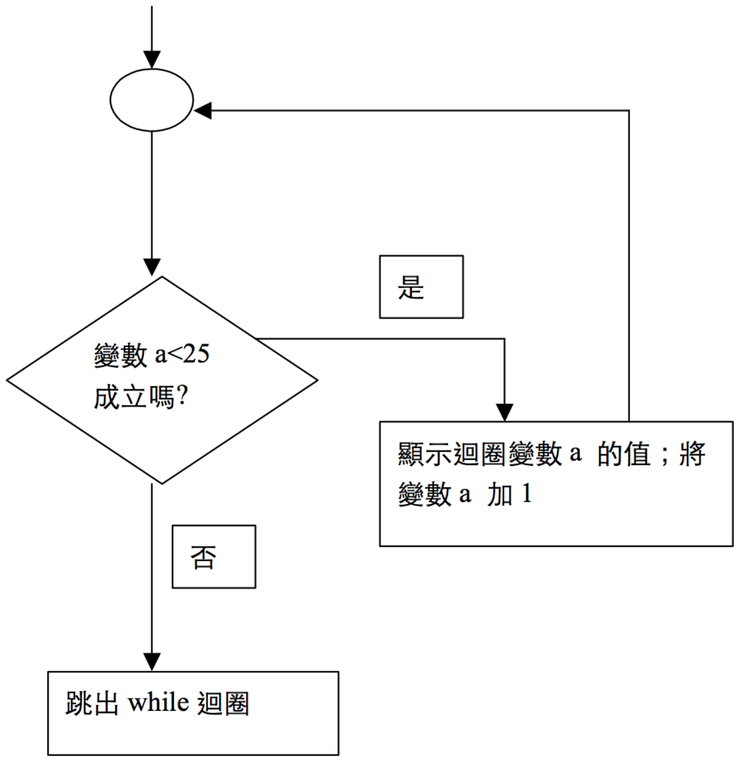
選取結構if

```
8
9  a = int(input("請輸入薪水:"))
10 if a < 50000:
11     print("薪水小於50000")
12 else:
13     print("薪水大於50000")
```

```
Console 4/A

Python 3.6.2 |Anaconda, Inc.|
Type "copyright", "credits" or

IPython 6.1.0 -- An enhanced I
薪水小於50000請輸入薪水:38000


In [2]:

In [2]: runfile('/Users/justin

請輸入薪水:85000
薪水大於50000

In [3]:
```

while迴圈

- 在if敘述中，條件後的敘述只執行一次，而在while敘述中，則可執行一次以上。While敘述的程序圖形中.選取結構和循序結構，都只執行程式敘述一次，如果我們要讓同一行程式重複執行好幾遍則要用迴圈敘述。迴圈敘述可以重複執行某一段程式好幾遍，直到條件的不成立才跳出這個迴圈。迴圈敘述：while、do……while。



迴圈結構for,while

迴圈結構for
迴圈結構while

迴圈結構for

- 語法:
- for 計數變數 in range(起始值,終始值):
  - 程式敘述

range(8,19)為8到18的數值

```
8
9 for i in range(8,19):
10     print("i的值:",i)
```

```
Console 5/A
Python 3.6.2 |An
Type "copyright"

IPython 6.1.0 --

In [1]: runfile(
i的值: 8
i的值: 9
i的值: 10
i的值: 11
i的值: 12
i的值: 13
i的值: 14
i的值: 15
i的值: 16
i的值: 17
i的值: 18
```

迴圈結構while

```
8
9 i=5
10 while i<=10:
11     print("i:",i)
12     i=i+1
```

```
Console 6/A
Python 3.6.2 |Ana
Type "copyright",

IPython 6.1.0 --

In [1]: runfile('
i: 5
i: 6
i: 7
i: 8
i: 9
i: 10

In [2]:
```

布林運算式

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Oct 26 14:30:21 2017

@author: justinwu
"""

x=True
y=False
print(x&y)
print(x|y)
if (x&y):
    print(x&y)
else:
    print(x&y)
z=True
print(z&y)
print(z|y)
```

```
In [30]: runfil
Users/justinwu/
False
True
False
False
True

In [31]:
```

continue繼續執行迴圈

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Oct 26 14:07:44 2017

@author: justinwu
"""

for i in range(8,19):
    if(i%2==1):
        continue
    print("i的值:",i)
```

```
In [19]: ru
Users/justi
i: 5
i: 6
i: 7
i: 8
i: 9

In [20]:
```

break跳出while迴圈

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Oct 26 14:09:11 2017

@author: justinwu
"""

i=5
x =True
while x:
    print("i:",i)
    i=i+1
    if(i>=10):
        break;

```

```
In [19]: runfile('
Users/justinwu/Des
i: 5
i: 6
i: 7
i: 8
i: 9

In [20]:
```

4-1邏輯運算子

- 邏輯運算子可以結合條件，以一個表達式判斷許多條件，而這些條件的結果不是真True就是假False。
- and稱為”與邏輯運算子”，只有當所有條件都成立時才會回傳真True，否則回傳假False。
- or稱為或邏輯運算子，只要運算式中一個條件成立就會回傳真True，只有當所有的條件都為假Fasle時，才會回傳假False。
- not為相反邏輯運算子，真True的條件加上not相反邏輯運算子時，就會變成假False；當假False的條件加上not相反邏輯運算子時，就會變成真True。

邏輯運算子

```
1   x=True
2   y=False
3   z=x and y
4   print(z)
5   z1=x or y
6   print(z1)
7   z2=not x
8   z3=not y
9   print(z2)
10  print(z3)
```

```
/Users/justinwu/PycharmProjects/Pyth
False
True
False
True

Process finished with exit code 0
```

4-2一個選擇的if敘述

- if 條件:

  敘述

一個選擇的if敘述

```
1   x=8
2   if x>5 :
3       print("x is bigger than 5")
```

trol_if

```
/Users/justinwu/PycharmProjects/Pytho
x is bigger than 5

Process finished with exit code 0
```

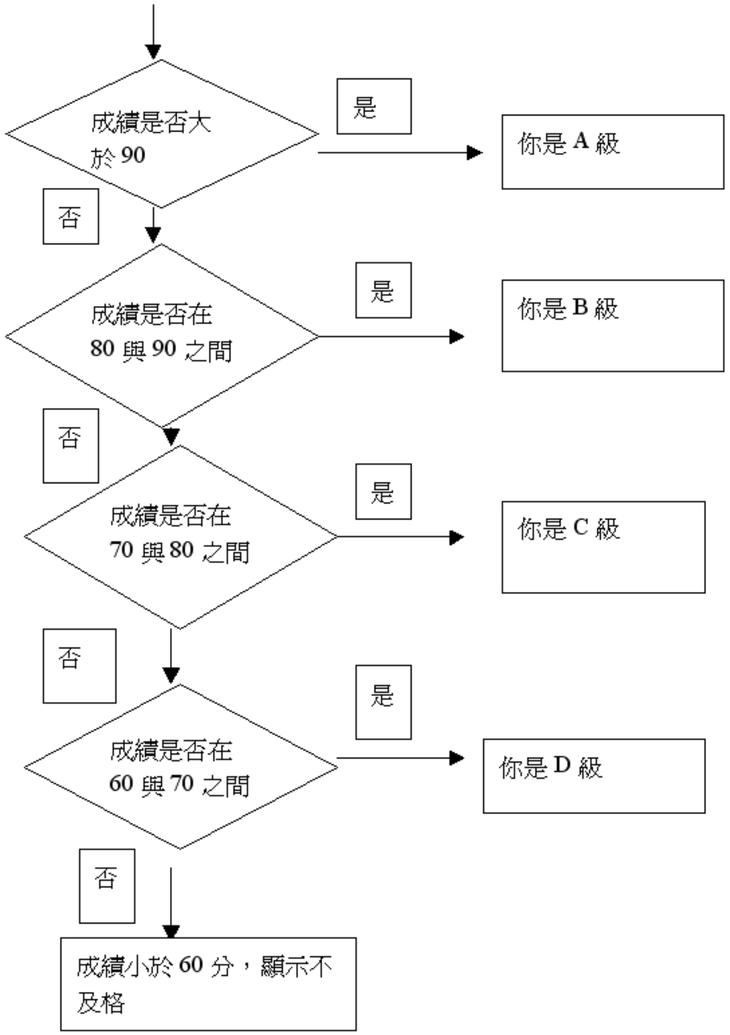
一個選擇的if控制

```
if (3>5):
    print("3大於5")

if (3<5):
    print("3小於5")

if (5==5):
    print("5等於5")

if (3>5)or(3<5):
    print("只要有一個成立就為真")

if (not(3>5)):
    print("不會3大於5,這是真")

if (3>5)and(3<5):
    print("HelloWorld")
```

```
3小於5
5等於5
只要有一個成立就為真
不會3大於5,這是真
```

4-3巢狀的if敘述與多重選擇決策

- 我們已經看過if敘述1個或2個的選擇，在此要用巢狀if敘述來撰寫多重選擇決策。例如我們的成績評等，90到100分的為A級，80到89的為B級，70到79的為C級，60到69的為D級，59分以下的為F級。如果我有一位學生分數為82分，他的分數在80到89分之間，則他的等級為B級。在這個例子中有5個等級來做選擇，就叫做多重選擇。


成績是否大於 90
是
你是 A 級
否
成績是否在 80 與 90 之間
是
你是 B 級
否
成績是否在 70 與 80 之間
是
你是 C 級
否
成績是否在 60 與 70 之間
是
你是 D 級
否
成績小於 60 分，顯示不及格

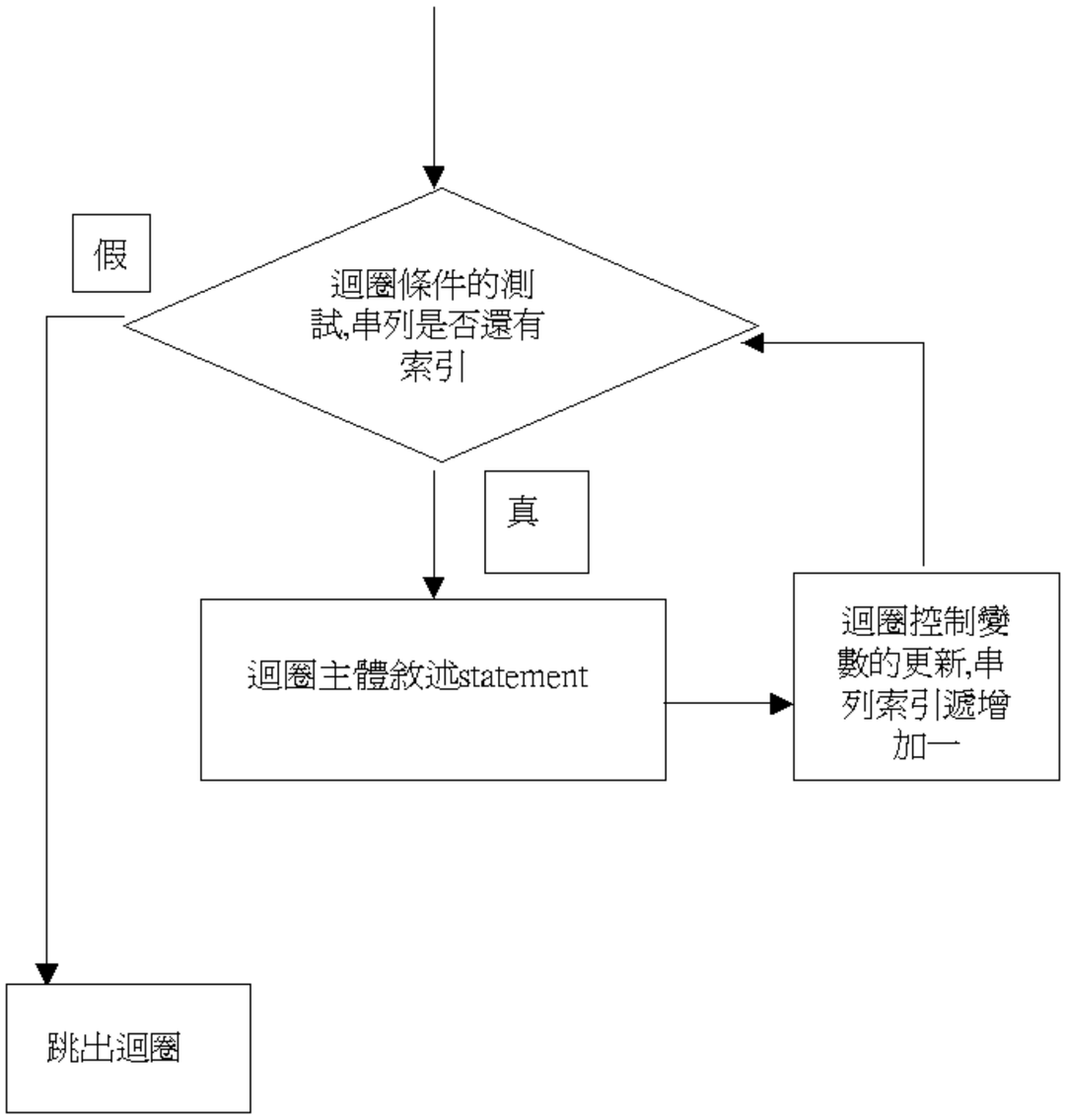
- 語法:
- if (條件1):
    - 敘述1
- elif (條件2):
    - 敘述2
- elif (條件3):
    - 敘述3
- elif (條件4):
    - 敘述4
- else (條件5):
    - 敘述5

```python
score = int(input("請輸入成績: "))
if(score>=90):
    print("你是A級")
elif (score >= 80) and (score<=89):
    print("你是B級")
elif (score >= 70) and (score<=79):
    print("你是C級")
elif (score >= 60) and (score <= 69):
    print("你是D級")
else :
    print("你不及格")
```

- 請輸入成績: 99
- 你是A級

4-4迴圈結構for

- Python提供for敘述作為迴圈，除了迴圈主體敘述(statement)，還有迴圈的條件結構。


假
迴圈條件的測試,串列是否還有索引
真
迴圈主體敘述statement
迴圈控制變數的更新,串列索引遞增加一
跳出迴圈

迴圈結構for

```
1  words =['airplane','cat','dog','bear','car']
2  for w in words:
3      print(w,len(w))
```

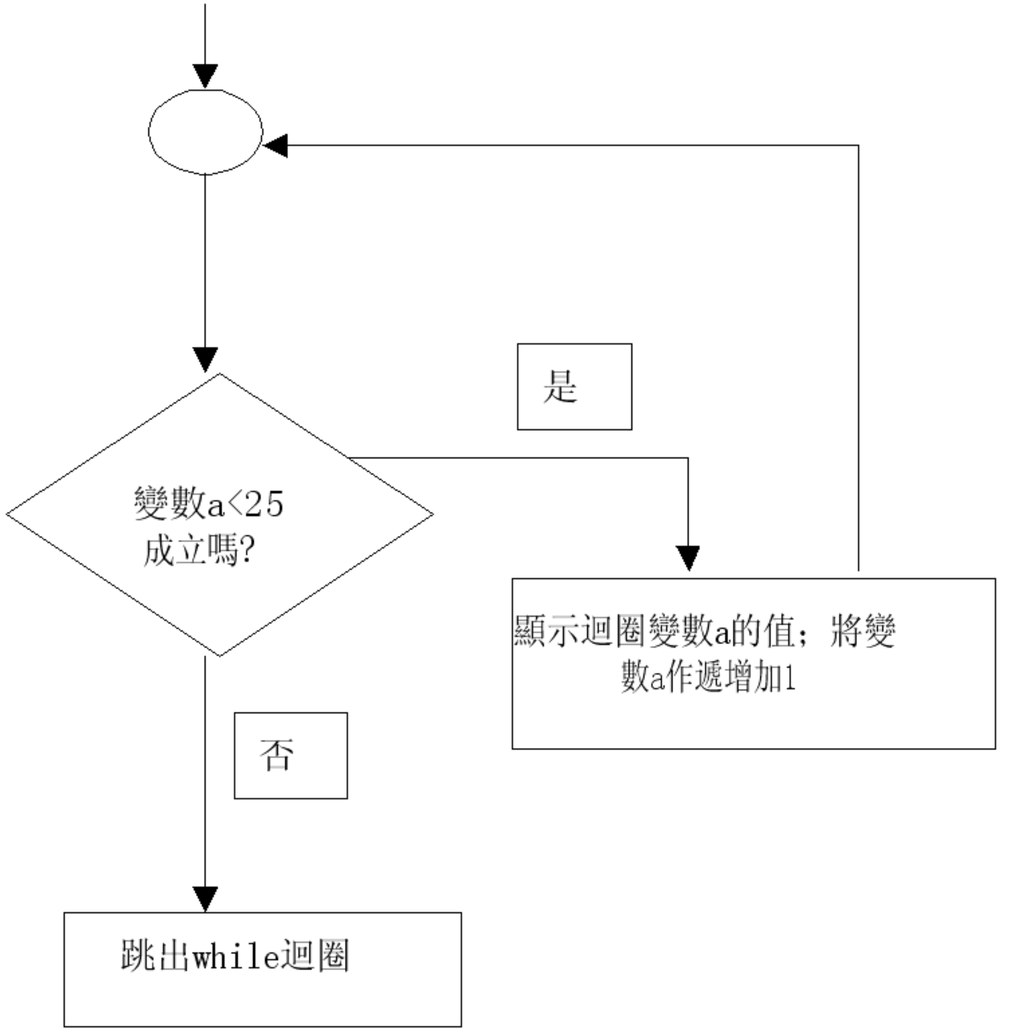
for w in words

pass | for

```
/Users/justinwu/untitled3/bin/python "/Users/justinwu/Desk
airplane 8
cat 3
dog 3
bear 4
car 3
```

4-5迴圈結構while

- 在if敘述中，條件後的敘述只執行一次，而在while敘述中，則可執行一次以上。While敘述的程序圖形中


是
變數a<25
成立嗎?
顯示迴圈變數a的值；將變
數a作遞增加1
否
跳出while迴圈

```python
a=20
while a < 25:
    a=a+1
    print(a)
```

pass while

```
/Users/justinwu/untitled3/bin/python
21
22
23
24
25
```

python

- Thanks.

Python 精通程式語言

函數

吳佳諺 老師

函數

5-1pass敘述沒做任何事
5-2預設參數值
5-3使用者自訂函數
5-4區域變數的生存空間
5-5遞迴函數
5-6函數參數與引數
5-7套件模組
5-8Python內建字串相關函數

5.函數

```
i = 10
def f():
    print(i)

i = 42
f()
```

函數與參數

```
14
15 def addf(x,y):
16     print(x+y)
17
18 i1=23
19 i2=12
20 addf(23,12)
21
22
```

return回傳

```python
i = 10
def f():
    print(i)

i = 42
f()

def addf(x,y):
    print(x+y)

i1=23
i2=12
addf(23,12)

def myreturn(x,y):
    return x*y

i3=2
i5=5
i8=myreturn(i3,i5)
print(i8)

def myreturn(a,x=2,y=3):
    return a*x*y

i3=2
i5=5
i8=myreturn(8,i3,i5)
print(i8)
```

```
Console 2/A
Python 3.6.2 |Ana
Type "copyright"

IPython 6.1.0 --

In [1]: runfile(
42
35
10
80

In [2]:
```

遞迴函數

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Oct 26 16:25:09 2017

@author: justinwu
"""

def factorial(n):
    if(n==0):
        return 0
    if(n==1):
        return 1
    else:
        return n*factorial(n-1)

print(factorial(1))
print(factorial(2))
print(factorial(3))
```

```
In [2]: runfile('/Us
wdir='/Users/justinw
1
2
6

In [3]:
```

費式係數

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Oct 26 16:28:50 2017

@author: justinwu
"""

def fibonaci(n):
    if(n==0):
        return 0
    if(n==1):
        return 1
    else:
        return fibonaci(n-1)+fibonaci(n-2)

print(fibonaci(1))
print(fibonaci(2))
print(fibonaci(3))
print(fibonaci(10))
print(fibonaci(20))
```

```
In [2]: runfile
wdir='/Users/ju
1
2
6

In [3]:
```

5-1pass敘述沒做任何事

```
1  def Hello():
2      pass
3
4  Hello()
5  print("Helloworld")
```

pass | initlog

```
/Users/justinwu/untitled3/bin/py
Helloworld
```

5-2預設參數值

```python
def ask_ok(prompt,retries=5,reminder='Please try again'):
    while True:
        ok=input(prompt)
        if ok in ('yes'):
            return True
        if ok in ('no'):
            retries=retries-1
            if retries<0:
                raise ValueError('invalid user response')
            print(reminder)


ask_ok('離開按yes或no:')
ask_ok('ok',2)
ask_ok('ok',2,'請再試一次')
```

pass function

```
/Users/justinwu/untitled3/bin/python "/Users/justinwu/Desktop/TOP/Pythor
離開按yes或no:yes
okno
Please try again
okno
Please try again
okno
```

5-3使用者自訂函數

```python
def ask_ok(prompt,retries=5,reminder='Please try again'):
    while True:
        ok=input(prompt)
        if ok in ('yes'):
            return True
        if ok in ('no'):
            retries=retries-1
            if retries<0:
                raise ValueError('invalid user response')
            print(reminder)


ask_ok('離開按yes或no:')
ask_ok('ok',2)
ask_ok('ok',2,'請再試一次')
```

pass function

```
/Users/justinwu/untitled3/bin/python "/Users/justinwu/Desktop/TOP/Pythor
離開按yes或no:yes
okno
Please try again
okno
Please try again
okno
```

5-4區域變數的生存空間

- 函數的參數為區域變數.函數的參數生存空間只在函數的範圍內
- 在函數內定義的變數,其作用範圍在函數內

區域變數的生存空間

```python
def ask_ok(prompt,retries=5,reminder='Please try again'):
    while True:
        ok=input(prompt)
        if ok in ('yes'):
            return True
        if ok in ('no'):
            retries=retries-1
            if retries<0:
                raise ValueError('invalid user response')
            print(reminder)


ask_ok('離開按yes或no:')
ask_ok('ok',2)
ask_ok('ok',2,'請再試一次')
```

prompt, retries,reminder

變數生存的範圍

```
pass   function
/Users/justinwu/untitled3/bin/python "/Users/justinwu/Desktop/TOP/Pythor
離開按yes或no:yes
okno
Please try again
okno
Please try again
okno
```

5-5遞迴函數

- 這是費氏系數的問題。費氏級數是由0和1開始，每一個費氏級數的數，是由前兩個費氏級數的合所組成。

費氏級數	0	1	1	2	3	5	8	13	21	34	55
索引	0	1	2	3	4	5	6	7	8	9	10

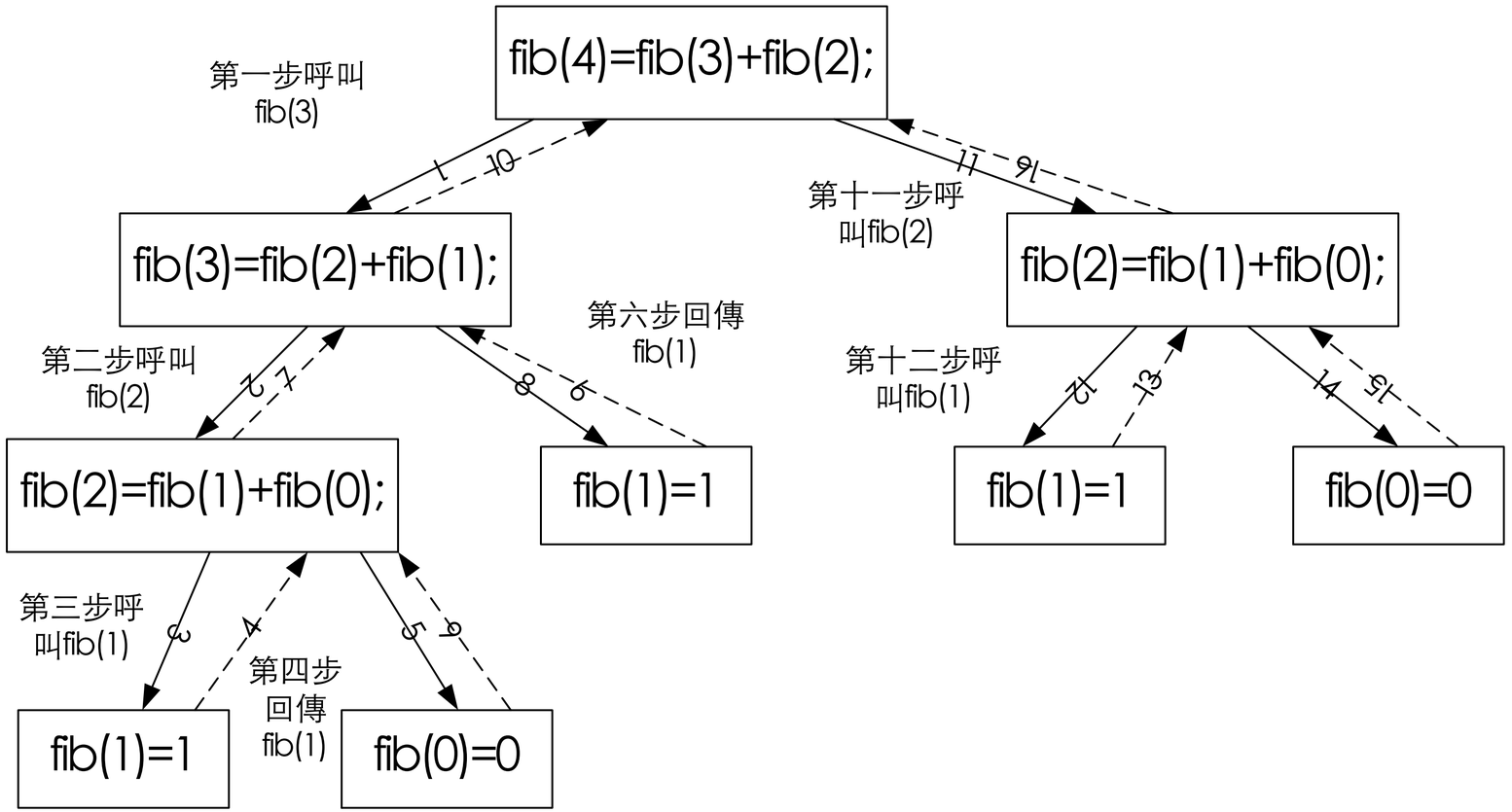
- 我們可以將費氏級數定義成。
- 遞迴函數有兩個條件。一個是最終遞迴的值，然後開始回傳；一個就是遞迴函數的運算式。這兩個條件就可以組成遞迴函數。fib()最終的值是fib(0)=0和fib(1)=1；遞迴函數的運算式是fib(index)=fib(index -1)+fib(index-2);。

```python
def fibonaci(n):
    if (n == 0):
        return 0
    if (n == 1):
        return 1
    else:
        return fibonaci(n - 1) + fibonaci(n - 2)

```

naci

```
/Users/justinwu/untitled3/bin/python "/Users/justinwu/D
3
```

遞迴函數


fib(4)=fib(3)+fib(2);
第一步呼叫
fib(3)
1
10
11
16
第十一步呼
叫fib(2)
fib(3)=fib(2)+fib(1);
fib(2)=fib(1)+fib(0);
第六步回傳
fib(1)
第二步呼叫
fib(2)
2
7
8
9
第十二步呼
叫fib(1)
12
13
14
15
fib(2)=fib(1)+fib(0);
fib(1)=1
fib(1)=1
fib(0)=0
第三步呼
叫fib(1)
3
4
5
6
第四步
回傳
fib(1)
fib(1)=1
fib(0)=0

5-6函數參數與引數

- 費氏係數使用迴圈方式實作

```
def fib(n):    # write Fibonacci series up to n
    """Print a Fibonacci series up to n."""
    a, b = 0, 1
    while a < n:
        print(a, end=' ')
        a, b = b, a+b
    print()
```

函數的名稱和參考位址

- 可以將使用者定義的函數分配給變數

```
17 fib(2000)
18 print(fib)
19 f = fib
20 print(f)
21 f(100)
```

將函數的參考位址列印出來

```
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597
<function fib at 0x10fc60488>
<function fib at 0x10fc60488>
0 1 1 2 3 5 8 13 21 34 55 89
```

參數預設值的給值

- 第九行將5分派給i變數
- 第十一行會將i分派給參數arg

```
 9 i = 5
10
11 def f(arg=i):
12     print(arg)
13
14 i = 6
15 f()
16
17
```

```
Console 1/A

In [9]: runf
wdir='/Users/
5
```

當函數預設值是物件參考時

- 當預設值是可變物件時像串列字典和類別物件實體,值會保存,一般值則不會

```
def f(a, L=[]):
    L.append(a)
    return L

print(f(1))
print(f(2))
print(f(3))
```

```
5

In [10]: runf
wdir='/Users/
[1]
[1, 2]
[1, 2, 3]
```

函數呼叫,每次重新分配串列物件

```
9  def f(a, L=None):
10     if L is None:
11         L = []
12     L.append(a)
13     return L
14
15 print(f(1))
16 print(f(2))
17 print(f(3))
18
```

```
Console 1/A

In [12]: run
function_sco
[1]
[2]
[3]

In [13]:
```

關鍵字引數

- 接受一個需要引數voltage和三個選項引數(state,action和type)

```
9  def parrot(voltage, state='僵硬', action='飛', type='挪威藍'):
10     print("-- 這隻鸚鵡將不會", action, end=' ')
11     print("假如你放", voltage, "伏特電壓.")
12     print("-- 可愛的羽毛,", type)
13     print("-- 這是", state, "!")
14
15
16 parrot(1000)                                          # 位置1引數
17 parrot(voltage=1000)                                  # 1個關鍵引數
18 parrot(voltage=1000000, action='VOOOOOM')             # 2個關鍵引數
19 parrot(action='飛飛', voltage=1000000)              # 2個關鍵引數
20 parrot('一百萬', '失去生命', '跳')          # 3個位置引數
21 parrot('一千', state='死去而被葬掉')  # 1個位置引數,1個關鍵引數
```

```
-- 這隻鸚鵡將不會 飛 假如你放 1000 伏特電壓.
-- 可愛的羽毛, 挪威藍
-- 這是 僵硬 !
-- 這隻鸚鵡將不會 飛 假如你放 1000 伏特電壓.
-- 可愛的羽毛, 挪威藍
-- 這是 僵硬 !
-- 這隻鸚鵡將不會 VOOOOOM 假如你放 1000000 伏特電壓.
-- 可愛的羽毛, 挪威藍
-- 這是 僵硬 !
-- 這隻鸚鵡將不會 VOOOOOM 假如你放 1000000 伏特電壓.
-- 可愛的羽毛, 挪威藍
-- 這是 僵硬 !
-- 這隻鸚鵡將不會 跳 假如你放 一百萬 伏特電壓.
-- 可愛的羽毛, 挪威藍
-- 這是 失去生命 !
-- 這隻鸚鵡將不會 飛 假如你放 一千 伏特電壓.
-- 可愛的羽毛, 挪威藍
-- 這是 死去而被葬掉 !
```

不合法的引數輸入

- #parrot() # 需要的引數沒有放入
- #parrot(voltage=5.0, 'dead') # 關鍵字引數後需要有關鍵字引數
- #parrot(110, voltage=220) # 一樣的引數帶入兩次
- #parrot(actor='John Cleese') # 沒有關鍵字引數叫actor

函數參數*name接受實體tuple,
函數參數**name接受字典

- *name需要定義在**name之前
- 字典包含所有關鍵字引數

```
def cheeseshop(kind, *arguments, **keywords):
    print("-- Do you have any", kind, "?")
    print("-- I'm sorry, we're all out of", kind)
    for arg in arguments:
        print(arg)
    print("-" * 40)
    for kw in keywords:
        print(kw, ":", keywords[kw])

cheeseshop("Limburger", "It's very runny, sir.",
           "It's really very, VERY runny, sir.",
           shopkeeper="Michael Palin",
           client="John Cleese",
           sketch="Cheese Shop Sketch")
```

```
-- Do you have any Limburger ?
-- I'm sorry, we're all out of Limburger
It's very runny, sir.
It's really very, VERY runny, sir.
----------------------------------------
shopkeeper : Michael Palin
client : John Cleese
sketch : Cheese Shop Sketch
```

任意數量的引數串列

- *args代表任意數量的引數

```
9 def concat(*args, sep="/"):
10     return sep.join(args)
11
12 print(concat("earth", "mars", "venus"))
13
14
15 print(concat("earth", "mars", "venus", sep="."))
```

- earth/mars/venus
- earth.mars.venus

**d輸入字典當關鍵字引數

```
9  def parrot(voltage, state='僵硬', action='飛'):
10     print("-- 這隻鸚鵡將不會", action, end=' ')
11     print("假如你放", voltage, "伏特電壓.")
12     print("-- 這是", state, "!")
13
14
15 d={"voltage":1000000,"state":'失去生命',"action":'VOOOOOM'}
16 parrot(**d) # 位置1引數
```

```
-- 這隻鸚鵡將不會 VOOOOOM 假如你放 1000000 伏特電壓.
-- 這是 失去生命 !
```

Lambda運算式

- Lambda關鍵字可以建立傳回一個匿名函數
- Lambda函數可以被使用,在函數物件被需要時
- 使用Lambda運算式當回傳匿名函數

語法:　Lambda 參數 : 運算式

```
def make_incrementor(n):
    return lambda x: x + n

f = make_incrementor(42)
print(f(0))
print(f(1))
```

```
[(4, 'fou

In [24]:
LambdaExp
42
43
```

lambda運算式傳送小函數當引數

```
9 pairs = [(1, 'one'), (2, 'two'), (3, 'three'), (4, 'four')]
10
11 pairs.sort(key=lambda pair: pair[1])
12
13 print(pairs)
```

```
[(4, 'four'), (1, 'one'), (3, 'three'), (2, 'two')]
```

文件字串

```
9  def my_function():
10     """文件說明.
11
12     這沒有作任何事.
13     """
14     pass
15
16 print(my_function.__doc__)
```

```
function')
文件說明.

    這沒有作任何事.


In [31]:
```

函數註釋

- 使用者自訂函數的註釋
- 函數註釋定義在函數的__annotations__屬性
- ->代表回傳註釋
- :冒號表示def敘述的結束

```
9 def f(ham: str, eggs: str = 'eggs') -> str:
10     print("Annotations:", f.__annotations__)
11     print("Arguments:", ham, eggs)
12     return ham + ' and ' + eggs
13
14 print(f('spam'))
15
```

```
Annotations: {'ham': <class 'str'>, 'eggs': <class 'str'>, 'return': <class 'str'>}
Arguments: spam eggs
spam and eggs
```

5-7套件模組

- 5-7-1自訂套件模組
- 5-7-2模組搜尋路徑

5-7-1自訂套件模組

- 我們可以在開設的forecastModule目錄中放__init__.py的檔案，這樣Python 就會將這個forecastModule目錄當作一個套件，而這個目錄名稱就是套件名稱，__init__.py可以是空的檔案
- __init__.py放到forecastModule套件中
- MyModule/ForecastWeather.py
- 模組daily放到forecastModule套件中
- MyModule/forecastModule/daily.py
- 模組weekly放到forecastModule套件中
- MyModule/forecastModule/weekly.py

MyModule/ForecastWeather.py

- 從forecastModule套件輸入daily模組和weekly模組
- enumerate()函數會拆解一個串列,並且得到串列的每個組成元素

```
9  from forecastModule import daily,weekly
10 print('每週天氣預測')
11 for number,outlook in enumerate(weekly.forecast(),1):
12     print(number,outlook)
13
14 print("每日天氣預測",daily.forecast())
```

MyModule/forecastModule/daily.py

模組daily包含forecast()函數

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Jan 11 19:38:33 2018

@author: justinwu
"""

def forecast():
    return '和昨天相同'
```

MyModule/forecastModule/weekly.py

- 模組weekly包含forecast()函數

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Jan 11 19:39:22 2018

@author: justinwu
"""

def forecast():
    return ['下雨','下大雨','陰天','晴天','大太陽','颱風天']
```

- 不同模組所執行的內容
- 顯示每週天氣預測和每日天氣預測

每週天氣預測
1 下雨
2 下大雨
3 陰天
4 晴天
5 大太陽
6 颱風天
每日天氣預測 和昨天相同

5-7-2模組搜尋路徑

- Python會到sys標準模組的path變數內的一串目錄名稱串列去搜尋
- 第一行空白行是一個”空字串,代表目前檔案的目錄

模組搜尋路徑

- 這是在Mac電腦內的檔案路徑,使用anaconda3

```
import sys
for Fpath in sys.path:
    print(Fpath)
```

```
/Users/justinwu/anaconda3/lib/python36.zip
/Users/justinwu/anaconda3/lib/python3.6
/Users/justinwu/anaconda3/lib/python3.6/lib-dynload
/Users/justinwu/anaconda3/lib/python3.6/site-packages
/Users/justinwu/anaconda3/lib/python3.6/site-packages/aeosa
/Users/justinwu/anaconda3/lib/python3.6/site-packages/IPython/extensions
/Users/justinwu/.ipython
```

5-8Python內建字串相關函數

- len()函數取得字串長度
- split()函數來使用分隔符號來將字串分解成短字串的串列
- join()函數將字串串列串成一個字串
- +運算子連接字串
- *用來複製字串

len()函數取得字串長度

- 一個中文字長度為1,一個英文字長度也為1,特殊字元長度也為1,例如換行符號
- x='林小寶'
- y=""
- z='Hello World'
- k=' '
- m='\n'
- n=‘\t'
- print(len(x))
- print(len(y))
- print(len(z))
- print(len(k))
- print(len(m))
- print(len(n))

3

0

11

1

1

1

split()函數

- split()函數來使用分隔符號來將字串分解成短字串的串列
- 使用’\n’換行符號作分隔符號,並將醜小鴨分解成一個短字串的串列.

split()函數

- poem='''呱呱呱呱呱
- 醜小鴨呀醜小鴨
- 腿兒短短腳掌大
- 長長脖子扁嘴巴
- 走起路來搖呀搖
- 愛到河邊去玩耍
- 喉嚨雖小聲音大
- 可是只會呱呱呱'''
- print(poem.split('\n'))

['呱呱呱呱呱', '醜小鴨呀醜小鴨', '腿兒短短腳掌大', '長長脖子扁嘴巴', '走起路來搖呀搖', '愛到河邊去玩耍', '喉嚨雖小聲音大', '可是只會呱呱呱']

join()函數

- join()函數將字串串列串成一個字串
- 要將poem串列串接起來,並且使用換行符號’\n’來分隔
- poem=['呱呱呱呱呱', '醜小鴨呀醜小鴨', '腿兒短短腳掌大', '長長脖子扁嘴巴', '走起路來搖呀搖', '愛到河邊去玩耍', '喉嚨雖小聲音大', '可是只會呱呱呱']
- print(‘\n'.join(poem))

- 呱呱呱呱呱
  醜小鴨呀醜小鴨
  腿兒短短腳掌大
  長長脖子扁嘴巴
  走起路來搖呀搖
  愛到河邊去玩耍
  喉嚨雖小聲音大
  可是只會呱呱呱

+運算子連接字串

- x='林小寶'
- y="詹羽彤"
- print(x+'和'+y)
- print(‘林小寶'+"詹羽彤")

  林小寶和詹羽彤

  林小寶詹羽彤

*用來複製字串

- x='林小寶'*5+'\n'
- y="詹羽彤"*2+'\n'
- z='Hello World'
- print(x+y+z)

  林小寶林小寶林小寶林小寶林小寶

  詹羽彤詹羽彤

  Hello World

» Thanks.

Python 精通程式語言

類別

吳佳諺 老師

6.類別

- 6-1類別class與封裝
- 6-2建構函數__new__()
- 6-3封裝

python

6.類別

```
8  class MyClass:
9      #範例屬性參考
10     i=12345
11
12
13 print(MyClass.i)
14
15
16 class Complex:
17     #實體建構
18     def __init__(self,realpart,imagpart):
19         self.r = realpart
20         self.i = imagpart
21
22 x=Complex(3.0,-4.5)
23 print(x.r,x.i)
24
25
```

成員屬性與成員方法

```
8
9 class MyClass2:
10     #範例屬性參考
11     i=12345
12     def f(self):
13         return 'hello world'
14
15 x=MyClass2()
16 print(x.i)
17 print(x.f())
18
19
```

類別和實體變數

- __init__(self,..)為建構函數,實體化物件時會呼叫它
- self為自己這個物件

```
>>> class Dog:
        kind = 'small dog'
        def __init__(self,name):
            self.name = name


>>> d =Dog('small dog')
>>> e = Dog('very small dog')
>>> print(d.kind)
small dog
>>> print(e.kind)
small dog
>>> print(d.name)
small dog
>>> print(e.name)
very small dog
>>>
```

__init__(self)建構物件,
__del__(self)解構物件

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Oct 27 04:52:38 2017

@author: justinwu
"""

class MyClass:
    i=12345

print(MyClass.i)

class Complex:
    def __init__(self,realpart,imagepart):
        self.r=realpart
        self.i=imagepart
    def __del__(self):
        print('delete object')
x=Complex(3.0,-4.5)
print(x.r,x.i)
x=None
```

```
In [4]: runfile('/
wdir='/Users/justi
12345
3.0 -4.5
delete object

In [5]:
```

6-1類別class與封裝

類別是抽象的，而物件就是類別的實體。在這我們將介紹物件類別圖來解釋類別。

物件導向程式設計(object oriented programming簡稱OOP)有幾個特色，分別為類別封裝、繼承inheritance、多型polymorphism。

資料封裝指的是在類別成員(資料欄或方法)宣告存取控制。使用公用public、私有者__，私有者__只有本類別的方法允許使用，公用者public任何類別都允許存取。

衍生類別為本類別的次類別(subclass)或稱為子類別、延伸類別(extended class)，本類別稱為子類別的父類別(super class)或稱為父類別(祖先類別)。

類別語法

class 類別名稱:

<敘述1>

<敘述2>

.....

<敘述N>

類別成員屬性與方法

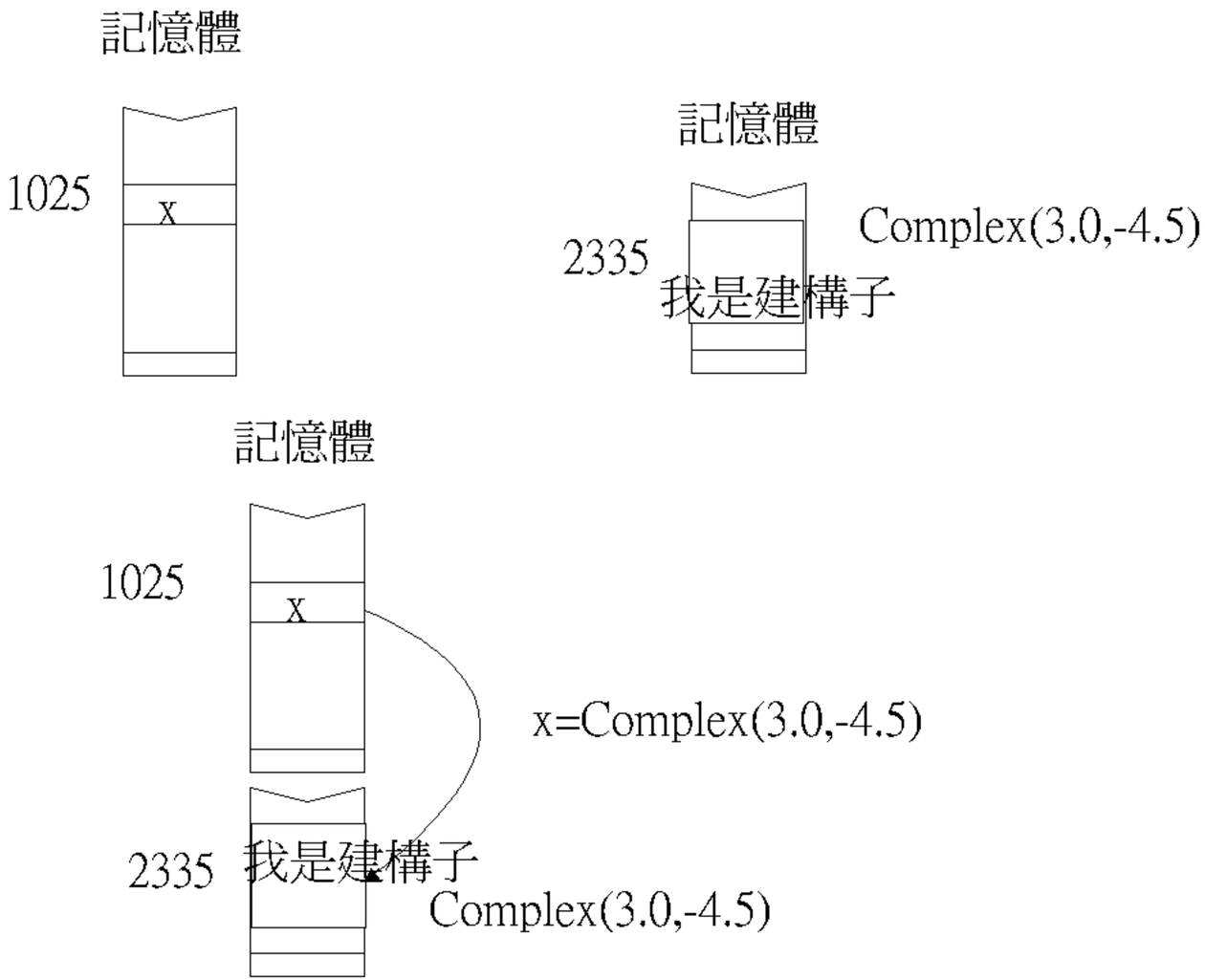
- 類別是由資料和方法所組成。資料宣告就是儲存在每個類別物件裏的資料；方法就是每個物件所提供的服務。

6-2建構函數__new__()

- 當我們建立物件時，我們常要初始化物件的實體變數。初始化實體變數，是最經常看到的初始化。建構子是當我們使用__new__()來新增類別實體時，Python會自動執行該函數，以便初始化物件__init__()。
- x=Complex(3.0,4.0)，Python會先呼叫__new__()來建立Complex(3.0,4.0)物件,然後會呼叫__init__()來初始化Complex物件

- 建立物件第一步是新增變數x。第二步驟是Complex(3.0,4.0)，也就是新增Complex(3.0,4.0)類別的物件，在主記憶體上為它配置空間。第三步驟是將剛剛在記憶體上新增物件的位置給x變數，這樣就完成參考(物件位置的分配)，也就是將物件的位置給Complex類別型態的x變數。

建構子

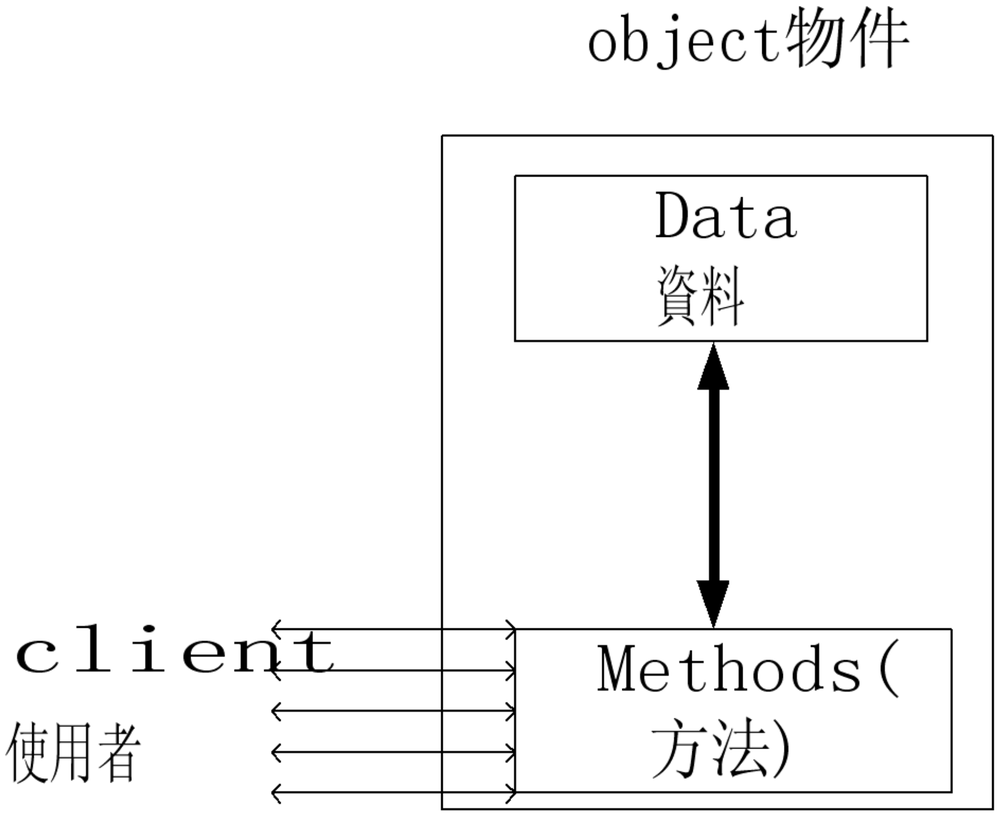

記憶體
1025
x
記憶體
2335
Complex(3.0,-4.5)
我是建構子
記憶體
1025
x
x=Complex(3.0,-4.5)
2335
我是建構子
Complex(3.0,-4.5)

6-3封裝

- 物件需要封裝，來確保資料不會直接被不正當的存取。
- 物件需要封裝於其它的系統。物件應該只開特定的方法來和程式其它部份交談，以確保程式的穩定性。
- 這些方法定義了介面，為物件和使用它們程式間的介面。

- 使用者clinet應該呼叫物件object的方法，而再使用物件的方法來存取物件object內部的資料。
- 這是確保物件資料的安全性。使用者不直接存取object物件資料，而透過方法介面去存取。
- Python並沒有強制資料成員一定要被封裝。

封裝



- **Class Private** 是以二個下底線 __ 開頭，而且不超過一個下底線 _ 結尾的的變數名稱，例如 **__name 或 __tricks**，但是**__init__()** 就不是 class private，但這種兩個下底線開頭和結尾叫特殊方法

Class Private 是以二個下底線 __ 開頭

```python
class Dog:
    def __init__(self,name):
        self.__name=name
        self.__tricks=[] #建立串列

    def add_trick(self,trick):
        self.__tricks.append(trick)
    def printTrick(self):
        print(self.__name+str(self.__tricks))

d = Dog('小黑')
e = Dog('小白')

d.add_trick('翻滾')
e.add_trick('靜止')
d.printTrick()
e.printTrick()
```

add_trick()和printTrick()為Dog類別的成員函數

- 小黑['翻滾']
- 小白['靜止']

python

- Thanks.

Python 精通程式語言

繼承

吳佳諺 老師

7.繼承

7-1物件導向程式語言

7-2父類別和子類別

7-3父類別與子類別的關係

7-4多型、動態聯繫

7-5實作繼承,多型

7.繼承

- class 子類別(父類別):
- 　　敘述1
- 　　敘述2

__為私有存取控制修飾,只有該類別方法才能存取

```python

class Vehicle:
    def __init__(self,name,engine):
        self.__name = name
        self.__engine = engine

    def getName(self):
        return self.__name

    def getEngine(self):
        return self.__engine

    def setEngine(self,engine):
        self.__engine = engine


class Car(Vehicle):
    def __init__(self,name,engine,electric):
        super().__init__(name,engine)
        self.__electric = electric

    def getCarName(self):
        print("名子"+self.getName())
        print("引擎"+self.getEngine())
        print("電動車"+self.__electric)

    def getAuto(self):
        print("自動駕駛車")

myCar = Car("特斯拉","磁電Engine","電力")
myCar.getCarName()
print(myCar.getAuto())
```

Console 2/A

```
Python 3.6.2 |Anacon
Type "copyright", "c

IPython 6.1.0 -- An
名子特斯拉
引擎磁電Engine
電動車電力
自動駕駛車
None

In [2]:

In [2]:
```

多重繼承

- class 子類別(父類別1,父類別2,父類別3,..):
- 　　敘述1
- 　　敘述2
- 當子類別繼承 (inheritance) 超過一個來源的時候，會以寫在最左邊的父類別優先繼承，多個父類別如果有相同名稱的屬性 (attribute) 與方法 (method) ，就會以最左邊的父類別優先。

```python
class Vehicle:
    def __init__(self,name,engine):
        self.__name = name
        self.__engine = engine

    def getName(self):
        return self.__name

    def getEngine(self):
        return self.__engine

    def setEngine(self,engine):
        self.__engine = engine

class Electric:
    def __init_(self,PowerElectric):
        self.__PowerElectric = PowerElectric

    def getPower(self):
        return self.__PowerElectric

    def setPower(self,PowerElectric):
        self.__PowerElectric = PowerElectric
```

Console

```
名子:特斯拉
引擎:磁電Engine
電動車:電力
自動駕駛車

In [13]:
```

```python
class Car(Vehicle,Electric):
    def __init__(self,name,engine,PowerElectric,auto):
        super().__init__(name,engine)
        self.setPower(PowerElectric)
        self.__Auto = auto

    def getCarName(self):
        print("名子:"+self.getName())
        print("引擎:"+self.getEngine())
        print("電動車:"+self.getPower())

    def getAuto(self):
        return self.__Auto


myCar = Car("特斯拉","磁電Engine","電力","自動駕駛車")
myCar.getCarName()
print(myCar.getAuto())
```

多型

子類別和父類別
有同名的
getEngine()名稱

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

class Vehicle:
    def __init__(self,name,engine):
        self.__name=name
        self.__engine=engine

    def getName(self):
        return self.__name
    def getEngine(self):
        return self.__engine

class Car(Vehicle):
    def __init__(self,name,engine,electric):
        super().__init__(name,engine)
        self.__electric=electric

    def getEngine(self):
        return ("超級")

    def getAuto(self):
        print("自動駕駛車")

myCar=Car("特斯拉","磁電Engine","電力")
myCar.getAuto()
print(myCar.getEngine())
```

```
In [17]: r
car_in.py'
自動駕駛車
超級

In [18]:
```

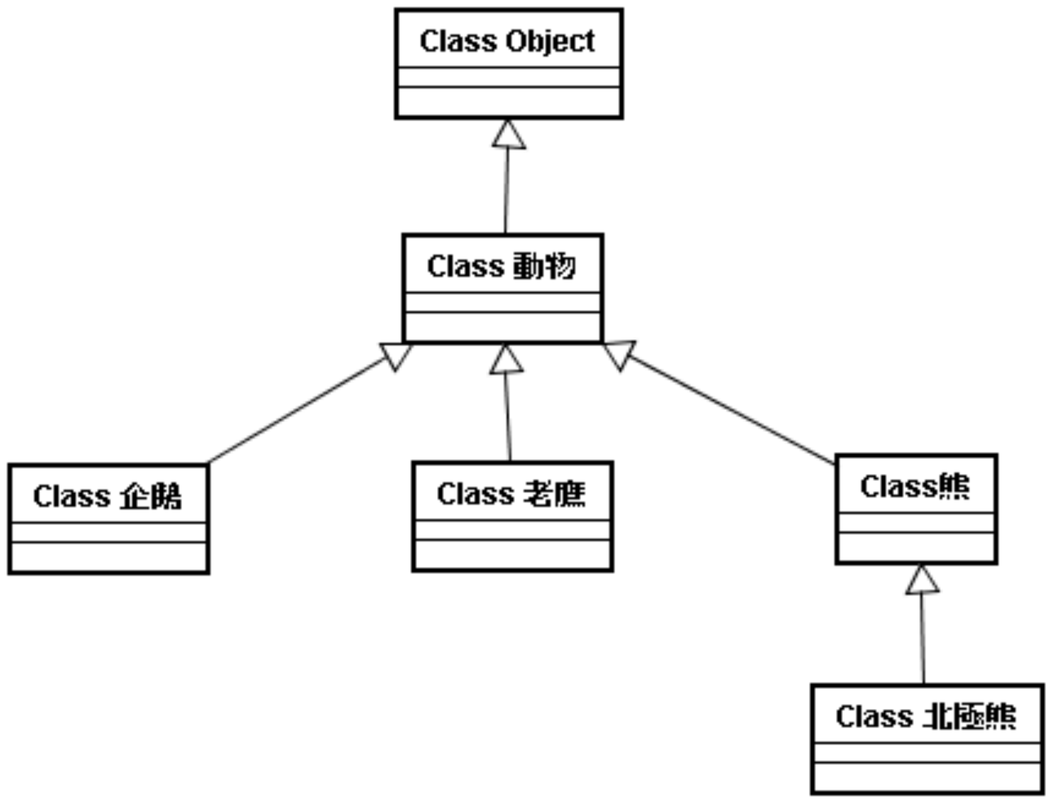
7-1.物件導向程式語言

- 物件導向程式設計中，繼承是一個很重要的特色，它是讓軟體模組可以再使用的一個機制。當建立類別時，我們可以使用繼承機制來使用已存在類別的屬性和方法。這已存在的類別，我們稱為父類別或基礎類別，而使用它的類別稱為子類別或延伸類別。

- 衍生類別為本類別的次類別(subclass)或稱為子類別。延伸類別(extended class)，本類別的延伸類別稱為上層類別(super class)或稱為父類別(祖先類別)。

子類別衍生特性

- Object類別是在最上層。我們寫的動物類別繼承了Object類別的屬性和方法。企鵝、老鷹和熊都是動物，它們都有動物的特徵屬性。動物會走路，動物會吃東西，動物會養小寶寶，而企鵝、老鷹和熊都繼承了動物的這些屬性和方法。
- 因此動物是企鵝、老鷹和熊的父類別。
- 企鵝會在水中游泳，這是企鵝的特徵。北極熊繼承了熊的特徵。熊有大爪子，熊喜歡獨居，而北極熊也繼承了這些特徵。而北極熊生長在北極，北極熊的毛是白色的，這是北極熊特有的特徵。



pass沒做什麼

» 新增Employee()欄位

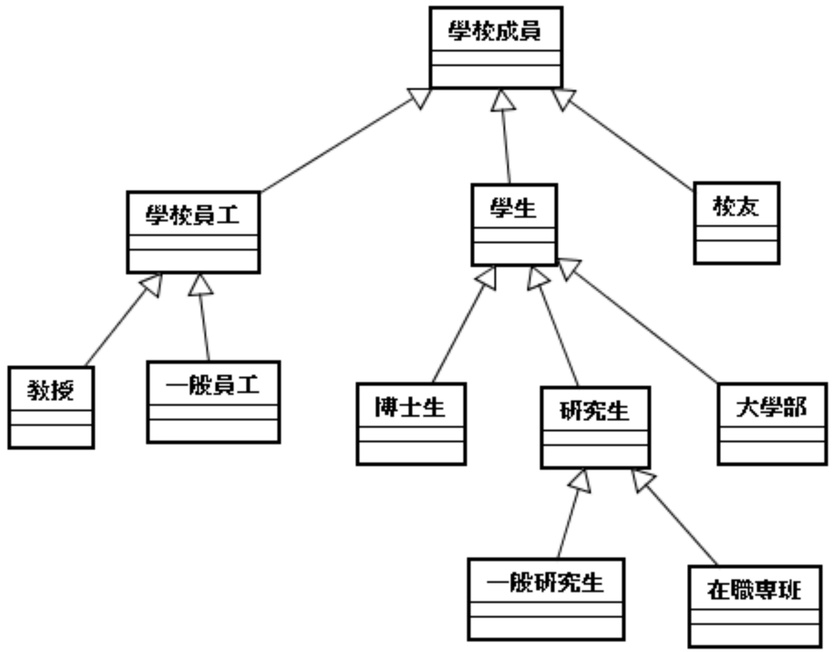
```
1   class Employee:
2       pass
3
4   john=Employee()
5   john.name='John Doe'
6   john.dept='Computer lab'
7   john.salary=1000
8   print(john.name)
9   print(john.dept)
10  print(john.salary)
```

iter

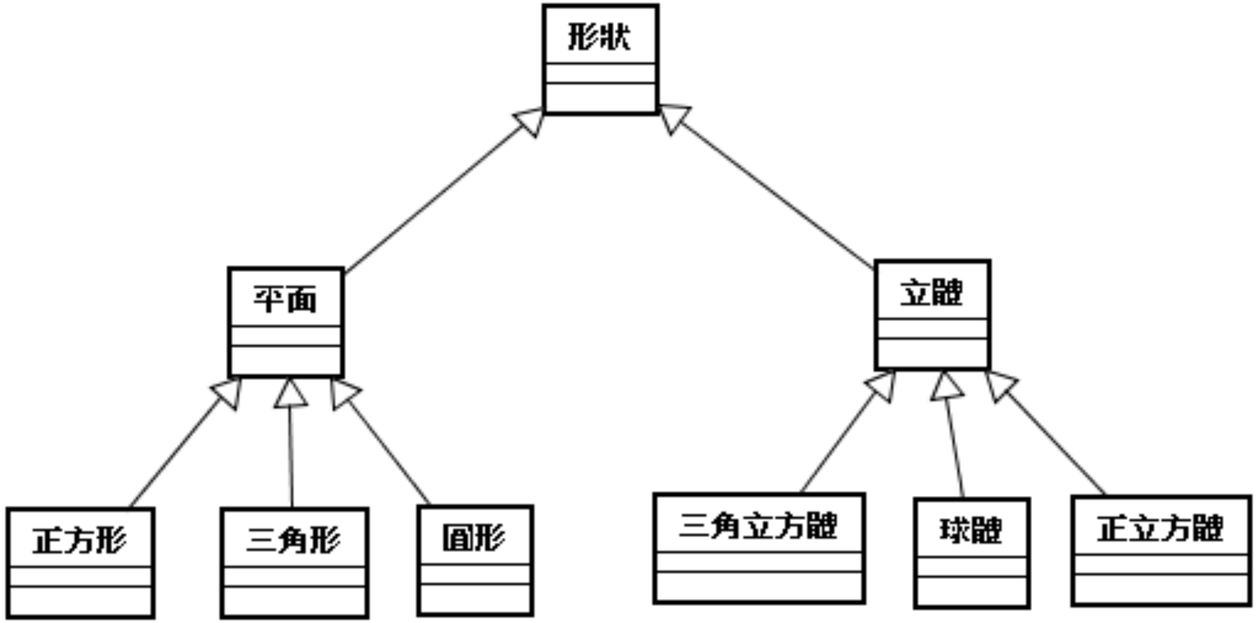
```
/Users/justinwu/untitled4/bin/pyt
John Doe
Computer lab
1000
```

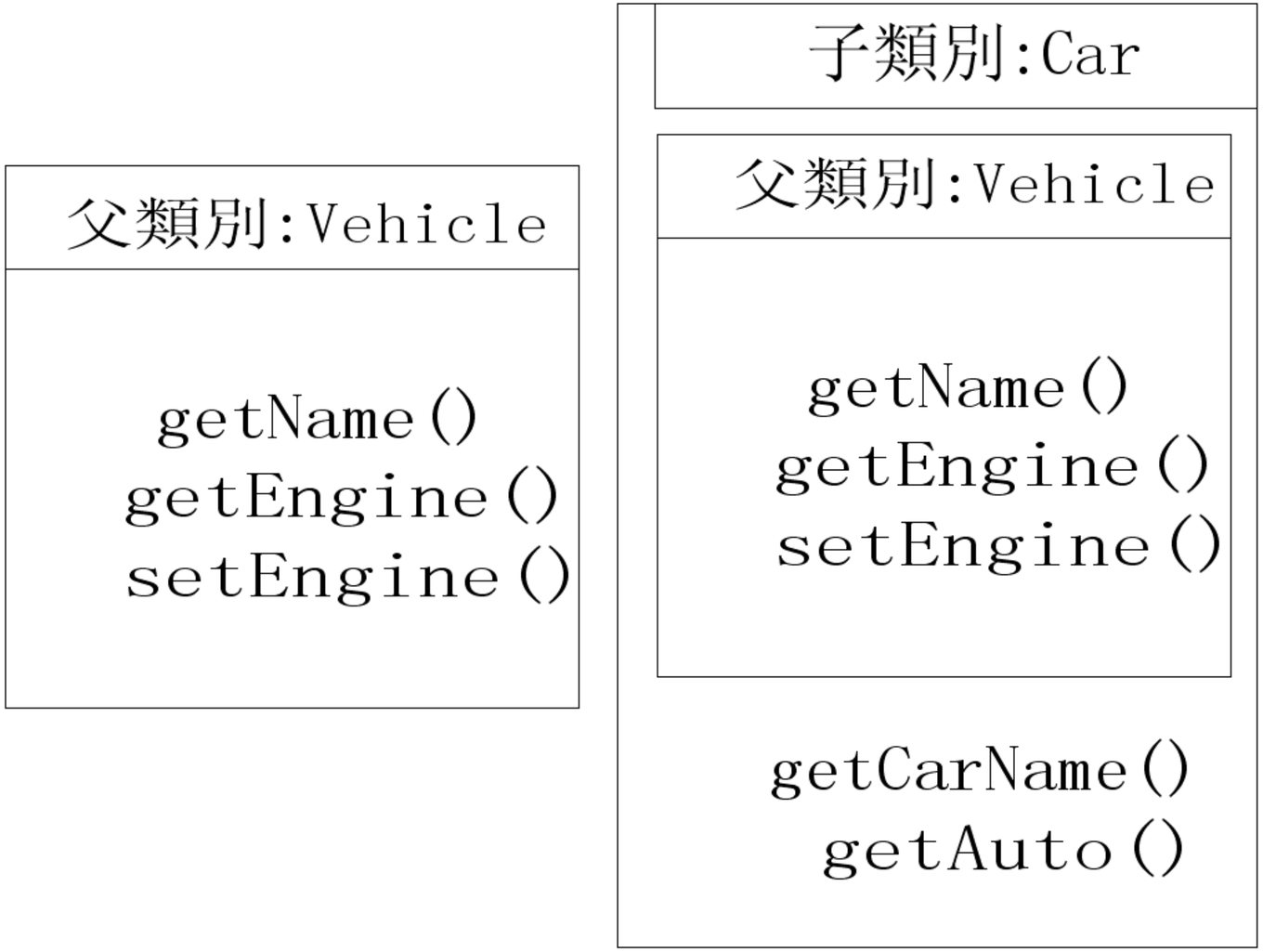
7-2父類別和子類別

- 學生的父類別是學校成員，學生的子類別是博士生、研究生和大學部學生。每一個博士生、研究生和大學部學生都是學校成員。因此每一個子類別的物件都是父類別的物件，而一個父類別可以有很多的子類別，因此代表父類別的物件比代表子類別的物件還多。
- 一個研究生他是學生，也是學校成員。教授是員工，也是學校成員。



- 平面繼承自形狀，而立體也繼承自形狀。三角形、四邊形及圓形則繼承自平面。
- 三角立方體、球體和正立方體則繼承自立體。當一個類別參與繼承關係時，它就會和其他類別產生關係。圓形是平面的一種，也是形狀的一種。正立方體是立體的一種，也是形狀的一種。



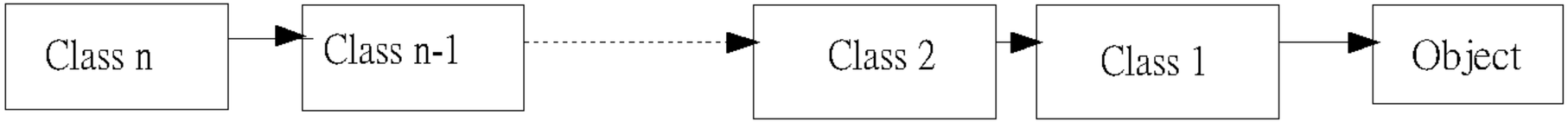

父類別:Vehicle
getName()
getEngine()
setEngine()
子類別:Car
父類別:Vehicle
getName()
getEngine()
setEngine()
getCarName()
getAuto()

7-3父類別與子類別的關係

- 物件導向的優點之一繼承(程式可重複再使用性)。父親和兒子有相同的基因，有相同的血型和，而且思想行為都很相似，這就是繼承。在程式設計中，父親有的兒子就可以繼承，因此程式碼可以不用再寫，這樣就降低了相同的程式重複出現，而降低軟體程式的大小。因此不要重複的使用相同的程式碼，是物件導向繼承的重點。
- 物件導向程式設計另外一個重點是封裝。為了降低模組與模組之間，類別與類別之間的存取複雜度，因此要將資料封裝起來，以免造成資料存取的不一致性。資料的存取，只能在該類別中。要存取另外一個類別的資料，最好透過該類別的函數或者管理類別來存取，來保證資料存取的安全與一致性。

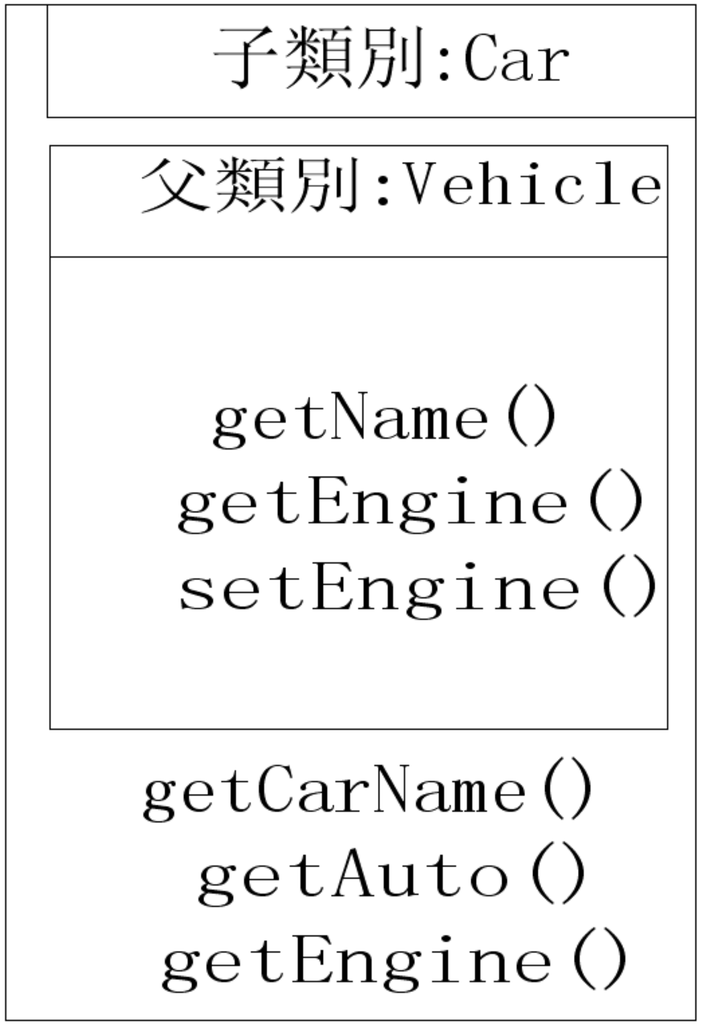
7-4多型、動態聯繫

- 子類別是父類別的特殊化，每一個子類別的實體都含有父類別的實體。例如每一個外聘員工都是員工，但，不是每個員工都是外聘員工。子類別的物件可以用來當作父類別型態的物件。這個就稱為多型(Polymorphism)。



多型

子類別getEngine()和父類別getEngine()為多型,
同名異式



7-5實作繼承

- Shape.py

```
import math
class Point:
    def __init__(self,x=0,y=0):
        self.x=x
        self.y=y

    def __eq__(self,other):
        return self.x==other.x and self.y==other.y

    def __repr__(self):
        return "Point({0.x!r},{0.y!r})".format(self)

    def __str__(self):
        return "({0.x!r},{0.y!r})".format(self)

```

Circle類別改寫Point類別 __repr__()方法,多型

- Shape.py

```
class Circle(Point):
    def __init__(self, radius, x=0, y=0):
        super().__init__(x, y)
        self.radius = radius

    def area(self):
        return math.pi * (self.radius ** 2)

    def __repr__(self):
        return "Circle({0.x!r},{0.y!r})".format(self)

    def __eq__(self, other):
        return "Circle({0.radius!r},{0.x!r},{0.y!r})".format(self)

    def __str__(self):
        return repr(self)
```

use.py輸入Shape.py

- 新增Shape.Point()物件實體和Shape.Circle()物件實體

Shape.py × use.py ×

```
1  import Shape
2  p=Shape.Point(20,30)
3  c=Shape.Circle(5,20,30)
4  print(c.area())
```

- 78.53981633974483

*repr()*函數會對給定的物件呼叫*__repr__()*特殊方法

```python
import Shape
p=Shape.Point(20,30)
c=Shape.Circle(5,20,30)
print(c.area())
print(repr(p))#repr()函數
print(repr(c))#會對給定的物件呼叫__repr__()特殊方法
print(c.radius)
```

use

```
/Users/justinwu/untitled4/bin/python "/Users/justin
78.53981633974483
Point(20,30)
Circle(20,30)
5
```

python

- Thanks.

Python 精通程式語言
資料結構

吳佳諺 老師

8.異常或錯誤處理

8.異常或錯誤處理

9.使用matplotlib畫圖

8.異常或錯誤處理

```
>>> while True:
        try:
                x = int(input("Please enter a number: "))
                break
        except ValueError:
                print("input error")
```

```
Please enter a number: f
input error
Please enter a number: f
input error
Please enter a number: ffffff
input error
Please enter a number:
```

異常或錯誤處理

```python
import sys
try:
    #f = open('myle.txt')
    f = open('mysql.py')
    s = f.readline()
    i = int(s.strip())
except OSError as err:
    print("OS error: {0}".format(err))
except ValueError:
    print("Could not convert data to an integer.")
except:
    print("Unexpected error:", sys.exc_info()[0])
```

Name	Type	Size	
s	str	1	#!/usr/bin/python

Variable explorer | File ex

IPython con

Console 1/A

```
In [5]: runfile('/Users/justinwu/Desktop/exception.py', wdi
OS error: [Errno 2] No such file or directory: 'myle.txt'

In [6]: runfile('/Users/justinwu/Desktop/exception.py', wdi
Could not convert data to an integer.

In [7]:
```

使用raise關鍵字丟出例外

```
8
9  import sys
10
11 def displaySalary(salary):
12     if salary<0:
13         raise ValueError("薪水為正")
14     print("薪水="+str(salary))
15
16 try:
17     #f = open('myle.txt')
18     Salary = eval(input("請輸入薪水:"))
19     displaySalary(Salary)
20 except OSError as err:
21     print("OS error: {0}".format(err))
22 except ValueError:
23     print("錯誤:輸入薪水值為正")
24 except:
25     print("Unexpected error:", sys.exc_info()[0])
```

Console 2/A

```
Unexpected error: <class 'SyntaxError'>

In [14]: runfile('/Users/justinwu/Deskto

請輸入薪水:80000
薪水=80000

In [15]: runfile('/Users/justinwu/Deskto

請輸入薪水:x
Unexpected error: <class 'NameError'>

In [16]: runfile('/Users/justinwu/Deskto

請輸入薪水:-10000
錯誤:

In [17]: runfile('/Users/justinwu/Deskto

請輸入薪水:-10000
錯誤:輸入薪水值為正
```

檔案處理

- fp=open(‘檔案名稱’,’檔案開啟模式’)

模式字串	當開啟檔案已存在	當開啟檔案不存在
r	開啟唯獨的檔案	產生異常錯誤
w	清除檔案內容後寫入	建立寫入檔案
a	開啟檔案從檔尾後開始寫入	建立寫入檔案
r+	開啟讀寫的檔案	產生錯誤
w+	清除檔案內容後讀寫內容	建立讀寫檔案
a+	從檔案尾巴開使讀寫	建立讀寫檔案

開啟,關閉及寫入檔案

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Oct 28 06:39:38 2017

@author: justinwu
"""
fp=open('file.txt','w')
if fp !=None:
    print('檔案開啟成功')
fp.close()

fp=open('file.txt','w')
if fp !=None:
    fp.write("小白")
fp.close()

fp=open('file.txt','w')
if fp !=None:
    fp.write("宇哲")
fp.close()
```

```
In [8]: runfile
Users/justinwu/
檔案開啟成功

In [9]:
```

讀取檔案

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Oct 28 06:46:03 2017

@author: justinwu
"""

fp=open('file.txt','r')
if fp !=None:
    str=fp.read()
    print(str)
fp.close()

fp=open('file.txt','r')
if fp !=None:
    strList=fp.readlines()
    print(strList)
fp.close()
```

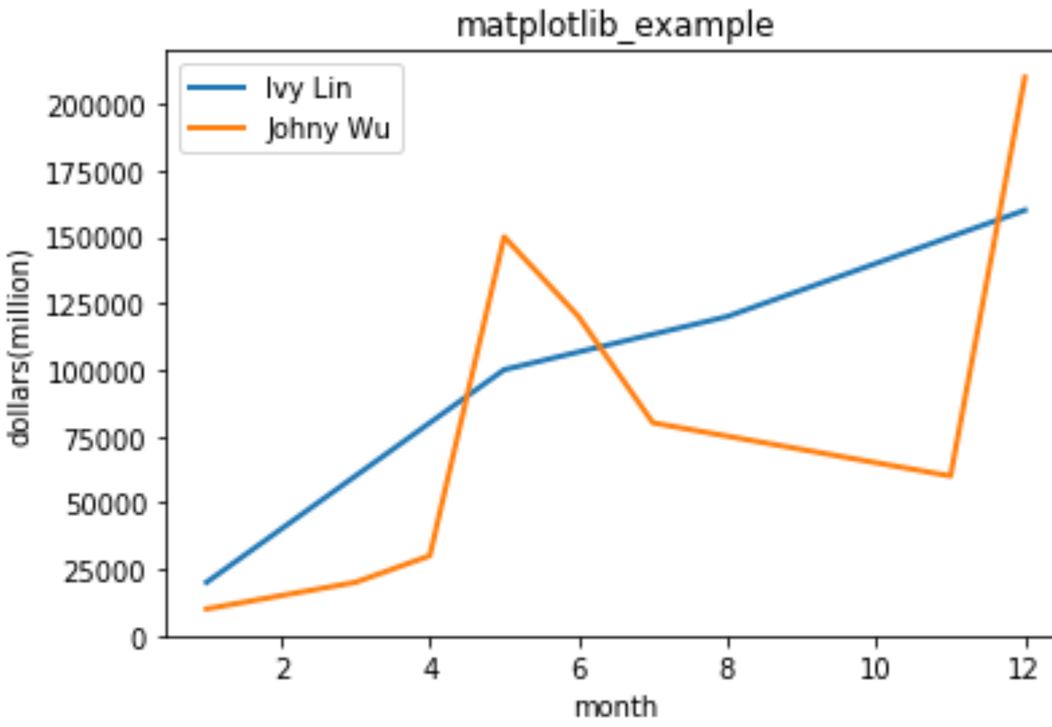
```
In [11]: runfil
Users/justinwu/
宇哲
['宇哲']

In [12]:
```

9.使用matplotlib畫圖

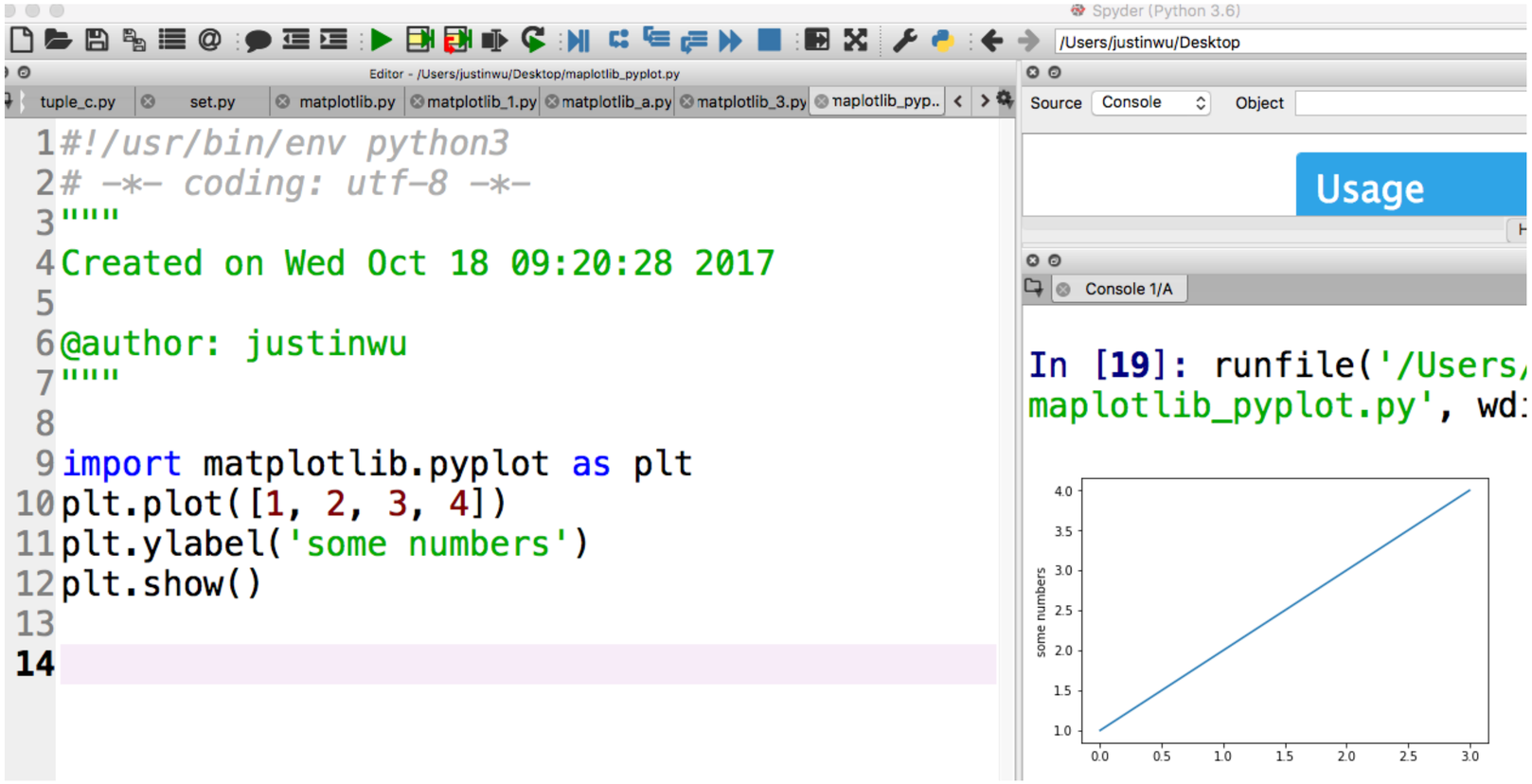
- Matplotlib.pyplot是畫圖的命令集合函數.每一個pyplot函數可以建立或修改圖形

使用matplotlib畫圖

```
6  @author: justinwu
7  """
8  import matplotlib.pyplot as mpt
9
10 month1 = [1,2,3,4,5,8,10,12]
11 month2 = [1,3,4,5,6,7,11,12]
12 sales1 = [20000,40000,60000,80000,100000,120000,140000,160000]
13 sales2 = [10000,20000,30000,150000,120000,80000,60000,210000]
14
15 mpt.plot(month1,sales1,lw=2,label='Ivy Lin')
16 mpt.plot(month2,sales2,lw=2,label='Johny Wu')
17 mpt.xlabel('month')
18 mpt.ylabel('dollars(million)')
19 mpt.legend()
20 mpt.title('matplotlib_example')
21 mpt.show()
```



plt.plot([1,2,3,4])預設是X軸,而Y軸是我們輸入的資料串列[1,2,3,4].



第一個[1,2,3,4]參數是X軸,第二個參數是Y軸

Spyder (Python 3.6)

/Users/justinwu/Desktop

Editor - /Users/justinwu/Desktop/matplotlib_a1.py

tuple.py | tuple_c.py | set.py | matplotlib.py | matplotlib_1.py | matplotlib_a1.py | matplotlib_3.py

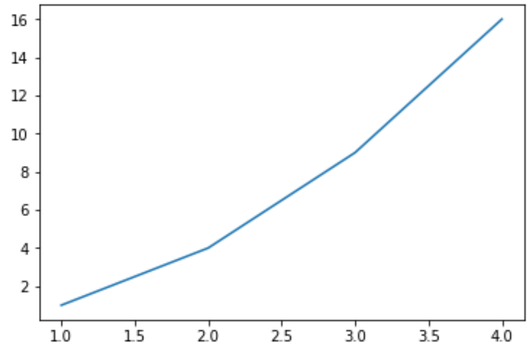
```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Oct 18 08:51:23 2017

@author: justinwu
"""

import matplotlib.pyplot as plt


plt.plot([1, 2, 3, 4], [1, 4, 9, 16])
plt.show()

```

Source | Console | Object

Usage

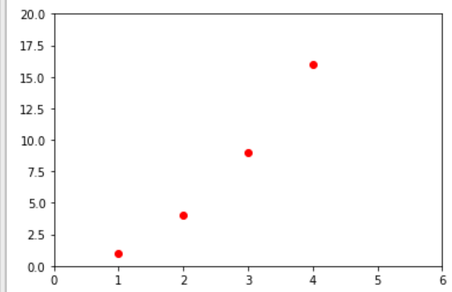
Console 1/A

```
In [24]: runfile('/Use
wdir='/Users/justinwu/
```



plot()第三個參數是格式字串點plot,'ro'為顯示紅色圓圈

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Oct 18 09:34:44 2017

@author: justinwu
"""
import matplotlib.pyplot as plt
#第三個參數是格式字串點plot,'ro'為顯示紅色圓圈
plt.plot([1, 2, 3, 4], [1, 4, 9, 16], 'ro')
plt.axis([0, 6, 0, 20])
plt.show()
```



'r--'紅色虛線,'bs'藍色矩形,'g^'綠色三角形

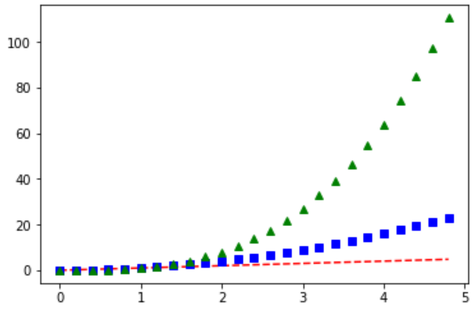
Editor - /Users/justinwu/Desktop/python/example/plot_s.py

plot_ro.py | plot_s.py

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Oct 18 09:42:54 2017

@author: justinwu
"""

import numpy as np

# 0到5每步0.2
t = np.arange(0., 5., 0.2)

# red dashes, blue squares and green triangles
# 'r--'紅色虛線,'bs'藍色矩形,'g^'綠色三角形
plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3,
plt.show()
```

Console 1/A

```
In [33]: runfile('/Users/ju
plot_s.py', wdir='/Users/ju
```



python

- Thanks.

Python 精通程式語言

網站擷取使用Python

吳佳諺 老師

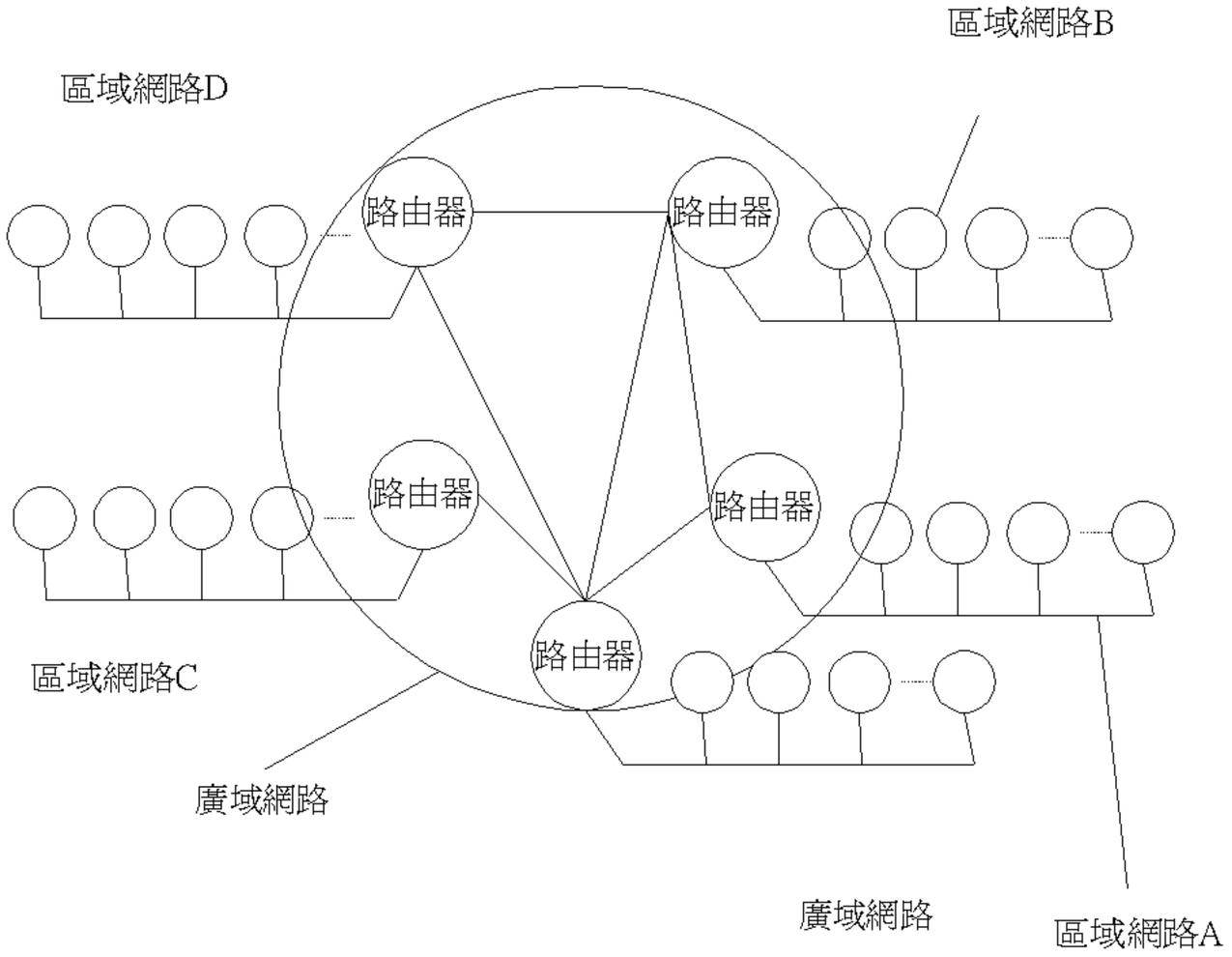
1.下載及安裝Python軟體
2.Python直譯器與計算機
3.資料結構
4.控制結構
5.函數
6.類別
7.繼承
8.異常或錯誤處理
9.使用matplotlib畫圖
10.網站擷取使用Python
11.行程和執行緒
12.資料及檔案處理

網站擷取使用Python

- Python的標準網路函式庫
- http負責管理所有用戶端的工作
  - client負責用戶端的工作
  - Server負責編寫Python web伺服器
- Urlib會在http之上執行
  - requestx負責處理用戶端請求
  - Response負責處理伺服器回應
  - Parse負責解析URL

廣域網路

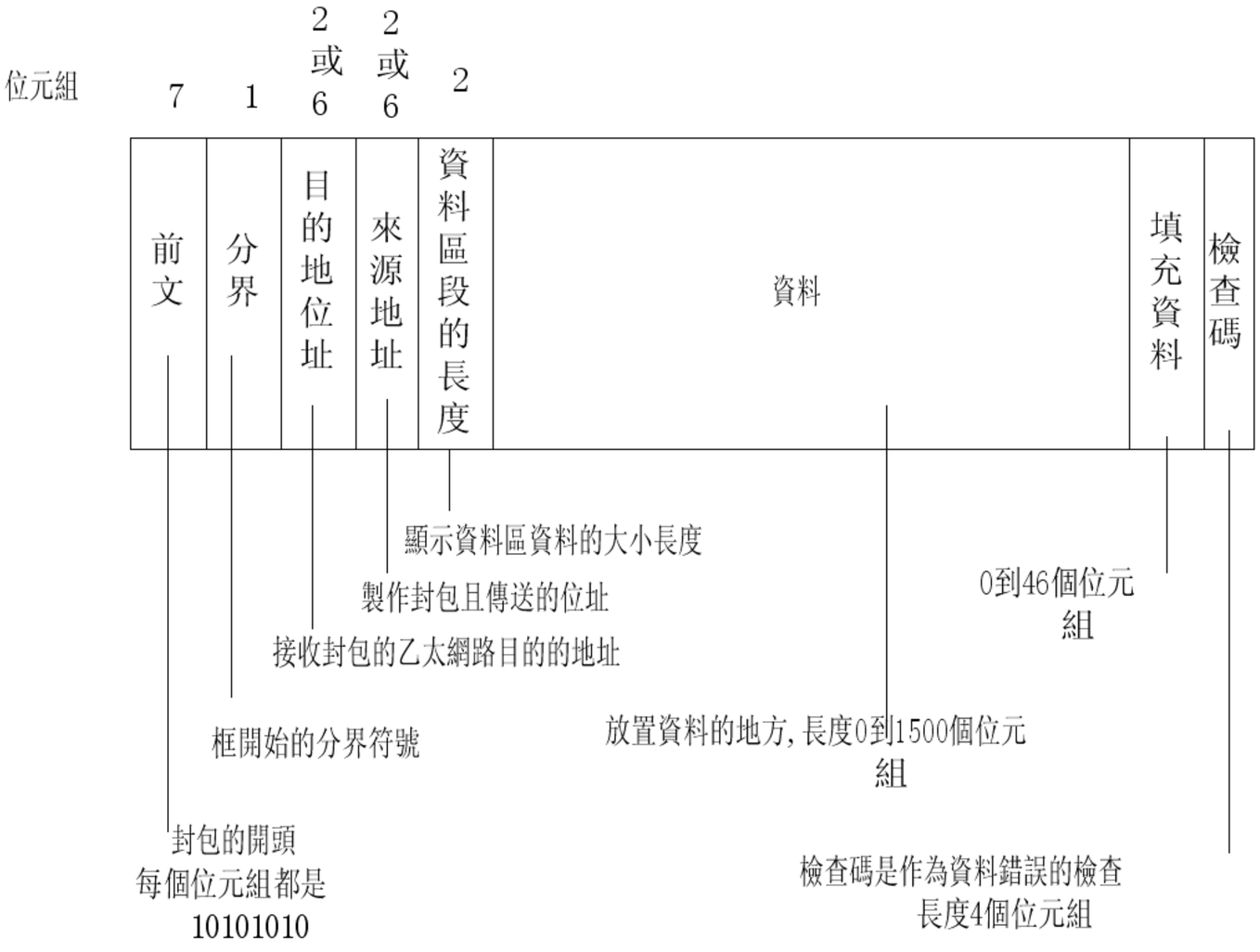
廣域網路稱為Wide Area Networks，它跨越了非常大的地理區域，通常是國家，或大型的數個區域。
Arpanet開始使用，然後發展成世界型的網路。因為廣域網路的每一個節點(網站)在實際的地理位置都非常的分散，因此我們的連接方式有電話線和衛星通訊。
www網路的廣域網路提供我們遠端的主機來存取資料。我們經過區域網路，再經過Router路由器，經過廣域網路，來存取遠端主機的資料，因此而提供世界網路服務。
我們電腦將資料處理成很多連續的封包來傳送，經過網路，再由路由器控制封包傳輸的路徑,將資料封包傳送到指定的地方。


區域網路B
區域網路D
路由器
路由器
路由器
路由器
路由器
區域網路C
廣域網路
廣域網路
區域網路A

封包與連接方法

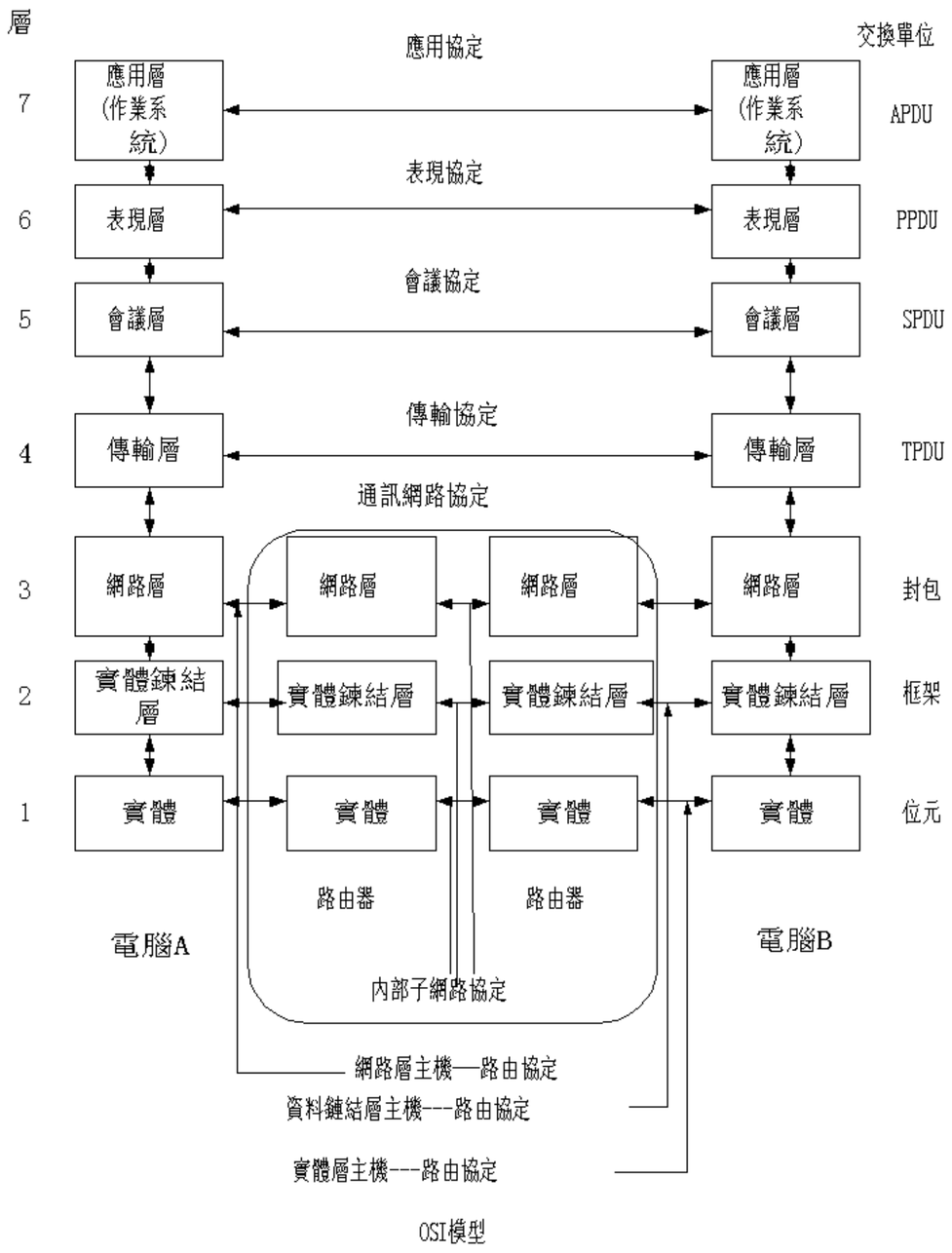
我們將資料分割成許多的封包(Packet)或Frame來傳送。這是PPP(Point To Point)點對點的傳輸框架封包。我們將我們要傳輸的資料放到封包的資料區，最大可放置1500個位元組。位址欄位是設定為11111111，表示所有節點站都可以接收這個封包。控制欄內定值是00000111，這表示為一個無編號框架。點對點的傳輸不使用序號來提供可靠的傳輸。控制欄是判別資料欄內是哪一類型的封包，定義的碼為LCP、NCP、IP、IPX或AppleTalk。檢查碼通常是2個位元組，檢查碼是用來檢查資料傳送時有無遺失或錯誤。

當我們要將網路上的兩個節點連接，我們可以採取線路切換、訊息切換和封包切換。線路切換是當兩個節點要連接時，就會有一條設定的線路連線建立，而這條連線只有這兩個節點可以使用，其它的節點則不可以使用，來保持兩個節點間連線的順暢。訊息切換是當兩個節點要通訊時，將路徑動態的分配給這兩個節點來連接。封包切換是將資料分個成數個封包，內容包含目的地及出發點，然後由路由器轉送到鄰近的節點，最後送到目的地。


位元組
7
1
2或6
2或6
2
前文
分界
目的地位址
來源地址
資料區段的長度
資料
填充資料
檢查碼
顯示資料區資料的大小長度
製作封包且傳送的位址
接收封包的乙太網路目的的地址
0到46個位元組
放置資料的地方,長度0到1500個位元組
框開始的分界符號
封包的開頭
每個位元組都是
10101010
檢查碼是作為資料錯誤的檢查
長度4個位元組

OSI參考模型

OSI總共有七個層次分別是實體層、資料鏈結層、網路層、傳輸層、會議層、表現層和應用層。
而TCP/IP通訊協定則是ARPANET參考模式，它運用於全球網際網路。目前我們經常使用的通訊協定則為TCP/IP協定。
因為TCP/IP把OSI模式給簡化，因此較為簡單與方便使用與了解。
TCP/IP通訊協定的應用層包含了OSI模型的應用層、表現層與會議層。TCP/IP通訊協定的傳輸層則和OSI協定的傳輸層一樣。
TCP/IP通訊協定的網路層則和OSI協定的網路層一樣。而TCP/IP的實體鏈結層則和OSI的實體鏈結層和實體一樣。



電腦A可以經過OSI通訊模式來和電腦B溝通，
而第一層、第二層和第三層為以硬體來製作。

網路名稱

1.TCP/IP:Transport Communication Protocal / Ineternet Protocal網路傳輸協定.

2.IP:網路位址61.218.29.2

4.Domain Name:網路名稱如.com,.tw,.net

5.DNS:Domain Name Server名稱伺服器.將61.218.29.2轉成aasir.com

6.E-Mail:電子郵件chaiyen@cm1.hinet.net

7.URL:網址如aasir.com

8.ISP:網際網路服務提供

9.HomePage:網頁

10.網站擷取使用Python

- 輸入urllib物件
- conn.status為網頁連接狀態

```
import urllib.request as ur
url='http://www.google.com'
conn=ur.urlopen(url)
print(conn)
mydata=conn.read()
print(mydata)
print(conn.status)
print(conn.getheader('Content-Type'))
for key,value in conn.getheaders():
    print(key,value)
```

200表示連接網路成功

```
200
text/html; charset=Big5
Date Tue, 02 Jan 2018 09:35:24 GMT
Expires -1
Cache-Control private, max-age=0
Content-Type text/html; charset=Big5
P3P CP="This is not a P3P policy! See g.co/p3phelp for more info."
Server gws
X-XSS-Protection 1; mode=block
X-Frame-Options SAMEORIGIN
Set-Cookie 1P_JAR=2018-01-02-09; expires=Thu, 01-Feb-2018 09:35:24
GMT; path=/; domain=.google.com.tw
Set-Cookie
NID=120=nMmiizDr3Fq25gRkrJ0z6ZzwFlC3bwdmektWLQ5BocxBbDwAGoJKlLIoc5
87UutuAPI38kxZV3me86cgXM6YnYglSm-
AFqvWdfrVovcbIMikdTAJzaWnCaa8gufdpu6W; expires=Wed, 04-Jul-2018
```

HTML標籤

HTML為我們作網頁的基礎，而在網頁的表達上，是以HTML來表達。HTML為一超文字標籤語言，我們在此介紹較為重要的標籤並且實作。

<HR>	加上分格線
<HTML>	網頁標籤
<I>	斜體字
<IMG SRC>	插入圖片
<INPUT TYPE NAME VALUE>	輸入欄位

標籤 屬性名稱	說明
<!--…-->	註解
<A HREF TARGET>	指定超連結的分割視窗
<A HREF=#錨的名子>	指定錨名稱的超連結
<A HREF>	指定超連結
<B>	粗體字
<BODY>	顯示本文
 	換行
<CAPTION ALIGN>	設定表格標題位置
<CAPTION>	為表格標題位置
<CENTER>	向中對齊
<FONT FACE>	指定字型

標籤 屬性名稱	說明
<FONT SIZE>	指定字體大小
<FORM ACTION>	指定互動表單的處理方法
<FORM METHOD>	互動表單的傳送方法
<FRAME MARGINHEIGHT>	設定視窗的上下邊界
<FRAME MARGINWIDTH>	設定視窗的左右邊界
<FRME NAME>	分割視窗的名稱
<FRME NORESIZE>	鎖定分割視窗大小
<FRAME SCROLLING>	設定分割視窗的捲軸
<FRAME SRC>	將網頁連結分割視窗
<FRAMESET COLS>	將視窗分割成左右的子視窗
<FRAMESET ROWS>	將視窗分割成上下的子視窗
<H1>	設定文字大小
<HEAD>	標示文件資訊

<P>	分段
<TABLE>	表格標籤
<TABLE BORDER=1>	設定表格的寬線高度
<TABLE CELLPADDING>	設定資料欄位的邊界
<TABLE CELLSPACING>	設定表格線的寬度
<TABLE HEIGHT>	表格的高度
<TABLE WIDTH>	表格的寬度
<TD ALIGN>	表格欄位之左右對齊
<TD>	表格欄位的標籤
<TD WIDTH>	表格欄位寬度
<TEXTAREA NAME ROWS COLS>	文字輸入方塊
<TEXTAREA WRAP>	文字是否換行
<TH>	標格的標頭欄位
<TITLE>	文件標題
<TR>	表格的列標籤

<html>標籤範例實作

範例練習:第一個網頁welecome.html

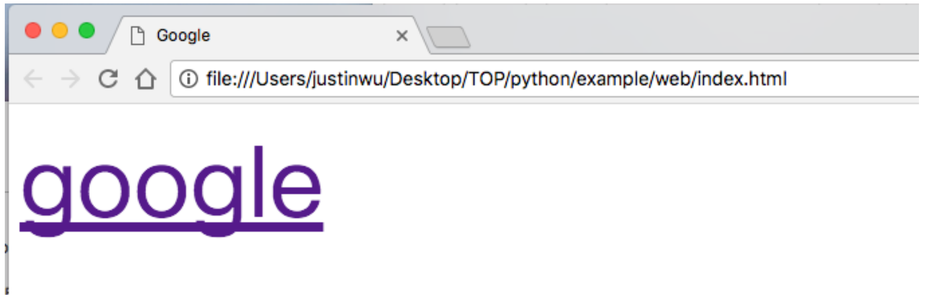
第一行的<html>和最後一行的</html>成對出現。

第2行的<head>和第4行的</head>成對出現。

第五行<body>和第七行</body>成對出現。

<a href=http://www.google.com>google</a>是超連結標籤

```
1 <html>
2 <head>
3 <title>Google</title>
4 </head>
5 <body>
6 <a href="http://www.google.com">google</a>
7 </body>
8 </html>
```



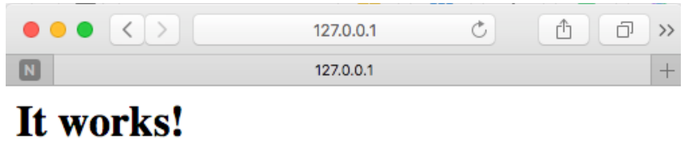
Web伺服器

- 啟動網路http伺服器,0.0.0.0代表任何IP

- $ python -m http.server

```
60-250-191-81:~ justinwu$ python -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/)
```

- 在網頁輸入127.0.0.1的位址,就可以顯示網頁



Bottle

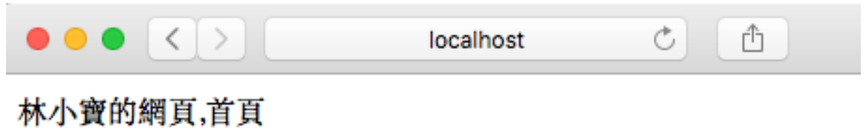
- 安裝bottle
- $pip install bottle
- 使用route去對應URL到程式碼
- port=9999是連接埠
- 主機名稱是localhost

```
9 from bottle import route, run
10
11 @route('/')
12 def home():
13   return "林小寶的網頁,首頁"
14
15 run(host='localhost', port=9999)
```

執行http://localhost:9999/

- 顯示林小寶的網頁,首頁

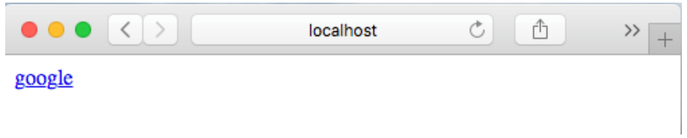
```
IPython 6.1.0 -- An enhanced Interactive Python.

In [1]: runfile('/Users/justinwu/Desktop/TOP/python/example/web/
Users/justinwu/Desktop/TOP/python/example/web')
Bottle v0.12.13 server starting up (using WSGIRefServer())...
Listening on http://localhost:9999/
Hit Ctrl-C to quit.
```



static_file()指定目前的目錄'.'
和檔案同目錄

```
8
9 from bottle import  run, static_file,route
10
11 @route('/')
12 def www():
13     return static_file('index.html', root='.')
14
15 run(host='localhost', port=9999)
```

```
In [7]: runfile('/Users/justinwu/Desk
www.py', wdir='/Users/justinwu/Deskto
Bottle v0.12.13 server starting up (u
Listening on http://localhost:9999/
Hit Ctrl-C to quit.
```



Flask

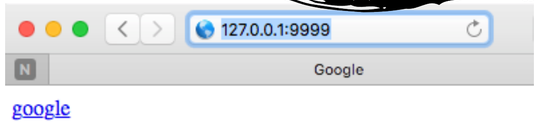
- 安裝Flask
- $ pip install flask

```python
from flask import Flask

app = Flask(__name__, static_folder='.', static_url_path='')

@app.route('/')
def home():
    return app.send_static_file('index.html')

@app.route('/echo/<thing>')
def echo(thing):
    return 'Hello:'+thing

app.run(port=9999, debug=True,use_reloader=False)
```

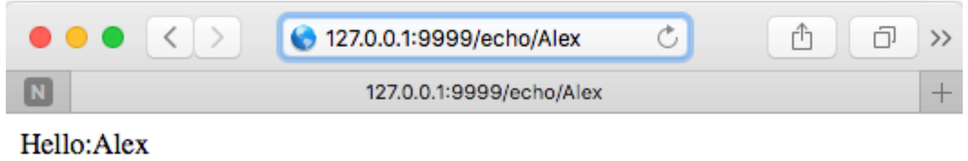
Flask

- 安裝Flask
- $ pip install flask
- Flask的靜態目錄是放置檔案目前目錄’.’
- static_url_path=‘’將url對應至index.html檔案
- app.run()設定debug=True , use_reloader=False
- @app.route(‘/echo/<thing>’)會對應到echo(thing)

- 這是127.0.0.1:9999對應到index.html



- 這是127.0.0.1:9999/echo/Alex會對應到echo 函數,並將Alex帶入到thing參數



- 使用get來傳遞資料

```
127.0.0.1 - - [05/Jan/2018 14:46:05] "GET /echo/Alex HTTP/1.1" 200 -
127.0.0.1 - - [05/Jan/2018 14:46:34] "GET /echo/Alex HTTP/1.1" 200 -
```

Web服務與自動化

- 輸入webbrowser模組,網路瀏覽器控制器
- 打開瀏覽器使用webbrowser.open()

```
Python 3.6.3 Shell
Python 3.6.3 (v3.6.3:2c5fed86e0, Oct  3 2017, 00:32:08)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> WARNING: The version of Tcl/Tk (8.5.9) in use may be unstable.
Visit http://www.python.org/download/mac/tcltk/ for current information.

>>> import webbrowser
>>> url='http://tw.yahoo.com'
>>> webbrowser.open(url)
True
```

- webbrowser.open_new()使用新視窗來打開url

```
>>> import webbrowser
>>> url='http://tw.yahoo.com'
>>> webbrowser.open_new(url)
True
```

BeautifulSoup抓Scrape網頁

- 安裝BeautifulSoup
- $pip install beautifulsoup4

- HTML的a元素代表連結,href是連接屬性,代表它的連接對象

BeautifulSoup抓Scrape網頁

- sudo easy_install pip
- $pip install beautifulsoup4

```python

def get_links(url):
    import requests
    from bs4 import BeautifulSoup as soup
    result = requests.get(url)
    page = result.text
    doc = soup(page)
    links = [element.get('href') for element in doc.find_all('a')]
    return links

if __name__ == '__main__':
    import sys
    url='http://www.yahoo.com'
    print('Links in', url)
    for num, link in enumerate(get_links(url), start=1):
        print(num, link)
    print()

```

```
In [15]: runfile('/Users/justinwu/Desktop/TOP/python/
example/web/beautiful', wdir='/Users/justinwu/Desktop/
TOP/python/example/web')
Links in http://www.yahoo.com
1 /
2 http://tw.mail.yahoo.com
3 http://tw.news.yahoo.com
4 http://tw.stock.yahoo.com
5 https://tw.news.yahoo.com/weather/
6 https://tw.sports.yahoo.com/
7 https://tw.video.yahoo.com/
8 https://tw.celebrity.yahoo.com/
9 https://www.flickr.com/
10 https://tw.buy.yahoo.com/?co_servername=sUH&c2=sUH
11 https://tw.mall.yahoo.com/?_co=pUH&_co2=pUH
12 https://tw.bid.yahoo.com/?
co_servername=aUH&co_servername2=aUH
13
```

抓到a連結網頁的屬性

```python
@author: justinwu
"""
from urllib.error import HTTPError
from bs4 import BeautifulSoup
import sys
from urllib.request import urlopen


def getTitle(url):
    try:
        html = urlopen(url)
    except HTTPError as e:
        print(e)
        return None
    try:
        Obj = BeautifulSoup(html, "html.parser")
        title = Obj.body.a
    except AttributeError as e:
        return None
    return title

title = getTitle("http://www.google.com")
if title == None:
    print("找不到標題")
else:
    print(title)
```

num	int	1	228
url	str	1	http://www.yahoo.com

Variable explorer | File explorer | Help

IPython console

Console 1/A

```
In [33]: runfile('/Users/justinwu/Desktop/TOP/python/example/web/
scraper.py', wdir='/Users/justinwu/Desktop/TOP/python/example/web')
<a class="gb1" href="http://www.google.com.tw/imghp?hl=zh-
TW&amp;tab=wi">圖片</a>

In [34]:
```

python

- Thanks.

Python 精通程式語言

行程和執行緒

吳佳諺 老師

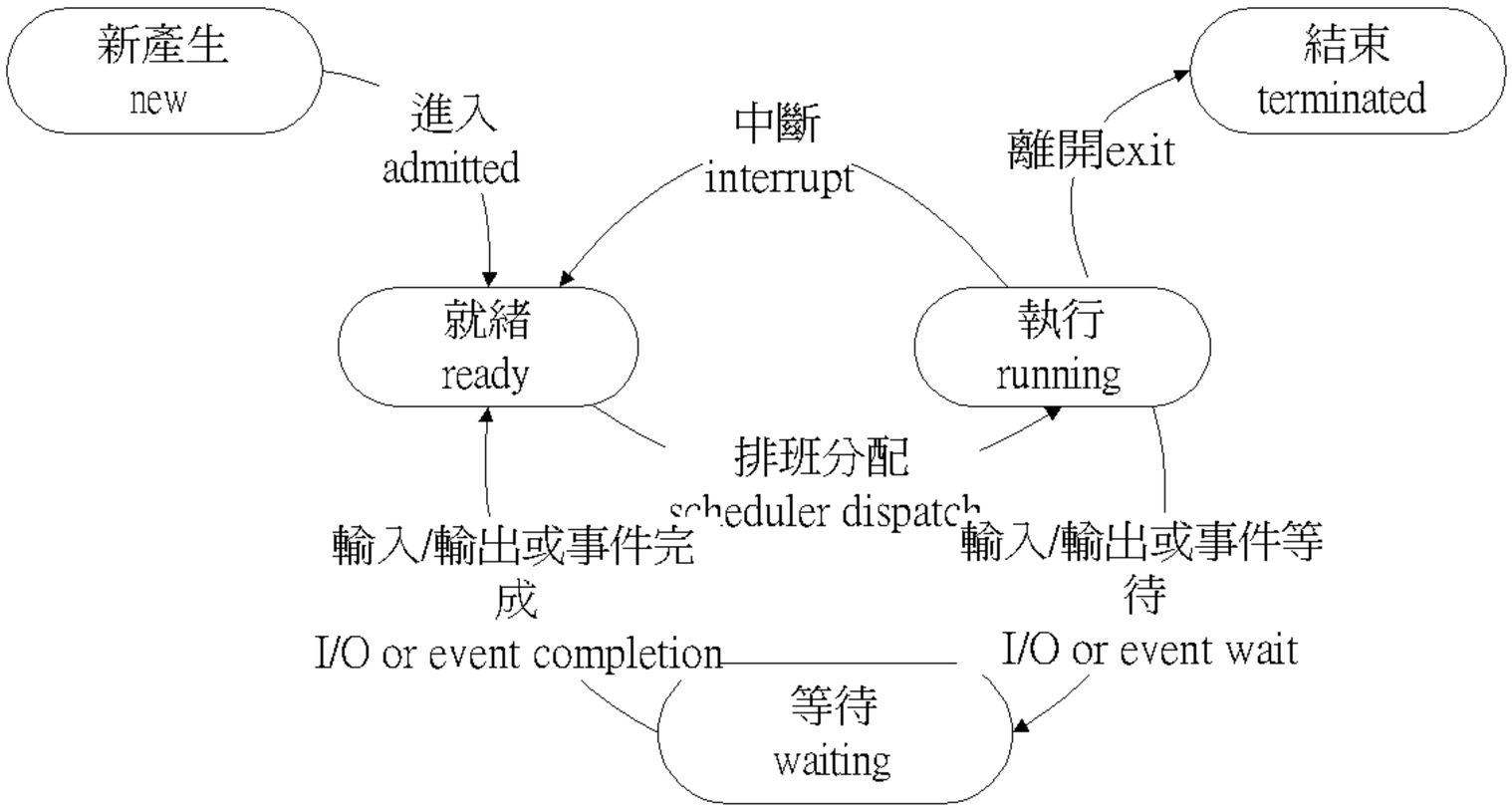
python

1.下載及安裝Python軟體
2.Python直譯器與計算機
3.資料結構
4.控制結構
5.函數
6.類別
7.繼承
8.異常或錯誤處理
9.使用matplotlib畫圖
10.網頁
11.行程和執行緒
12.資料及檔案處理

python

行程和執行緒

一個行程就是一個程式正在執行。
當我們在終端機下達指令時，系統就會建立一個行程，而當我們的程式執行完時，這個行程就被終止了。
在一個分時系作業系統，允許多個使用者使用電腦系統，而許多行程也同步的被執行。
當我們執行程式時，電腦作業系統就會使用new幫我們產生新的行程，而當我們的行程就緒(ready)時，而核心又排班分配CPU中央處理器給他時，他就會開始執行running，直到執行結束terminated(或Zombie)。
當然在過程中也有可能發生像系統呼叫的System call而發生中斷，或者改成其它行程執行(被preempt)，這時我們的行程就會回到就緒ready的狀態。
在過程中也可能發生像輸入/輸出I/O或事件等待(sleep)，此時，CPU是在閒置waiting的狀態，而當我們的輸入/輸出或事件完成時(被wake up)，才會到就緒ready，等待下一次排班分配CPU中央處理器，直到結束(zombie或terminated)。


新產生
new
進入
admitted
中斷
interrupt
離開exit
結束
terminated
就緒
ready
執行
running
排班分配
scheduler dispatch
輸入/輸出或事件完成
I/O or event completion
輸入/輸出或事件等待
I/O or event wait
等待
waiting

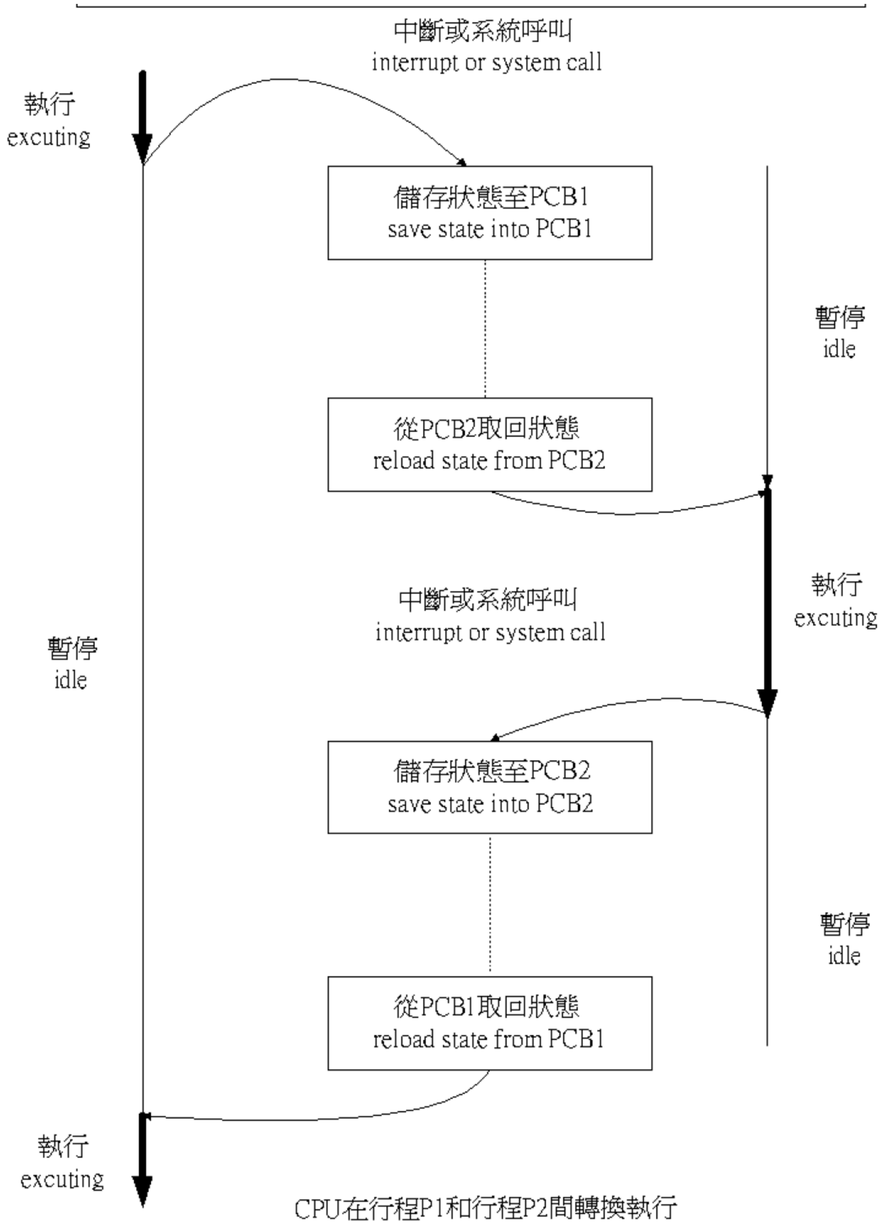
行程狀態圖

這是分時系統當行程P1和行程P2作執行上的內容轉換Content switch。

當行程P1在執行時，因為作業系統的中斷或系統呼叫，這時行程P1就要先將它的Proccess Control Block(行程控制表PCB1)給除存起來。

而這時後CPU中央處理器會載入所呼叫的行程P2，這時會載入行程P2的Proccess Control Block(行程控制表PCB2)，這時CPU中央處理器就會執行行程P2，而當執行行程P2一定的時間，又會將行程P2的Proccess Control Block(行程控制表PCB2)給儲存起來。

而此時CPU中央處理器就會載入行程P1，而執行行程P1。



CPU在行程P1和行程P2間轉換執行

這就是行程狀態控制表。PCB(Proccess Control Block)含有行程的許多內部資訊。

行程狀態：可以是new、ready、running、waiting或halted。

程式計數器：指明該行程下次要執行的指令位置。

CPU暫存器：其數量和類別依不同形式CPU而有不同。有累加器(accumulators)、索引暫存器(index register)、堆疊指標(stack pointer)、位址暫存器(address register)….。

記憶體管理資訊：這包括存取記憶體的基底暫存器(base register)和限制暫存器(limit register)、分頁表(page table)或記憶體系統的分段表(segment table)的資訊。

CPU排班法則和相關資訊：包括行程的優先順序、排班法則。

CPU會計資訊：CPU的使用時間、帳號、工作和行程的編號。

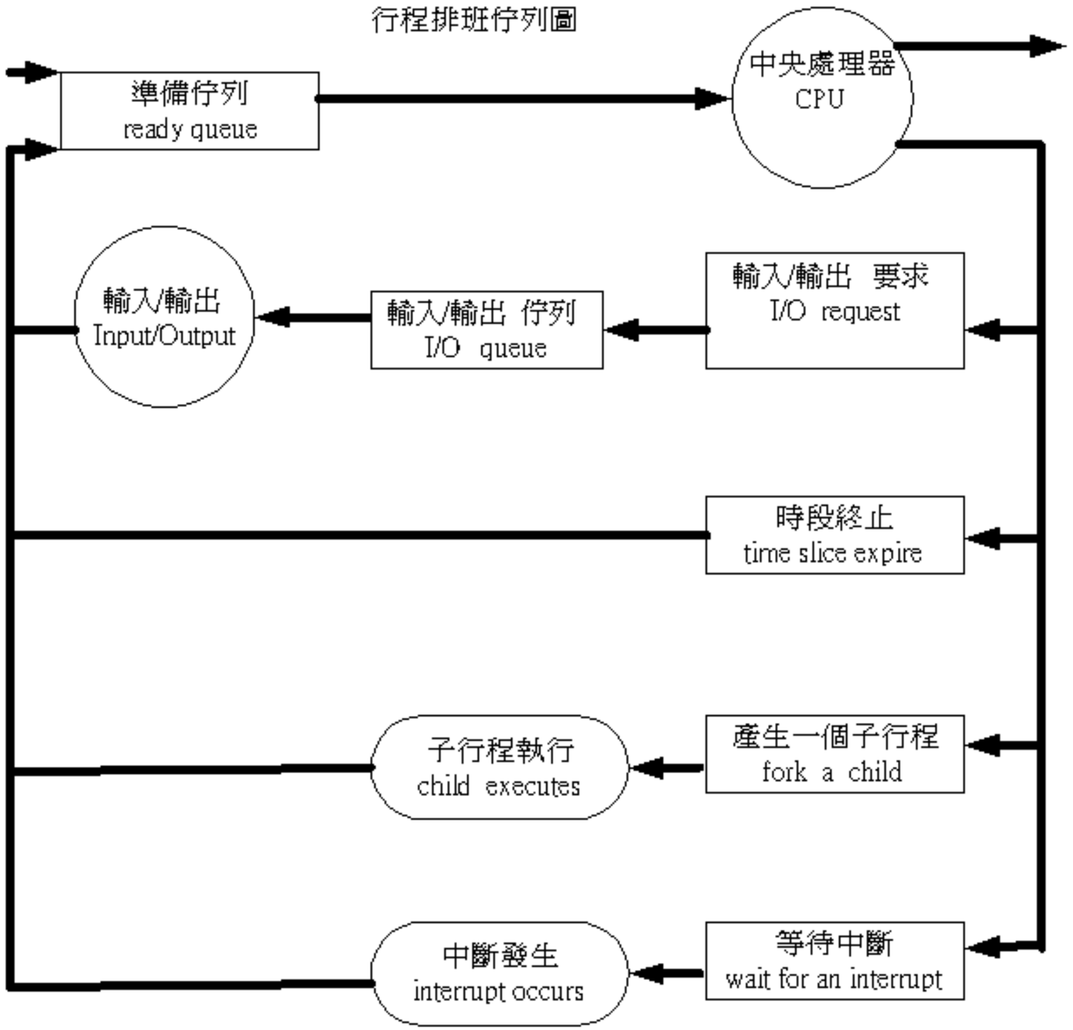
輸入/輸出狀態資訊：這包括了行程的輸入/輸出裝置、開啟檔案的串列。

指標pointer	行程狀態process state
行程號碼process number	
程式計數器program counter	
暫存器registers	
記憶體限制memory limits	
已開啓檔案表list of open files .	
. . . . .	

行程控制表PCB

分時系統的目地是將CPU中央處理器在不同的行程之間不斷的轉換，以便讓使用者可以在自己的行程執行時與它交談。
我們個人電腦的CPU處理器，它們必需在旁佇列中等待，一直到CPU有空時，才可能排到它們執行。
當我們新增new一個新的行程時，它最初是置於就緒佇列中。每一個長方形就是一個佇列，圓圈代表執行行程的資源，箭頭代表執行的方向。當我們所執行的程式或行程在進行時，它會先配置CPU，然後有可能發出輸入/輸出的要求，然後置於一個I/O佇列中。
程式也可能fork產生新的行程。行程也可強行的離開CPU然後就回到就緒佇列中。

行程排班佇列圖



行程排班佇列圖

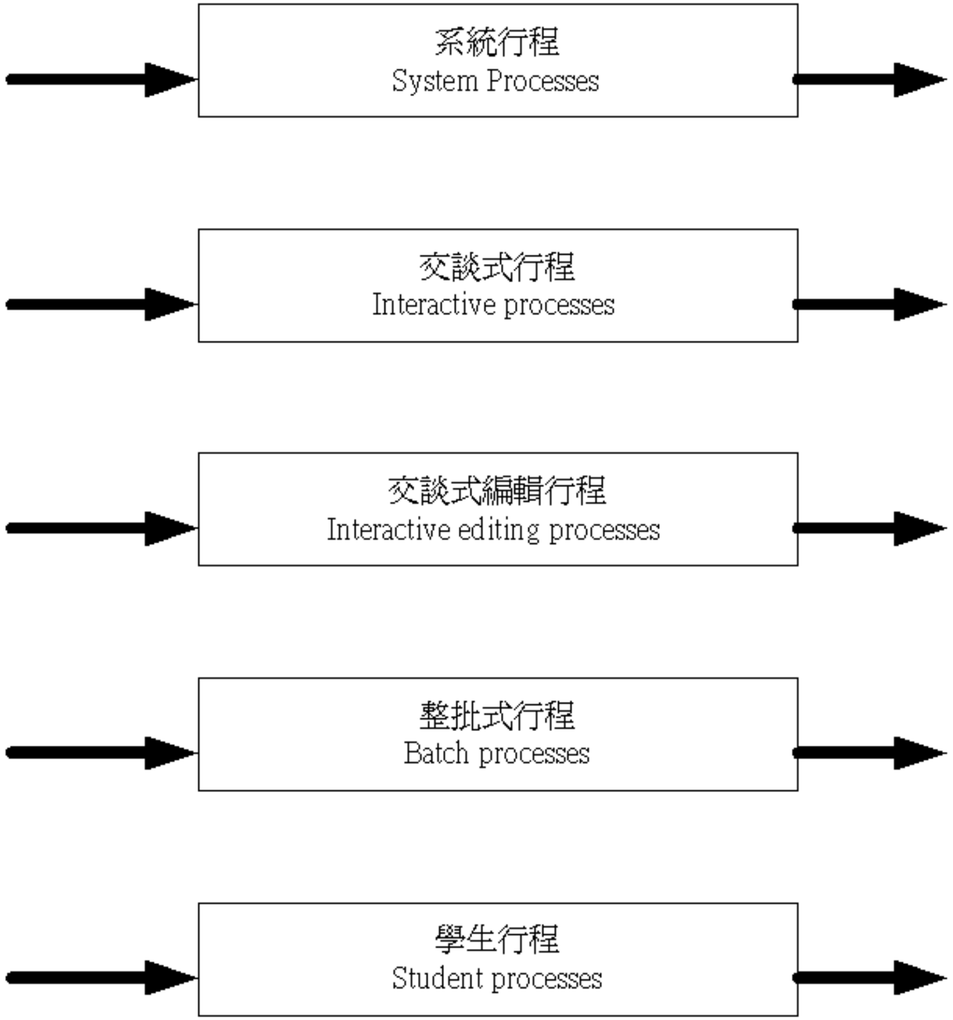
多層佇列之排班法則Multilevel Queue Scheduling，是根據行程的性質將它們分成幾個不同的小組，最典型的分類方式是區分為foreground”前景執行”或background”背景執行”。

前景foreground代表交談式的行程，而背景Background代表的是整批作業的行程。一般來說前景行程的優先權高於背景執行。

多層佇列排班法則將就緒佇列區分為五種獨立的佇列，行程一般都是按照行程得性質決定，例如行程的優先權、記憶體的大小、作業系統設計的目地等。

在我們的例子中，前景的行程使用Round-Robin依序循序排班的排班法則，而背景的行程使用FCFS先來先作排班法則。因此，前景行程的優先權大於背景行程。作業系統使用多層佇列之排班法則。

多層佇列之排班法則
Multilevel Queue Scheduling



行程的屬性

我們可以使用ps來觀看目前行程的狀態。我們加入一些參數，就可以看到我們所要指定的情況。
指令：ps
參數:
-a：顯示所有在終端機執行的行程資訊，包括其它使用者的行程。
-e或 -A：顯示所有在系統執行的行程資訊。
-j：顯示工作的資訊(包含Parent PID、group ID,session ID)。
-l：顯示行程的狀態。
-r：顯示狀態正在執行的行程。
-u：顯示使用者行程的資訊。
-x：顯示行程的所有資訊，而沒有包含TTYs。
-f：顯示行程間的階層與關係。
我們使用ps a顯示所有在終端機執行的行程資訊，包括其它使用者的行程。

- Window 10使用tasklist指令

顯示行程資訊

```
60-250-191-83:~ chaiyenwu$ ps -a
  PID TTY           TIME CMD
  713 ttys000    0:00.02 login -pf chaiyenwu
  714 ttys000    0:00.02 -bash
  724 ttys000    0:00.00 ps -a
```

```
C:\Users\User>tasklist

映像名稱                       PID 工作階段名稱        工作階段 #    RAM使用量
========================= ======== ================ =========== ============
System Idle Process              0 Services                   0          8 K
System                           4 Services                   0      1,308 K
smss.exe                       364 Services                   0      1,156 K
csrss.exe                      500 Services                   0      5,616 K
wininit.exe                    600 Services                   0      6,764 K
services.exe                   672 Services                   0     10,740 K
lsass.exe                      680 Services                   0     21,768 K
svchost.exe                    804 Services                   0      3,692 K
fontdrvhost.exe                832 Services                   0      3,728 K
svchost.exe                    868 Services                   0     35,524 K
svchost.exe                     68 Services                   0     16,796 K
svchost.exe                    548 Services                   0     17,344 K
svchost.exe                   1064 Services                   0      4,364 K
svchost.exe                   1132 Services                   0     19,708 K
svchost.exe                   1464 Services                   0     21,028 K
svchost.exe                   1476 Services                   0     17,820 K
svchost.exe                   1568 Services                   0      9,520 K
nvvsvc.exe                    1576 Services                   0      9,764 K
nvSCPAPISvr.exe               1596 Services                   0      6,504 K
svchost.exe                   1648 Services                   0      7,432 K
svchost.exe                   1712 Services                   0     11,780 K
svchost.exe                   1752 Services                   0     11,848 K
svchost.exe                   1836 Services                   0      9,500 K
```

Windows 10
顯示行程資訊
tasklist

行程平行運算

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Jan  3 16:57:04 2018

@author: chaiyenwu
"""

from multiprocessing import Pool

def f(x):
    return x*x

if __name__ == '__main__':
    p = Pool(5)
    print(p.map(f, [1, 2, 3]))
```

```
In [18]: runfile('
python/example/pro
Users/chaiyenwu/De
process_threads')
[1, 4, 9]

In [19]:
```

Process.start()啟動行程

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Jan  3 17:01:33 2018

@author: chaiyenwu
"""

from multiprocessing import Process

def f(name):
    print('hello', name)

if __name__ == '__main__':
    p = Process(target=f, args=('bob',))
    p.start()
    p.join()
```

```
In [22]: ru
AAA/python/
wdir='/User
example/pro
hello bob

In [23]:
```

os.getpid()得到行程編號

```python
from multiprocessing import Process
import os

def info(title):
    print(title)
    print('module name:', __name__)
    if hasattr(os, 'getppid'):  # only av
        print('parent process:',
              os.getppid())
    print('process id:', os.getpid())
def f(name):
    info('function f')
    print('hello', name)

if __name__ == '__main__':
    info('main line')
    p = Process(target=f, args=('bob',))
    p.start()
    p.join()
```

```
process_id.py', wdir='/Users/chaiyenwu
Desktop/AAA/python/example/process_thr
main line
module name: __main__
parent process: 820
process id: 1353
function f
module name: __main__
parent process: 1353
process id: 2006
hello bob

In [24]:
```

Queue可以用在行程和執行緒儲存資料

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Jan  3 17:08:12 2018

@author: chaiyenwu
"""
from multiprocessing import Process, Queue

def f(q):
    q.put([58, None, 'hello'])

if __name__ == '__main__':
    q = Queue()
    p = Process(target=f, args=(q,))
    p.start()# prints "[58, None, 'hello']"
    print(q.get())
    p.join()
```

```
In [25]: runfile('/User
AAA/python/example/proc
queue.py', wdir='/Users
AAA/python/example/proc
[58, None, 'hello']

In [26]:
```

行程的生產與消費

```python
import multiprocessing as mp

def washer(dishes, output):
    for dish in dishes:
        print('洗', dish)
        output.put(dish)

def dryer(input):
    while True:
        dish = input.get()
        print('擦乾', dish)
        input.task_done()

dish_queue = mp.JoinableQueue()
dryer_proc = mp.Process(target=dryer, args=(dish_queue,))
dryer_proc.daemon = True
dishes = ['大碗', '小碗', '鍋子', '筷子','湯匙']
washer(dishes, dish_queue)
dryer_proc.start()
dish_queue.join()
```

先洗後擦

Usage

Variable explorer File e

IPython co

Console 1/A

```
comsume.py', wdir
chaiyenwu/Desktop
example/process_t
洗 大碗
洗 小碗
洗 鍋子
洗 筷子
洗 湯匙
擦乾 大碗
擦乾 小碗
擦乾 鍋子
擦乾 筷子
擦乾 湯匙
```

行程

```python
import multiprocessing
import os

def do_this(what):
    whoami(what)

def whoami(what):
    print("行程 %s says: %s" % (os.getpid(), what))

if __name__ == "__main__":
    whoami("主程式")
    for n in range(3):
        p = multiprocessing.Process(target=do_this,
          args=("do_this函數 %s" % n,))
        p.start()
```

```
In [4]: runfile('/Users/chaiyenwu/
Desktop/AAA/python/example/
process_threads/processes.py',
wdir='/Users/chaiyenwu/Desktop/AAA/
python/example/process_threads')
行程 992 says: 主程式
行程 1046 says: do_this函數 0
行程 1047 says: do_this函數 1
行程 1048 says: do_this函數 2

In [5]:
```

行程

```python
import multiprocessing
import time
import os

def whoami(name):
    print(" %s, 執行緒 %s" % (name, os.getpid()))
def loopy(name):
    whoami(name)
    start = 1
    stop = 880000
    for num in range(start, stop):
        print("\tNumber %s of %s. 執行!"
                                % (num, stop))
        time.sleep(1)
if __name__ == "__main__":
    whoami("main")
    p = multiprocessing.Process(target=loopy,
                                args=("loopy",))
    p.start()
    time.sleep(5)
    p.terminate()
```

Name	Type	Size	Value
key	str	1	Connection
n	int	1	3

Variable explorer | File explorer | Help

IPython console

Console 1/A

```
In [23]: runfile('/Users/chaiyenwu/
Desktop/AAA/python/example/
process_threads/processes_fun.py',
wdir='/Users/chaiyenwu/Desktop/AAA/
python/example/process_threads')
 main, 執行緒 748
 loopy, 執行緒 12975
        Number 1 of 880000. 執行!
        Number 2 of 880000. 執行!
        Number 3 of 880000. 執行!
        Number 4 of 880000. 執行!
        Number 5 of 880000. 執行!

In [24]:
```

以執行緒Thread來執行

```python
Created on Wed Jan  3 16:17:12 2018

@author: chaiyenwu
"""
import threading

def do_this(what):
    whoami(what)

def whoami(what):
    print("執行緒Thread %s 跑: %s" % (
            threading.current_thread(), what))
if __name__ == "__main__":
    whoami("主程式:")
    for n in range(3):
        p = threading.Thread(target=do_this,
          args=("函數 %s" % n,))
        p.start()
```

Variable explorer | File explorer | Help

IPython console

Console 2/A

```
process_threads/threads.py', wdir='/
Users/chaiyenwu/Desktop/AAA/python/
example/process_threads')
執行緒Thread <_MainThread(MainThread,
started 140736015815488)> 跑: 主程式:執行
緒Thread <Thread(Thread-4, started
123145465651200)> 跑: 函數 0執行緒Thread
<Thread(Thread-5, started
123145470906368)> 跑: 函數 1執行緒Thread
<Thread(Thread-6, started
123145476161536)> 跑: 函數 2

In [2]:
```

執行緒的鎖

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Jan  3 17:16:09 2018

@author: chaiyenwu
"""

import _thread

a_lock = _thread.allocate_lock()

with a_lock:
    print("a_lock is locked while this executes")
```

```
In [25]: runfile('/Users/chaiyenwu/
Desktop/AAA/python/example/
process_threads/queue.py', wdir='/
Users/chaiyenwu/Desktop/AAA/python/
example/process_threads')
[58, None, 'hello']

In [26]: runfile('/Users/chaiyenwu/
Desktop/AAA/python/example/
process_threads/thread_lock.py',
wdir='/Users/chaiyenwu/Desktop/AAA/
python/example/process_threads')
a_lock is locked while this executes

In [27]:
```

生產與消費使用執行緒

```
8  import threading, queue
9  import time
10 def washer(dishes, dish_queue):
11     for dish in dishes:
12         print ("洗"+dish+"\n")
13         time.sleep(1)
14         dish_queue.put(dish)
15 def dryer(dish_queue):
16     while True:
17         dish = dish_queue.get()
18         print ("擦乾"+dish+"\n")
19         time.sleep(2)
20         dish_queue.task_done()
21 dish_queue = queue.Queue()
22 for n in range(2):
23     dryer_thread = threading.Thread(target=dryer,
24                                     args=(dish_queue,))
25     dryer_thread.start()
26 dishes = ['大碗', '小碗', '鍋子', '筷子','湯匙']
27 washer(dishes, dish_queue)
28 print(dish_queue.join())
```

先洗後擦

```
洗大碗

洗小碗
擦乾大碗


洗鍋子
擦乾小碗


洗筷子
擦乾鍋子


擦乾筷子
洗湯匙


擦乾湯匙

None
```

join()等待執行緒呼叫終止terminated

Condition條件鎖

- notify_all()叫醒所有等待條件鎖Condition的執行緒
- 搭配with子句使用

Condition條件鎖

```python
import threading
import time

logging.basicConfig(level=logging.DEBUG,
                    format='%(asctime)s (%(threadName)-2s) %(message)s',
                    )
def producer(cond):
    logging.debug('Starting producer thread')
    with cond:
        logging.debug('Making resource available')
        cond.notifyAll()

def consumer(cond):
    logging.debug('Starting consumer thread')
    t = threading.currentThread()
    with cond:
        cond.wait()
        logging.debug('Resource is available to consumer')

condition = threading.Condition()
cu1 = threading.Thread(name='consumer_1', target=consumer, args=(condition,))
cu2 = threading.Thread(name='consumer_2', target=consumer, args=(condition,))
pro = threading.Thread(name='Producer', target=producer, args=(condition,))

cu1.start()
time.sleep(2)
cu2.start()
time.sleep(2)
pro.start()
```

Usage

Here you can get help of

Variable explorer | File explorer | Help

IPython console

Console 1/A

```
available to consumer
2018-01-04 06:08:42,795 (c2) Resource is
available to consumer

In [2]: runfile('/Users/justinwu/Desktop/
TOP/python/example/process_threads/
with.py', wdir='/Users/justinwu/Desktop/
TOP/python/example/process_threads')
2018-01-04 06:12:15,720 (consumer_1)
Starting consumer thread
2018-01-04 06:12:17,721 (consumer_2)
Starting consumer thread
2018-01-04 06:12:19,727 (Producer)
Starting producer thread
2018-01-04 06:12:19,729 (Producer) Making
resource available
2018-01-04 06:12:19,731 (consumer_1)
Resource is available to consumer
2018-01-04 06:12:19,734 (consumer_2)
Resource is available to consumer

In [3]:
```

開始消費資源

python

- Thanks.

Python 精通程式語言

資料及檔案處理

吳佳諺 老師

python

1.下載及安裝Python軟體
2.Python直譯器與計算機
3.資料結構
4.控制結構
5.函數
6.類別
7.繼承
8.異常或錯誤處理
9.使用matplotlib畫圖
10.網頁
11.行程和執行緒
12.資料及檔案處理

python

檔案處理

- fp=open(‘檔案名稱’,’檔案開啟模式’)

模式字串	當開啟檔案已存在	當開啟檔案不存在
r	開啟唯獨的檔案	產生異常錯誤
w	清除檔案內容後寫入	建立寫入檔案
a	開啟檔案從檔尾後開始寫入	建立寫入檔案
r+	開啟讀寫的檔案	產生錯誤
w+	清除檔案內容後讀寫內容	建立讀寫檔案
a+	從檔案尾巴開使讀寫	建立讀寫檔案

開啟,關閉及寫入檔案

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Oct 28 06:39:38 2017

@author: justinwu
"""
fp=open('file.txt','w')
if fp !=None:
    print('檔案開啟成功')
fp.close()

fp=open('file.txt','w')
if fp !=None:
    fp.write("小白")
fp.close()

fp=open('file.txt','w')
if fp !=None:
    fp.write("宇哲")
fp.close()
```

```
In [8]: runfile
Users/justinwu/
檔案開啟成功

In [9]:
```

讀取檔案

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Oct 28 06:46:03 2017

@author: justinwu
"""

fp=open('file.txt','r')
if fp !=None:
    str=fp.read()
    print(str)
fp.close()

fp=open('file.txt','r')
if fp !=None:
    strList=fp.readlines()
    print(strList)
fp.close()
```

```
In [11]: runfil
Users/justinwu/
宇哲
['宇哲']

In [12]:
```

讀寫資料

- 檔案輸入輸出
- mode第一個字母代表操作
- mode第二個字母是檔案的類型
- t或無代表文字，b代表二進位

開啟寫入文字檔模式wt

write()寫入文字檔

```
7  poem='''呱呱呱呱呱
8  醜小鴨呀醜小鴨
9  腿兒短短腳掌大
10 長長脖子扁嘴巴
11 走起路來搖呀搖
12 愛到河邊去玩耍
13 喉嚨雖小聲音大
14 可是只會呱呱呱'''
15 print(len(poem))
16
17 fout=open('song.txt','wt')
18 fout.write(poem)
19 fout.close()
20
```

Console 1/A

```
In [21]: ru
python/exam
chaiyenwu/D
61

In [22]:
```

開啟寫入二進位檔模式wb

```
8  binaryData=bytes(range(0,128))
9  print(len(binaryData))
10 fout=open('BinaryFile','wb')
11 fout.write(binaryData)
12 fout.close()
13
14 fileIn=open('BinaryFile','rb')
15 bdata=fileIn.read()
16 len(bdata)
17 for i in bdata:
18     print(bdata[i])
```

```
Help can also be shown
Variable explorer   File explorer   Help
IPython console
Console 1/A
AAA/python/example/data/BinaryFile.py',
wdir='/Users/chaiyenwu/Desktop/AAA/python/
example/data')
128
0
1
2
```

用with來關閉檔案

- Python可清除已開啟的檔案環境管理器context managers.
- 語法:　運算式 as 變數

```
 9 binaryData=bytes(range(0,128))
10 print(len(binaryData))
11 fout=open('BinaryFile','wb')
12 fout.write(binaryData)
13 fout.close()
14
15 with open('BinaryFile','rb') as fileIn:
16     bdata=fileIn.read()
17     len(bdata)
18     for i in bdata:
19         print(bdata[i])
```

```
Console 1/A

In [21]: ru
Desktop/AAA
write.py',
Desktop/AAA
61

In [22]:
```

文字檔案結構JSON

- JavaScript Object Notation(JSON)是網路的資料傳輸交換物件.
- JSON為文字檔案結構,常用在網路網頁傳輸
- 使用json.dumps(Employee)將資料結構Employee轉成JSON字串
- 使用json.loads(company_employee)來將JSON物件company_employee轉成Python資料結構字典company2
- company2.keys()顯示資料字典的鍵

文字檔案結構JSON

```python
import json

Employee={
"employee_1":{
"姓名":   "林小寶",
"年齡":   25,
"地址": "台北市復興南路199號2F",
"電話":
{
"公司":   "0922828835",
"住家":   "02-23567890"
}},
"employee_2":{
"姓名":   "江美麗",
"年齡":   35,
"地址": "台北市敦化南路19號2F",
"電話":
{
"公司":   "02-25879099",
"住家":   "02-56890989"
}}}
```

Employee	dict	2	{'empl…
company2	dict	2	{'empl…

Variable explorer | File explorer | Help

IPython console

Console 1/A

```
{
  "employee_1": {
    "姓名": "林小寶",
    "年齡": 25,
    "地址": "台北市復興南路199號2F",
    "電話": {
      "公司": "0922828835",
      "住家": "02-23567890"
    }
  },
  "employee_2": {
    "姓名": "江美麗",
    "年齡": 35,
    "地址": "台北市敦化南路19號2F",
    "電話": {
      "公司": "02-25879099",
      "住家": "02-56890989"
    }
```

ensure_ascii=False,indent=2設定dump為中文字

```
14 "地址": "台北市復興南路199號2F",
15 "電話":
16 {
17 "公司":   "0922828835",
18 "住家":   "02-23567890"
19 }},
20 "employee_2":{
21 "姓名":   "江美麗",
22 "年齡":   35,
23 "地址": "台北市敦化南路19號2F",
24 "電話":
25 {
26 "公司":   "02-25879099",
27 "住家":   "02-56890989"
28 }}}
29 #,ensure_ascii=False,indent=2設定dump為中文字
30 company_employee=json.dumps(Employee,
31                 ensure_ascii=False,indent=2)
32 print(company_employee)
33 company2=json.loads(company_employee)
34 print(company2)
35 print(company2.keys())
36
```

Employee	dict	2	{'empl…
company2	dict	2	{'empl…
company_employee	str	1	{

Variable explorer | File explorer | Help

IPython console

Console 1/A

```
      "公司": "02-25879099",
      "住家": "02-56890989"
    }
  }
}
{'employee_1': {'姓名': '林小寶',
'年齡': 25, '地址': '台北市復興南路199
號2F', '電話': {'公司':
'0922828835', '住家':
'02-23567890'}}, 'employee_2':
{'姓名': '江美麗', '年齡': 35, '地
址': '台北市敦化南路19號2F', '電話':
{'公司': '02-25879099', '住家':
'02-56890989'}}}
dict_keys(['employee_1',
'employee_2'])

In [31]:
```

python

- Thanks.

Python 精通程式語言

SQLite資料庫

吳佳諺老師

SQLite資料庫

- 1.關聯式資料庫與SQLite
  - 在Window 10安裝SQLite
  - 在Mac安裝SQLite3
  - 使用Python連接資料庫
- 2.SQL語言
- 3.SQLite3資料庫模組

1.關聯式資料庫與SQLite

- 關聯式資料庫
- SQLite資料庫引擎
- SQL語言的基礎

關聯式資料庫與SQLite

- 資料庫是一種資料儲存單位，經過組織的資料集合，出勤管理系統、倉庫管理系統、進銷存系統。

- 資料庫系統是由資料庫和資料庫管理系統所組成，資料庫管理系統是一套管理資料庫的應用程式。

關聯式資料庫

- 關聯式資料庫系統是目前資料庫系統的主流，資料庫管理系統都屬於關聯式資料庫管理系統，例如：Access、MySQL、SQL Server和Oracle等。
- 關聯式資料庫是由一個或多個資料表所組成，在多個資料表間使用欄位的資料值來建立連接，以便實作資料表之間的關聯性。

在Window 10
安裝SQLite3

- 從www.sqlite.org下載SQLite
- SQLite3.exe，放入Windows目錄下。

進入命令列

1)建立資料庫檔:

>SQLite3 d:\myreview.sqlite

就生成了一個myreview.sqlite在d槽。

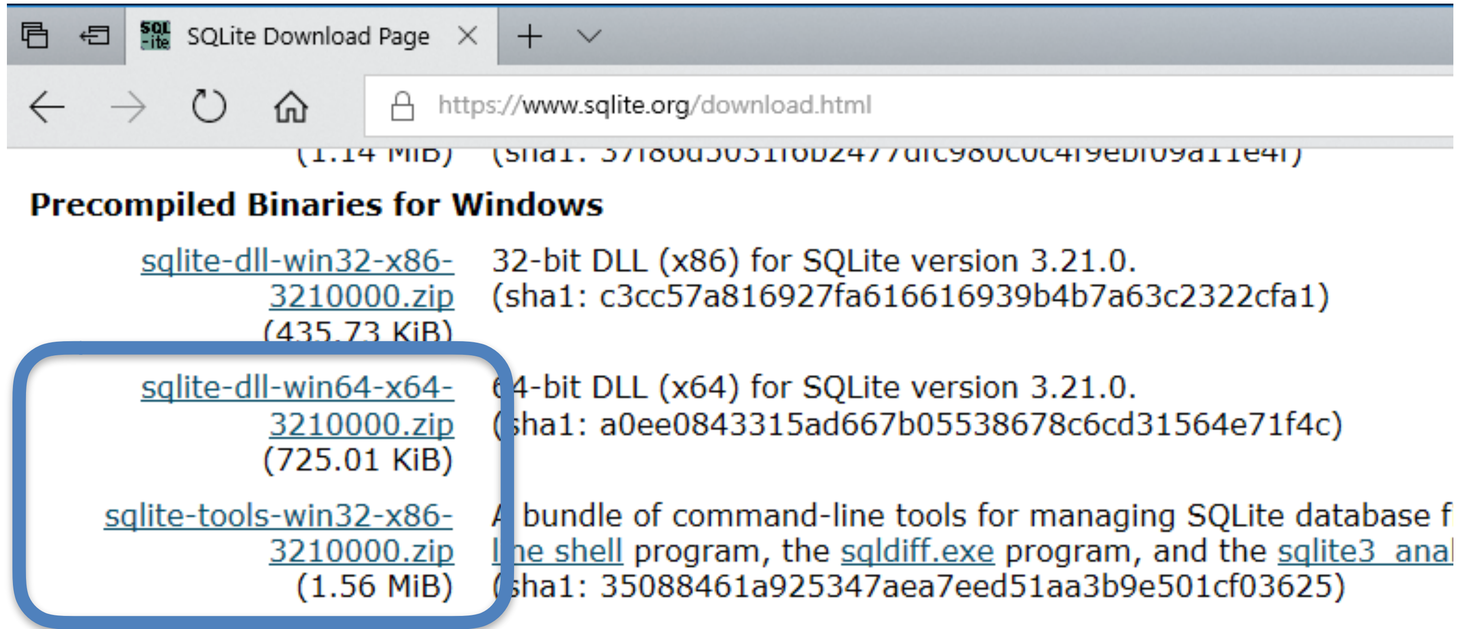
這樣同時SQLite3連接myreview.sqlite資料庫

2).help可以看看有什麼命令

>.help

下載SQLite

- 下載sqlite-dll-win64-x64-3210000.zip檔和sqlite-tools-win32-x86-3210000.zip檔並且解壓縮到c:\sqlite目錄下
- sqlite3.def, sqlite3.dll 和sqlite3.exe三個檔案,放到同一個C:\sqlite目錄下

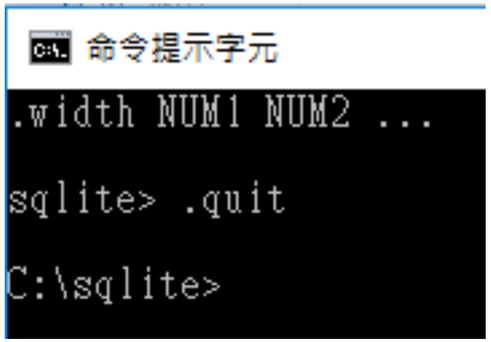


- 執行sqlite3.exe就可以進入到sqlite資料庫

命令提示字元 - sqlite3

```
C:\>cd sqlite

C:\sqlite>dir
 磁碟區 C 中的磁碟是 系統保留
 磁碟區序號:  EA6E-790E

 C:\sqlite 的目錄

2018/01/07  下午 04:10    <DIR>          .
2018/01/07  下午 04:10    <DIR>          ..
2018/01/07  下午 04:09           454,144 sqldiff.exe
2018/01/07  下午 04:05           742,414 sqlite-dll-win64-x64-3210
2018/01/07  下午 04:09    <DIR>          sqlite-tools-win32-x86-32
2018/01/07  下午 04:08         1,634,357 sqlite-tools-win32-x86-32
2018/01/07  下午 04:10             5,435 sqlite3.def
2018/01/07  下午 04:10         1,742,848 sqlite3.dll
2018/01/07  下午 04:09           740,352 sqlite3.exe
2018/01/07  下午 04:09         1,965,056 sqlite3_analyzer.exe
               7 個檔案       7,284,606 位元組
               3 個目錄  157,421,453,312 位元組可用

C:\sqlite>sqlite3
SQLite version 3.21.0 2017-10-24 18:55:49
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite>
```

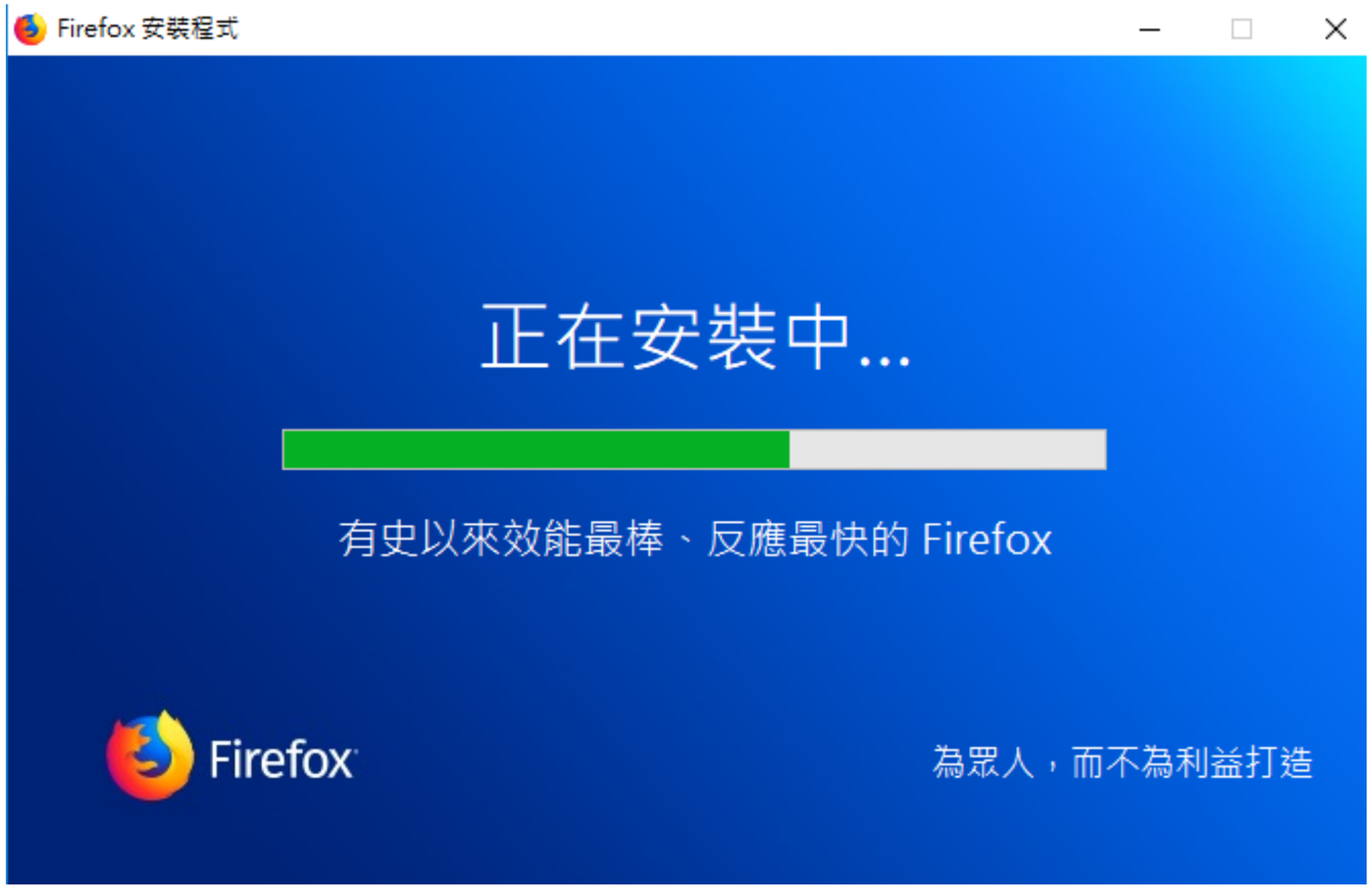
- 使用.quit就可以離開sqlite資料庫



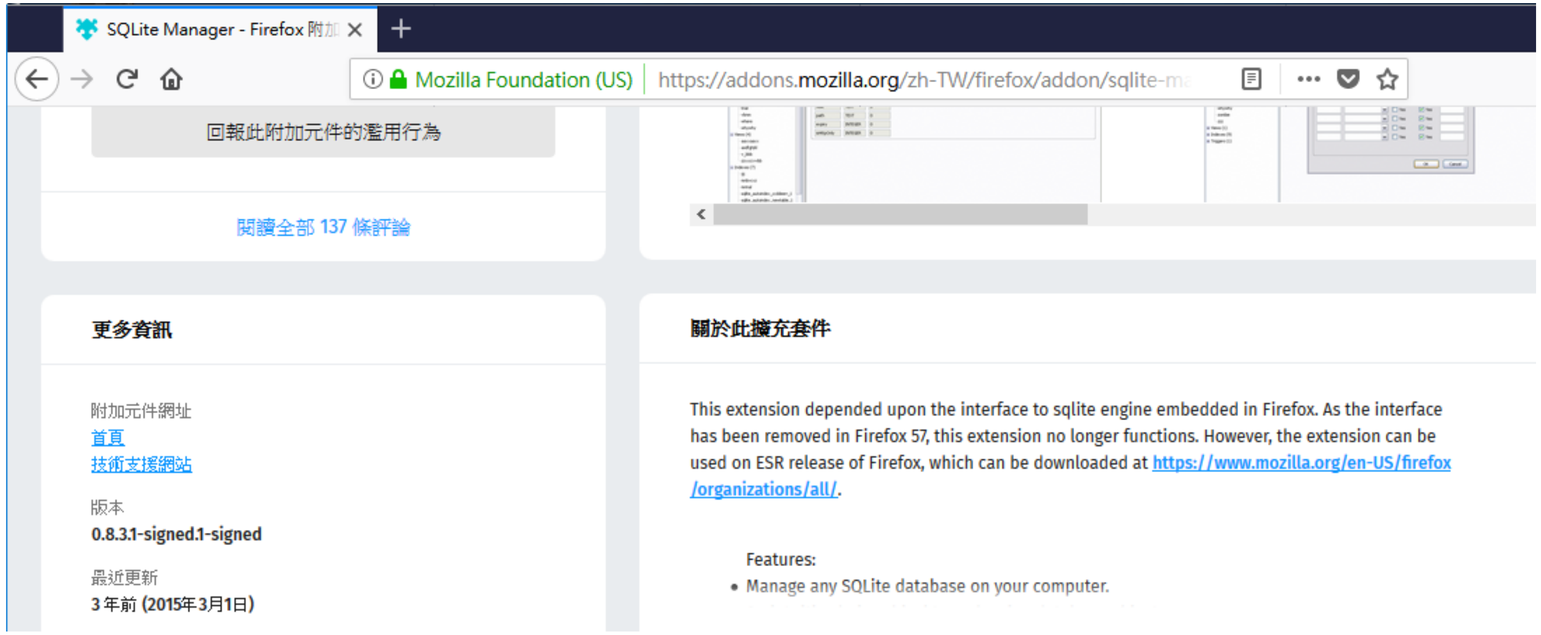
安裝Firefox

- 先安裝Firefox並且再安裝sqlite Manager

這是安裝Firefox中

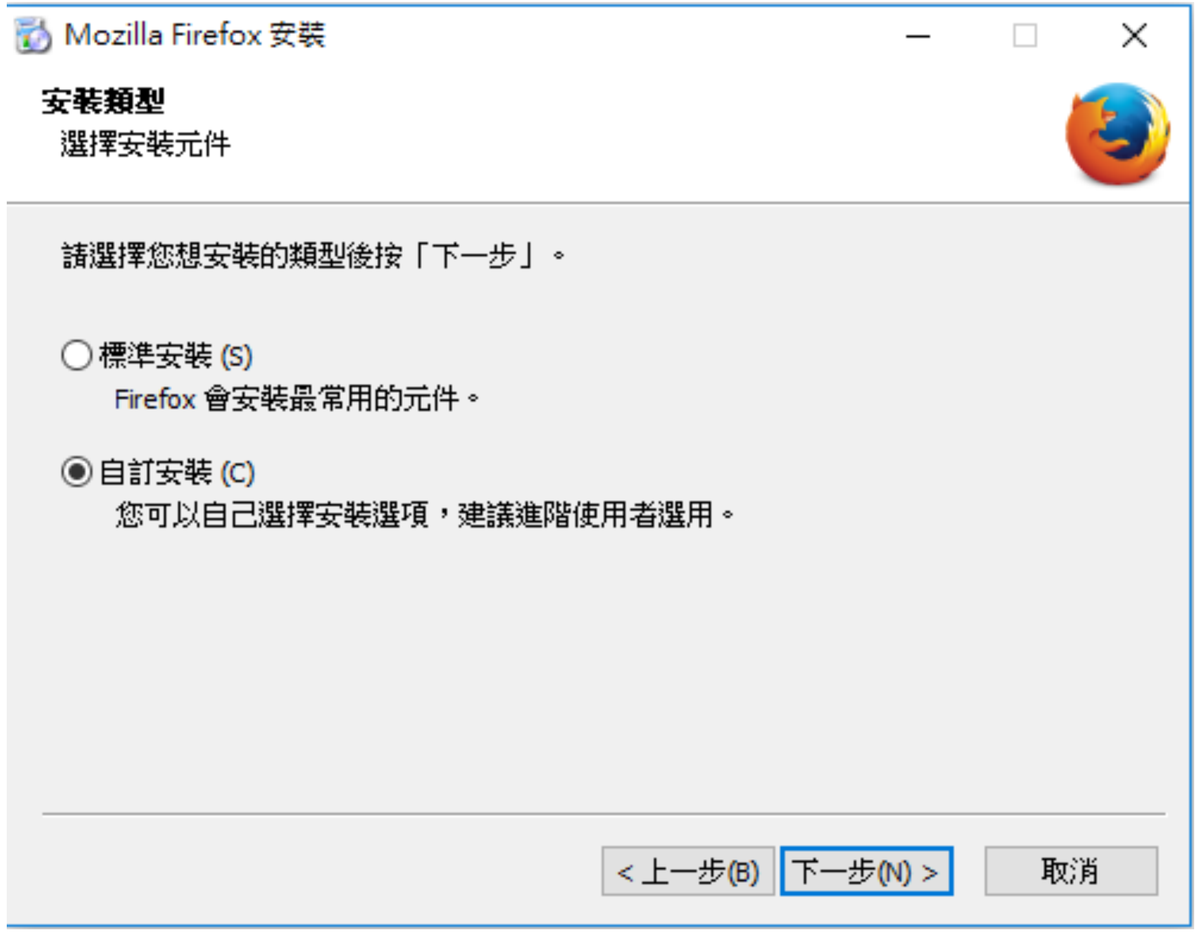


- 選取擴充套件安裝繁體中文和SQLite Manager

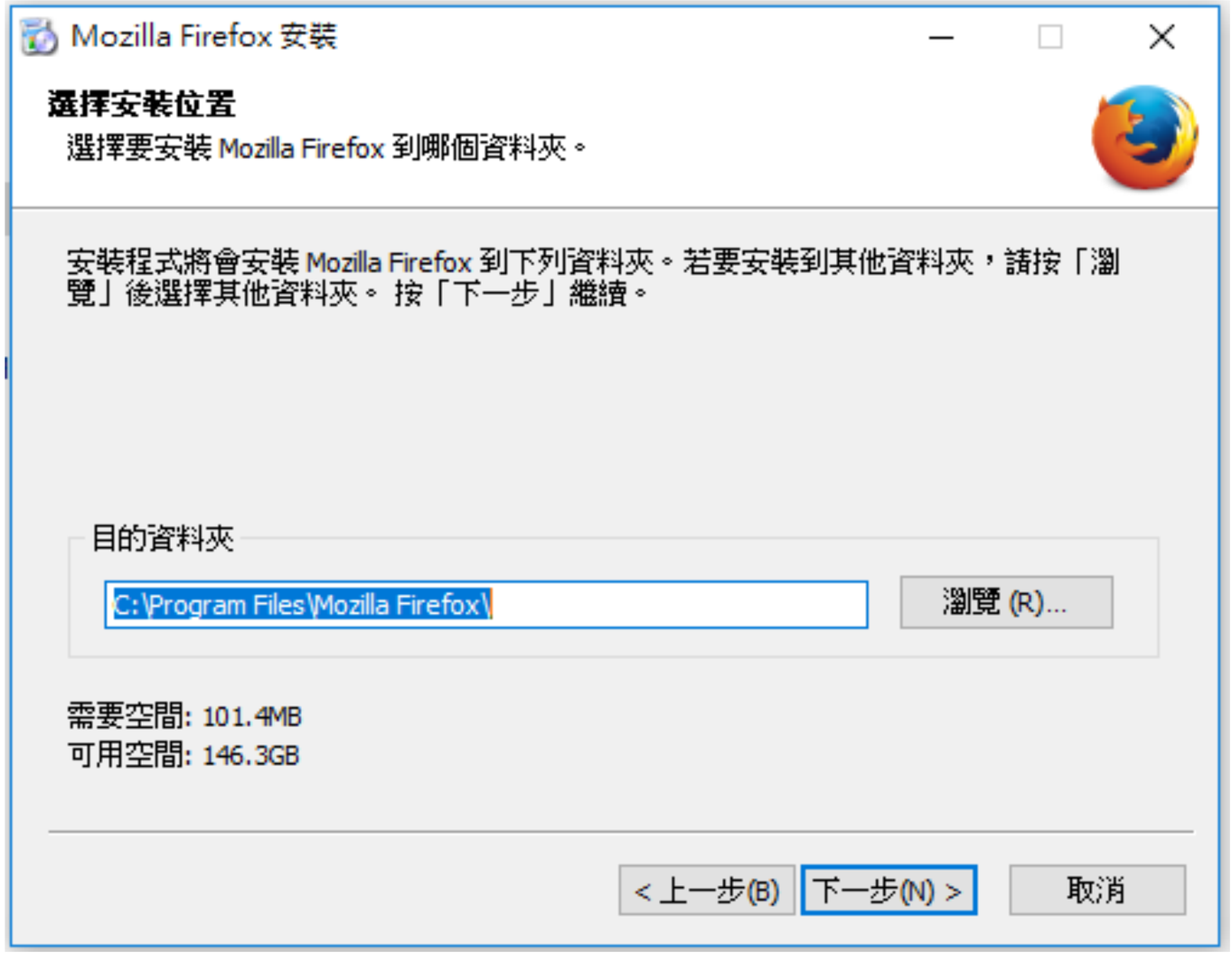


安裝Firefox

Mozilla Firefox 安裝
歡迎使用 Mozilla Firefox 安裝精靈
本精靈會引導您一步步安裝 Mozilla Firefox。
安裝前建議您關閉其他的常駐程式，為了更新相關的系統檔案可能必須重新啟動電腦。
按「下一步」繼續。
下一步(N) >
取消

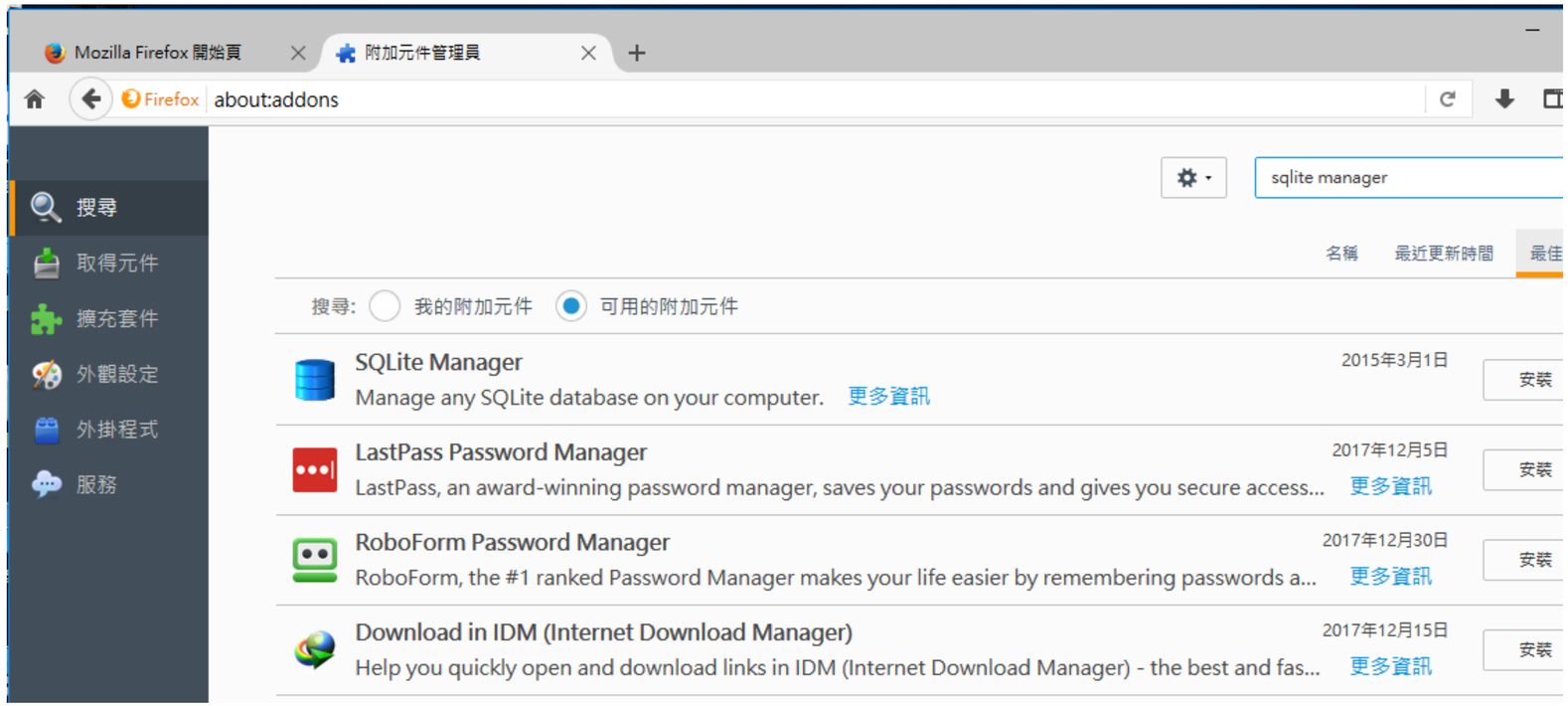

Mozilla Firefox 安裝
安裝類型
選擇安裝元件
請選擇您想安裝的類型後按「下一步」。
標準安裝 (S)
Firefox 會安裝最常用的元件。
自訂安裝 (C)
您可以自己選擇安裝選項，建議進階使用者選用。
< 上一步(B)
下一步(N) >
取消

python

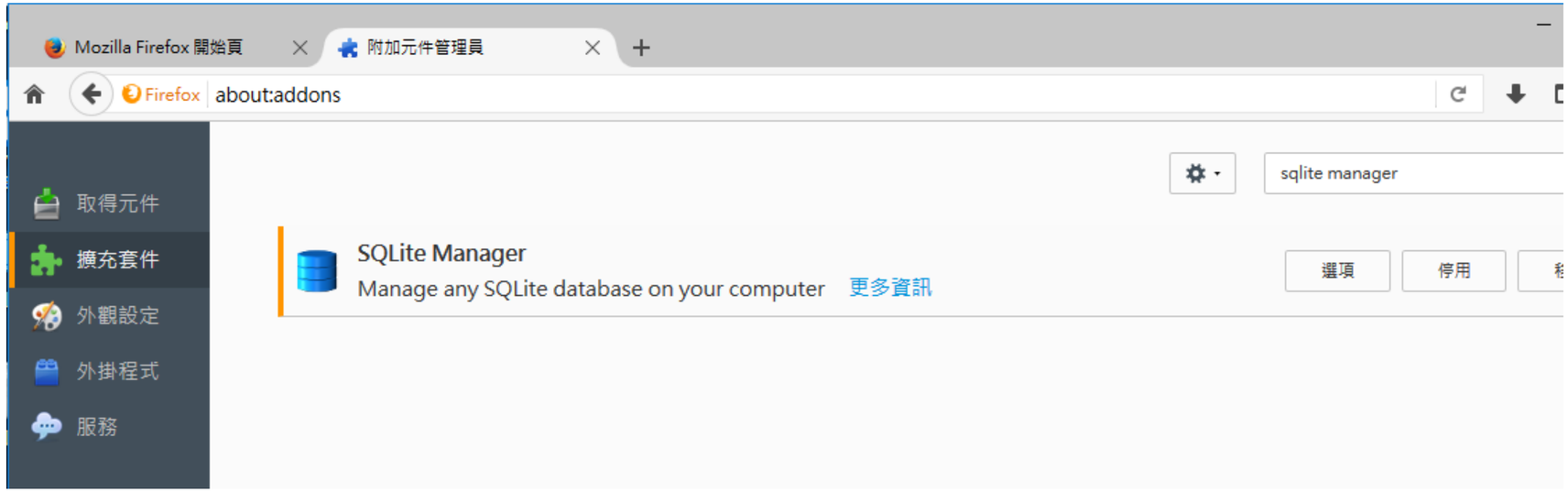


SQLite

安裝完Firefox後
安裝SQLite Manager

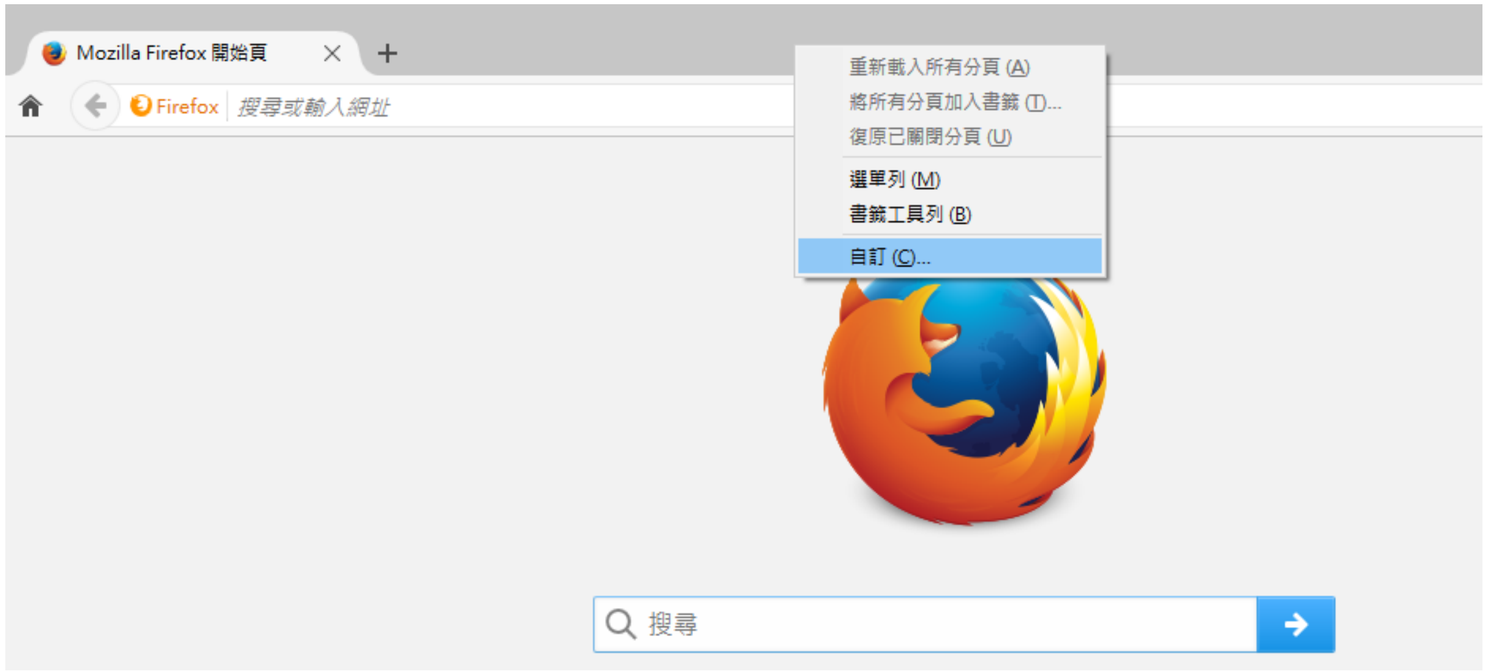


在擴充套件安裝選取
SQLite Manager

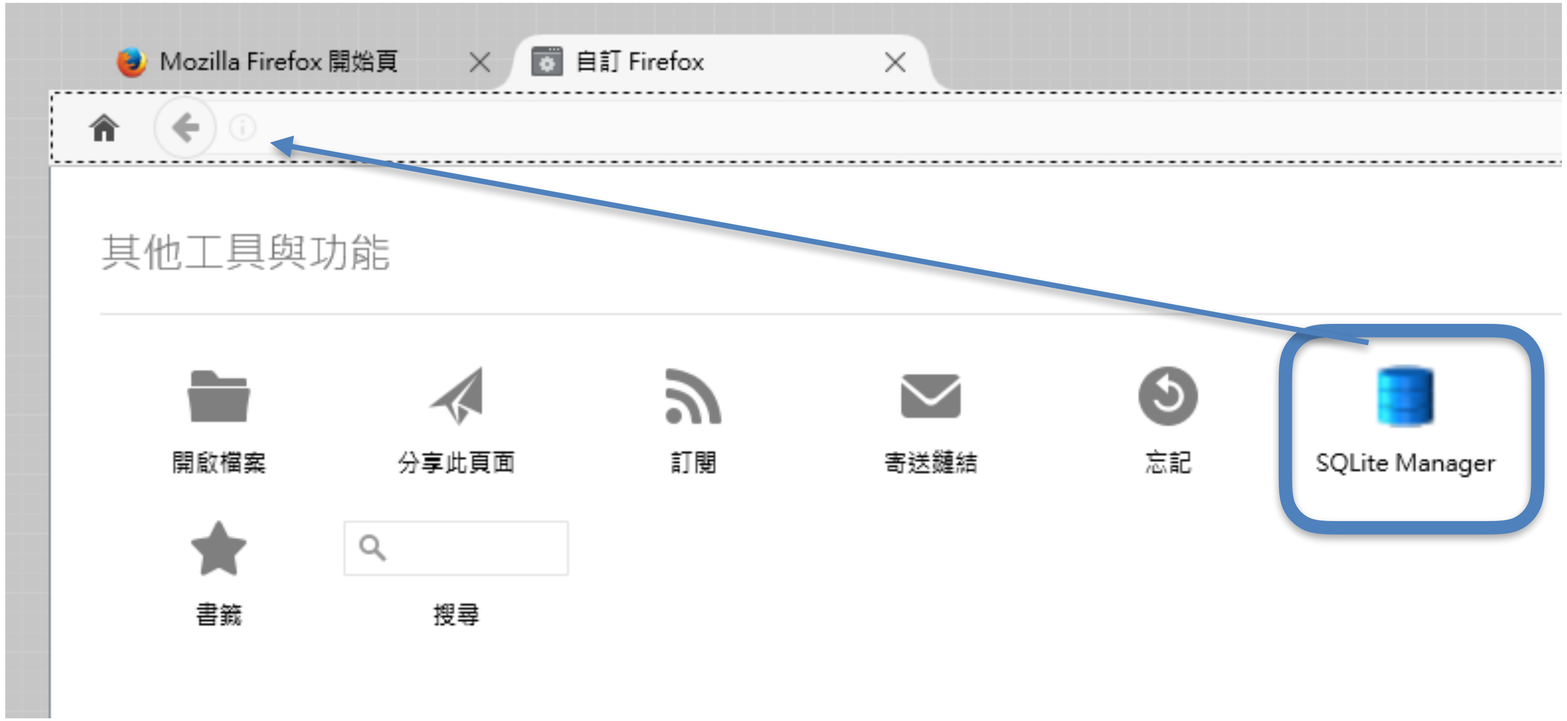


設定
SQLite Manager

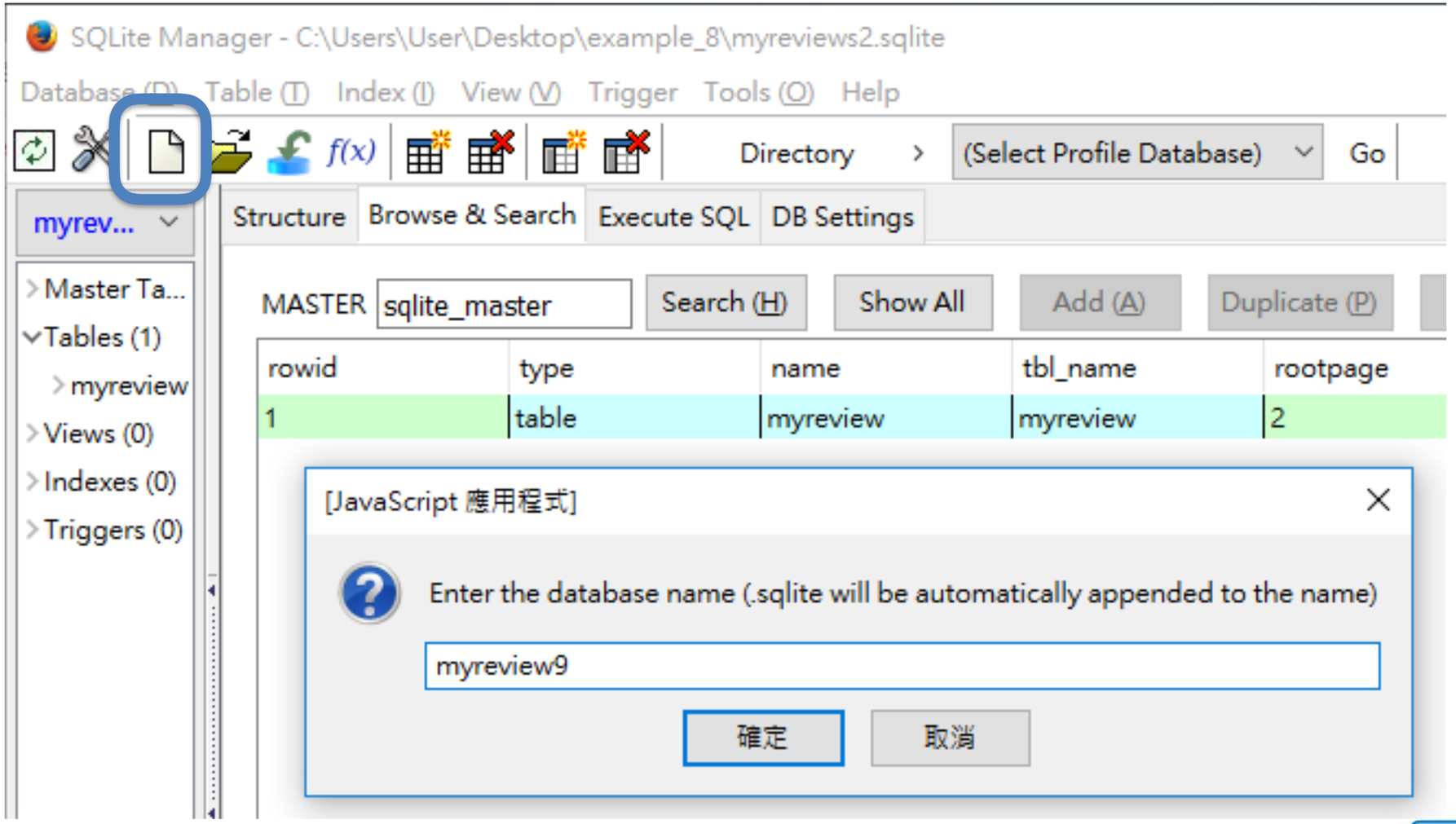
- 在工具列按下滑鼠右鍵->選取自訂



將SQLite Manager
拉到工具列



新增資料庫
myreview9



這是在Windows 10 執行的情況

- 連接myreview9.sqlite資料庫,Python檔案和資料庫要放在同一個目錄

```python

@author: justinwu
"""
import sqlite3
import os
conn=sqlite3.connect('myreview9.sqlite')
mycur=conn.cursor()
mycur.execute('create table myreview(
  review TEXT,sentiment integer,date text)')
example1='I would like the movie'
mycur.execute("insert into myreview(
 review,sentiment,date) values (
 ?,?,DATETIME('now'))",(example1,1))
example2='I do not like the movie'
mycur.execute("insert into myreview(
        review,sentiment,date) values (
    ?,?,DATETIME('now'))",(example2,0))
conn.commit()
conn.close()
```

Name	Type	Size	Value
example1	str	1	I would like the movie
example2	str	1	I do not like the movie

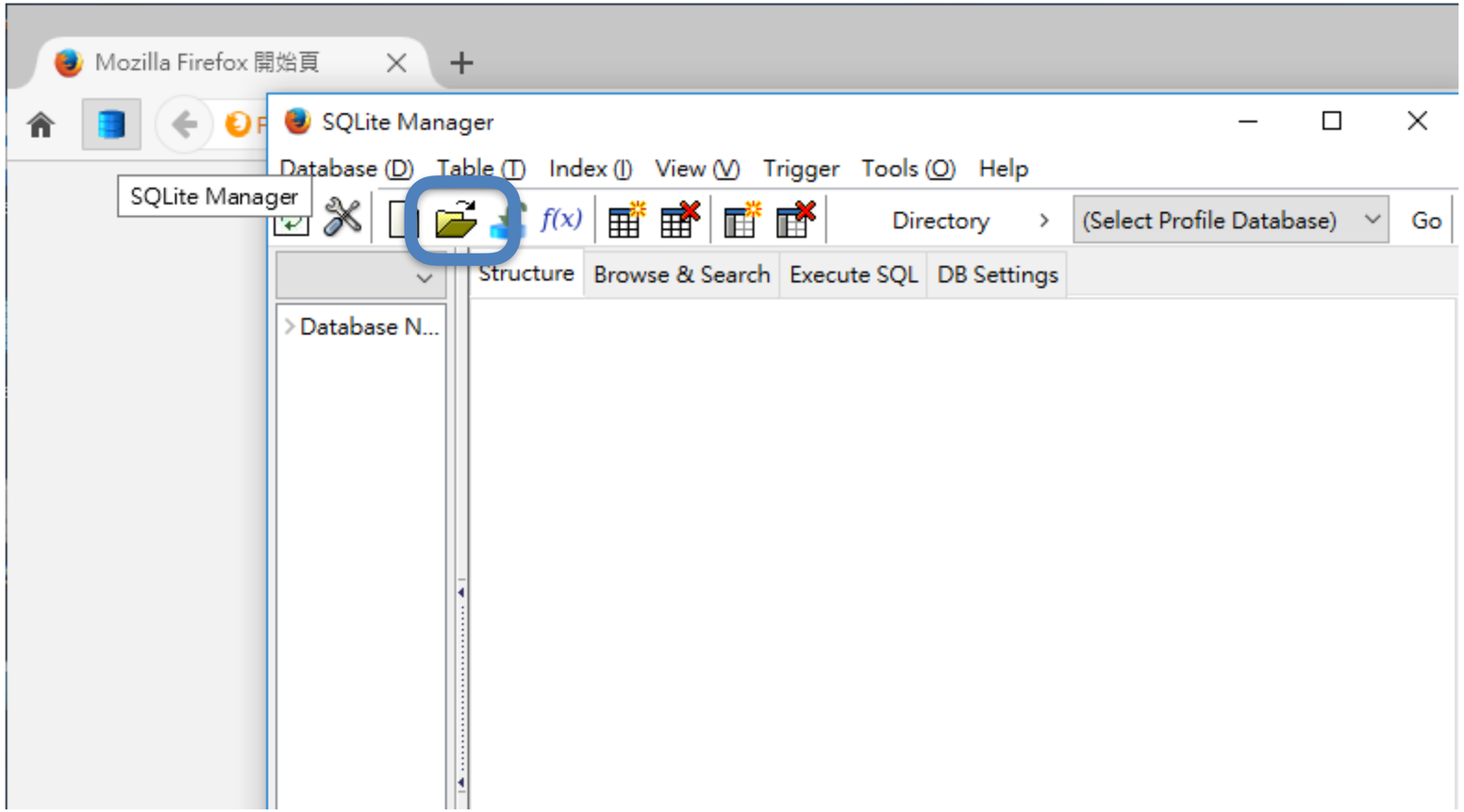
Variable explorer | File explorer | Help

IPython console

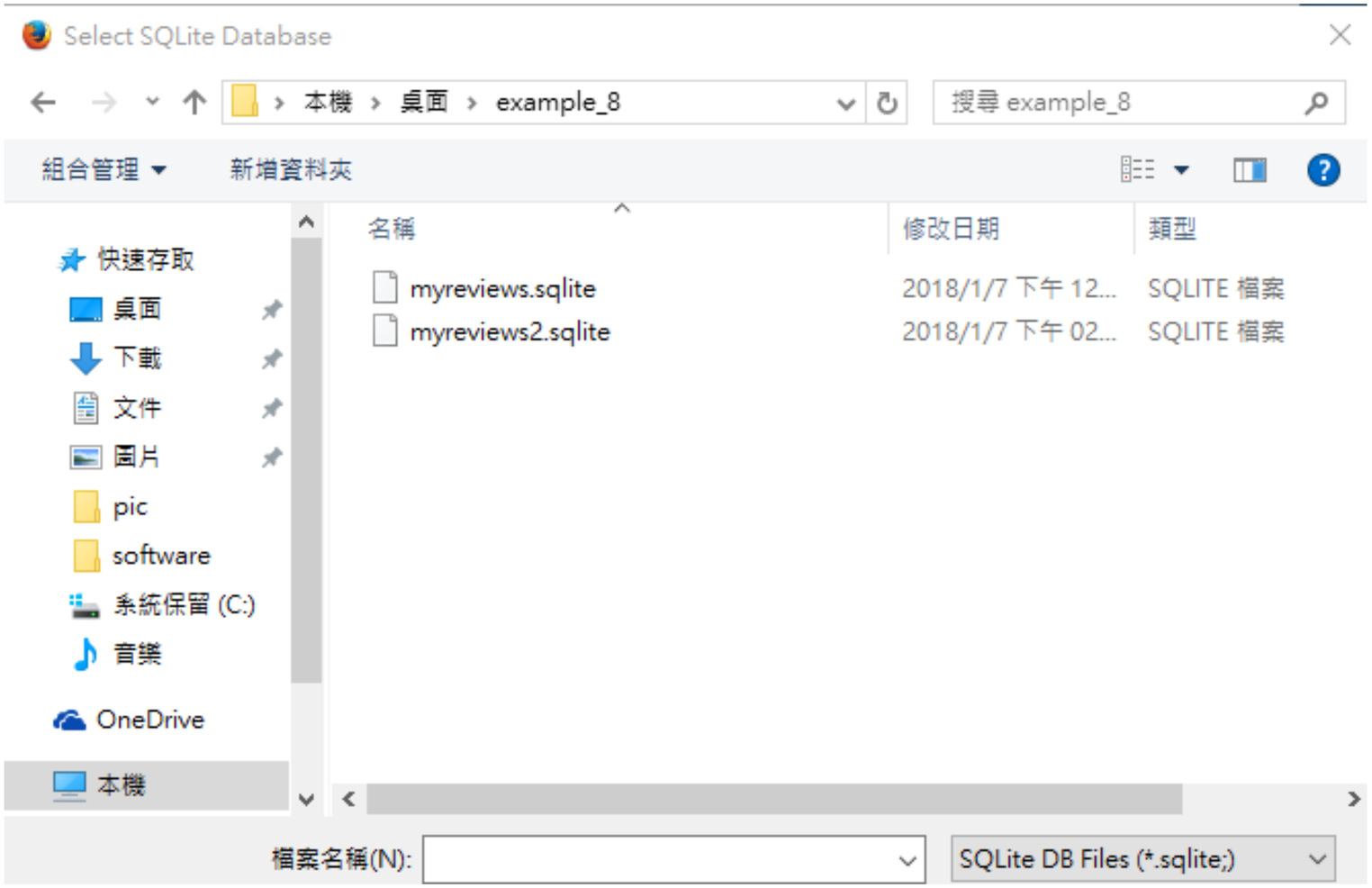
Console 1/A

```
In [5]: runfile('C:/Users/User/Desktop/
example_8/sqlite3.py', wdir='C:/Users/User/
Desktop/example_8')

In [6]:
```

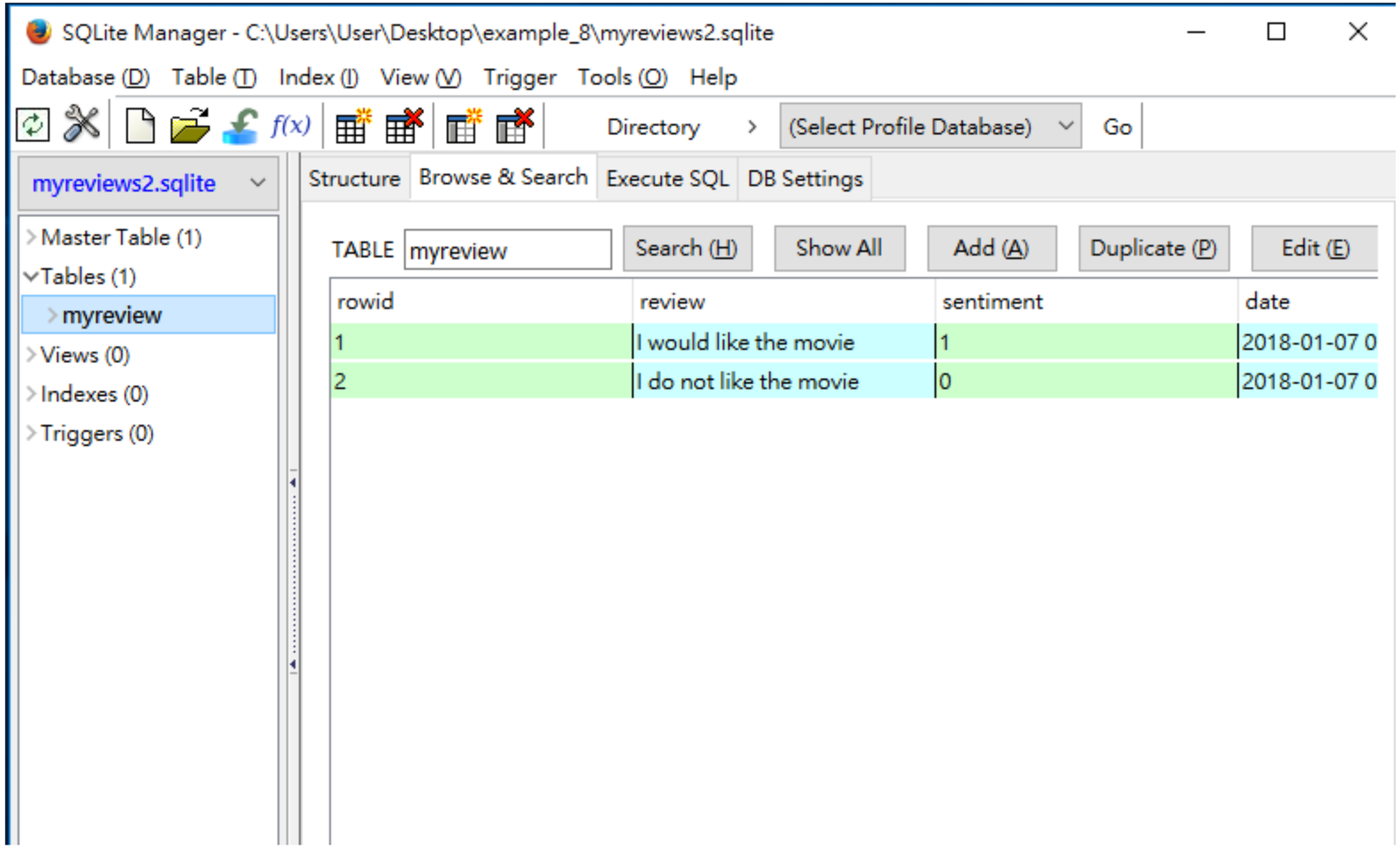
選取資料庫


Mozilla Firefox 開始頁
SQLite Manager
SQLite Manager
Database (D) Table (T) Index (I) View (V) Trigger Tools (O) Help
f(x)
Directory
(Select Profile Database)
Go
Structure
Browse & Search
Execute SQL
DB Settings
Database N...

選取
myreview2.sqlite



這是顯示的結果


SQLite Manager - C:\Users\User\Desktop\example_8\myreviews2.sqlite
Database (D) Table (T) Index (I) View (V) Trigger Tools (O) Help
Directory
(Select Profile Database)
Go
myreviews2.sqlite
Structure
Browse & Search
Execute SQL
DB Settings
Master Table (1)
Tables (1)
myreview
Views (0)
Indexes (0)
Triggers (0)
TABLE myreview
Search (H)
Show All
Add (A)
Duplicate (P)
Edit (E)
rowid
review
sentiment
date
1
I would like the movie
1
2018-01-07 0
2
I do not like the movie
0
2018-01-07 0

這是在Windows 10
執行選取查詢

Spyder (Python 3.6)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\User\Desktop\example_8

Editor - C:\Users\User\Desktop\example_8\select.py

temp.py sqlite3.py select.py

```python
Created on Sun Jan  7 12:24:27 2018

@author: justinwu
"""


import sqlite3
import os
conn=sqlite3.connect('myreviews.sqlite')
mycur=conn.cursor()
mycur.execute('select * from myreview')
myresults=mycur.fetchall()
conn.close()
print(myresults)

```

Variable explorer

Name	Type	Size	Value
my...	...	2	[('I would like ...

Variable explorer File explorer Help

IPython console

Console 1/A

```
In [4]: runfile('C:/Users/User/Desktop/example_8/select.py', wdir='C:/Users/User/Desktop/example_8')
[('I would like the movie', 1, '2018-01-07 04:43:46'), ('I do not like the movie', 0, '2018-01-07 04:43:46')]
```

在Mac安裝SQLite3

- 到https://www.sqlite.org/download.html
- 下載原始碼sqlite-autoconf-*.tar.gz
- `$tar xvfz sqlite-autoconf-3210000.tar.gz`
- `$cd sqlite-autoconf-3210000`
- `$./configure --prefix=/usr/local`
- `$make`
- `$make install`

```
60-250-191-83:sqlite chaiyenwu$ tar xvfz sqlite-autoconf-3210000.tar.gz
x sqlite-autoconf-3210000/
x sqlite-autoconf-3210000/compile
x sqlite-autoconf-3210000/config.guess
x sqlite-autoconf-3210000/configure
x sqlite-autoconf-3210000/sqlite3ext.h
x sqlite-autoconf-3210000/aclocal.m4
x sqlite-autoconf-3210000/install-sh
```

python

```
60-250-191-83:sqlite chaiyenwu$ cd sqlite-autoconf-3210000
60-250-191-83:sqlite-autoconf-3210000 chaiyenwu$ ./configure --prefix=/usr/local
```

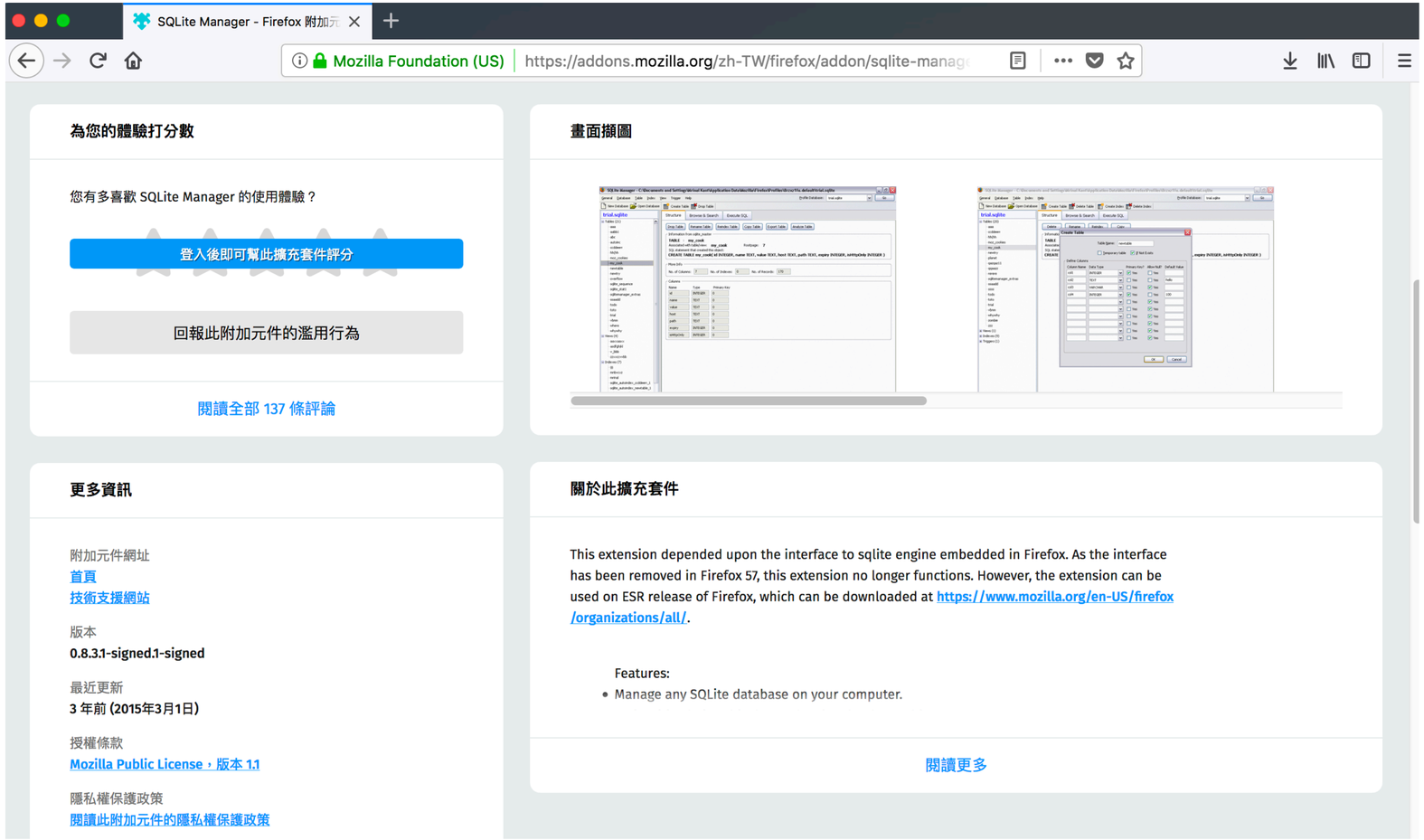
```
60-250-191-83:sqlite-autoconf-3210000 chaiyenwu$ sudo make install
Password:
 ./install-sh -c -d '/usr/local/lib'
 /bin/sh ./libtool   --mode=install /usr/bin/install -c   libsqlite3.la '/usr/local/l
ib'
libtool: install: /usr/bin/install -c .libs/libsqlite3.0.dylib /usr/local/lib/libsqli
te3.0.dylib
libtool: install: (cd /usr/local/lib && { ln -s -f libsqlite3.0.dylib libsqlite3.dyli
b || { rm -f libsqlite3.dylib && ln -s libsqlite3.0.dylib libsqlite3.dylib; }; })
libtool: install: /usr/bin/install -c .libs/libsqlite3.lai /usr/local/lib/libsqlite3.
la
libtool: install: /usr/bin/install -c .libs/libsqlite3.a /usr/local/lib/libsqlite3.a
libtool: install: chmod 644 /usr/local/lib/libsqlite3.a
libtool: install: ranlib /usr/local/lib/libsqlite3.a
 ./install-sh -c -d '/usr/local/bin'
  /bin/sh ./libtool   --mode=install /usr/bin/install -c sqlite3 '/usr/local/bin'
libtool: install: /usr/bin/install -c sqlite3 /usr/local/bin/sqlite3
 ./install-sh -c -d '/usr/local/include'
 /usr/bin/install -c -m 644 sqlite3.h sqlite3ext.h '/usr/local/include'
 ./install-sh -c -d '/usr/local/share/man/man1'
 /usr/bin/install -c -m 644 sqlite3.1 '/usr/local/share/man/man1'
```

QLite

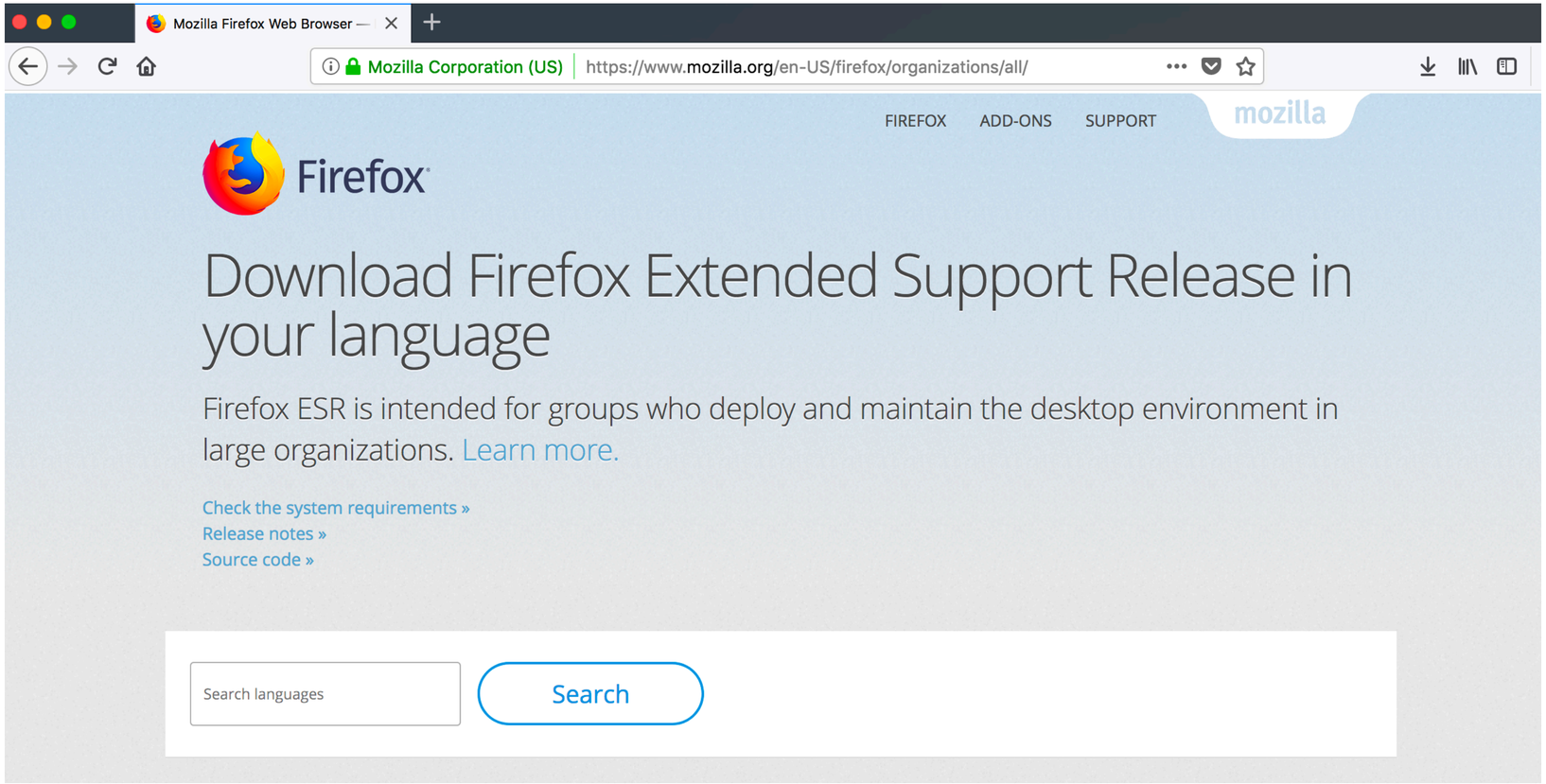
- 使用sqlite3指令登錄

```
60-250-191-83:sqlite-autoconf-3210000 chaiyenwu$ sqlite3
SQLite version 3.20.1 2017-08-24 16:21:36
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite>
```

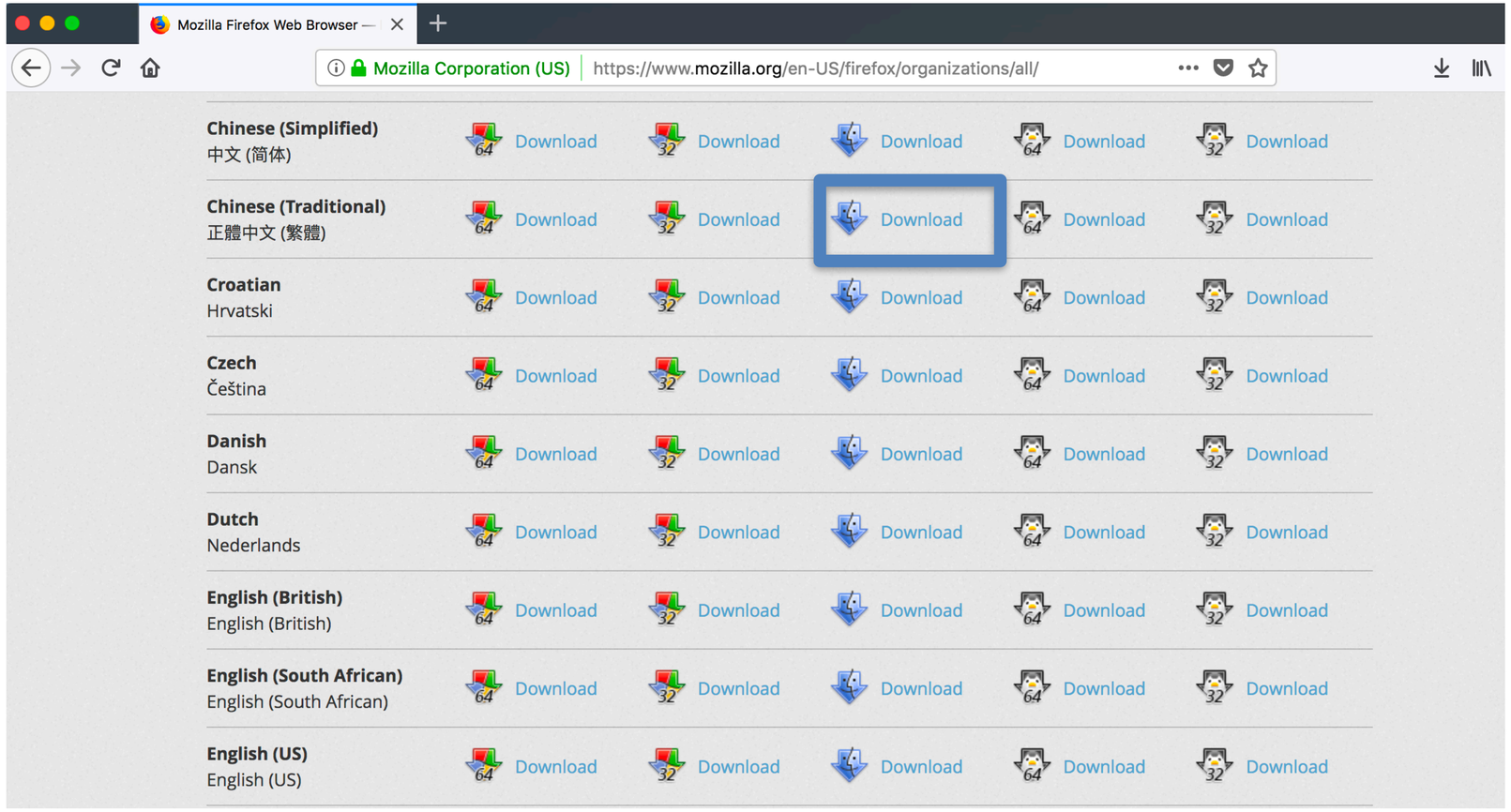
安裝firefox瀏覽器



安裝中文版的Firefox延伸



- 選取繁體中文mac檔


Mozilla Firefox Web Browser
Mozilla Corporation (US)
https://www.mozilla.org/en-US/firefox/organizations/all/
Chinese (Simplified)
中文 (简体)
Chinese (Traditional)
正體中文 (繁體)
Croatian
Hrvatski
Czech
Čeština
Danish
Dansk
Dutch
Nederlands
English (British)
English (British)
English (South African)
English (South African)
English (US)
English (US)
Download

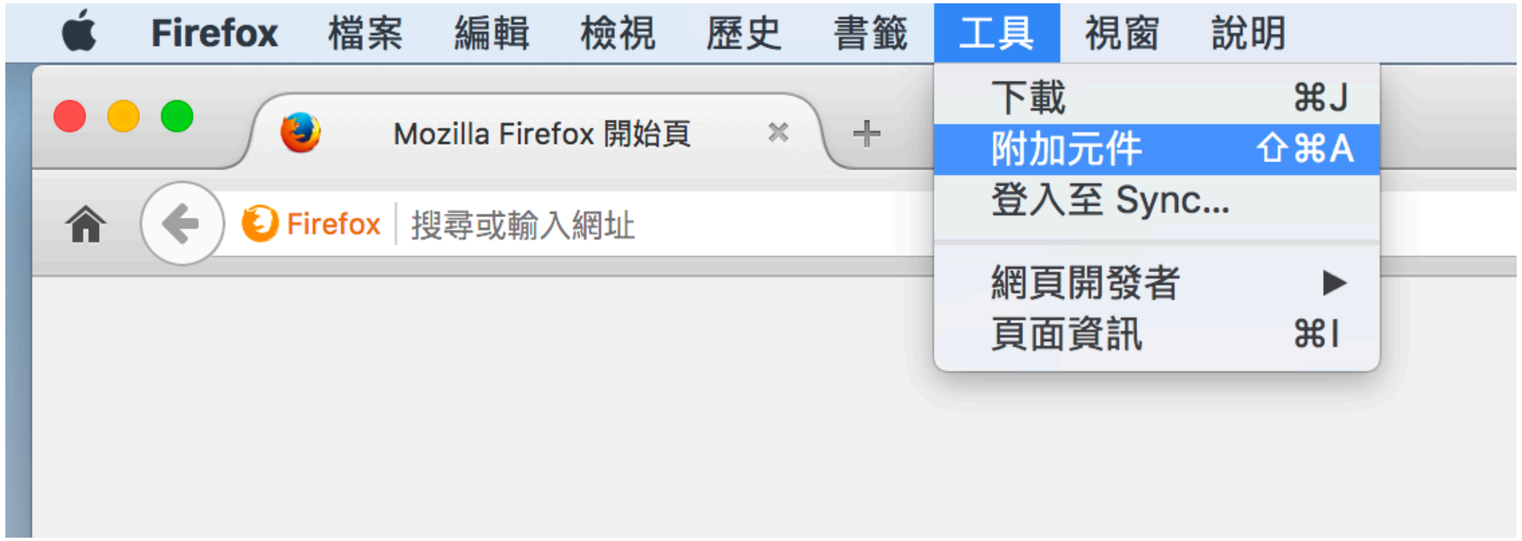
儲存並且安裝 Firefox瀏覽器

python

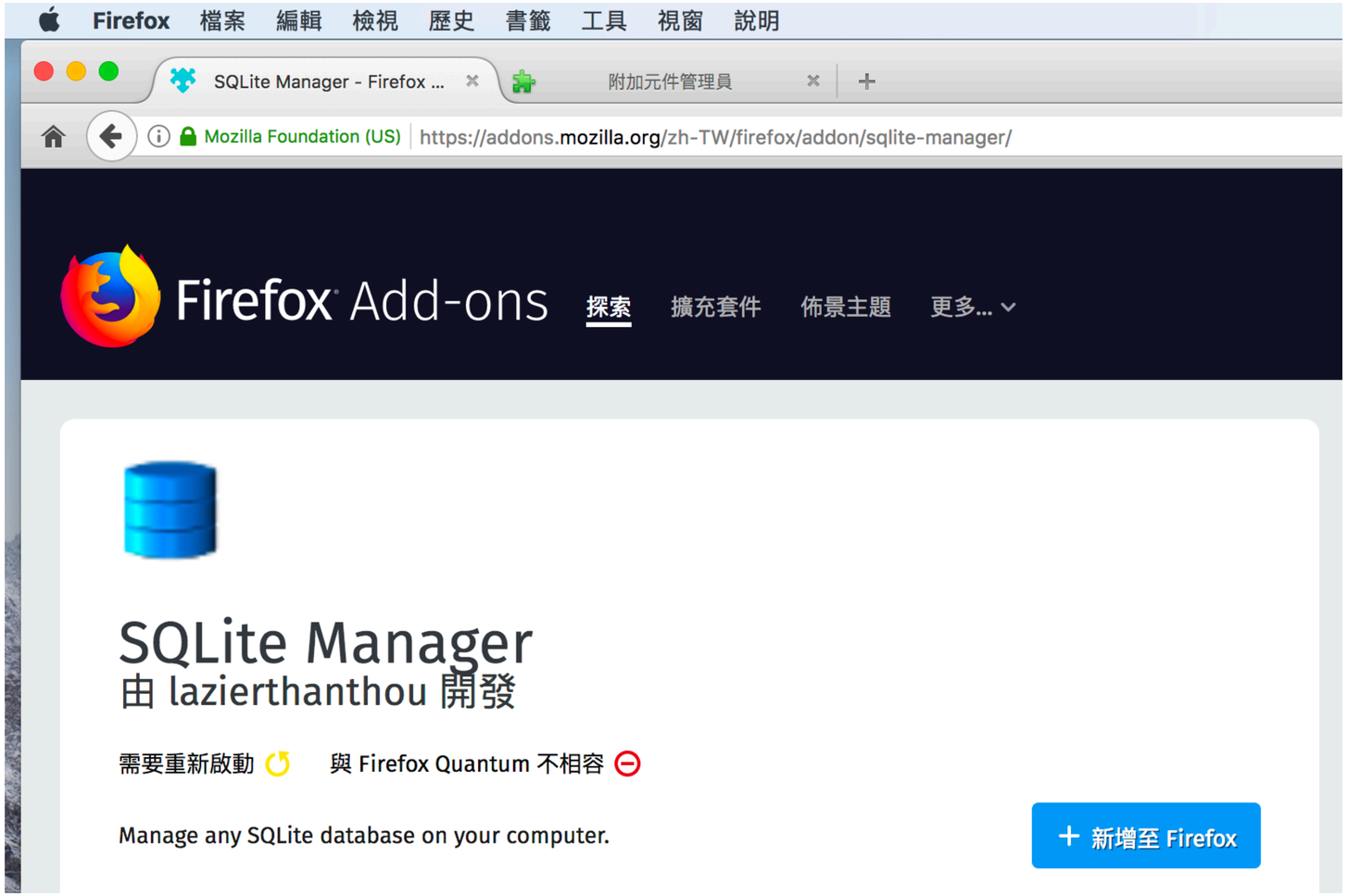
SQLite

安裝SQLite Manager

- 選取瀏覽器->工具->附加元件

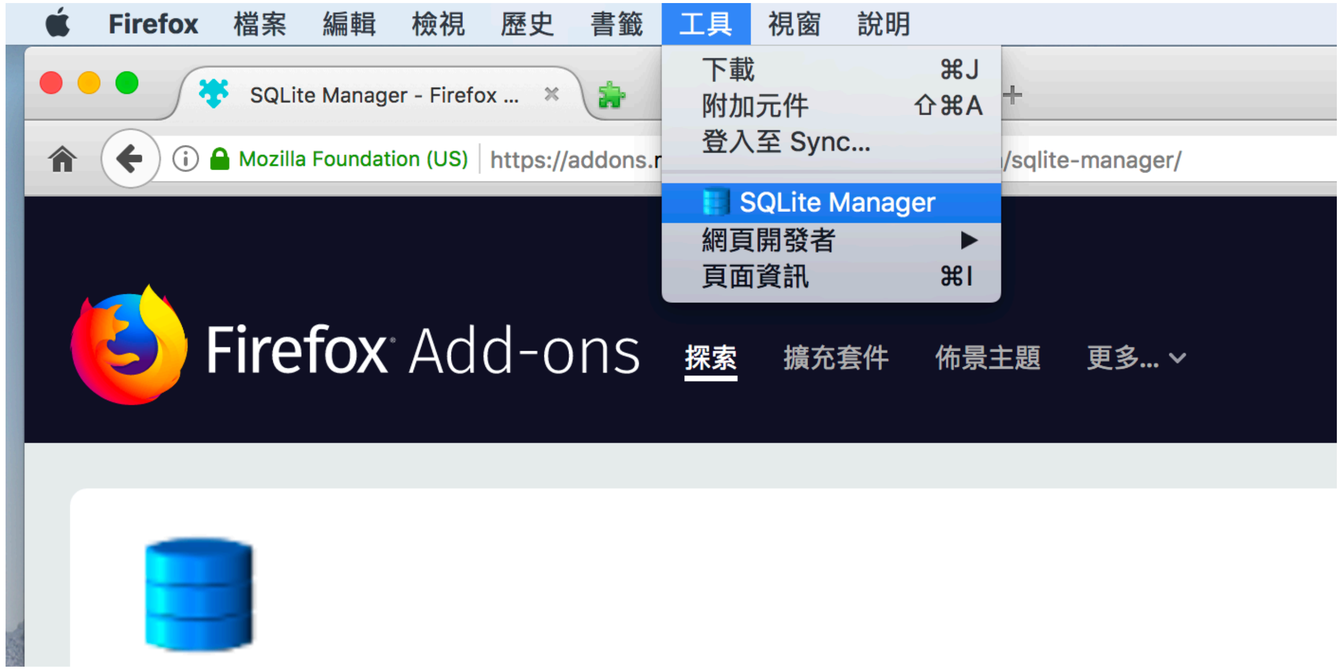


將SQLite Manager新增到瀏覽器firefox

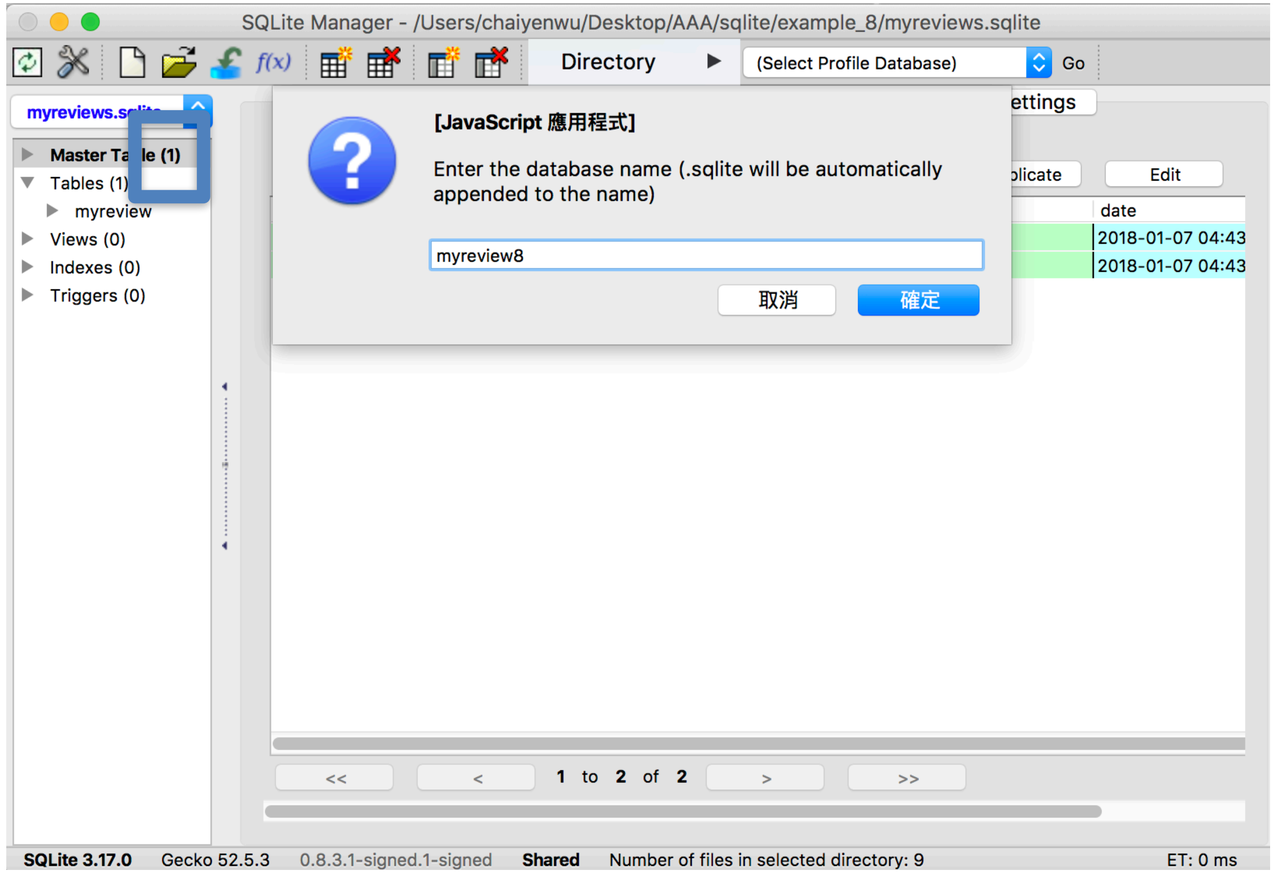


安裝及打開
SQLite Manager

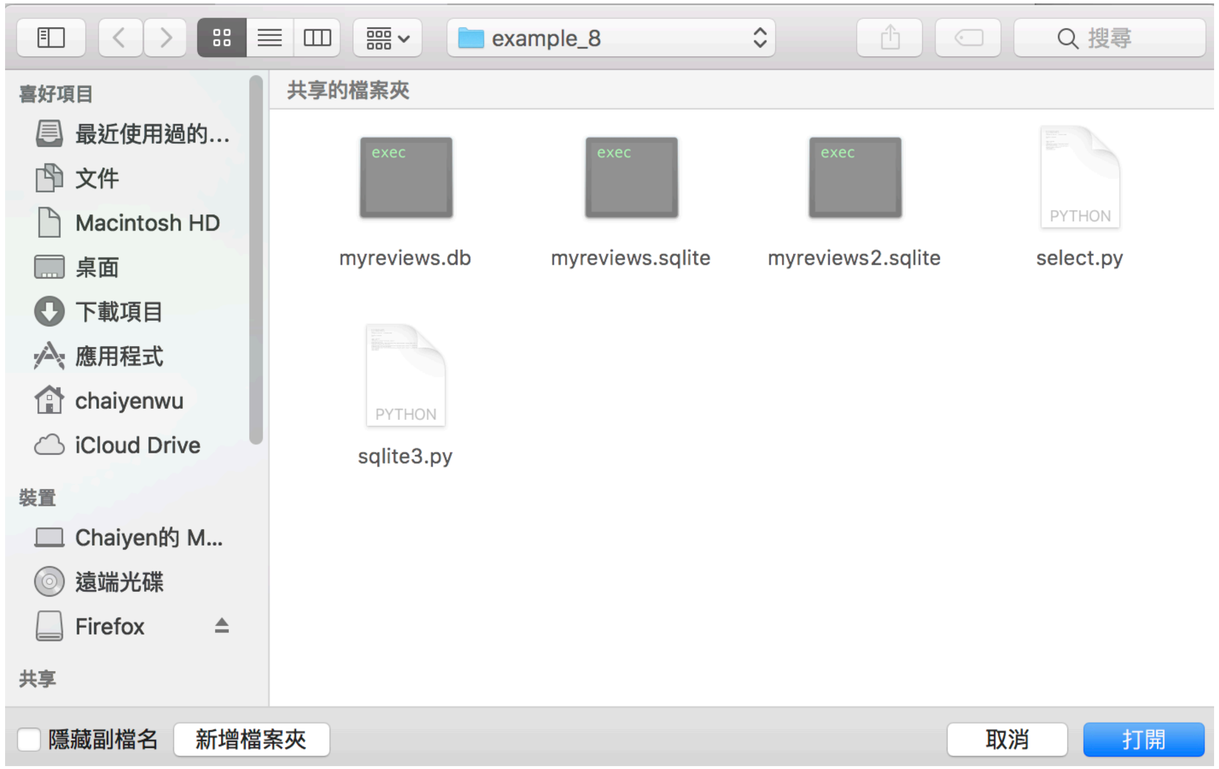
- 在Firefox選取工具->SQLite Manager



新增myreview8
資料庫



選取資料庫
放置的地方


example_8
搜尋
喜好項目
最近使用過的...
文件
Macintosh HD
桌面
下載項目
應用程式
chaiyenwu
iCloud Drive
裝置
Chaiyen的 M...
遠端光碟
Firefox
共享
共享的檔案夾
exec
myreviews.db
exec
myreviews.sqlite
exec
myreviews2.sqlite
PYTHON
select.py
PYTHON
sqlite3.py
隱藏副檔名
新增檔案夾
取消
打開

SQLite資料庫引擎

- SQLite是目前世界上最廣泛使用的免費資料庫引擎，一套實作大部分SQL 92標準的函數庫，支援交易的SQL資料庫引擎，網址為：http://www.sqlite.org/。
- SQLite是資料庫系統：
  - SQLite資料庫只是一個檔案，可以直接使用檔案權限來管理資料庫，而不用自行處理資料庫的使用者權限管理。
  - 單一檔案的SQLite資料庫。
  - SQLite不需要啟動。

2.SQL語言

- SQL標準的資料庫語言，可以存取和更新資料庫的記錄資料。SQLite支援的SQL指令主要分成兩大類，如下所示：
  - 資料定義語言（DDL）：建立資料表和定義資料表欄位。
  - 資料操作語言（DML）：資料表記錄的查詢、插入、刪除和更新。

SQL語言的基礎 - CREATE TABLE建立資料表

- CREATE TABLE 指令可以在資料庫建立資料表，如下所示：

```
create table myreview(review
TEXT,sentiment integer,date text)
```

SQL語言的基礎 - SELECT查詢記錄(語法)

- 在SQL語言的資料查詢指令只有一個SELECT指令，其基本語法如下所示：

  SELECT 欄位1, 欄位2 FROM 資料表
  WHERE 條件
  ORDER BY 欄位清單

- SELECT指令的欄位1~2為記錄的欄位，conditions為查詢條件，從資料表取回符合WHERE子句條件的記錄，顯示欄位1和2，並且以ORDER BY子句的欄位來排序。

SQL語言的基礎 - SELECT查詢記錄(語法)

- 在SQL語言的資料查詢指令只有一個SELECT指令，其基本語法如下所示：

  SELECT 欄位1, 欄位2 FROM 資料表
  WHERE 條件
  ORDER BY 欄位清單

- SELECT指令的欄位1~2為記錄的欄位，條件為查詢條件，從資料表取回符合WHERE子句條件的記錄，顯示欄位1和2，並且以ORDER BY子句的欄位來排序。

SQL語言的基礎 - SELECT查詢記錄("*"符號)

- SELECT指令可以使用"*"符號代表資料表的所有欄位，表示取回資料表記錄的所有欄位，如下所示：
  - select * from myreview

SELECT查詢記錄(WHERE子句-1)

- SQL查詢如果是單一條件，在WHERE子句條件：
  - 文字欄位需要使用單引號括起，例如：書名為Swift時的SQL指令字串，如下所示：

    SELECT * FROM contacts WHERE name='Swift'
  - 數字欄位不需要使用單引號括起，例如：價格為55元的SQL指令，如下所示：

    SELECT * FROM contacts WHERE value=55

INSERT新增記錄

- SQL新增記錄INSERT指令可以新增一筆記錄例如：在myreview資料表新增一筆記錄的SQL指令：

insert into myreview(review,sentiment,date) values ('I would like the movie’, 1,DATETIME('now'))

- SQL指令的括號中是欄位清單（不用全部，但需包含所有not null欄位），字串欄位值使用單引號括起，數字沒有。

UPDATE更新記錄

- SQL更新記錄UPDATE指令是將資料表內符合條件的記錄，更新其欄位內容，例如：在 myreview 資料表更改 sentiment 的SQL指令，如下所示：

```
UPDATE myreview SET sentiment =1
      WHERE review='I will like the movie'
```

- SQL指令的WHERE條件為review欄位，然後使用SET子句更新 sentiment 欄位資料。

DELETE刪除記錄

- SQL刪除記錄DELETE指令是將資料表內符合條件的記錄刪除掉。在 myreview 資料表刪除記錄的SQL指令，如下所示：

```
delete from myreview WHERE review='I will like
  the movie'
```

- SQL指令的WHERE條件為review欄位，也就是將符合review條件的記錄刪除掉。

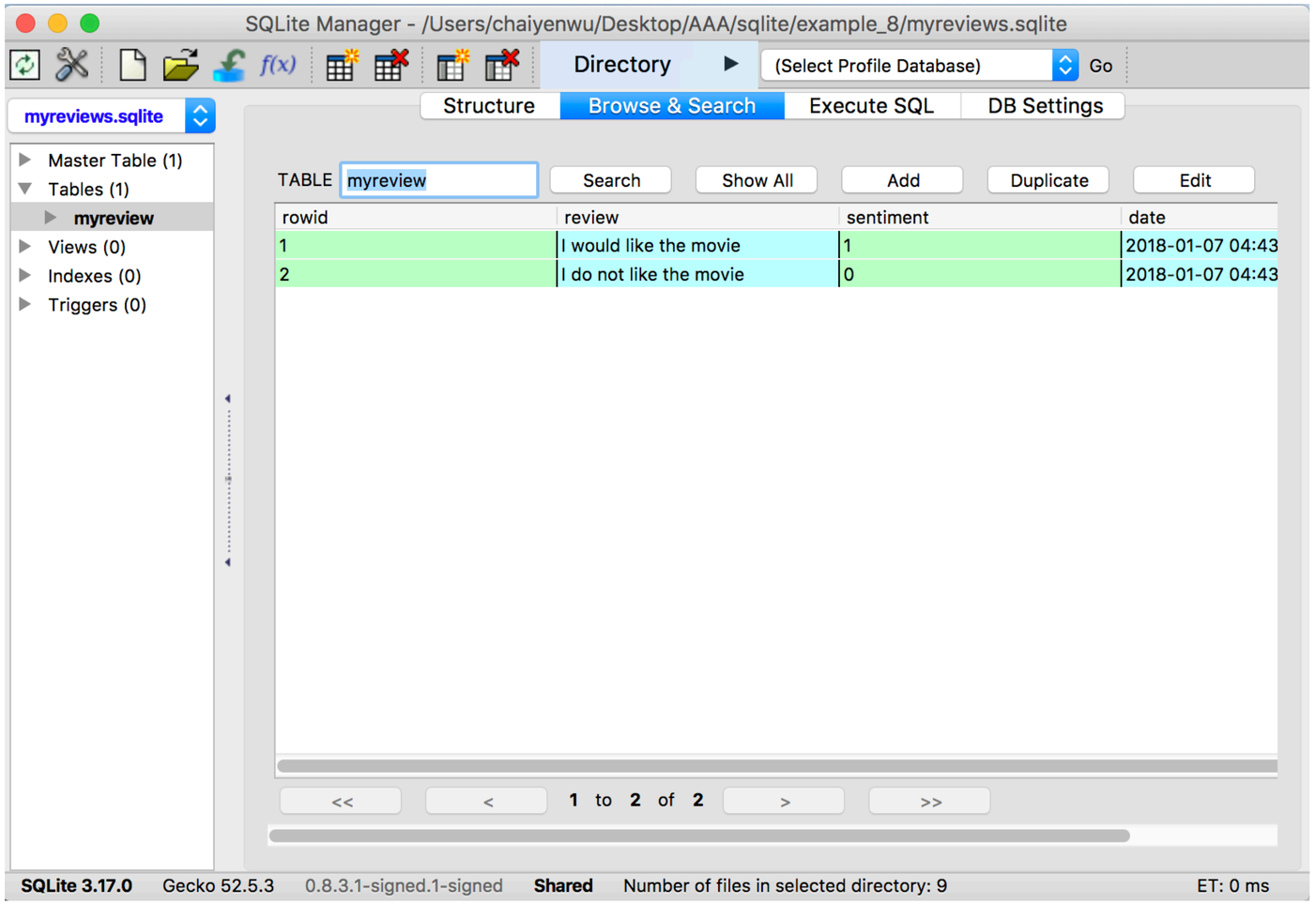
python

- 使用python連接資料庫並且建立資料表

```
import sqlite3
import os
conn=sqlite3.connect('myreviews8.sqlite')
mycur=conn.cursor()
mycur.execute('create table myreview(review
        TEXT,sentiment integer,date text)')
example1='I would like the movie'
mycur.execute("insert into myreview(review,
  sentiment,date) values (?,?,DATETIME('now'))",(example1,1))
example2='I do not like the movie'
mycur.execute("insert into myreview(review
   ,sentiment,date) values (?,?,DATETIME('now'))",(example2,0))
conn.commit()
conn.close()
```

SQLite

這是觀看
資料庫資料


SQLite Manager - /Users/chaiyenwu/Desktop/AAA/sqlite/example_8/myreviews.sqlite
Directory
(Select Profile Database)
Go
myreviews.sqlite
Structure
Browse & Search
Execute SQL
DB Settings
Master Table (1)
Tables (1)
myreview
Views (0)
Indexes (0)
Triggers (0)
TABLE myreview
Search
Show All
Add
Duplicate
Edit
rowid
review
sentiment
date
1
I would like the movie
1
2018-01-07 04:43
2
I do not like the movie
0
2018-01-07 04:43
<<
<
1 to 2 of 2
>
>>
SQLite 3.17.0
Gecko 52.5.3
0.8.3.1-signed.1-signed
Shared
Number of files in selected directory: 9
ET: 0 ms

使用Python

連接Sqlite資料庫

- 使用Python連接Sqlite資料庫並且執行選取查詢select

```python
import sqlite3
import os
conn=sqlite3.connect('myreviews.sqlite')
mycur=conn.cursor()
mycur.execute('select * from myreview')
myresults=mycur.fetchall()
conn.close()
print(myresults)
```

```
sqlite/example_8/select.py', wdir='/Users/
chaiyenwu/Desktop/AAA/sqlite/example_8')
[('I would like the movie', 1, '2018-01-07
04:43:46'), ('I do not like the movie', 0,
'2018-01-07 04:43:46')]

In [2]:
```

3.SQLite3資料庫模組

```
#輸入sqlite3模組
import sqlite3
#建立和myreviews.sqlite資料庫連接
conn=sqlite3.connect('myreviews.sqlite
')
#使用資料庫指標cursor()
mycur=conn.cursor()
#使用資料庫指標mycur執行SQL指令
mycur.execute()
```

SQLite3資料庫模組

- #送出指令並交由資料庫執行使用commit()函數
- conn.commit()
- #關閉資料庫使用close()函數
- conn.close()

SQLite3資料庫模組

- #輸入sqlite3模組
- import sqlite3
- import os
- #建立和myreviews.sqlite資料庫連接
- conn=sqlite3.connect('myreviews.sqlite')
- #使用資料庫指標cursor()
- mycur=conn.cursor()
- #使用資料庫指標mycur執行SQL指令
- mycur.execute('select * from myreview')
- #使用指標mycur的fetchall()函數來讀取所有資料
- myresults=mycur.fetchall()
- #關閉資料庫使用close()函數
- conn.close()

python

insert語法

```
9 #輸入sqlite3模組
10 import sqlite3
11 import os
12 #建立和myreviews.sqlite資料庫連接
13 conn=sqlite3.connect('myreviews.sqlite')
14 #使用資料庫指標cursor()
15 mycur=conn.cursor()
16 #使用資料庫指標mycur執行SQL指令
17 example3='I will like the movie'
18 #sql指令insert插入資料
19 mycur.execute("insert into myreview(review,sentiment,date)
20              values (?,?,DATETIME('now'))",(example3,0))
21 conn.commit()
22 #關閉資料庫使用close()函數
23 conn.close()
```

SQLite

update語法

```python
Created on Sun Jan  7 20:05:28 2018

@author: justinwu
"""

#輸入sqlite3模組
import sqlite3
import os
#建立和myreviews.sqlite資料庫連接
conn=sqlite3.connect('myreviews.sqlite')
#使用資料庫指標cursor()
mycur=conn.cursor()
#使用資料庫指標mycur執行SQL指令
example3='I will like the movie'
#sql指令insert插入資料

mycur.execute("UPDATE myreview SET sentiment =1
        WHERE review='I will like the movie'")
conn.commit()
#關閉資料庫使用close()函數
conn.close()
```

```
Variable explorer   File explorer   Help
IPython console
Console 1/A

In [7]: runfile('/Users/justinwu/Desktop/TOP/Python Strong/sqlite/
example_8/select.py', wdir='/Users/justinwu/Desktop/TOP/Python Strong/
sqlite/example_8')
[('I would like the movie', 1, '2018-01-07 04:43:46'), ('I do not like t
movie', 0, '2018-01-07 04:43:46'), ('I will like the movie', 0,
'2018-01-07 12:02:37')]

In [8]: runfile('/Users/justinwu/Desktop/TOP/Python Strong/sqlite/
example_8/update.py', wdir='/Users/justinwu/Desktop/TOP/Python Strong/
sqlite/example_8')

In [9]: runfile('/Users/justinwu/Desktop/TOP/Python Strong/sqlite/
example_8/select.py', wdir='/Users/justinwu/Desktop/TOP/Python Strong/
sqlite/example_8')
[('I would like the movie', 1, '2018-01-07 04:43:46'), ('I do not like t
movie', 0, '2018-01-07 04:43:46'), ('I will like the movie', 1,
'2018-01-07 12:02:37')]

In [10]:
```

- Thank You.

Python 精通程式語言
人機界面及影像處理

吳佳諺 老師

9.使用matplotlib畫圖
10.網站擷取使用Python
11.行程和執行緒
12.資料及檔案處理
13.SQLite資料庫
14.人機界面及影像處理
15.Python遊戲設計

人機界面及影像處理

- 14-1人機界面Tkinter
- 14-2Pillow影像處理

14-1人機界面Tkinter

- 在圖形化介面，一個事件代表了一些動作，例如按下按鈕，拖拉滑鼠，在鍵盤上按下按鈕，在視窗上按下關掉視窗的按鈕，或者是任何期望引發回應的動作。

14-1-1事件驅動程式設計

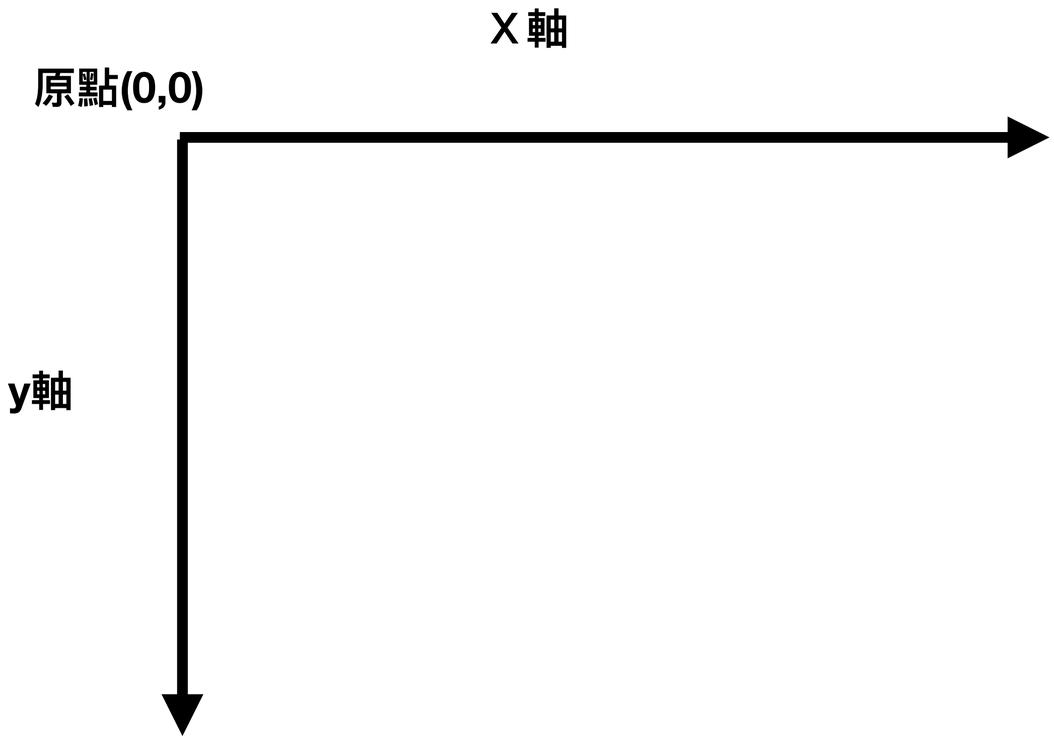
- 事件驅動程式設計使用告知和回應的方法來作程式設計。
- 告知物件被稱為事件(event)。
- 使用事件和事件處理。
- 一個事件(event)就是一個物件，它作用在另外一個稱為傾聽的物件上。
- 事件的傳送稱為啟動事件。
- Canvas.bind()函數為註冊傾聽的函數物件.
- mycanvas.bind("<Button-1>", self.MouseEvent)

傾聽的函數物件 MouseEvent()

- 當按下滑鼠左鍵就會去執行MouseEvent()函數

```
25    def MouseEvent(self, event):
26        print("滑鼠座標", event.x, event.y)
27        print("滑鼠在視窗位置", event.x_root, event.y_root)
28        print("按下哪一個滑鼠按鈕 ", event.num)
```

14-1-2座標軸



使用tkinter套件

- 從tkinter人機界面套件輸入所有模組
- from tkinter import *
- #建立最上層視窗
- root = Tk()

python

- #從tkinter人機界面套件輸入所有模組
- from tkinter import *
- class Window(Frame):
- #初始化視窗
- def __init__(self, master=None):
- Frame.__init__(self, master)
- self.master = master
- #建立最上層視窗
- root = Tk()
- #建立視窗實體
- app = Window(root)

14-1-3Tkinter按鈕

- 建立主視窗

  root = Tk()

- 設定視窗大小400*300

  root.geometry("400x300")

  app = Window(root)#建立視窗實體

  root.mainloop() #主迴圈mainloop

Tkinter按鈕

```python
from tkinter import *

class Window(Frame):
    def __init__(self, master=None):
        Frame.__init__(self, master)
        self.master = master
        self.init_window()
    #建立初始化視窗
    def init_window(self):
        #改變主視窗的名稱
        self.master.title("人機界面GUI")
        #使用包裝管理將widget視窗放到主視窗容器內
        self.pack(fill=BOTH, expand=1)
        # 建立按鈕實體
        quitButton = Button(self, text="Quit離開")
        # 將按鈕實體放到(100,100)座標位置
        quitButton.place(x=100, y=100)
root = Tk()
#視窗大小設定
root.geometry("200x200")
app = Window(root)
root.mainloop()
```

14-1-4Tkinter事件處理

- quitButton = Button(self, text=“Exit",command=self.client_exit)
- 按下Quit離開按鈕就會執行client_exit()函數離開視窗
- self.pack()將視窗加入包裝管理容器

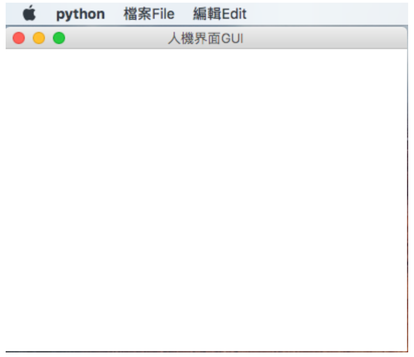
Tkinter事件處理

```python
from tkinter import *  #從tkinter人機界面套件輸入所有模組
class Window(Frame):
    ##初始化視窗
    def __init__(self, master=None):
        # Frame類別初始化,並將主視窗帶入.
        Frame.__init__(self, master)
        #參考master的tk視窗
        self.master = master
        #執行init_window()
        self.init_window()
    #init_window () 建立
    def init_window(self):
        #改變我們主視窗的名稱
        self.master.title("GUI")
        #使用包裝管理將widget視窗放到主視窗容器內
        self.pack(fill=BOTH, expand=1)
        #建立按鈕實體
        quitButton = Button(self, text="Exit",command=self.client_exit)
        #放置按鈕到視窗主體中
        quitButton.place(x=0, y=0)
    def client_exit(self):
        exit()
root = Tk()#建立主視窗
root.geometry("400x300")#設定視窗大小400*300
app = Window(root)#建立視窗實體
root.mainloop()  #主迴圈mainloop
```

14-1-5Tkinter選單

- 使用Menu()選單
- file.add_command()函數將命令加入到選單中

```
9  from tkinter import *
10 #Frame類別是在tkinter模組中
11 class Window(Frame):
12     def __init__(self, master=None):
13         Frame.__init__(self, master)
14         self.master = master
15         self.init_window()
16     def init_window(self):
17         self.master.title("人機界面GUI")
18         self.pack(fill=BOTH, expand=1)
19         menu = Menu(self.master)
20         self.master.config(menu=menu)
21         file = Menu(menu)
22         file.add_command(label="離開Exit", command=self.client_exit)
23         menu.add_cascade(label="檔案File", menu=file)
24         edit = Menu(menu)
25         edit.add_command(label="Undo")
26         menu.add_cascade(label="編輯Edit", menu=edit)
27     def client_exit(self):
28         exit()
29 root = Tk()
30 root.geometry("400x300")
31 app = Window(root)
32 root.mainloop()
```



14-1-6Tkinter影像圖片

- 使用選單載入圖片和文字

使用pip安裝pillow

- PIL為Python 影像函式庫
- from tkinter import *
- #安裝pillow
- from PIL import Image, ImageTk

```
60-250-191-81:~ justinwu$ pip install pillow
Requirement already satisfied: pillow in ./anaconda3/lib/python3.6/site-packages
Requirement already satisfied: olefile in ./anaconda3/lib/python3.6/site-package
s (from pillow)
```

建立Window類別

- 設定Window類別及__init__()初始建構函數

```
12 class Window(Frame):
13     def __init__(self, master=None):
14         Frame.__init__(self, master)
15         self.master = master
16         self.init_window()
17     def init_window(self):
18         self.master.title("人機界面GUI")
19         #使用包裝管理將widget視窗放到主視窗容器內
20         self.pack(fill=BOTH, expand=1)
21         menu = Menu(self.master)#建立選單實體
22         self.master.config(menu=menu)
23         file = Menu(menu)#建立檔案物件
24         #加入離開命令command到選單選項,當事件發生執行client_exit()函數
25         file.add_command(label="Exit離開", command=self.client_exit)
26         #將檔案加入到選單
27         menu.add_cascade(label="File檔案", menu=file)
28         edit = Menu(menu)#建立檔案物件
29         #加入顯示圖片命令command到選單選項,當事件發生執行showImg()函數
30         edit.add_command(label="顯示圖片", command=self.showImg)
31         #加入顯示圖片命令command到選單選項,當事件發生執行showText()函數
32         edit.add_command(label="顯示文字", command=self.showText)
33         #將檔案加入到選單
34         menu.add_cascade(label="編輯Edit", menu=edit)
```

顯示圖片文字

- 使用ImageTk.PhotoImage()來載入圖像
- img.place()放置圖片到特定位置
- text.pack()將文字打包管理放到容器中

```
35    def showImg(self):
36        load = Image.open("python.png")
37        render = ImageTk.PhotoImage(load)
38        #標籤可以是圖片或文字
39        img = Label(self, image=render)
40        img.image = render
41        img.place(x=0, y=0)
42    def showText(self):
43        text = Label(self, text="Hi 小文您好啊!")
44        text.pack()
45    def client_exit(self):
46        exit()
```

建立Tkinter介面實體

- 建立主視窗

  root = Tk()

- 設定視窗大小380*300

  root.geometry("380x300")

  app = Window(root)#建立視窗實體

  root.mainloop() #主迴圈mainloop

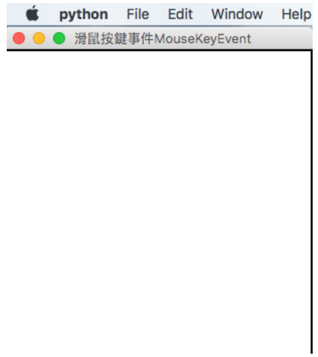
```
48 root = Tk()
49 root.geometry("380x300")
50 app = Window(root)
51 root.mainloop()
```

14-1-7Tkinter滑鼠鍵盤事件

- 一個例外物件也是一個事件，丟出例外就是啟動事件，傾聽者就是一個抓到區塊，它會抓到例外或事件。

- 當傾聽者接收各種事件時，傾聽物件會指派將發生什麼的方法。這些處理事件的方法稱為事件處理，而我們程式設計師就會定義這些方法。事件啟動物件(例如按紐)，和它的事件處裏傾聽者的關係。
- 元件啟動事件，傾聽者物件使用事件參數來呼叫事件處理函數。

按下滑鼠就會顯示資訊




```
鍵盤按鍵符號?  9
鍵盤按鍵碼keycode?  6029369
鍵盤輸入字元  9
鍵盤按鍵符號?  8
鍵盤按鍵碼keycode?  5963832
鍵盤輸入字元  8
鍵盤按鍵符號?  Shift_L
鍵盤按鍵碼keycode?  131074
鍵盤輸入字元
滑鼠座標 153 135
滑鼠在視窗位置 153 181
按下哪一個滑鼠按鈕  1
滑鼠座標 165 111
滑鼠在視窗位置 165 157
按下哪一個滑鼠按鈕  1
滑鼠座標 206 199
滑鼠在視窗位置 206 245
按下哪一個滑鼠按鈕  1
```

使用Canvas的bind()方法連接鍵盤及滑鼠

- mycanvas.pack()將畫布打包管理並且放到容器中
- <key>當按下某一個按鍵事件
- <Button-1>滑鼠左邊按鈕
- <Button-3>滑鼠右邊按鈕
- event.keycode從鍵盤輸入按鍵碼給按鍵事件
- event.x,event.y是事件的x,y軸座標位置

使用Canvas的bind()方法連接鍵盤及滑鼠

```python
from tkinter import *

class MouseKeyEvent:
    def __init__(self,window):
        window.title("滑鼠按鍵事件MouseKeyEvent")
        mycanvas = Canvas(window, bg = "white", width = 300, height = 300)
        mycanvas.pack()
        #和鍵盤按鍵聯繫
        mycanvas.bind("<Key>", self.KeyEvent)
        mycanvas.focus_set()
        #和滑鼠按鍵<Button-1>聯繫
        mycanvas.bind("<Button-1>", self.MouseEvent)
        window.mainloop()
    def KeyEvent(self, event):
        print("鍵盤按鍵符號? ", event.keysym)
        print("鍵盤按鍵碼keycode? ", event.keycode)
        print("鍵盤輸入字元 ", event.char)
    def MouseEvent(self, event):
        print("滑鼠座標", event.x, event.y)
        print("滑鼠在視窗位置", event.x_root, event.y_root)
        print("按下哪一個滑鼠按鈕 ", event.num)

mywindow = Tk()
MouseKeyEvent(mywindow)
```

13-2Pillow影像處理

- 使用pip安裝pillow
- RGB三顏色與像素
- 使用PIL套件的Image物件
- 使用Image圖片的crop()函數來切割圖片
- 複製圖片並且貼上圖片
- 調整圖片長寬
- 將本目錄的所有圖片都加上PNG.png圖片

使用pip安裝pillow

- PIL為Python 影像函式庫
- #安裝pillow

```
60-250-191-81:~ justinwu$ pip install pillow
Requirement already satisfied: pillow in ./anaconda3/lib/python3.6/site-packages
Requirement already satisfied: olefile in ./anaconda3/lib/python3.6/site-package
s (from pillow)
```

- from PIL import Image

RGB三顏色與像素

- 紅R綠G藍B
- 像素是由0-255的數值
- (255,255,255)由RGB組成白色

```
(255, 255, 255, 255)
(255, 0, 0, 255)
(255, 255, 255, 255)
(0, 128, 0, 255)
(255, 255, 255, 255)
(0, 0, 255, 255)
```

getpixel()是得到座標的像素

```
from PIL import Image,ImageColor

myIm=Image.new('RGBA',(100,100))
myIm.getpixel((0,0))
for i in range(100):
    for j in range(50):
        myIm.putpixel((i,j),(255,255,255))

for i in range(100):
    for j in range(50,100):
        myIm.putpixel((i,j),\
           ImageColor.getcolor('red','RGBA'))

print(myIm.getpixel((0,0)))
print(myIm.getpixel((0,50)))
myIm.save('Pixel_1.png')
```

putpixel()是設定座標位置的像素

```
26 myIm=Image.new('RGBA',(100,100))
27 myIm.getpixel((0,0))
28 for i in range(100):
29     for j in range(50):
30         myIm.putpixel((i,j),(255,255,255))
31
32 for i in range(100):
33     for j in range(50,100):
34         myIm.putpixel((i,j),\
35             ImageColor.getcolor('green','RGBA'))
36
37 print(myIm.getpixel((0,0)))
38 print(myIm.getpixel((0,50)))
39 myIm.save('Pixel_2.png')
```

ImageColor.getcolor()設定藍色

```
myIm=Image.new('RGBA',(100,100))
myIm.getpixel((0,0))
for i in range(100):
    for j in range(50):
        myIm.putpixel((i,j),(255,255,255))

for i in range(100):
    for j in range(50,100):
        myIm.putpixel((i,j),\
            ImageColor.getcolor('blue','RGBA'))

print(myIm.getpixel((0,0)))
print(myIm.getpixel((0,50)))
myIm.save('Pixel_3.png')
```

```
(255, 255, 255, 255)
(255, 0, 0, 255)
(255, 255, 255, 255)
(0, 128, 0, 255)
(255, 255, 255, 255)
(0, 0, 255, 255)
```

使用PIL套件的Image物件

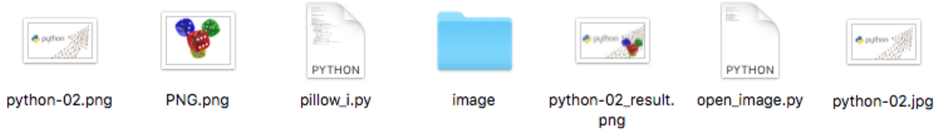
- 可以將png圖片轉存jpg檔

```
from PIL import Image
LOGO_F= 'python-02.png'
logoIm = Image.open(LOGO_F)
width,heigh=logoIm.size
print(width,heigh)
print(logoIm.filename)
print(logoIm.format)
print(logoIm.format_description)
print(logoIm.save('python-02.jpg'))
```

```
8859 4985
python-02.png
PNG
Portable network graphics
None
```

可以將png圖片轉存jpg檔

- 將python-02.png轉存成python-02.jpg



使用Image圖片的crop()函數來切割圖片

```
from PIL import Image
LOGO_F= 'python-02.png'
logoIm = Image.open(LOGO_F)
width,heigh=logoIm.size
print(width,heigh)
logoIm=logoIm.crop((5500,2000,6500,3000))
print(logoIm.save('mycrop.png'))
width,heigh=logoIm.size
print(width,heigh)
```

- 8859 4985
- None
- 1000 1000

複製圖片並且貼上圖片

使用logoIm.copy()拷貝圖片

- 使用logoIm.past()函數貼上圖片

```
from PIL import Image
LOGO_F= 'python-02.png'
logoIm = Image.open(LOGO_F)
myCopy = logoIm.copy()
width,heigh=logoIm.size
print(width,heigh)
logoIm=logoIm.crop((6000,2000,7000,3000))
myCopy.paste(logoIm,(100,100))
myCopy.paste(logoIm,(2000,2000))
myCopy.save('new_fixed.png')
```

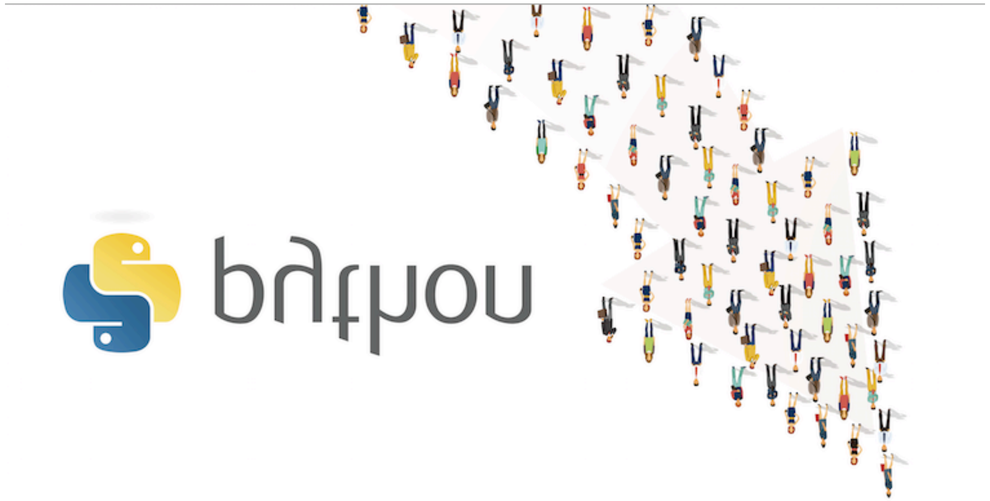
調整圖片長寬

```python
from PIL import Image
LOGO_F= 'python-02.png'
logoIm = Image.open(LOGO_F)
print(logoIm.size)
width,heigh=logoIm.size
logoIm_1=logoIm.resize((int(width/4),int(heigh/4)))
print(logoIm_1.size)
logoIm_1.save('new_half_fixed.png')
logoIm_2=logoIm.resize((width+2000,heigh))
print(logoIm_2.size)
logoIm_2.save('new_width.png')
logoIm.rotate(180).save('new_rotate_180.png')
logoIm.transpose(Image.FLIP_TOP_BOTTOM).save('new_hori_flip.png')
```

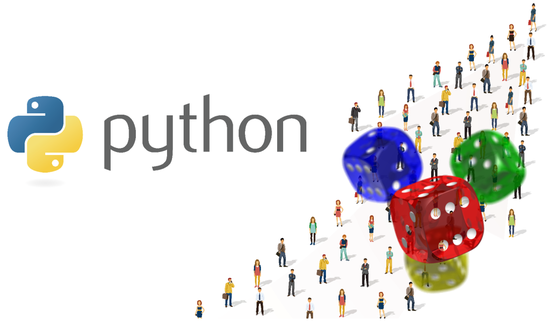
```
(8859, 4985)
(2214, 1246)
(10859, 4985)
```

將圖片旋轉180度

將圖片水平上下翻轉



將本目錄的所有圖片都加上PNG.png圖片




```
更新圖片大小 python-02.png...
將圖片加入LOGO_F python-02.png...
```

將本目錄的所有圖片都加上PNG.png圖片

- import os
- from PIL import Image
- S_SIZE = 1800
- LOGO_F= 'PNG.png'
- logoIm = Image.open(LOGO_F)
- logoWidth, logoHeight = logoIm.size
- #新增image目錄
- os.makedirs('image', exist_ok=True)

搜尋目前目錄的所有檔案
選取所有png檔和jpg檔或檔名為LOGO_F

```
#搜尋目前目錄的所有檔案#選取所有png檔和jpg檔或檔名為LOGO_F
for filename in os.listdir('.'):
    if not (filename.endswith('.png') or filename.endswith('.jpg')) \
       or filename == LOGO_F:
        continue
    im = Image.open(filename)#打開LOGO_F檔案
    width, height = im.size
    #檢查影像大小是否要被修正
    if width > S_SIZE and height > S_SIZE:
        if width > height:
            height = int((S_SIZE / width) * height)
            width = S_SIZE
        else:
            width = int((S_SIZE / height) * width)
            height = S_SIZE
        # 更新圖片大小
        print('更新圖片大小 %s...' % (filename))
        im = im.resize((width, height))
    # 將圖片加入LOGO_F.
    print('將圖片加入LOGO_F %s...' % (filename))
    im.paste(logoIm, (width - logoWidth, height - logoHeight), logoIm)
    # 儲存圖片.
    im.save(os.path.join('image', filename))
```

- Thanks.

Python 精通程式語言
Python遊戲設計

吳佳諺 老師

9.使用matplotlib畫圖
10.網站擷取使用Python
11.行程和執行緒
12.資料及檔案處理
13.SQLite資料庫
14.人機界面及影像處理
15.Python遊戲設計

Python遊戲設計

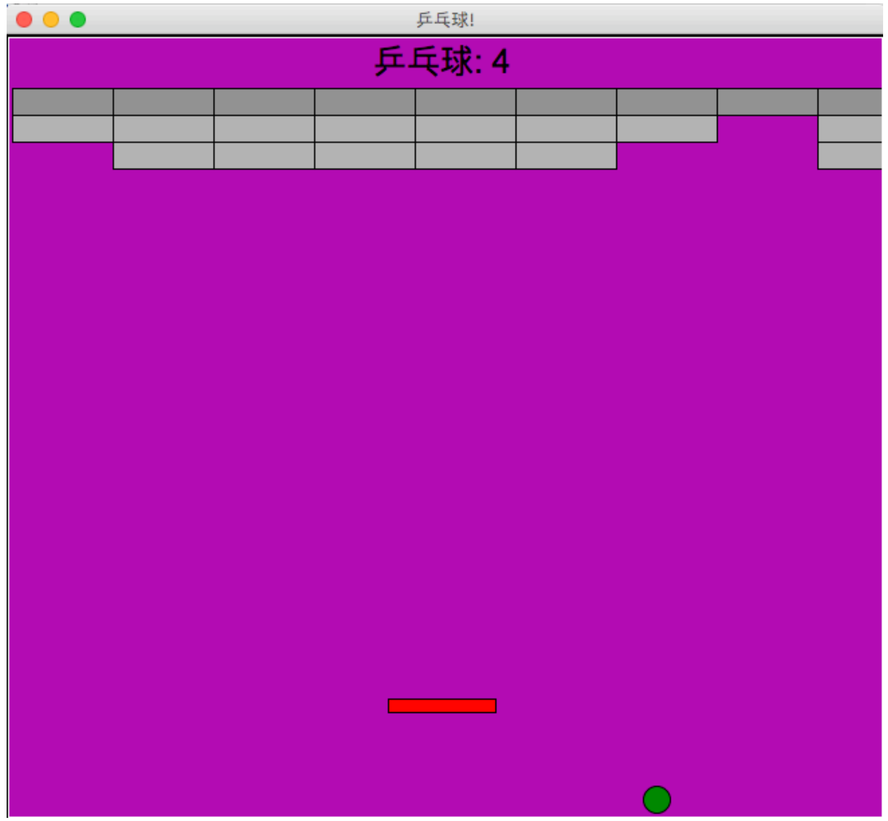
- 15-1乒乓球遊戲
- 15-2人機界面Tkinter
- 15-3乒乓球遊戲實作

15-1乒乓球遊戲

- 遊戲開始按下空白鍵
- 乒乓球掉到底下,乒乓球減少一顆
- 乒乓球撞到灰色磚塊可以消滅磚塊
- 球撞到黑色磚塊會變灰色磚塊
- 磚塊全部撞完,你贏囉
- 遊戲結束囉

遊戲開始按下空白鍵

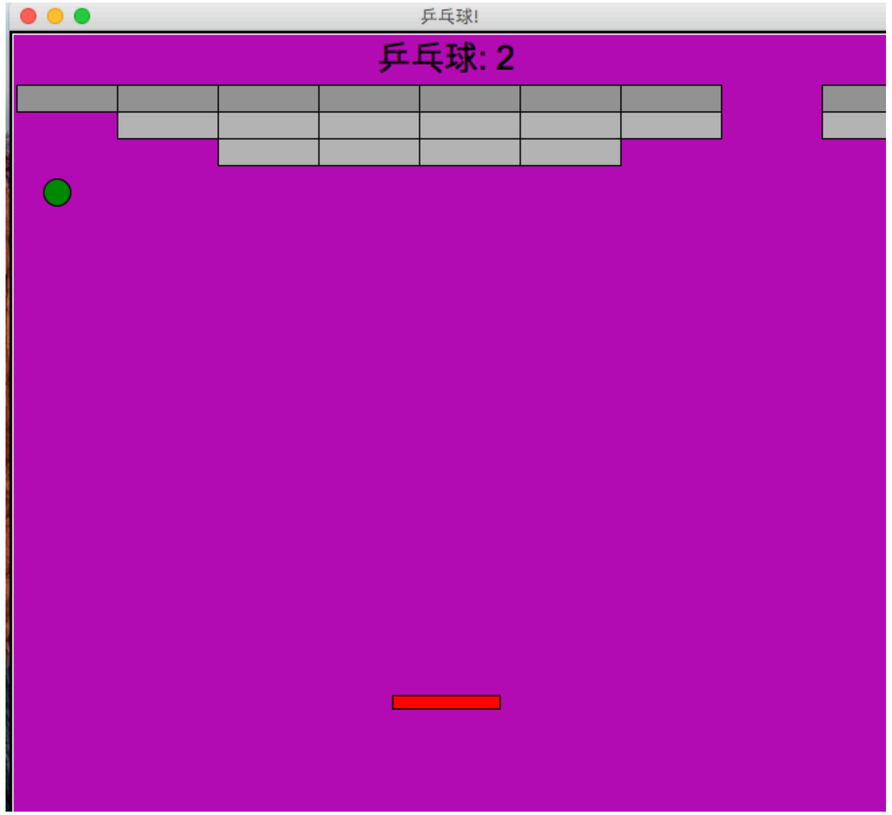
乒乓球掉到底下,乒乓球減少一顆



剩下乒乓球3顆

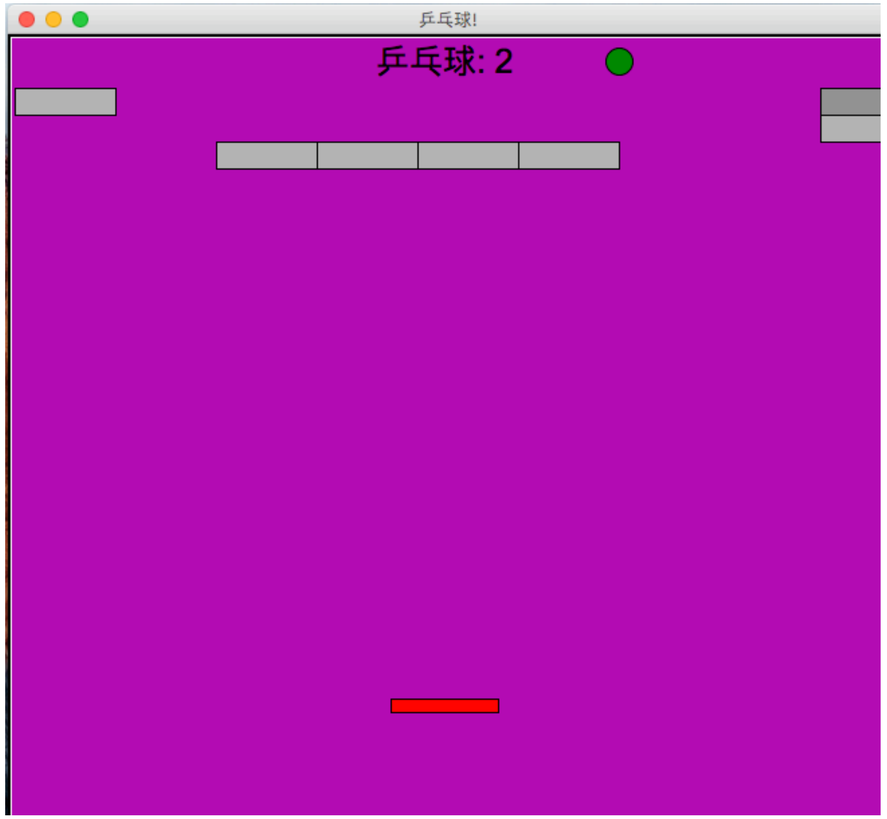
剩下乒乓球2顆

乒乓球撞到灰色磚塊可以消滅磚塊



球撞到黑色磚塊會變灰色磚塊

- 上面灰黑色磚塊撞到,會變灰色



剩下乒乓球1顆

剩下乒乓球0顆

磚塊全部撞完,你贏囉

遊戲結束囉

- 當乒乓球剩下0顆時,遊戲結束囉

15-2人機界面Tkinter

- 在圖形化介面，一個事件代表了一些動作，例如按下空白鍵按鈕，在鍵盤上按下左右鍵按鈕，在視窗上按下關掉視窗的按鈕，或者是任何期望引發回應的動作。

15-2-1事件驅動程式設計

- 事件驅動程式設計使用告知和回應的方法來作程式設計。
- 告知物件被稱為事件(event)。
- 使用事件和事件處理。
- 一個事件(event)就是一個物件，它作用在另外一個稱為傾聽的物件上。
- 事件的傳送稱為啟動事件。
- Canvas.bind()函數為註冊傾聽的函數物件.

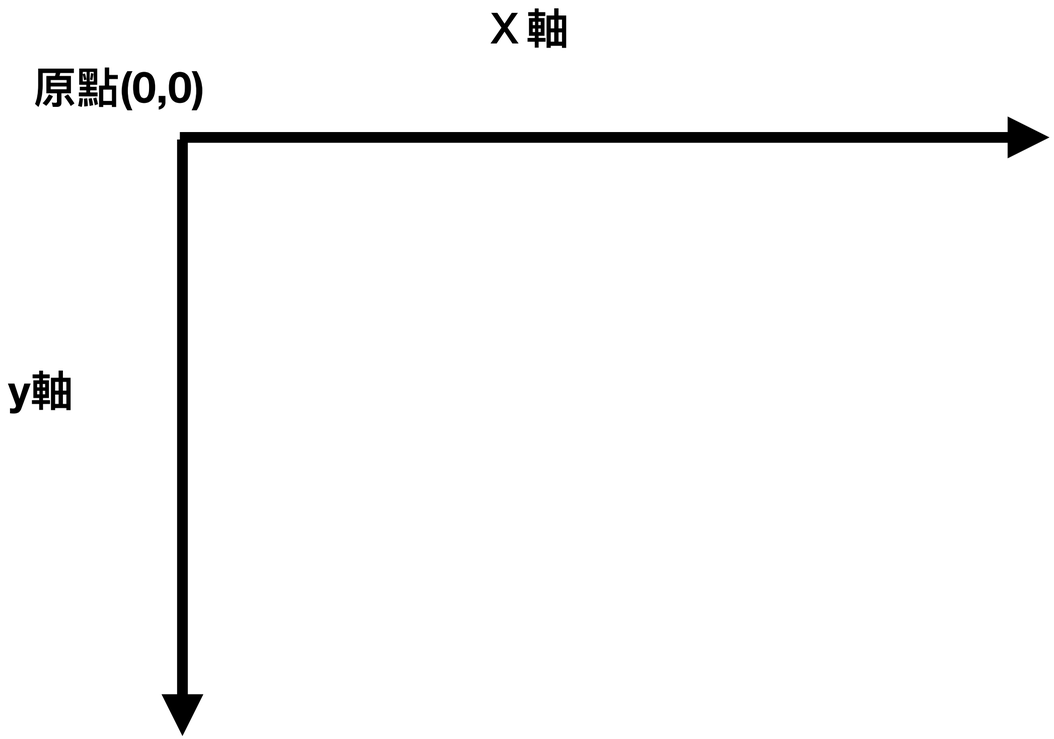
```
self.canvas.bind('<Left>',lambda _: self.paddle.move(-30))
self.canvas.bind('<Right>',lambda _: self.paddle.move(30))
```

```
self.canvas.bind('<space>', lambda _: self.start_game())
```

傾聽的函數物件start_game()

```
def start_game(self):
    self.canvas.unbind('<space>')
    self.canvas.delete(self.text)
    self.paddle.ball = None
    self.game_loop()
```

15-2-2座標軸



使用tkinter套件

- 從tkinter人機界面套件輸入所有模組
- from tkinter import *
- #建立最上層視窗
- master = Tk()

python

- #從tkinter人機界面套件輸入所有模組
- from tkinter import *
- class MyGame(Frame):#繼承自父類別Frame
- #初始化視窗
- master = Tk()
- master.title('乒乓球!')
- super(MyGame, self).__init__(master)
- #建立最上層視窗
- master = Tk()
- #建立視窗實體
- game = MyGame()

15-3乒乓球遊戲實作

- MyGameObject類別
- MyBall繼承自MyGameObject類別
- MyPaddle類別繼承自MyGameObject類別
- MyBrick類別繼承自MyGameObject類別
- MyGame類別繼承自Frame類別
- 開始遊戲

MyGameObject類別

```
 9 from tkinter import *
10
11 class MyGameObject(object):
12     def __init__(self, mycanvas, item):
13         self.canvas = mycanvas
14         self.item = item
15
16     def position(self):
17         return self.canvas.coords(self.item)
18
19     def move(self, c_x, c_y):
20         self.canvas.move(self.item, c_x, c_y)
21
22     def delete(self):
23         self.canvas.delete(self.item)
```

MyBall繼承自MyGameObject類別

```
26 #MyBall繼承自MyGameObject類別
27 class MyBall(MyGameObject):
28     def __init__(self, canvas, x, y):
29         self.radius = 10
30         self.direction = [1, -1]
31         self.speed = 10
32         item = canvas.create_oval(x-self.radius, y-self.radius,
33                                   x+self.radius, y+self.radius,
34                                   fill='green')
35         super(MyBall, self).__init__(canvas, item)
36 
37     def ball_update(self):
38         coords = self.position()
39         width = self.canvas.winfo_width()
40         if coords[0] <= 0 or coords[2] >= width:
41             self.direction[0] *= -1
42         if coords[1] <= 0:
43             self.direction[1] *= -1
44         x = self.direction[0] * self.speed
45         y = self.direction[1] * self.speed
46         self.move(x, y)
```

MyGameObject類別

```
48      def ball_collide(self, game_objects):
49          coords = self.position()
50          x = (coords[0] + coords[2]) * 0.5
51          if len(game_objects) > 1:
52              self.direction[1] *= -1
53          elif len(game_objects) == 1:
54              game_object = game_objects[0]
55              coords = game_object.position()
56              if x > coords[2]:
57                  self.direction[0] = 1
58              elif x < coords[0]:
59                  self.direction[0] = -1
60              else:
61                  self.direction[1] *= -1
62
63          for game_object in game_objects:
64              if isinstance(game_object, MyBrick):
65                  game_object.hit()
```

MyPaddle類別繼承自MyGameObject類別

```
68 class MyPaddle(MyGameObject):
69     def __init__(self, canvas, x, y):
70         self.width = 80
71         self.height = 10
72         self.ball = None
73         item = canvas.create_rectangle(x - self.width / 2,
74                                        y - self.height / 2,
75                                        x + self.width / 2,
76                                        y + self.height / 2,
77                                        fill='red')
78         super(MyPaddle, self).__init__(canvas, item)
79
80     def set_ball(self, ball):
81         self.ball = ball
82
83     def move(self, offset):
84         coords = self.position()
85         width = self.canvas.winfo_width()
86         if coords[0] + offset >= 0 and coords[2] + \
87                                 offset <= width:
88             super(MyPaddle, self).move(offset, 0)
89             if self.ball is not None:
90                 self.ball.move(offset, 0)
```

MyBrick類別繼承自MyGameObject類別

```
93 class MyBrick(MyGameObject):
94     COLORS = {1: '#aaaaaa', 2: '#888888', 3: '#000000'}
95
96     def __init__(self, canvas, x, y, hits):
97         self.width = 75
98         self.height = 20
99         self.hits = hits
100        color = MyBrick.COLORS[hits]
101        item = canvas.create_rectangle(x - self.width / 2,
102                                       y - self.height / 2,
103                                       x + self.width / 2,
104                                       y + self.height / 2,
105                                       fill=color, tags='brick')
106        super(MyBrick, self).__init__(canvas, item)
107
108    def hit(self):
109        self.hits -= 1
110        if self.hits == 0:
111            self.delete()
112        else:
113            self.canvas.itemconfig(self.item,
114                                   fill=MyBrick.COLORS[self.hits])
```

MyGame類別繼承自Frame類別

```
116 class MyGame(Frame):
117     def __init__(self):
118         master = Tk()
119         master.title('乒乓球!')
120         super(MyGame, self).__init__(master)
121         self.lives = 5
122         self.width = 650
123         self.height = 580
124         self.canvas = Canvas(self, bg='#aa11aa',
125                              width=self.width,
126                              height=self.height,)
127         self.canvas.pack()
128         self.pack()
129         self.items = {}
130         self.ball = None
131         #將Paddle()物件分配給paddle成員屬性
132         self.paddle = MyPaddle(self.canvas, self.width/2, 500)
133         self.items[self.paddle.item] = self.paddle
134         for x in range(5, self.width - 5, 75):
135             self.addBrick(x + 37.5, 50, 2)
136             self.addBrick(x + 37.5, 70, 1)
137             self.addBrick(x + 37.5, 90, 1)
138         self.hud = None
139         self.setupGame()
140         self.canvas.focus_set()
141         self.canvas.bind('<Left>',lambda _: self.paddle.move(-30))
142         self.canvas.bind('<Right>',lambda _: self.paddle.move(30))
```

MyGame類別

```
145     def setupGame(self):
146             self.addBall()
147             self.ball_update_lives_text()
148             self.text = self.draw_text(300, 200,
149                                        '請按下空白鍵開始')
150             self.canvas.bind('<space>', lambda _: self.start_game())
151
152     def addBall(self):
153         if self.ball is not None:
154             self.ball.delete()
155         paddle_coords = self.paddle.position()
156         x = (paddle_coords[0] + paddle_coords[2]) * 0.5
157         #新增MyBall()物件,並且將它分配給成員屬性ball
158         self.ball = MyBall(self.canvas, x, 310)
159         self.paddle.set_ball(self.ball)
160
161     def addBrick(self, x, y, hits):
162         brick = MyBrick(self.canvas, x, y, hits)
163         self.items[brick.item] = brick
```

MyGame類別

```
165     def draw_text(self, x, y, text, size='55'):
166         font = ('Arial', size)
167         return self.canvas.create_text(x, y, text=text,
168                                        font=font)
169
170     def ball_update_lives_text(self):
171         text = '乒乓球: %s' % self.lives
172         if self.hud is None:
173             self.hud = self.draw_text(325, 20, text, 25)
174         else:
175             self.canvas.itemconfig(self.hud, text=text)
176
177     def start_game(self):
178         self.canvas.unbind('<space>')
179         self.canvas.delete(self.text)
180         self.paddle.ball = None
181         self.game_loop()
```

MyGame類別

```
183     def game_loop(self):
184         self.checkCollisions()
185         num_bricks = len(self.canvas.find_withtag('brick'))
186         if num_bricks == 0:
187             self.ball.speed = None
188             self.draw_text(300, 200, '你贏囉!')
189         elif self.ball.position()[3] >= self.height:
190             self.ball.speed = None
191             self.lives -= 1
192             if self.lives < 0:
193                 self.draw_text(300, 200, '遊戲結束囉')
194             else:
195                 self.after(1000, self.setupGame)
196         else:
197             self.ball.ball_update()
198             self.after(50, self.game_loop)
199
200     def checkCollisions(self):
201         ball_coords = self.ball.position()
202         items = self.canvas.find_overlapping(*ball_coords)
203         objects = [self.items[x] for x in items if x in self.items]
204         self.ball.ball_collide(objects)
```

50ms後再次執行
game_loop()函數

開始遊戲

```
if __name__ == '__main__':
    game = MyGame()
    game.mainloop()
```

15-4安裝pygame

- 使用pip install pygame來安裝pygame

```
60-250-191-81:~ justinwu$ pip install pygame
Requirement already satisfied: pygame in ./anaconda3/lib/python3.6/site-packages
```

15-5聲音

- pygame使用mixer模組來播放聲音
- music類別可以播放較長的聲音
- 初始化建構聲音init(),load()載入聲音,set_volume()可以設定聲音大小(0到1)

```
111 class MyGame(Frame):
112     def __init__(self):
113         master = Tk()
114         master.title('乒乓球!')
115         super(MyGame, self).__init__(master)
116         pyg.mixer.init()
117         pyg.mixer.music.load('By_the_Pool.mp3')
118         pyg.mixer.music.set_volume(0.28)
```

使用pyg.mixer.music.play()

來播放音樂

- 使用pyg.mixer.music.play()來播放音樂

```
def start_game(self):
    self.canvas.unbind('<space>')
    self.canvas.delete(self.text)
    self.paddle.ball = None
    pyg.mixer.music.play()
    self.game_loop()
```

- Thanks.