台灣人工智慧學校

Python 快速上手 part1

William

資料與程式碼: 程式碼與練習題解答
影片播放列表: 影片播放列表
投影片 PDF : 投影片PDF下載連結

「版權聲明頁」

why python

Why Python?

- 深度學習的框架幾乎都支援 Python

Caffe

Caffe2

TensorFlow

theano

Microsoft CNTK

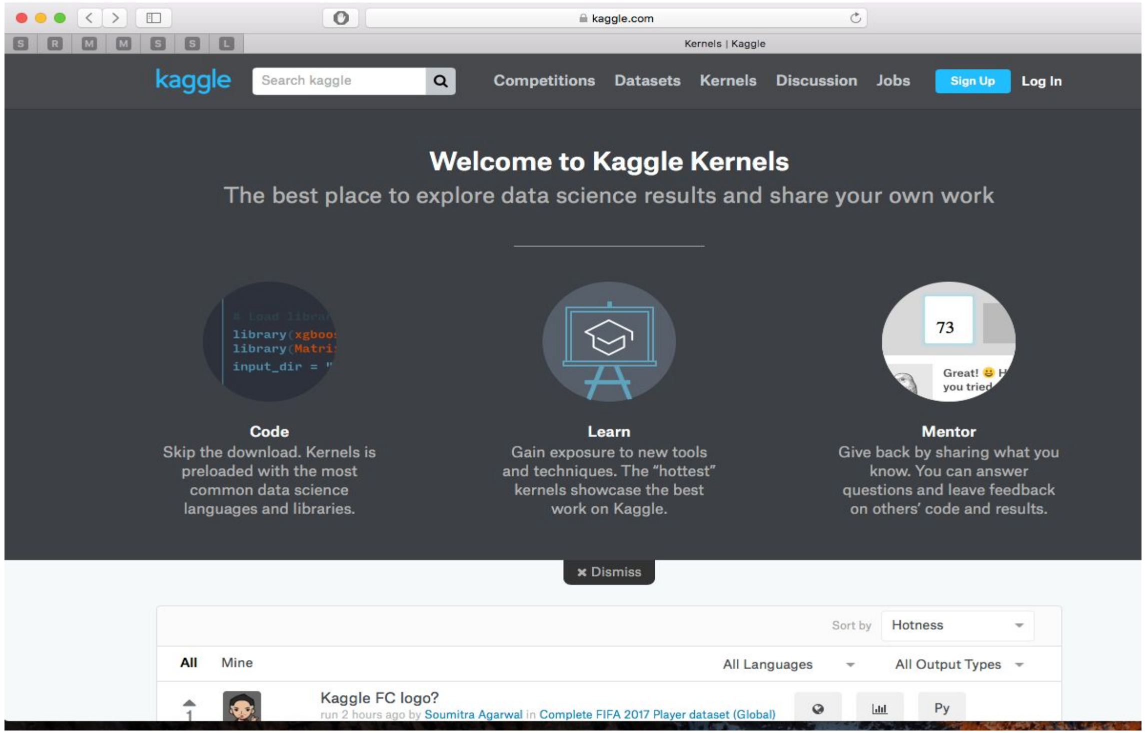
dmlc mxnet

K

PYTORCH

Why Python?

- 資料科學中的主流語言



Welcome to Kaggle Kernels

The best place to explore data science results and share your own work

Code
Skip the download. Kernels is preloaded with the most common data science languages and libraries.

Learn
Gain exposure to new tools and techniques. The "hottest" kernels showcase the best work on Kaggle.

Mentor
Give back by sharing what you know. You can answer questions and leave feedback on others' code and results.

Kaggle FC logo?
run 2 hours ago by Soumitra Agarwal in Complete FIFA 2017 Player dataset (Global)

Why Python?

- 深度學習的框架幾乎都支援 Python

Caffe

Caffe2

TensorFlow

theano

Microsoft CNTK

dmlc mxnet

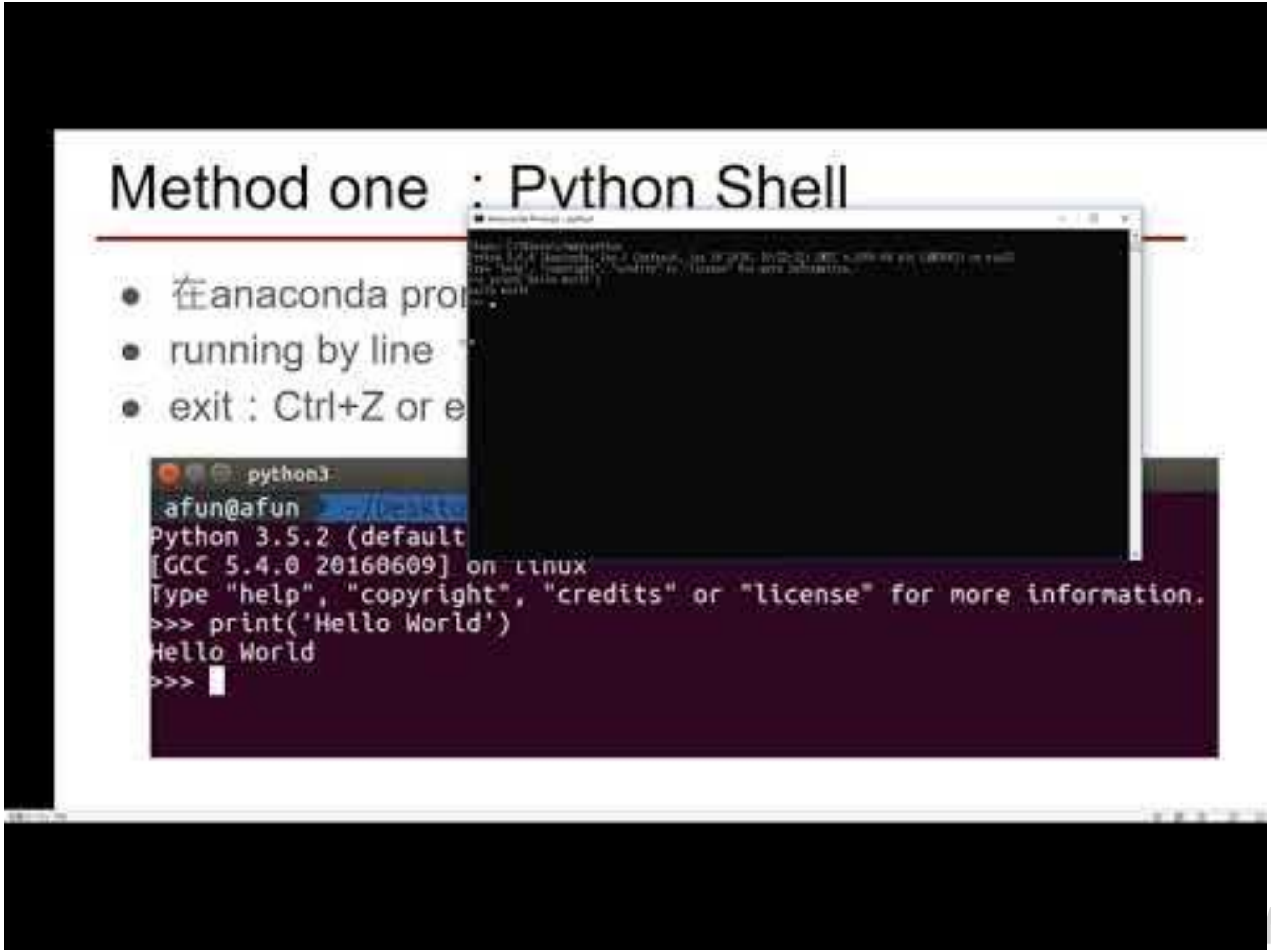
K

PYTORCH

Why Python?

- 簡單好學!
- Hello world in Python
- print("Hello world")

本機端環境建置



Method one ：Python Shell

- 在anaconda pro
- running by line
- exit：Ctrl+Z or e

```
python3
afun@afun
Python 3.5.2 (default
[GCC 5.4.0 20160609] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello World')
Hello World
>>> 
```

Before installing ...

- Highly recommend learning Python 3.x
  - Different syntax
  - Different implementation
  - No more support for Python 2.7

Anaconda

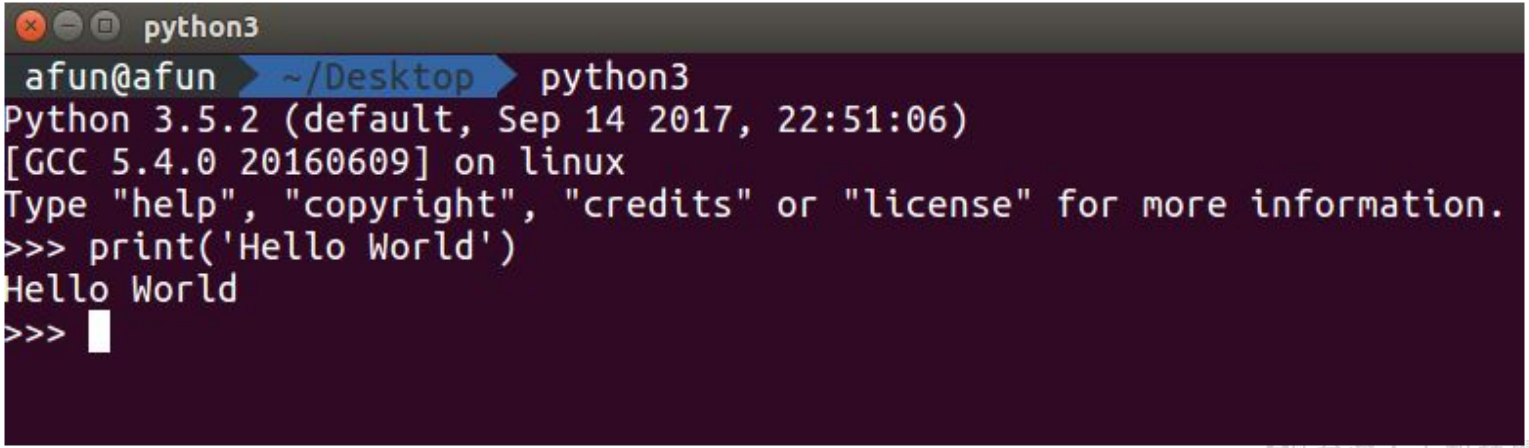
- 除了 Python，許多資料分析常用的套件也都包含在內
- Windows / Linux / Mac OS
- Download
  - 目前python 3.7尚有許多問題，建議下載Anaconda3 5.2版(python 3.6)
- Anaconda Prompt
- conda install

To use Python in Anaconda, there are three methods ...

- Python Shell
- Ipython
- jupyter notebook

Method one ：Python Shell

- 在anaconda prompt 輸入python
- running by line
- exit：Ctrl+Z or exit()

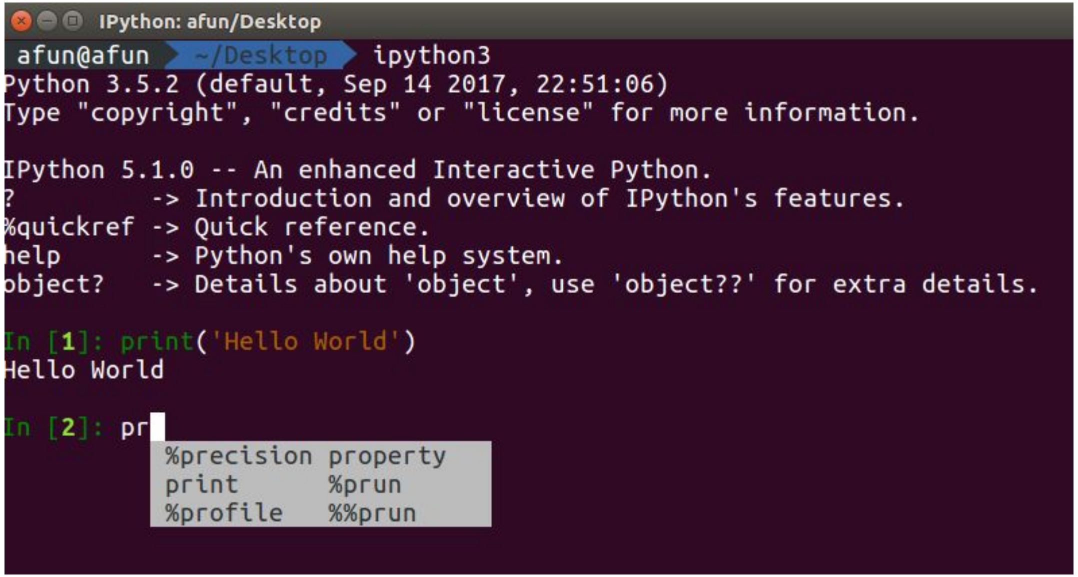


```
afun@afun  ~/Desktop  python3
Python 3.5.2 (default, Sep 14 2017, 22:51:06)
[GCC 5.4.0 20160609] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello World')
Hello World
>>> 
```

Method two：ipython

- $ pip install ipython
- 在anaconda prompt 輸入ipython
- include magic code
- running by line
- TAB for hint

- exit：exit



```
afun@afun  ~/Desktop  ipython3
Python 3.5.2 (default, Sep 14 2017, 22:51:06)
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]: print('Hello World')
Hello World

In [2]: pr
          %precision property
          print      %prun
          %profile   %%prun
```

jupyter notebook

Introduction to Jupyter notebook

- Code is divided into cells to control execution

- Ideal for exploratory analysis and model building

Download the Image

[illegible]

```
[illegible]
```

但對於許多人來說...

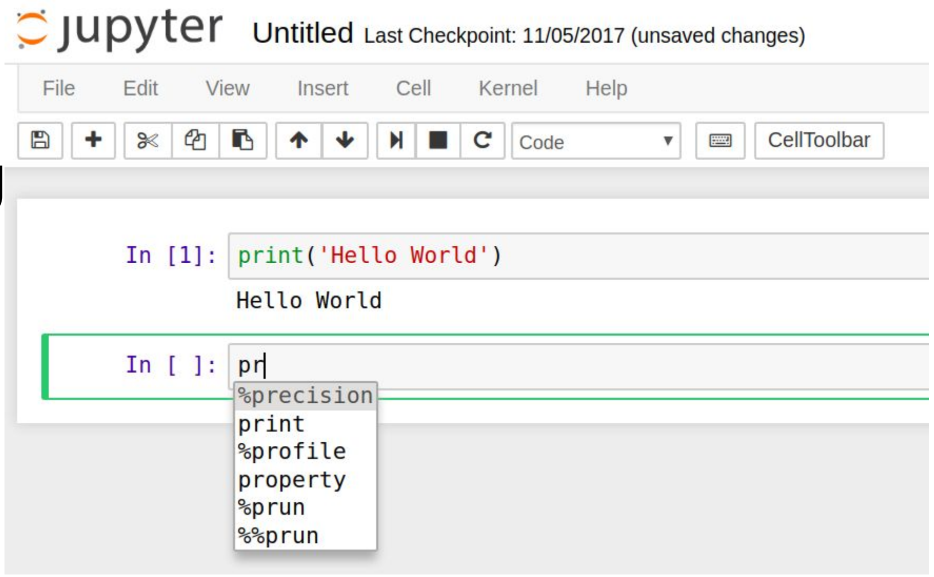
- 一看到命令字元介面...

沒有按鈕...沒有游標...滑鼠不能用...

吓到吃手手

Method three：jupyter notebook

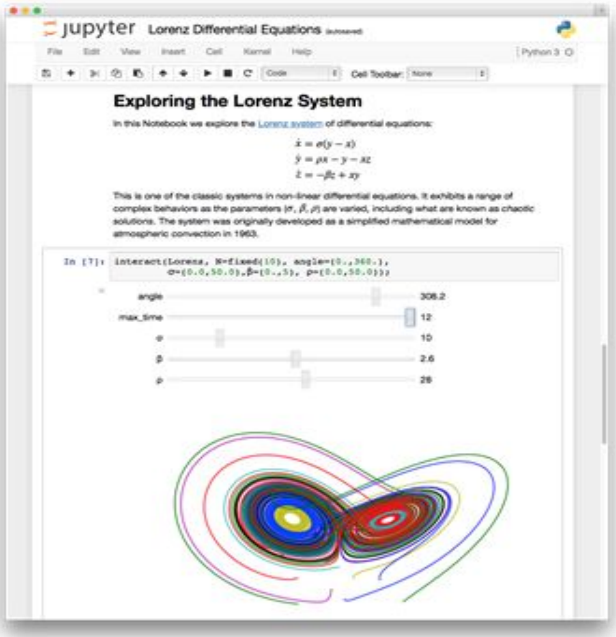
- $ pip install jupyter
- include magic code
- running by cell
- TAB for autocomplete
- SHIFT+TAB for docstring



```
In [1]: print('Hello World')

        Hello World

In [ ]: pr
        %precision
        print
        %profile
        property
        %prun
        %%prun
```

Introduction to Jupyter notebook

- Jupyter is an anagram of: Julia, Python, and R
- Supports multiple content types: code, narrative text, images, movies, etc



Exploring the Lorenz System

In this Notebook we explore the Lorenz system of differential equations:

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = \rho x - y - xz$$
$$\dot{z} = -\beta z + xy$$

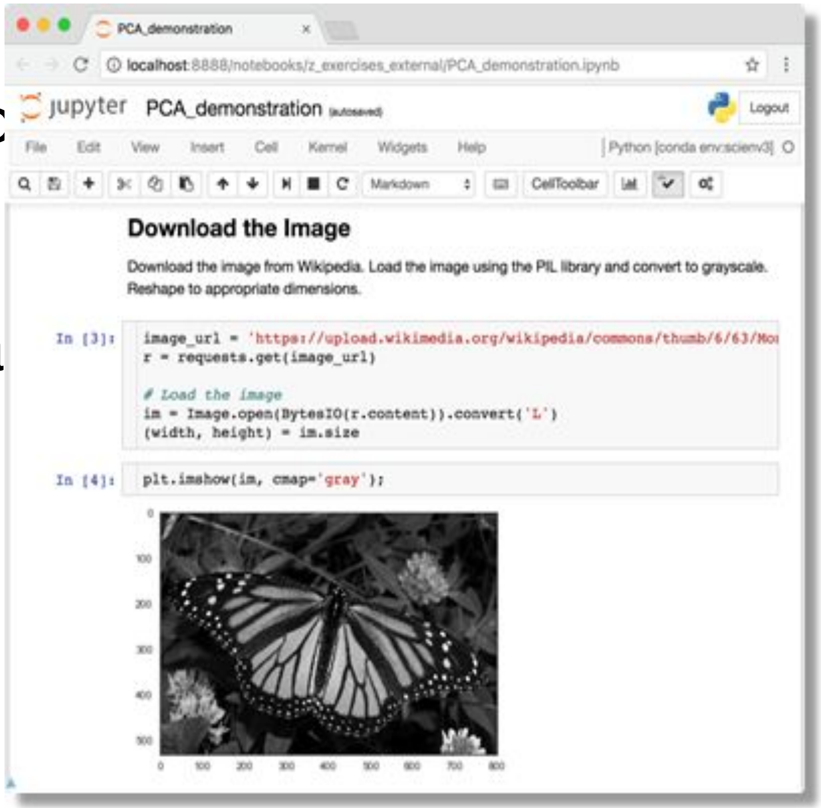
This is one of the classic systems in non-linear differential equations. It exhibits a range of complex behaviors as the parameters (σ, β, ρ) are varied, including what are known as chaotic solutions. The system was originally developed as a simplified mathematical model for atmospheric convection in 1963.

```
In [7]: interact(Lorenz, N=fixed(10), angle=(0.,360.),
                 σ=(0.0,50.0),β=(0.,5), ρ=(0.0,50.0));
```

angle	308.2
max_time	12
σ	10
β	2.6
ρ	28

Introduction to Jupyter notebook

- Code is divided into cells to cc
- Ideal for exploratory analysis a building


PCA_demonstration
localhost:8888/notebooks/z_exercises_external/PCA_demonstration.ipynb
jupyter PCA_demonstration (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Python [conda env:scienv3]
Markdown
CellToolbar

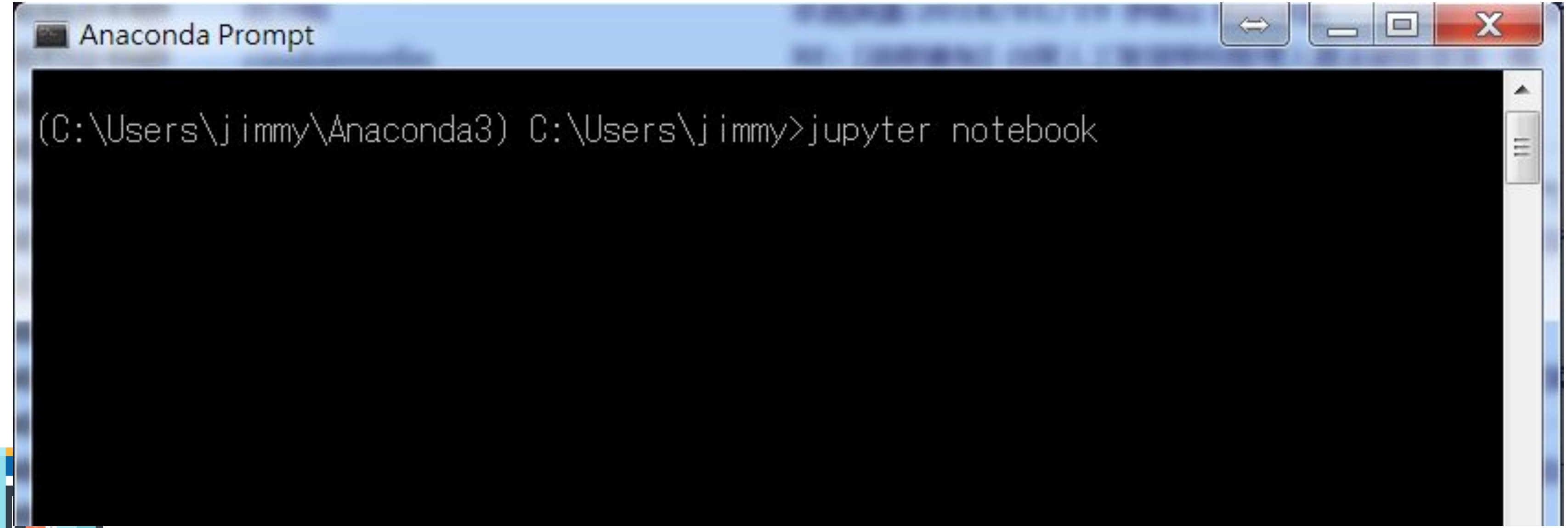
Download the Image

Download the image from Wikipedia. Load the image using the PIL library and convert to grayscale. Reshape to appropriate dimensions.

```
In [3]: image_url = 'https://upload.wikimedia.org/wikipedia/commons/thumb/6/63/Mo
        r = requests.get(image_url)

        # Load the image
        im = Image.open(BytesIO(r.content)).convert('L')
        (width, height) = im.size

In [4]: plt.imshow(im, cmap='gray');
```

如何開啟 Jupyter notebook (本機)

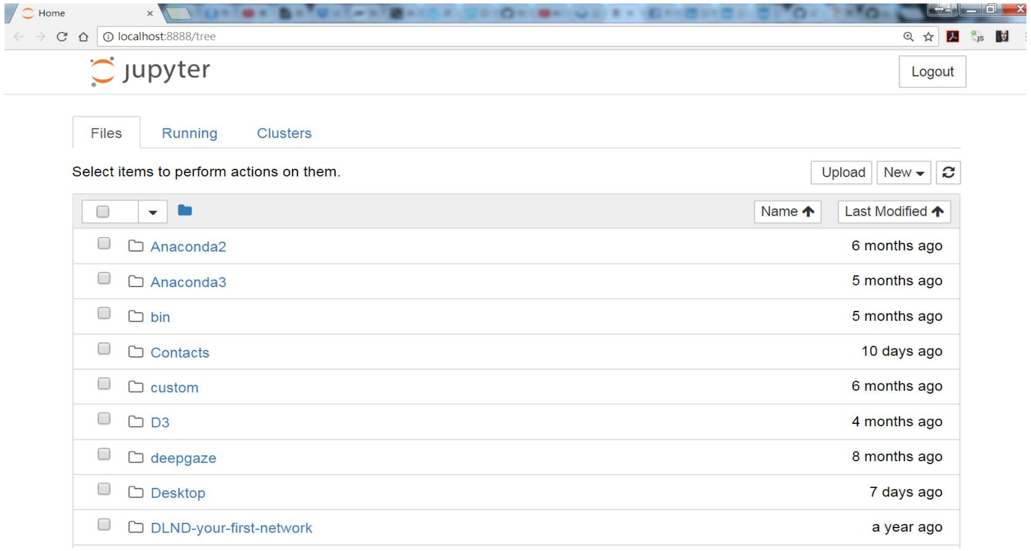
- 安裝好 Anaconda 後，請打開 Anaconda Prompt，並輸入 jupyter notebook



```
(C:\Users\jimmy\Anaconda3) C:\Users\jimmy>jupyter notebook
```

順利開啟！

- 會自動在您預設的瀏覽器中打開 Jupyter notebook
- 顯示的資料則是 terminal 輸入的當前路徑 (預設使用者名稱)



Home

localhost:8888/tree

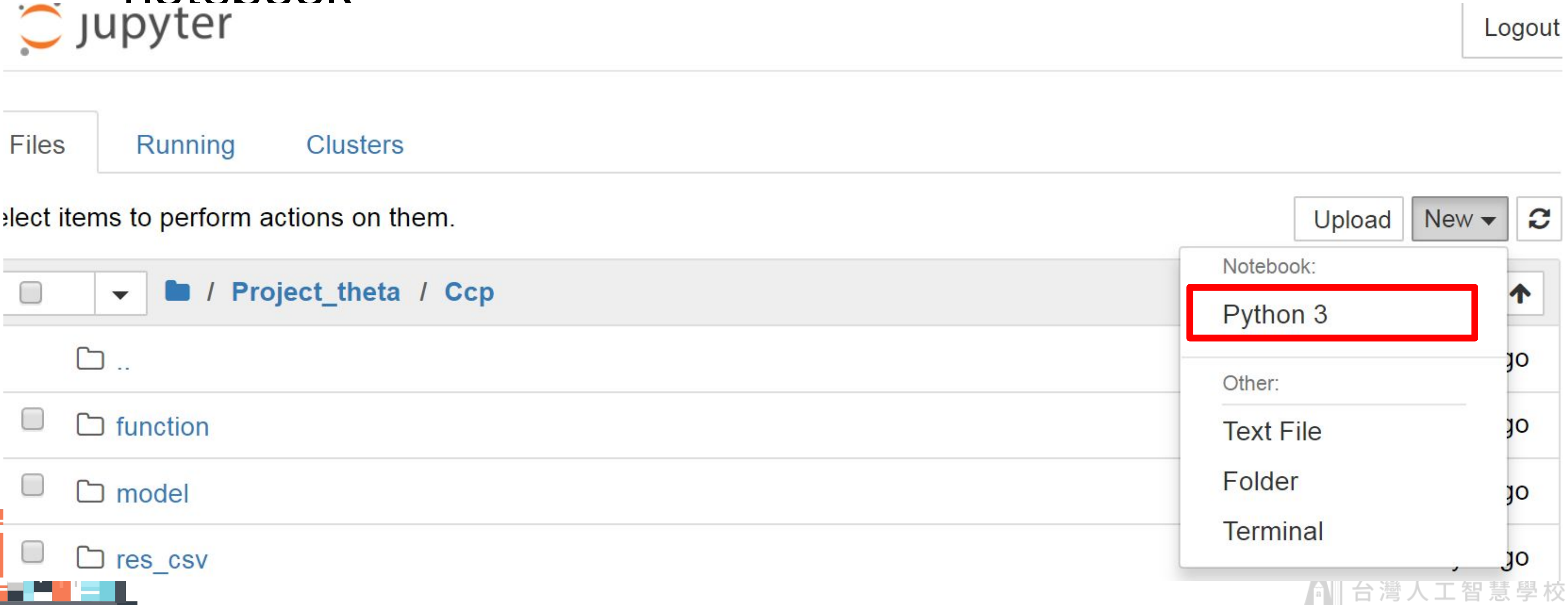
Files Running Clusters

Select items to perform actions on them.

Name	Last Modified
Anaconda2	6 months ago
Anaconda3	5 months ago
bin	5 months ago
Contacts	10 days ago
custom	6 months ago
D3	4 months ago
deepgaze	8 months ago
Desktop	7 days ago
DLND-your-first-network	a year ago

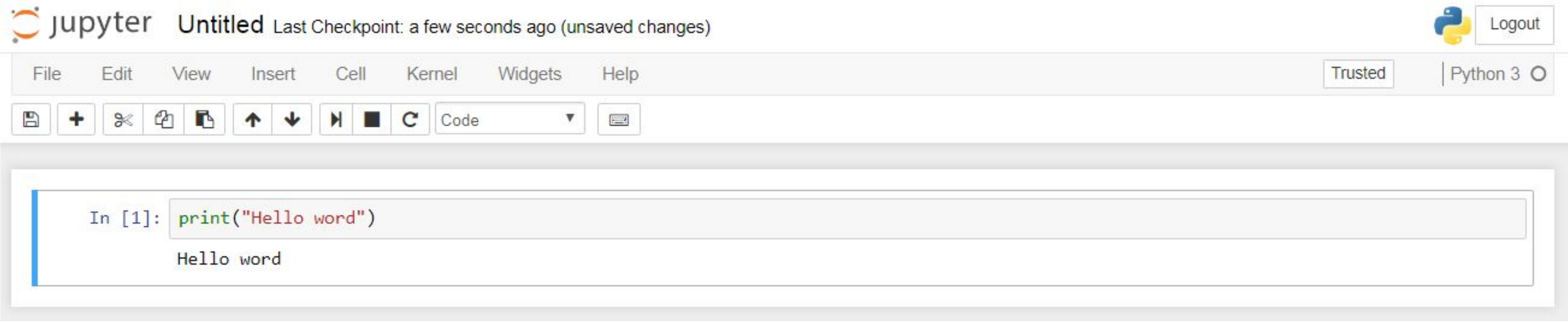
開啟 notebook

- 在 New 鍵下，選擇 Python 3，即可開啟新的 notebook



開啟 notebook

- 完成！可以開始輸入 code 囉

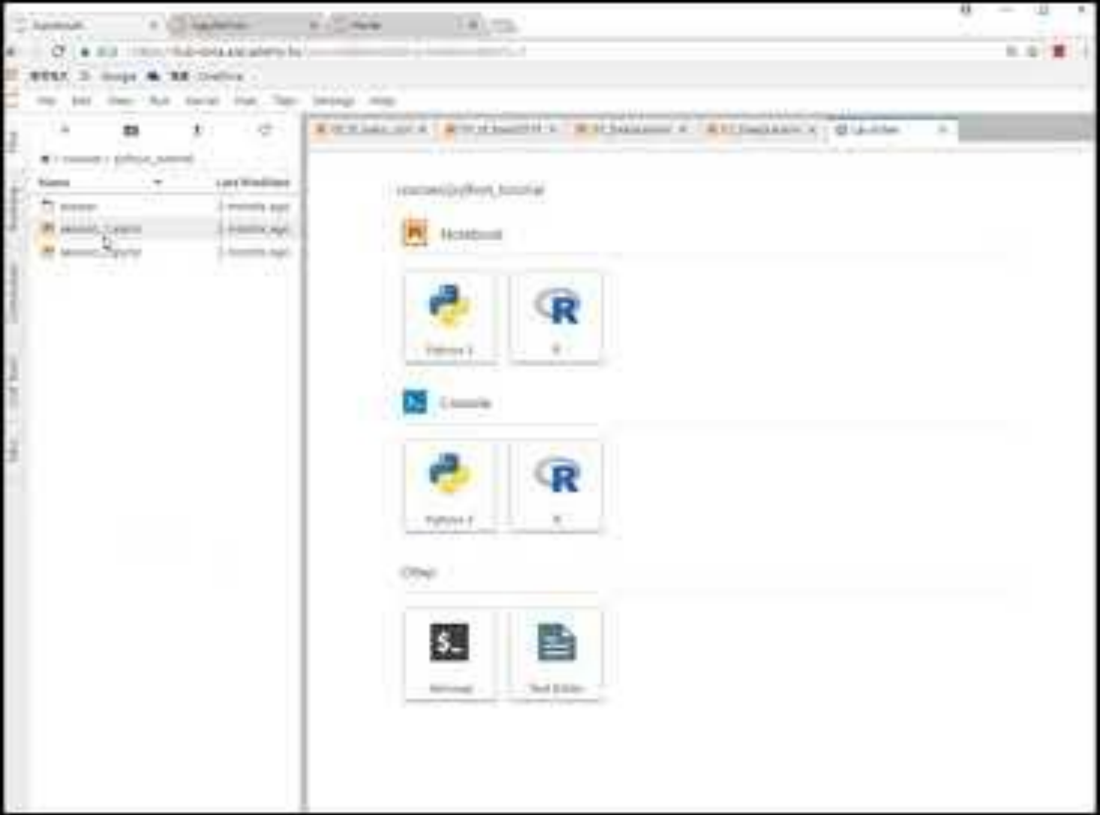


Untitled Last Checkpoint: a few seconds ago (unsaved changes)

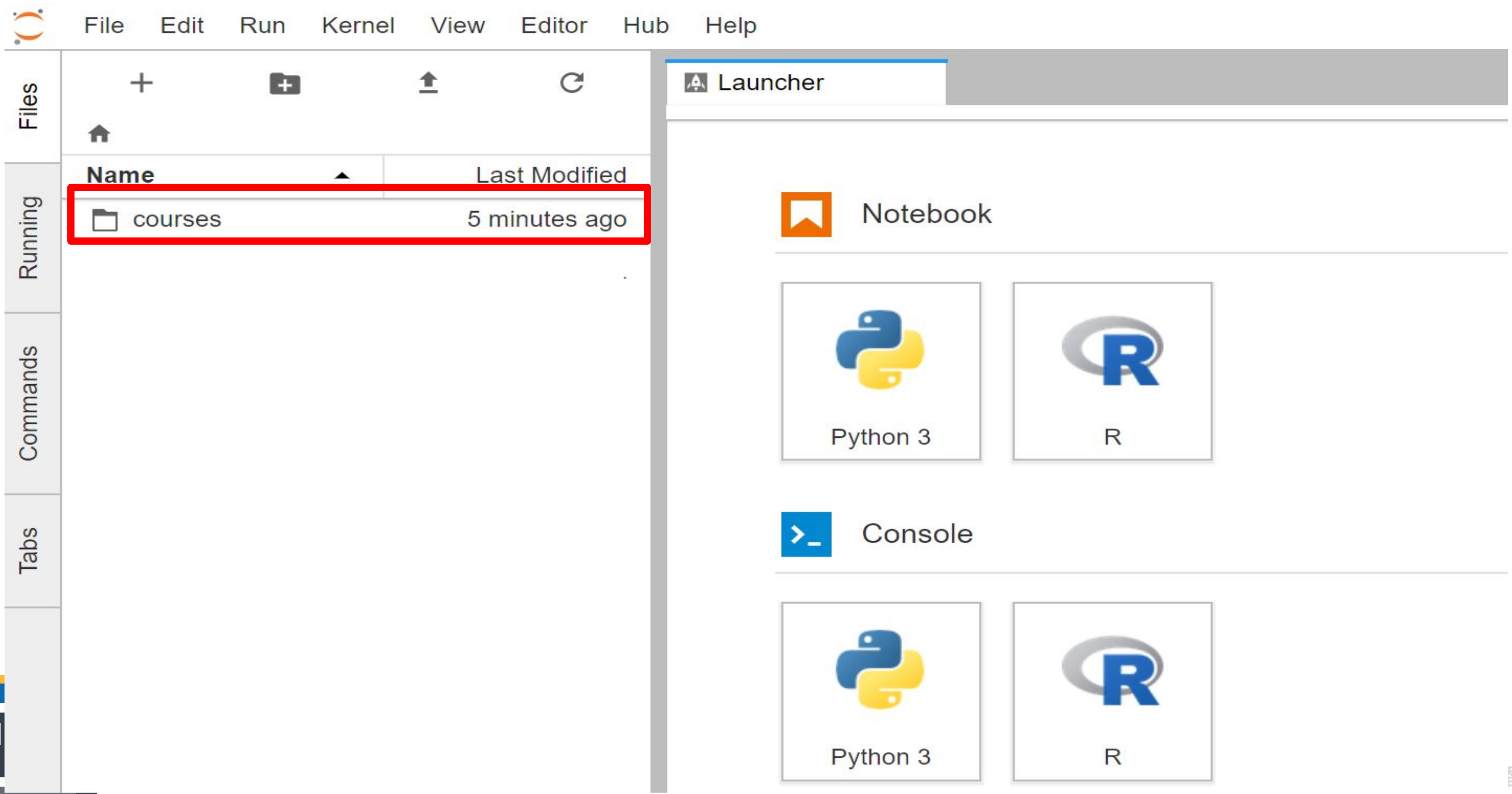
```
In [1]: print("Hello word")

        Hello word
```

- jupyter notebook 會自動儲存
- code 的結果會即時顯示

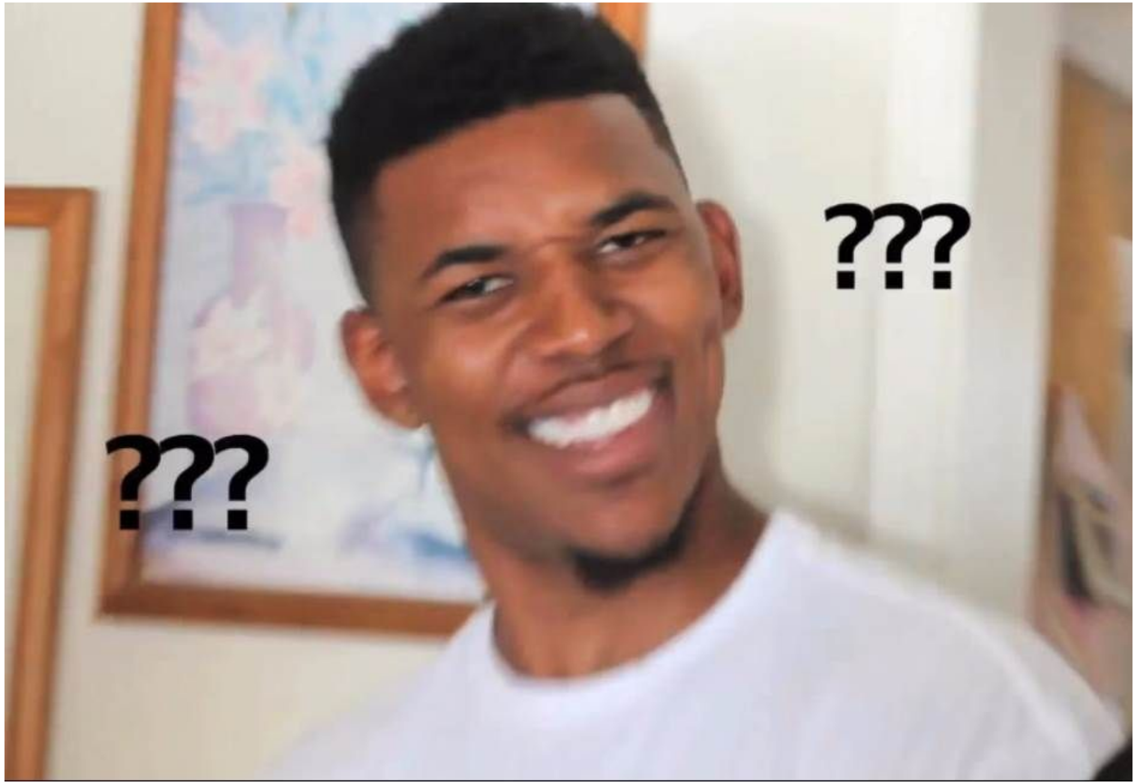
AIA Server 使用


Notebook
R
Console
R
Other

AIA Server- jupyter notebook



明明不一樣，你跟我說這都是jupyter notebook?



Tree and Lab

- Lab：較人性化的介面，但功能不完善

- Tree：功能相對較完善

下載課程資料

- 為維護課程資料，courses 中的檔案皆為 read-only，如需修改請 cp 至自身的環境中
- 打開 terminal，輸入

```
cp -r courses-tpe/python_programming mypython
```

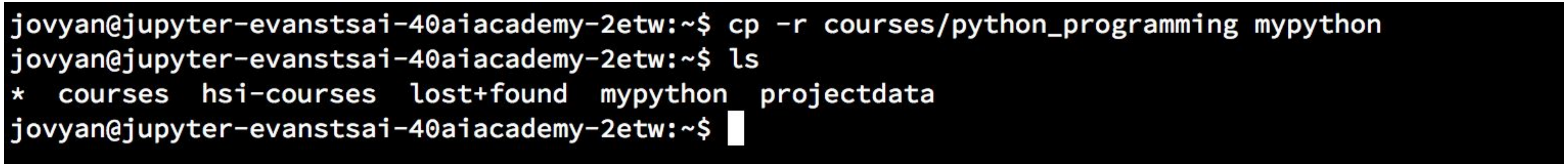
- 今後的課程，如果需要下載課程資料都會使用這樣的方式

下載課程資料

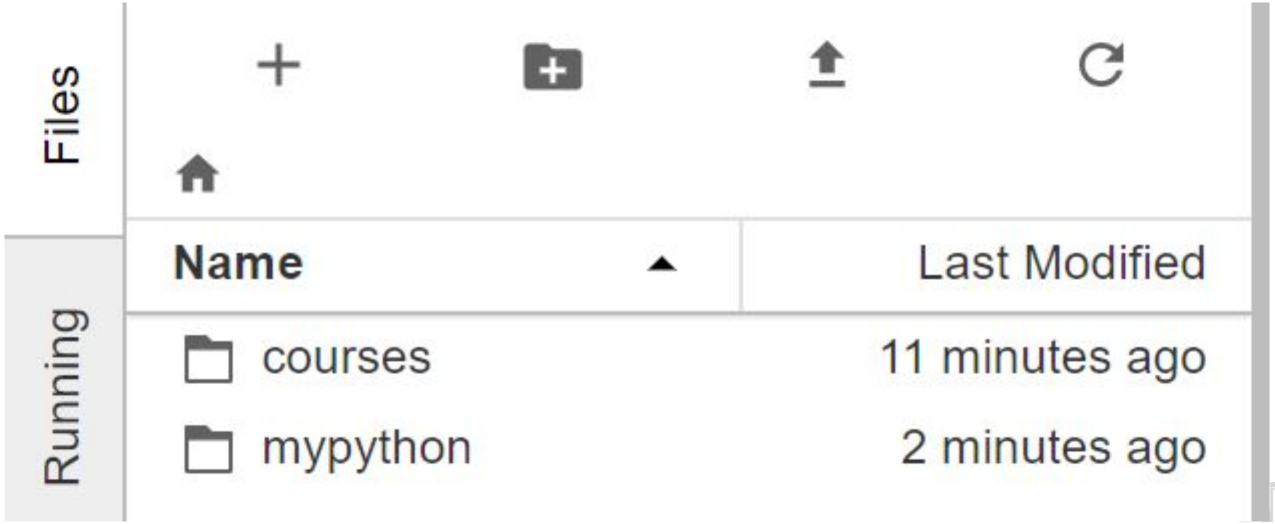
- 為維護課程資料，courses 中的檔案皆為 read-only，如需修改請 cp 至自身的環境中
- 打開 terminal，輸入

```
cp -r courses-tpe/python_programming mypython
```

- 今後的課程，如果需要下載課程資料都會使用這樣的方式



```
jovyan@jupyter-evanstsai-40aiacademy-2etw:~$ cp -r courses/python_programming mypython
jovyan@jupyter-evanstsai-40aiacademy-2etw:~$ ls
*  courses  hsi-courses  lost+found  mypython  projectdata
jovyan@jupyter-evanstsai-40aiacademy-2etw:~$ 
```



Jupyter快捷鍵(非重點!請勿著墨太久!)

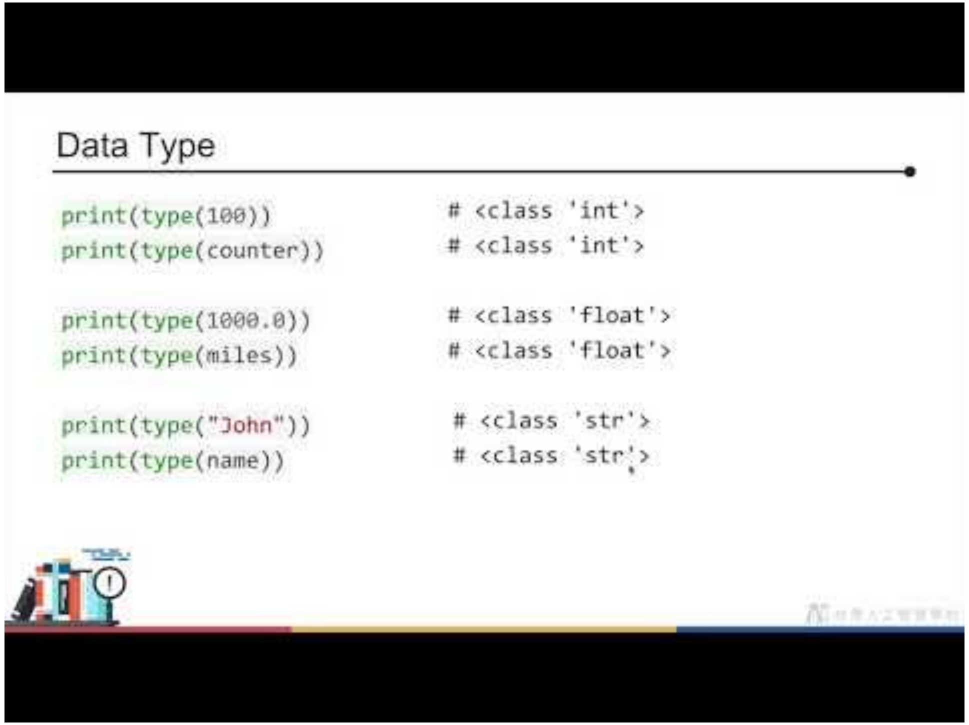
- 補充在另外獨立的slide中。
- slide連結：https://docs.google.com/presentation/d/1rBOmUrPdYcal24EOw7FV6dVQohDDwLDeKxE9RiXK6lY/
- 稍微會寫python但沒用過jupyter notebook的建議可以先看一下這份補充；若沒有學過程式，也可以先開始後續python課程，稍微了解了程式語言後再回來參考喔。

台灣人工智慧學校

Python 程式設計

Felix

Basic Syntax



Data Type

```
print(type(100))          # <class 'int'>
print(type(counter))      # <class 'int'>

print(type(1000.0))       # <class 'float'>
print(type(miles))        # <class 'float'>

print(type("John"))       # <class 'str'>
print(type(name))         # <class 'str'>
```

Variables

```python
# assignment
a = 1
b = c = 5

# assign multiple objects to multiple variables.
a, b, c = 1, 2, "John"
print(a)  # 1
print(b)  # 2
print(c)  # John
```

Data Type

```
counter = 100          # An integer assignment
miles   = 1000.0       # A floating point
name    = "John"       # A string

print(counter)   # 100
print(miles)     # 1000.0
print(name)      # John
```

Data Type

```
print(type(100))                # <class 'int'>
print(type(counter))            # <class 'int'>

print(type(1000.0))             # <class 'float'>
print(type(miles))              # <class 'float'>

print(type("John"))             # <class 'str'>
print(type(name))               # <class 'str'>
```

Keywords

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

Arithmetic Operators

Symbol	Task Performed
+	Addition
-	Subtraction
/	division
%	mod
*	multiplication
//	floor division
**	to the power of

Arithmetic Operators

```
add = 1 + 1       # 2
sub = 1 - 1       # 0
div = 4 / 2       # 2
mod = 4 % 3       # 1
mul = 2 * 3       # 6
f_div = 5 // 2 # 2
power = 2 ** 3 # 8
```

Comparison Operators

Symbol	Task Performed
==	True, if it is equal
!=	True, if not equal to
<	less than
>	greater than
<=	less than or equal to
>=	greater than or equal to

Comparison Operators

```python
a, b = 10, 20
```

```python
a == b      # False
a != b      # True
a < b       # True
a > b       # False
a <= b      # True
a >= b      # False
```

Built-in Functions

e.g. print( ), type( ), int( ) and str( )

```
integer = 123
string = "456"

s_to_i = int(string)      # int now
i_to_s = str(integer)     # str now

print(type(s_to_i))       # <class 'int'>
print(type(i_to_s))       # <class 'str'>
```

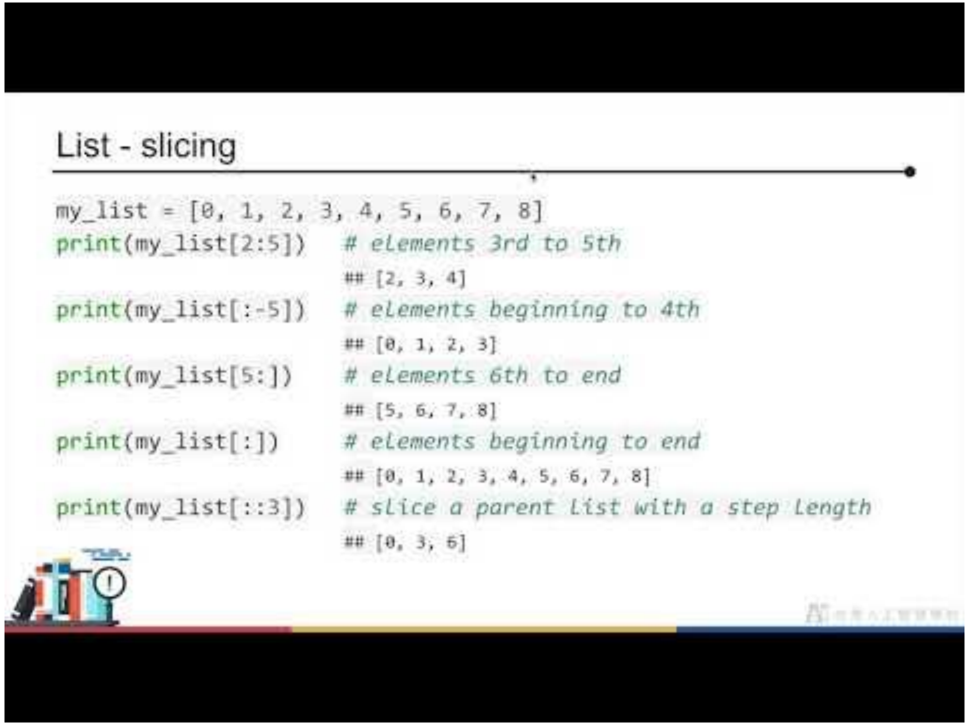
練習 - part 1

Q1. 輸入兩個整數數字，計算兩數字之加、減、乘、除的結果，並且列印出來。

```
Example Output:
第一個數字？ 20
第二個數字？ 10
20 + 10 = 30
20 - 10 = 10
20 * 10 = 200
20 / 10 = 2
```

hint1: 利用內建 input() 取得輸入數字，並且利用 int() 將輸入字串轉成整數。
hint2: num1 + num2 = sum 可利用 print(num1, "+", num2, "=", num1 + num2) 印出。

Data Structures



List - slicing

```
my_list = [0, 1, 2, 3, 4, 5, 6, 7, 8]
print(my_list[2:5])     # elements 3rd to 5th
                        ## [2, 3, 4]
print(my_list[:-5])     # elements beginning to 4th
                        ## [0, 1, 2, 3]
print(my_list[5:])      # elements 6th to end
                        ## [5, 6, 7, 8]
print(my_list[:])       # elements beginning to end
                        ## [0, 1, 2, 3, 4, 5, 6, 7, 8]
print(my_list[::3])     # slice a parent list with a step length
                        ## [0, 3, 6]
```

Numbers

```python
# Output: <class 'int'>
print(type(5))

# Output: <class 'float'>
print(type(5.0))

# Output: <class 'complex'>
c = 5 + 3j
print(type(c))
```

Lists

```
# empty list
my_list = []

# list of integers
my_list = [1, 2, 3]

# list with mixed datatypes
my_list = [1, "Hello", 2.3]

# nested list
my_list = ["mouse", [8, 4, 6]]
```

List - index

```
my_list = ['h','e','l','l','o']

print(my_list[0])      # Output: h

print(my_list[1])      # Output: e

# my_list[5.0]      # Error! Only integer can be used for indexing

n_list = ["Happy", [2,0,1,8]]      # Nested List
print(n_list[1][3])                # Output: 8
```

List - negative indexing

```
my_list = ['p','r','o','b','e']

print(my_list[-1])    # Output: e

print(my_list[-5])    # Output: p
```

List - slicing

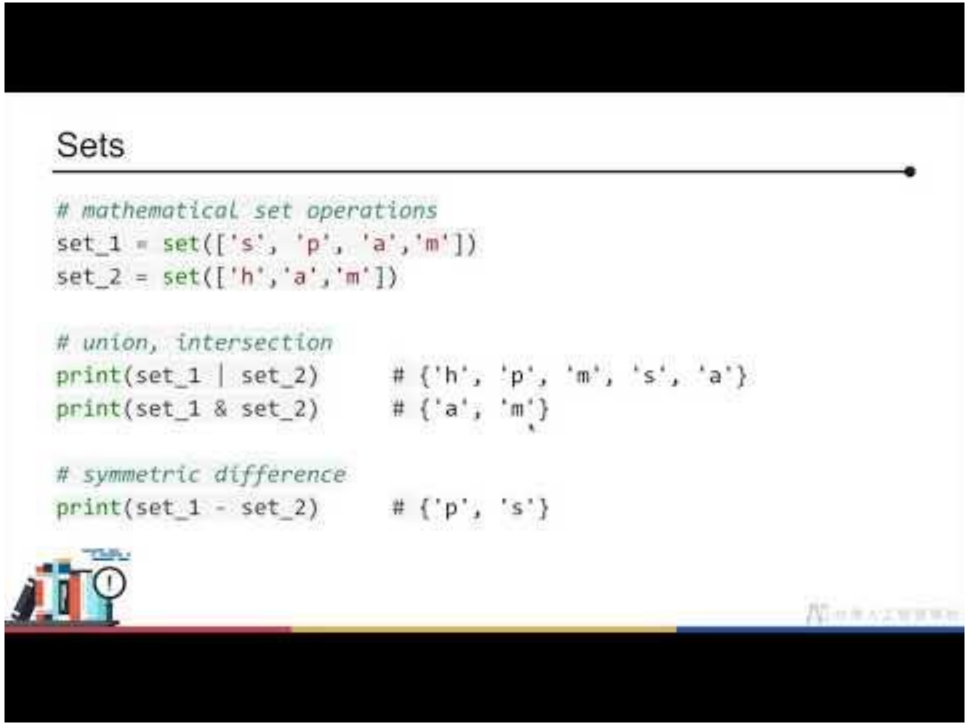
```
my_list = [0, 1, 2, 3, 4, 5, 6, 7, 8]
print(my_list[2:5])    # elements 3rd to 5th
                       ## [2, 3, 4]
print(my_list[:-5])    # elements beginning to 4th
                       ## [0, 1, 2, 3]
print(my_list[5:])     # elements 6th to end
                       ## [5, 6, 7, 8]
print(my_list[:])      # elements beginning to end
                       ## [0, 1, 2, 3, 4, 5, 6, 7, 8]
print(my_list[::3])    # slice a parent list with a step length
                       ## [0, 3, 6]
```

Built-in List Methods

```
num_list = [0, 0, 1, 2, 3, 4, 5, 6, 7, 8]

# append() is used to add an element at the end of the list.
num_list.append(9)

# remove() takes a single element as an argument and removes it
from the list.
num_list.remove(9)
```

Built-in List Methods

```python
# index() is used to find the index value of a particular element.
num_list.index(5)

# pop() takes a single argument (index) and removes the element
present at that index from the list.
result = num_list.pop(7)

print(result)       # 6
print(num_list)     # [0, 0, 1, 2, 3, 4, 5, 7, 8]
```



Sets

```python
# mathematical set operations
set_1 = set(['s', 'p', 'a','m'])
set_2 = set(['h','a','m'])

# union, intersection
print(set_1 | set_2)      # {'h', 'p', 'm', 's', 'a'}
print(set_1 & set_2)      # {'a', 'm'}

# symmetric difference
print(set_1 - set_2)      # {'p', 's'}
```

Tuples

```
# empty tuple
my_tuple = ()
print(my_tuple)          # Output: ()

# tuple having integers
my_tuple = (1, 2, 3)
print(my_tuple)          # Output: (1, 2, 3)
```

Strings

```python
# all of the following are equivalent
my_string = 'Hello'
print(my_string)

my_string = "Hello"
print(my_string)
```

Strings

```
my_str = 'Hello World!'
print('my_str = ', my_str)    # my_str =  Hello World!

# first character, last character
print(my_str[0])        # H
print(my_str[-1])       # !

# slicing 3nd to 5th character
print(my_str[2:5])      # llo
```

Strings

```
str1 = 'Hello'
str2 = 'World!'

# using +
print(str1 + str2)      # HelloWorld!

# using *
print(str1 * 3)         # HelloHelloHello
```

Built-in Strings Methods

```python
my_string = "hello world"

print(my_string.find("he"))          # Output: 0
print(my_string.capitalize())        # Output: Hello world
print(my_string.upper())             # Output: HELLO WORLD
print(my_string.endswith("d"))       # Output: True

print(my_string.split(" "))          # Output: ['hello', 'world']

print(my_string.replace("hello", "Nihao"))  # Output: Nihao world
```

Sets

```
# set of integers
my_set = {1, 2, 3}
print(my_set)        # {1, 2, 3}

# set of mixed datatypes
my_set = {1.0, "Hello", (1, 2, 3)}
print(my_set)        # {'Hello', 1.0, (1, 2, 3)}
```

Sets

```python
# mathematical set operations
set_1 = set(['s', 'p', 'a','m'])
set_2 = set(['h','a','m'])

# union, intersection
print(set_1 | set_2)        # {'h', 'p', 'm', 's', 'a'}
print(set_1 & set_2)        # {'a', 'm'}

# symmetric difference
print(set_1 - set_2)        # {'p', 's'}
```

Dictionary

```
# empty dictionary
my_dict = {}

# dictionary with integer keys
my_dict = {1: 'a', 2: 'b'}

# dictionary with mixed keys
my_dict = {'name': 'Tom', 1: 23}
```

Dictionary

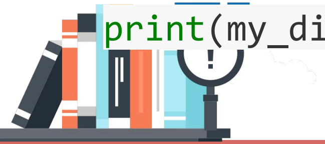
```
# Another define
my_dict = dict()

# add elements
my_dict['One'] = '1'
my_dict['OneTwo'] = 12
print (my_dict)         # {'One': '1', 'OneTwo': 12}

# update value
my_dict['One'] = 111
print (my_dict)         # {'One': 111, 'OneTwo': 12}
```

Dictionary

```python
# Merge two lists to a dictionary.
names = ['One', 'Two', 'Three', 'Four', 'Five']
numbers = [1, 2, 3, 4, 5]
merged_dict = dict(zip(names, numbers))

print(merged_dict)      # {'One': 1, 'Two': 2, 'Three': 3, 'Four': 4, 'Five': 5}
```

Dictionary Methods

```
my_dict = {'name':'Jack', 'age': 16, 'gender':'man'}

# remove a particular item
print(my_dict.pop('gender'))       # man
print(my_dict)                     # {'name': 'Jack', 'age': 16}

# Returns view of dictionary's (key, value) pair
print(my_dict.items())             # [('name', 'Jack'), ('age', 16)]

# Return a new view of the dictionary's keys.
print(my_dict.keys())              # ['name', 'age']

# remove all items
my_dict.clear()
print(my_dict)                     # {}
```



練習 - part 2

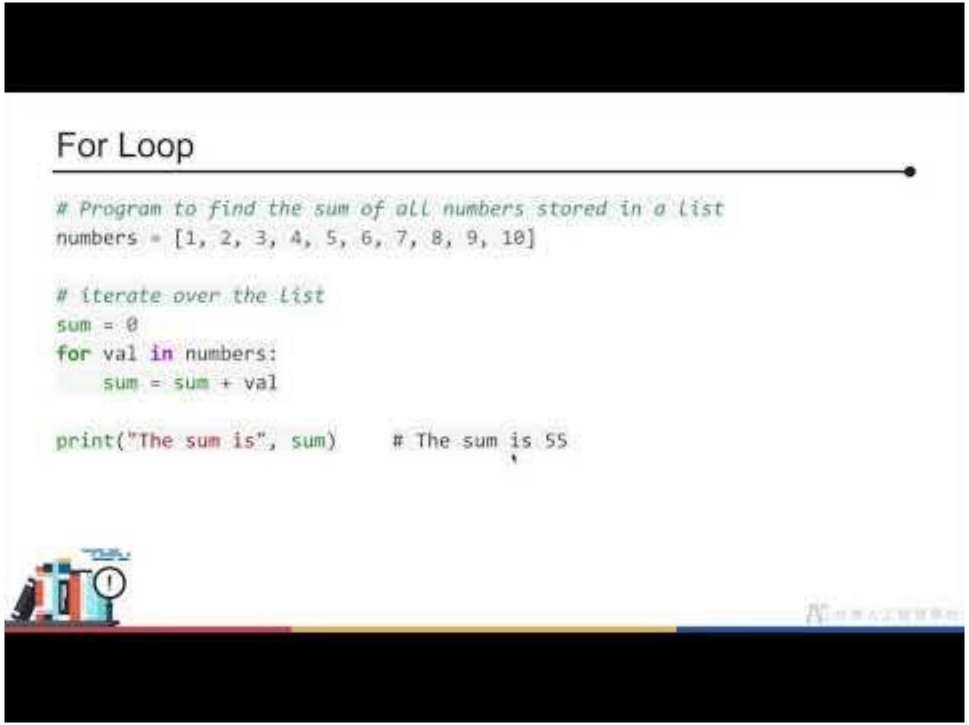
Q1. 給定一個 a_list = [3, 7, 6, 2, 9, 4, 1], 請列印出下列結果:
(1) 第2個元素
(2) 最後一個元素
(3) 第3到第5個元素的列表

Q2.

編號	姓名
s1	John
s2	Tom
s3	Lisa

(1) 將上述表格資料, 存成 Dictionary 的資料結構。(key = 編號, value = 姓名)
(2) 列印出該 key 為 s2 的 value 的值
(3) 添加人員 編號 s4, 姓名 Mana
(4) 刪除人員 John

Control Flow



For Loop

```python
# Program to find the sum of all numbers stored in a list
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

# iterate over the list
sum = 0
for val in numbers:
    sum = sum + val

print("The sum is", sum)     # The sum is 55
```

if

```
num = 3
if num > 0:
    print(num, "is a positive number.")

num = -1
if num > 0:
    print(num, "is a positive number.")

## Output: 3 is a positive number.
```

if … else

```
num = -1
if num >= 0:
    print(num, "Positive or Zero")
else:
    print(num, "is a Negative number")

## Output: -1 is a Negative number.
```

if … elif … else

```python
num = 0
if num > 0:
    print("Positive number")
elif num == 0:
    print("Zero")
else:
    print("Negative number")

## Output: Zero
```

Logical - or, and

```
num = 5

if num == 0 or num == 1:
    print("Zero or One")
elif num >= 2 and num <= 10:
    print("From 2 to 10")
else:
    print('More')

## Output: From 2 to 10
```

is, not

```
num = 4
```

```
# num == 4
if num is 4:
    print("num is 4")
```

```
# !(num == 5)
if not num == 5:
    print("num is not 5")
```

```
# num != 6
if num is not 6:
    print("num is not 6")
```

```
# !(num == 7)
if not num is 7:
    print("num is not 7")
```

For Loop

```
# Program to find the sum of all numbers stored in a list
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

# iterate over the list
sum = 0
for val in numbers:
    sum = sum + val

print("The sum is", sum)       # The sum is 55
```

For loop with range()

```python
# range(stop)
# range(start, stop[, step])

numbers = [1, 2, 3, 4, 5, 6]

# iterate over the list using index
for i in range(len(numbers)):
    print("number", numbers[i])


# iterate over the list using 2 steps
for i in range(0, len(numbers), 2):
    print("2 steps", numbers[i])
```

```
# Output
  number 1
  number 2
  number 3
  number 4
  number 5
  number 6

# Output
  2 steps 1
  2 steps 3
  2 steps 5
```

For loop with enumerate( )

```
pets = ('Dogs', 'Cats', 'Turtles', 'Rabbits')

for index, pet in enumerate(pets):
    print(index, pet)



# Output:
  0 Dogs
  1 Cats
  2 Turtles
  3 Rabbits
```

While Loop

```
n = 10

# initialize sum and counter
sum = 0
i = 1

while i <= n:
    sum = sum + i
    i = i+1     # update counter

# print the sum
print("The sum is", sum)     # The sum is 55
```

Nested Loop

```python
for i in range(0, 2):
    for j in range(0, 2):
        print("i=", i, "j=", j, ", i*j=", i*j)
```

```
# Output:
  i= 0 j= 0 , i*j= 0
  i= 0 j= 1 , i*j= 0
  i= 1 j= 0 , i*j= 0
  i= 1 j= 1 , i*j= 1
```

break, continue and pass

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
# break
for val in numbers:
    if val >= 4:
        break
    print(val)
```

```
# Output
  1
  2
  3
```

```
# pass
for val in numbers:
    pass
```

```
# continue
for val in numbers:
    if val >= 3 and val <=8:
        continue
    print(val)
```

```
# Output
  1
  2
  9
  10
```

List comprehension

```
# make new lists by using iterable
squares = []
for x in range(10):
    squares.append(x**2)
print(squares)               # [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]


# equivalently
squares = [x**2 for x in range(10)]
print(squares)               # [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

List comprehension

```
# with if
squares = [x**2 for x in range(10) if x % 2 == 0]
print(squares)          ## [0, 4, 16, 36, 64]



# equivalently
squares = []
for x in range(10):
    if x % 2 == 0:
        squares.append(x**2)

print(squares)          ## [0, 4, 16, 36, 64]
```

練習 - part 3-1

Q1. 建立一個驗證密碼的小程式，程式 內建一組字串密碼，請使用者輸入一組字串密碼，比對密碼是否輸入正確。

```
Expected Result:
      請輸入密碼: Passw0rd
      密碼正確
      or
      請輸入密碼: adfgg
      密碼錯誤
```

Q2. 給予一個列表，計算出列表中元素為 2的倍數的和。
Sample List : [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

```
Expected Result : 30
```

練習 - part 3-2

Q3. 輸入人物的身高、體重，計算出該人物的 BMI
公式：BMI = 體重(公斤) / 身高*身高 (公尺)

P.S. 於2002年4月公布臺灣成人肥胖標準：
BMI＜18.5 為過輕，
18.5≦BMI＜24 為正常體重，
24≦BMI＜27 為過重，
BMI≧27 即為肥胖

Q4. 印出 1 到 50，但如果是 3 的倍數就印 Fizz，如果是 5 的倍數就印 Buzz，如果同時是 3 和 5 的倍數就印 FizzBuzz。

課後問卷

親愛的學員您好:

　　為了解課程內容的安排是否恰當，想請各位學員給我們一些回饋，各位寶貴的意見將能協助我們設計出更優質的課程！

問卷連結