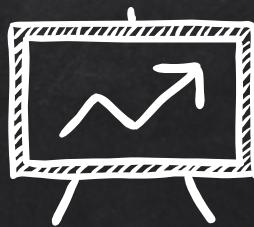


PYTHON X 資料分析

2017.0317



PYTHON X 資料分析

2017.0317



KRISTEN

DATA SCIENTIST AT WAVE-IN COMMUNICATION INC.

E-COMMERCE DATA MINING
TELECOM DATA ANALYTICS



課程簡介

- ✓ 資料結構 (Data structure)
- ✓ 資料處理 (Data processing)
- ✓ 資料探索 (Exploratory Data Analysis)
- ✓ 資料視覺化 (Data visualization)
- ✓ 資料分類與分群 (Classification and Clustering)



課程簡介

- ✓ 套件
- ✗ Numpy
- ✗ Panda
- ✗ Matplotlib
- ✗ Scikit-learn



課程資訊

✓ 課程時間：

✗ 3/17, 3/24, 4/7, 4/28, 5/5, 5/12, 5/19, 6/2

✗ 週五 PM 7:00- PM 9:00

✓ 課程網站：

✗ <https://github.com/kristenchan/Python-Data-Analysis>



聯絡方式

✓ Email :

✗ kristenchan.py@gmail.com

✓ 寄信主旨 :

✗ HW_第幾次作業_姓名

e.g. 陳小明交第三次作業 → HW_3_陳小明

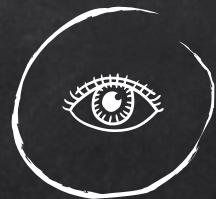
✗ Q_姓名



使用軟體

- ✓ Python 3
- ✓ Anaconda
- ✓ Jupyter Notebook

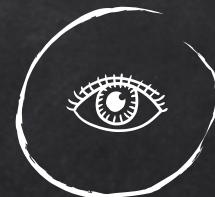
WHY PYTHON



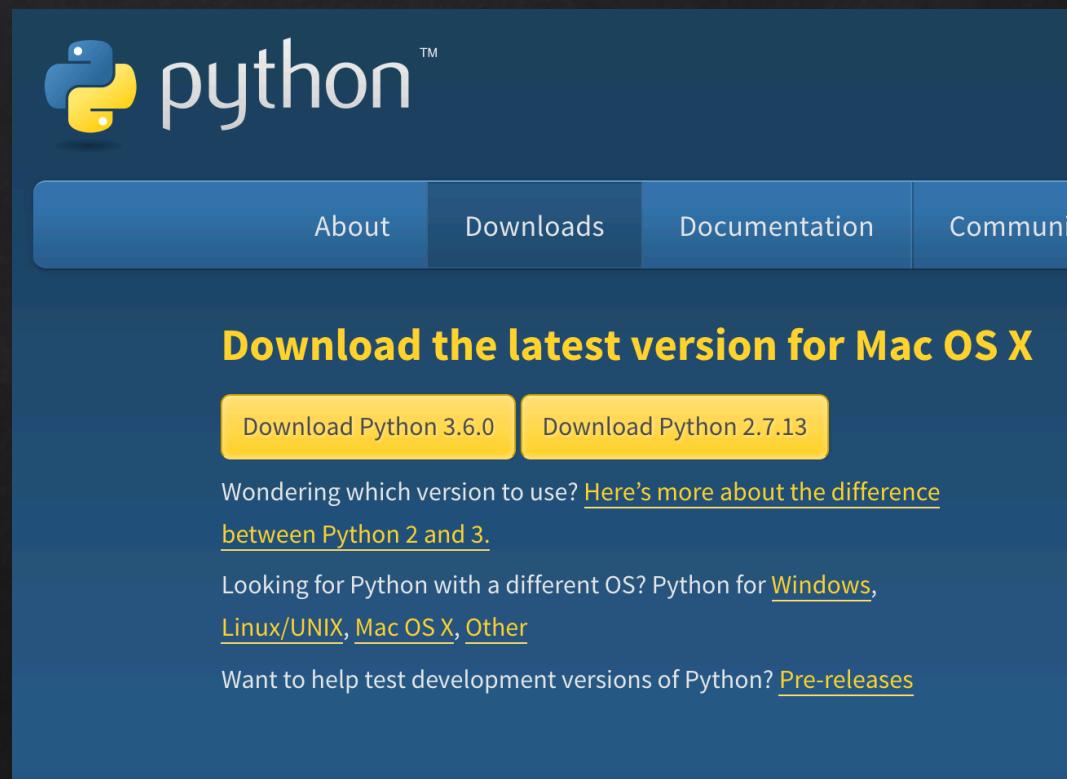
Why Python

- ✓ 簡單易學
- ✓ 開源免費
- ✓ 跨平台
- ✓ 豐富的函式庫

WHY PYTHON 3



Why Python 3



The screenshot shows the Python.org homepage with a dark blue header. The Python logo (a blue and yellow stylized 'P') is on the left. The word "python" is written in white lowercase letters next to it, with a trademark symbol (TM) at the top right. Below the header is a navigation bar with four tabs: "About", "Downloads", "Documentation", and "Community". The "Downloads" tab is highlighted with a yellow background. The main content area has a dark blue background. A large yellow button in the center says "Download the latest version for Mac OS X". Below this button are two smaller yellow buttons: "Download Python 3.6.0" and "Download Python 2.7.13". To the right of these buttons is a link: "Wondering which version to use? [Here's more about the difference between Python 2 and 3.](#)" Further down, there is another link: "Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [Mac OS X](#), [Other](#)". At the bottom, there is a link: "Want to help test development versions of Python? [Pre-releases](#)".

Ref: <https://www.python.org/>



Applications for Python

- ✓ Web程式 - Django、Flask
- ✓ Game
- ✓ 網頁爬蟲 - Scrapy
- ✓ 資料分析/機器學習 - numpy, scipy, matplotlib
- ✓ 自然語言處理 - nltk



Anaconda

 **ANACONDA**
Powered by Continuum Analytics®

Log In Get Support Search Contact

PRODUCTS COMMUNITY CONSULTING TRAINING ABOUT RESOURCES

**DOWNLOAD
ANACONDA NOW**

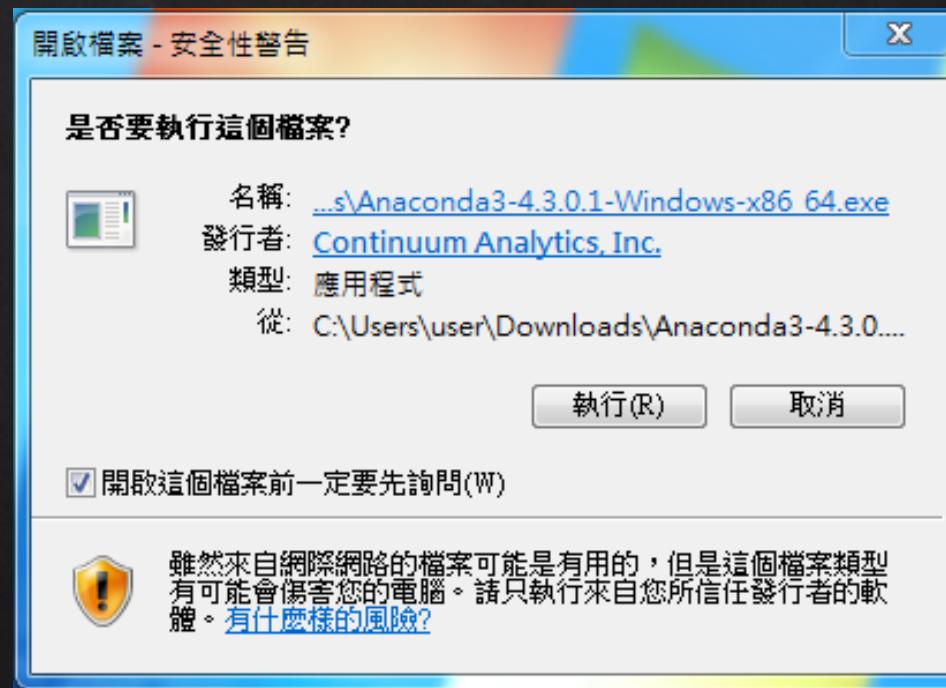
Download for   



GET SUPERPOWERS



Anaconda



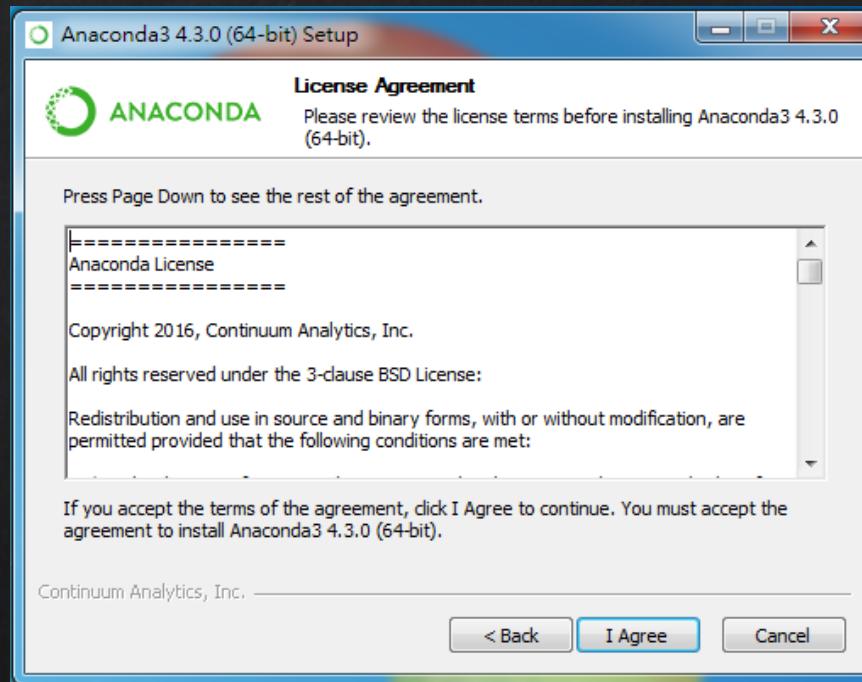


Anaconda



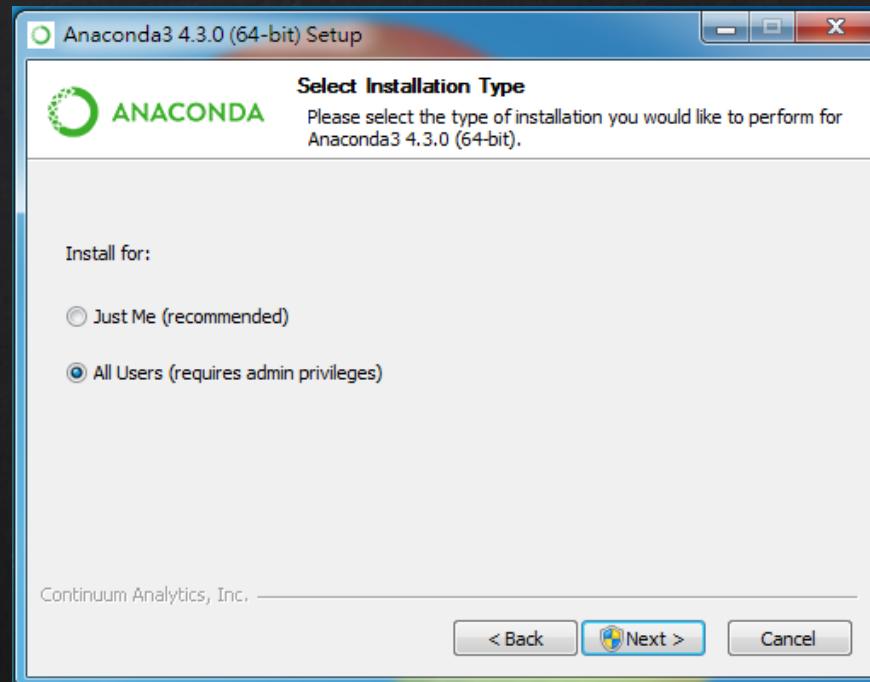


Anaconda



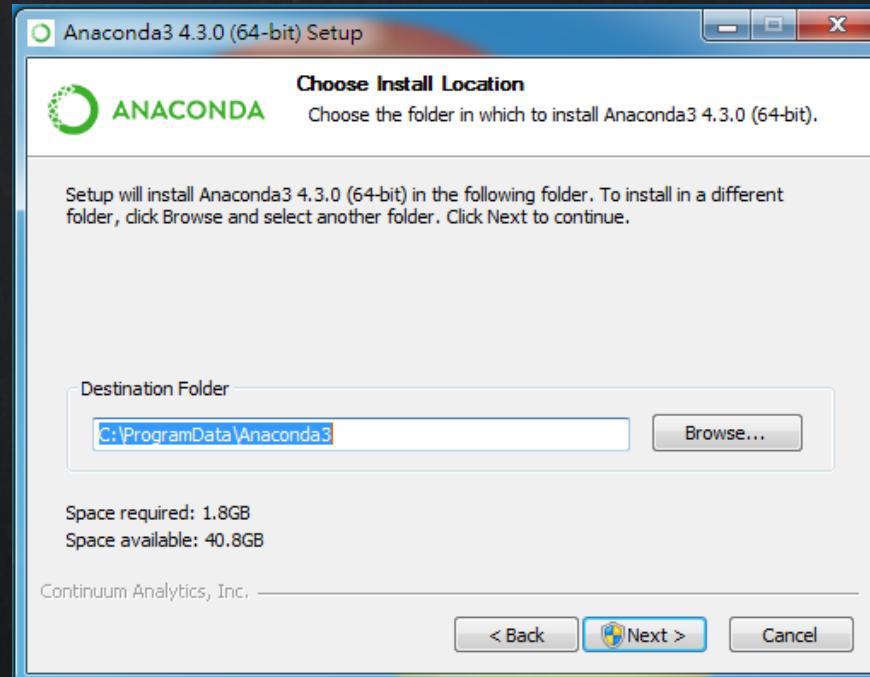


Anaconda



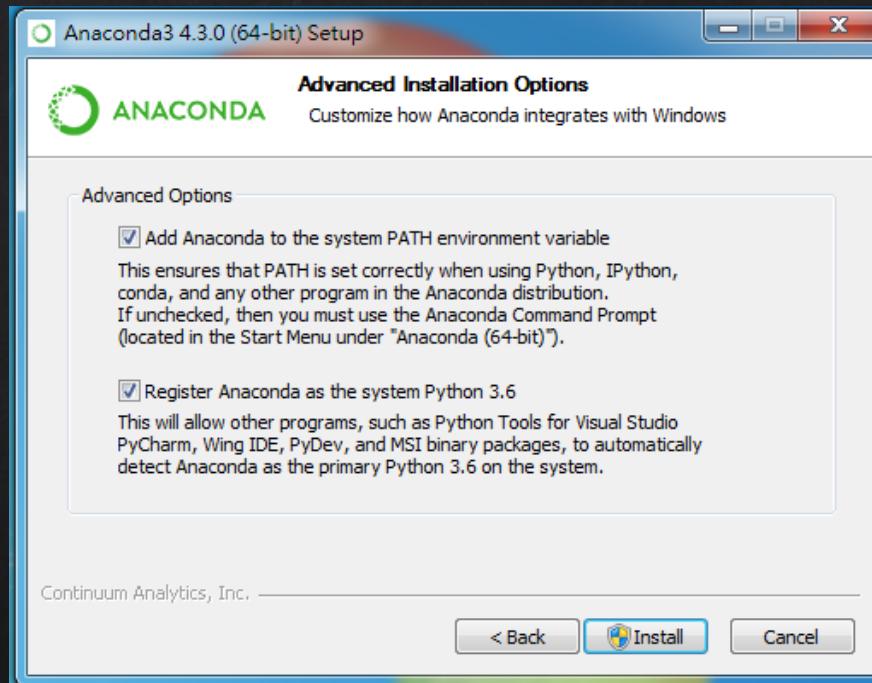


Anaconda



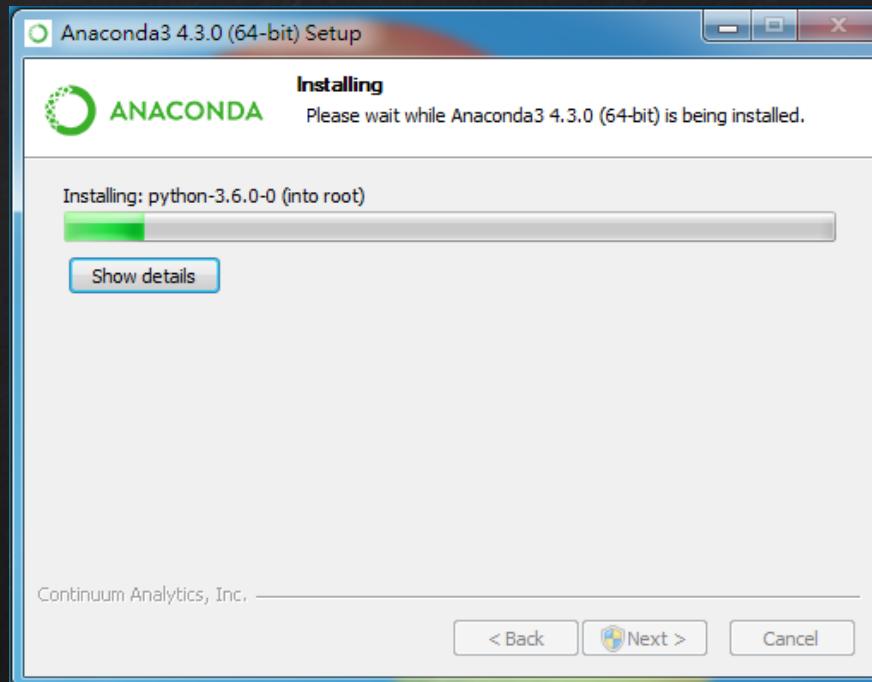


Anaconda



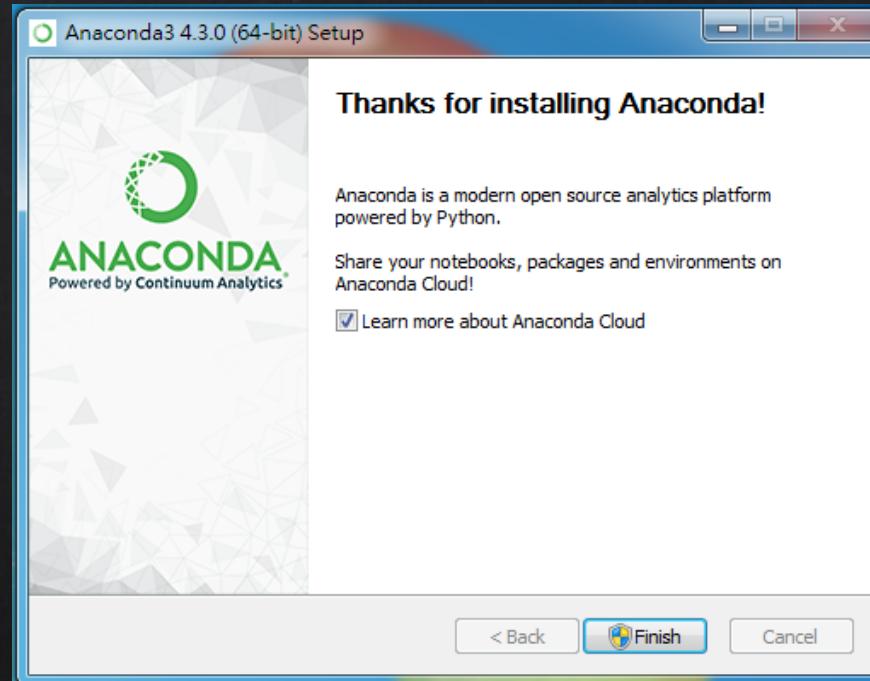


Anaconda





Anaconda



WHY JUPYTER NOTEBOOK



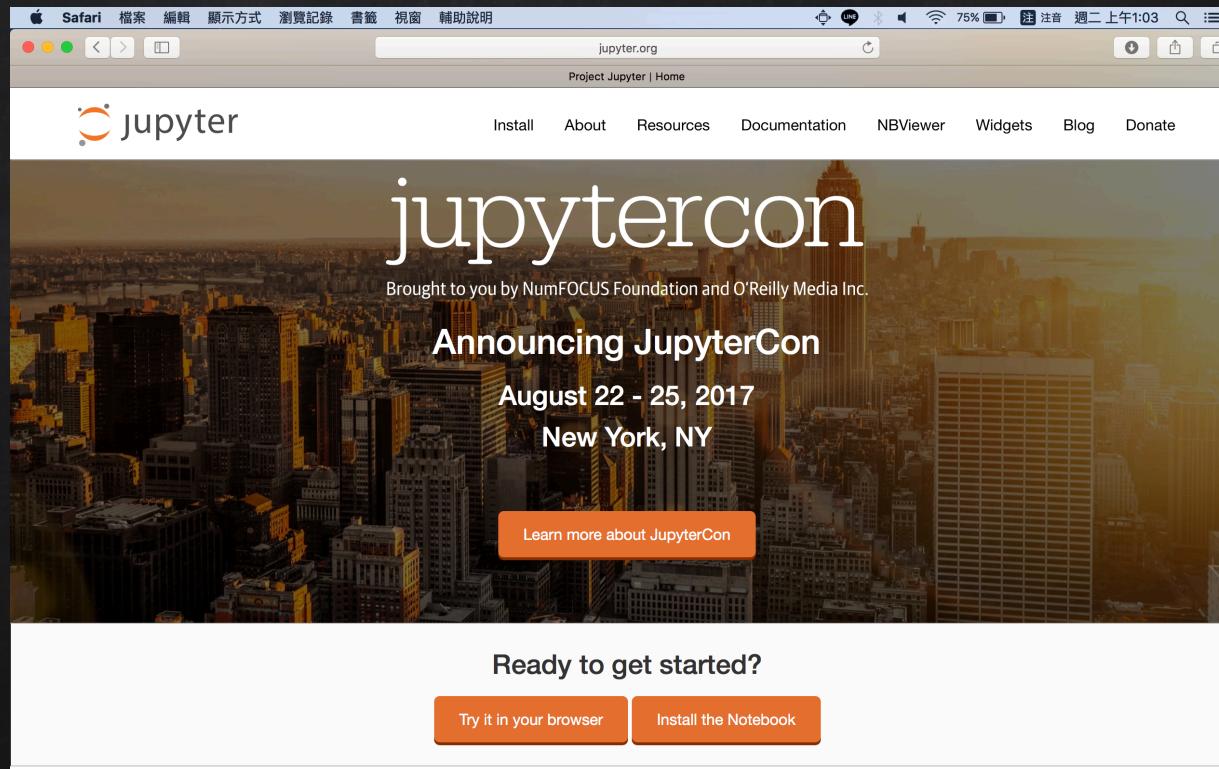
Why Jupyter Notebook

- ✓ 互動式計算介面
- ✓ 支援多種程式語言
- ✓ 支援markdown

Ref: <https://www.wikiwand.com/zh-tw/Markdown>



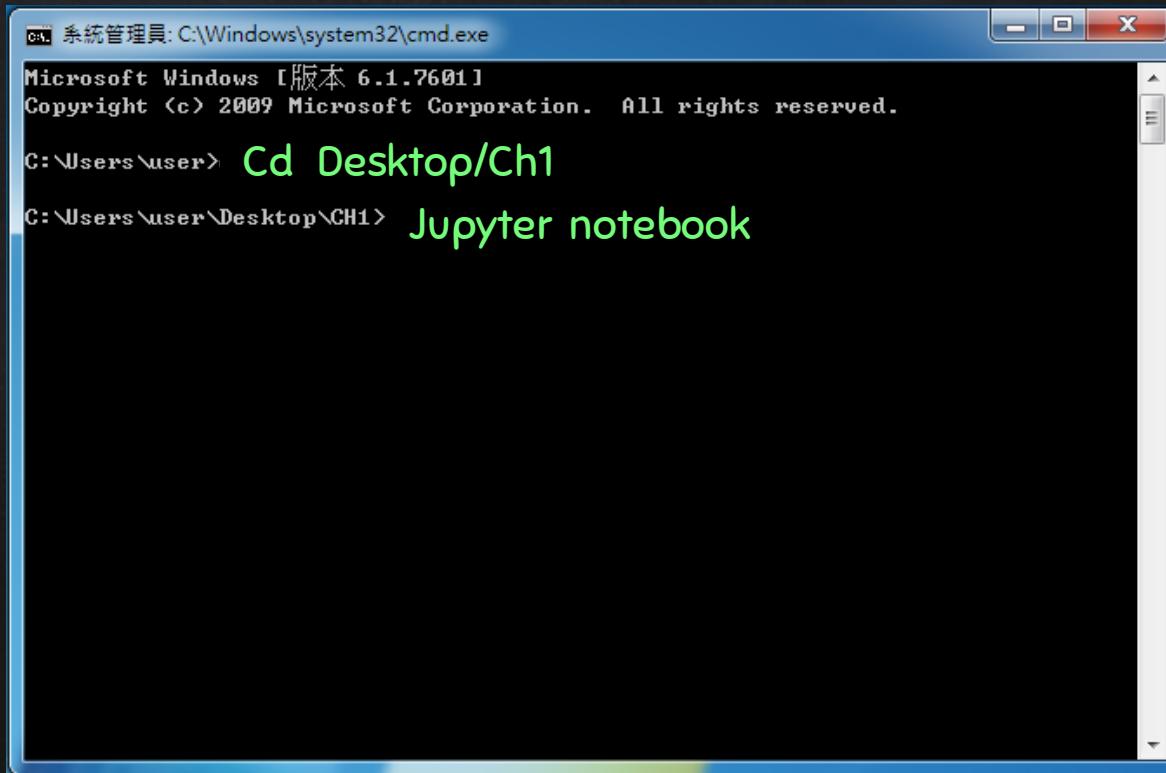
Jupyter Notebook(Online)



The screenshot shows the official Jupyter website (jupyter.org) displayed in a Safari browser. The page features a dark background with a city skyline at sunset. At the top, there's a navigation bar with links for 'Install', 'About', 'Resources', 'Documentation', 'NBViewer', 'Widgets', 'Blog', and 'Donate'. The main heading 'jupytercon' is prominently displayed in large white letters. Below it, text reads 'Brought to you by NumFOCUS Foundation and O'Reilly Media Inc.' followed by 'Announcing JupyterCon' and 'August 22 - 25, 2017 New York, NY'. A button labeled 'Learn more about JupyterCon' is visible. At the bottom, a white box contains the text 'Ready to get started?' with two buttons: 'Try it in your browser' and 'Install the Notebook'.



Open your Jupyter Notebook !!

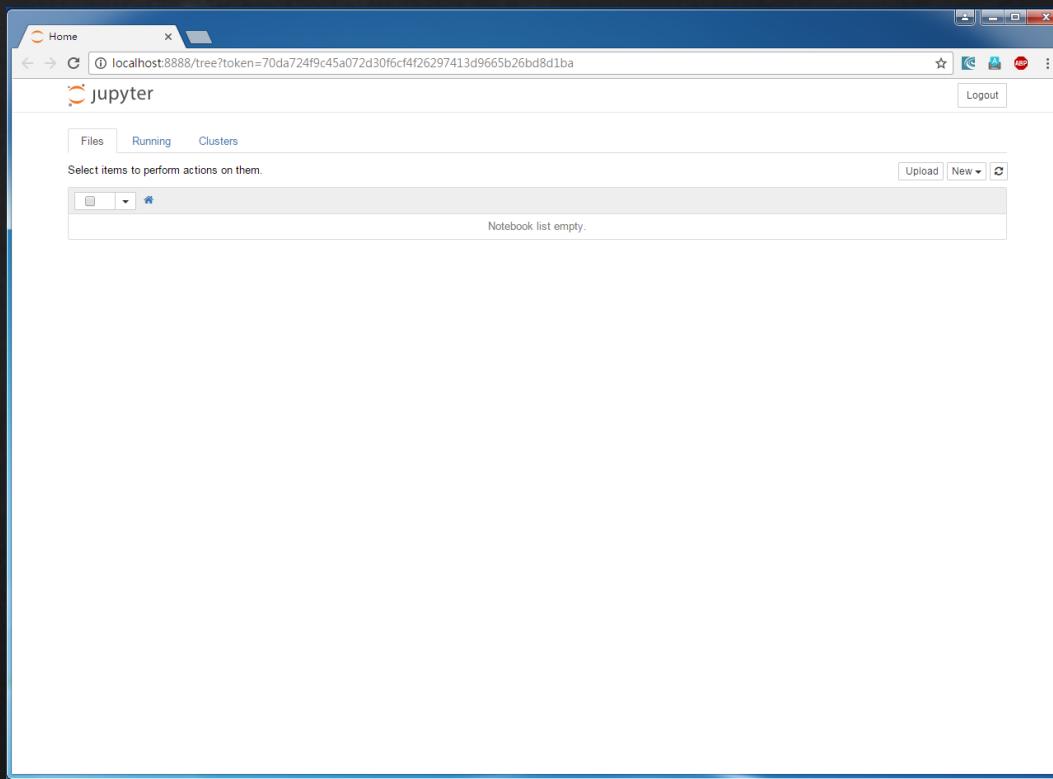


```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\user> Cd Desktop\Ch1
C:\Users\user\Desktop\CH1> Jupyter notebook
```

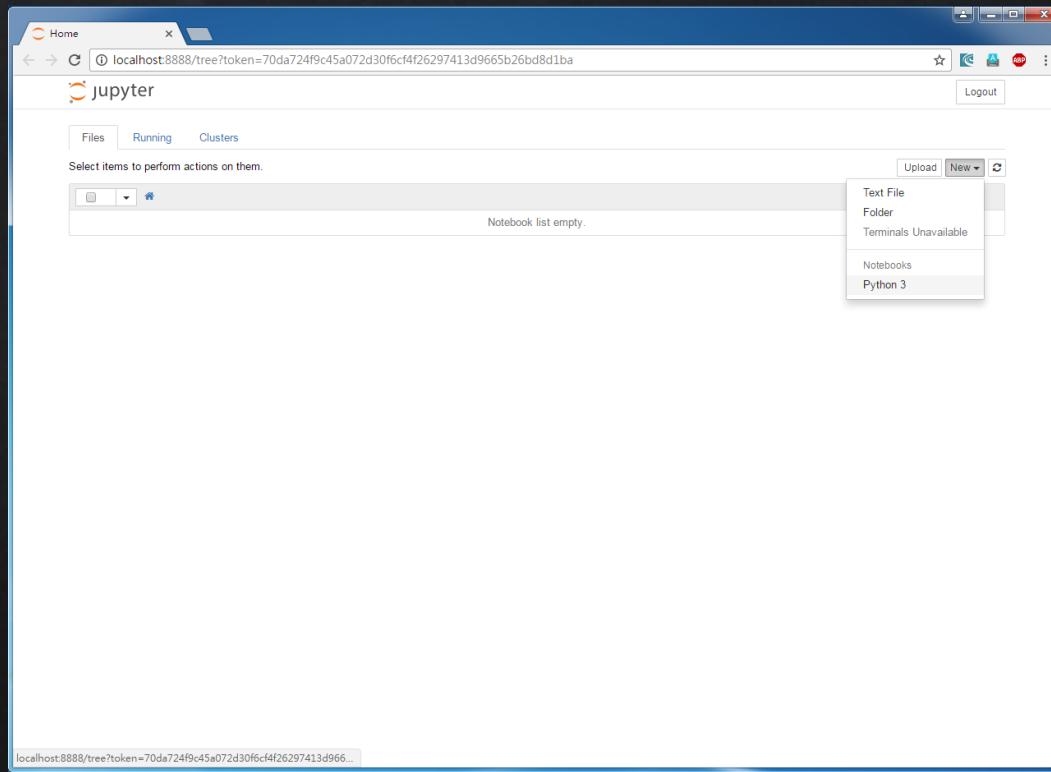


Open your Jupyter Notebook !!





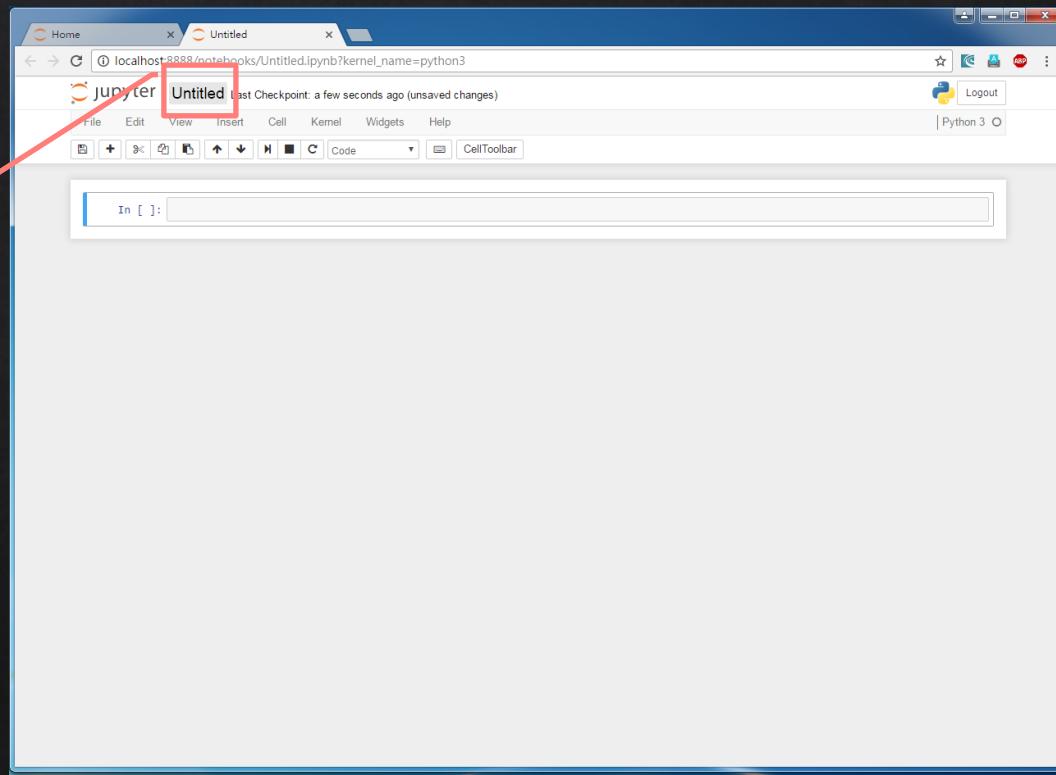
Open your Jupyter Notebook !!





Open your Jupyter Notebook !!

改檔名





Why “Hello World!!” ?

- ✓ 第一支程式
- ✓ Ref : https://zh.wikipedia.org/zh-tw/Hello_World



Hello Python!!

```
In [1]: print('Hello Python !!')  
Hello Python !!
```

```
In [2]: print('Hello Python !!') #我的第一個python程式  
Hello Python !!
```

: Python 註解



The Zen of Python

In [3]: `import this`

The Zen of Python, by Tim Peters

```
Beautiful is better than ugly.  
Explicit is better than implicit.  
Simple is better than complex.  
Complex is better than complicated.  
Flat is better than nested.  
Sparse is better than dense.  
Readability counts.  
Special cases aren't special enough to break the rules.  
Although practicality beats purity.  
Errors should never pass silently.  
Unless explicitly silenced.  
In the face of ambiguity, refuse the temptation to guess.  
There should be one-- and preferably only one --obvious way to do it.  
Although that way may not be obvious at first unless you're Dutch.  
Now is better than never.  
Although never is often better than *right* now.  
If the implementation is hard to explain, it's a bad idea.  
If the implementation is easy to explain, it may be a good idea.  
Namespaces are one honking great idea -- let's do more of those!
```

變數命名規則

VARIABLE NAME RULES



Variable Name Rules

✓ First character :

A letter (a - z, A - Z) or underscore (_)

✓ Other characters :

Letters (a - z, A - Z), numbers (0-9) or (_)

✓ No :!, @, #, \$, %, - etc.

✓ Reserved words 保留字



Variable Name Rules -- Reserved words

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	



Variable Name Rules

```
In [4]: _variable = 1
```

```
In [5]: variable = 1
```

Good

```
In [6]: first_variable = 1
```

```
In [7]: $variable = 1
```

```
File "<ipython-input-7-1ea8277c23c2>", line 1
$variable = 1
^

SyntaxError: invalid syntax
```

```
In [8]: 1variable = 1
```

```
File "<ipython-input-8-ae24453ae98c>", line 1
1variable = 1
^

SyntaxError: invalid syntax
```



Variable Name Rules

```
In [4]: _variable = 1
```

```
In [5]: variable = 1
```

```
In [6]: first_variable = 1
```

```
In [7]: $variable = 1  
        File "<ipython-input-7-1ea8277c23c2>", line 1  
              $variable = 1  
                   ^  
SyntaxError: invalid syntax
```



```
In [8]: 1variable = 1  
        File "<ipython-input-8-ae24453ae98c>", line 1  
              1variable = 1  
                   ^  
SyntaxError: invalid syntax
```

變數賦值

ASSIGNMENT STATEMENTS

交換變數
SWAP VARIABLES

資料型態

DATA TYPE



Data Type

Type	Example
int	10
float	3.14
boolean	True False
string	'10'



Data Type

✓ `type(x)`

使用`type`來得知 `x` 的型態

運算子 OPERATOR



Operator

- ✓ 算術運算子(Arithmetic Operator)
- ✓ 關係運算子(Comparison Operator)
- ✓ 指派運算子(Assignment Operator)
- ✓ 邏輯運算子(Logical Operator)



Operator -- Arithmetic Operator

運算子	功能
+	加法運算
-	減法運算/負號
*	乘法運算
**	指數運算
/	除法運算
//	整除
%	取餘數



Exercise_1

Q. 現在台北溫度是 19°C ，請將 $^{\circ}\text{C}$ 轉成 $^{\circ}\text{F}$

公式： $\text{Fahrenheit} = \text{Celsius} \times (9/5) + 32$



Operator -- Comparison Operator

運算子	說明
<	小於
>	大於
<=	小於等於
>=	大於等於
==	等於
!=	不等於



Operator -- Assignment Operator

運算子	說明
=	將右邊的運算結果,指定給左邊的變數
+=	將左邊變數值加右邊的值,再指定給左邊的變數
-=	將左邊變數值減右邊的值,再指定給左邊的變數
*=	將左邊變數值乘右邊的值,再指定給左邊的變數
/=	將左邊變數值除右邊的值,再指定給左邊的變數
%=	將左邊變數值除右邊的值,再指定餘數給左邊的變數



Operator -- Logical Operator

運算子	說明
&	且(And)
	或(Or)

資料型態

DATA TYPE II



Data Type

Type	Example
list	[1, 2, 'YA']
tuple	(1, 2, 'YA')
dictionary	{'Key1': 'value1', 'Key2': 'value2'}



Data Type -- list

- ✓ 由 [] 將一串以 , 分開的值包起來

e.g. my_list = [1, 2, 'YA']

- ✓ list 中的元素不必放相同的資料型態
(int, float, boolean, string)

- ✓ Python index 從 0 開始

e.g. my_list[0] = 1
my_list[1] = 2



Data Type -- list

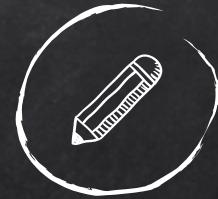
- ✓ `list.append(x)`

在list後面加入一個新的元素x，直接放入該元素

- ✓ `list.extend(x)`

在list後面加入一個新的元素x，先將元素展開再放入

int, string, list, tuple



Data Type -- list

✓ `list.insert(索引位置 , x)`

新增元素 `x`，並放在指定的索引位置之前

`int, string, list, tuple`



Exercise_2

- Q. 請將 weather_list[9] 中的 list ['North'] 後面
加入一個風力 2 級 的資訊
['North', 2]



Exercise_2

```
In [75]: weather_list[9].insert(len(weather_list[9]),2)
print(weather_list)
```

```
['Taipei', 18.5, 'rainy', '80%', ['Northeast', 4], 'Taichung', 21.5, 'cloudy', '73%', ['North', 2], 'Kaohsiung', 25.9
, 'sunny', '69%', ('Northwest', 3)]
```



Data Type -- tuple

- ✓ 由 () 將一串以 , 分開的值包起來

e.g. my_tuple = (1, 2, 'YA')

- ✓ tuple 跟 list 很像
- ✓ tuple 不能新增、刪除、修改



Data Type -- tuple

✓ `tuple.count(x)`

元素 x 在 tuple 中出現幾次

✓ `tuple.index(x)`

元素 x 在 tuple 的索引位置是哪裡



Data Type -- dictionary

✓ 由 {} 將一群由 Key,Value 配對組合而成的東西 包起來

e.g. my_dict = {'Key1': 'value1', 'Key2': 'value2'}

✓ Key有大小寫之分

✓ 不必放相同的資料型態

✓ Key不能重複，否則會被後面取代

✓ 沒有順序之分

資料型態轉換

DATA TYPE CONVERSION



Data Type Conversion

Type	說明
<code>int(x)</code>	<code>string , float</code> \rightarrow int
<code>float(x)</code>	<code>string , int</code> \rightarrow float
<code>str(x)</code>	<code>int , float , list , tuple , dict</code> \rightarrow string



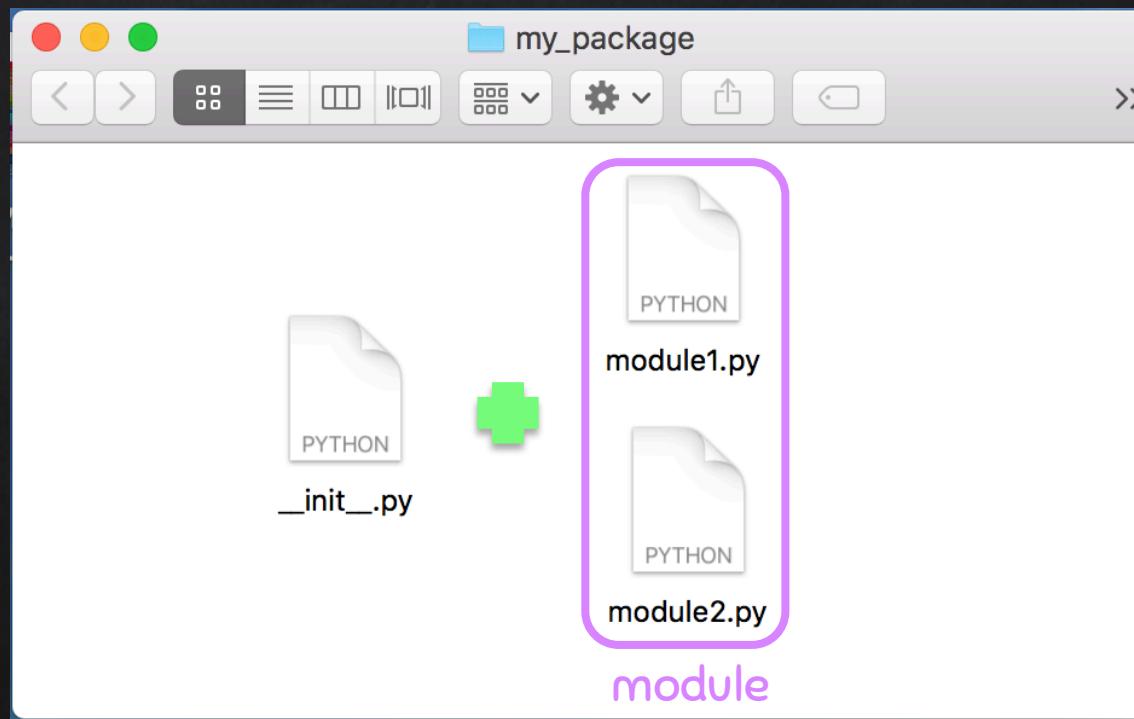
Data Type Conversion

Type	說明
<code>list(x)</code>	<code>string, tuple, dict</code> → list
<code>tuple(x)</code>	<code>string, list</code> → tuple

PACKAGE & MODULE



Package & Module



Numpy



Numpy

✓ `import numpy as np`

引入 numpy 套件

✓ `numpy.array(x)`

建立 numpy.array
[1, 2, 3]



Numpy

✓ `.mean()`

計算平均數

✓ `np.std(x)`

計算標準差



Homework

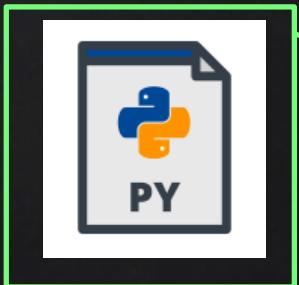
HW1. 請簡介一下你的背景

HW2. 利用 numpy 套件，計算高雄未來一週的平均溫度

Keelung = [22,20,21,18,18,19,21]

NOTE. HW_1_陳小明

主旨



HW_1_陳小明.ipynb