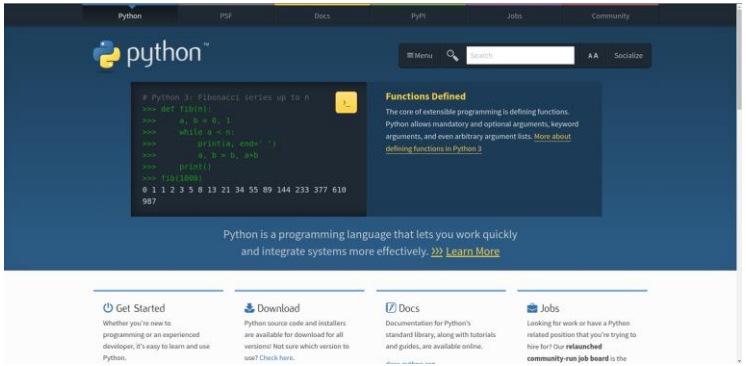
Lecture 1: Basics

python



- https://www.python.org/
- 廣為使用及開源的免費軟體之一
- 跨平台使用：Linux, Windows, Mac
- 可使用別人提供的library
- 亦可開放自己整合的library貢獻給開源社群

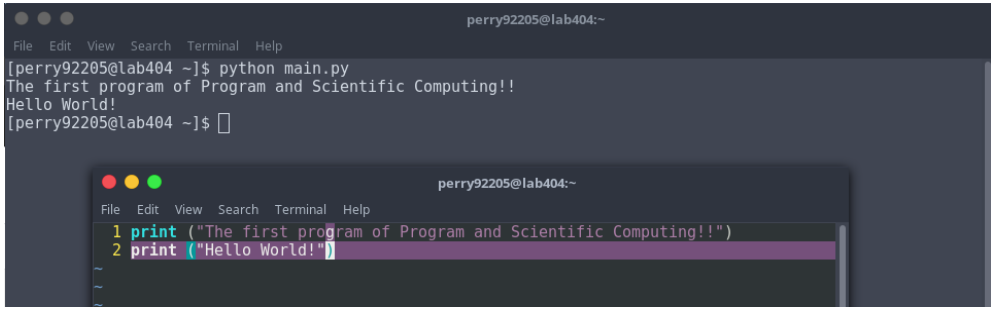
- 你可以在study主機上使用python，或在你的筆電上安裝、使用整合好的Anaconda

連到study執行python

- 互動式指令列：在>>>後輸入python指令，按下enter後馬上執行（結束時，按下ctrl + d）

```
Python 3.6.1 |Anaconda 4.4.0 (64-bit)| (default, May 11 2017, 13:09:58)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> print ("Hello World!!")
Hello World!!
>>>
```

- .py檔 (類似GrADS的.gs概念)
  - 先用nano編輯檔案，串接數個python指令（e.g. test.py)
  - 在unix下輸入 python filename (e.g. python test.py)



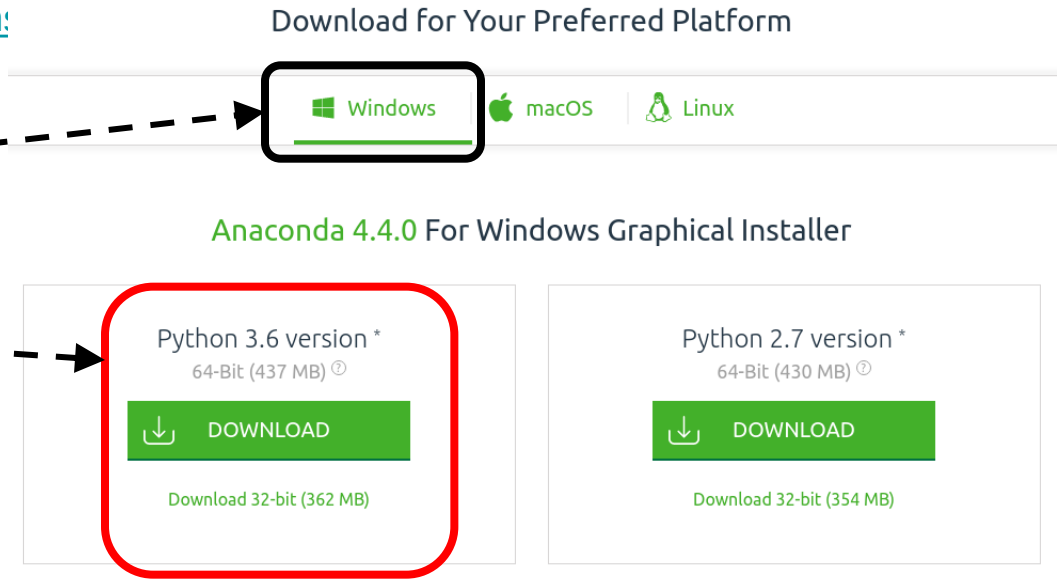
在你的筆電上：Let’s use Anaconda!

1. 已安裝好許多常用的科學、數學、工程、數據分析的 library
2. 完全開源(open)和免費(free)
3. 支持不同版本的 Python (2.7、3.6)，可自由切換 (-> 安裝不同版本即可使用)
4. 提供 spyder 視窗介面

下載 Anaconda

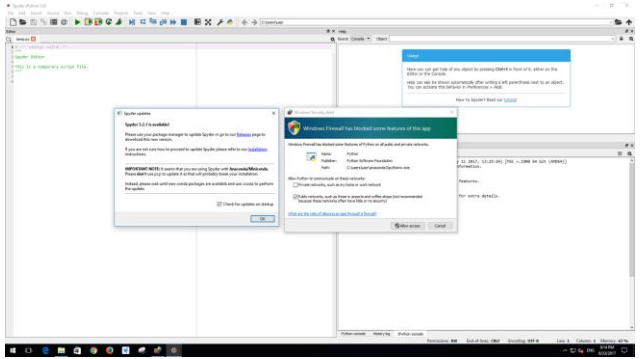
- download link: https://www.continuum.io/downloads
- 安裝教學
  - Windows: https://docs.continuum.io/anaconda/install/windows
  - Mac: https://docs.continuum.io/anaconda/install/mac-os
  - Linux: https://docs.continuum.io/anaconda/in

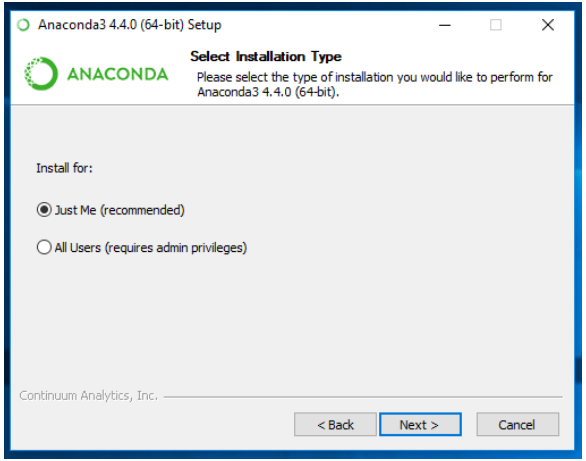
- 選擇自己電腦相對應的系統 (Windows/Mac)
- 選擇 python**3.X**的 64-bit version

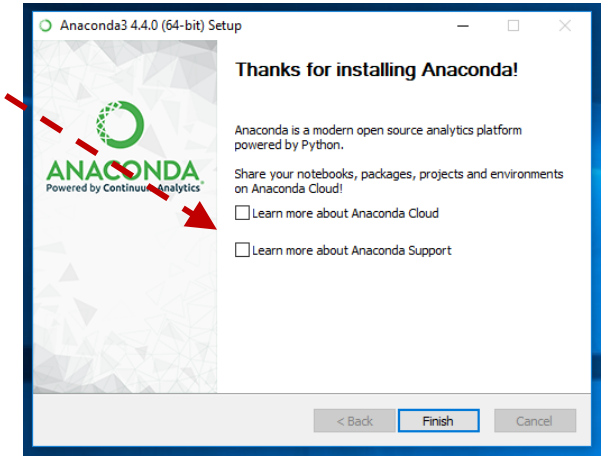


安裝 Anaconda

1. 雙擊下載好的程式
2. 可以直接套用預設值 (一直next)
3. 結束前，取消兩個選項
4. 執行spyder
   (在工作開始列搜尋”spyder”)
5. 出現如下畫面
   a. 右邊防火牆”允許存取”
   b. 左邊取消”check for updates”







spyder編輯器

- 左半邊編輯.py檔
  - 會有提示可用的function
  - 按下tab鍵也會出現提示可用的function
- 按工具列的綠色箭頭執行程式
- 右下角出現執行結果
- 右上角列出已存在的變數清單、指令查詢等

The Basics of Python:

Variables
Arithmetic Operations
IF
For Loop
Import module

Python: Basic syntax

- ”#”符號後是註解（comment），可以在一列的開頭或中間加入
- reserved word or built-in function (變數取名請避開！)
  - **and**, exec, **not**, **as**, finally, **or**, assert, **for**, pass, except
  - **break**, from, **print**, class, global, raise, **continue**, **if**, return
  - def, **import**, try, **del**, **in**, **while**, **elif**, **is**, with, **else**, lambda, yield
- 縮排視為不同的block (在IF判斷式或迴圈的段落中使用)
  - 縮排可以用tab或是數個空格(至少一個空格)。
  - 空格的數量不同，視為不同的block (bug很容易因為這一點而發生)
- python的每個變數視為一個object。（稍後說明）

Python variables

- 不需要事先宣告變數，直接用”=“ assign value（賦值）即可。
  - x=3.14 → 實數變數

- 變數類型（data type）根據被賦予的值決定（之後如果被assign不同類型的數值，該變數的類型就會直接改變）
  - x=‘text’ → 字串變數

- 確認變數的data type: type(x)
- 變數名稱中的大小寫要完全一致（a、A會當作不同的變數）
- python可以在同一個指令中對多個變數賦值
  - x, y = 2, 1 → x = 2, y = 1
- 若想要移除變數，使用del x y
- 常見的data type: number, string, list, tuple, boolean

Python variable types

- Number 數值
  - int : a = 11
  - float: a = 1.1e-18
  - complex: a = 4. + 7j
- String 字串 （ 用“ “ 或 ‘ ‘夾起）
  - e.g. x = “Hello World!” or x = ‘Hello World!’
  - 取出字串的局部：x [0:3]
    - 注意：index 從 0 開始計算！
    - [a:b] -- begin at index a and end before index b (e.g., x[0:3] -> “Hel”)
  - 不可以對字串的局部做更改，e.g. ~~x[0:3] =”Yo!”~~ (這個是錯誤的語法)
- Boolean 邏輯
  - 只有True, False兩種值，根據邏輯判斷（IF condition）的結果決定
  - 兩個boolean 變數做運算，會以True=1, False=0 做整數運算

Python variables: list and tuple

- list, tuple
  - 類似陣列的概念，但可以混雜儲存不同型態的資料，如下所示
  - List： x = ['abcd', 786 , 2.23, 'john', 70.2]
  - Tuple: y = ('abcd', 786 , 2.23, 'john', 70.2)
  - assignment: list 使用 [], tuple 使用(), 每個元素都用”，”分開
  - list 的大小及元素可以改變。tuple則不行 (類似常數陣列)
  - sub-list 或是 sub-tuple語法類似取出字串局部：
    x[0:2] → ['abcd', 786]
    y[2:4] → (786, 2.23)
  - 之後會對list有更多介紹

Convert data type

有時候需要不同型態的資料轉換，例如將字串”10”轉成整數10

下列為常用的built-in function

- int (x)
- float (x)
- str (x)

Arithmetic Operators

- 加法 **+**
  - string 相加：形成新的字串
    - x = “Hello”, y = “World”
    - c = x + y → c = “HelloWorld”
- 減法 **-**
- 乘法 ***
  - string 乘一個數字(int)：字串重複幾次
    - x = “Yo!”
    - y = x * 3 → y = “Yo!Yo!Yo!”
- 除法 **/**
  - 注意：兩個整數相除，結果是實數
    - x = 21, y = 10
    - c = x / y → c = 2.1
- 次方 ****
- 取餘數 **%**
- 整除至最近整數 **//**
  - 相除後取最接近的整數（整實數），結果的類型取決於兩個變數的類型：

9//2 = 4
9.0//2.0 = 4.0
11.0//3 = 4.0

Arithmetic Operators

- 如果要進行下面的運算 a= a+b （用a+b的結果為a重新賦值）
- 在python可以改寫成： a += b

- 所以算術運算符號可以有

  +=

  -=

  *=

  /=

  %=

  **=

  //=

Python built-in function

- **sqrt**(x)
- **abs**(x)
- **max**(x1, x2,...)
- **min**(x1, x2,...)
- **ceil**(x)：無條件進位
- **floor**(x)：無條件捨去
- **round**(x,n)：四捨五入到小數點下第n位（n可省略，則使用default n = 0）

- 下面函數必須先引入math工具才能使用，在.py檔先寫入
  import math
- math.**exp**(x)
- math.**log**(x)
- math.**log10**(x)
- math.**cos**(x), math.**sin**(x), math.**tan**(x), math.**acos**(x), math.**asin**(x), math.**atan**(x)
- math.**degrees**(x)：converts angle x from radians to degrees.
- math.**radians**(x)：converts angle x from degrees to radians.

Block IF and while

- **if condition1 :**
  statement(s)

  elif condition2 :
  statement(s)

  else:
  statement(s)

- **while condition :**
  statement(s)
  (repeatedly executes the statement(s) as long as the given condition is true)

- 邏輯判斷符號

==	(equal)
!=	(not equal)
>	(greater than)
>=	(greater than or equal to)
<	(less than)
<=	(less than or equal)

is, is not

- is : 條件比 ”==” 更為嚴格，必須連data type 都相同
- 右圖為範例
  (a = 4, data type : int)

- 邏輯運算

and	交集
or	聯集
not	非

```
>>> a = 4
>>> print (a)
4
>>> a is 4.
False
>>> a is 4.0
False
>>> a is 4
True
>>> a == 4.
True
>>> a == 4
True
```

IF and while: syntax

- 不同數量的空格（或tab）縮排，會被視為不同的if block
- 邏輯判斷式 可以不用加括號”()”，但是如果有多個或很複雜，建議加上括號以利閱讀
- 不管if, elif, else, for, while，都需要加上 :
- Statement如果很簡短，可以直接在:後面寫出，例如
  if a % 2 == 0: print (“a is even number”)
  (while、for loop也適用)

Example of “if condition”

```
1 year = 2017
2 print ("This year is %d" %(year))
3 if (year%400 == 0) or (year%4 == 0 and year%100 != 0):
4     print ("This year is a leap year")
5 else:
6     print ("This year is a common year")
7
```

```
[perry92205@lab404 ~]$ python main.py
This year is 2017
This year is a common year
```

For loop 迴圈

- **for x in [list] :**
  statement(s)

  (iterates over the items in a sequence, such as elements in a list or in a string)

- **e.g.,**
  for x in [1,2,3] :
  y=x+2
  print(y)

  也可寫成
  for x in range(1,4) :
  y=x+2
  print(y)

- **range**(…)
  經常用來控制迴圈的次數，但注意default是從0開始
- range(**n**)
  從0開始到n-1的整數
  e.g., range(3) → 0,1,2
- range(**nmin, nmax, interval**)
  從nmin開始，以間隔interval遞增到nmax-1的整數
  e.g., range(1,8,2) → 1, 3, 5, 7

Example for loop

- for char in message: 逐一取出message字串變數裡的元素 (字元)

```
File  Edit  View  Search  Terminal  Help
1 message = "Hello World!!"
2 for char in message:
3     print (char)
4
```

```
[perry92205@lab404 ~]$ python main.py
H
e
l
l
o

W
o
r
l
d
!
!
```

Simple input/output

- **input**(“…”) 在螢幕上顯示字串，並等待使用者輸入字串
  - x = input (“input your name: “) → 螢幕上會顯示訊息 input your name:，使用者輸入的內容，會用字串類型存到x變數
  - 就算使用者輸入數字，仍然是以字串類型儲存（之後會介紹如何切割字串（split），轉為數值）

- **print** (...) 顯示在螢幕上
  - e.g. print (“Hello World!!”)
  - 若要一次輸出多個變數至螢幕上，直接以逗號分開 e.g. print(x,y,z)
  - 也可以**format output**（之後介紹）

Python is an “object-oriented” programming language

- 生活中的例子：每個人都有自己的特徵以及可以從事的行動
  - 物件(object)：不同型態的人
  - 特徵(attribute)：人名、身高、膚色...
  - 工作(method)：寫作業、批改作業、出作業...
  - 不同型態的object有特定的attribute and method（例如：「學生」可”寫作業”但不能”批改作業”）
- 推演到python
  - 不同的data type視為不同的物件（object），譬如int, string
  - 變數儲存的”數值（或字串）”則為其attribute
  - 不同的object有各自的methods（會使用到這個物件的attribute的函式），使用dir(x)可以列出清單
  - 要使用物件x的B method，語法是：x.B(...)
  - 用help(x.B)可以查詢指令用法
  - __method__是特殊指令，一般不會用到

```
>>> x = "Hello World!"
>>> dir (x)
['__add__', '__class__', '__contains__', '__delattr__', '__dir__
'__getattribute__', '__getitem__', '__getnewargs__', '__gt__', '
ter__', '__le__', '__len__', '__lt__', '__mod__', '__mul__', '__
 '__repr__', '__rmod__', '__rmul__', '__setattr__', '__sizeof__'
casefold', 'center', 'count', 'encode', 'endswith', 'expandtabs'
num', 'isalpha', 'isdecimal', 'isdigit', 'isidentifier', 'islowe
tle', 'isupper', 'join', 'ljust', 'lower', 'lstrip', 'maketrans'
ust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', '
slate', 'upper', 'zfill']
>>> help (x.upper)
```

Modules or package（也稱為library）

- ”modules” or “packages”：通常是其他人寫好的函數或指令，打包成一個「工具庫」方便分享使用。（非python內建，可能會需要下載安裝）
  - 本課程會用到的library: numpy (array), matplotlib (基本繪圖), basemap (地圖), netCDF (讀nc檔)

- 使用方式：**import** module_name
  - 在執行工具庫內的指令之前，就要先把modules 引進來（不然python不會認得非內建的指令）
  - 使用工具庫內的指令：**module_name.function** (or **module_name.constant**)
  - 可以使用**import** module_name **as** xxx (xxx是你自己取的縮寫）
    使用指令的語法就簡化成xxx.function，例如
    import math as **m**
    x=**m**.cos(m.pi)

Online resource

官方教學文

1. https://docs.python.org/3.6/tutorial/index.html
2. https://www.tutorialspoint.com/python3/index.htm
3. https://docs.python.org/3.6/

其他網路資源/線上課程

1. http://tech-marsw.logdown.com/blog/2014/09/03/getting-started-with-python-in-ten-minute
2. http://blog.techbridge.cc/2016/12/17/python101-tutorial/
3. coursera: https://www.coursera.org/learn/python