Python

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”Anaconda” (整合安裝、編輯、執行python程式的介面)

連到study執行python: 基本指令

先進入python互動式環境，再輸入python指令

- 在unix/linux的指令模式，輸入**python** ，按下enter
- 進入python環境後，出現>>>等待你下python指令
- 在>>>後輸入python指令，按下enter後馬上執行並顯示結果
- 結束時，按下ctrl + d跳出python環境

```
[weitingc@study python]$ python
Python 3.4.5 (default, May 29 2017, 15:17:55)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-11)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> print('Hello World')
Hello World
>>>
```

The Very Basics of Python:

Print
Variables
Arithmetic Operations
Input from keyboard
Import module

基本指令: print(…) 顯示在螢幕上

- 在print函式中放進想要顯示在螢幕上的內容
  - 顯示一串文字 (字串，用單引號 ‘ ’包起來)
    print (‘Hello World!!’)
  - 顯示幾個數字，函式內以逗號分開
    print(3.1, 2, 4.5)
  - 顯示幾個變數所儲存的數值(或字串內容)
    x=3
    y=2
    print(x,y)

基本指令: 數值計算

- 在指令列可以輸入運算式進行數值的計算，類似計算機的功能，按下enter就會顯示結果
  - 加 **+**
  - 減 **-**
  - 乘 *****
  - 除 **/**
  - 次方 ******
  - 取餘數 **%**
  - 整除至無條件捨去的整數（整實數） **//**
  - () 內優先運算

```
>>> 2+3
5
>>> (2+3)*4
20
>>> (2+3)*4-0.5/2
19.75
>>>
>>> 2**2
4
>>>
>>>
```

基本指令: 變數，以及為變數【賦值】

- 變數就像是一個可以儲存資料的盒子
- 變數名稱是盒子上的標籤，例如 x
- 用 = 把資料存進變數裡，賦予變數內容，如 x = 3
  - 口語化的意思就是「令變數x為3」
  - 等號左邊是變數名稱，右邊是資料，
  - 資料的種類可以是字串、數字，或運算式
    - y = x + 3.1
    - z = ‘hello’
- 變數的「型態」就根據賦值的資料種類決定，
  - 可以用type查詢

```
>>> x = 3
>>> type(x)
<class 'int'>
>>>
>>> y = x + 3.1
>>> type(y)
<class 'float'>
>>>
>>> z = 'hello'
>>> type(z)
<class 'str'>
>>>
```

Python 變數型態: 數值與字串

- Number 數值
  - 整數int : a = 11
  - 實數float: a = 1.1e-18 (科學記號表示法，1.1 x 10^{-18})
  - 複數complex: a = 4. + 7j

- String 字串 （ 用“ “ 或 ‘ ‘夾起 ）
  - x = “Hello World!” or x = ‘Hello World!’
  - 取出字串的局部：x[0:4] → ‘Hell’ x[0:5] → ‘Hello’
    - 方括號內填入開始與結束+1的index，用 : 分開
    - 注意：index 從 0 開始計算！
    - 一旦賦值後，不可以對字串的局部做更改，e.g. x[0:3] =”Yo!” 是錯誤的語法!!

Python 變數的取名規則

- 變數名稱中的大小寫是有區別的（a、A會當作不同的變數）

- 變數取名請避開內建函數或指令，否則會造成錯誤
  - and, as, assert, break, class, continue, def, del, else, elif, except, exec, finally, for, from, global, if, import, in, is, lambda, not, or, pass, print, raise, return, try, while, with, yield…

變數可以進行運算

- x = -3.0
  y = 3.1
  print(x + y) → 0.1
  print(x // y) → -1.0
  print(x**2) → 9.0

  print(abs(x)) → 3.0
  print(max(x,y)) → 3.1

- **內建的基本數學函數(不用安裝module)**
- **abs**(x) : 絕對值
- **max**(x1, x2,...) : 找出最大值
- **min**(x1, x2,...) : 找出最小值
- **round**(x,n)：四捨五入到小數點下第n位（n若省略，則使用預設 n = 0 ）

一些關於變數運算的細節: 整數與實數混合運算

- 兩個整數相除，結果是<u>實數</u>
  - x = 21
    y = 10
    z = x / y
    print(z) → 2.1

- 整除到無條件捨去的最近整數（整實數） //
  - 結果的類型取決於兩個變數的類型：

(兩個都整數)	9//2	→ 4	(整數)
(兩個都實數)	9.0//2.0	→ 4.0	(整實數)
(混型運算)	11.0//3	→ 3.0	(整實數)

一些關於變數運算的細節: 字串的”運算”

- 字串 “相加”：形成新的字串
  - x = ‘Hello’
    y = ‘World’
    z = x + y
    print(z) → ‘HelloWorld’

- 字串乘一個整數(int)：字串重複幾次
  (特例，其他字串變數與數值變數的混型運算會得到錯誤訊息!)
  - x = ’Yo!’
    y = x * 3
    print(y) → ‘Yo!Yo!Yo!’

字串變數跟數值進行運算會造成錯誤!

會出現錯誤訊息

```
>>> x = '123.4'
>>> y = 100
>>> print(x+y)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: Can't convert 'int' object to str implicitly
>>>
```

(不要慌張，從錯誤訊息判斷可能出錯的原因)

將變數型態轉換，再進行運算

可以利用內建函數將不同型態的變數(資料)轉換，再做運算

- int (x) 轉為整數
- float (x) 轉為實數
- str (x) 轉為字串

```
>>> print( float(x)+y )
223.4
>>>
>>> print( x + str(y) )
123.4100
>>>
>>> print( int(x) )
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: invalid literal for int() with base 10: '123.4'
>>>
```

變數賦值與移除

- python可以在同一個指令中對多個變數賦值
  - a, b, c = 2, 1, 0 → a = 2, b = 1, c=0
  - 等號左邊是變數名稱清單，右邊是依序要儲存的資料，用逗號分開

- 若想要移除變數，使用del
  - del x,y,a,b,c

.py檔: 串接多個python指令的腳本(script)

- 先用nano(或任何文字檔編輯器)編輯一個純文字檔案，檔名要是.py結尾
- 檔案內容是數個python指令串接(一個指令一行)
- “執行” .py檔: 直接在 unix/linux指令列，或使用spyder (後面說明)

- 例子: 用nano看看square.py的內容

```
# This is an example of python code
#
# Let the user input one number and output
# the square of the number on screen
#
# Wei-Ting (2018/09/14)

# user input
x=input('Please input a number:')

# conver string to float number
y=float(x)

# output result of square on screen
print('Its square is',y**2)

~
```

連到study執行python: .py檔

不用進入python環境，直接”執行”.py檔

- 在unix/linux的指令模式，輸入**python** xxx.py 按enter 就可依序執行檔案內的這串指令(e.g. python square.py)

```
[weitingc@study python]$
[weitingc@study python]$ python square.py
Please input a number:1.5
Its square is 2.25
[weitingc@study python]$
```

input 讓使用者輸入字串，為變數賦值

- **a = input()**
  會等使用者輸入(輸入後按enter)，並以字串型態存入 a 這個變數中

- **b = input**(“name:”)
  螢幕上顯示訊息 “name:”，使用者輸入的內容，會用字串類型存到b變數

- 就算使用者輸入數字，仍然是以字串型態儲存。可以之後用int(a), float(a)轉換型態

.py檔基本規則: # 註解符號

- **#** 符號之後的文字會被判斷為註解，不會當作指令執行
(可以在一行的開頭或中間加入)

- 為什麼要寫註解?
  - 在寫程式的時候先掌握基本架構（程式的草稿）
  - 讓寫程式的人可以對程式內容進行口語化的說明、提醒，幫助自己或他人日後使用、修改時方便參考、回憶

- 可以考慮寫成註解的事項
  - 在程式開頭，解釋整個程式的用處、基本架構、版本、更新日期
  - 列出各變數代表的意義，與相對應的單位
  - 解釋每一段程式要執行的功能、計算相對應的原始方程式
  - 其他對自己或程式讀者重要的提醒

.py檔基本規則: 縮排對齊

- 縮排空格的數量如果不同，會被判定為不同的段落
  (主要在IF判斷式或迴圈的段落中使用)

- 隨興縮排可能造成執行錯誤→養成每行開頭對齊的習慣!

預習問題—第二部分

- 用nano或其他編輯器打開square.py檔案，瀏覽裡面的程式。

(1) 變數x是什麼類型？你是怎麼判斷的？

(2) x = input('Please input a number:')這行，會讓程式執行時發生什麼事情？

(3) 第一行開頭的 #，有甚麼作用?

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: math (數學), numpy (array), matplotlib (基本繪圖)

- 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**)

import math

下面函數必須先引入math工具才能使用，在.py檔開頭先寫入
import **math**

- math.**sqrt**(x)：平方根
- math.**ceil**(x)：無條件進位
- math.**floor**(x)：無條件捨去
- math.**exp**(x)：指數函數
- math.**log**(x)：自然對數，math.**log10**(x)：以10為底對數
- math.**cos**(x), math.**sin**(x), math.**tan**(x), math.**acos**(x), math.**asin**(x), math.**atan**(x)：三角函數
- math.**degrees**(x)：converts angle x from radians to degrees.
- math.**radians**(x)：converts angle x from degrees to radians.

import module with abbreviation

- 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)

在你的筆電上安裝使用 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: https://www.anaconda.com/distribution/#download-section

- 安裝教學
  - Windows: https://docs.anaconda.com/anaconda/install/windows/
  - Mac: https://docs.anaconda.com/anaconda/install/mac-os/

Anaconda Installers

- 選擇自己筆電相對應的作業系統

Windows

Python 3.8

64-Bit Graphical Installer (466 MB)

32-Bit Graphical Installer (397 MB)

MacOS

Python 3.8

64-Bit Graphical Installer (462 MB)

64-Bit Command Line Installer (454 MB)

Linux

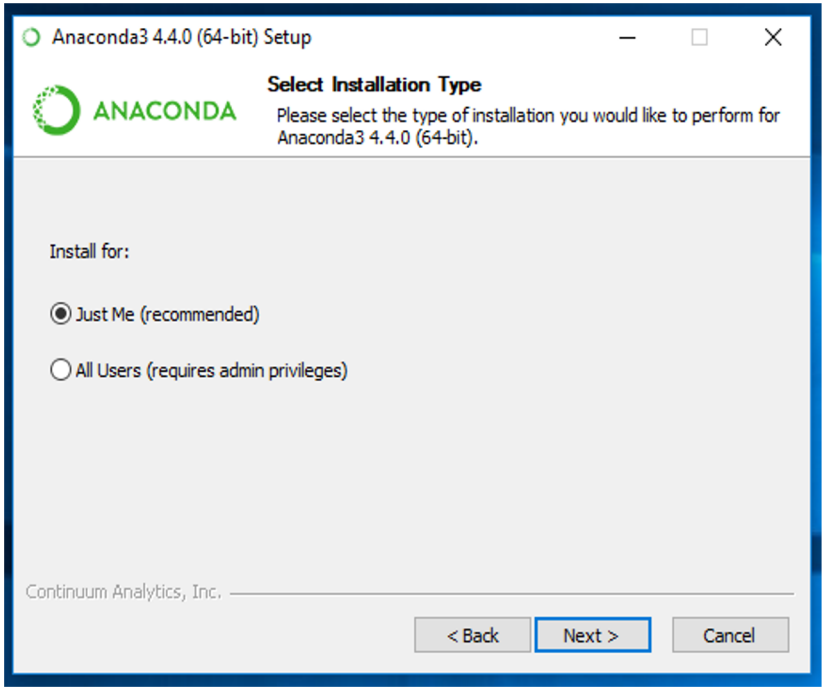
Python 3.8

64-Bit (x86) Installer (550 MB)

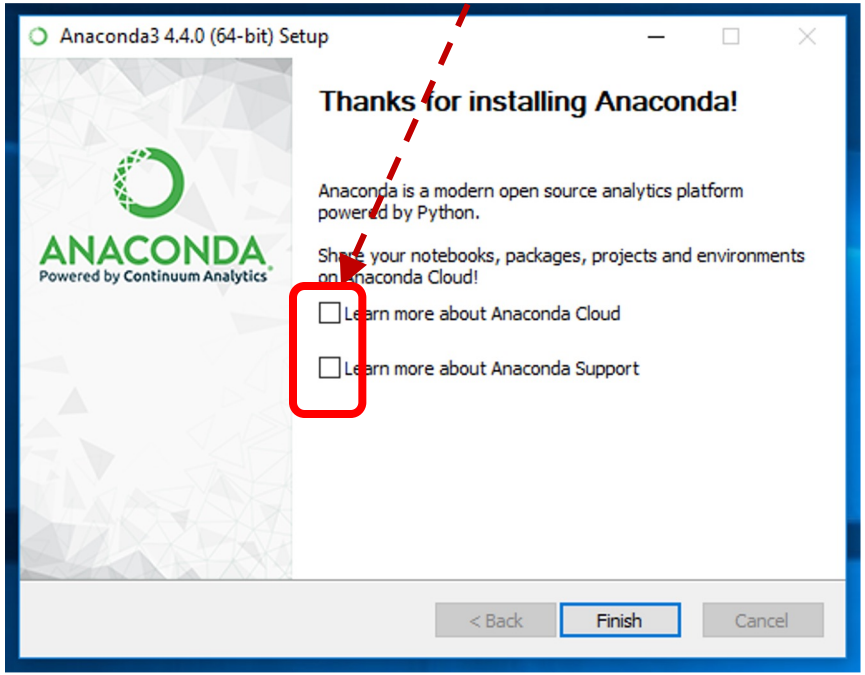
64-Bit (Power8 and Power9) Installer (290 MB)

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. 按Finish之前，**取消**兩個選項



spyder編輯器

安裝好Annaconda之後，會一併裝好spyder

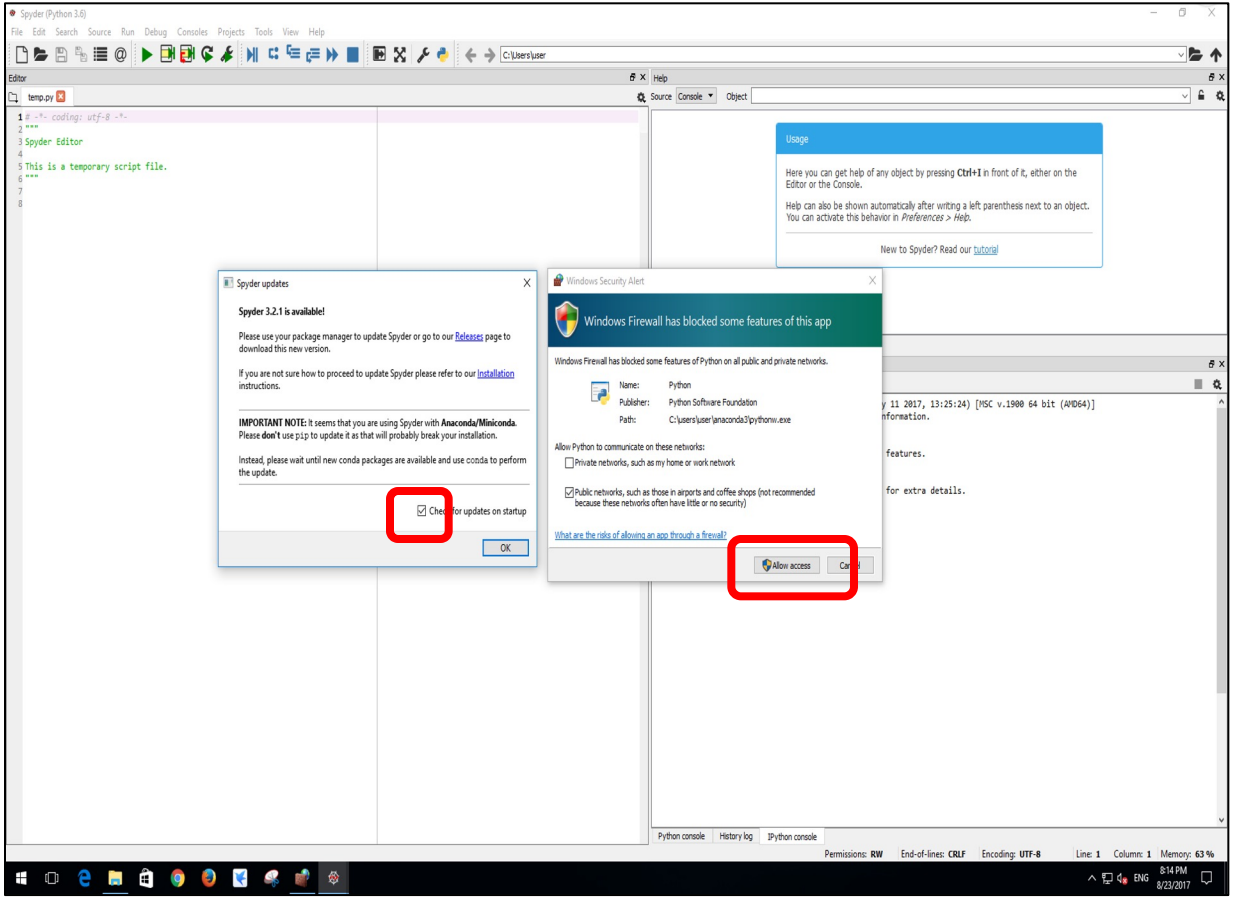
1. 在工作開始列搜尋spyder

2. 點icon執行

3. 出現如右畫面

a. 右邊的防火牆設定”允許存取”

b. 左邊取消”check for updates”，按OK



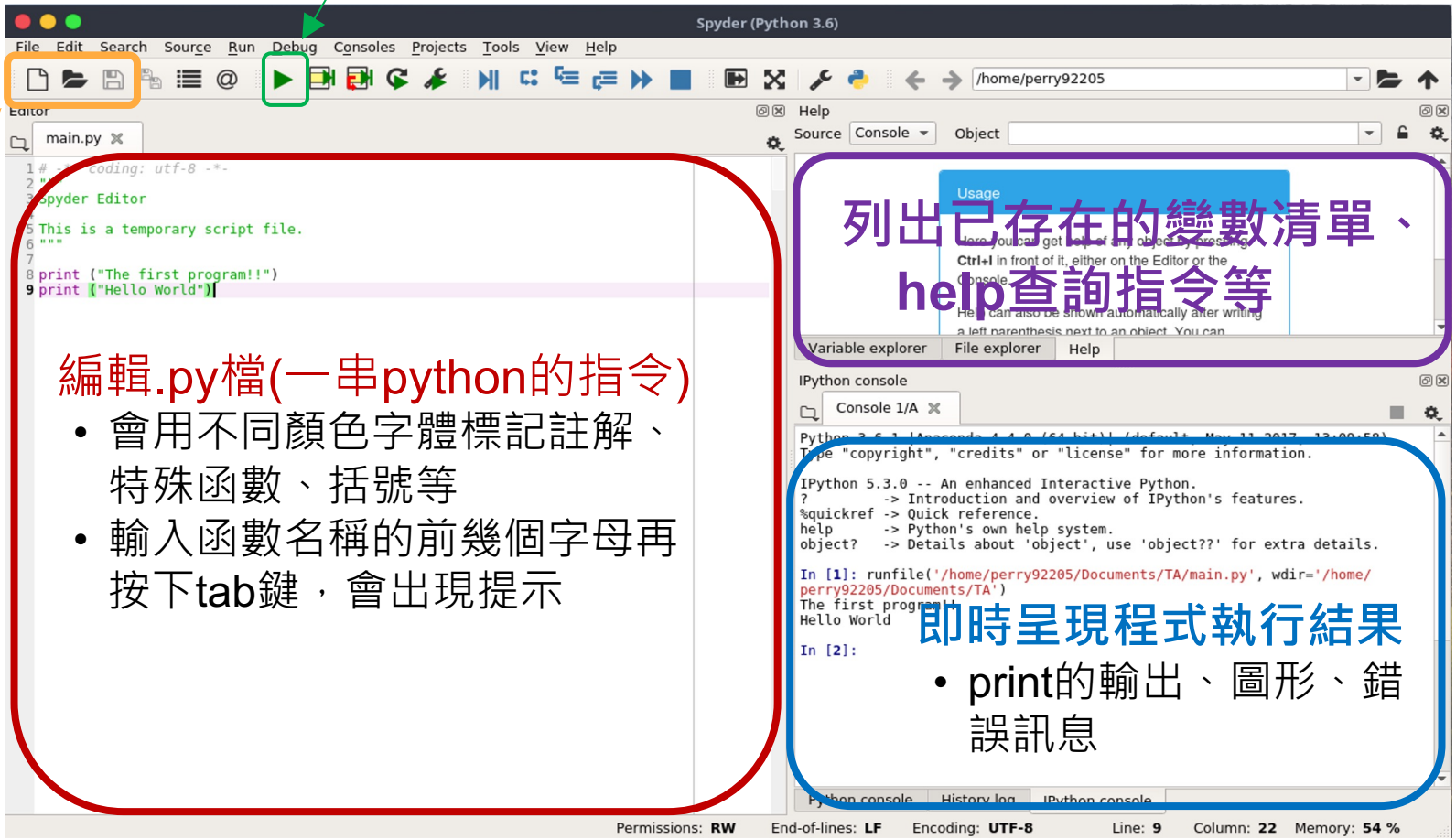
Spyder介面

點工具列的**綠色箭頭**
依序執行.py檔的指令

開啟新的
.py檔

開啟存在
的.py檔

存檔



列出已存在的變數清單、help查詢指令等

編輯.py檔(一串python的指令)

- 會用不同顏色字體標記註解、特殊函數、括號等
- 輸入函數名稱的前幾個字母再按下tab鍵，會出現提示

即時呈現程式執行結果

- print的輸出、圖形、錯誤訊息

體驗spyder介面

- 用mobaxterm從study上把 square.py下載到你的筆電
  - 用spyder開啟並執行square.py

- 寫一個新的 CtoF.py，
  - 螢幕上顯示字串 ‘please input a temperature in Celcius’，接收使用者輸入一個數值(攝氏溫度)
  - 計算出對應的華氏溫度，將數值顯示在螢幕上
  - 執行並測試結果是否正確

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
  - 不同變數型態為不同的物件（object），譬如int, string
  - 物件（變數）儲存的”數值（或字串）”的特徵則為其attribute
  - 要顯示物件x的attribute A，語法是： x.A
  - 不同物件有各自的methods（會使用到這個變數數值的函式），
  - 要使用物件x的method B，語法是： x.B(...)
  - 用dir(x)可以顯示物件x相關的attribute and method
  - 用help(x.B)可以查詢method指令用法 (e.g., help(math.atan))

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

運算式的簡化

- 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作業繳交注意事項

- 所有檔案（程式、圖檔、資料檔） 必須上傳到study主機上的指定位置
- 例如本週作業請放在 ~/hw1/
  - 先移動到你的家目錄(**cd**按enter，可用**pwd**確定是在/home/B09/b092090xx/下面）
  - 用**mkdir hw1**創出新目錄
  - 再**cd hw1**移動到新目錄裡面，開始編輯程式，或上傳程式到這裡
- 評分是根據程式在study上執行的結果，所以上傳後務必在study上測試執行。

- 請設定你的資料夾權限，讓助教可以進去改作業:

```
chmod +rx /home/B09/b092090xx
```

(你的學號)

- Mac系統從study主機把檔案從筆電上傳到study的方法 (xx是你的學號)
  - 在terminal 打下面的指令

```
scp /Desktop/hw1b.py b092090xx@140.112.66.200:/home/B09/b092090xx/hw1/.
```

要上傳的檔案存放在筆電的位置

要上傳到Study主機上的位置

再輸入study主機的密碼

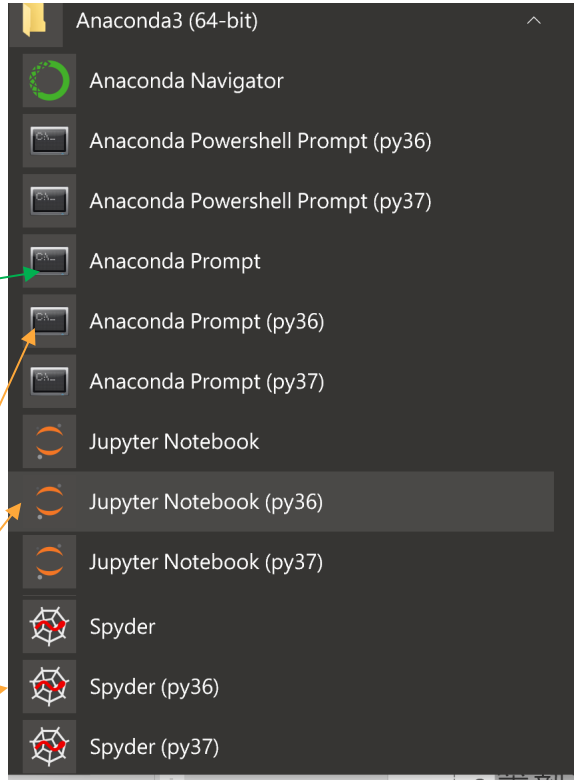
- Mac系統從study主機把檔案下載到筆電的方法
  - 在terminal 打下面的指令

```
scp b092090xx@140.112.66.200:/home/B09/b092090xx/hw1/hw1.py /Desktop/.
```

要下載的檔案在Study主機上的位置

下載後檔案要存放在筆電的位置

再輸入study主機的密碼

如何更改Anaconda中的Python版本?
(例如: 從python3.7 改成 3.6)

- 有些library或package只支援特定版本的python，如果程式中使用到這些library，就要在這個版本的python環境中才能正確執行
- 在你的筆電創造另一個python版本的環境:
  - 在Anaconda選單中點選Anaconda Prompt，叫出指令介面
  - 依序輸入下面的指令 (下面的例子是建立python 3.6的環境，py36是自己取的名稱)
    - python --version
    - conda create -n py36 python=3.6 anaconda
    - conda activate py36
- 等執行完畢後，Anaconda的工作列選單會出現python 3.6的prompt, jupyter notebook與spyder