

Python基礎 & rabboni

講師：隋建德

Python Installation



- Python的安裝有幾種方式
 - Python 官網下載
 - <https://www.python.org/downloads/>
 - 額外的library 需要PIP 安裝
 - Anaconda
 - <https://www.anaconda.com/distribution/>
 - 懶人包，內有許多Library

Python 版本



- Python 2.X or Python 3.X?
 - 建議安裝Python3
 - 已有許多lib不對Python2進行更新

Python的特色

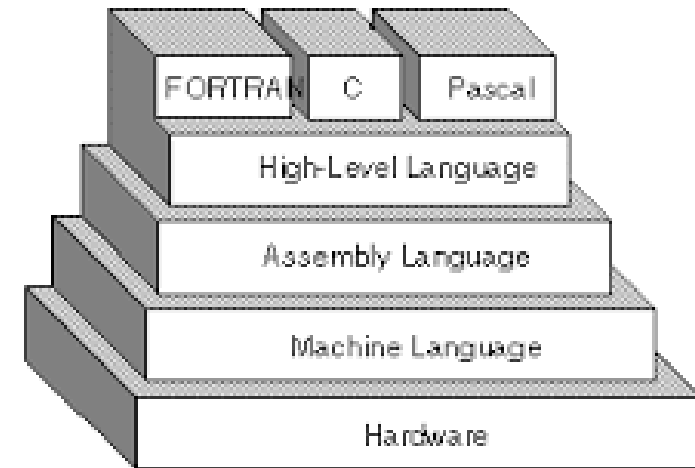


- 易學
- High-Level 程式語言
- Interpreted language
 - Line by line
- Cross-platform Language
 - Run on Windows, Linux, MacOS
- Free and Open Source
 - 有很多Open Source可使用
- GUI Programming Support
 - 可以寫GUI介面程式

High-Level?



- Low-Level
 - Assembly Language
 - Machine Language
 - 主要給電腦、機器看的語言
 - 不易讀
- High-Level
 - C++,python,java
 - 易讀
 - 一行High-level其實內涵數行low-level



Interpreted Language?



- 一行行執行
 - Python, C++

```
1  a = 0
2  a = a + 1 ## a = 1
3  b = a + 1 ## b = 2
```

- 非一行行執行
 - Verilog

```
1
2  a = 0
3  #-----
4  a <= a + 1 ## a = 1
5  b <= a + 1 ## b = 1
6  #-----
-
```

Hello World



- 建立一個檔案 Hello.py
- 打入
 - `print ('Hello World')`
- 執行
 - `python Hello.py`
- 會顯示
 - Hello World

Coding IoT 2.0

Python的資料型態

Data Type



- Dynamic Type

- 不需要像C等語言需指定變數的型態
- $A = 4$
- $A = \text{'SSS'}$
- 可以在程式碼中改變資料型態
 - 注意: 因此要注意使用的變數，否則會很容易改動到以往的變數

- Strong Type

- 不同型態的變數不能混用
- $1 + \text{"SS"} = ?$

數值型態



- 數字
 - Int
 - Float
- Operation
 - +
 - -
 - *
 - /
 - //
 - %
 - **

Sequence Types



- String
 - 字串
- List
 - 序列、列表、串列
- Tuple
 - 多元組

String



- 字串
- 以 `"""` 或 `"` 隔開來
- `A = "Hello"`
- `"""` 與 `"`
 - 無差別
 - 主要可拿來混用
- `A = "Owen is 'Excellent' !"`
- 可串接
 - `C = "Hello" + "World" ## C = "HelloWorld"`

List



- 物件的列表
- 內容物可以任意長度、型態
- `A = [000, " AioT" , 666]`
- 可以動態刪減新增

Tuple



- (000 , "AioT" , 666)
- 不可刪減新增
- 佔用空間較小

List常用操作：Indexing



- List 以0為起點
 - `A = [000, "AioT", 666]`
 - `A[0] = 000`
 - `A[1] = "AioT"`
 - `A[2] = 666`
- 可取倒數項目
 - `A = [000, "AioT", 666]`
 - `A[-1] = 666`
 - `A[-2] = "AioT"`

List常用操作 : Slicing



- 取段落項目,含起點不含終點
- `A = [000 , "AioT" , 666]`
- `A[0:2] = [000 , "AioT"]`
- `A[0:] = [000 , "AioT" , 666]`

List 常用操作：刪減新增



- `A = []`
 - 給空list
- `A.append("AioT")`
 - `A = ["AioT"]`
- `A.append(666)`
 - `A = ["AioT", 666]`
- `del A[1]`
 - `A = ["AioT"]`

List 常用操作：刪減新增



- `A = ["AioT"], B = [" is ",666]`
 - 兩個不同list
- `A.extend(B)`
 - `A = ["AiOT"," is ",666]`

List 常用操作 : 刪減新增



- `A = ["AiOT"," is ",666]`
- `A.remove(" is ")`
 - `A = ["AiOT",666]`
- `A.insert(" is ",1)`
 - `A = ["AiOT"," is ",666]`

List 常用操作：其他



- `A = [4,2,3,6,7,1]`
- `A.sort()`
 - `A = [1,2,3,4,6,7]`
- `A.reverse()`
 - `A = [7,6,4,3,2,1]`

Data Type : Dictionary



- Dictionary 含有兩元素
 - Key 跟 Value
- $D = \{ \text{"Owen"} : 90, \text{"Cindy"} : 100 \}$
- $D[\text{"Owen"}] = ?$
 - $D[\text{"Owen"}] = 90$

Dictionary 操作



- `D = { "Owen" : 90 , "Cindy" : 100 }`
- 更新
 - `D ["Owen"] = 100`
 - `D = { "Owen" : 100 , "Cindy" : 100 }`
- 新增
 - `D ["Andy"] = 40`
 - `D = { "Owen" : 100 , "Cindy" : 100 , "Andy" : 40 }`
- 刪除
 - `del D ["Andy"]`
 - `{ "Owen" : 100 , "Cindy" : 100 }`

Coding IoT 2.0

Python語法

縮排、斷句



- 換行
 - C++ : ; 分號代表換行
 - Python : 不用分號
- 判斷迴圈內外 (if, while, for...etc)
 - C++ : 使用{} 判斷
 - Python : 使用縮排判斷 建議使用4個space 非 Tab

If 判斷式



If condition :

.....

elif condition:

.....

else:

.....

If 判斷式 範例



```
score = 59
```

```
if score >= 60:
```

```
    print ("及格")
```

```
elif score > 50 :
```

```
    print ("差一點!")
```

```
else:
```

```
    print ("尚須努力!")
```

If 判斷式 – 多層



```
If condition :  
    if condition :  
        .....  
    else:  
        .....  
elif condition:  
    .....  
else:  
    .....
```

While 迴圈

While condition:

.....

小心使用，condition 沒設好會無限



While 迴圈 範例



```
a = 0
while a < 10:
    a += 2
    print ("a=",a)
```

```
a = 2
a = 4
a = 6
a = 8
a = 10
```

For 迴圈



- For in range
 - for 變數 in range(起點,終點,間隔):
 - for 變數 in range(終點):
 - 起點=0,間隔=1
 - 不含終點
- For in enumerate
 - 在list上使用
 - for item in list :
 - for index, item in list :
- For in dictionary
 - for key in dictionary:

For 迴圈 – range 範例



```
for i in range(0,10,2):  
    print (i)
```

0 2 4 6 8

For 迴圈 – enumerate 範例



```
a = ["AioT", 666, 777]
for index, item in a:
    print (index,":",item)
```

0 : AioT

1 : 666

2 : 777

For 迴圈 – dictionary 範例



```
D = {"Owen" : 90, "Cindy": 20}
```

```
for key in D :
```

```
    print (key, " = ", D[key])
```

```
Owen = 90
```

```
Cindy = 20
```

function



- 使用場合
 - 將常用的function 獨立出來
 - 將過長的片段獨立出一個function，使程式碼易讀
- 養好將必要片段獨立出來的習慣！
 - 使程式碼易讀性高
 - 利於團隊合作

Function 例子



```
def square(x):  
    y = x*x  
    return y  
print ("ans :",square(2))
```

ans : 4

Function



```
def square(x,y=2):
```

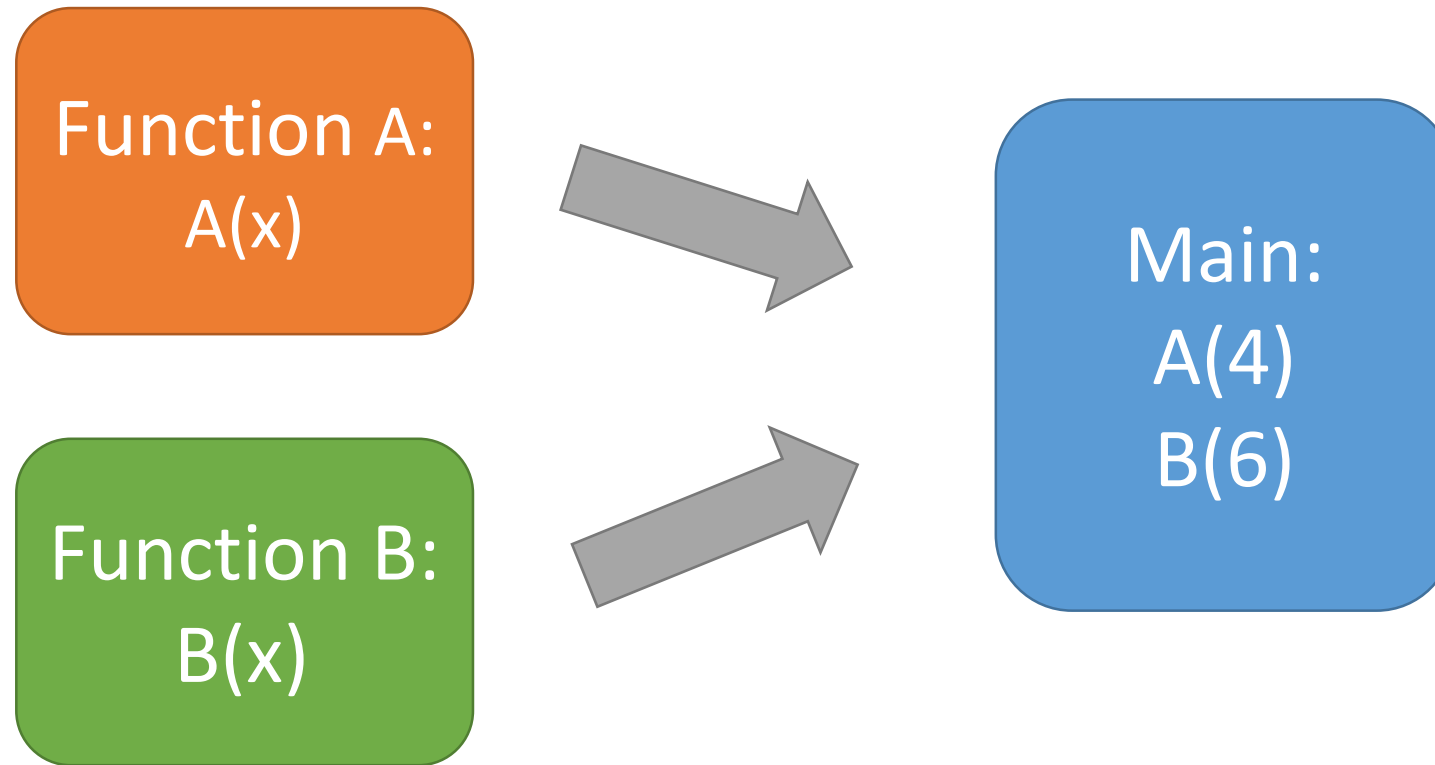
```
    y = x*x+y
```

```
    return y
```

```
print ("ans : ",square(2))
```

```
ans : 6
```

Function flow & block



Packages and module

Packages & module?



- Module
 - 一個獨立的.py檔就可以是一個module
- Packages
 - 內含許多module
 - 可從pypi上下載第三方開發的packages

Module



- 兩個檔案
 - Add.py
 - 加法module
 - Main.py
 - 主要程式碼
- 資料夾/
 - Add.py
 - Main.py
- 可將不同module視為不同function
 - 使用module可增加易讀性

呼叫 (Main.py)

```
import Add
```

```
.....
```

```
from Add import f
```

```
.....
```

```
import Add as a
```

```
.....
```

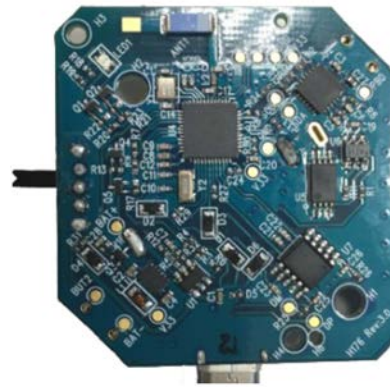


Package



- 下載安裝
 - 打開 命令提示視窗
 - 輸入 `pip install package`
- 常見package
 - numpy
 - array 內有許多優於list的function
 - matplotlib
 - 畫圖用
 - **rabboni**
 - 由愛物課程團隊開發完成
 - 專門給rabboni使用!

Rabboni



Rabboni 簡介



- 內含六軸感測器(sensor)
 - 3-axis 加速度計
 - 3-axis 陀螺儀
- 支援兩種連接方式，可將資料實時(Real-time)讀出
 - 藍芽無線連線
 - USB傳輸線
- 內涵Counter
 - 可自訂counter的threshold
 - threshold為加速度三軸的平方開根號
 - 分 目前counter與保存counter兩種，後者會存在裝置內

藍芽模式

- 將dongle藍芽接受器插上電腦USB
 - 開啟rabboni，按下藍芽廣播鍵
 - 執行Python程式
-
- 有效距離
 - 約10m，依環境而定



USB模式

- 將Rabboni使用USB插上電腦
- 執行Python USB相關code
- 活動性較差(有線)



如何使用Python操作Rabboni



- 安裝Rabboni library
 - `pip install rabboni == 1.73` (1.73版本)
- 此library僅限rabboni使用
 - 非rabboni裝置，library會無效
- Rabboni library 可以做到什麼事？

Rabboni Library API

- 能夠做到?(基本)
 - 掃描與顯示附近藍芽裝置
 - Rabboni連線/斷線
 - 取得連線狀態
 - 讀取六軸資料
 - 讀取counter的數值
 - 重置counter
 - 設定counter 的 threshold
 - 將接收的資料寫入csv檔案
 - 將接收的資料輸出圖檔
 -等



擁有Rabboni能幹嘛？

- Sensor相關都可以做到！
 - 開車遊戲
 - 利用Gyro轉動量
 - 貪吃蛇方向遊戲
 - 利用Acc方向
- 進階？
 - 利用AI計算人難以計算的目標
 - 游泳泳姿？
 - 跑步姿勢？
 - 步長？
 - 睡眠品質？

