

Maze Demo



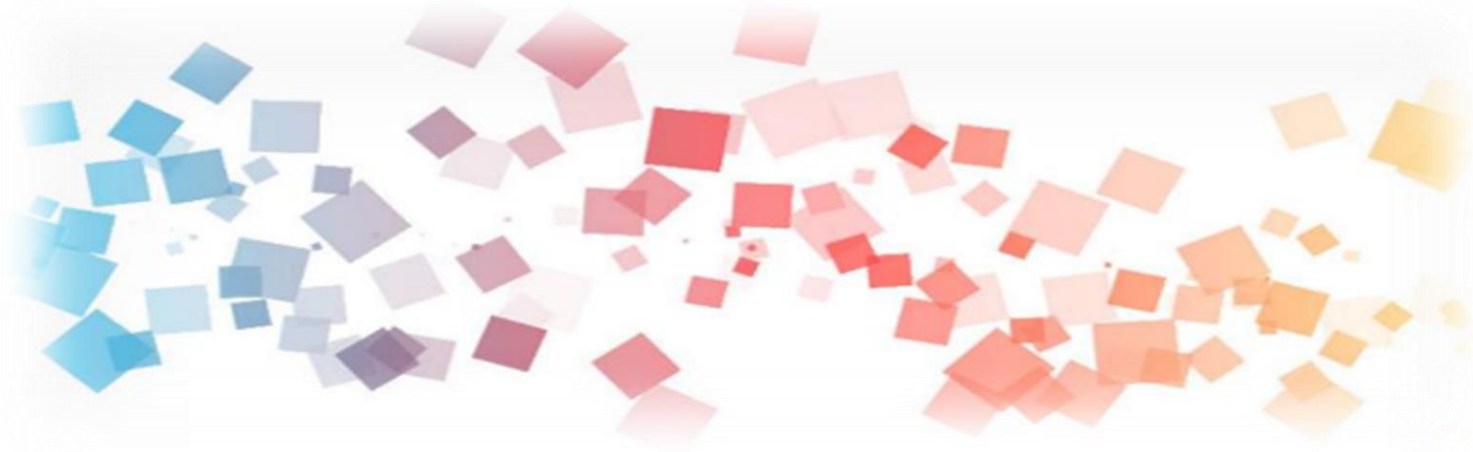
## AIOT Coding 智慧物聯- Python基礎 & rabbit

Python 簡介以及連結rabbit的程式設計

Course ID: 20pythonF1

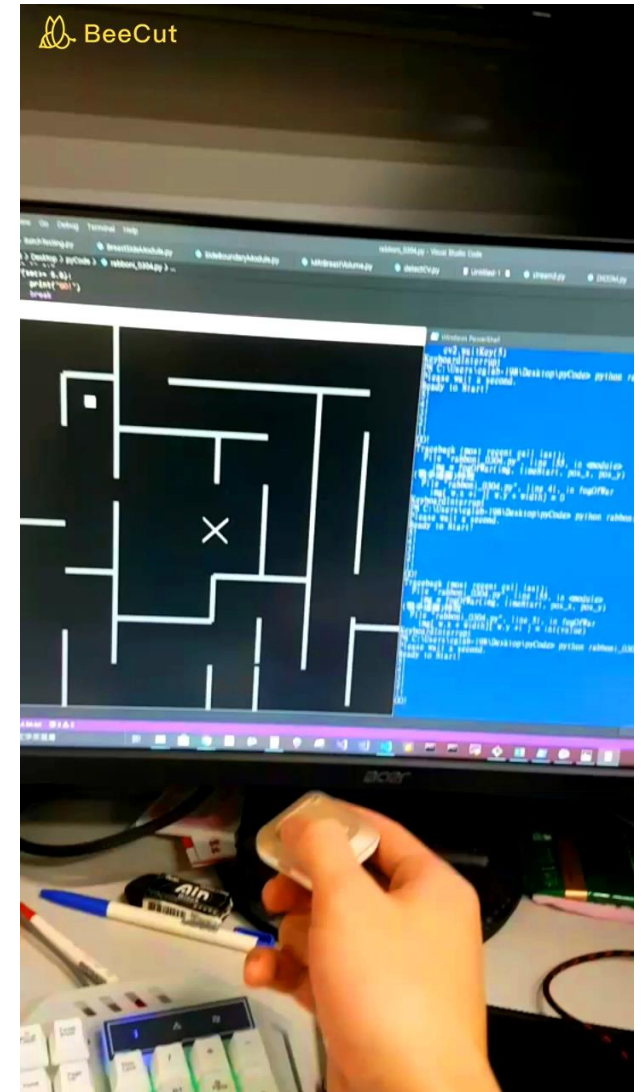
國立交通大學電子工程系教授  
國立交通大學半導體學院副院長  
國立交通大學策略辦公室執行長

[twtstella@gmail.com](mailto:twtstella@gmail.com) 03-5731664

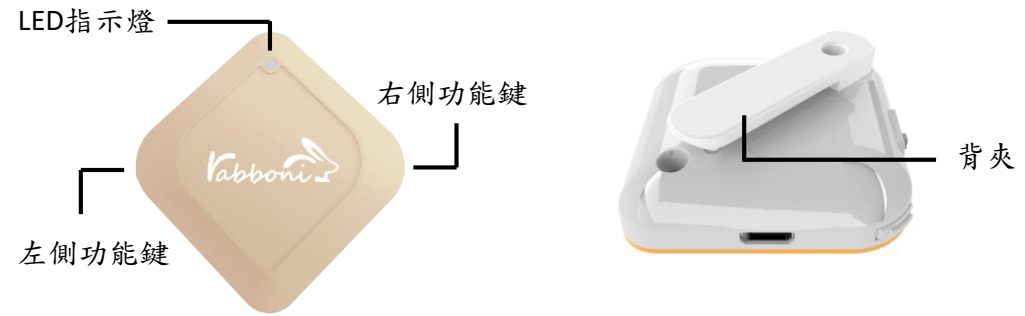


## Rabboni - Maze Demo

蔡孟哲 - 資科工所



## yabboni-Usage



- yabboni內建六軸重力感測器 (IMU: Initial Measurement Unit)、BLE藍芽傳輸及運算元件
- 可即時傳輸感測讀值並提供取樣頻率及動態範圍之多樣選擇
- 配有LED燈，指示yabboni運作狀態及電量顯示。
- yabboni 提供Android感測訊號擷取APP及各式程式教育應用 API
- Scratch, Python, Unity, Java, App Inventor
- 專為 AIoT 程式教育、APP開發、AI智慧感測互聯或各種智慧化應用之動作偵測相關研究開發使用。

# yabboni-感測參數介紹

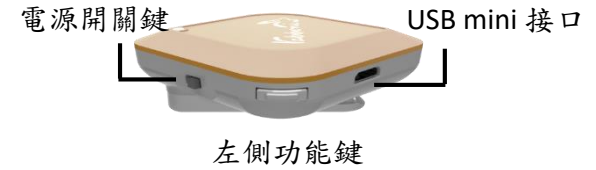
Gyro Full Scale Range	Gyro Sensitivity	Accel Full Scale Range
(°/sec)	(LSB/°/sec)	(g)
±250	65.5	±2
±500	32.8	±4
±1000	16.4	±8
±2000	8.2	±16

電池容量	120mAh 鋰離子充電電池
充電方式	USB mini 充電
無線傳輸	Bluetooth 4.0 BLE
充電時間	30分鐘
待機時間	5天 (電源開關鍵OFF)
連續使用時間	8 小時
支援作業系統	藍芽：Android USB：系統Windows 7以上

為了提高可靠性，還可以為每個軸配備更多的傳感器。一般而言IMU要安裝在被測物體的重心上。

# yabboni-操作功能介紹

電源開關鍵	單刀開關	On/off 標示
左側功能鍵	(短按1秒)	計數紀錄開始與結束(LED紅燈)
右側功能鍵	(短按1秒)	藍芽廣播開啟，與藍芽裝置配對(LED綠燈)
	(長按5秒)	電量顯示
LED電量指示燈號	(紅)	錄影指示燈、電量小於30%
	(橘)	關機指示燈、電量小於70%
	(綠)	配對指示燈、電量大於70%



[綠燈閃爍]藍芽廣播中



[紅燈閃爍]計數記錄中



[長按右鍵5秒]可以確認電量狀態



電量大於70%



電量介於70% 到30%



電量小於30%

# yabboni-配件介紹



yabboni本體 (正面)

yabboni背夾(拆卸須將螺絲工具)



提供使用者跑步或行進間  
yabboni主體與鞋面穩固  
結合，確保動作的正確偵測。



yabboni本體 (背面)

魔鬼氈手腕帶 · 寬2公分、長27.5公分



提供使用者跑步或行進間yabboni主體  
與鞋面穩固結合，確保動作的正確偵測。



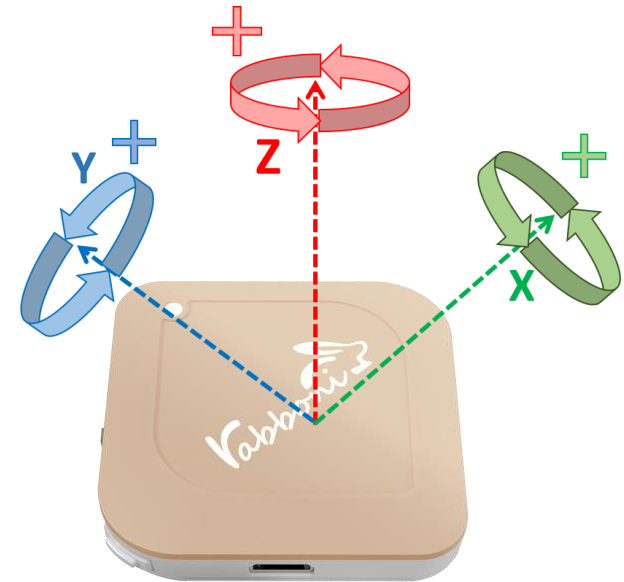
USB Type A轉接 USB mini線 ·  
可提供傳輸數據以及充電功能。

USB轉接線一條

# yabboni-軸向定義

直線軸：X/Y/Z加速度 (Acceleration)

環狀軸：X/Y/Z 角速度 (Gyro)





## Install

# yabboni PC UI 連線

1. rabboni\_pc\_UI下載/解壓縮資料夾(rabboni\_PC\_ui) :

<https://reurl.cc/QprO60>

2. 解壓縮檔中找到/建立捷徑

3. 執行

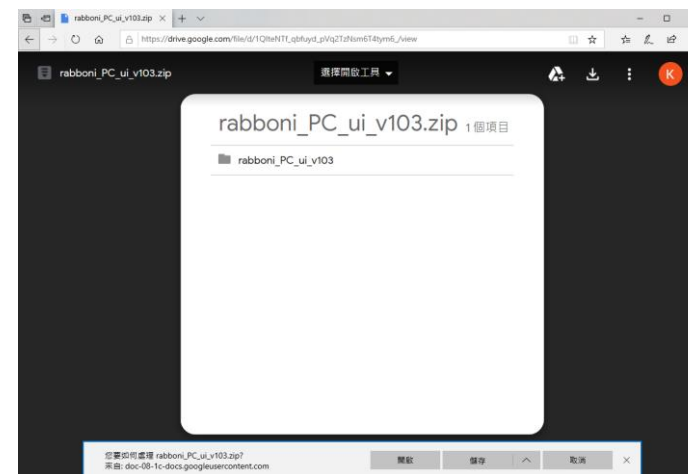
3

rabboni-PC

- rabboni-PC.application
- rabboni-PC.exe
- rabboni-PC.exe.config
- rabboni-PC.exe.manifest

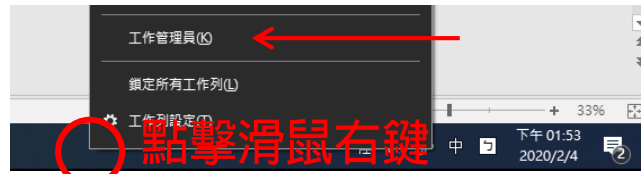
下載並解壓縮檔案 2

rabboni\_PC\_ui\_v103.zip

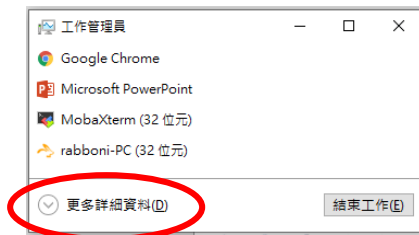


## 如果yabboni PC UI 連線程式無法開啟

1. 執行工作管理員 (在工作列上按右鍵或同時按下Ctrl+Alt+Del，選擇”工作管理員”)



2. 點擊「更多詳細資訊」

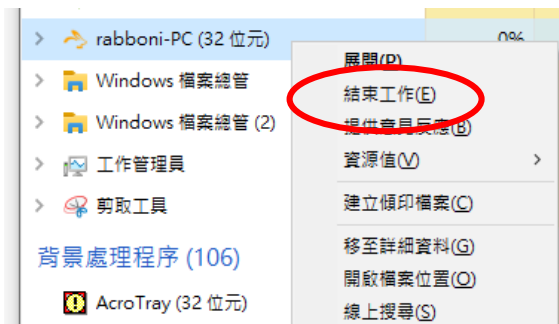


2. 找到仍在背景執行的rabboni程式

> rābboni-PC (32 位元)

0.1%	80.1 MB	0 MB/秒
------	---------	--------

3. 點擊右鍵選擇「結束工作」





## γabboni - PC UI介紹



1. USB：點擊透過USB連線
2. Bluetooth：點擊透過藍芽連線
3. MAC：輸入裝置MAC的地方
4. Scratch：點擊可以連到 Scratch
5. 驅動門檻：設定內建加速度公式 $\sqrt{x^2 + y^2 + z^2}$ 並計算驅動次數結果的門檻(要大於多少算一次)
6. 裝置驅動記錄數/Reset：紀錄驅動次數在
7. 驅動：搖動超過門檻會回傳 1
8. 新驅動紀錄數/Reset：每次重新連線回重新計數
9. X/Y/Z方向加速度 (1g=9.8m/sec<sup>2</sup>)
10. X/Y/X方向角速度 (degree/sec)
11. 參數設定：設定rabboni內的加速度以及角速度偵測範圍及 sampling rate。

## γabboni-USB連線

1. 打開Scratch UI
2. 連結USB



3. 點擊USB連結按鈕  
即可開始與電腦連線傳輸數據。

數字開始變動就是成功連線，變動數值就是三軸的加速度以及三軸的角速度。如果有問題的話就把檔案關起來重開。跳動值為量測值（含雜訊值），因此 Sensor 靜置仍會有跳動值。



Resource

## Python安裝(1/4)



- <https://www.python.org/downloads/>
- 配合AIOT 應用請下載3.7.5
- 額外的library 需要 PIP 安裝

**Download the latest version for Windows**

[Download Python 3.8.1](#)

Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [Mac OS X](#), [Other](#)

Want to help test development versions of Python? [Prereleases](#), [Docker images](#)

Looking for Python 2.7? See below for specific releases

Looking for a specific release?  
Python releases by version number:

Release version	Release date	Click for more
<a href="#">Python 3.5.9</a>	Nov. 2, 2019	<a href="#">Download</a> <a href="#">Release Notes</a>
<a href="#">Python 3.5.8</a>	Oct. 29, 2019	<a href="#">Download</a> <a href="#">Release Notes</a>
<a href="#">Python 2.7.17</a>	Oct. 19, 2019	<a href="#">Download</a> <a href="#">Release Notes</a>
<a href="#">Python 3.7.5</a>	Oct. 15, 2019	<a href="#">Download</a> <a href="#">Release Notes</a>
<a href="#">Python 3.8.0</a>	Oct. 14, 2019	<a href="#">Download</a> <a href="#">Release Notes</a>
<a href="#">Python 3.7.4</a>	July 8, 2019	<a href="#">Download</a> <a href="#">Release Notes</a>
<a href="#">Python 3.6.9</a>	July 2, 2019	<a href="#">Download</a> <a href="#">Release Notes</a>

[View older releases](#)

## Python安裝(3/4)

- 下載window安裝版本

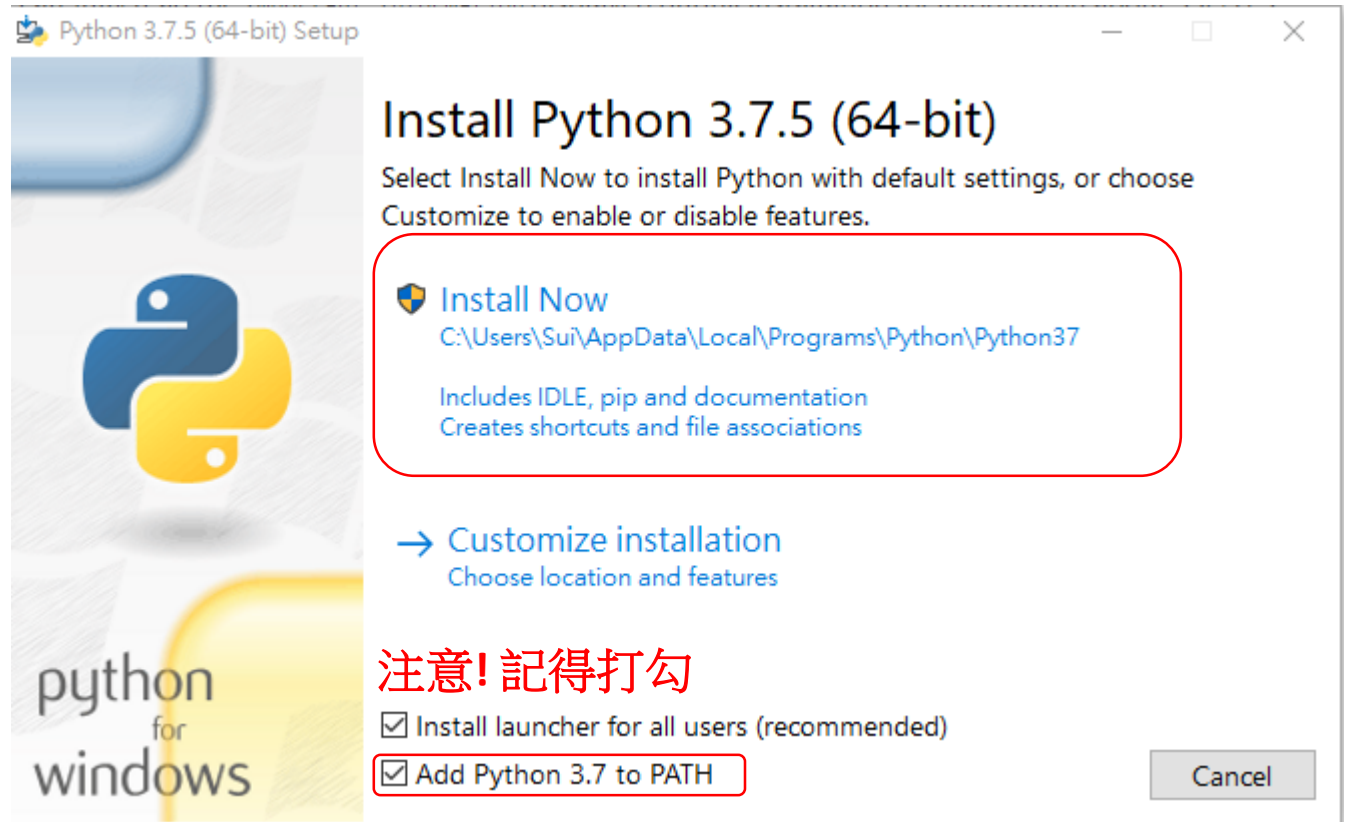
- 請先注意電腦中是否已安裝過python
- 若有其他版本請先卸載
- 建議只安裝一個版本
- 多版本的操作較複雜

### Files

Version	Operating System	Description	MD5 Sum	File Size	PGP
<a href="#">Gzipped source tarball</a>	Source release		1cd071f78ff6d9c7524c95303a3057aa	23126230	<a href="#">SIG</a>
<a href="#">XZ compressed source tarball</a>	Source release		08ed8030b1183107c48f2092e79a87e2	17236432	<a href="#">SIG</a>
<a href="#">macOS 64-bit/32-bit installer</a>	Mac OS X	{Deprecated} for Mac OS X 10.6 and later	cd503606638c8e6948a591a9229446e4	35020778	<a href="#">SIG</a>
<a href="#">macOS 64-bit installer</a>	Mac OS X	for macOS 10.9 and later	20d9540e88c6aaba1d2bc1ad5d069359	28198752	<a href="#">SIG</a>
<a href="#">Windows help file</a>	Windows		608cafa250f8baa11a69bbfcb842c0e0	8141193	<a href="#">SIG</a>
<a href="#">Windows x86-64 embeddable zip file</a>	Windows	for AMD64/EM64T/x64	436b0f803d2a0b393590030b1cd59853	7500597	<a href="#">SIG</a>
<a href="#">Windows x86-64 executable installer</a>	Windows	for AMD64/EM64T/x64	697f7a884e80ccaa9dff3a77e979b0f8	26777448	<a href="#">SIG</a>
<a href="#">Windows x86-64 web-based installer</a>	Windows	for AMD64/EM64T/x64	b8b6e5ce8c27c20bfd28f1366ddf8a2f	1363032	<a href="#">SIG</a>
<a href="#">Windows x86 embeddable zip file</a>	Windows		726877d1a1f5a7dc68f6a4fa48964cd1	6745126	<a href="#">SIG</a>
<a href="#">Windows x86 executable installer</a>	Windows		cfe9a828af6111d5951b74093d70ee89	25766192	<a href="#">SIG</a>
<a href="#">Windows x86 web-based installer</a>	Windows		ea946f4b76ce63d366d6ed0e32c11370	1324872	<a href="#">SIG</a>

## Python 安裝 (4/4)

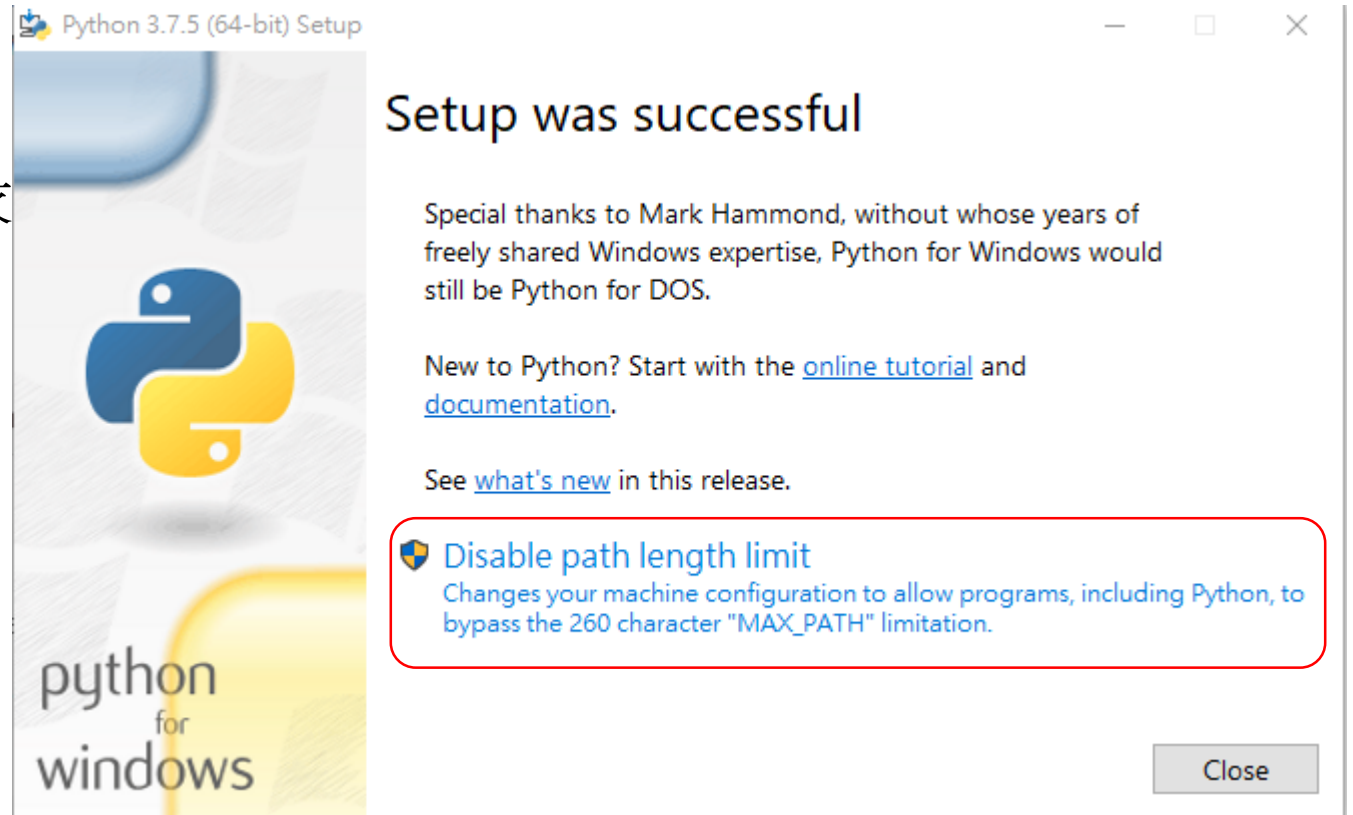
- 打勾Add Python 3.7...
- 點擊Install Now






## Python 安裝 (2/2)

- 跑完進度後
- 若有右圖的按鍵，請按下
  - 有的電腦安裝不會有此要求

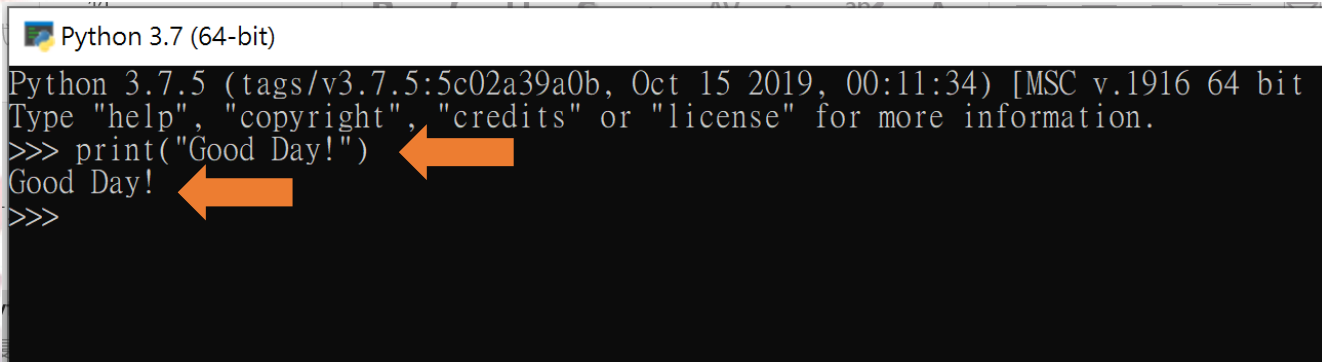




## 編譯器

- 下載安裝後從 [開始]  可看見新增:  
點擊Python 3.7 可於命令列模式下執行，

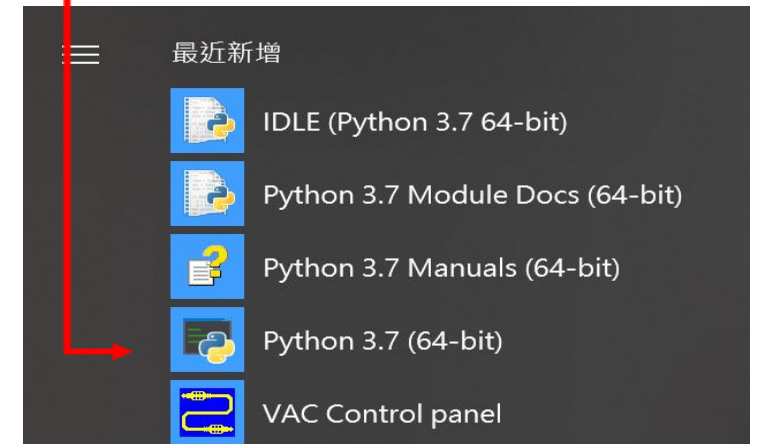
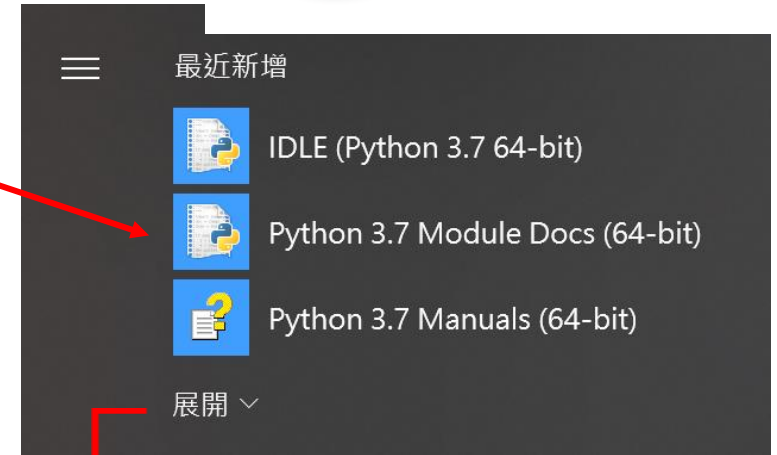
>>> print("Good Day!") Enter ( @print 要小寫)



```
Python 3.7 (64-bit)
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)]
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Good Day!")
Good Day!
>>>
```

Two orange arrows point to the input line and the output line in the terminal window.

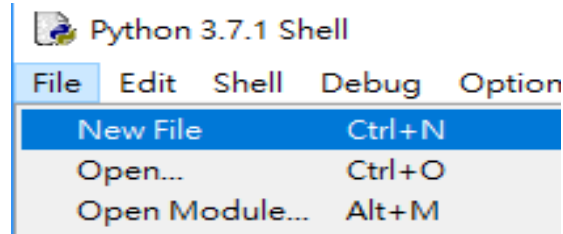
這就完成第一個程式啦!



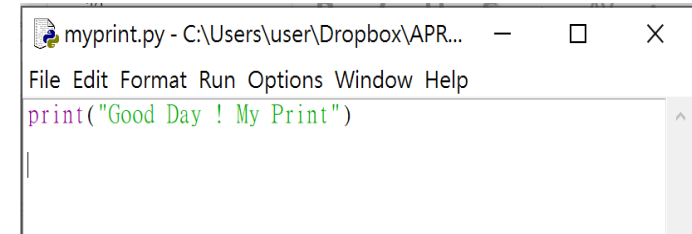
點擊 IDLE 編譯器可開始編輯程式  
坊間有許多編譯器可依喜好選擇



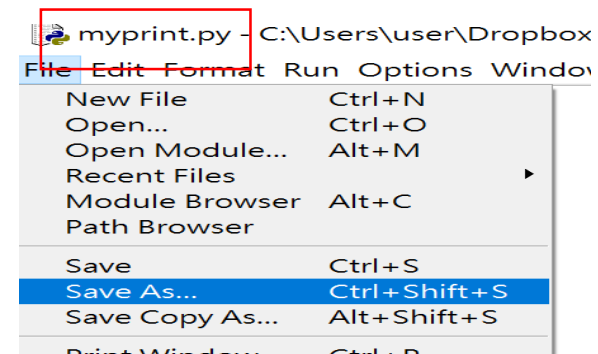
1. 開新檔案 File new



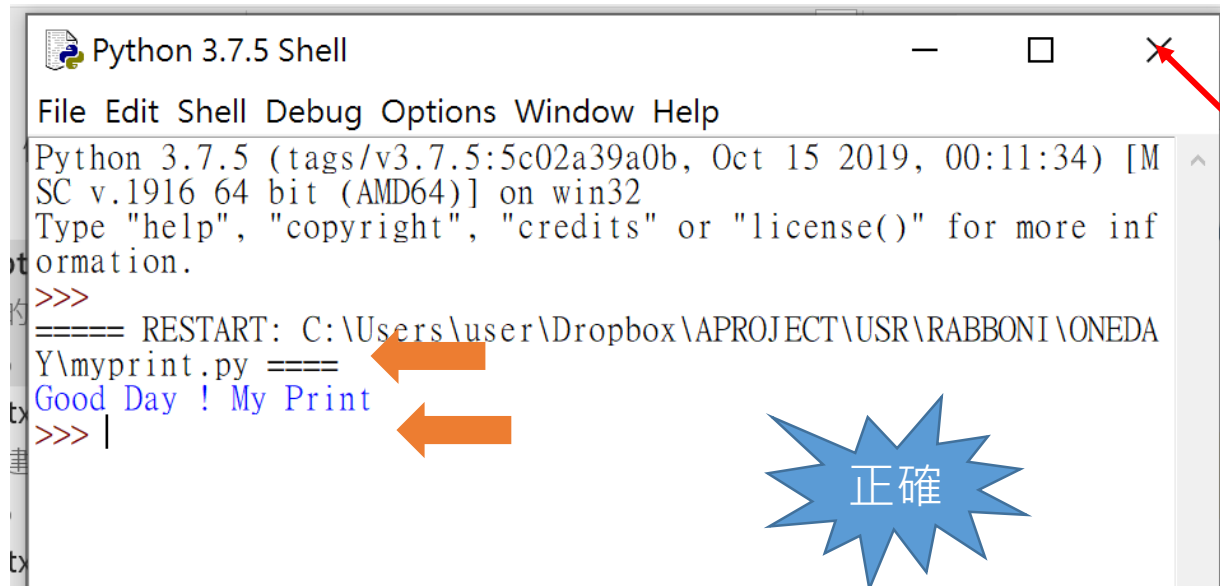
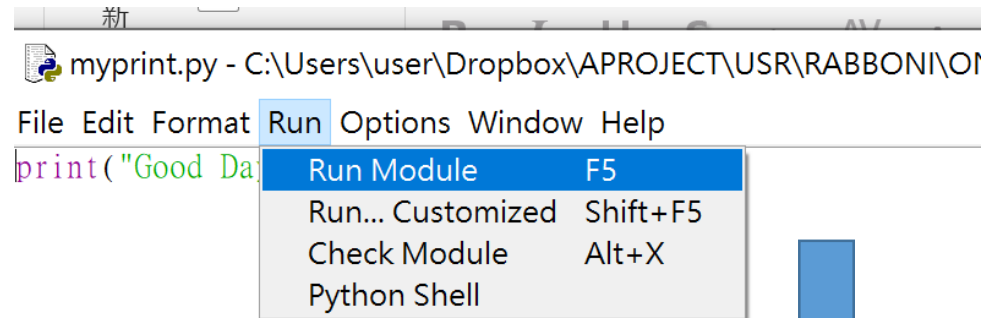
2. 打上指令: `print("Good Day! My Print.")`



3. save as  
設定檔名 ( myprint ) python 自動加上副檔名 .py



## 3. Run Module



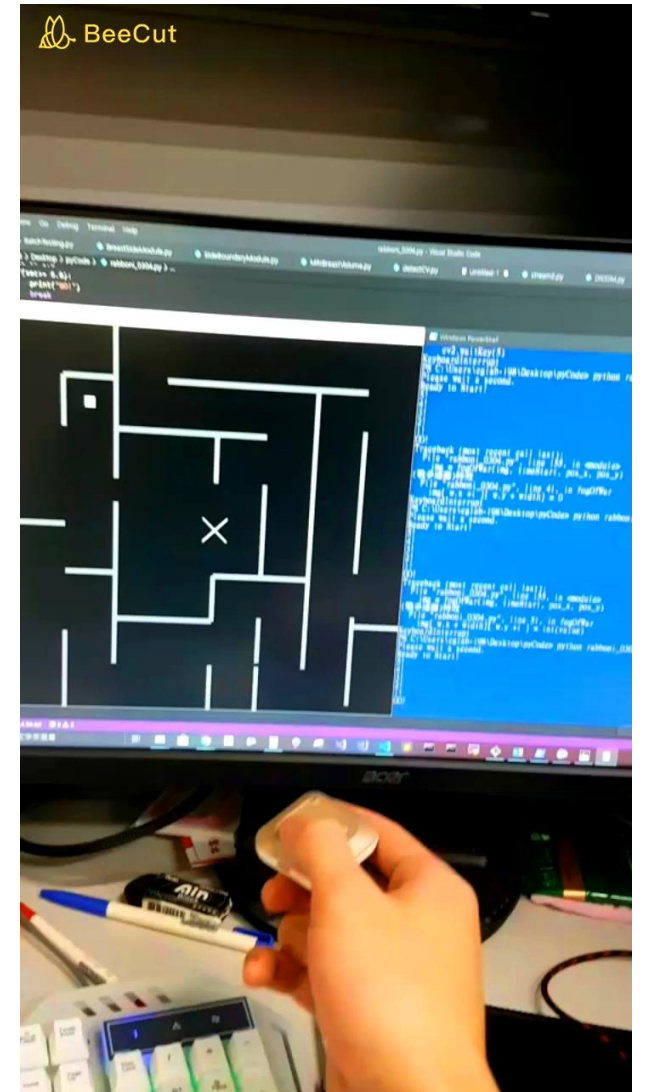
Close 回到編輯畫面



這就完成第一個程式

# Rabboni - Maze Demo

蔡孟哲 - 資科工所

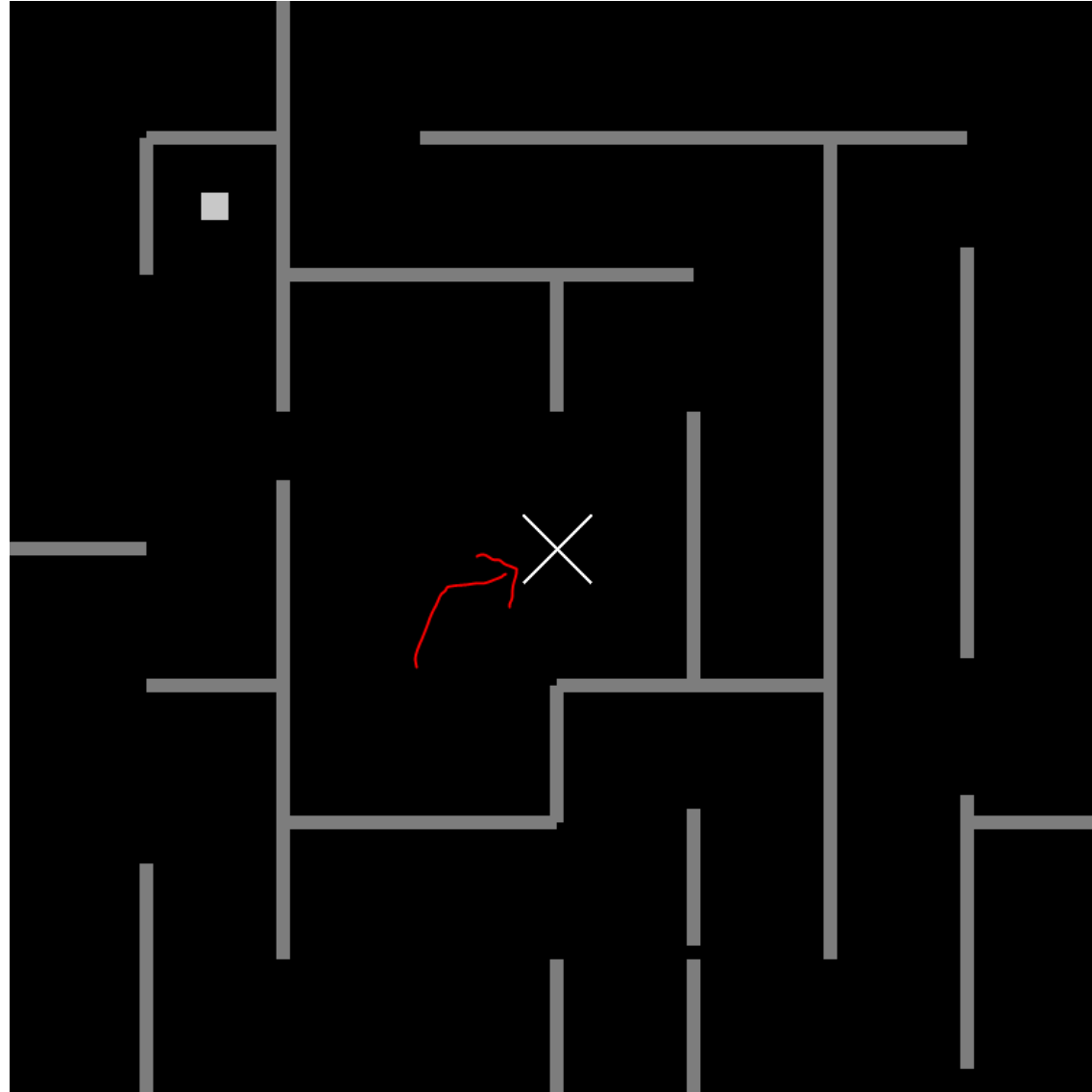


## 前置工作：

- 安裝python 3.7
- 安裝rabboni – python API
- 安裝 **numpy** , 在cmd上執行 **pip install numpy**
- 安裝 **OpenCV** 以及 **OpenCV – python API**

不好意思使用到額外套件有點麻煩 請見諒

# User, Wall and Destination





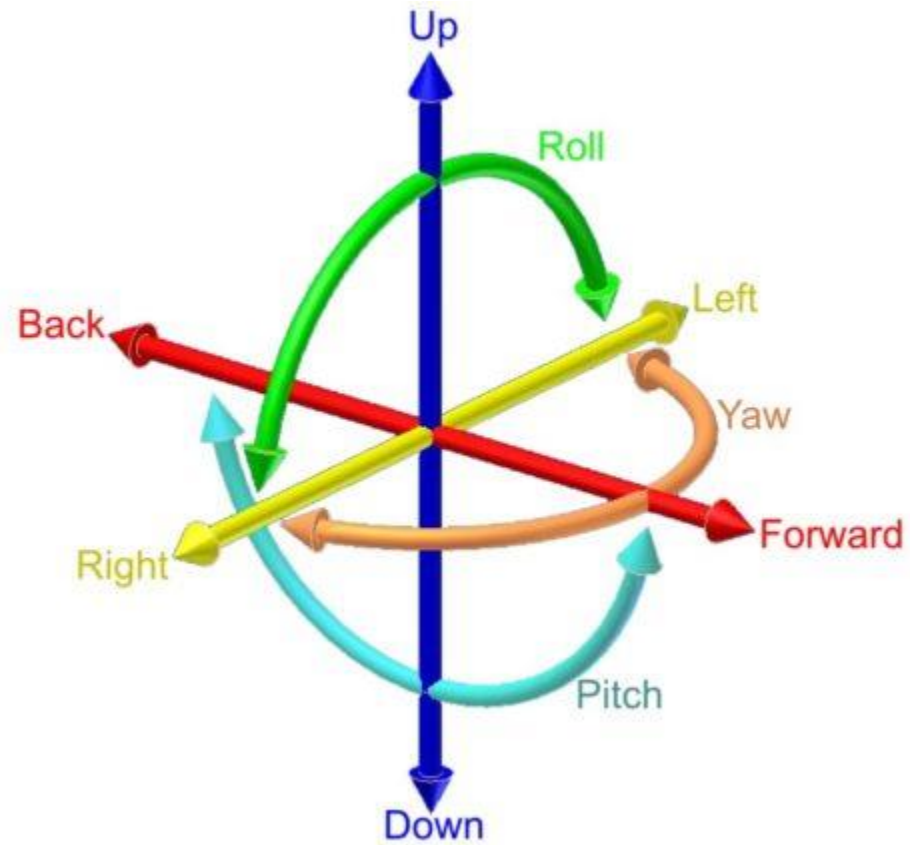
# User, Wall and Destination

```
for i in range(-25,25): .....#角色位置(pos_x, pos_y)上畫叉叉, 寬度為3
    img[ pos_x +i][ pos_y +i] = 255
    img[ pos_x +i +1][ pos_y +i] = 255
    img[ pos_x +i][ pos_y +i +1] = 255
for i in range(-25,25):
    img[ pos_x +i][ pos_y -i] = 255
    img[ pos_x +i][ pos_y -i -1] = 255
    img[ pos_x +i +1][ pos_y -i] = 255

cv2.imshow('Please Walk to The Cube.', img) .....#更新視窗內容
cv2.waitKey(2)
```

- 畫出角色(即叉叉)

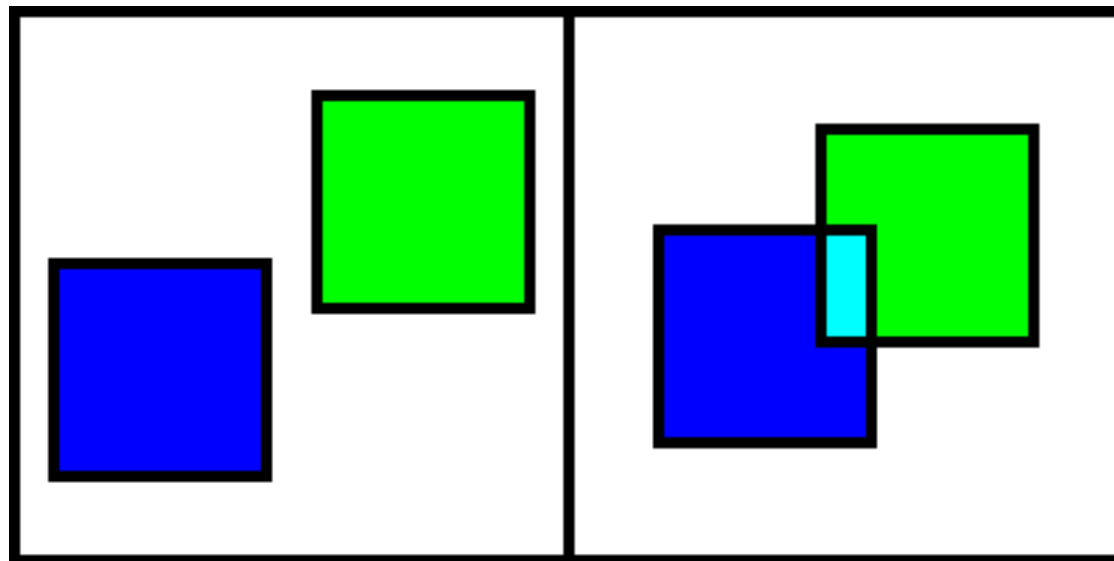
# Move - Accelerometer



# Move - Accelerometer

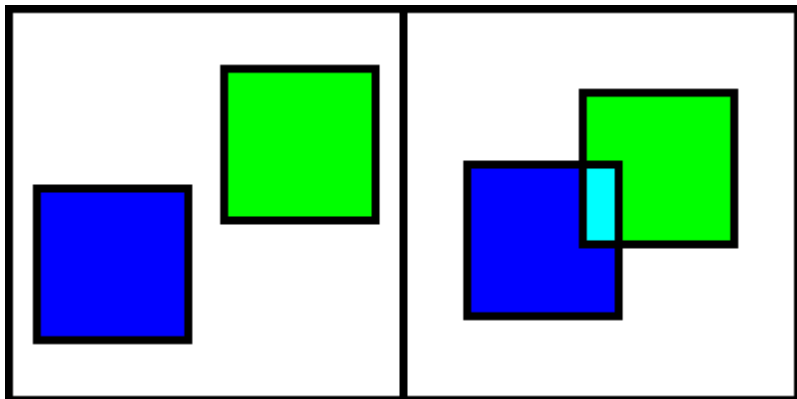
```
101
102 # 遊戲開始
103 timeStart = time.time()
104 while True:
105     if(abs(rabo.Accx) > abs(rabo.Accy)):
106         if(rabo.Accx > 0.5): .....#依靠Rabboni來移動
107             tem_y = pos_y - 2
108         if(rabo.Accx < -0.5):
109             tem_y = pos_y + 2
110     if(abs(rabo.Accx) < abs(rabo.Accy)):
111         if(rabo.Accy > 0.5):
112             tem_x = pos_x + 2
113         if(rabo.Accy < -0.5):
114             tem_x = pos_x - 2
115
```

# Collision Detection

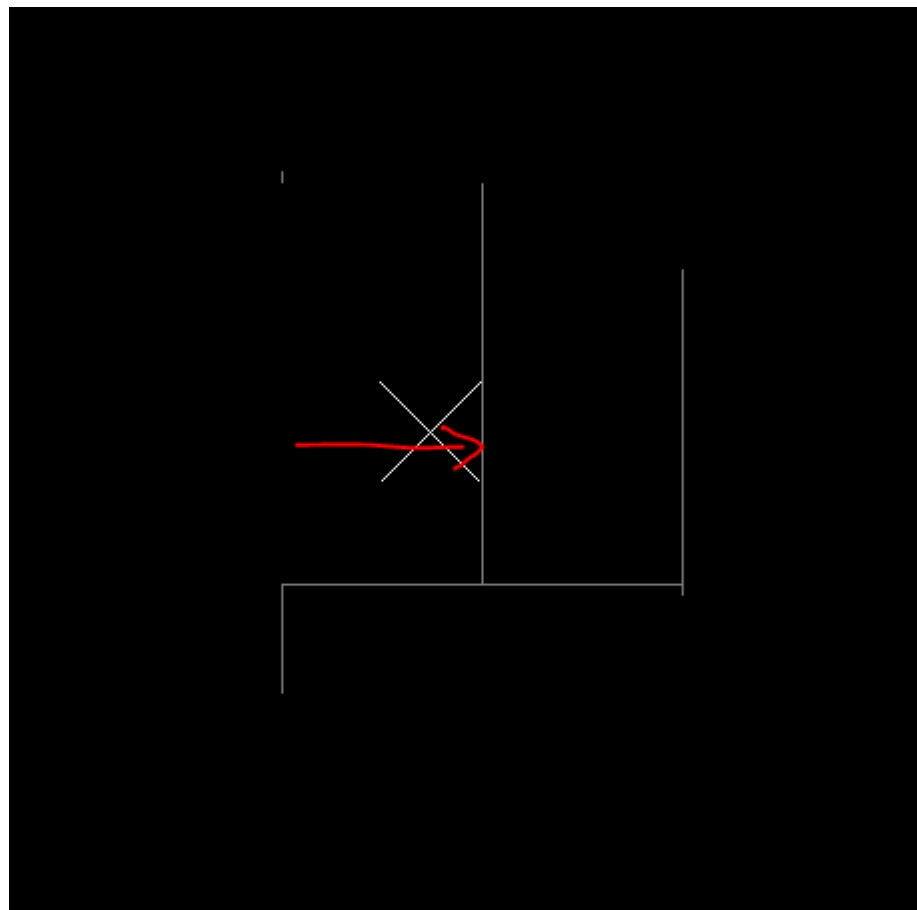


- 達到被牆壁卡住的效果

# Collision Detection



- 達到被牆壁卡住的效果



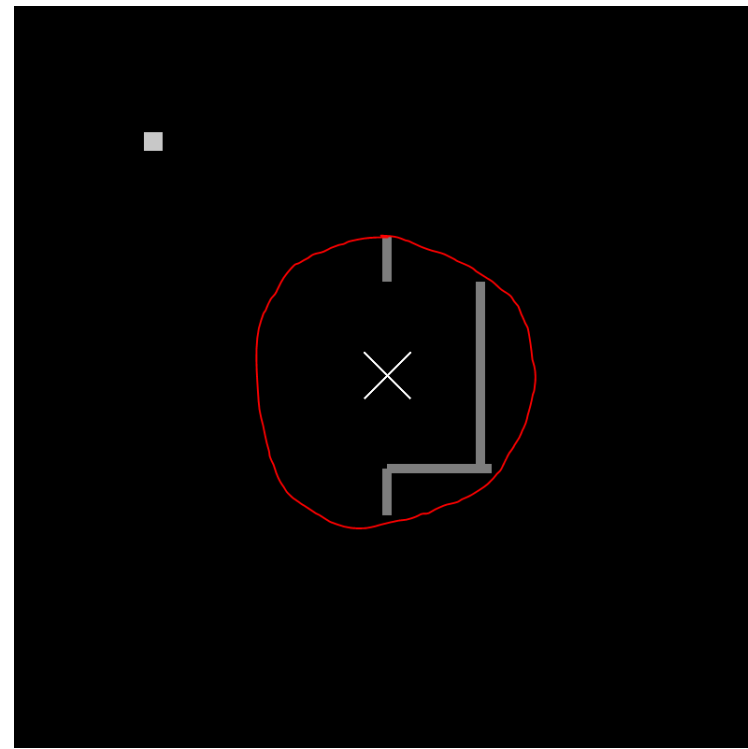
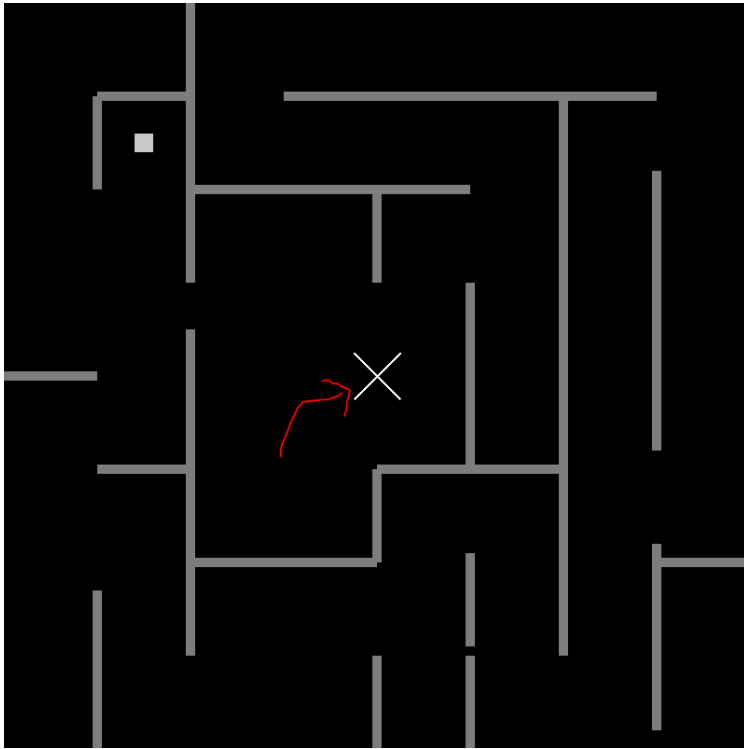
# Collision Detection

```
115
116     ...if( 25 <= tem_x < 775 and 25 <= tem_y < 775):.....#collision detect(碰撞偵測)
117     ...     flag = 0
118     ...     for i in range(-25,25):
119     ...         ...if(img[ tem_x +i][ tem_y +i] > 0):
120     ...             ...flag = 1
121     ...         for i in range(-25,25):
122     ...             ...if(img[ tem_x +i][ tem_y -i] > 0):
123     ...                 ...flag = 1
124     ...         ...if(flag == 0):
125     ...             ...pos_x = tem_x
126     ...             ...pos_y = tem_y
127
```

- 達到被牆壁卡住的效果



# Fog of War (戰爭迷霧效果)



- 離圓心太遠的牆壁消失

# Fog of War (戰爭迷霧效果)

圓公式：

$$(x - a)^2 + (y - b)^2 < r^2$$

X = wall's x position

Y = wall's y position

a = user's x position

b = user's y position

- 離圓心太遠的牆壁消失

# Fog of War (戰爭迷霧效果)

```
24 def fogOfWar(img, timeStart, pos_x, pos_y): ..... # Fog of War(戰爭迷霧)特效
25     value = 125.0
26     timeNow = time.time()
27     if(timeNow - timeStart > 10.0): #遊戲開始10秒後 啟動Fog of War(戰爭迷霧效果)
28         value = 225.0 - ((timeNow-timeStart)*10)
29
30     for index, w in enumerate(dataset1): # 垂直的牆
31         for i in range(100):
32             if(((w.x+i-pos_x)*(w.x+i-pos_x) + (w.y-pos_y)*(w.y-pos_y)) < 150*150): ..... # 圓公式
33                 for width in range(-5, 5):
34                     img[ w.x+i ][ w.y+width ] = 125
35             else: ..... # 離角色太遠的牆的顏色漸漸衰弱
36                 if(value > 0.0):
37                     for width in range(-5, 5): ..... # 牆的寬度為10
38                         img[ w.x+i ][ w.y+width ] = int(value)
39                 else:
40                     for width in range(-5, 5):
41                         img[ w.x+i ][ w.y+width ] = 0
42
43     for index, w in enumerate(dataset2): # 水平的牆
44         for i in range(100):
45             if(((w.x-pos_x)*(w.x-pos_x) + (w.y+i-pos_y)*(w.y+i-pos_y)) < 150*150):
46                 for width in range(-5, 5):
47                     img[ w.x+width ][ w.y+i ] = 125
48             else:
49                 if(value > 0.0):
50                     for width in range(-5, 5):
51                         img[ w.x+width ][ w.y+i ] = int(value)
52                 else:
53                     for width in range(-5, 5):
54                         img[ w.x+width ][ w.y+i ] = 0
55     return img
56
```

# Destination



- 抵達目的, 獲勝!

# Destination

```
11
12 def winGame(img, pos_x, pos_y): ..... # 贏得遊戲
13     .... for k in range(255, 0, -1): ..... # 遊戲結束後 角色及方塊慢慢消失
14         .... for i in range(-10, 10):
15             .... for j in range(-10, 10):
16                 .... img[150+i][150+j] = k
17             .... for i in range(-25, 25):
18                 .... img[pos_x+i][pos_y+i] = k
19             .... for i in range(-25, 25):
20                 .... img[pos_x+i][pos_y-i] = k
21         .... cv2.imshow('Please Walk to The Cube.', img)
22         .... cv2.waitKey(10)
23
```

Thank you for your attention

## 常見問題

Rabboni 連結指令: 請點擊參考



### Rabboni with python

Python 安裝與前置作業

- 請先至 Python 官方網站下載最新版 python(windows)  
<https://www.python.org/downloads/>
- 使用 python 安裝 rabboni 專屬的 library
  - 使用系統管理員開啟命令提示字元(CMD)
  - 輸入: `pip install rabboni==1.73` (目前最新版本 1.73)



### Rabboni 介紹

- 內含 6 軸 sensor, 3-axis 加速度與 3-axis 陀螺儀
  - 規格: ICM-20689
  - 可使用 Python 藍芽與 USB API 將值讀出來
- 內置 counter 計數功能
  - 可使用 Python 藍芽與 USB API 將值讀出來
  - 可使用 Python 藍芽與 USB API 將 counter 歸零
- 內建兩種連接模式
  - 藍芽模式:
    - 將 dongle 插上電腦 USB
    - 開啟 rabboni, 按下藍芽廣播鍵
    - 執行 python 藍芽相關 code(如 example code)
  - USB 模式:
    - 將 rabboni 使用 USB 連上電腦
    - 執行 python USB 相關 code(如 example code)

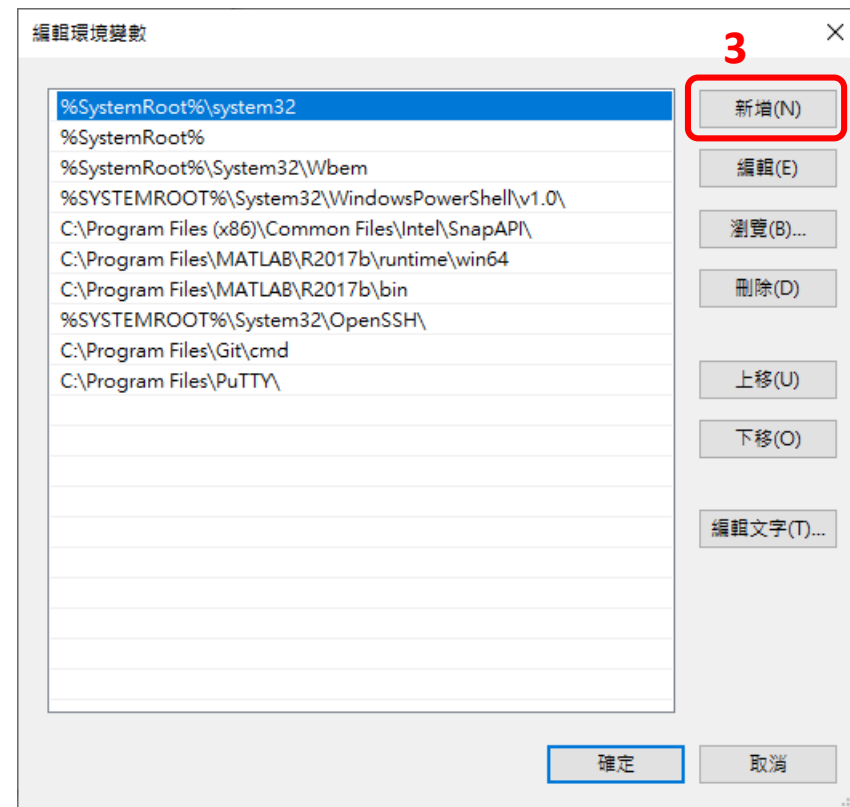
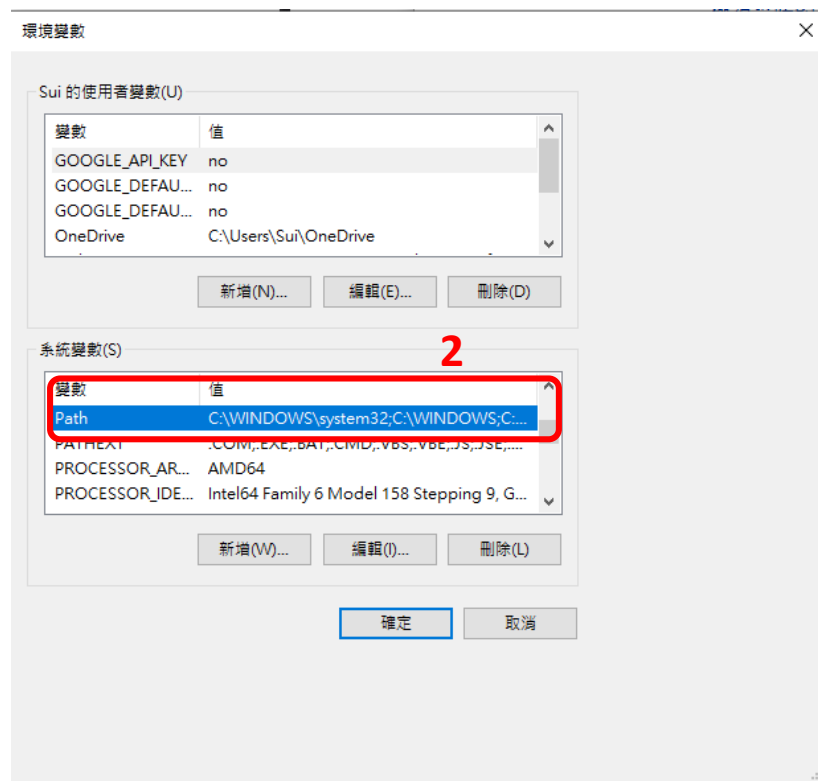


## 尋找不到python、pip指令

- Q:
  - 在CMD視窗輸入 python, pip無反應
  - 輸入 python, pip顯示 “python” 不是外部或內部命令….
- A:
  - 0.沒安裝好Python，卸載再安裝一次或是執行1~5步驟
  - 1.先找到Python安裝的地方，並複製位置
    - 通常會在 C:\Users\Username\AppData\Local\Programs\Python\Python37
    - Username是自己電腦帳戶名稱
  - 2.開啟電腦「控制台」->「系統」->「進階系統設定」->「環境變數」
  - 3.系統變數的視窗中尋找 Path 變數打開
  - 4.新增 Python的位置跟Python script的位置
  - 5.按確定退出

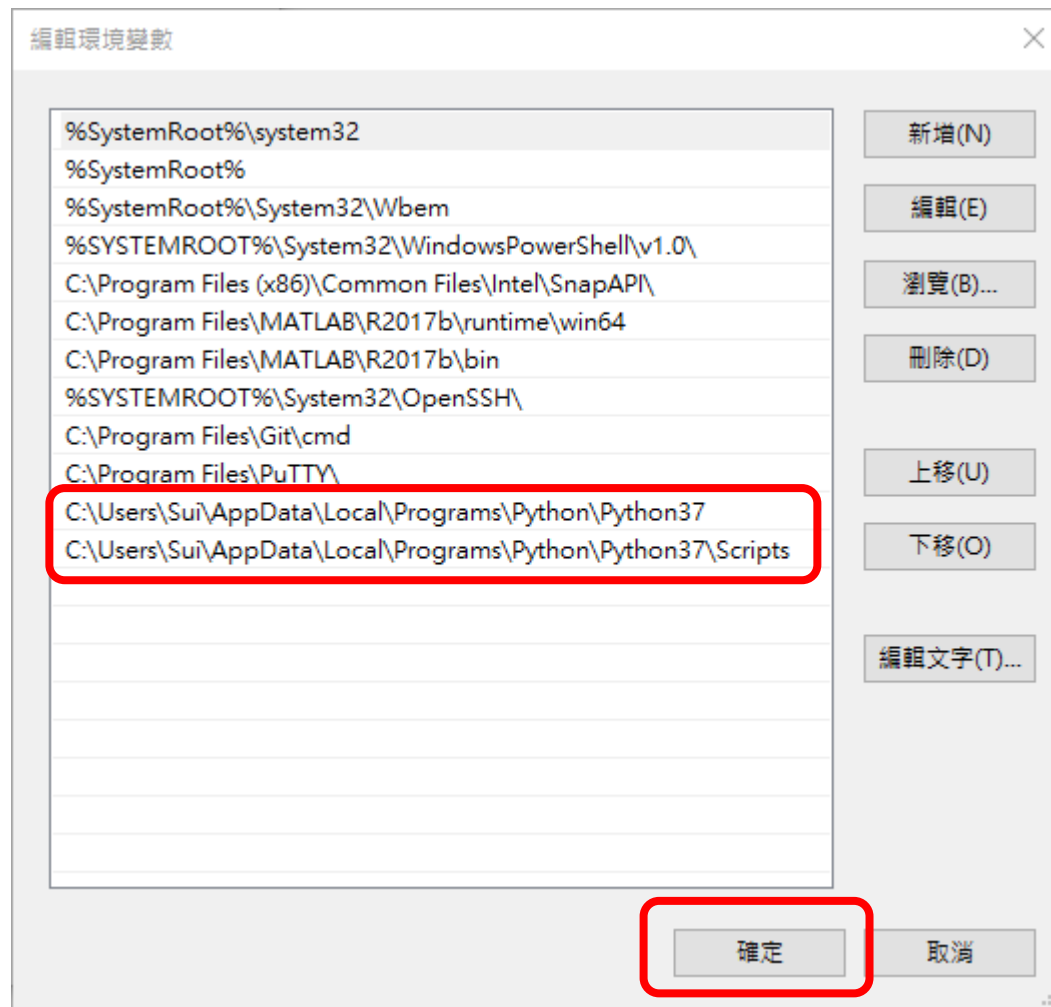


## 尋找不到python、pip指令



## 尋找不到python、pip指令

- 新增 Python的位置
- 新增 Python script的位置
- 按確定退出



## 安裝好Rabboni但Python卻說找不到

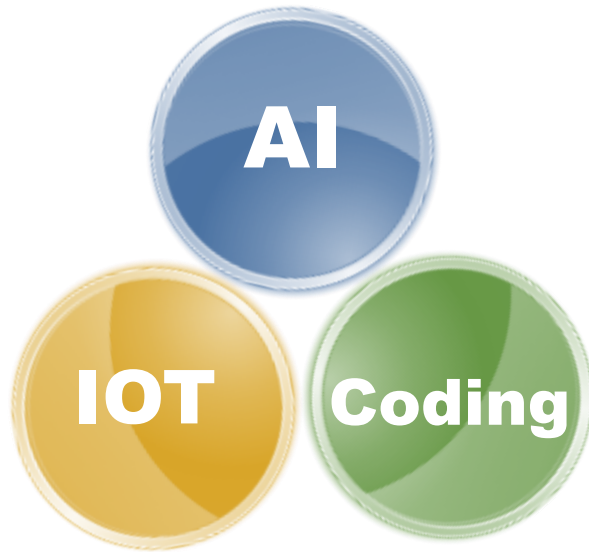
- Q:

- from rabboni import \*
- 顯示 No module named 'rabboni'

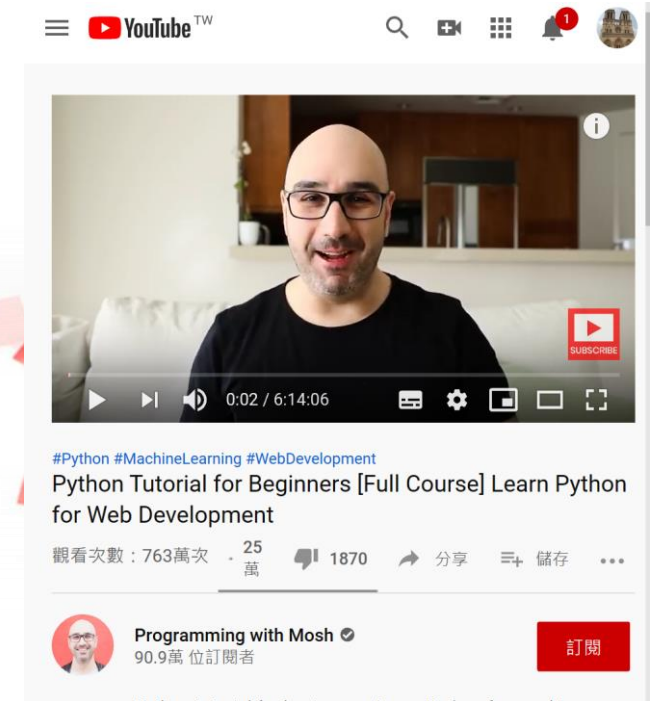
```
>>> from rabboni import *
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'rabboni'
>>>
```

- A:

- 原因有幾種可能
- 1.先確定是否有安裝好rabboni，重新進行第九頁內容
- 2.若確定有安裝rabboni，可能是該電腦存在多個python
  - rabboni安裝到 python1 但執行時是用python2
  - 解決方法:
    - 1.卸載用不到的Python再重新安裝rabboni
    - 2.檢查上頁的環境變數，python跟Python/Script是否都是都在同個資料夾，若非，將其改成同一個資料夾，再重新安裝rabboni



WITH **FUN!**



[https://youtu.be/\\_uQrJ0TkZlc](https://youtu.be/_uQrJ0TkZlc)