



# Assignment 0

This assignment is based on the Assignment 2, 3, and 4 of SC001 from stanCode



作業檔案下載

由於進階班需要非常扎實的基礎才能幫助同學們往上蓋出程式巨塔，這份作業將包含撰寫程式最重要的基本邏輯：靈活使用變數 (Variables)、使用者互動之對畫框 (Console)、各式迴圈 (Loops)、並熟悉運算法則 (Expression)

本次作業請勿使用任何 Python 資料結構 (如 list, dict, tuple, ...)

所有題目都改編自 SC001 的作業/考試，請下載上方檔案進行編寫

本分作業目的為提供同學一個是否能順利進行進階班的最低標準  
若您可以獨立完成每一題，歡迎您加入 stanCode 的大家庭：)

繳交期限：收到後一週內，回傳至 [stancode.tw@gmail.com](mailto:stancode.tw@gmail.com)

## Problem 1 - class\_reviews.py

請同學們幫忙 stanCode 寫出一個 SC001, SC101 期末評分計算程式！  
( 您可以假設在課程名稱的部分使用者只會輸入 'SC001' 及 'SC101' 這兩個字串，  
且大小寫不會影響您最後輸出之結果計算 )

首先，您的程式將先詢問課號 ( SC001 或是 SC101 )，接著詢問該課號所獲得之分數 ( 一定是整數 )。您的程式將分別計算每個課號所獲得之「最高分」、「最低分」以及「平均分數」。使用者將輸入 -1 來當作輸入結束的指示！若您程式撰寫正確，將可以重現下方截圖每一行之內容：

```
Which class? Sc101
Score: 10
Which class? sc001
Score: 7
Which class? sC001
Score: 9
Which class? SC001
Score: 9
Which class? -1
=====SC001=====
Max (001): 9
Min (001): 7
Avg (001): 8.333333333333334
=====SC101=====
Max (101): 10
Min (101): 10
Avg (101): 10.0
```

眼尖的同学一定有注意到，SC101 只有在剛剛的例子中獲得一筆評分 ( 10 分 )！這個時候，「最高分」、「最低分」以及「平均分數」都將是 10 分，唯平均分數的資料型態是 float 請同學們小心

然而，若使用者輸入的課號只有 SC001，您的程式執行結束的當下，應該會隨之產生如下方截圖所示之每一行文字/數字：

```
Which class? sc001
Score: 7
Which class? sc001
Score: 10
Which class? sc001
Score: 9
Which class? -1
=====SC001=====
Max (001): 10
Min (001): 7
Avg (001): 8.6666666666666666
=====SC101=====
No score for SC101
```

同理可證，若使用者輸入的課號只有 SC101，您的程式執行結束應該會產生如下方截圖所示之每一行文字/數字：

```
Which class? Sc101
Score: 10
Which class? sC101
Score: 9
Which class? SC101
Score: 9
Which class? sc101
Score: 10
Which class? -1
=====SC001=====
No score for SC001
=====SC101=====
Max (101): 10
Min (101): 9
Avg (101): 9.5
```

最後，如果使用者一開始便在輸入課號時輸入 -1，那我們就要印出 **No class scores were entered** 的字樣，如下圖所示：

```
Which class? -1
No class scores were entered
```

## Problem 2 - coin\_flip\_runs.py

再來要請同學模擬投擲一枚硬幣直到該枚硬幣達到使用者輸入的「連續次數 ( 以下簡稱 num\_run ) 」之結果！

硬幣的正面我們以 'H' 表示；硬幣的背面我們以 'T' 表示。您的程式將使用 random 包裹去隨機產生 'H' 或是 'T'，並想辦法在 num\_run 次數達到的時候終止程式

舉例來說，如下圖，程式一開始在畫面印出 **Let's flip a coin!** 的字樣；緊接著，使用者在看到 **Number of runs:** 的字樣後輸入一個整數 ( 以下簡稱 num\_run )

```
Let's flip a coin!
Number of runs: 1
TT
```

若 num\_run 的數值為 1，您的程式將在「連續出現 1 次的 'H' 或 'T' 當下結束執行」並將投擲過程印出！請注意：整個過程是隨機的！如下如第二次執行產生的畫面 ( 有很大的機率與第一次執行結果不同 )

```
Let's flip a coin!
Number of runs: 1
HTHH
```

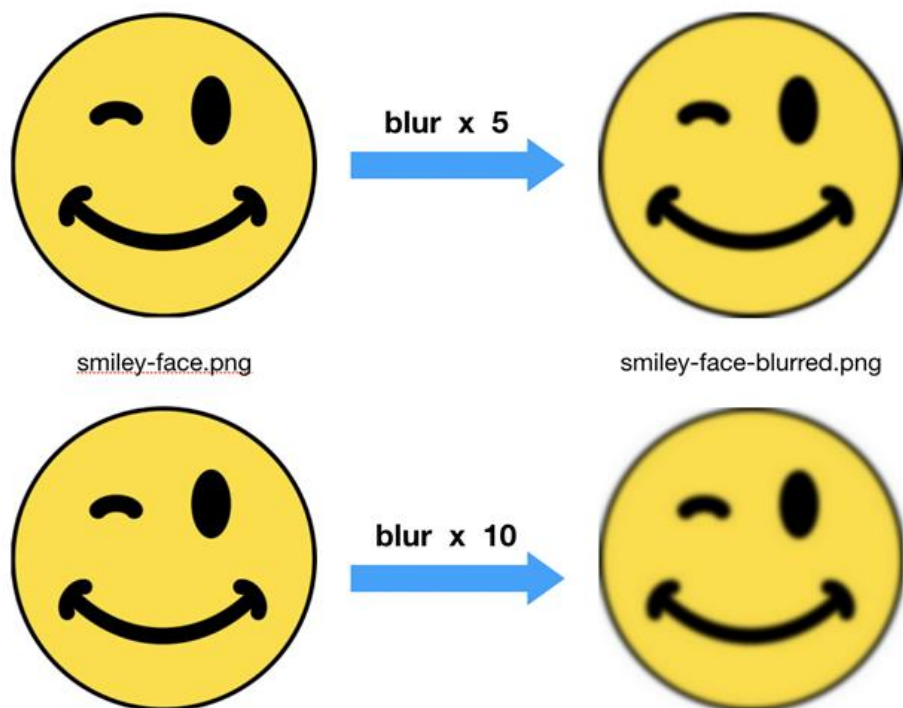
同理可證，若 num\_run 的數值為 5，您的程式將在「連續出現 5 次的 'H' 或 'T' 當下結束執行」並將投擲過程印出！請注意：下圖之紅線是我們後製上去的

```
Let's flip a coin!
Number of runs: 5
THHHHHTTTHTHTHHTTTHTHTHH
  └──┬──┘ └──┬──┘ └──┬──┘ └──┬──┘ └──┬──┘
    1 run  2 run  3 run  4 run  5 run
```

### Problem 3 - blur.py

第三題要請同學編輯 `def blur(img)` 並 `return` 一張將原圖 `img` 模糊處理的影像。  
我們使用的方法是將原本 pixel 數值改成此 pixel 與其身邊相鄰 pixels 之平均值

您這題所撰寫的程式碼請控制在 30 行以內 ( 提示：使用 4 個 for loop )  
找鄰居的演算法將在 SC101 最後一份派上用場！



假設我們有一個座標為  $(x, y)$  的 pixel，它模糊後的 `new_r`, `new_g`, `new_b` 數值應該為  $(x, y)$  與其周圍八個點  $(x-1, y)$ ,  $(x+1, y)$ ,  $(x-1, y-1)$ ,  $(x, y-1)$ ,  $(x+1, y-1)$ ,  $(x-1, y+1)$ ,  $(x, y+1)$ ,  $(x+1, y+1)$  的平均。舉例來說，下圖  $(2, 1)$  點模糊後的新數值應該為  $(52, 41, 55)$

	0	1	2	3	4
0	(14, 97, 63)	(84, 22, 99)	(74, 38, 69)	(16, 17, 18)	(85, 75, 75)
1	(21, 18, 45)	(66, 53, 88)	<b>(32, 67, 12)</b>	(95, 65, 35)	(6, 0, 2)
2	(37, 29, 61)	(28, 49, 31)	(47, 21, 94)	(31, 41, 51)	(246, 84, 13)
3	(82, 33, 90)	(42, 43, 44)	(15, 80, 50)	(60, 40, 12)	(188, 45, 1)

$$52 = (84+74+16+66+32+95+28+47+31) / 9$$

$$41 = (22+38+17+53+67+65+49+21+41) / 9$$

$$55 = (99+69+18+88+12+35+31+94+51) / 9$$

以下七點請注意：

- 此題的程式碼請勿超過 30 行 ( 請使用 4 個 for loop 來找鄰居 ) 這個概念將在進階班最後一份作業派上用場！
- 請務必將平均出來的值存在一個全新的 `new_img`，千萬不要用新得到平均數值來改變舊影像 ( 您將使用 `new_img = SimpleImage.blank(new_w, new_h)` 來製造空白的影像 `new_img` )
- 位在角落的點，例如上圖之 `(0, 0)`，只會有三個鄰居 `(0, 1)`, `(1, 0)`, `(1, 1)`
- 位在邊上的點，例如上圖之 `(2, 0)`，只會有五個鄰居 `(1, 0)`, `(1, 1)`, `(2, 1)`, `(3, 0)`, `(3, 1)`
- 請注意 `def blur(img)` 接收的 `img` 已經是一張照片，並不是檔名
- 在 `def main()` 裡我們使用 for loop 呼叫您要編輯的 `blur` 來達到多次的模糊效果 ( 如下圖之程式碼所示 )
- 此題的運算量極大，大約需要一分鐘的運算時間，請同學耐心等待

```
def main():
    old_img = SimpleImage("images/smiley-face.png")
    old_img.show()

    blurred_img = blur(old_img)
    for i in range(4):
        blurred_img = blur(blurred_img)
    blurred_img.show()
```

# 評分標準

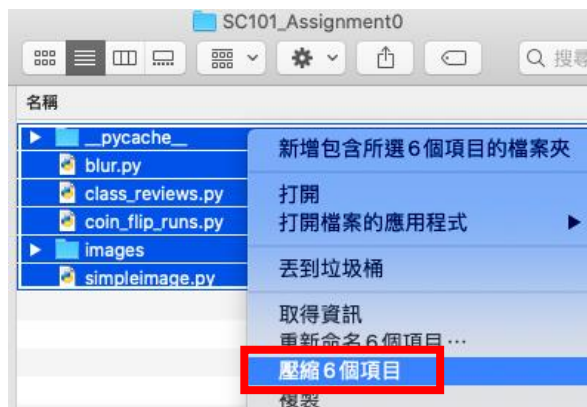
**Functionality** - 程式是否有通過我們的基本要求？程式必須沒有 bug、能順利完成指定的任務、並確保程式沒有卡在任何的無限環圈 ( Infinite loop ) 之中

**Style** - 好的程式要有好的使用說明，也要讓人一目瞭然，這樣全世界的人才能使用各位的 code 去建造更多更巨大更有趣的程式。因此請大家寫**精簡扼要**的使用說明、function 敘述、單行註解

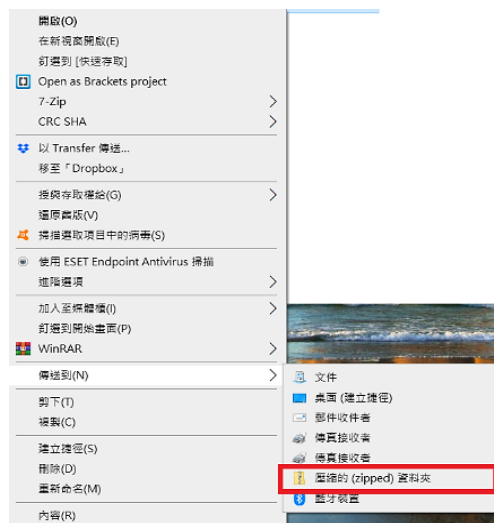
# 作業繳交

1. 以滑鼠「全選」作業資料夾內的所有檔案，並壓縮檔案。請見下圖說明。

macOS：按右鍵選擇「壓縮 n 個項目」

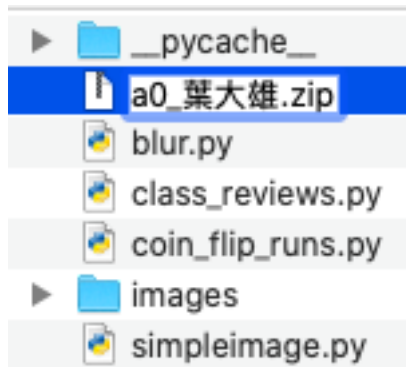


Windows：按右鍵選擇「傳送到」→「壓縮的(zipped)資料夾」



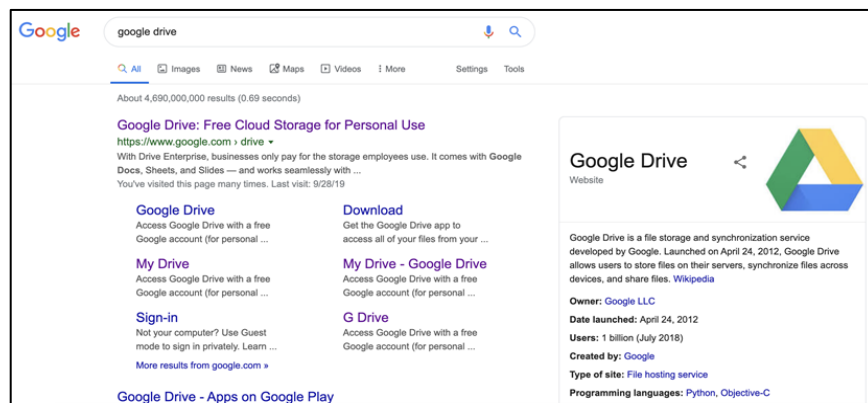


2. 將壓縮檔(.zip)重新命名為「a0\_中文姓名」，如：



3. 將命名好的壓縮檔(.zip)上傳至 Google Drive ( 或任何雲端空間 )

1) 搜尋「google drive」



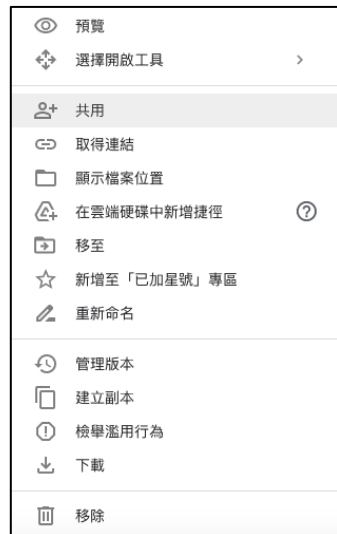
2) 登入後，點選左上角「新增」→「檔案上傳」→ 選擇作業壓縮檔(.zip)



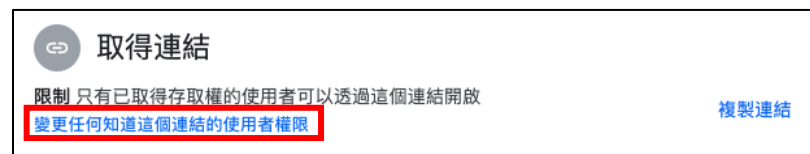


#### 4. 開啟連結共用設定，並複製下載連結

1) 對檔案按右鍵，點選「共用」



2) 點擊「變更任何知道這個連結的使用者權限」後，權限會變為「可檢視」



3) 點選「複製連結」



#### 5. 將連結回傳至 stancode.tw@gmail.com

( 之後會是將連結上傳至開課後的作業提交表單 )



Should you have any idea or questions, please feel free to contact.