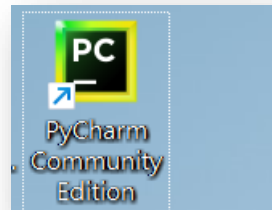


白河國中資訊科技課程:PYTHON

壹、我們課程執行 PYTHON 程式的平台環境

有兩種:

一、單機版 Pycharm



二、線上版 Google Colab

(一)登入 google 教育帳號:

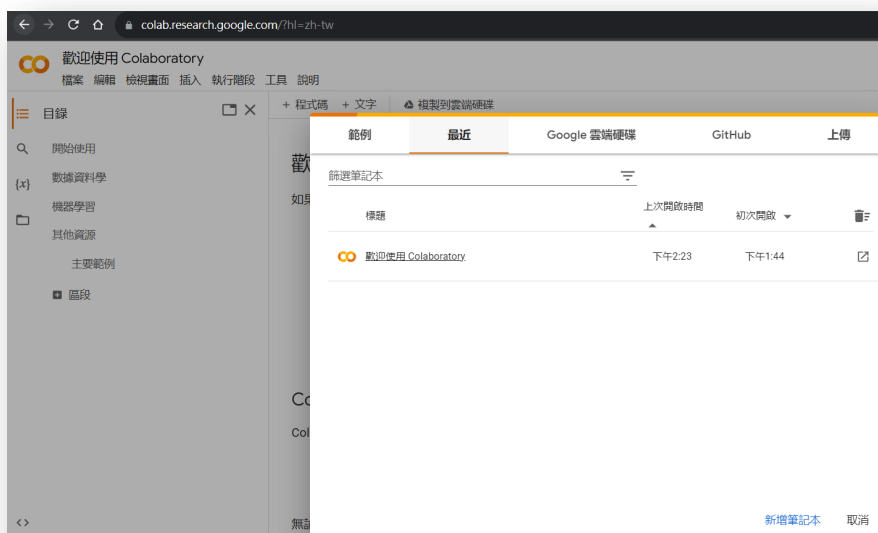
同學點選網站: https://www.google.com.tw/?hl=zh_TW

右上角登入 GOOGLE 教育帳號:

帳號規則: 大部分同學帳號為 stxxxxxxx@go.edu.tw , 例如你的 openID 為 st1234567 ,

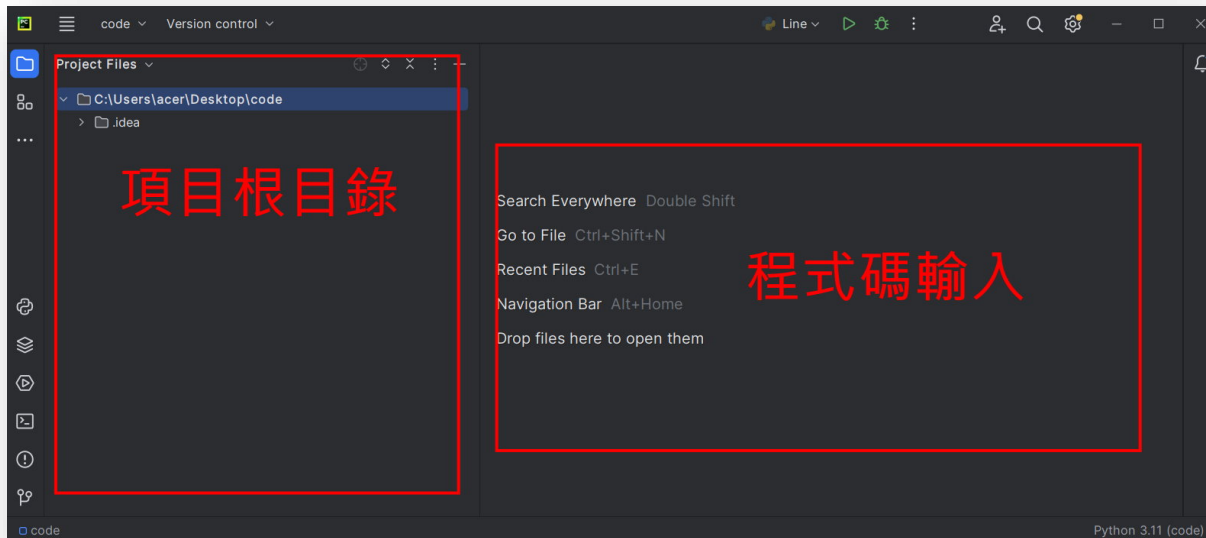
那你的帳號就是 st1234567@go.edu.tw

(二)點選 colab 網站: <https://colab.research.google.com/?hl=zh-tw>



貳、單機版 Pycharm 建立 python 文件

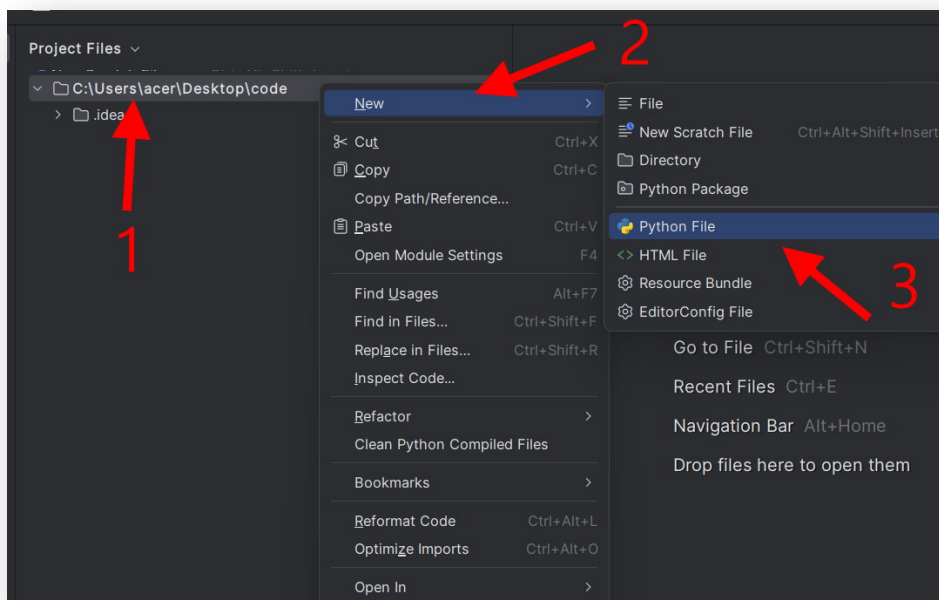
一、打開桌面 Pycharm 程式



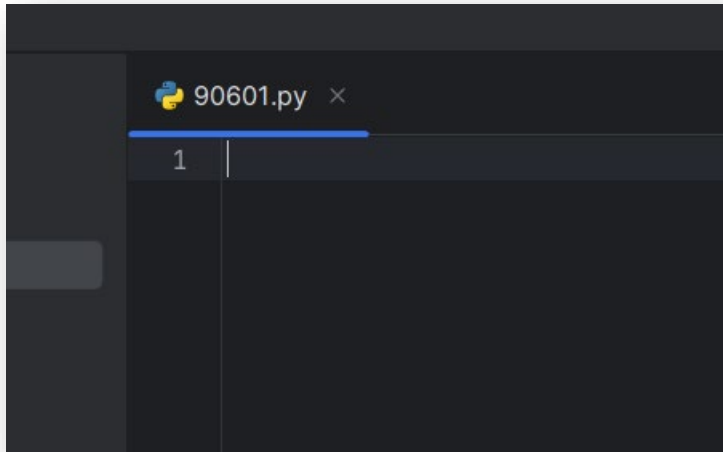
二、新建或打開文件並書寫程式碼

新建：在左側根目錄內部任一位置 - 右鍵 - [new] - [Python File] - 輸入檔名(班級+座號) - OK

註:將來檔案會上傳伺服器執行，檔名不能使用中文



這裡老師輸入 90601 檔名，各位發現除了檔名之外，副檔名出現了 [.py]，這是 python 檔案的格式。



今日課程: Python 運算符號分類 1(5)

運算符號分五類:

- ✓ 算術運算子
- ✓ 賦值運算子
- ✓ 複合賦值運算子
- ✓ 比較運算子
- ✓ 邏輯運算子

【邏輯運算子】

「邏輯運算子」可用以判斷「and」(且)、「or」(或) 和「not」(非) 三種邏輯狀態，運算的結果只有 True 或是 False 兩種。

比較運算子可以使用 & 代替 and · | 代替 or。

一、「and」：且

當 a 是 True，b 也是 True，結果是 True，但只要 a、b 其中一個是 False，結果就是 False。口訣:都真才真!

a and b	a = True	a = False
b = True	True	False
b = False	False	False

舉例:

我們令 $a=0$, $b=1$, $c=2$, 我們使用 `print` 指令讓電腦來判斷式子正確與否 , 並使用 `and` 指令進行邏輯運算:

```
1 a = 0
2 b = 1
3 c = 2
4
5 # 1. and :都真才真
6 print( (a>b) and (c>b) )
7
8 print( (a<b) and (b<c) )
```

我們觀察第 6 行指令:知道 $a>b$ 為假(False) , $c>b$ 為真(True) , 根據 `and` 口訣:都真才真 , 按右鍵執行 , 結果會呈現 False 。

False

以此類推 , 觀察第 8 行指令: 知道 $a<b$ 為真(True) , $b<c$ 為真(True) , 根據 `and` 口訣:都真才真 , 按右鍵執行 , 結果會呈現 True 。

True

二、「or」：或

只要 a、b 其中一個是 True，結果就是 True。口訣：一真則真，都假才假!

a or b	a = True	a = False
b = True	True	True
b = False	True	False

舉例：

我們令 $a=0$ ， $b=1$ ， $c=2$ ，我們使用 print 指令讓電腦來判斷式子正確與否，並使用 or 指令進行邏輯運算：

```
10 #2. or :一真則真，都假才假
11 print( (a>b) or (c>b) )
12
13 print( (a>b) or (b>c) )
```

我們觀察第 11 行指令：知道 $a>b$ 為假(False)， $c>b$ 為真(True)，根據 or 口訣：一真則真，按右鍵執行，結果會呈現 True。

True

以此類推，觀察第 13 行指令：知道 $a > b$ 為假(False)， $b > c$ 為假(False)，根據 or 口訣：一真則真，都假才假，按右鍵執行，結果會呈現 False。

False

三、「not」：取反義

如果 a 為 True，not a 的結果 False，如果 a 為 False，not a 的結果 True。

舉例：

我們令 $a=0$ ， $b=1$ ， $c=2$ ，我們使用 print 指令讓電腦來判斷式子正確與否，並使用 not 指令進行邏輯運算：

```
15 #3. not :取反義
16 print(not False)
17
18 print(not c>b)
```

觀察第 16 行指令:知道 (False)為假，前面加上 not 取之反義，按右鍵執行，結果會呈現 True。

True

以此類推，觀察第 18 行指令: 知道 $c > b$ 為真(True)，前面加上 not 取之反義，按右鍵執行，結果會呈現 False。

False

注意：邏輯運算子前後也可以加上括號，當有括號的時候，則必須優先計算!

【今日作業】

到 classroom 第 17 堂作業填答:

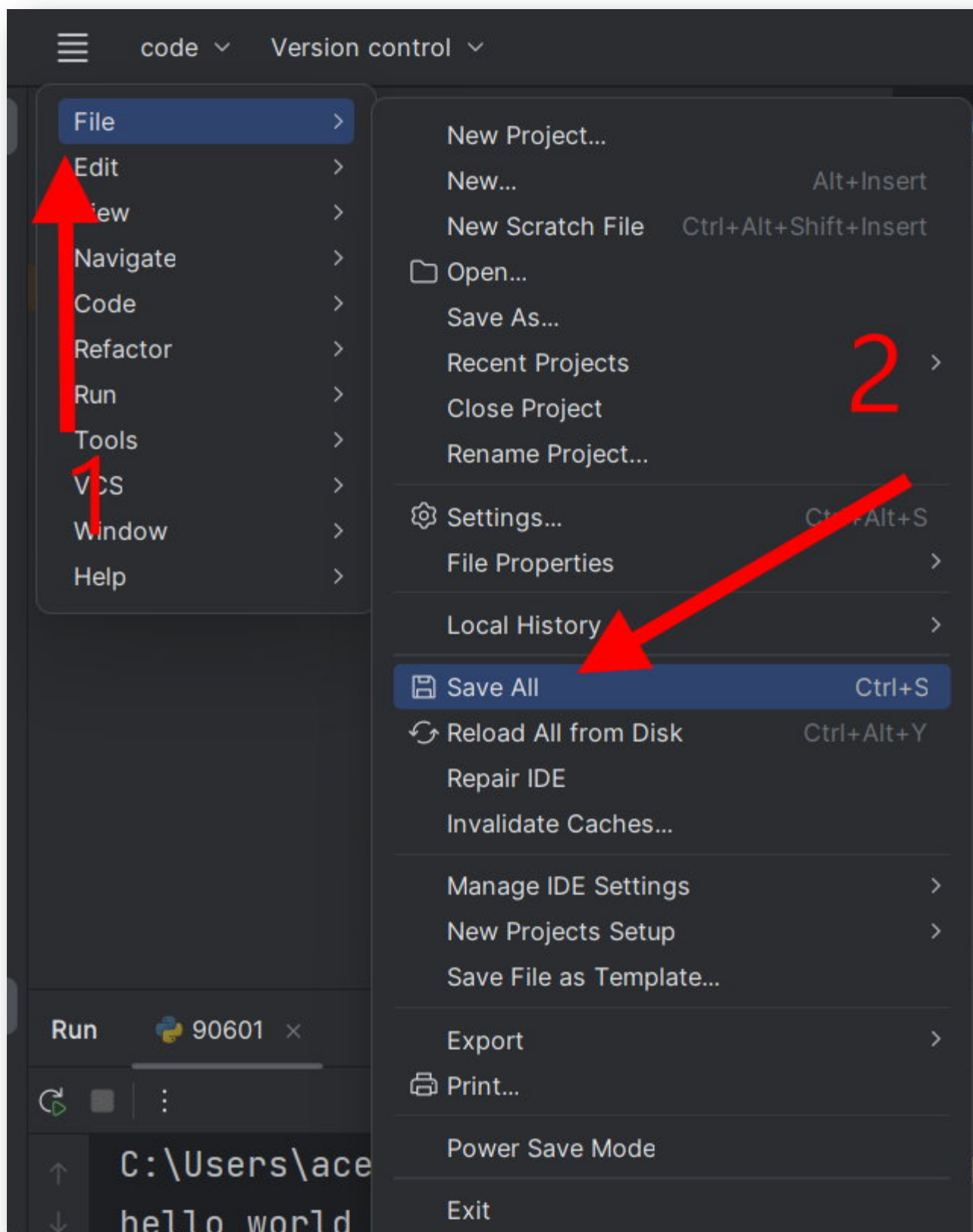
1. 執行 `print((2>5) or (3==2))` 的結果是? 並簡單解釋理由。
2. $x = 1$, $y = 2$, $z = 3$ 執行指令 `print ((y > x) and (y < z))` 的結果是? 並簡單解釋理由。
3. 承上題，執行指令 `print(not ((y > x) and (y < z)))` 的結果是? 並簡單解釋理由。

4. `x = True , y = False` 執行指令 `print ((x or y) and x)`

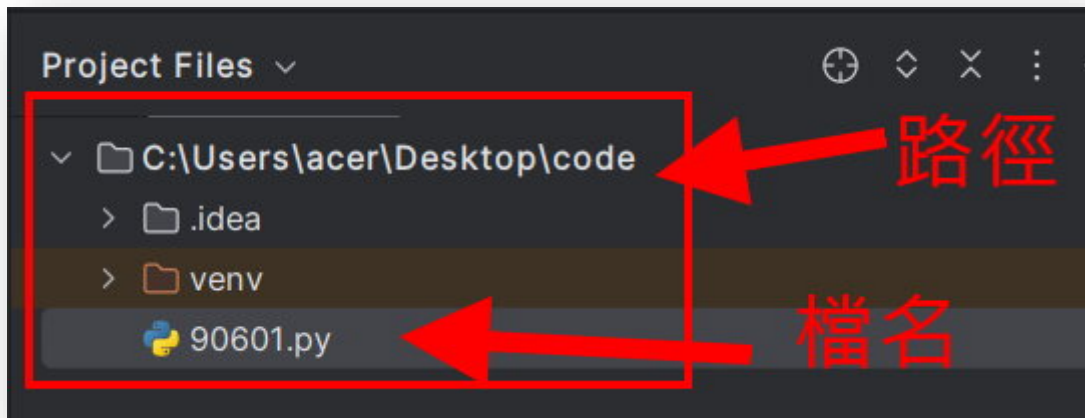
的結果是？並簡單解釋理由。

三、儲存檔案，將檔案上傳到 classroom。

如何儲存檔案



檔案會儲存在畫面路徑的資料夾內:



請把這個檔案上傳到 google classroom 的作業區