stanCode

標準程式教育機構

Assignment 4

This assignment is based on the Assignment 3 of CS106AP at Stanford University

作業檔案下載

這份作業將訓練同學 python 基本影像處理技能來打造一個屬於自己的 photoshop！

我們將使用史丹佛老師 Nick Parlante 撰寫的程式 simpleimage.py 進行影像偵測 (Pixel Detection)、與像素操作 (Pixel Manipulation)，我們也會在第五題帶領同學熟悉綠屏處理 (remove green screen) 的技巧，為作業最後一題 P 圖競賽做準備！

期末考結束會請同學票選 最佳 P 圖賞(Best Photoshop Award)
得獎者將獲得獎學金 500 元！同學們加油！

本次作業的繳交時間為 02/05 (五) 23:59
作業估計需要時間為 **8 小時**
本次作業不接受遲交

如果作業卡關 **歡迎各位到社團提問**，也非常鼓勵同學們互相討論作業之 **概念，但請勿把 code 給任何人看**（也不要把程式碼貼在社團裡）分享妳/你的 code 會剝奪 其他學生獨立思考的機會，也會讓其他學生的程式碼與你/妳的極度相似，使防抄襲軟體認定有抄襲嫌疑

Problem 1 - fire.py

影像分析應用裡很重要的一環就是請電腦自動偵測是否有森林大火發生，並及早通知當地政府前往救災。在 Assignment 4 資料夾 **images** 裡有一個名為 **greenland-fire.png** 的衛星圖像。請同學編輯 **def highlight_fires(filename)** 這個 function 並標記出大火發生的區域（如下圖所示）

以下三點重點提醒：

- 如果一個 **pixel.red** 的數值大於該 pixel **RGB 數值平均** 乘上我們定義的 **HURDLE_FACTOR** 即為我們想要的大火 pixel
- 當您偵測到大火 pixel 時，請將該 **pixel.red** 數值設成 **255**、**pixel.green** 數值設成 **0**、**pixel.blue** 數值設成 **0**
- 當您偵測到非大火 pixel 時，請將該 pixel 設成 **grayscale** (RGB 數值都等於他們的平均)

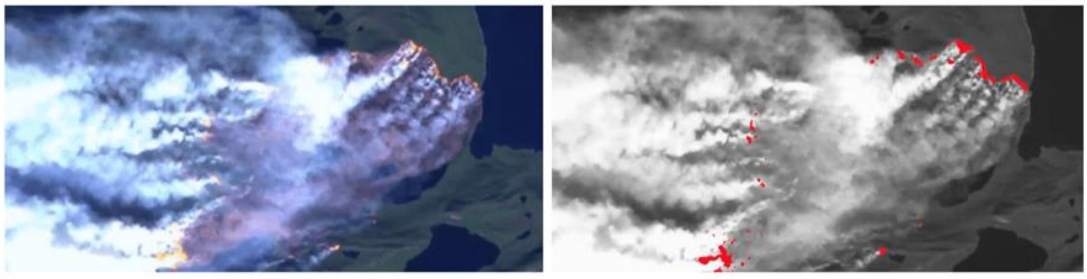


Figure 1. **greenland-fire.png** 森林大火影像處理前（左圖）與處理後（右圖）

Problem 2 - mirror_lake.py

如果我們將一風景影像與其倒轉影像上下並排，往往可以產生美不勝收的湖面鏡像錯覺。請同學在 **mirror_lake.py** 檔案並製造一個跟 **mt-rainier.jpg** 同寬 (width)、兩倍高 (height) 的新影像。新影像的上半部與 **mr-rainier.jpg** 一模一樣，但下半部為上半部的鏡面對稱（如下圖所示）。換句話說，上半部最上方的 pixels 要完美複製到下半部最底層的 pixels

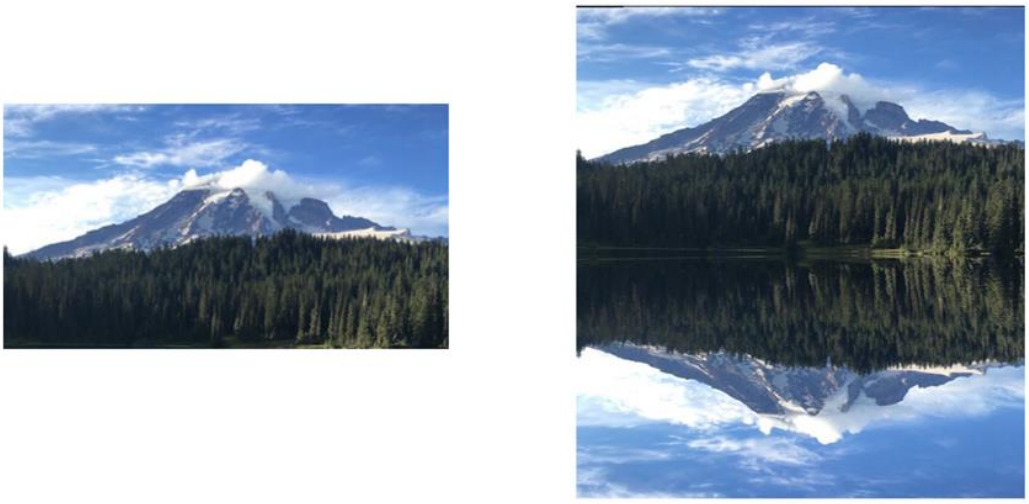
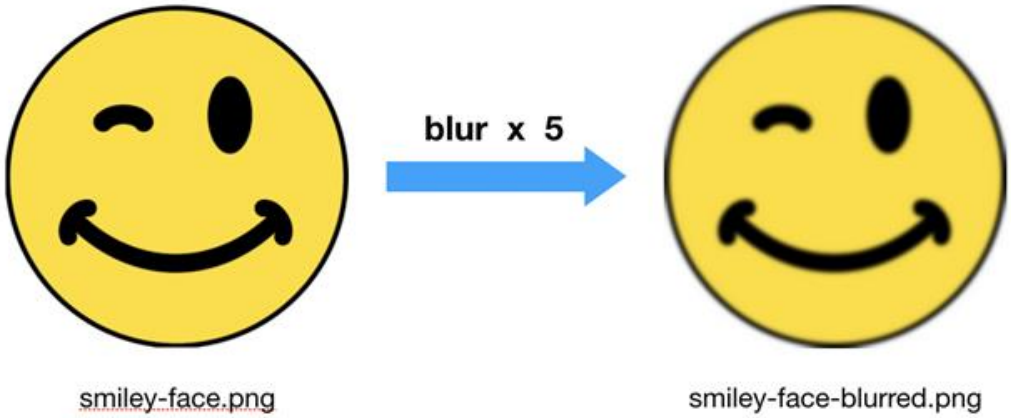
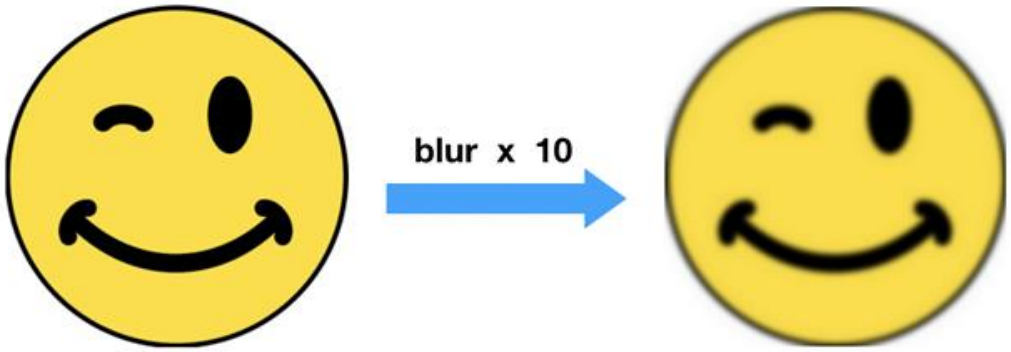


Figure 2. **mr-rainier.jpg** 影像（左圖）與您輸出之新影像（右圖）

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





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)	**(32, 67, 12)**	(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

以下五點請注意：

- 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（您應該會使用到 **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)**

- 在 def main( ) 裡我們使用 for loop 呼叫您要編輯的 blur 四次來達到更明顯的模糊效果（如下圖之程式碼所示）。然而，因為第一次模糊處理的對象為 old_img，因此，雖然 for loop 只反覆四次，但其實您的原圖已經被模糊處理了五次。（同學可以將 for loop 的次數改為 9 次，您的圖像就可以被模糊十次了！）

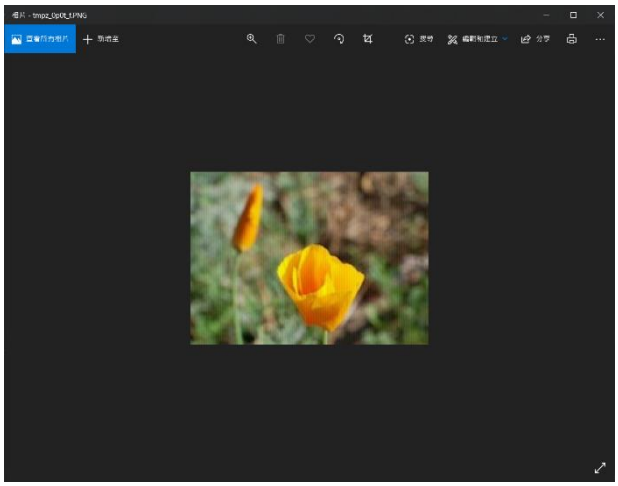
- 此題的運算量極大，大約需要**一分鐘**的運算時間，請同學耐心等候

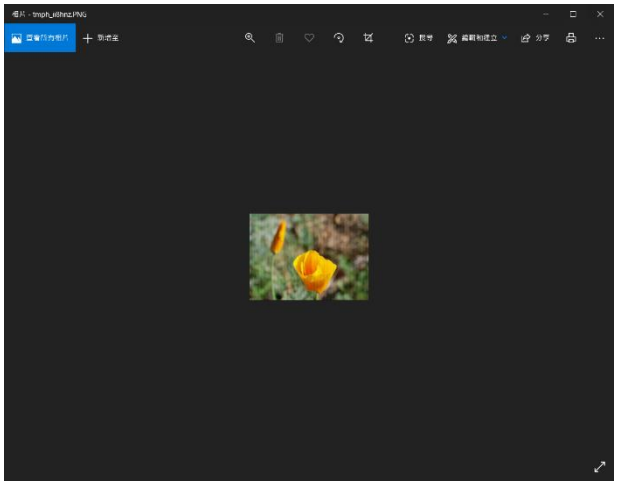
```
def main():
    old_img = SimpleImage("images/smiley-face.png")
    old_img.show()

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

Problem 4 - shrink.py

做影像處理時常常會需要用縮放圖片的這個功能，因此第四題要請同學編輯 def shrink(filename) 並 return 一張將原圖的寬和高等比例縮小 1/2 的新影像！

這題困難之處不在於程式撰寫的部分，而在於想法！同學可以思考一下：縮小之後的影像資料量比原圖少許多，那我們該怎麼選取原圖影像資料才能達到縮小的目標呢？演算法有許多種！歡迎同學發揮創意寫出自己的縮放程式！**請勿使用 make_as_big_as**





Problem 5 - greenscreen.py

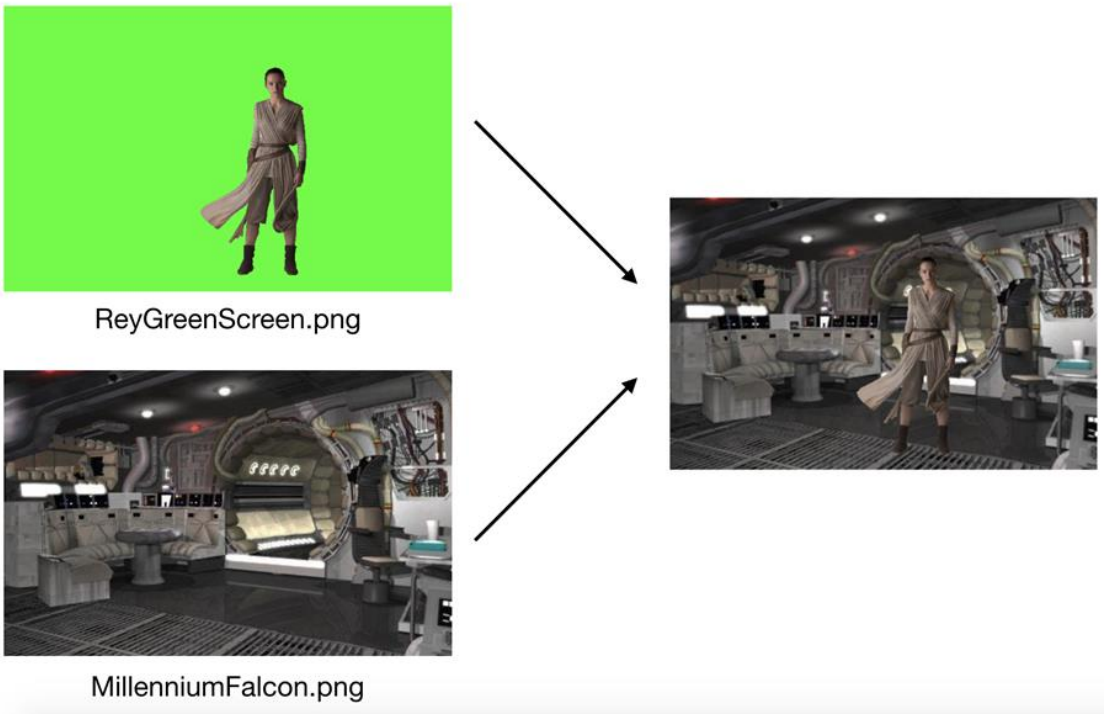
第五題要請同學完成好萊塢電影特效最重要的一環 - 綠屏處理

您將 **return** 一張綠屏 pixels 都被變成 **MillenniumFalcon.png** 的新影像。而**綠屏 pixel** 在這邊定義為 green 數值大於「red, blue 較大數值的兩倍」。舉例來說，若某個 pixel 的 (red, green, blue) 分別為 (23, 83, 42)，它**不能**被稱為綠屏 pixel 因為 red, blue 較大的數值為 42，而 83 並沒有大於 2*42 == 84

下方的程式碼應該會派上用場：

```
bigger = max(pixel.red, pixel.blue)  # returns the one that is bigger
```

請您一個 pixel 一個 pixel 地去查看它是否為綠屏；如果是，請將相對應位置的 (red, green, blue) 以 MillenniumFalcon.png 影像的 (new_red, new_green, new_blue) 取代。完成之後您應該會看到右下圖之影像



Problem 6 - best_photoshop_award.py

最後一題，要請同學發揮創意，將自己的綠屏照片 P 到其他圖片。您的作品會在 **02/07(日)** 上課讓全班投票，我們將頒發獎學金 500 元給最高票得獎者

- 輸入圖檔有兩個方法：(1.) 直接將想要的檔案拖拉到 PyCharm 裡名為 **image_contest** 的資料夾 (2.) 打開電腦中名為 Assignment 4 的資料夾並將檔案放入 **image_contest**

- 若同學要調整名為 **background** 的影像讓他跟您自己的影像 **figure** 大小一樣，您可以使用

background.make_as_big_as(figure)

以下四點重點提醒：

- 請在 **def main( )** 的 function comments 撰寫創作理念/靈感來源
- 請將所有您用到的原檔圖片放置於 **image_contest** 之資料夾中
- 一個人只會有一張影像參加競賽，因此請同學們務必挑選最滿意的作品再上傳作業檔案

最後想跟各位同學說加油！教學團隊非常期待看到各位的成果。發揮創意，獎金一定是妳/你的！

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迴圈（Infinite loop）之中.

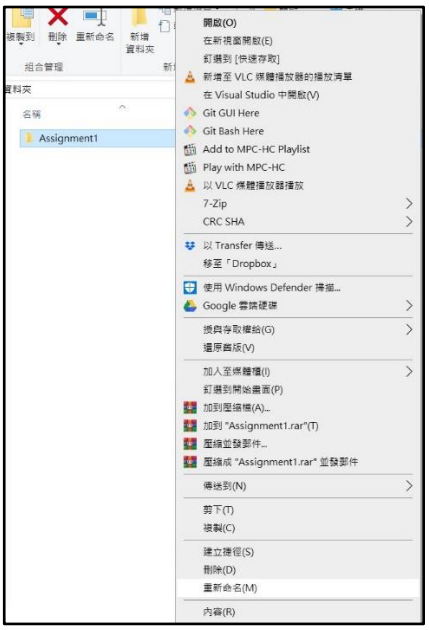
Style - 如同我們在課堂上所說，好的程式要有好的使用說明，也要讓人一目瞭然，這樣全世界的人才能使用各位的 code 去建造更多更巨大更有趣的程式。因此請大家寫**精簡扼要** 的 main( ) 程式概要、function 敘述、單行註解.

作業繳交

恭喜各位完成 Assignment 4! 大家應該要對自己的成就感到驕傲，因為這份作業跟史丹佛大學的學生作業非常相似，代表你們跟世界各國的菁英一樣厲害了

請同學於 **02/05 (五) 23:59** 前依照下圖將作業上傳

找到作業資料夾，按右鍵

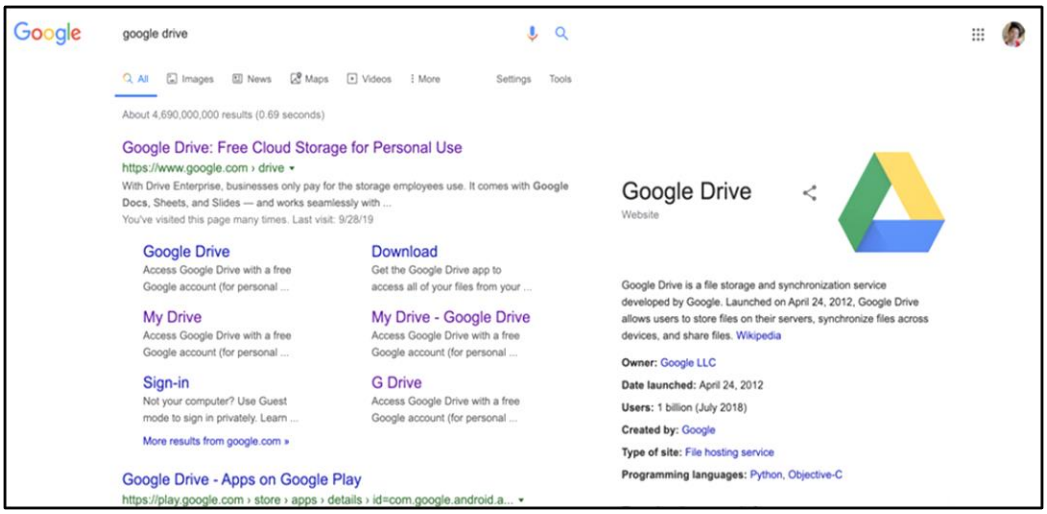


點選重新命名

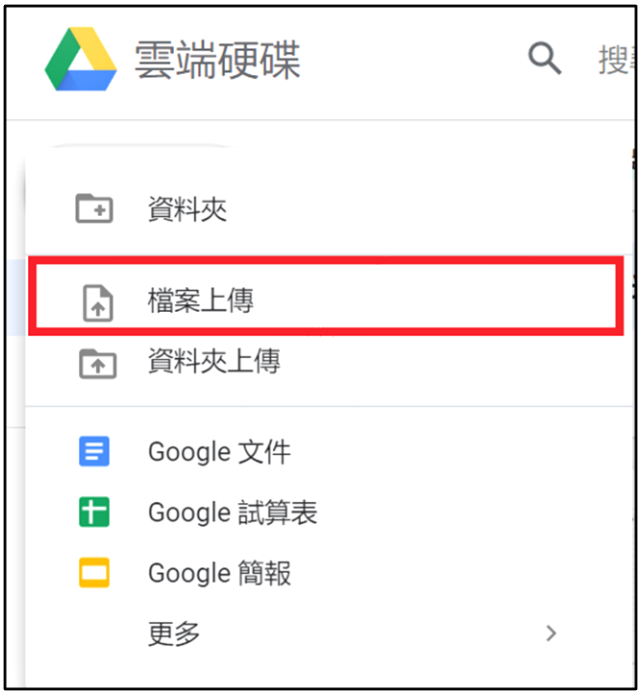
請命名成「**Assignment4_**中文姓名」的格式

Windows 請點選「傳送到」->「壓縮的(zipped)資料夾」

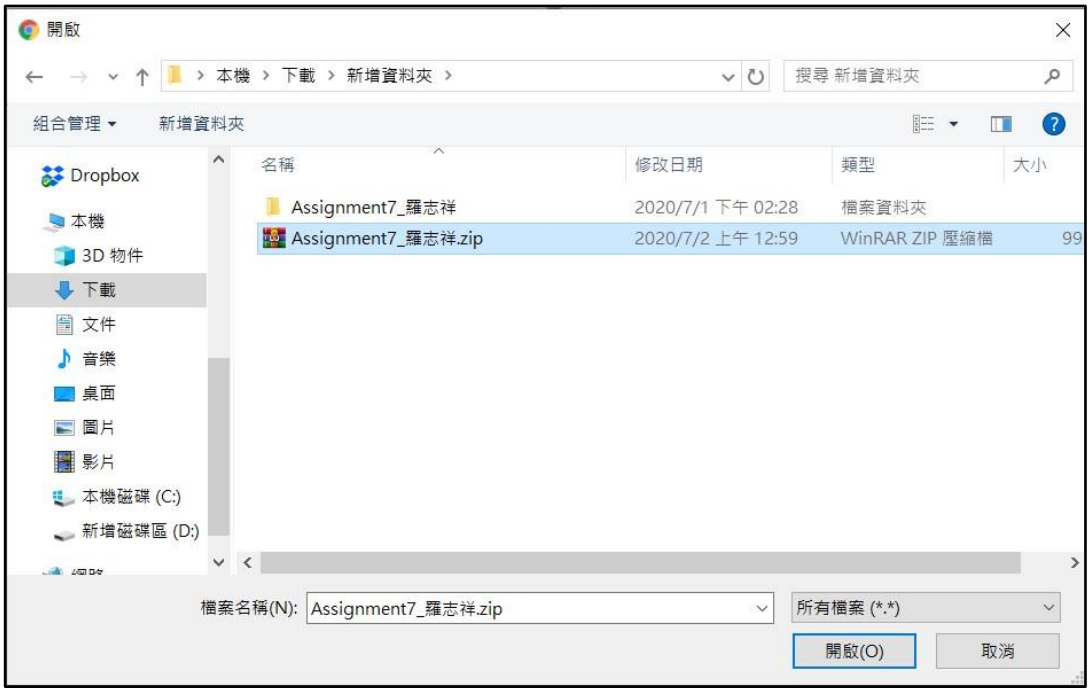
Mac 請點選 Compress "Assignment4_中文姓名"



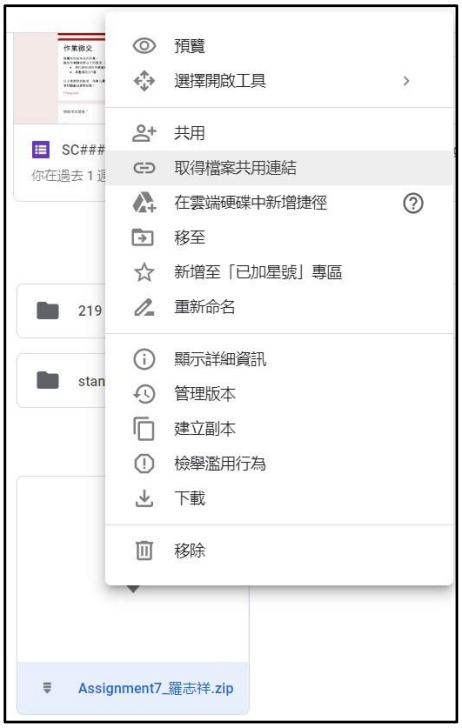
上網搜尋 **Google Drive**



點選 File upload（檔案上傳）



找到剛剛壓縮的 Assignment4 檔案後，點選 Open（開啟）



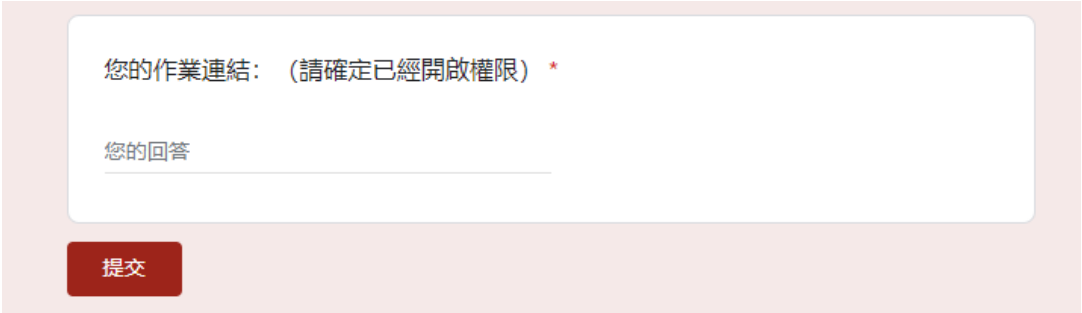
上傳完成後，對檔案按右鍵，選擇 **Get shareable link**（取得檔案共用連結）

將「**Restricted**」（限制）改成「**Anyone with the link**」（知道連結的使用者）

再按下 **複製連結**

點開以下 **google** 表單：

https://forms.gle/zDWdLW4mUYbG4aR49



填入相關資訊，並在最後附上剛剛複製的連結

確認無誤後按下提交

stanCode - 標準程式教育機構

Should you have any questions please feel free to contact us.