

stanCode

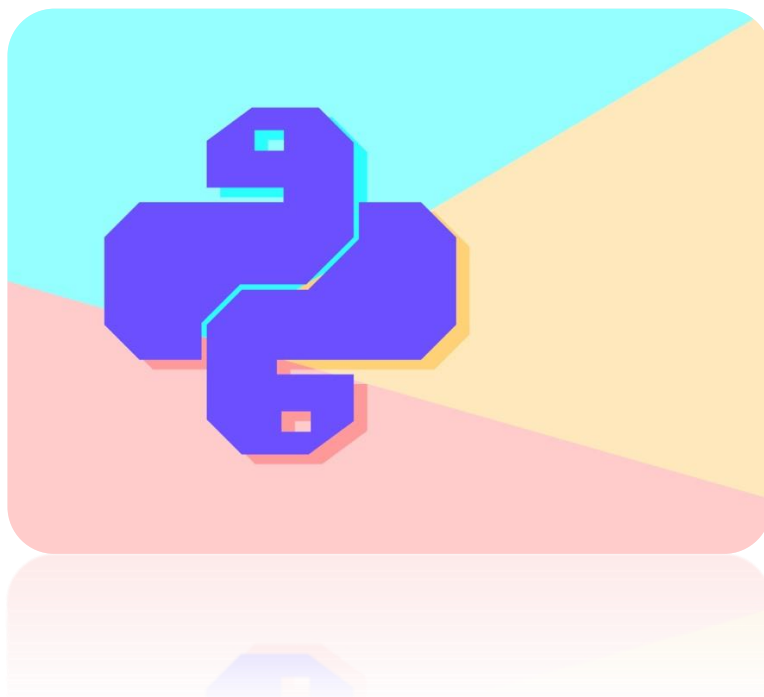
標準程式教育機構

Assignment 1

This assignment is based on handouts written by Jerry Liao, Shang-Chun Tai

Image credit: devclass.com

作業檔案下載



歡迎各位同學來到 SC101

這份作業將帶領同學熟悉由史丹佛老師 **Sam Redmond** 撰寫的 **campy** 包裹，習得 **Python** 繪圖、動畫製作、遊戲製作的基本技能！其中，滑鼠偵測 (**Mouse Event**) 將是下一份打磚塊作業的重要基石。透過本次作業題目，同學們不僅可以建構打磚塊遊戲裡的關鍵技術，也將慢慢了解電腦科學裡非常具有挑戰性的 **class & object** 概念：繪圖元素都是一個 **object**，而每個可以製造繪圖元素的工廠都是一個 **class**

同學們提交的 **Problem 1 – my_drawing.py** 作品將於 **Week3** 課堂上票選出
「最佳 **GObjects** 繪圖大賞」冠軍

得獎者將獲得獎學金 **500** 元！期待看到同學們的創意巨作！

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（也不要將程式碼貼在社團裡）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



Image credit: Bella Wang

Problem 1 – my_drawing.py

同學們上課時練習了如何使用 **campy** 資料夾裡面的 **GOval, GRect** 畫一張臉，再
來這個部分請同學發揮創意自由揮灑！展現您的創意，讓 **GOval、GRect、GLine、**
GLabel 排列出不一樣的美（若您想要使用上課畫的臉也行）請記得將創作點子/理
念寫在 **def main()** 下方的程式註解：

您將使用位於 **campy.graphics.gobjects** 這個檔案裡的四個大程式 (Classes)，並從那些大程式中呼叫您的「小小兵」(Objects/Instances)

若要製作一個名為 **face**，寬 100、高 200 的 **GOval** 並將之放置於 (101, 101):

```
face = GOval(100, 200, x=101, y=101)
```

若要製作一個名為 **window**、寬 800、高 400 的 **GWindow**:

```
window = GWindow(width=800, height=400)
```

若要將 **face** 夾到 **window** 上:

```
window.add(face)
```

然而，如果 Jerry 要改變 **face** 邊框顏色為洋紅色，Jerry 要「請 **face** 做絕招」

```
face.color = 'magenta';
```

下圖是往年競賽的優勝作品給大家參考。非常期待看到您的創意！

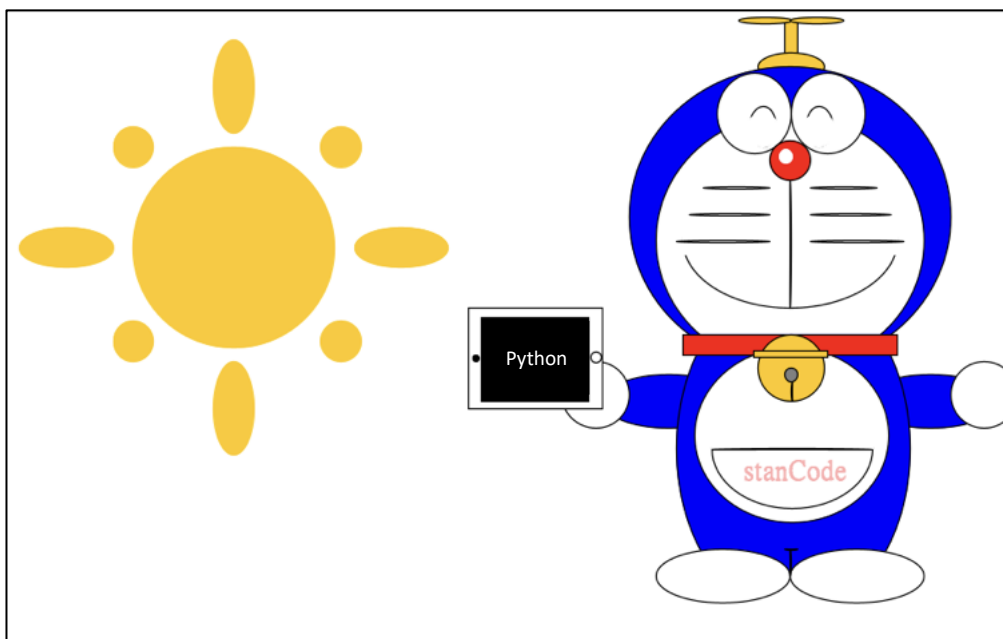
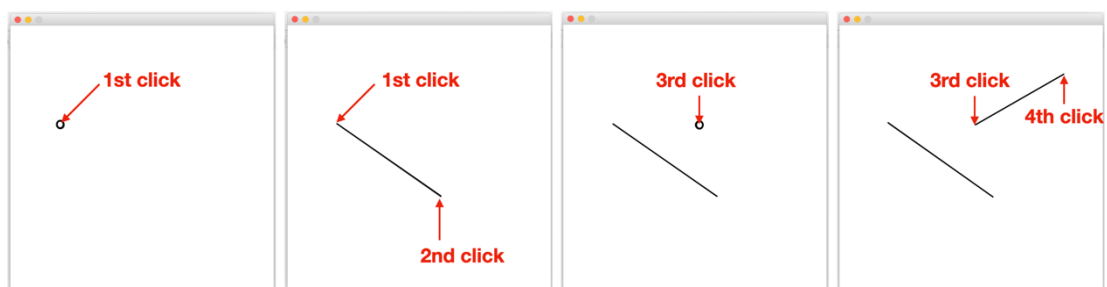


Image credit: Cheng-Chu Chung

Problem 2 – draw_line.py

在上課範例中，我們成功地寫出了「小畫家」應用程式裡面的畫筆！現在讓我們來擴充小畫家的功能：讓使用者可以在視窗上連續的任意兩點畫出一條線！

當使用者第一次在視窗上點擊 (click)，視窗會以滑鼠點選的位置為圓心畫一個黑色的空心圓，告訴使用者剛剛點擊的位置（空心圓的直徑我們定義在名叫 **SIZE** 的常數）當使用者於視窗再度點擊 (click)，程式將以此點為終點，在視窗畫出一條線。請注意：整個過程使用者只有點擊視窗，沒有拖拉 (Drag)。程式執行畫面的範例請見下圖（紅色的箭頭與文字是我們後製上去的！真正的程式畫面只有圓形與線條）



- 左一圖片是使用者第一次點擊視窗之畫面
- 左二圖片是使用者第二次點擊而產生線條的畫面 - 線條起點為第一次點擊產生之圓心、終點為滑鼠第二次點擊的座標。請注意：線條出現後，第一次點擊的圓就會從畫面上消失
- 左三圖片是使用者第三次點擊視窗的畫面；左四圖片則是使用者第四次點擊視窗的畫面。請注意：第三次點擊視窗確實有出現圓圈，只是因為第四次點擊而消失了

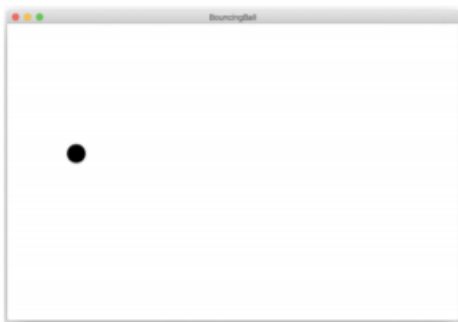


Problem 3 – bouncing_ball.py

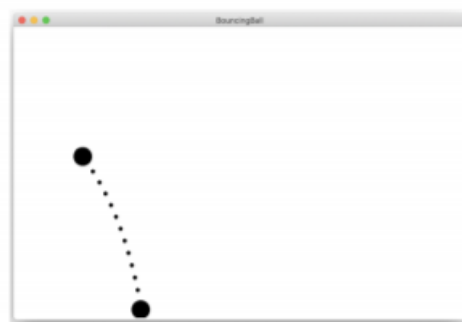
最後一題將是下週魔王作業 - 打磚塊 - 的前哨戰！各位同學將模擬球的彈跳與碰撞（下方六張圖為一顆實心黑球連續彈跳過程之示意圖。虛線是後製上去的，以輔助題目解釋）

影片說明：

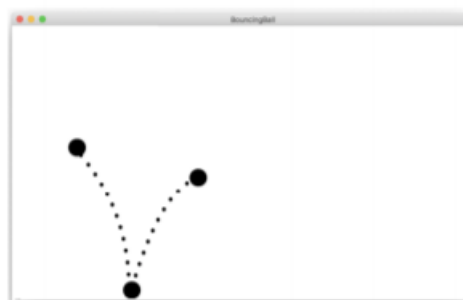
https://drive.google.com/file/d/1vPaz3uptAGF1LcvL81TH_urtQUa3V-0c/view?usp=sharing



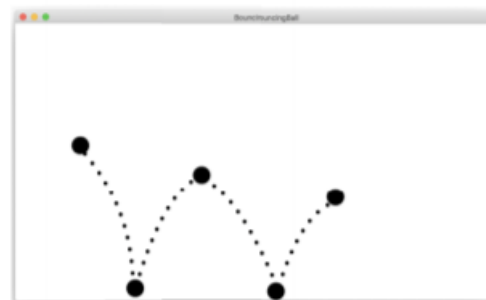
圖(一) 球的起始位置



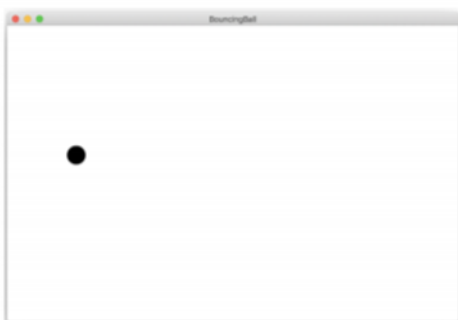
圖(二) 第一次與地板反彈之示意圖



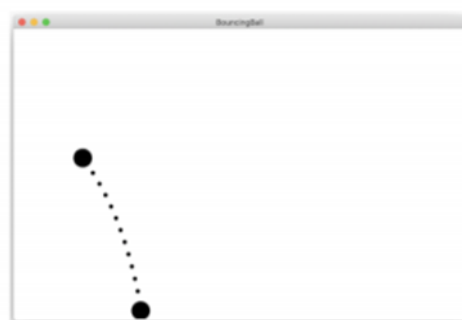
圖(三) 第一次反彈至最高點示意圖



圖(四) 第二次反彈至最高點示意圖



圖(五) 球超出視窗後回到原位示意圖



圖(六) 點按視窗，球又再次墜落

Constant 介紹（請務必使用下方的每一項）

- **VX** - 球的水平速度
- **DELAY** - 動畫停格多少毫秒(ms)
- **GRAVITY** - 模擬重力加速度。每一圈 **while loop** 垂直速度要加上的數值
- **SIZE** - 球的直徑
- **REDUCE** - 每一次反彈，球垂直速度所剩的比例
- **X_START, Y_START** - 球的起始位置

請製作一個動畫（在使用者點擊視窗時才開始執行）讓球可以模擬受重力影響而彈跳的過程！

球彈跳過程的水平速度請保持 **VX**。但由於垂直速度受重力影響，往下時會越來越快！因此，同學在每一次動畫更新球移動的距離時（也就是 **ball.move**）請在垂直速度加上我們定義的常數 **GRAVITY** 來模擬受重力的影響結果

當球與地板碰撞，垂直速度將改為向上（且受重力影響的關係，球的速度呈現越來越慢的趨勢）為了模擬與地板物理碰撞的現象，「垂直速度」會因為碰撞而損失變成「碰撞前的 **0.9**（也就是我們定義的常數 **REDUCE**）」

當整顆球離開視窗最右側時，請讓球**回到原初始位置**，等待使用者在視窗任一處點擊，球才會再度開始彈跳、碰撞

請注意：球在彈跳、碰撞過程**不會受使用者點擊視窗影響**。當使用者重複了碰撞模擬過程 **3** 次（即球超出右側視窗 **3** 次）球將回到原初始位置，且這時若使用者再點擊視窗，程式不會有任何反應

這題最困難的地方莫過於「動畫要在使用者點擊後」才能開始執行，且我們要確保遊戲可以重複這個特性 **3** 次。同學可以思考：什麼樣的程式設計可以產生類似閘門的功能：將程式卡住，等待使用者點擊後才打開開關？

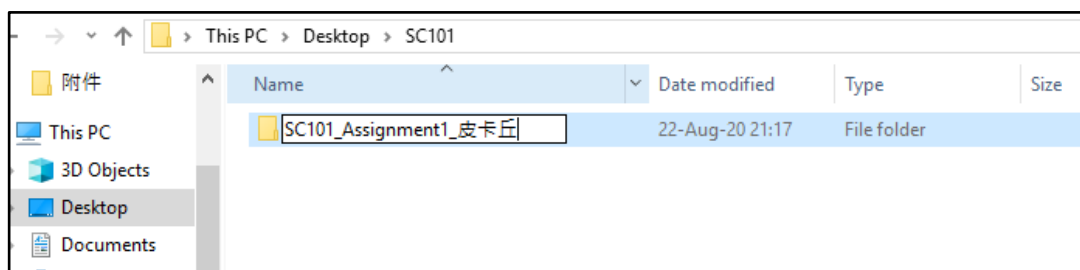
若同學遇到球在地板彈不起來（好像卡在地板上下震動）這是因為您定義反彈為速度方向變號 – 由正轉負、由負轉正。想一想：我們是否可以讓球碰到地板後確保速度一定是向上的？

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環圈（**Infinite loop**）之中

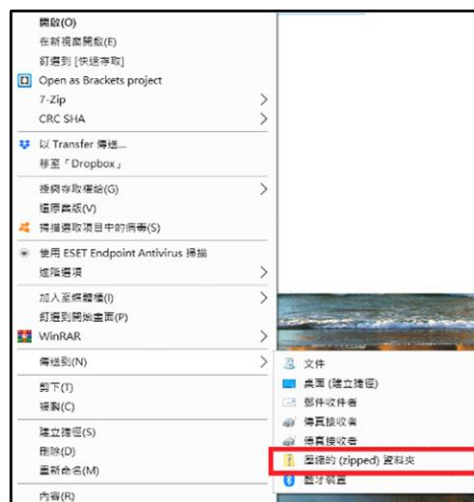
Style - 如同我們在課堂上所說，好的程式要有好的使用說明，也要讓人一目瞭然，這樣全世界的人才能使用各位的 **code** 去建造更多更巨大更有趣的程式。因此請大家寫**精簡扼要**的使用說明、**function** 敘述、單行註解

作業繳交



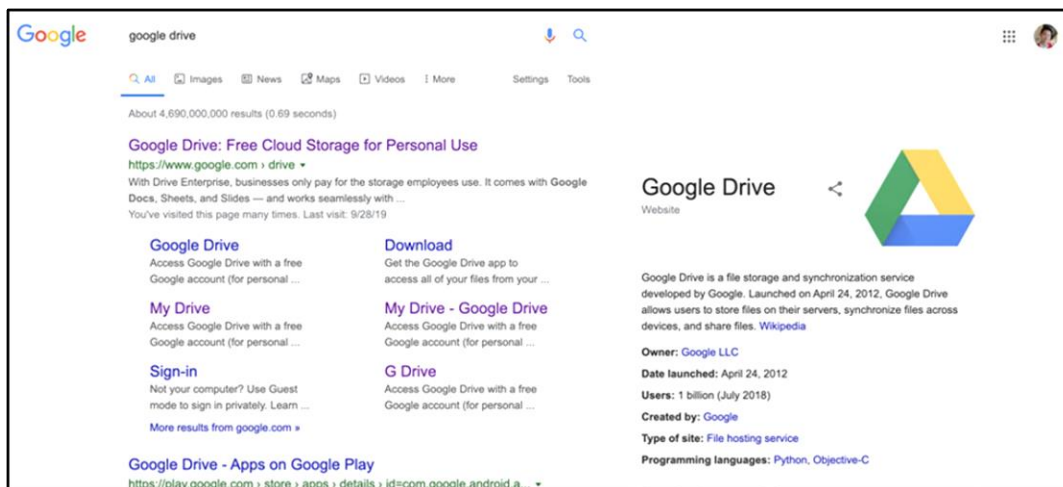
找到作業資料夾，重新命名成以下格式「**SC101_Assignment#_中文姓名**」

如：**SC101_Assignment1_皮卡丘**



Windows 請點選「傳送到」->「壓縮的(zipped)資料夾」

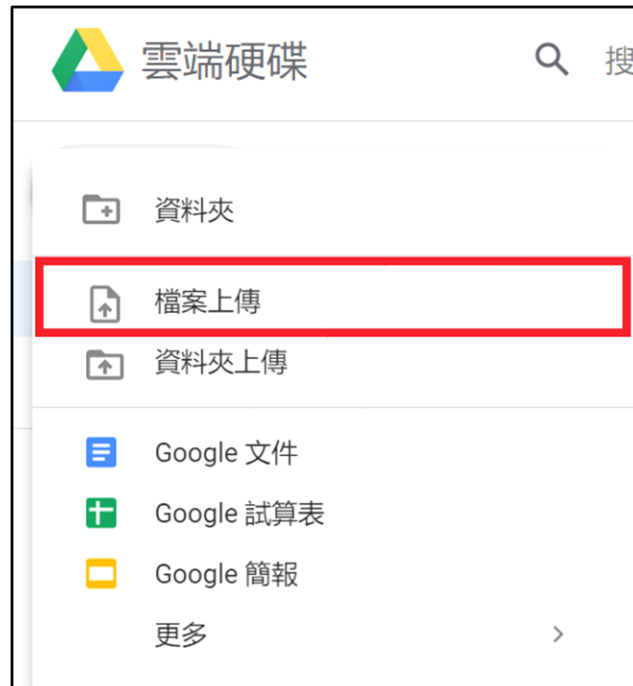
Mac 請點選 **Compress "SC101_Assignment1_皮卡丘"**



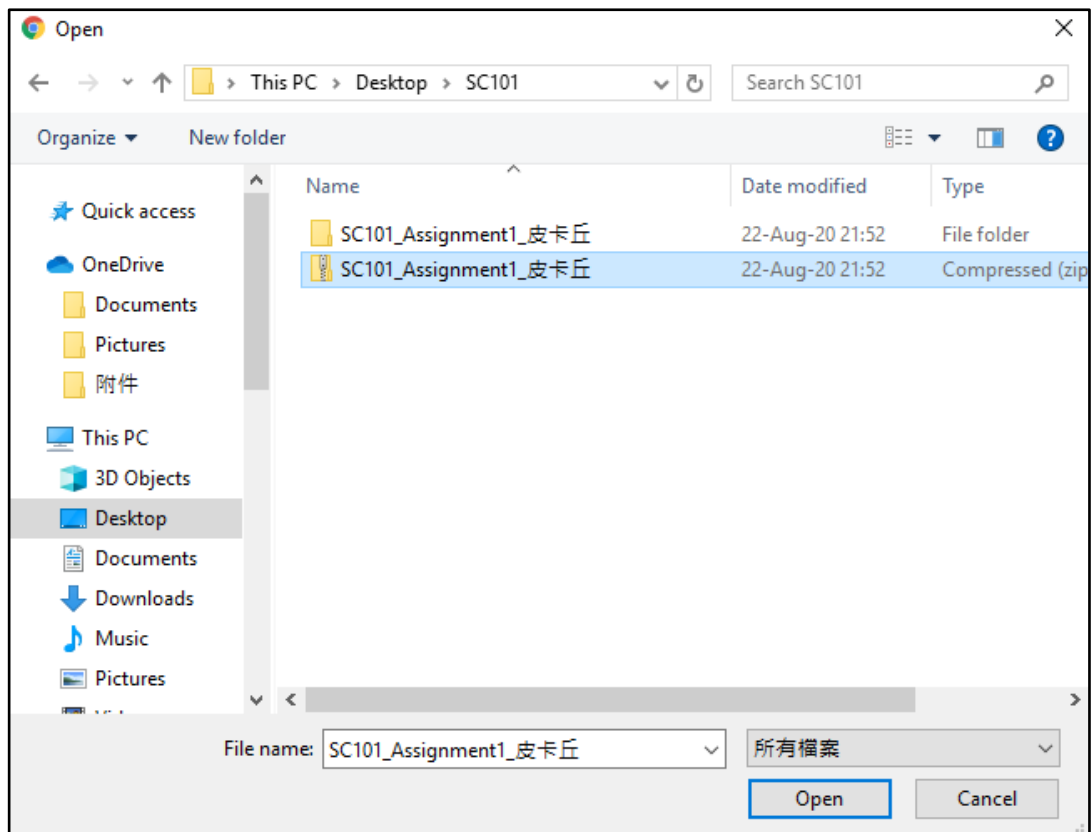
上網搜尋 Google Drive



點選 New（新增）



點選 **File upload** (檔案上傳)



找到剛剛壓縮的 **SC101_Assignment1_皮卡丘.zip** 檔案後，
點選 **Open**（開啟）



上傳完成後，對檔案按右鍵，選擇 **Share**（共用）



點選右上角 **Get shareable link**（開啟連結共用設定）

與他人共用

開啟連結共用設定

連結共用設定已開啟 [瞭解詳情](#)

知道連結的人均可以檢視 ▼

複製連結

https://drive.google.com/file/d/1XG0m8R6jdQs-tKP2FQzOnZBuQzpcEo07/view?usq

使用者

輸入名稱或電子郵件地址...

這個檔案的檢視者可以查看註解和建議。 [瞭解詳情](#)

完成

進階

複製連結，傳至社團提供的「作業題交表單」

stanCode

stanCode - 標準程式教育機構

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