

課程名稱

計算機程式 (Computer Programming)

課程目標

本課程旨在培養學生掌握計算機程式設計的基本概念與實務能力，主要透過 Python 程式語言進行教學，目標是使學生具備設計程式解決問題的能力，並能將學到的技能應用於機械設計相關的工程問題中。

課程內容

1. 基礎程式設計概念

- Python 環境設置與簡介
- 變數與資料型態
- 基本輸入與輸出
- 運算符與邏輯表達式

2. 流程控制

- 條件判斷 (if-else)
- 迴圈結構 (for 與 while)
- 巢狀控制結構與應用

3. 函式與模組

- 自訂函式
- 函式的參數與回傳值
- 使用內建模組與第三方模組

4. 資料結構

- 列表 (List)、元組 (Tuple)、集合 (Set) 與字典 (Dictionary)
- 資料結構的應用與處理

5. 檔案處理

- 讀取與寫入檔案
- 檔案處理中的錯誤檢測

6. 進階主題

- 物件導向程式設計 (OOP) 基本概念
- 類別與物件的定義與使用
- 繼承與多型的應用

7. 機械設計應用

- 使用 Python 進行簡單的工程計算
- 開發與機械設計相關的簡單工具或模擬

W7

```
1 a="我已經學會簡單的python程式了"
2
3 for i in range(1, 11):
4     print(1, a)
```

Filename: .py

Run Output 清除輸出區 清除繪圖區 Reload

[illegible]

W9

Brython. triangle1

1 add to 100

```
1 n = 5 # 總共的行數
2 space = ' '
3 for i in range(n): # 對於每一行
4     # 印出空格
5     print(space * (n - i - 1), end='') # 在每行前面印出空格
6     # 印出星號
7     print('*' * (2 * i + 1)) # 每行印出 2*i +
8
9 from IPython.display import IFrame
10
```

Filename: .py

```
*
***
*****
*****
*****
*****
*****
Traceback (most
recent call last):
File
```

Brython. triangle2

從 1 累加到 100 part2:

1 add to 100 cango_three_gears BSnake AI Tetris Rotating Block

```
1 n = 5 # 定義金字塔的總行數，這裡我們用 n=5，表示金字塔共有 5 行
2 space = ' '
3 # 第一部分：打印上半部（包括中間行）
4 for i in range(1, n + 1): # 從第1行到第n行，i代表行數
5     # 這裡我們打印空格，讓星號可以靠右對齊
6     # 空格數量是 (n - i)，因為第一行需要 4 個空格，第二行需要 3 個...
7     print(space * (n - i), end='') # end='' 防止換行，讓星號在同一行
8
9     # 打印星號。每一行的星號數量是 i，從 1 開始逐行增加
10    # 第一行 1 顆星，第二行 2 顆星，第三行 3 顆星...這樣增加
11    print('*' * i) # 每行輸出 i 個星號
12
13    # 第二部分：打印下半部（去除中間行）
14    for i in range(n - 1, 0, -1): # 這裡我們從 n-1 行開始，逐行減少
15        # 空格的數量依然是 (n - i)，這是因為下半部的對稱性
16        # 比如當 i=n-1 時，空格數量是 1，當 i=n-2 時，空格數量是 2...
17        print(space * (n - i), end='') # 同樣使用 end='' 來避免換行
18
19        # 打印星號數量為 i，從 n-1 開始逐行減少
20        # 所以當 i = n-1 時，打印 n-1 顆星，當 i = n-2 時，打印 n-2 顆星...
21
```

Filename: .py

```
*
 *
**
 ***
****
*****
*****
****
 ***
**
 *
```

<completed in 6.30

w10_hw

題目一

心得與做法：

利用 codespace、brython、jupyter、cmd、scite 執行程式，我覺得我有學習到每個編譯器的操作過程

把前面學到的東西統整起來 有學習到程式相關的東西

題目二

心得與做法：

scite、codespace、jupyter、brython、cmd 的正確與錯誤程式樣式

了解到正確與錯誤的語法差異

題目三

心得與做法：

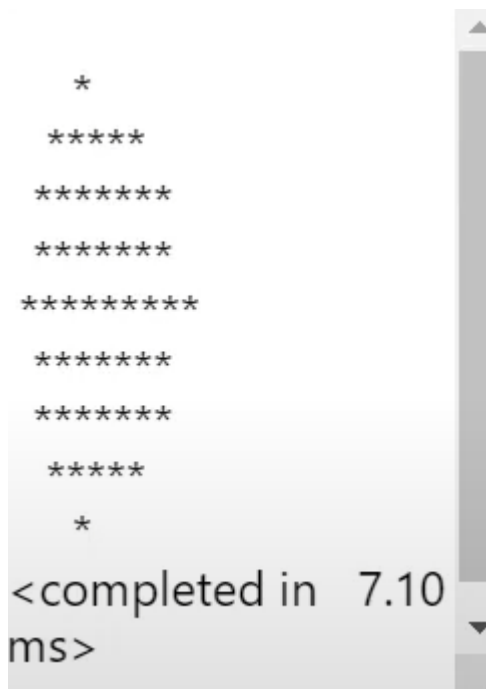
利用 scite、cmd、codespace、jupyter、brython 進行三個物理及一個微積分程式，讓我更熟悉每一個程式編譯器的操作模式，也讓我了解物理及數學問題可以使用程式解答

w11_hw

1. 已知可以用 Python 程式由上往下列印三角形 *，或由左往右列印三角形 *，請將此程式延伸到可以在 10x10 的字元區域中，以 "*" 字元列印出圓型區域。完成後請將程式碼存入個人 Gist 後，直接在個人作業網站中的 Brython 頁面中執行。（提示：?src=gist_script_URL）

心得：有稍微看一下程式內容，嘗試去理解意思。

解題過程：利用*由左而右即由上而下做三角形，我利用 chatgpt 幫求出答案



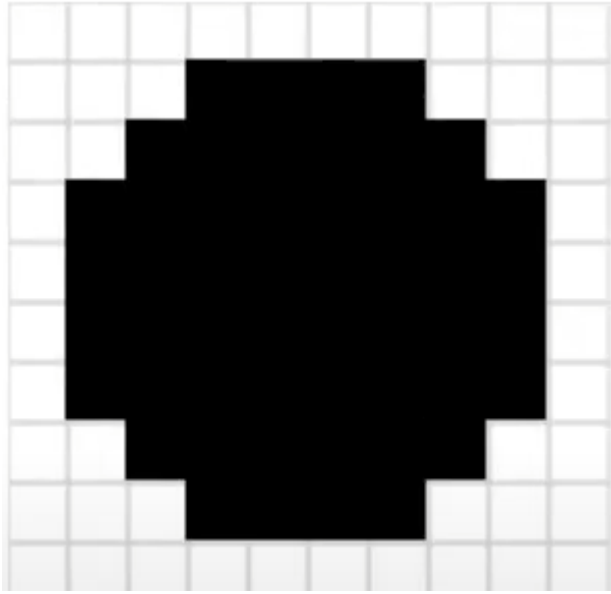
```

      *
     *****
    *****
   *****
  *****
 *****
 *****
 *****
  *****
   *
<completed in 7.10
ms>
```

2. 已知可以在網頁中利用 Brython 繪製方格與塗色，如下連結，請設法修改此一程式，直接將程式存入個人的 Gist，並利用黑色方塊圍出一個圓形區域。

心得：有稍微看一下程式內容，嘗試去理解意思。

解題過程：利用 chatgpt 求解得出程式內容。



w12_hw

1. 請各組組員分別利用 Brython 以及 Jupyter lab 環境，執行可攜程式 notebook 目錄中的 01_string_input_print.ipynb 至 10_for_search.ipynb 筆記本程式，完成後請將能在 Brython 執行的程式碼存入個人 Gist 後，將各程式執行連結放入 w12_hw 網頁上。最後請說明各範例中所使用的程式語法以及執行心得。

心得：透過 Brython 和 Jupyter Lab 執行筆記本程式，加深了對 Python 語法的理解，例如字串處理、迴圈與搜尋演算法的應用。過程中熟悉了 Brython 的瀏覽器環境，提升了程式執行與問題解決的能力。

程式碼：

```
n = 5

space = "_"
'''

for i in range(n): #對於每一行
    #印出空格
    print(space * (n - i - 1), end='')
'''

for i in range(n): #對於每一行
    #印出空格
    print(space * (n - i - 1), end='')
'''

for i in range(n): #對於每一行
    #印出空格
    print(space * (n - i - 1), end='')
    print("目前的數字為 " + str(i))
```


w13_hw

```
from
browser
import
html

    from browser import document as doc

    # 利用 html 建立 canvas 超文件物件
    canvas = html.CANVAS(width=400, height=400)
    brython_div = doc["brython_div1"]
    brython_div <= canvas

    ctx = canvas.getContext("2d")
    ctx.lineWidth = 1

    # 設定對角線顏色為藍色
    ctx.strokeStyle = 'blue'
    ctx.beginPath()
    ctx.moveTo(40, 40)
    ctx.lineTo(270, 270) # 第一條對角線
    ctx.stroke()

    # 繪製第一個正方形，邊框顏色為紅色
    ctx.strokeStyle = 'red' # 設定正方形邊框顏色為紅色
    ctx.beginPath()
    ctx.rect(40, 40, 230, 230) # 設定正方形的位置和大小
    ctx.stroke() # 繪製邊框

    # 繪製第二個正方形，邊框顏色為紅色
    ctx.strokeStyle = 'red' # 設定邊框顏色為紅色
    ctx.beginPath()
    ctx.rect(140, 140, 220, 220) # 設定正方形的位置和大小
    ctx.stroke() # 繪製邊框

    # 繪製兩個正方形的外圍圓形，顏色為黑色
```

```

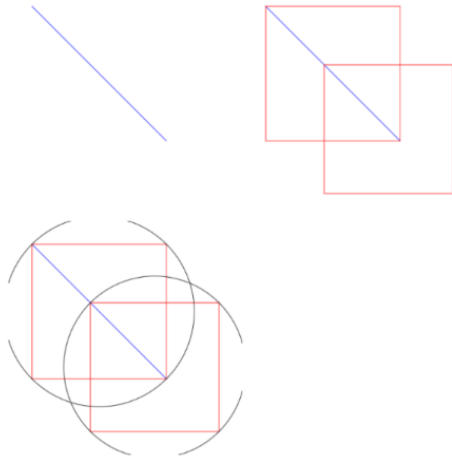
# 計算第一個圓的中心和半徑
center_x1 = 40 + 230 / 2 # 第一個正方形的 X 中心
center_y1 = 40 + 230 / 2 # 第一個正方形的 Y 中心
radius1 = (230 * (2 ** 0.5)) / 2 # 半徑為對角線的一半

# 設定圓的顏色為黑色
ctx.strokeStyle = 'black'
ctx.beginPath()
ctx.arc(center_x1, center_y1, radius1, 0, 2 * 3.14159) # 繪製圓
ctx.stroke() # 描邊

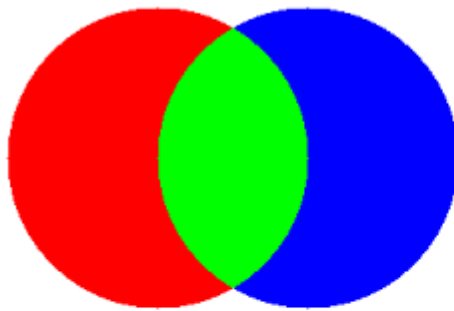
# 計算第二個圓的中心和半徑
center_x2 = 140 + 220 / 2 # 第二個正方形的 X 中心
center_y2 = 140 + 220 / 2 # 第二個正方形的 Y 中心
radius2 = (220 * (2 ** 0.5)) / 2 # 半徑為對角線的一半

# 繪製第二個圓
ctx.beginPath()
ctx.arc(center_x2, center_y2, radius2, 0, 2 * 3.14159) # 繪製圓
ctx.stroke() # 描邊

```



w14_hw



從 1 累加到 100 part2:

1 add to 100 | cango_three_gears | BSnake | AI Tetris | Rotating Block

```
20     canvas.get_context('2d')
11     return canvas, ctx
12
13 # 判斷點是否在圓內
14 def is_point_in_circle(px, py, cx, cy, r):
15     return (px - cx) ** 2 + (py - cy) ** 2 <= r ** 2
16
17 # 填充區域的主函式
18 def draw_circles(x1, y1, r1, x2, y2, r2):
19     canvas, ctx = initialize_canvas(400, 400)
20     ctx.clearRect(0, 0, canvas.width, canvas.height)
21
22     # 定義顏色
23     color_circle1 = "rgb(255, 0, 0)" # 圓1顏色 (紅色)
24     color_circle2 = "rgb(0, 0, 255)" # 圓2顏色 (藍色)
25     color_intersection = "rgb(0, 255, 0)" # 交集區域顏色 (綠)
26     color_background = "rgb(255, 255, 255)" # 背景顏色 (白)
27
28     # 掃描畫布中的每個像素
29     for py in range(0, canvas.height):
30         for px in range(0, canvas.width):
31             # 判斷點是否在圓內
```

w15_hw

- w15_anchor

```
1  # sum 初始值設為 0
2  sum = 0
3  init = 1
4  upto = 100
5  # 利用 for 重複迴圈與變數加法進行累加
6  for i in range(init, upto+1):
7      sum = sum + i
8  print("從" + str(init) + "累加到" + str(upto) + "=" + str(sum))
```

- w15_addto

```
1  def addto(start, end):
2      total = 0
3      for i in range(start, end + 1):
4          total += i
5      return total
```

- w15_add_only_even

```
1  def add_only_even(start, end):
2      total = 0
3      for i in range(start, end + 1):
4          if i % 2 == 0: # 判斷是否為偶數
5              total += i
6      return total
```

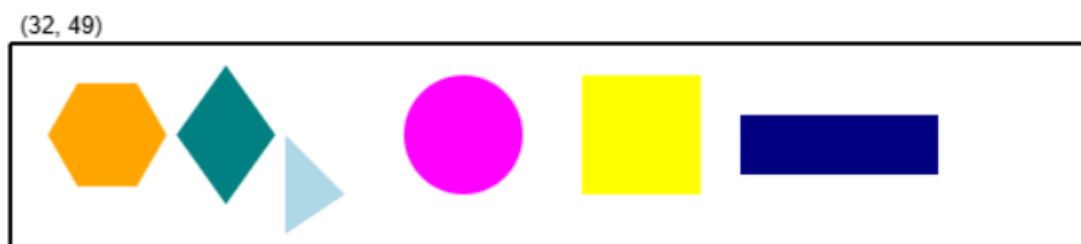
- w15_add_avoid_8

```
1  def add_avoid_8(start, end):
2      total = 0
3      for i in range(start, end + 1):
4          if '8' in str(i): # 如果數字包含 "8"，則跳過
5              continue
6          total += i
7      return total
```

w16_hw

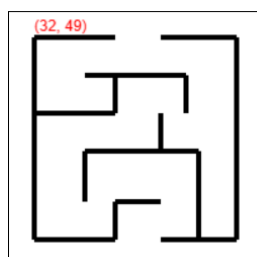
題目：請以自己的學號最後四碼作為繪圖的座標原點，（例如：若學號最後四碼為 3932，請將 (39, 32) 作為繪圖座標原點），並以 pixel=2 的黑色直線分別利用 Brython 繪圖，標示出向右為正的 X 軸，以及向下為正的 Y 軸，並且利用文字標示出原點座標（例如：(39, 32)），之後請自選最左側圖形的起始點座標，直接在頁面畫出下列圖像，繪圖程式除了必須直接列在頁面 html 檔案外，也請存入自己的 Gist 區域，並在所完成的圖像下方，以自選標題的 anchor 連結至 Gist，讓使用者點擊該連結後，可以在各自的 Brython 頁面中繪出下列圖像

心得：問了 chatgpt 很多問題，才解出這個題目，但到了期末也比較知道怎麼問問題以及問題出在哪裡並除錯。



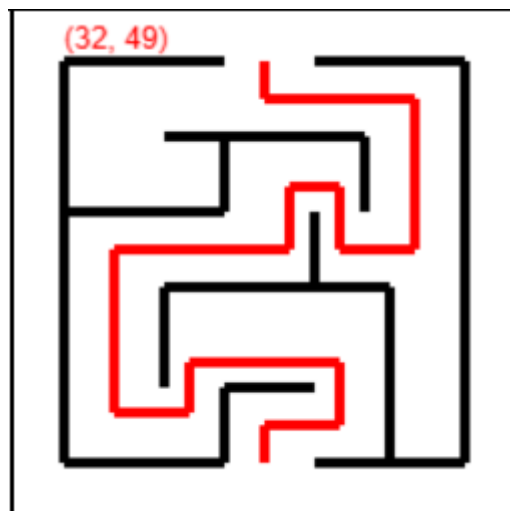
請以自己的學號最後四碼作為下列繪圖的左上方點座標，（例如：若學號最後四碼為 3932，請將 (39, 32) 作為繪圖左上方點座標），並以紅色文字標示出該點座標（例如：(39, 32)），之後請利用 pixel=2 的藍色直線，自訂迷宮畫布大小，直接在頁面畫出下列圖像，繪圖程式除了必須直接列在頁面 html 檔案外，也請存入自己的 Gist 區域，並在所完成的圖像下方，以自選標題的 anchor 連結至 Gist，讓使用者點擊該連結後，可以在各自的 Brython 頁面中繪出下列圖像）。

心得：先用 chatgpt 大概了解一下程式結構，在一步一步改成與題目一樣的圖形。



請以自己的學號最後四碼作為下列繪圖的左上方點座標，（例如：若學號最後四碼為 3932，請將 (39, 32) 作為繪圖左上方點座標），並以紅色文字標示出該點座標（例如：(39, 32)），之後請利用 pixel=2 的黑色直線，自訂迷宮畫布大小，直接在頁面畫出下列圖像，並且利用紅色連續線段標示從上方入口，走至下方出口的路線圖。

心得:用與 exam2 的相同觀念，將線條改成紅色的把路徑用程式呈現出來，越來越熟悉。



期末心得：

這學期的「計算機程式」課程，真的讓我對程式設計有了很大的突破與收穫。一開始學 Python 的時候，其實有點吃力，因為對變數、字串操作這些概念還不熟悉，但透過課堂上的一步步講解，加上 Jupyter Lab 的實作練習，我慢慢掌握了基礎語法，像是條件判斷、迴圈還有函式設計，後來甚至還接觸到了物件導向這種進階主題。特別是這次作業中，我在 Brython 和 Jupyter Lab 兩個環境裡，執行了從 01_string_input_print 到 10_for_search 的筆記本程式，這過程讓我學會了不同環境的特性和使用方式。Jupyter Lab 的即時執行結果很直觀，而 Brython 讓我覺得很酷，因為可以直接在瀏覽器執行 Python 程式，這對網頁應用來說真的很實用。實作的過程中，我不只對程式語法更熟悉了，還學會了怎麼把程式碼上傳到 Gist，並把執行連結分享到作業網站上。這不僅是寫程式，還是一次完整的學習流程，包含撰寫、測試、上傳與分享，讓我對程式設計的實用性有更深的體會。總的來說，這堂課讓我對程式設計有了更大的興趣，也讓我對自己的能力更有信心，希望未來可以用這些技能解決更多實際問題，甚至做出一些有趣的專案！