

E-game 派森試煉 教學應用

高雄市教育網路中心團隊

kiec-edu@mail.kh.edu.tw

研習課程

- Python快速入門篇
班級代碼：Ngr52
- E-Game篇
 - [課程模式](#)
 - 本課程在因材網有相同教材與影片
 - [自學模式](#) (300題)
 - [冒險歷程](#) (關卡模式)
- [問題集](#)
- [操作手冊](#)
- 服務信箱-教網中心教育企劃組
 - kiec-edu@mail.kh.edu.tw

E-Game

派森試練

- 派森試練

如何進入派森試練

登入 E-game

- [E-game](#)入口
 - 開放時間：
每日7~23時
- 教學影片
 - [達學堂](#)影片



點選 [打寇島]

- 打寇島 是 程式類 的島嶼

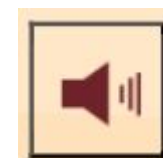




點選 [派森試煉]



開關聲音



派森試煉主畫面

- 四種模式
 - 課程模式
 - 自學模式
 - 冒險歷程
 - 線上對戰



派森試煉

- 按鈕的用途
(預設規劃)



課程模式



課程模式

課程模式						
1 名稱	2 單元	大綱	3 課程影片	4 課程講義	5 課後小練習	
1. Python開發環境	1 - 1	Python開發環境				
	1 - 2	撰寫第1個程式				
	1 - 3	編輯器選項設定				
	1 - 4	總結				冒險模式 1
2. 變數命名與指派	2 - 0	上次課程總複習				
	2 - 1	變數使用				
	2 - 2	變數命名				
	2 - 3	格式化輸出				
	2 - 4	格式化輸出				
	2 - 5	第2單元實作練習				
			Rows per page 10 < 1 2 3 4 5 6 7 >			



課程模式 - 課程影片

- 提供錄製的教學影片供學生可以重複觀看學習。



回主功能

返回



課程模式 - 課程講義

- 提供課程講義，免除學生抄寫並可以重點標記。

三 程式語言基本概念- 2變數命名與指派 4 / 12 | - 100% + | [Icons]

1 Python開發環境

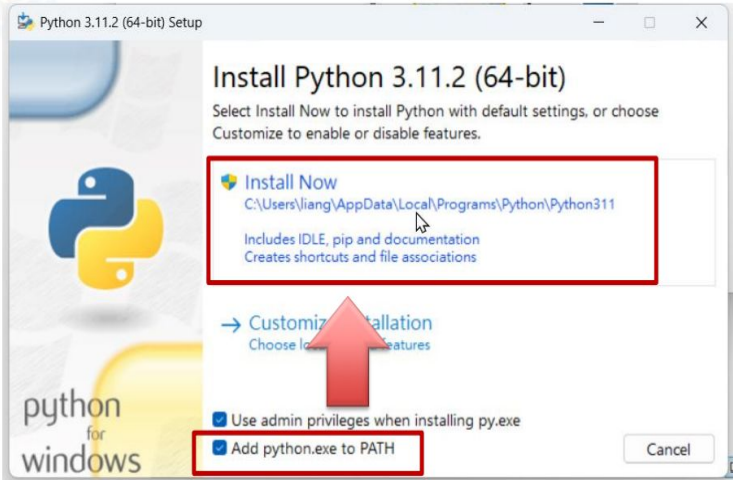
2 Python程式下載安裝

3 Python下載

4 安裝Python

安裝Python

1. 點擊執行python-3.11.2-amd64.exe
2. 安裝時要勾選「Add python.exe to PATH」後，再點選「Install Now」。





課程模式 - 課後小練習 / 介面說明

- 提供課後練習 - 了解自己的學習成效。

The screenshot shows the E-game interface with several key components and annotations:

- Header:** Features the "E-game" logo, a character, and various icons. On the right, there are buttons for "程式碼" (Code), "積木" (Blocks), "並列" (Parallel), a toggle switch, a score of "20", and buttons for "設定" (Settings) and "重置" (Reset).
- Left Panel (Annotations):**
 - 題目說明 (Question Description):** A yellow box with an arrow pointing to the problem text: "使用 input()取得姓名，輸出「Hello 姓名」(中間有一個空格)。提示: 可從選項複製正確的程式碼執行" (Use input() to get the name, output "Hello Name" (with a space in the middle). Hint: You can copy the correct code from the options to execute).
 - 選項 (Options):** A list of four code snippets: (A) `print('Hello', input())`, (B) `print(input(), 'Hello')`, (C) `print('Hello ', input())`, and (D) `print('Hello' + input())`.
 - 輸入說明 (Input Instructions):** States "可以為任何英文字或中文" (Can be any English letters or Chinese).
 - 輸出說明 (Output Instructions):** States "Hello 姓名。Hello 和姓名中間要有一格空格" (Hello Name. There must be a space between Hello and the name).
 - 範例輸入 #1 (Example Input #1):** A section for example input.
 - 返回 (Return):** A button at the bottom of the left panel.
- Right Panel (Annotations):**
 - 程式編輯區 (Code Editor):** A large area for writing code, indicated by a yellow box and an arrow.
 - 作答回饋 (Answer Feedback):** A section for receiving feedback on the answer, indicated by a yellow box and an arrow.
- Bottom Bar:** Contains buttons for "終端機測試" (Terminal Test), "資料測試" (Data Test), and "送出結果" (Submit Results).



課程模式 - 課後小練習 / 按鈕說明

The screenshot shows the E-game interface with various components and annotations:

- Header:** Features the "E-game" logo, a strawberry character, and navigation buttons: "</> 程式碼", "積木", "並列", a toggle switch, "20", "設定", and "重置".
- Left Panel:**
 - Instructions:** "使用 input()取得姓名，輸出「Hello 姓名」（中間有一個空格）。" and "提示: 可從選項複製正確的程式碼執行".
 - 選項 (Options):** A list of four Python code snippets (A, B, C, D) for printing "Hello" and user input.
 - 輸入說明 (Input Instructions):** "可以為任何英文字或中文".
 - 輸出說明 (Output Instructions):** "Hello 姓名。Hello 和姓名中間要有一格空格".
 - 範例輸入 #1:** A text input field with a "返回" (Return) button.
- Code Editor:** A central area for writing code, with a line number "1" at the top.
- Terminal/Output Area:** At the bottom, with tabs for "終端機" and "測試".

Annotations (Yellow boxes with red arrows):

- 程式編輯區切換顯示:** Points to the "</> 程式碼" button in the header.
- 程式碼字體大小:** Points to the "20" font size indicator in the header.
- 資料測資 測試:** Points to the "測試" (Test) button in the bottom right.
- 送出結果 進行評測:** Points to the "送出結果" (Submit Results) button in the bottom right.
- 程式編譯 執行:** Points to the "執行" (Run) button in the bottom right.
- 返回上一頁:** Points to the "返回" (Return) button in the bottom left.



課程模式 - 課後小練習 / 實作示範

</> 程式碼 積木 並列 20 設定 重置

使用 input()取得姓名，輸出「Hello 姓名」（中間有一個空格）。
提示: 可從選項複製正確的程式碼執行

選項

1

(A) print('Hello', input())
(B) print(input(), 'Hello')
(C) print('Hello ', input())
(D) print('Hello' + input())

輸入說明

可以為任何英文字或中文

輸出說明

Hello 姓名。Hello 和姓名中間要有一格空格

範例輸入 #1

返回

1 print('Hello ', input())

2 可以參考「選項」輸入程式碼

3 確認答案是否正確

輸入 [John]後執行

py.egame.k
Jhon
確定 取消

終端機 測試

終端機測試 資料測試 送出結果



課程模式 - 課後小練習

答案**正確**

```
1 print('Hello',
```

終端機

測試

測試結果

AC

範例 1

正確

結果說明

測資狀態

範例 1

+

輸入

John

預期輸出

Hello John

新增測資：

1. 使用者可以自行增加測資，用來驗證自己寫的程式能不能符合自己編寫的資料。
2. 這些新增的測試資料，不會進入隱藏測資的結果評分。

終端機測試

資料測試

送出結果

答案**錯誤**

```
1 print('Hello' + i
```

終端機

測試

測試結果

NA

範例 1

錯誤

結果說明

WA: Wrong Answer 表示答案錯誤，請仔細比對，務必符合題目要求

測資狀態

範例 1

+

輸入

John

預期輸出

Hello John

終端機測試

資料測試

送出結果



課程模式 - 課後小練習

E-game

</> 程式碼 積木 並列 20 設定 重置

```
1 print('Hello', input())
```

(A) print('Hello', input())
(B) print('Hello', input())
(C) print('Hello', input())
(D) print('Hello' + input())

輸入說明
可以為任何英文字或中文Python 開發環境 – 取得輸入資料

輸出說明
Hello 姓名。Hello 和姓名中間要有一格
空格

範例輸入 #1
John

範例輸出 #1
Hello John

返回

課程模式 評分結算

單元名稱	Hello John	評價說明	經驗值	距離下一個技能等級還差
測資 1	隱藏 1	☒ 測資全部通過 ☒ 執行時間 < 標準值 ☒ 硬體負荷通過	+ 9 點	79 點

送出結果 的 評分結算

AC

測資 1

輸入
John
輸出
Hello John
預期輸出
Hello John

終端機測試 資料測試 送出結果

課程模式 - 課後小練習

課程模式評分結算

Python 開發環境 - 取得
輸入資料

- ✓ 測資全部通過
- ✓ 執行時間 < 標準值
- ✓ 硬體負荷通過

+ 5 點

79 點

通過

測資評



輸入

John

輸出

Hello John

預期輸出

Hello John

測資 1

隱藏 1

測資 1

不通過



輸入

John

輸出

Hello
John

預期輸出

Hello
John

測資 1

隱藏 1

測資 1

WA: Wrong Answer



課程模式 - 課後小練習 操作展示影片

自學模式 (評測系統)



自學模式

自學模式

🏆 排行榜

難易度

分類

搜尋

5

4

3

1

2

序號	名稱	難易度	分類	建立時間	通過率
● 1	最大公因數	簡單	函數與資料型態, 公因數	2023-10-11 10:15:42	0.6183
● 2	進制轉換	簡單	函數與資料型態	2023-10-11 10:15:42	0.9181
● 3	列表中最大值和最小值	簡單	函數與資料型態	2023-10-11 10:15:42	0.9136
● 4	BMI 計算機	簡單	變數命名與指派, 運算子及運算元	2023-10-11 10:15:42	0.8245
● 5	簡易四則計算機	簡單	基本輸出輸入	2023-10-11 10:15:42	0.8996
● 6	三角形判斷	簡單	流程控制, 幾何	2023-10-11 10:15:42	0.8701
● 7	質數判斷	簡單	流程控制, 質數	2023-10-11 10:15:42	0.8996
● 8	計算圓面積	簡單	變數命名與指派, 模組	2023-10-11 10:15:42	0.9033
● 9	計算平均成績	簡單	變數命名與指派, 字元與字串, 容器資料	2023-10-11 10:15:42	0.9369
● 10	計算三角形面積	簡單	變數命名與指派, 幾何	2023-10-11 10:15:42	0.8683

每頁筆數

10

<

1

2

3

...

28

29

30

>

🎓 自學模式 - 分類

中等 困難

函數與資料型態

公因數

變數命名與指派

運算子及運算元

流程控制 幾何

質數 模組

字元與字串 容器資料

字典 排序 極值

迴圈 基本算數

資料型態 單向選擇

多向選擇 字串切片

格式化輸出 串列

集合 統計 元組

字典合併 遞迴

排序演算法 選擇排序

插入排序 搜尋演算法

搜尋 矩陣操作

字串操作 數學公式

自訂函數 簡單

學習人數通過率

難易度

分類

搜尋...



建立時間

通過率

2023-10-11 10:15:42

0.5388

2023-10-11 10:15:42

0.8416

2023-10-11 10:15:42

0.8636

2023-10-11 10:15:42

0.6857

2023-10-11 10:15:42

0.7213

2023-10-11 10:15:42

0.6866

2023-10-11 10:15:42

0.5349

2023-10-11 10:15:42

0.7879

2023-10-11 10:15:42

0.7097

2023-10-11 10:15:42

0.7754

Rows per page 10 < 1 2 3 ... 28 29 30 >



示範演練 - 01.最大公因數 (二數輸入方法)

輸入兩個正整數，求它們的最大公因數。

輸入說明

每組輸入一行，包含兩個正整數 a 和 b ($1 \leq a, b \leq 10000$)。

輸出說明

輸出一個正整數，
(為 a 和 b 二數的最大公因數。)

範例輸入 #1

24 30

範例輸出 #1

6

```
1  a, b = map(int, input().split()) # 讀取輸入
2
3  # 使用轉輾相除法求最大公因數
4  while b != 0:
5      a, b = b, a % b
6
7  # 輸出結果
8  print(a)
```

終端機 測試

6

示範演練 - 01.最大公因數

```
a, b = map(int, input().split())
```

- 程式說明

- `split()`：這是一個字串方法，用於將字串分割成多個部分，預設會**使用空格作為分割字元**。
- `map()`：這是一個內建函數，用於將指定的函數應用於可迭代物件（在這個例子中是字串列表）。在這個程式碼中，我們指定`int`函數作為要應用的函數，以便將字串列表中的每個元素轉換為**整數型別**。

總結起來，`a, b = map(int, input().split())`這行程式碼的作用是從使用者輸入中讀取一行文字，將其以空格分割成多個部分，並將每個部分轉換為整數型別，然後將這些整數分別賦值給變數`a`和`b`。

示範演練 - 01.最大公因數

輸入兩個正整數，求它們的最大公因數。

輸入說明

每組輸入一行，包含兩個正整數 a 和 b ($1 \leq a, b \leq 10000$)。

輸出說明

輸出一個正整數，
(為 a 和 b 二數的最大公因數。)

範例輸入 #1

24 30

範例輸出 #1

6

```
1 def gcd(a, b):
2     while b != 0:
3         a, b = b, a % b
4     return a
5
6 # 讀取輸入
7 a, b = map(int, input().split())
8
9 # 呼叫函數求解最大公因數
10 result = gcd(a, b)
11
12 # 輸出結果
13 print(result)
```

終端機 測試

6



示範演練 - 07.判斷質數 (答案要完全符合)

判斷一個正整數是否為質數。

輸入說明

輸入一個正整數 n (n 為正整數且不超過 1000)

輸出說明

如果 n 為質數，則輸出 'YES'，否則輸出 'NO'。

範例輸入 #1

3

範例輸出 #1

YES

```
1 n = int(input())
2 is_prime = True
3 if n < 2:
4     is_prime = False
5 else:
6     for i in range(2, int(n**0.5) + 1):
7         if n % i == 0:
8             is_prime = False
9             break
10 if is_prime:
11     print('YES')
12 else:
13     print('NO')
```

終端機

測試

🎓 示範演練 - 08計算圓面積 (數值要完全符合)

- 程式撰寫時會有不同的結果，但輸出要**完全符合**輸出結果 - 2位小數

```
1 r = int(input())
2 a = r*r*3.14
3 print(a)
```

錯誤

終端機 測試

28.26

```
1 r = int(input())
2 a = 3.14*r*r
3 print(a)
```

錯誤

終端機 測試

28.259999999999998

```
1 r = int(input())
2 a = 3.14159*r*r
3 print(round(a,2))
```

```
1 import math
2 r = int(input())
3 a = r*r*math.pi
4 print(round(a,2))
```

示範演練 - 08計算圓面積 (輸入不能有提示詞)

輸入**不能**有提示詞

```
1 import math
2 r = int(input())
3 a = r*r*math.pi
4 print(round(a,2))
```

終端機 測試

測試結果

AC

範例 1

正確

```
1 import math
2 r = int(input('輸入半徑'))
3 a = r*r*math.pi
4 print(round(a,2))
```

終端機 測試

測試結果

NA

範例 1

錯誤

單元名稱
計算至小數點後兩位

計算圓面積

```
1 import math
```

```
2 r = int(input())
```

```
3 a = r*r*math.pi
```

```
4 print(round(a,2))
```

評價說明

經驗值

距離下一個技能等級還差

✓ 測資全部通過

✓ 執行時間 < 標準值

✓ 硬體負荷通過

+ 4 點

25.5 點

一天最多4次可重複送增加
經驗值

測資評分

測資 1

隱藏 1

測試

測資 1

範例 1

輸入

AC 5

輸出

78.54

預期輸出

78.54

AC



示範演練 - 177平方根

請寫一個程式，輸入一個正整數，並輸出它的平方根。

如果輸入一個負數，則直接輸出 -1.00

輸入說明

輸入一個正整數 a ，正整數小於 10^6 。

輸出說明

輸出正整數的平方根，計算到兩位小數。

範例輸入 #1

4

範例輸出 #1

2.00

```
import math
# 輸入一個整數
a = int(input())
square_root = math.sqrt(a)
# 輸出結果，保留兩位小數
print(f"{square_root:.2f}")
```

177

錯在哪兒？

終端機	測試	
測試結果 AC	範例 1 正確	範例 2 正確

資料測試是正確，
但送出結果是錯的



自學模式 - 排行榜

排行榜

序號	姓名	等級	星星數	經驗值	玩家所在學校
1	慶00000員	1	6	157	市立鳳翔國中
2	慶00技	0	3	14	市立鳳翔國中

依星星數排序



自學模式 - 排行榜

排行榜

 個人冒險歷程	序號	姓名	角色等級	刷題成功次數	刷題成功率	經驗值	玩家所在學校
	1	陳O齊	0	19	0.3158	37	市立福山國小
	2	慶OO技	0	3	0.6667	14	市立鳳翔國中
	3	慶OOOOO員	1	2	1	157	市立鳳翔國中
	4	鍾O明	0	1	1	4	民眾

Rows per page 10 < 1 >

依刷題成功次數排序

冒險歷程 (關卡模式)

冒險歷程





冒險關卡

關卡模式

評分結算

Level 1

★ ★ ★

終端機 測試

Hello John

測資評分

測資 1 隱藏 1

測資 1

輸入 John WA: 英文或中文

輸出

預期輸出

RE

星級評價	評價說明	經驗值	距離下一個技能等級還差
	☐ 測資全部通過		
	☒ 執行時間 < 標準值	+ 0 點	100 點
	☒ 硬體負荷通過		

關卡模式

評分結算

Level 1

★ ★ ★

終端機 測試

Hello John

測資評分

測資 1 隱藏 1

測資 1

輸入 John

輸出 Hello John

預期輸出 Hello John

AC

星級評價	評價說明	經驗值	距離下一個技能等級還差
	☒ 測資全部通過		
	☒ 執行時間 < 標準值	+ 9 點	91 點
	☒ 硬體負荷通過		



關卡模式 - 第1關

使用 `print()` 印出 "Hello" 的字串。

輸入說明

(無)

輸出說明

輸出字串 Hello

範例輸入 #1

(無)

範例輸出 #1

Hello

1

終端機

測試



關卡10 - 問題說明

- 程式碼，為何系統「資料測試」通過了，但「送出結果」卻是失敗的？

先由輸入取得一系列由空白分隔的整數列，以及想在數列中搜尋的值。

將整數列分割並轉換成整數串列後，使用任何你所知道的方法將串列進行排序。

先輸出排序後的串列。再以二元搜尋法進行搜尋。

每次尋找必須輸出當次的起始值及結束值。如果找到則輸出找到索引值；如果找不到則輸出-1。

輸入說明

依序分二次輸入，第一次是一行以空格分開的整數；第二次則是一個整數，代表要在數列中搜尋的值。

輸出說明

先輸出排序後的串列。接著每次尋找都需要輸出起始值及結束值，中間以空白分隔。如果找到則輸出找到索引值；如果找不到則輸出-1。

範例輸入 #1

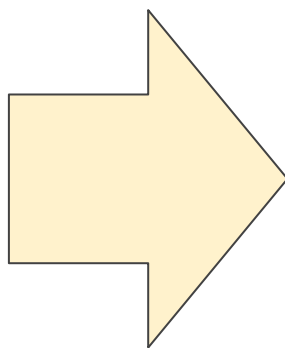
```
2 10 6 8 5
6
```

範例輸出 #1

```
[2, 5, 6, 8, 10]
0 4
2
```

關卡10 - 問題說明

- 「資料測試」通過了，但「送出結果」卻是失敗的





關卡10 - 問題處理修正

```
while start <= end:
    mid = (start + end) // 2
    if sorted_list[mid] == target:
        index = mid
        break
    elif sorted_list[mid] < target:
        start = mid + 1
    else:
        end = mid - 1
```

輸出尋找結果

```
if index != -1:
    print(start, end)
    print(index)
else:
    print(start, end)
    print(-1)
```

```
# 輸入整數列
input_list = list(map(int, input().split()))

# 輸入要搜尋的值
target = int(input())

# 排序整數列
sorted_list = sorted(input_list)

# 輸出排序後的串列
print(sorted_list)

# 進行二元搜尋
start = 0
end = len(sorted_list) - 1
index = -1

while start <= end:
    print(start, end)
    mid = (start + end) // 2
    if sorted_list[mid] == target:
        index = mid
        break
    elif sorted_list[mid] < target:
        start = mid + 1
    else:
        end = mid - 1

# 輸出尋找結果
if index != -1:
    print(index)
else:
    print(-1)
```

問題回應專區



問題回應專區

當您在使用派森試練有問題時，提供 2個管道，有專人為您服務。

- **電子郵件**：高雄市政府教育局教育網路中心 **教育企劃組**
 - kiec-edu@mail.kh.edu.tw
- 電話連絡： 07-713-6536 轉 37 趙小姐

請儘可能使用**電子郵件**連絡，因為有**截圖畫面**可以判斷是否為系統問題或是題目問題。

重要提示 - 評測系統注意事項

Q1. 輸入不可以有提示詞

- 在input()中，不可以有提示詞

```
1 n = int(input())
2 is_prime = True
```

終端機

測試

測試結果

AC

範例 1

正確



```
1 n = int(input('輸入正整數='))
2 is_prime = True
```

終端機

測試

測試結果

NA

範例 1

錯誤

範例 2

錯誤



Q2. 輸出必須完全符合要求

- 題目要求輸出YES及NO

輸出說明

如果 n 為質數，則輸出 'YES'，否則輸出 'NO'。

```
10 ✓ if is_prime:
11     print('YES')
12 ✓ else:
13     print('NO')
14
```

終端機 測試

測試結果

AC

範例 1

正確

範例 2

正確

```
10 ✓ if is_prime:
11     print('Yes')
12 ✓ else:
13     print('No')
14
```

終端機 測試

測試結果

NA

範例 1

錯誤

範例 2

錯誤

Q3. 如何一行輸入多個數 值

```
a, b = map(int, input().split())
```

函式說明

- `split()`：一個字串的方法，將字串分割成多個部分，預設使用**空格**作為分割字元。
- `map()`：一個內建函數，將指定的函數應用於可迭代物件。
 - 示範中指定`int`函數作為要應用的函數，將輸入字串列表中的每個元素轉換為整數型別。

上述這個程式輸入 24 30 時，`map`會把它們轉換成`a=24`，`b=30`來輸入。

Q4. 只要能達到輸出入要求，程式可有不同解法

輸入一個正整數 r

- 以 r 作為圓的半徑，
- 計算並輸出圓的面積（計算至小數點後兩位。）
- 圓周率請使用 `math` 模組的 `math.pi`
- 圓面積公式：

$$\text{圓面積} = \pi r^2$$

輸入說明

輸入一個正整數，代表圓的半徑，範圍在 1~1000

輸出說明

輸出圓的面積，計算至小數點後兩位

```
1 r = int( input())
2 a = round(3.14159*r*r ,2)
3 print(a)
```

終端機

測試

測試結果

AC

範例 1

正確

```
1 import math
2 r = int( input())
3 a = round(math.pi*r*r,2)
4 print(a)
```

終端機

測試

測試結果

AC

範例 1

正確



Q5. 資訊測試通過，但送出結果失敗

請寫一個程式，輸入一個正整數，並輸出它的平方根。

如果輸入一個負數，則直接輸出 -1.00

輸入說明

輸入一個正整數 a ，正整數小於 10^6 。

輸出說明

輸出正整數的平方根，計算到兩位小數。

範例輸入 #1

4

範例輸出 #1

2.00

```
import math
# 輸入一個整數
a = int(input())
square_root = math.sqrt(a)
# 輸出結果，保留兩位小數
print(f"{square_root:.2f}")
```

程式剛好符合測資，但是在隱藏測資中不符合，沒有寫到負數

錯在哪兒？

終端機	測試	
測試結果	範例 1	範例 2
AC	正確	正確

測資評分		測資 1	測資 2	隱藏 1
RE	輸入	4		
	輸出	2.00		
	預期輸出	AC 2.00		

Python

快速入門篇



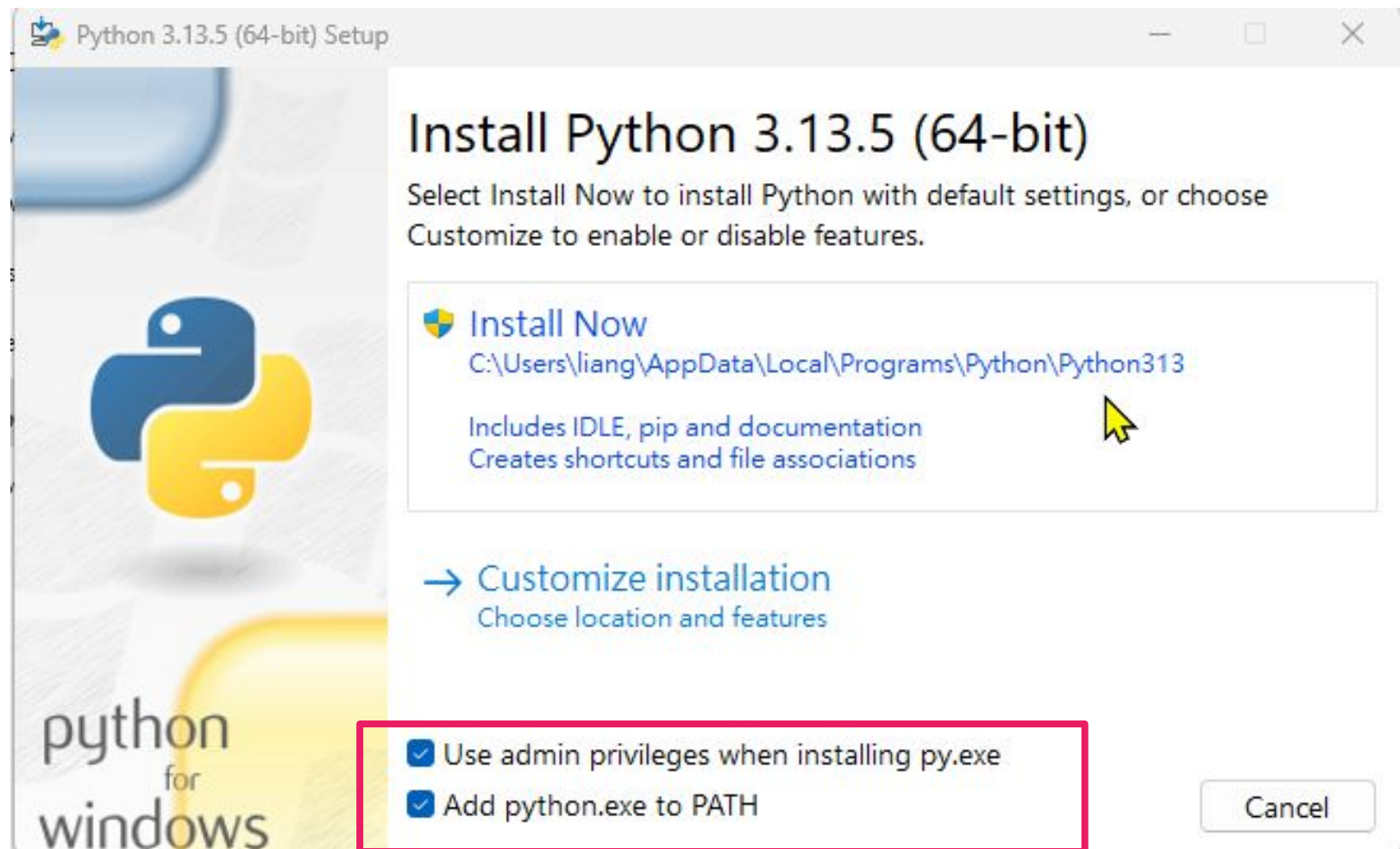
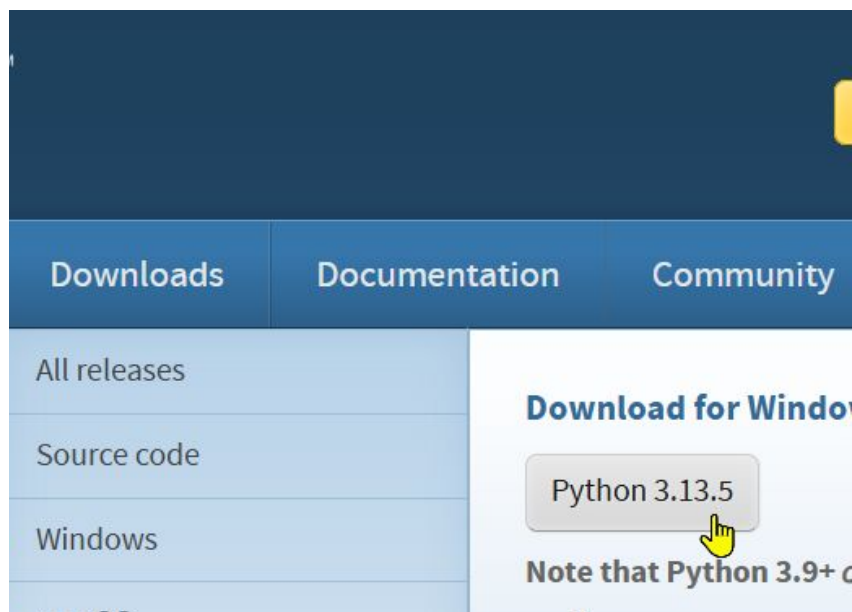


研習資料

- Python官方網站: <https://www.python.org/>
- Gemini : <https://gemini.google.com>
- ChatGPT: <https://chatgpt.com/>
- OneCompiler: <https://onecompiler.com/python>

安裝與設定

安裝





撰寫第一個程式

使用IDLE撰寫程式

- 點選 [開始/ Python 3.9/ IDLE]
- 設定字體「Option/Configure IDLE」
 - Fonts頁籤中，設「Consolas/25點」或 Coutier New
 - Shell/Ed頁籤中，設定行號
- 點選功能表 [File/ New File]
- 在編輯器中輸入程式碼後，按[Ctrl連結文字+S]鍵儲存
- 在編輯器中，點選功能表 [File/ Run]執行

哈囉



哈囉

- 請設計一個程式，讓使用者輸入名字後，電腦會將名字呈現在畫面上與使用者打招呼。

文字輸入

1 2 3
變數=input(["提示字串"])

- ① 變數：在 python 中，變數不用宣告，只要直接設定變數名稱並指定變數的值，就代表建立變數。

例 執行完右圖指令後，就會建立變數「a」，其值為 1。



※ 變數名稱可由大小寫英文字母、數字及底線 `_` 組成，且第一個字必須是字母。

- ② 等號 `=`：等號是「指派」的意思，也就是將等號右邊的值儲存至左邊的變數中。
- ③ 提示字串：此處的內容會呈現在畫面上，用以提示使用者要回答什麼。

※ 放在 `[]` 當中，代表是擴充內容，可以省略，但使用者可能不知道程式停下來要做什麼。

文字輸出



```
print(項目1[,項目2,...])
```

※ 語法中，所有符號均為半形符號。

- ① 方法名稱：有嚴格的大小寫限制，不能寫成 Print 喔！
- ② 小括號 ()：將指令內容包含其中。
- ③ 項目：要印出的內容，若是變數、數字，可直接填入；
若是文字或字串，則必須以單引號 ' ' 或雙引號 " " 包起來。
- ④ 中括號 []：中括號裡的指令為可擴充內容。※ 中括號不是語法一部分，所以不用寫出來。
例 假設只要輸出一個項目，可省略 [] 內容，指令寫成 `print(項目 1)`；若要輸出兩個項目，則指令寫成 `print(項目 1, 項目 2)`。
- ⑤ 逗號 ,：語法中有多個項目或參數時，各項目間以逗號分隔。

求平均數



求平均數

- 請設計一個程式，讓使用者輸入兩個數字後，再呈現兩個數字的總和與平均值。

資料型態

- ① 在 python 中，各種型態占用的記憶體空間都不同，在建立變數時若沒有特別指定，程式就會根據變數值選擇最適合的型態來儲存資料，以減少空間浪費。常見的資料型態有：

資料型態	說明	範例
int	整數：包含正整數、負整數、0	age=10
float	浮點數：包含小數點的數	weight=129.3
str	字串：即一串文字	name="哆啦 A 夢"
bool	布林值：只能是 True 或 False	isRobot=True

註 當我們想得知某筆資料是什麼型態時，可以使用 `type()` 來查看，例如：

程式碼	執行結果
<pre>name="哆啦 A 夢" # 將變數 name 賦值「哆啦 A 夢」 type(name) # 查看變數 name 的資料型態</pre>	str

- ② 數值資料（int、float）與字串的型態不同，無法相加。**例** `3+"3"` → 無法執行。
- ③ 整數和浮點數雖為不同型態，但皆為數值資料，程式會自動將整數轉換成浮點數來相加。
- ④ 使用 `input()` 所取得的數字是字串型態。**例** 使用者輸入 10，實際上回傳的會是 "10"。

資料型態轉換

資料型態的轉換方法很好記，因為方法名稱與要轉換成的資料型態相同。例如要將數字轉換成文字，就是 `str()`。

方法	說明	範例程式、輸出結果
<code>int()</code>	1. 將浮點數轉換成整數，小數點會被捨棄。 2. 將字串且值為整數的資料轉換成整數。	<code>int(2.5) → 2</code> <code>int("2") → 2</code> <code>int("2.0") → 無法轉換</code>
<code>float()</code>	1. 將整數轉換成浮點數。 2. 將字串且值為數值的資料轉換成浮點數。	<code>float(5) → 5.0</code> <code>float("5") → 5.0</code> <code>float("5.5") → 5.5</code>
<code>str()</code>	將任何數字、文字轉換成文字。	<code>str(20) → "20"</code>
<code>bool()</code>	除了「空字串、0、無輸入」會被轉換為 False，其餘所有文字、數字都會被轉換為 True。	<code>print(bool("")) → False</code> <code>print(bool("False")) → True</code>

算術運算子

- ① 算術運算子：主要用於數學計算，但有些符號與數學上常用的寫法不同。
- ② 數學運算順序：
 - ❶ 括號內先運算
 - ❷ 展開指數
 - ❸ 乘除法
 - ❹ 除法取餘數或商數
 - ❺ 加減法
- ③ 「+」可以運用在「數值相加」或「文字組合」上，但數值與文字無法以「+」組合。

運算子	說明	範例	結果
+	相加	15+2	17
-	相減	15-2	13
*	相乘	15*2	30
/	相除	15/2	7.5
%	除法取餘數	15%2	1
//	除法取商數的整數	15//2	7
**	指數	3**2	9

→ 即 3^2

單行輸入多變數

- 範例：輸入為三個整數用空格隔開，請輸出三數平均數。

```
a, b, c = input().split()
```

split 括號內放置分割子
預設為空白鍵

```
a = int(a)
```

```
b = int(b)
```

input 預設為字串 (str)
利用 **int** 轉換型態

```
c = int(c)
```

```
print((a + b + c) / 3)
```

前四行可合併為：

```
a, b, c = map(int, input().split())
```

計算學期成績



計算學期成績

- 請設計一個程式，讓使用者輸入各項成績後，再將各項成績轉換為學期成績，並判斷學期成績是否及格？（其中，作業成績占 40%，測驗成績占 40%，平時成績占20%，學期成績 60分為及格分數。）



進階題 - 計算學期成績

- 請設計一個程式，讓使用者輸入各項成績後，再將各項成績轉換為學期成績，並判斷學期成績等級？
 - 90分以上優等
 - 80分以上甲等
 - 70分以上乙等
 - 60分以上丙等
 - 60分以下要補考



關係運算子

關係運算子會比較左右兩側的值，成立回傳 True、反之回傳 False。

運算子	說明	範例①	回傳結果	範例②	回傳結果
==	兩者相等	5==5	True	5==6	False
【註】一個等號「=」在程式中是「指派」的意思，請見 P.99 說明。					
!=	不相等	5!=5	False	5!=6	True
>	大於	5>5	False	5>6	False
<	小於	5<5	False	5<6	True
>=	大於或等於	5>=5	True	5>=6	False
<=	小於或等於	5<=5	True	5<=6	True

【註】當有兩組以上的條件要「同時」進行判斷時，就必須使用「邏輯運算子」，包含 and（且）、or（或）、not（非），以進行多重判斷，使用時建議加上括號以利閱讀。

範例	回傳結果
(5==5) or (5==6)	True
not ((5==5) or (5==6))	False

選擇結構 – 單向



① `if` 指令：意思是「如果」，一組選擇結構只會有一個 `if`。

② 條件式：可以是判斷式、變數、布林值等。

❗ 建議使用括號將條件式包起來，以清楚呈現條件式的範圍。

③ 半形冒號 `:`：用於表示 `if` 指令的條件式敘述結束。

④ 若條件式成立（True），執行程式區塊 1，若不成立（False），則不會執行。

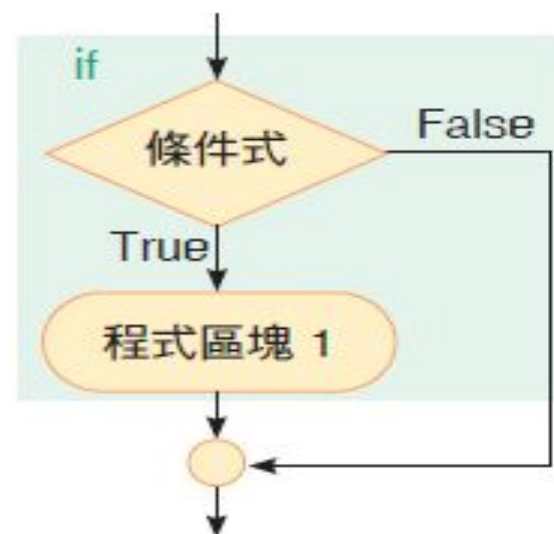
⑤ 縮排：

❶ 為了呈現程式的內外層結構，python 會使用「縮排」來表示。

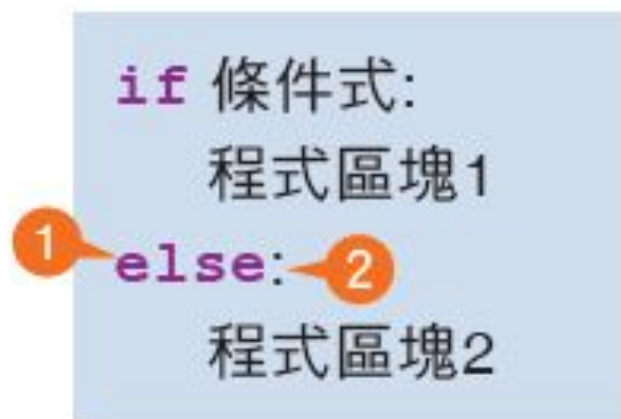
例 `if` 指令的下一行是選擇結構內層，換行時會自動縮排。

❷ 縮排的標準格式為 4 個 `半形空格`（或 1 個 `Tab 鍵`）。

（使用 2 ~ 6 個 `半形空格`，也可達成縮排效果，但同一段落的程式中，必須使用相同的空格數進行縮排，程式才能正常執行）



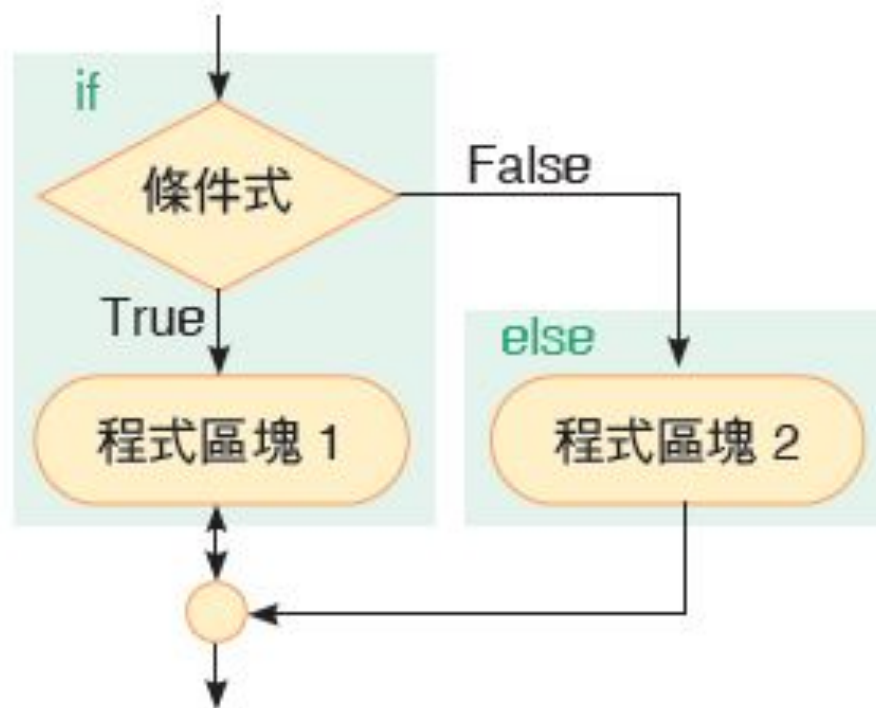
選擇結構 – 雙向



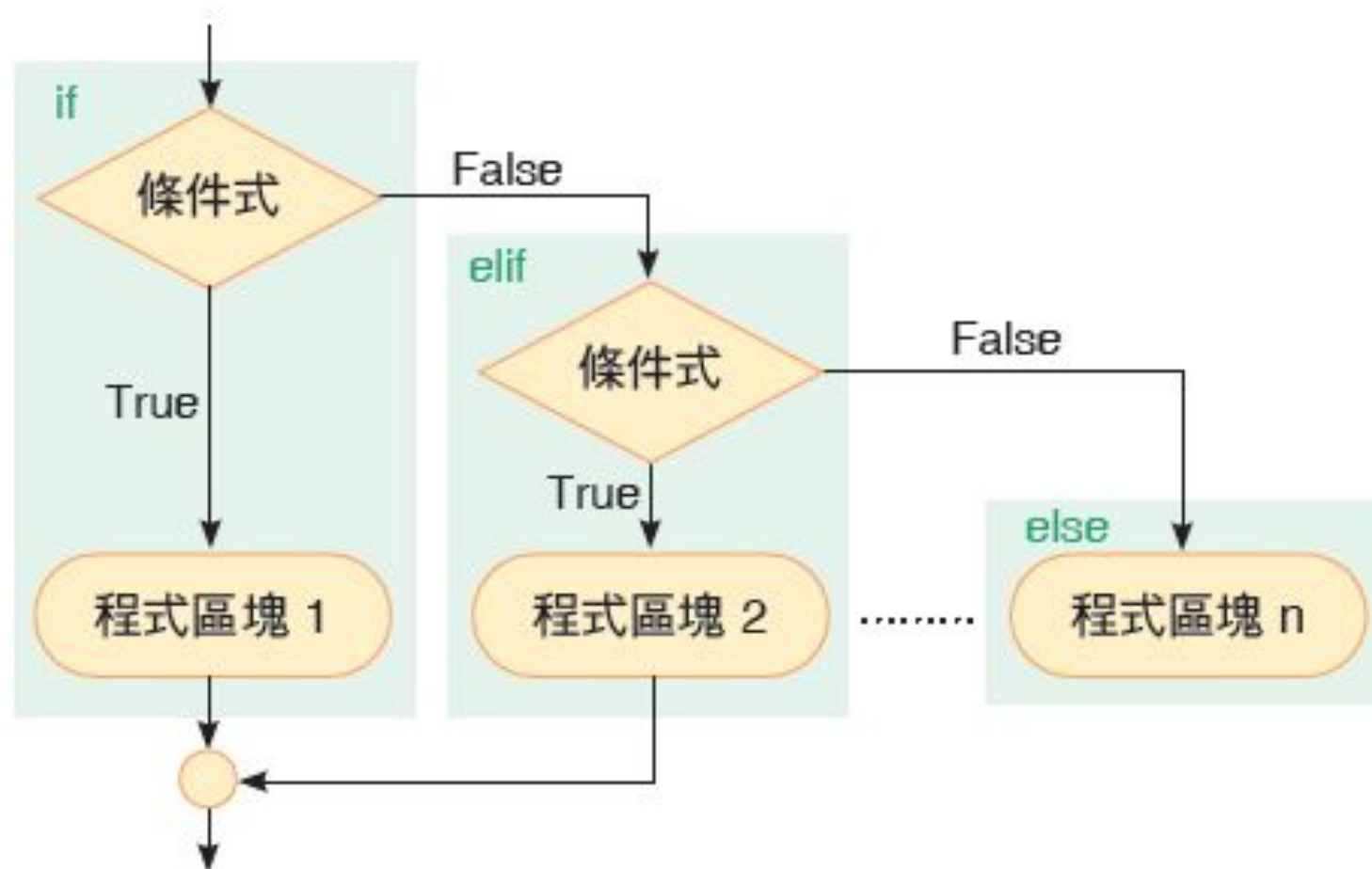
① `else` 指令：意思是「其他」，也就是當 `if` 條件不成立的其他狀況，都會執行 `else` 指令下的程式區塊 2。

❗ 一組選擇結構只能有一個 `else` 指令。

② `else` 指令本身不接條件式，直接以半形冒號結尾。



選擇結構 – 多向





練習題 - 計算BMI值

- 輸入體重及身高後，計算BMI值
- 判斷標準
 - BMI ≤ 18.5 時，體重太輕了
 - BMI > 18.5 且 BMI ≤ 24 時，體重很標準
 - BMI > 24 時，體重太重了

BMI身體質量指數

$$\text{BMI} = \frac{\text{體重(kg)}}{\text{身高(m)}^2}$$

累加計算



累加計算

- 請設計一個程式，讓使用者輸入數字 n 後，再計算 $1 + 2 + 3 + \dots + n$ 的值。

for迴圈

① **for** 變數 **in** range(重複次數):
程式區塊

②

! for 迴圈搭配 range() 使用時，功能與 Scratch 中的重複幾次積木類似。

- ① 「for 變數 in ...」為固定用法，常寫作「for i in ...」或「for n in ...」。
- ② 在 range() 中輸入的整數，就是程式區塊會重複執行的次數。

【程式範例】

程式碼	結果
<pre>thing=input("請輸入重要的事情：") for i in range(3): # 重複 3 次下一行程式 print(thing)</pre>	回答為「吃飯」 吃飯 吃飯 吃飯

串列

串列

串列的資料結構

程式的資料用變數來儲存，但大量的資料可以用串列來簡化

串列的語法

串列名稱 = [元素1, 元素2, ...]

串列的索引值由0開始,若是索引為負數-1則是倒數

```
for i in [1,2,3,4,5]:
```

```
    print(i)
```

```
for i in range(1,6):
```

```
    print(i)
```



進階題 - 任意數的所有因數

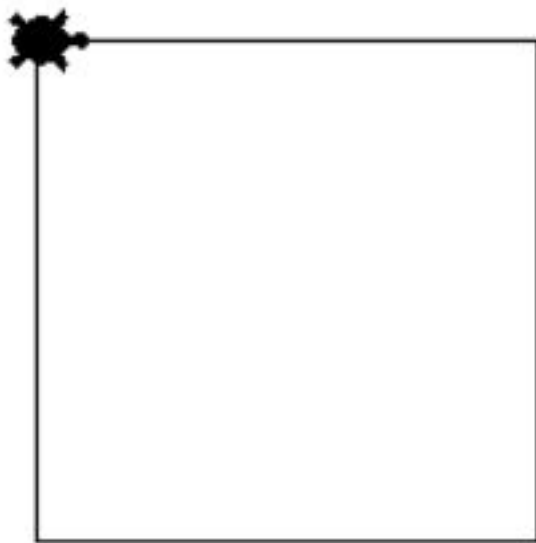
- 請設計一個程式，讓使用者輸入一個數字後，再找出該數字的所有因數。

4.正多邊形小畫家



畫正方形

- 請設計一個程式，讓使用者畫出一個邊長為 150 個單位的正方形。



繪圖模組——海龜

1. 前置步驟：Colab 並未內建海龜模組，因此我們必須①安裝套件、②匯入模組，才能正常執行，這兩個動作在每支使用到海龜的程式中都必須執行一次。

① 安裝套件：指令為 `!pip3 install 套件名`。

```
!pip3 install ColabTurtle
```

② 匯入模組：指令為 `import 模組 as 變數`。

```
import ColabTurtle.Turtle as t
```

當模組名稱較長，常搭配「as」指令將其設定為一個變數，方便呼叫使用。

➡ 上述指令將匯入 ColabTurtle 套件裡的 Turtle 模組（寫作 Colabturtle.Turtle），並以變數「t」來代表該模組。

! Python 有許多模組，包括繪圖、資料處理等，有些是內建好的，可直接以 import 指令匯入進行使用；有些則是由第三方開發，這時就必須先安裝再匯入才能使用。

繪圖模組——海龜

2. 使用模組：指令為 `模組名稱 (或變數). 方法 ()`。

匯入海龜模組後，就可以利用模組中的方法來進行圖形繪製。

例 `t.forward(90)` 代表前進 90 點，也可寫作 `ColabTurtle.Turtle.forward(90)`。

ColabTurtle 常用方法	說明
<code>initializeTurtle()</code>	初始化畫布，如清除筆跡、重設繪圖點等。
<code>forward()</code> 、 <code>backward()</code>	繪圖點向前、後移動指定距離。
<code>left()</code> 、 <code>right()</code>	繪圖點向左、右轉動指定角度。預設為向上。
<code>pendown()</code>	放下畫筆，開始繪製。預設為放下畫筆。 ※ 下筆狀態時，移動的路徑都會畫出線條。
<code>penup()</code>	提起畫筆，不繪製。
<code>position()</code>	回傳目前繪圖點坐標。預設起始點為畫布正中間。
<code>goto()</code>	前往指定坐標。ColabTurtle 中畫布坐標沒有負數， 畫布最左上角為 (0,0)，向右、向下增加。

將程式打包成執行檔



安裝PyInstaller

- 在電腦的命令提示視窗(CMD)中, 安裝PyInstaller
 - `pip install pyinstaller`
- 目錄執行檔
 - `pyinstaller 01.py`
- 單一執行檔
 - `pyinstaller -F 01.py`

課程結束



補充資料 - 測資評分說明

- **AC** : Accept 即表示通過，測資全部通過
- **NA** : Not Accept 表示在多個測資點的情況下，有部分測資點無法通過
- **WA** : Wrong Answer 表示答案錯誤，請仔細比對，務必符合題目要求
- **TLE** : Time Limit Exceed 表示【執行超過時間限制】
- **MLE** : Memory Limit Exceed 表示程序超過記憶體限制
- **OLE** : Output Limit Exceed 表示程序輸出超過限制
- **RE** : Runtime Error 執行時錯誤，通常為記憶體配置錯誤 如使用超過陣列大小的位置
- **RF** : Restricted Function 表示使用了被禁止使用的函式，並在錯誤訊息中指明使用了什麼不合法的函式。（取消）
- **CE** : Compile Error 表示編譯錯誤
SE: System Error 包含 Compile, Runtime 等未定義的錯誤均屬於 System Error（保留）

補充資料 - 星級評價說明



補充資料 - 星級評價說明 一顆星

- **【達成】** 下列所有目標
 - NA: Not Accept 表示在多個測資點的情況下，有部分測資點無法通過
 - 執行時間 < 標準值 // 待封閉測試取平均過關標準時間後訂定之
 - CPU使用率介於 70-90%【或】記憶體使用率介於 70-90%
- **【沒有】** 下列任一狀態
 - WA: Wrong Answer 表示答案錯誤，請仔細比對，務必符合題目要求
 - TLE: Time Limit Exceed 表示【執行超過時間限制】
 - MLE: Memory Limit Exceed 表示程序超過記憶體限制
 - OLE: Output Limit Exceed 表示程序輸出超過限制
 - RE: Runtime Error 表示執行時錯誤，通常為記憶體配置錯誤 如：使用了超過陣列大小的位置
 - RF: Restricted Function 表示使用了被禁止使用的函式，並在錯誤訊息中指明使用了什麼不合法的函式。
 - CE: Compile Error 表示編譯錯誤SE: System Error 包含 Compile, Runtime 等未定義的錯誤均屬於 System Error
 - 無使用「輔助模式」進行遊戲



補充資料 - 星級評價說明 二顆星

- **【達成】** 下列所有目標
 - NA: Not Accept 表示在多個測資點的情況下，有部分測資點無法通過
 - 執行時間 < 標準值 // 待封閉測試取平均過關標準時間後訂定之
 - CPU使用率介於 50-69%【或】記憶體使用率介於 50-69%
- **【沒有】** 下列任一狀態
 - WA: Wrong Answer 表示答案錯誤，請仔細比對，務必符合題目要求
 - TLE: Time Limit Exceed 表示【執行超過時間限制】
 - MLE: Memory Limit Exceed 表示程序超過記憶體限制
 - OLE: Output Limit Exceed 表示程序輸出超過限制
 - RE: Runtime Error 表示執行時錯誤，通常為記憶體配置錯誤 如：使用了超過陣列大小的位置
 - RF: Restricted Function 表示使用了被禁止使用的函式，並在錯誤訊息中指明使用了什麼不合法的函式。
 - CE: Compile Error 表示編譯錯誤SE: System Error 包含 Compile, Runtime 等未定義的錯誤均屬於 System Error
 - 無使用「輔助模式」進行遊戲



補充資料 - 星級評價說明 三顆星

- **【達成】** 下列所有目標
 - AC Accept 即表示通過，測資全部通過
 - 執行時間 < 標準值 // 待封閉測試取平均過關標準時間後訂定之
 - CPU使用率小於50% **【且】** 記憶體使用率小於50%
- **【沒有】** 下列任一狀態
 - NA: Not Accept 表示在多個測資點的情況下，有部分測資點無法通過
 - WA: Wrong Answer 表示答案錯誤，請仔細比對，務必符合題目要求
 - TLE: Time Limit Exceed 表示**【執行超過時間限制】**
 - MLE: Memory Limit Exceed 表示程序超過記憶體限制
 - OLE: Output Limit Exceed 表示程序輸出超過限制
 - RE: Runtime Error 表示執行時錯誤，通常為記憶體配置錯誤 如：使用了超過陣列大小的位置
 - RF: Restricted Function 表示使用了被禁止使用的函式，並在錯誤訊息中指明使用了什麼不合法的函式。(取消)
 - CE: Compile Error 表示編譯錯誤SE: System Error 包含 Compile, Runtime 等未定義的錯誤均屬於 System Error (保留)
 - 無使用「輔助模式」進行遊戲



補充資料 - 星級評價說明 未過關

- **【達成】** 下列任一目標
 - WA: Wrong Answer 表示答案錯誤，請仔細比對，務必符合題目要求
 - TLE: Time Limit Exceed 表示【執行超過時間限制】
 - MLE: Memory Limit Exceed 表示程序超過記憶體限制
 - OLE: Output Limit Exceed 表示程序輸出超過限制
 - RE: Runtime Error 表示執行時錯誤，通常為記憶體配置錯誤 如：使用了超過陣列大小的位置
 - RF: Restricted Function 表示使用了被禁止使用的函式，並在錯誤訊息中指明使用了什麼不合法的函式。
 - CE: Compile Error 表示編譯錯誤SE: System Error 包含 Compile, Runtime 等未定義的錯誤均屬於 System Error
 - 執行時間 > 標準值 // 待封閉測試取平均過關標準時間後訂定之
 - 使用「輔助模式」進行遊戲者
- **【沒有】** 下列任一狀態
 - AC Accept 即表示通過，測資全部通過
 - NA: Not Accept 表示在多個測資點的情況下，有部分測資點無法通過