關於本書

在資訊科技深入到各行各業、AI 快速起飛的現代，程式設計已經不再是資訊專業人員的限定技能，有更多人想要學習程式設計，例如行銷人員可以撰寫程式統計商品銷售數字、辦公室人員可以撰寫程式自動化處理重複性工作、金融人員可以撰寫程式分析股價變動趨勢。

在眾多程式語言當中，**又以 Python 爲初學者的首選，因爲其語法簡潔、易學易用、功能強大，適合各種應用場景**。雖然如此，還是有不少人半途而廢，因為許多書籍往往著重在鉅細靡遺的語法教學，忽略了初學者無法消化這麼多，最終的結果就是受挫放棄。

隨著 **ChatGPT**、**Google Gemini**、**Microsoft Copilot** 等 AI 助理的出現，程式設計這項技能開始有了轉變，**傳統的學習方式是以語法爲導向，而 AI 時代的學習方式是以解決問題爲核心**，當人們面對問題時，首先要思考的是如何解決問題，接著才是撰寫程式碼，然後測試程式碼。

因此，**程式設計除了要懂得語法，還要培養解決問題的邏輯思維**，才能有效率地跟 AI 助理溝通，生成符合需求的程式碼，然後進一步閱讀並測試程式碼，確保程式碼是經過完整測試、正確無誤且有效率的。

在本書中，我們會在每章的開頭講解重要的語法與主題，然後在結尾的地方透過「**ChatGPT 程式助理**」專欄，示範如何有效率地和 AI 助理合作，讓初學者不再苦苦糾纏於繁瑣的語法，能夠快速寫出正確、有用的程式！當然這些提示工程技巧並不限定於 ChatGPT，你也可以舉一反三、靈活運用在 Gemini、Copilot 等 AI 助理。

本書內容

首先，在第 1 章中，我們會介紹如何使用 Anaconda 和 Colab 撰寫 Python 程式，其中 **Anaconda** 是安裝在本機電腦的開發環境，功能齊全，適合開發者；**Colab** 是在雲端運行的開發環境，透過瀏覽器就能運行，適合初學者。

接著，在第 2 ~ 9 章中，我們會以範例為導向，講解 Python 的語法，包括變數、型別與運算子、數值與字串處理、容器型別、流程控制、函式、模組與套件、檔案存取、例外處理、類別與物件等。

最後，在第 10 ~ 13 章中，我們會介紹幾個實用的 Python 套件，包括：

- **pillow**：圖像處理，例如轉換色彩模式、調整大小、裁剪圖片、旋轉與翻轉、濾鏡、繪製文字、繪製圖形等。
- **matplotlib**：繪製圖表，例如折線圖、散布圖、長條圖、直方圖、圓餅圖等。
- **tkinter**：建立圖形使用者介面，例如視窗、標籤、按鈕、輸入方塊、對話方塊、核取按鈕、選項按鈕、功能表、圖形等。
- **Requests**、**Beautiful Soup**：網路爬蟲，例如抓取臺灣銀行牌告匯率資料、從「yahoo! 股市」抓取即時股價等。

「**ChatGPT 程式助理**」專欄則有和 AI 助理合作解決問題、撰寫程式的技巧，例如查詢語法和範例；撰寫、修正與優化程式；閱讀並測試程式碼；除錯；幫程式加上註解或 try...except 語法；解決流程錯誤與無窮迴圈；撰寫邏輯複雜的程式；透過設計與撰寫函式來解決問題；查看與解決程式錯誤所造成的例外；根據資料判斷要使用哪種圖表並撰寫程式；根據附圖與文字敘述撰寫 GUI 程式；解決網路爬蟲程式失敗等。

聯絡方式

碁峰資訊網站 https://www.gotop.com.tw/；國內學校業務處電話－台北 (02)2788-2408、台中 (04)2452-7051、高雄 (07)384-7699。

4-1 list (串列)

在本章中，我們要介紹一些有別於數值與字串的**容器型別** (container type)，包括 **list** (串列)、**tuple** (元組)、**set** (集合) 與 **dict** (字典)，其比較歸納如下。之所以稱為「容器型別」，主要是因為它們就像容器，能夠包含多個資料，當程式要處理大量資料時，就可以使用容器型別。

容器型別	list (串列)	tuple (元組)	set (集合)	dict (字典)
前後符號	[]	()	{}	{}
有無順序	有	有	無	無
可否改變內容	可以	不可以	可以	可以

4-1-1 建立 list

list (串列) 可以包含多個**有順序、可改變內容**的資料，稱為**元素**。list 的前後以**中括號**標示，元素以**逗號**分隔，型別不一定要相同。元素會按順序排列，可以透過**索引** (從 0 開始) 來存取元素，也可以新增、刪除或修改元素，適合用來處理生活中的資料，例如考試成績、學生名單、購物清單、城市名稱等。我們可以使用**中括號 []** 建立 list，例如下面的敘述是建立一個 list 並指派給變數 A：

```
❶   ❷
A = [10, '倫敦', 20]
        ❸
```

元素	值
A[0]	10
A[1]	'倫敦'
A[2]	20

❶ list 的名稱。

❷ list 的前後以中括號標示。

❸ 包含 10、'倫敦'、20 三個元素，中間以逗號分隔。我們可以透過 list 的名稱與索引來存取元素，例如 A[0]、A[1]、A[2] 代表 10、'倫敦'、20。

list 亦可包含 list，例如下面的敘述是建立一個巢狀串列並指派給變數 B：

```
B = [10, [21, 22], 30]
```

❶ 巢狀串列的名稱。

❷ 第二個元素是另一個 list，我們可以透過 list 的名稱與兩個索引來存取元素，例如 B[1][0]、B[1][1] 代表 21、22。

元素	值
B[0]	10
B[1]	[21, 22]
B[1][0]	21
B[1][1]	22
B[2]	30

NOTE

★ 串列的資料是有順序的，所以像 [1, 2, 3] 和 [3, 2, 1] 雖然包含相同的元素，卻是不同的串列，因爲元素的順序不同。

★ **空串列**指的是沒有包含元素的串列，也就是 **[]**。

★ 我們也可以使用內建函式 **list()** 建立串列，例如：

```
In [1]: list('ABC')          # 從字串建立包含 'A', 'B', 'C' 的串列
Out[1]: ['A', 'B', 'C']
In [2]: list({10, 20, 30})  # 從集合建立包含 10, 20, 30 的串列
Out[2]: [10, 20, 30]
```

★ **str.split()** 方法可以用來將字串分隔成串列，例如：

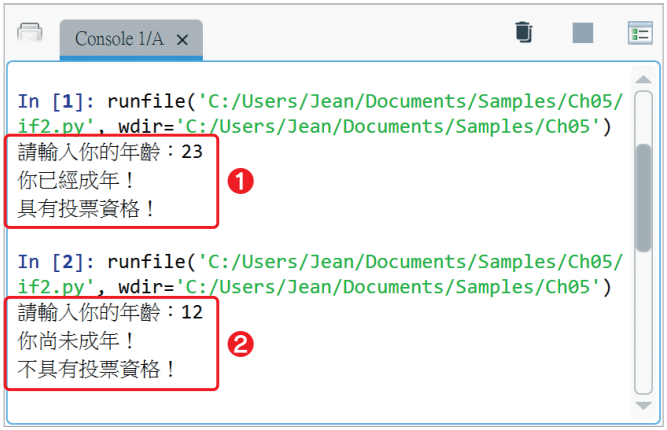
```
In [1]: 'x y z'.split()      # 根據空白將字串分隔成串列
Out[1]: ['x', 'y', 'z']
In [2]: 'x,y,z'.split(',')  # 根據逗號將字串分隔成串列
Out[2]: ['x', 'y', 'z']
```

前一節的例子有個缺點，若輸入的年齡不到 20 歲，程式就不會顯示任何訊息，而這難免讓人感到疑惑，此時，可以使用 if...else 來改寫，如下，注意 if 區塊和 else 區塊裡面的敘述要縮排，同時縮排要對齊。

\Ch05\if2.py

```
01  age = int(input('請輸入你的年齡：'))
02
03  if age >= 20:
04      print('你已經成年！')
05      print('具有投票資格！')
06  else:
07      print('你尚未成年！')
08      print('不具有投票資格！')
```

執行結果如下，若第 01 行輸入的年齡大於等於 20，例如 23，第 03 行的條件式 age >= 20 的結果為 True，就執行 if 區塊裡面的敘述，顯示第 04、05 行的訊息；相反的，若第 01 行輸入的年齡小於 20，例如 12，第 03 行的條件式 age >= 20 的結果為 False，就執行 else 區塊裡面的敘述，顯示第 07、08 行的訊息。



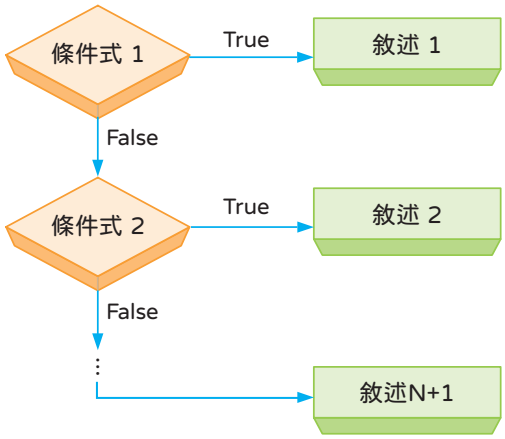
❶ 輸入 23 會顯示此訊息　❷ 輸入 12 會顯示此訊息

5-2-3 if...elif...else (若...就...否則 若...)

if...elif...else 的語法如下，一開始先檢查「條件式 1」，若結果為 True，就執行「敘述 1」，否則檢查「條件式 2」，若結果為 True，就執行「敘述 2」，…，依此類推，若所有條件式的結果均為 False，就執行「敘述 N+1」，所以「敘述 1」~「敘述 N+1」只有一組會被執行。

```
❶ if 條件式 1:
        敘述 1
❷ elif 條件式 2:
        敘述 2
   ...
❸ else:
        敘述 N+1
```

❶ 條件式 1 爲 True 就執行敘述 1
❷ 條件式 2 爲 True 就執行敘述 2
❸ 所有條件式均爲 False 就執行敘述 N+1



撰寫邏輯複雜的程式（計算綜所稅）

在學會流程控制後，我們可以試著請 ChatGPT 撰寫功能更強大、邏輯更複雜的程式，有時光靠自己想並不是很周全，可能會有邏輯錯誤或想到腦袋打結太燒腦，此時，ChatGPT 就可以派上用場。

舉例來說，假設要計算個人綜所稅的應納稅額，於是到國稅局網站查詢最新的累進稅率，找到如下圖表 tax_rate.jpg：

級距	綜合所得淨額	稅率
1	0~590,000	5%
2	590,001~1,330,000	12%
3	1,330,001~2,660,000	20%
4	2,660,001~4,980,000	30%
5	4,980,001以上	40%

我們可以把這個圖表上傳給 ChatGPT，並輸入提示詞「**已知個人綜合所得稅的稅率級距如附圖，請撰寫一個 Python 程式，令它要求輸入綜合所得淨額，然後根據稅率級距計算應納稅額。**」，如下，若你無法上傳圖表（或許是因為免費額度用完了），可以改用文字來說明稅率級距。

得到如下回答，裡面有程式與說明，我們將程式複製到 Spyder 做測試，並存檔為 \Ch05\tax.py 供你參考。很幸運的，經過測試後，確認程式 OK。

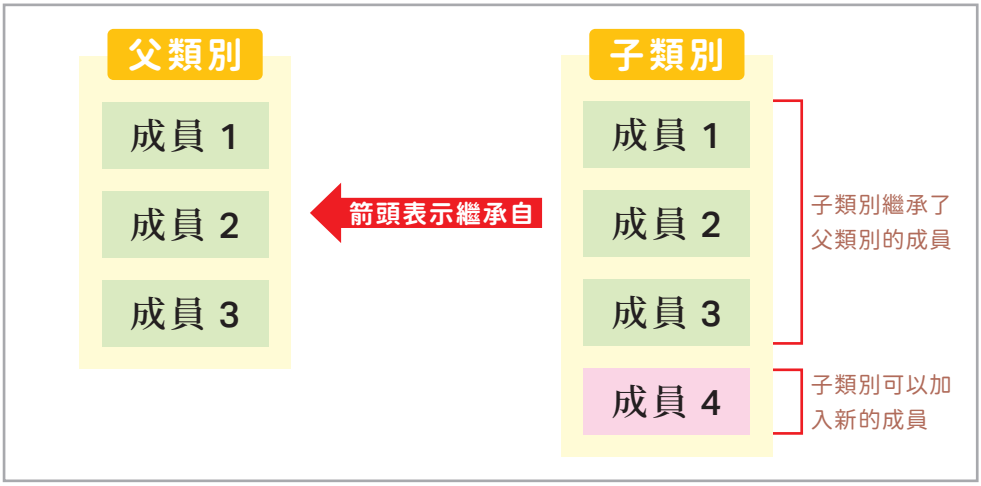
再度提醒，ChatGPT 生成的答案是隨機的，你所得到的回答不見得跟書上的相同，也不一定完全正確。**請務必對 AI 生成的程式碼保持審慎的態度，徹底了解與測試每一行程式碼**，切勿驚艷於 AI 強大的生成能力，就全盤接受，忽略了其中可能隱藏的錯誤或漏洞。也正因為需要替程式碼把關，所以你還是得學好 Python，才能看得懂程式並與 ChatGPT 有效的互動。

9-3 繼承

繼承 (inheritance) 是物件導向程式設計的主要特性之一，Python 允許一個類別繼承另一個類別的成員，也就是從現有的類別定義出新的類別，而不必重新撰寫相同的程式碼，有助於提升程式的重複使用性及擴展性。

我們將這個現有的類別稱為**父類別** (parent class) 或**基底類別** (base class)，而新的類別稱為**子類別** (child class) 或**衍生類別** (derived class)。子類別繼承了父類別的非私有成員，同時可以加入新的成員，或**覆蓋** (override) 繼承自父類別的方法，也就是在子類別內重新定義繼承自父類別的某個方法。

舉例來說，假設要定義兩個類別用來表示「火車」和「公車」，由於它們具有「交通工具」共同的特質與動作，為了不要重複定義，我們可以先定義具有一般性的 Transport 類別作為父類別，裡面有行車速度、動力來源、乘車人數等屬性，以及發動、停止等方法；接著可以繼承此類別，定義具有特殊性的 Train 和 Bus 兩個子類別，然後在子類別內加入獨有的特質與動作，例如火車的車廂數目、公車是否為雙層巴士等，同時可以在子類別內覆蓋發動、停止等方法，因為火車和公車的發動與停止方式不同。



9-3-1 定義子類別

我們一樣是使用 **class** 關鍵字定義子類別，**差別在於子類別名稱後面要加上小括號和父類別名稱**，若父類別不只一個，中間以逗號分隔，其語法如下，子類別內的敘述可以是屬性、方法、建構子、解構子等成員：

```
class 子類別 ( 父類別 ):
    敘述

class 子類別 ( 父類別 1, 父類別 2, …):
    敘述
```

下面是一個例子，其中類別 B 是類別 A 的子類別，繼承了類別 A 的非私有成員，包括 x 屬性和 M1() 方法，同時加入新的成員 y 屬性。

\Ch09\inheritance1.py

```
01 class A:
02     x = ' 我是類別 A 定義的 x 屬性 '
03
04     def M1(self):
05         print(' 我是類別 A 定義的 M1() 方法 ')
06
07 class B(A):
08     y = ' 我是類別 B 定義的 y 屬性 '
09
10 obj = B()
11 print(obj.x)
12 print(obj.y)
13 obj.M1()
```

類別 A

類別 B 是類別 A 的子類別

透過類別 B 的物件存取 x、y、M1() 等成員

10-5 旋轉與翻轉圖片

我們可以使用圖片物件的 **rotate()** 方法根據參數所指定的角度來旋轉圖片，例如 20 表示逆時針旋轉 20 度，-20 表示順時針旋轉 20 度。旋轉後的圖片可能有部分區域超出邊界被裁切掉，若要避免這種情況，可以加上選擇性參數 **expand=True** 自動擴展圖片，下面是一個例子，你可以從中比較有無自動擴展的效果。

\Ch10\rotate.py

```
from PIL import Image

# 開啓圖片 ( 建立圖片物件 )
im = Image.open('E:\\girl.png')

# 旋轉 20 度
im_rotated1 = im.rotate(20)
# 另存新檔
im_rotated1.save('E:\\new_girl1.png')

# 旋轉 20 度並自動擴展
im_rotated2 = im.rotate(20, expand=True)
# 另存新檔
im_rotated2.save('E:\\new_girl2.png')
```

❶ 原始圖片　❷ 旋轉 20 度 (部分區域被裁切掉)　❸ 旋轉 20 度並自動擴展

下面是一個例子，假設有一家公司想要了解顧客的年齡分布，以更精準地調整產品和行銷策略，於是蒐集顧客的年齡樣本資料，然後分組並繪製成直方圖，從執行結果可以看出，顧客的年齡是集中在 20 ~ 40 歲。

\Ch11\hist.py

```
import matplotlib.pyplot as plt

# 顧客的年齡樣本資料
ages = [18, 21, 25, 25, 26, 27, 28, 29, 34, 35, 35, 36, 37, 38, 40,
45, 47, 50, 55, 60, 70, 72, 81]
# 自行定義分組
bins = [10, 20, 30, 40, 50, 60, 70, 80, 90]
# 繪製直方圖
plt.hist(ages, bins, color='pink')

# 設定標題與軸標籤
plt.title('Customer Age Distribution')
plt.xlabel('Age Groups')
plt.ylabel('Number of Customers')
# 顯示圖表
plt.show()
```