



# { 環環相CODE }

👉 拼湊到編碼：從Scratch到Python無縫過渡

由Scratch入門

由建立興趣及  
成就感入門

循序漸進學習Python

延伸各式能力



快搜 期刊大師



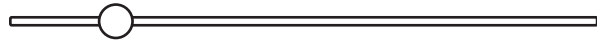
或掃 QR Code 下載



# ✦ 作者介紹 ✦



高慧君 |  
高中教師



賴錦緣 |  
國立臺中科技大學教授

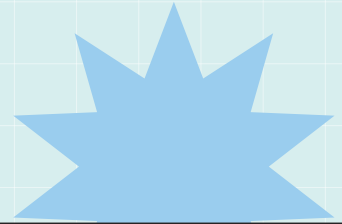


魏澤人 |  
國立陽明交通大學教授





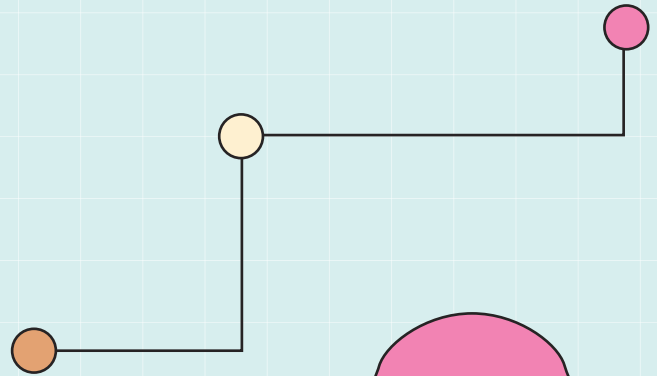
# 目次



1 Scratch 的設計哲學初探 ..... P2~P6

2 從積木式 (Scratch) 語言  
到文字式 (Python) 語言 ..... P7~P9

3 Python 生態系的介紹與應用 ..... P10~P23



## 前言

隨著時代的進步，程式語言從早期的科學計算用途轉變為問題解決的工具，而 Scratch 和 Python 作為兩種程式語言，承擔了初學者從簡單邏輯到進階思維的過渡角色。Scratch 由美國麻省理工大學 (MIT) 開發，旨在讓程式設計變得更加容易入門；然而，當學習者希望進一步掌握更強大的工具並應用到更廣泛的領域時，Python 則是最佳的下一步選擇。

本書將帶領讀者從拼湊式的 Scratch 過渡到文字式的 Python，介紹兩者的設計理念與特點，並探索 Python 的強大生態系統。希望透過這本書，讀者能夠掌握程式設計的核心知識，並為未來的進階學習做好準備。



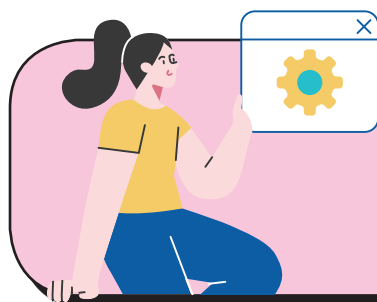
# 01 Scratch 的設計哲學初探

## 一、Scratch 的誕生

隨著時代的演進，程式設計從早期科學計算用途，轉變為問題解決的邏輯設計與物理的動力機械控制。現代為程式初學者量身訂製的程式語言，已引入多媒體運用與互動控制等功能，其中美國麻省理工學院於 2007 年推出 Scratch 後，隨即引起廣泛的注意與重視。Scratch 背後有什麼設計哲學，促成它會那麼成功地成為小孩子入門程式設計的一項重要教學軟體呢？

## 二、Scratch 的設計理念

Scratch 與其它程式語言有什麼差異呢？研究團隊的創始人 Seymour Papert 在 1968 年從 LISP 語言的基礎裡創立 Logo 程式語言，作為孩子們進入程式設計領域很好的入門磚，但是為什麼早期的 Logo 或其它程式語言沒有普遍地流行呢？



Papert 曾經指出好的程式教學語言應該要有二個特色：

- 一、低門檻：要能夠很容易入門。
- 二、高天花板：要能夠讓學習者有機會發展出複雜作品。

繼 Papert 之後，美國麻省理工大學媒體實驗室裏，由 Michael Resnick 所領導終身幼兒園（Lifelong Kindergarten）研究團隊，致力於開發新的技術與活動。他們認為學習應該就像在幼兒園裏面玩耍一樣，可以體驗充滿樂趣的創造過程。

研發團隊與樂高公司密切合作發展出可程式化積木組（Lego Mindstorms），在合作的過程中，他們發現當小孩子拿到一盒實體積木，就會開始拼湊與組合積木，過程中會不斷地帶來新點子，而就在小孩子玩耍時，計畫與目標是有機式地演化，伴隨著結構與故事的發展。

研發團隊被小孩子玩積木的方式觸發靈感，在 2007 年創造出由下而上拼湊式的積木程式設計教學軟體 Scratch，除了基於 Papert 提出的二項特色做為核心原則的設計理念，並且融入第三項特色：



三、寬廣牆面：  
要能夠迎合不同學習興趣與學習方式的人。

### 三、Scratch 的特色

Scratch 成功地將複雜的程式設計創作變成像堆積木一樣簡易的過程，讓程式設計不再是工程師才能涉足的領域，小學生都可以透過積木設計學會程式語言。就像樂高積木一樣，在 Scratch 中，程式積木的形狀會引導小孩子什麼樣的積木是能被放在一起，並嚐試用不同順序與組合來看看會發生什麼事，因此，它沒有傳統程式語言艱澀的語法與嚴謹的邏輯，小孩子可以開始用拼湊（tinkering）的方式來寫程式。

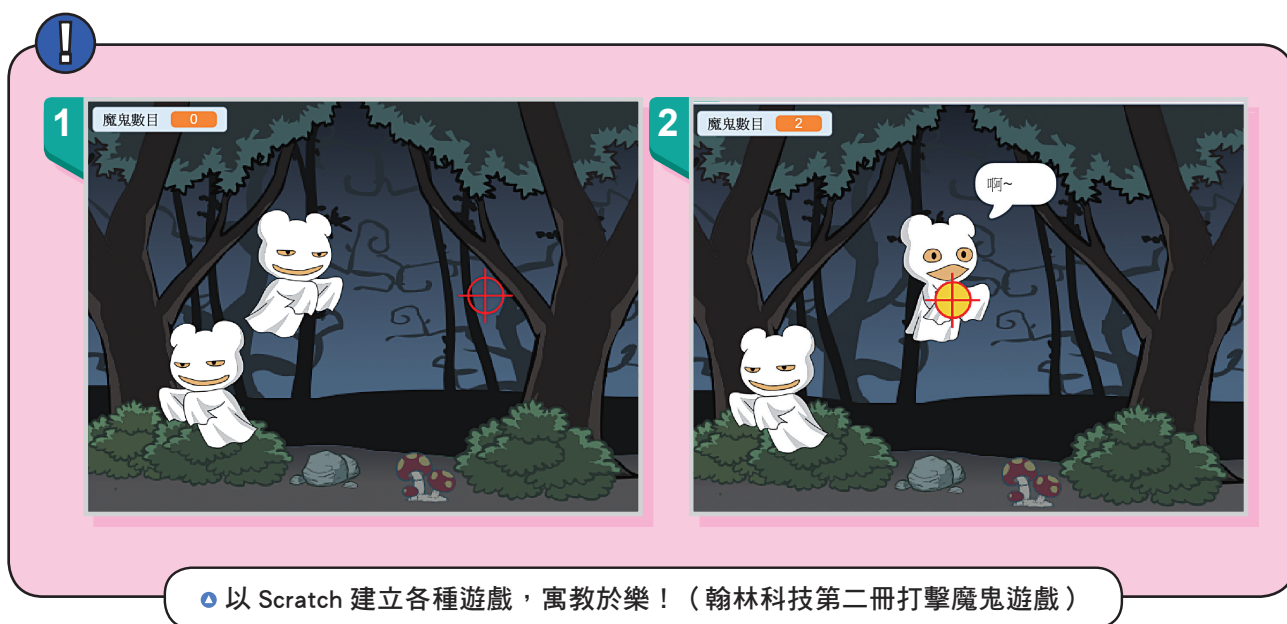
Scratch 這個本身就高度隱含了拼湊的含義，它來自於嘻哈音樂的 DJ，會用雙手來回地轉動碟片來製造出音樂，將音樂有創意地拼湊組合在一起。



在 Scratch 中，活動也是類似的，小孩子用程式積木來拼湊圖像、動畫、照片、音樂、聲音等，而且 Scratch 被設計為高互動性，只要在程式積木上點一下就可以執行，甚至於可以在程式執行中修改程式碼，所以可以隨時立即與互動地實驗新想法。

在 Scratch 程式編輯區中，就好像真實世界中的桌面，可以放置多餘的程式積木，代表著訊息有點亂或實驗性都是無所謂的。大多數的程式語言使用由上而下的設計，而非由下而上的拼湊，在 Scratch 中寫程式，強調互動與漸增性風格，入門的門檻是低的，而寫程式的經驗是可玩耍的，研發團隊希望小孩子在使用 Scratch 的經驗是舒適的。

研發團隊認為當小孩子在製作對個人有意義的專題時，學得最好也享受其中最多，因此在設計 Scratch 時有融入了多樣性與個人化這二項先決的準則。每個小孩子喜好的事物不太一樣，Scratch 被設計為能支援各種不同型態的專題（故事、遊戲、動畫、模擬等），以因應不同小孩子廣泛的興趣，讓他們能夠製作自己感興趣的作品。在 Scratch 中，小孩子可以畫圖、錄音、匯入照片或音樂等來建構出個人化的素材，增加學習樂趣。而 Scratch 選擇以 2D 影像來展現視覺的多樣性。



## 四、Scratch 的社群分享與學習

Scratch 不僅風靡了全世界，更重要的是，它鼓勵使用者創造自己的作品並且分享，官網上來自超過 150 個國家的使用者，在這個社群創造並分享了超過兩千八百萬個作品，包括複製著名遊戲、文藝創作、描述生活經驗，或是賀卡等數不盡的型態，成為程式設計教育的範本。

Scratch 官網除了提供大量作品外，也提供了很好的機制來讓小孩子學習用分享的作品進行社交活動，形成了一個專業社群。教師也可以為班級建立專屬的藝廊 (Studio)，促進更密集的区域性社群成果分享與交流。除此之外，網站提供了很好的搜尋機制，使用者可以輕鬆搜尋作品名稱、特定帳號等。



翰林「一起學程式」統整各冊程式素材，免帳號登入



立即體驗



E-Game 以遊戲闖關模式，熟悉 Scratch 程式編碼邏輯！



立即體驗

## 五、結語

雖然 Scratch 本身強調由下而上思考，提供機會給小孩子主動建構自己的知識，但是由於小孩子的經驗不夠，多數往往停留在下方上不來，所以對很多學習者來說，只發揮了 Scratch 低門檻與牆面寬廣的特性。

筆者一直思考著如何能夠調高小孩子學習 Scratch 的天花板，把 Scratch 學到精，而不是流於湊熱鬧，在親身經歷了 8 年的 Scratch 高中生（初學者）教學歲月裡，深刻地感受到循序漸進的教材設計的重要性。

一份好的教材能夠引起學習者的興趣，讓他們願意主動探索與理解程式碼，並且提供鷹架讓學生一步一步提昇自己的視野。有些學生能夠舉一反三，有些學生能夠舉一反一，有些學生需要舉三反一，因此，要能迎合多數學生需求的程式教材，應該要能夠對於重要的程式觀念有充份的範例讓學生學習，並且要能夠同步培養學生的思考力，讓學生能夠掌握重要概念後，可以從一個問題學習遷移到另外一個問題，從一個情境學習遷移到另外一個情境，而這個能力正是運算思維所強調的抽象化。





# 02 從積木式 (Scratch) 語言 到文字式 (Python) 語言

## 一、Scratch 的教育價值與挑戰

Scratch 本身強調由下而上思考，提供機會給小孩子主動建構自己的知識。但是小孩子的認知資源有限，所以透過積木式的 Scratch 程式，雖然能夠讓小孩子先跳過程式語法的困擾，在不用擔心語法錯誤的情況下，盡情地用拼湊的方式來創作程式作品。然而，因為小孩子的經驗不夠，多數往往停留在下方上不來。



筆者們認為在 Scratch 教材設計方面，就要能夠在有限的資源中，設計出最簡要的知識，這樣才能夠達到「Less is More (少即是多)」的效果。

## 二、選擇 Python 作為銜接語言的考量

在開發了 Scratch 積木式程式教材之後，筆者們一直在思考要選用哪一個語言作為銜接 Scratch 之後的文字式語言。

在高中資訊課程部份，自 98 學年度開始實施的課綱資訊科技必修 2 學分課程中，以電腦與問題解決單元與程式設計最為重要，而這二個單元最強調的是培養垂直式思考能力，就是要把問題想得很深入。然而，受限於時數過少，往往只能表面地介紹解題工具的特性。

在多年的教學經驗中，我們一直思考著如何培養垂直式思考能力，而這也是我們在高中端課程選擇使用 Python 語言的最主要因素之一。雖然市面上有各式各樣的程式編輯工具可以解決問題，在基於培養學生垂直式思考的信仰之下，我們認為高中階段學生能夠精熟一種語言即可，因此，我們認為 Python 最適合用來作為銜接 Scratch 之後的文字式語言。

### 三、Python 的設計理念

Python 是 1991 年由 Guido van Rossum 所創作，目前最新的版本是 3.12.4（2024 年 6 月 6 日，穩定版本）。除了原本的「結構化程式」架構外，Python 亦是「物件導向程式設計」語言。不同於以往的程式語言（如：C, C++, C#, Java, Visual Basic）是使用編譯的方式來執行程式，Python 本身選擇使用直譯的方式來執行程式，是目前排名第一的程式語言。Python 的設計理念是「優雅」、「明確」、「簡單」等三項原則：

#### ▶ 優雅

相較於傳統的程式語言（以 C 為例），Python 使用「縮排」來精簡程式碼區塊，C 語言使用「大括號」來定義程式碼區塊，因此，在程式碼的可讀性方面，Python 的程式碼就比較類似日常生活中的英文閱讀。

#### ▶ 明確

相較於傳統的程式語言（以 C 為例），Python 程式碼可以接受使用中文做為變數或函數名稱，在英文的部份，有區分大小寫不同。Python 使用「小括號」、「中括號」、「大括號」來表示跨起多行的區塊，省略掉像 C 語言一樣的終止字元（分號），另外，搭配「縮排」的使用，使得程式碼區塊閱讀與寫作都顯得更明確。

#### ▶ 簡單

相較於傳統的程式語言（以 C 為例），Python 不需要事先宣告變數，也不需要事先定義變數的資料型態，因此，在撰寫程式碼時，初學者不需要了解太多的電腦背後基本原理，就可以進行問題解決程式設計學習。Python 的程式碼就比較類似日常生活中的英文寫作。

#### 四、結語

除了上述的概念外，我們選擇 Python 做為銜接 Scratch 之後的文字式語言，最主要原因是 Python 是由「Python 軟體基金會」所經營，它是一個從 2001 年成立的非營利組織，該基金會其中一項重要的工作，是跟「開放原始碼促進會（Open Source Initiative）」合作，確保相關的使用授權。



Python 軟體基金會，是一個致力於 Python 程式語言的非營利組織

基於這項工作的成果，Python 在全世界擁有廣大的社群，因此，有各式各樣的第三方開源套件，可以廣泛地運用在不同的領域。有人形容 Python 像是強力膠，配合不同套件它就會變成不同的工具，以便解決各式各樣的問題。因此，我們不必選用其它的數位工具，就可以解決許多問題，一方面又符合垂直式思考能力的精神。



# 03 Python 生態系的介紹與應用

## 一、Python 的崛起與多元生態系

Python 是目前最流行的程式語言之一，在許多不同的領域中，都被廣泛使用，形成了一個龐大的生態系，有著各式各樣的程式庫、開發工具。幾乎所有你能想到的功能，都能從網路上找到相關的 Python 套件。

## 二、為什麼 Python 生態系如此龐大？

為什麼 Python 會有這麼龐大的生態系呢？以下是一些可能的原因：

### 1 簡單易學，人人都愛

Python 的進入門檻較低。想印出 "Hello, World!" 只需要寫 `print("Hello, World!")` 就行了，因此成為最熱門的入門程式語言之一，因為使用者基礎龐大，也就有更多人來開發各種有趣的工具和套件。

### 2 應用廣泛，無所不能

Python 不只是用來寫小程式，它還能做數據分析、人工智慧、網站開發。光是使用內建模組，就能用 `json` 模組處理資料。用 `os` 模組來控制電腦作業系統的功能。用 `math` 模組做數學計算。涵蓋範圍之廣，就像是知名全球連鎖速食店或者便利商店一樣，不管到世界上什麼地方都能看得到。因而變成讓人放心且信賴的選擇。



### 3 跨平台、到處都能用

不管你用的是 Windows、Mac 還是 Linux，Python 都能使用。也有很多網站，能讓你在雲端使用 notebook 介面執行 Python 程式碼，來進行各種實驗和計算。

### 4 開源社群傳統

Python 本身是開源軟體，也繼承了開源社群的共享開放傳統，當你遇到問題時，在網路發問，常常都能很快得到幫忙。這種開放和友善的氛圍，讓更多人願意為 Python 開發新的工具和套件。

## 三、Python 的「內建電池」

雖然 Python 有著龐大的第三方套件，但 Python 本身的內建功能就相當完善。想像你買了一台超酷的遙控車，打開盒子時發現，不僅有車子，還附帶了電池、遙控器和各種額外配件，一應俱全，讓你可以立即開始遊玩。這就是 Python 的「batteries included」（內建電池）哲學！

現在讓我們來看看 Python 標準模組中，究竟自帶了哪些「電池」吧：

### 1 常見的標準功能

- `os`：作業系統小助手，能幫你與操作系統互動，比如創建或刪除文件夾。例子：`os.mkdir("這是一個新資料夾")` 就能創建一個新資料夾。
- `datetime`：時間管理大師，能處理日期和時間相關的所有事情。例子：`datetime.now()` 能告訴你現在幾點幾分。
- `math`：數學計算高手，提供各種數學運算，如 `math.pi` 給你圓周率的值，`math.sqrt(16)` 計算平方根。
- `random`：隨機數生成器，能生成隨機數，洗牌，隨機選擇等。例子：`random.choice(["剪刀","石頭","布"])` 可以幫你出拳猜拳。

## ② 資料處理相關



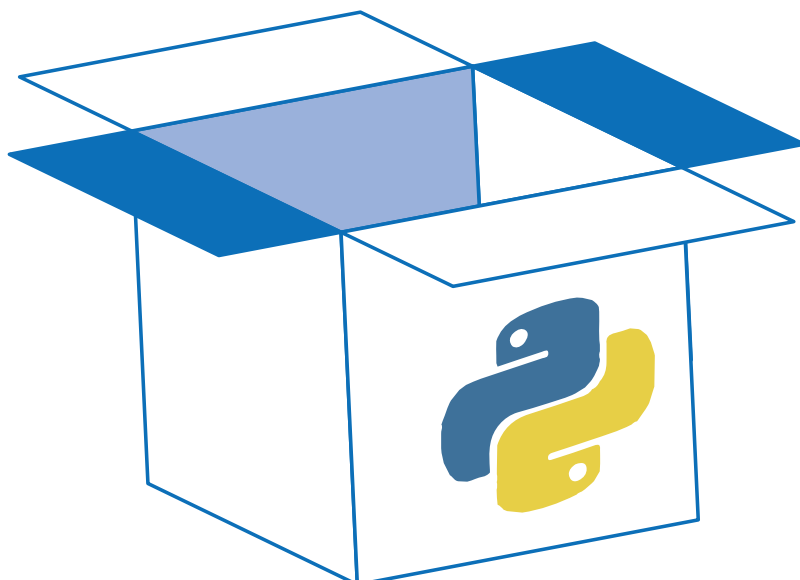
- Python 內建 `json`、`csv`、`tomllib` 模組來處理常見的檔案格式。比方 JSON 是網路上面常見，常用於網路數據交換。你完全不用自己去寫，只要使用 `json.loads({'name': '小明', 'age': 15})` 可以將 JSON 字符轉成 Python 物件。
- `sqlite3`：是一個輕量級資料庫，速度快，安裝使用容易。在 Python 中，你甚至不需要額外安裝第三方套件，標準庫就直接支援。



## ③ 網路相關支援



- `urllib`：網路通信好幫手，能抓取網路資料。比方 `urllib.request.urlopen("https://www.python.org").read()` 可以獲取 Python 官網的內容。
- `http.server`：迷你網站服務器，只需在命令行輸入 `python -m http.server 8000`，就能快速創建一個簡單的小型網站，分享電腦中的文件或者網頁內容。
- `html.parser`：HTML 解析專家，幫助你分析和提取網站中的 HTML 網頁的內容。
- `webbrowser`：讓 Python 控制你的網頁瀏覽器。比方 `webbrowser.open_new_tab("https://www.wikipedia.org")` 可以在網頁瀏覽器的新分頁打開維基百科網頁。



#### ④ 演算法以及資料結構

- Python 的資料型態以及演算法，可以節省你在撰寫演算法細節上的時間。比方 `itertools.combinations` 這個函數可以幫你生成一個集中所有可能的組合。

#### 舉例來說：

```
from itertools import combinations
# 假設我們有以下不同種類的水果
fruits = ['蘋果', '香蕉', '橘子', '葡萄']
# 我們想要選擇 2 個不同水果的所有可能組合
for combo in combinations(fruits, 2):
```

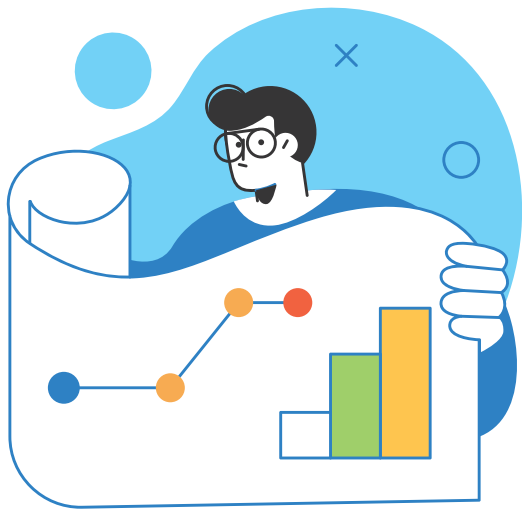
```
    print(combo)
```

輸出結果：

```
('蘋果', '香蕉')
('蘋果', '橘子')
('蘋果', '葡萄')
('香蕉', '橘子')
('香蕉', '葡萄')
('橘子', '葡萄')
```



想像一下，你想知道哪兩種水果混和在一起做出的果汁味道最適配，你想要窮舉所有可能性來試試看，使用 `combinations` 函數，就能輕鬆幫你列出所有可能的組合。



這些只是冰山一角，除了以上的模組外，Python 標準庫還能在 windows 電腦播放聲音、用 tkinter 製作視窗程式或者寫遊戲、讀取許多不同壓縮檔的內容等各種任務，且不需要額外安裝任何東西。這就是 Python 內建電池的特點。

#### 四、Python 生態系：驚奇無限的遊樂園

軟體的世界就像是一個龐大的遊樂園，你可以在其中到處探索。在不同地方你可以找到各種道具積木，來幫助你探索。但可惜的是，很多積木是不相容的，因為他們是使用不同的程式語言以及開發環境設計的。因此你不容易將各種不同的功能結合。

而有人稱 Python 為一種膠水語言，能讓各種原本不相容的積木組合在一起。當然，實際上還是需要有人先寫出一段簡短的程式，把這個積木包起來，讓 Python 能使用他。Python 不是唯一被拿來當成膠水使用的語言，只是因為 Python 的使用者基數龐大，又有開源社群共享分享傳統，所以有非常多的人幫忙將各式各樣的軟體打包，以致於實際上來說，你可以在 Python 生態系中找到各式各樣的積木。

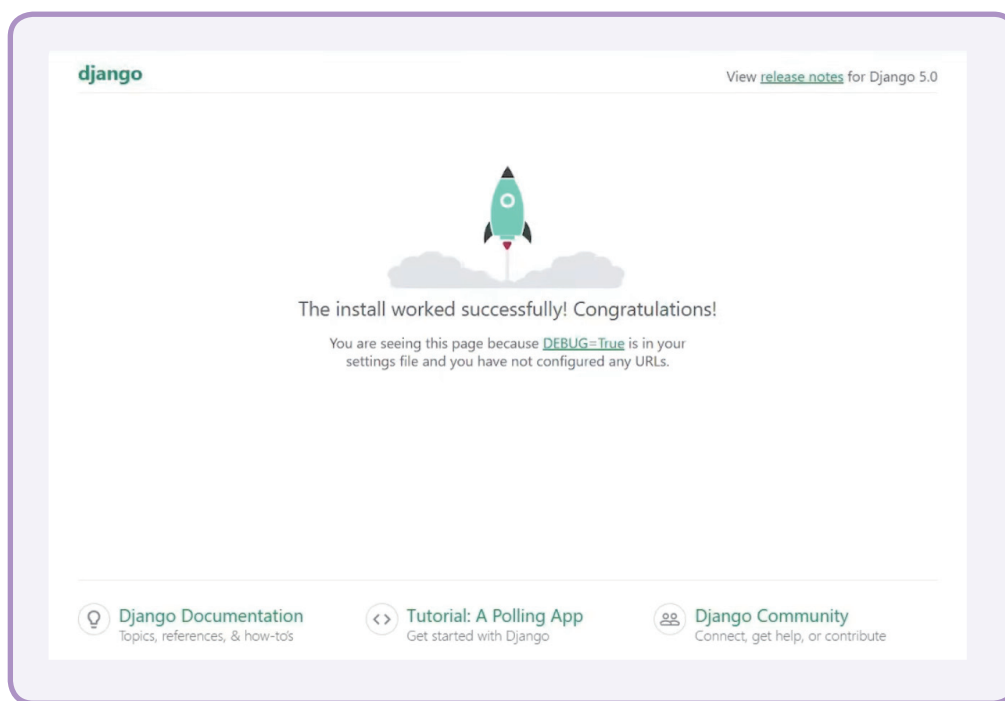
以下我們介紹一些不同領域中的 Python 套件：

##### ① 網頁開發 </>

Python 是適合用來開發網站的工具之一，目前 Python 網頁開發世界的三大框架為 Django、Flask 和 FastAPI。每個框架都有其獨特的優勢和應用場景。



Django 是全功能網站框架，包含了架設網站上所需的各種功能，提供了從資料庫操作到後台管理的解決方案，適合完整大型的網站開發。Django 有自己的規範，自成一個生態圈，也有許多第三方的套件支援，適合搭建標準的網站的開發。



• 預設的 Django 頁面

Flask 跟 Django 正好相反，非常輕巧靈活。作為一個輕量級的 Web 框架，Flask 給予開發者極大的自由度。它的核心非常簡潔，跟 Django 的全能相反，只專注在 Web 框架上，透過其他套件來增添功能，像是操作資料庫或後臺管理、網頁模板等等。好處是靈活自由，但沒有單一的規範。當你有特定需求時，Flask 可以讓你掌控每一個細節。適合中小型的網站原型，也可以搭配其他框架使用。



• Flask 是一個使用 Python 編寫的輕量級 Web 應用框架

FastAPI 是網頁開發領域的新秀。顧名思義，FastAPI 特別擅長構建高效能的 API，其速度可以媲美 NodeJS 和 Go。雖然不像是 Django 這樣重量級，但規範上利用了現代的 Python 的類型提示特性，不僅提高了程式可靠性，還能自動生成 API 文檔。很多新的軟體專案選擇使用 FastAPI。

註：應用程式介面（Application Programming Interface，簡稱 API），是一組規則和工具，讓不同軟體之間可以互相溝通，並共享功能或資料。



# FastAPI

● FastAPI 適合構建 API 與微服務

除了框架外，SQLAlchemy 常常用來搭配 Flask 和 FastAPI 來操作資料庫（Django 有自己的 ORM 來操作資料庫），它支持多種資料庫，能以高層抽象的 ORM 方式來存取資料庫內容，也有底層操控能力。而 Jinja2 是常見的模板引擎，用來動態生成網站內容，可與 Flask, FastAPI 一起使用。

註：物件關聯對映（Object Relational Mapping，簡稱 ORM），是將程式中的物件轉換為資料庫表格的技術，方便操作資料庫。



● SQLAlchemy 是 ORM 工具，簡化資料庫操作，支援多種資料庫

## ② 遊戲開發 </>

在 2D 遊戲開發方面，Pygame 是最有名的套件，許多初學者和獨立開發者的首選。PythonArcade 是另外一個選擇，提供了更高性能和簡潔的語法，適合快速原型製作和教育用途。



• Pygame 是跨平台 Python 模組，專為電子遊戲設計



特別值得一提的是 Ren'Py，這是一個專門用於創建視覺小說和互動式故事遊戲的引擎。Ren'Py 使用簡單的腳本語言，讓即使是編程新手也能快速上手，創作出引人入勝的故事型遊戲。它支持分支對話、複雜的劇情選擇，以及豐富的視覺和音頻效果，使其成為敘事驅動型遊戲開發的理想選擇。

• Ren'Py 視覺小說引擎是一款開放原始碼的自由軟體引擎，用來創作透過電腦敘述故事的視覺小說

在 3D 遊戲開發領域，Panda3D 提供了強大的 3D 渲染和物理模擬能力。Godot 引擎雖然不是 Python 專用，但其 Python-like 腳本語言 GDScript 使 Python 開發者能夠輕鬆過渡，同時享受其全面的編輯器和跨平台能力。Unity 以及 Unreal engine 其實也有 Python 腳本或者 API 支援，但相對來說並不是最流行的解決方案。



# PANDA3D

• Panda3D 是開源 3D 引擎，支援遊戲開發與視覺模擬

Blender 是一個完整強大的 3D 創作平台，內建 Python API。開發者可以利用 Python 控制 Blender 的各種功能，從模型操作到動畫製作，除了增加工作效率外，也能結合 Python 的能力，擴增各種功能。



• Blender 是免費開源 3D 創作工具，用於建模、動畫、渲染、特效製作

此外，PyMesh 能幫忙處理遊戲的 3D Mesh，而 PyBullet 則提供了遊戲物理模擬功能。

### ③ 資料科學及人工智慧 </>

在資料科學及人工智慧領域，Python 是最主要的程式語言。

在資料科學方面，Python 的生態系統提供了一系列強大的工具，使得數據的處理、分析和可視化變得更加高效和直觀。

這個生態系統的基石是 NumPy，它為科學計算提供了高效的多維數組操作。建立在 NumPy 之上的 Pandas 則進一步擴展了數據操作的能力，提供了 DataFrame 這樣的結構，使得複雜的數據清理和轉換變得簡單。



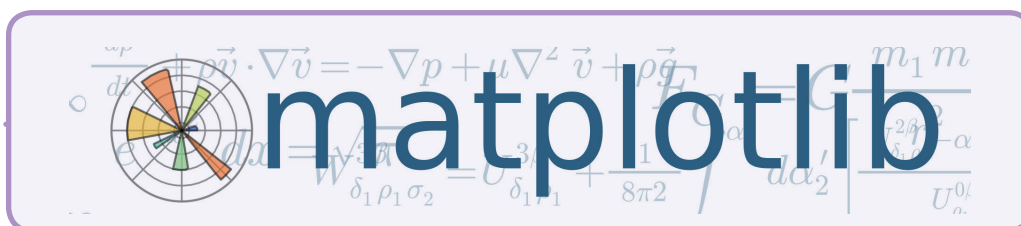
• NumPy 是科學計算庫，提供高效數值運算和多維陣列處理功能

對於更專業的統計分析，SciPy 補充了 NumPy 的功能，提供了更多高級的數學工具，包括演算法、線性代數和信號處理等。當需要處理大規模資料時，也有如 Dask、Ray 等等工具，提供平行或是分散式計算的能力。

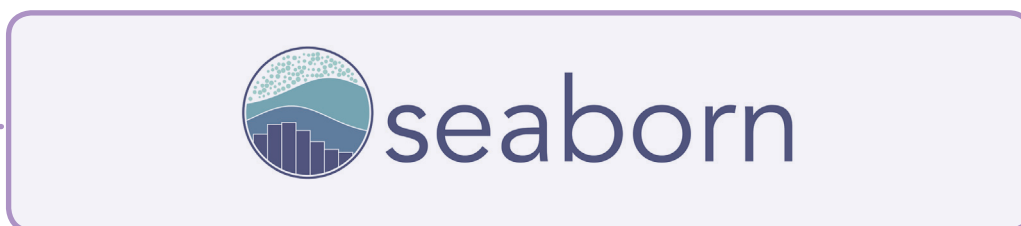


• SciPy 擴展 NumPy 功能，支援優化、積分、統計等計算

數據可視化是資料科學中不可或缺的部分，Matplotlib 在這方面扮演著核心角色。它提供了豐富的繪圖功能，能夠創建各種類型的統計圖表。Seaborn 則在 Matplotlib 的基礎上，提供了更高階的介面以及更美觀的圖表風格。plotly 是另外一套也很流行的資料可視套件，更著重於在網頁上呈現互動的圖表。



• Matplotlib 用於創建靜態、互動式和動畫式視覺化圖表



• Seaborn 基於 Matplotlib，提供美觀的統計圖表



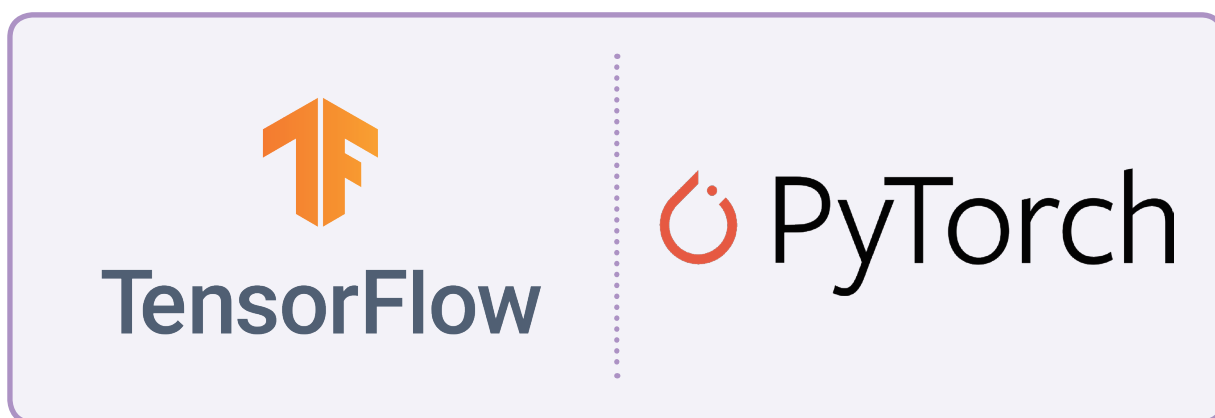
• plotly 是互動式視覺化工具，適用於動態和網頁嵌入圖表製作

在機器學習方面，Scikit-learn 是最流行的套件之一，它提供了大量的機器學習演算法，包括分類、回歸、分群等，並且具有一致且易用的 API。對於計算機視覺任務，scikit-image, OpenCV-Python 提供了豐富的圖像處理功能及演算法，而 Pillow 則更適合於基本的圖像操作。



- Scikit-learn 是機器學習庫，提供分類、回歸、聚類等多種算法工具

深度學習領域則是 TensorFlow 和 PyTorch 這兩個框架佔有率最高。由於人工智慧以及深度學習的蓬勃發展，可以在網路上找到大量的人工智慧模型及程式碼，絕大多數都是基於這兩個框架中的一個。



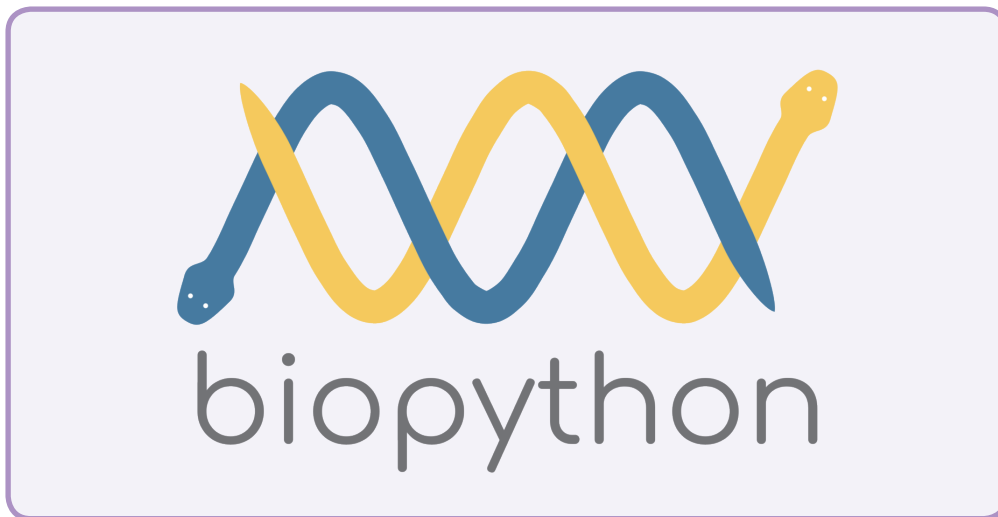
- TensorFlow 和 PyTorch 是兩大深度學習框架，前者擅長生產部署，後者靈活易用，適合研究與開發

自然語言處理是人工智慧的另一個重要分支，NLTK 是一個有名的套件，提供非常多樣的演算法和工具。一般使用上，spaCy 可能是更容易使用的選擇，提供了高效的工業級語言處理工具。而 Transformers 提供了最先進的深度學習語言模型工具，有許多預先訓練好的模型可以很方便的使用，比方 BERT、GPT、Llama 等。

## ④ 科學及數學 </>

Python 有許多科學和數學套件，被研究人員和專業人士廣泛使用。像是 numpy 就是數值計算的重要工具。後來的人工智慧及資料科學生態系，也是以 Python 在科學計算的生態系為基礎。除了資料科學及人工智慧，Python 也有很多其他學門的研究工具建立在這個基礎上。

Biopython 是生物資訊研究者，為了解決研究上遇到的需求所開發的套件。對於基因組學研究，pysam 是一個用於讀取和操作 SAM/BAM 文件的重要套件。



• Biopython 提供序列分析、結構操作等生物數據處理功能

醫學研究者經常使用 Nilearn 來分析神經影像數據。MNE-Python 則是另一個強大的工具，主要用於腦電圖（EEG）和腦磁圖（MEG）數據的處理和可視化。



• Nilearn 用於大腦影像數據可視化與機器學習



• MNE-Python 用於處理和可視化 EEG、MEG 等腦電數據

Astropy 是一個功能豐富的天文學套件，提供了從基本天文計算到複雜數據分析的各種功能。它可以處理天文坐標系統、時間轉換，以及各種天文數據格式。



• Astropy 提供天文數據處理、坐標轉換、時間操作等工具

在物理學領域，QuTiP (Quantum Toolbox in Python) 是量子力學模擬的工具。對於粒子物理學，則有 PyROOT 來串接是 CERN 的 ROOT 框架，用於高能物理數據分析。

VPython 是一個能操作 3D 模擬世界的 Python 環境，特別適合於物理學教育和科學可視化，它使得複雜的 3D 模擬變得簡單直觀，特別是在力學、電磁學和現代物理學等領域。它幫助學生將抽象的物理概念轉化為可視化的模型。

SageMath 是一個開源數學系統，整合了許多開源軟體，適合輔助代數、幾何、數論等多個數學領域的研究。對於圖論研究，NetworkX 是一個方便的套件，提供了一個合適的圖論資料結構，能將圖論演算法應用在實際問題上。



• SageMath 整合多種數學工具，用於代數、幾何、統計等計算



## 5 GUI 軟體及 App </>

在桌面 GUI 開發領域，內建的 Tkinter，適合初學者。而 PyQt、PySide 可以用來開發 Qt 風格的 GUI 軟體。可以使用 Qt 生態系豐富的小部件和工具。對於較為簡單的程式 PySimpleGUI 也是一個不錯的選擇。

而 dash、gradio、streamlit 可以用來開發一些控制面板類型的介面，並且用瀏覽器呈現。特別適合作為科學計算、人工智慧、資料科學等應用的介面。



在手機及平板 App 開發方面，Python 並不是最主流的選擇，但還是有不少可用的框架。Kivy 是一個跨平台框架，能讓開發者使用相同的程式碼來創建適用於 Android 和 iOS 的應用。它的設計理念注重靈活性和創新性，特別適合開發遊戲和多媒體應用。

另一個選擇是 BeeWare，它的目標也是讓開發者能用同一份 Python 程式碼，同時開發 iOS、Android、Windows、MacOS、Linux、Web、tvOS 的應用程式，並且目標是讓產生的程式介面更貼近原生風格。



- BeeWare 能讓開發者用 Python 構建原生桌面和移動應用

## 五、Python，開啟你的程式冒險

Python 就像是打開新世界的鑰匙，上面列出的領域，僅僅是其中的一小部分。其他像是物聯網、自動化及網路爬蟲、金融、區塊鏈、地理、機器人、音樂、多媒體等等，也都有套件來幫助你起步，探索各種新領域，進而將各種不同結合。

# 學習 Q & A



Q：如何從 Scratch 銜接到 Python ？

A：翰林獨家以 Scratch 對照 Python，輔以概念拆解 + 舉例說明，知識輕盈好吸收！

## Python 版面設計

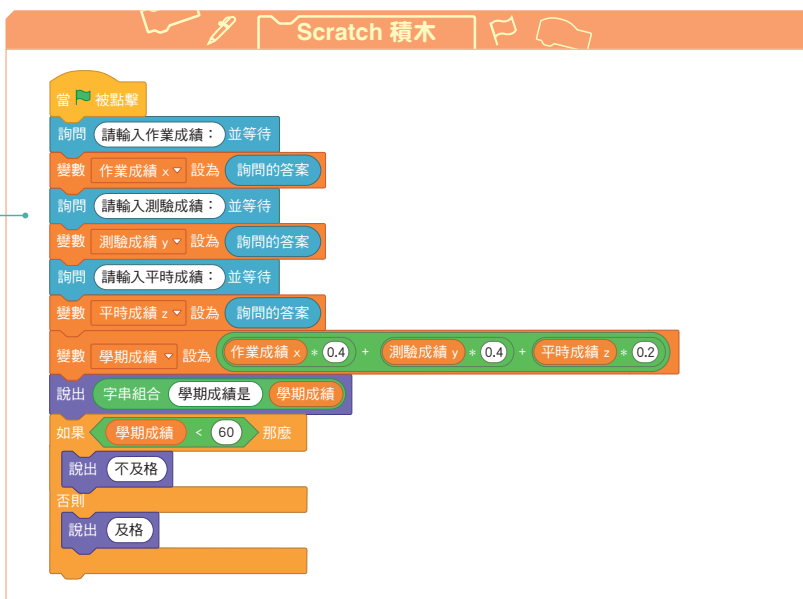
將 Scratch 圖像化程式與 Python 文字型程式碼左右對照、清楚比對，搭配中英對照、程式說明，概念分類清楚統整。

### 範例：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。）

### 例題挑選

挑選一二年級學習過例題進行延伸，加速理解語法邏輯。



### 概念統整

針對基礎概念做主題式統整，搭配舉例說明完整學習。

**Python 中，可使用的關係運算符號如下表所示。**

關係運算符號	意義	範例	運算結果
==	相等	8 == 5	False
		8 == 8	True
!=	不相等	8 != 5	True
		8 != 8	False
>	大於	8 > 5	True
		2 > 8	False
<	小於	8 < 5	False
		2 < 8	True
>=	大於或等於	8 >= 5	True
		8 >= 8	True
		8 >= 9	False
<=	小於或等於	8 <= 5	False
		8 <= 8	True
		2 <= 5	True

**《程式碼》**

```
print(3 + 6 == 2 + 5) # 9==9, 結果不成立, False
print(7 - 2 != 5 + 4) # 5!=9, 結果不成立, True
print(4 + 5 > 3 + 7) # 9>10, 結果不成立, False
print(100 < 100) # 100<100, 結果不成立, True
print(100 >= 100) # 100>=100, 結果不成立, False
print(1 + 3 <= 6 - 2) # 4<=4, 結果成立, True
```

當成立就返回 True，不成立就返回 False。

**Python 中，可使用的關係運算符號如下表所示。**

在日常生活，我們經常會遇到製作決策的情況，例如：如果下雨了，就帶把傘出門。程式的執行也是類似的情況，根據條件式運算結果，執行不同的程式區塊。最簡單或最複雜的單向選擇結構，都統稱為：

```
if 條件式:
    程式區塊
```

if 敘述後面接著條件式，條件式的運算結果是布林值，當條件式為 True 時，就會執行程式區塊的敘述；當條件式為 False 時，不會執行程式區塊，在條件式之後必須有冒號，底下的程式區塊必須縮排。

**《程式碼》**

```
a = 10
b = 8
if (a > b):
    print('變數 a 比較大') # 10>8, 結果成立。
```

**《執行結果》**

```
變數 a 比較大
```

**《條件式》**

```
c = int(input('請輸入分數 1-5 分: '))
if (c >= 3):
    print('滿堂')
```

**《執行結果》**

```
請輸入分數 1-5 分: 4
滿堂
請輸入分數 1-5 分: 3
滿堂
請輸入分數 1-5 分: 1
```

中英對照

保留字		變數	
if	如果	grade	成績
else	否則		
elif ( else if )	否則如果		
False	假		
True	真		

保留字是文字式程式語言中，已經具有特定意義的英文單字，不可用來作為使用者自訂的變數名稱。

中英對照

將常用英文指令整理成表，降低語言門檻，學習不受限！



```

Python 程式碼
1 x = int(input('請輸入作業成績：'))
2 y = int(input('請輸入測驗成績：'))
3 z = int(input('請輸入平時成績：'))
4 grade = x * 0.4 + y * 0.4 + z * 0.2
5 print('學期成績是', grade)
6 if (grade < 60):
7     print('不及格')
8 else:
9     print('及格')
    
```

程式說明

逐行對應 Python 程式碼，說明對照更清楚。

# 將輸入的字串轉換為數字存到變數 x。  
 # 將輸入的字串轉換為數字存到變數 y。  
 # 將輸入的字串轉換為數字存到變數 z。  
 # 計算學期成績存到變數 grade。  
 # 呈現學期成績。  
 # 使用雙向選擇結構判斷結果。  
 # 當 grade<60，呈現不及格。  
 # 當 grade>=60，呈現及格。

小提示

程式區塊的縮排

在 Python 中，自動縮排預設為四個空白字元，也可以使用一個 Tab 鍵，但須注意同一個程式只能使用同一種方式，不可同時使用四個空白字元和 Tab 鍵，否則執行會產生錯誤。

概念拆解 + 舉例說明 = 程式概念扎實！

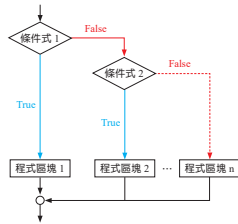
概念 多向選擇結構

如果發生的情況不只兩種，就可以透過使用 elif 做更進一步的選擇，語法如下：

```

if 條件式 1:
    程式區塊 1
elif 條件式 2:
    程式區塊 2
elif 條件式 3:
    ...
else:
    程式區塊 n
    
```

當其中一個條件式為 True 時，就執行相對應的程式區塊，並且當所有條件式皆為 False 時，就執行 else 裡面的程式區塊。



例如

《程式碼》

```

f = int(input('請評分餐點 1~5 分：'))
if (f == 5):
    print('非常滿意') # 假設 1 是 5=5，結果成立。
elif (f >= 3):
    print('滿意') # 假設 2 是 3>=3，結果成立。
else:
    print('不滿意') # 假設 3 是 1>=3，結果不成立。
    
```

《執行結果》

假設 1  
 請評分餐點 1~5 分：5  
 非常滿意

假設 2  
 請評分餐點 1~5 分：3  
 滿意

假設 3  
 請評分餐點 1~5 分：1  
 不滿意



新增舉例 & 執行結果，教學效率提升！

# 學習 Q & A



Q：如何降低 Python 的撰寫難度？

A：以填空的方式完成 Python 撰寫，降低學習難度。

## 專題更新

將過去 1A2B 專題替換，難度大大調降！學生可自行設計繪製的圖樣，操作更簡單有趣！

### 2-3 Python 程式設計的應用

經過前面的課程，同學們已經熟悉了基本的輸入、輸出、迴圈及繪圖的功能，接下來我們要統整學習過的觀念來完成一個程式。

#### 範例：你想畫什麼，我來畫給你看

請設計一個程式，有三個選項讓使用者選擇要畫三角形、六邊形或五角星的圖形。然後再讓使用者輸入想要的圖形數量，畫出對應的圖形結果。



216



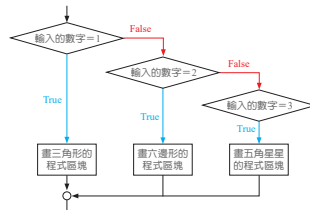
#### 問題分析

我們可以將這個程式拆解為幾個部分如下：

- 如何匯入 turtle 模組，並定位起始位置？
- 執行時，如何設定起始位置，並考慮畫完圖形時，不會超出畫面？
- 如何產生選擇讓使用者輸入數字，決定要畫的圖形與數量？

編號	1	2	3
圖形	三角形	六邊形	五角星

- 執行時，如何製作選擇？



- 如何使用迴圈畫出三角形、六邊形和五角星？

編號	1	2	3
圖形 (●為起點)	右轉 120°	右轉 60°	右轉 144°

- 如何使用迴圈達成重複畫圖形的結果？

資訊科技

Python 圖形設計的應用

217

### 解題步驟

**問題拆解 1** 如何匯入 turtle 模組，並定位起始位置？

**步驟 1** 匯入 turtle 模組，並定位畫筆的起始位置。

1. 匯入 turtle 模組，並提筆。
2. 請同學想想看，設定起始位置時，虛線框要填入什麼？

```
import turtle
t = turtle.Turtle()
t.penup()
t.goto( )
```

**問題拆解 2** 如何產生選項讓使用者輸入數字，決定要畫的圖形與數量？

**步驟 2** 製作選單，並詢問使用者想畫的圖形與數量。

1. 設定圖形變數 draw\_what 與數量變數 draw\_times。
2. 請同學想想看，虛線框要填入什麼？

```
draw_what = (input('輸入想畫的圖形(1.三角形 2.六邊形 3.五角星星:'))
draw_times = (input('你想畫幾個這樣的圖形:'))
```

**問題拆解 3** 如何判斷輸入的數字，分別代表何種圖形？

**步驟 3** 讓程式判斷輸入的數字所代表的圖形。

1. 當輸入的數字為 1，對應的圖形為三角形，以此類推。
2. 請同學想想看，虛線框要填入什麼？

```
if draw_what == 1:
    畫三角形的程式區塊
elif draw_what == :
    畫六邊形的程式區塊
    畫五角星星的程式區塊
```

畫圖形的程式區塊，再完成填碼。



填空式問答，操作更簡單！

218

專題 4 頁完成！



**問題拆解 4** 如何使用迴圈畫出三角形、六邊形和五角星星？

**步驟 4** 撰寫畫三角形、六邊形和五角星星的程式區塊。

1. 請同學想想看，考量畫布大小，畫出一個三角形時，虛線框要填入什麼？
2. 請完成畫六邊形和五角星星的程式區塊。

```
t.pendown()
for j in range(3):
    t.forward( )
    t.right( )
t.penup()
```

#畫三角形的程式區塊。  
#設定三角形邊數。  
#設定三角形邊長。  
#設定三角形角度。

想一想，繪製六邊形和五角星星，旋轉角度分別是多少呢？



**問題拆解 5** 如何使用迴圈達成重複畫圖形的結果？

**步驟 5** 結合前面兩個步驟，畫出指定數量的圖形。

1. 當畫出一個三角形後，產生固定的間隔，再重複畫三角形時，才不會重疊。
2. 請同學想想看，執行計次迴圈時，虛線框內要填入什麼？
3. 請同學想想看，設定畫三角形的間隔距離時，虛線框要填入什麼？
4. 請完成畫六邊形和五角星星的間隔距離設定。

```
for i in range( ):
    if draw_what == 1:
        畫三角形的程式區塊
        t.forward( )
```

#執行計次迴圈。  
#設定間隔距離。

請完成整個專題程式，接著想想看，如果使用者輸入 1 到 3 以外的數字，程式會有什麼反應？

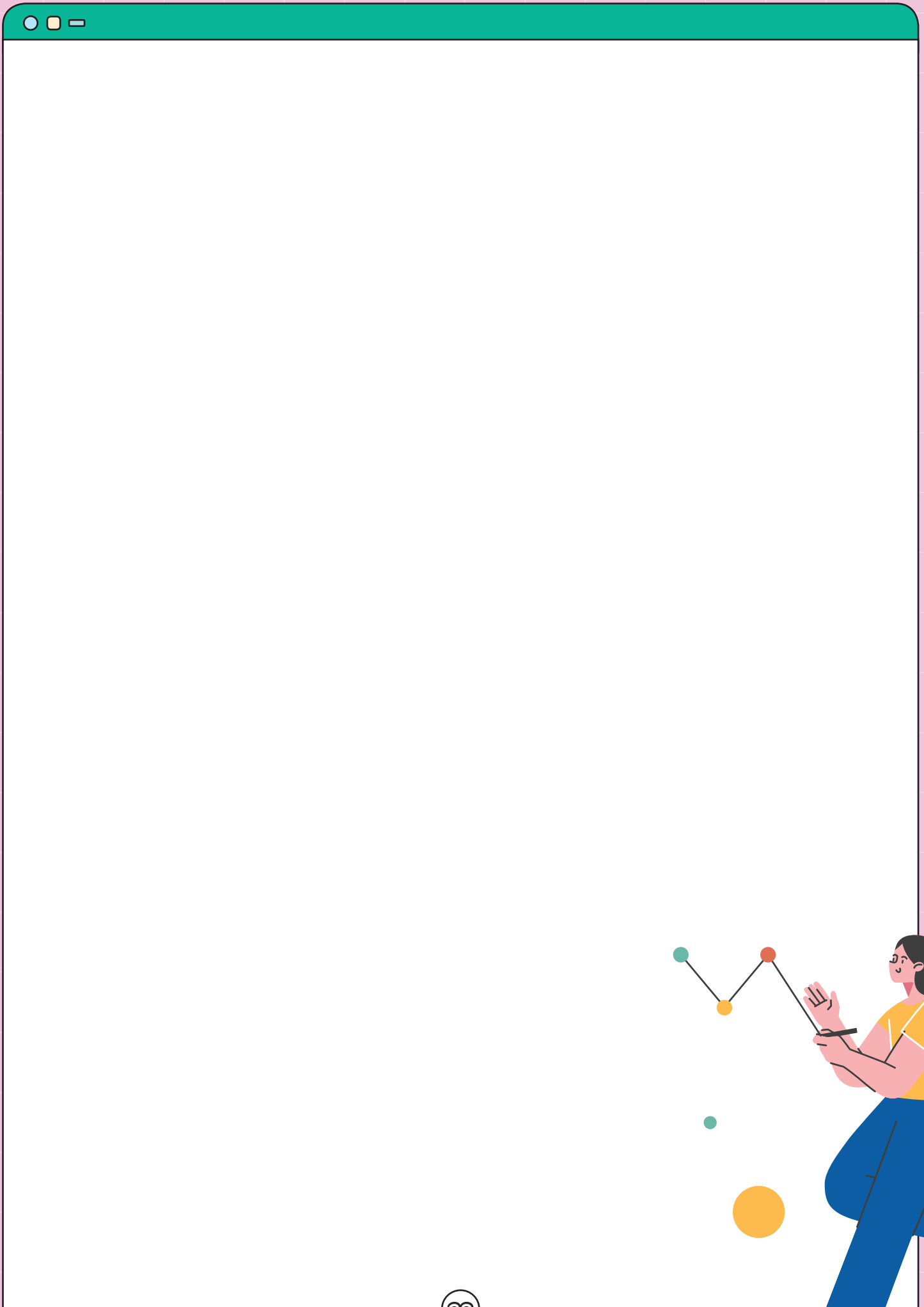


資訊科技

Python 程式設計應用

219

# 筆記頁



數十種天然草本提煉



喉立爽  
爽喉軟糖

特殊嚼感 ✨ 清涼舒暢



天氣變化  
喉嚨卡卡?



上課上班  
急需聲援?



揪團歡唱  
影響發揮?



賽事吶喊  
尷尬燒聲?



讀者限定



輸入折扣碼

HONEYSONG

享超取免運費!



服務專線：0800-52-5959