

序言

本書是利用 Python 程式設計語言製作遊戲與學習演算法的入門書。

Python 目前已是於軟體開發以及學術研究領域普及的程式語言，也是許多企業與教育機構使用的主流程式語言之一。此外，基本資訊技術人員考試也新增了 Python 這項語言，所以許多學習資訊處理的人都有機會接觸它。

Python 之所以如此受歡迎，理由在於：

- 語法簡單，只要幾行程式，就能寫出與其他程式語言一樣程式。
- 寫好的程式可立刻執行，開發效率非常優異。
- 函式庫非常豐富，而且非常簡單好用。

在眾多程式語言之中，Python 特別容易上手、易學，這也是它如此普及的理由之一。

本書的重點在於學習演算法，主要會從帶領初學者入門的程式設計基礎開始，一步步從簡單的演算法學到高階的演算法，讓每位讀者都能讀懂本書的內容。裡頭所提及的演算法是解決問題的步驟或手段。學會演算法之後，就能具備解決各類問題的能力，所以一直有不少人強調學習演算法的重要性。或許大家覺得演算法很難，但請大家不要太擔心，本書會在製作遊戲的過程中，帶著大家學會各種演算法。

希望大家都能一邊開發遊戲，一邊快樂地學習程式設計與演算法。

廣瀨豪

在此先介紹與大家一起學習演算法的登場人物，以及說明一些在開始之前，應該先知道的事情，例如該如何閱讀本書或使用支援網頁的方法。

》》 登場人物簡介

本書登場的是下列兩位人物，他們會幫助大家正確地了解書中內容。「鳩山莉香」是負責補充說明的助手，「豐川優斗」則是與大家一起學習的年輕人。



鳩山莉香

於慶王大學理工學部學習資訊處理技術的理科女子。大學畢業後，進入軟體製作公司「Python Systems」服務，在技術部門負責開發程式。其優秀的技術得到上司青睞，因此被任命為公司內部的指導員。

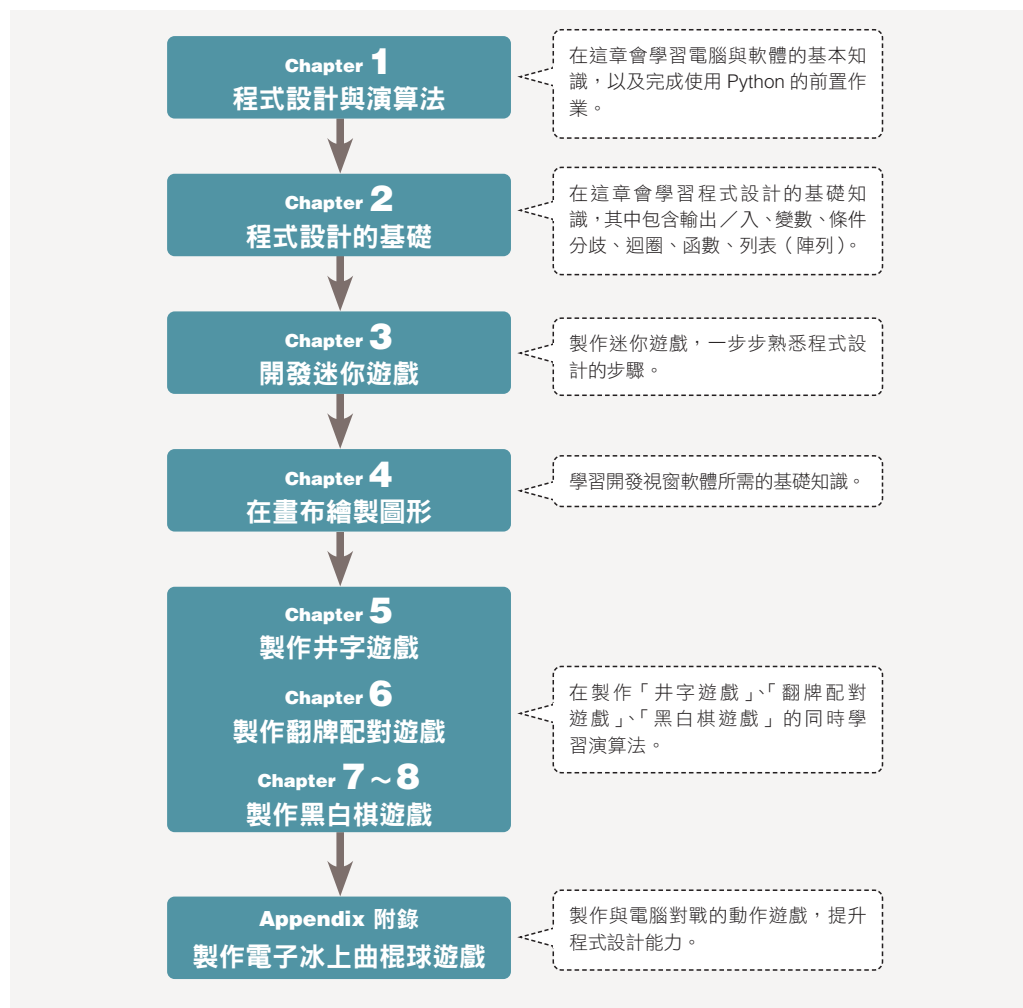


豐川優斗

明收大學經濟學系畢業後，進入 Python Systems 業務銷售部門服務。Python Systems 規定每位新進員工都必須於技術部門進行員工訓練，所以目前在鳩山底下學習程式設計。

》》》 本書的學習流程

本書將依照下列的步驟學習程式設計與演算法。



筆者的建議

就算遇到很難的內容，也不需要自己當下全盤了解，只需要先貼張便條紙，標註一下，然後讀完整章再說。讀完整章之後，請回到剛剛覺得很難的部分。學習程式設計時，偶爾會遇到這種看完其他的部分，原本不懂的部分就豁然開朗的情況，所以建議大家不要太過執著某個部分，先讀完一遍再說。由於本書的主題是遊戲開發，所以請大家放輕鬆，開心地學吧！

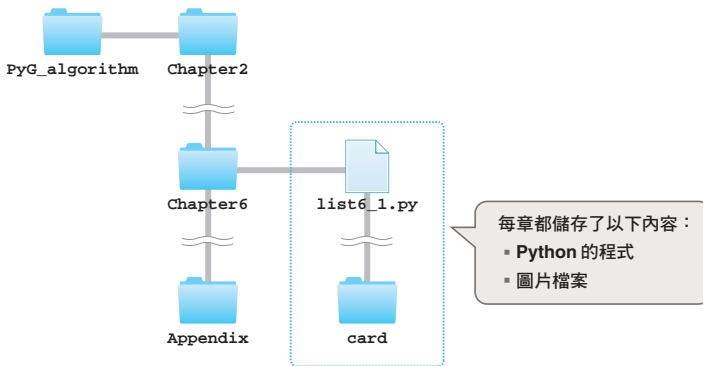
》》》 範例程式的使用方法

本書介紹的程式可於支援網頁下載。請大家至以下網址下載：

<http://books.gotop.com.tw/download/ACG006800>

下載的是以密碼加密的 ZIP 壓縮檔。必須先輸入本書第 295 頁的密碼解壓縮檔案之後，才能使用。

範例檔是依照下圖的結構儲存，每章的範例檔都存在不同的資料夾裡。至於使用的是哪一個程式，則會在程式碼的上方註明檔案名稱。如果自行撰寫的程式無法正常執行，請開啟該章的資料夾，參考其中的範例檔。



》》》 程式碼的撰寫方式

本書介紹的程式是由行編號、程式碼、解說這三個欄位組成。如果程式碼太長，無法寫成一行，就會插入空白，讓程式碼換到下一行。

程式碼 ▶ 範例

列編號	程式	說明
1	<code>import tkinter</code>	載入 tkinter 模組
2		
3	<code>def banmen():</code>	顯示棋盤的函數
4	<code> for y in range(8):</code>	迴圈 x 從 0 遞增至 7
5	<code> for x in range(8):</code>	迴圈 y 從 0 遞增至 7
6	<code> X = x*80</code>	棋格的 X 座標
7	<code> Y = y*80</code>	棋格的 Y 座標
8	<code> cvs.create_rectangle(X, Y, X+80,</code> <code>Y+80, outline="black")</code>	繪製以 (X, Y) 為左上角的正方形
9		
10	<code>root = tkinter.Tk()</code>	建立視窗物件
11	<code>這個函數的處理與之前的內容相同</code>	省略
:	:	

無法寫成一行的程式碼會換到下一行

省略的部分會記載省略的原因

有些剛剛接觸程式設計的初學者會有「電腦的程式到底是什麼？」「演算法到底是什麼？」這類疑問。所以本書要在開始學習程式設計之前，先回答這些問題。就算是很熟悉電腦的人，或許也會從中得到一些新的知識，還請大家先讀過一遍喇！

接下來則是要帶著大家在電腦安裝 **Python**，完成程式設計所需的前置作業。

程式設計 與演算法

1 Chapter



電腦與程式設計語言

要了解電腦的程式就必須先了解電腦如何運作。接下來就為大家說明。

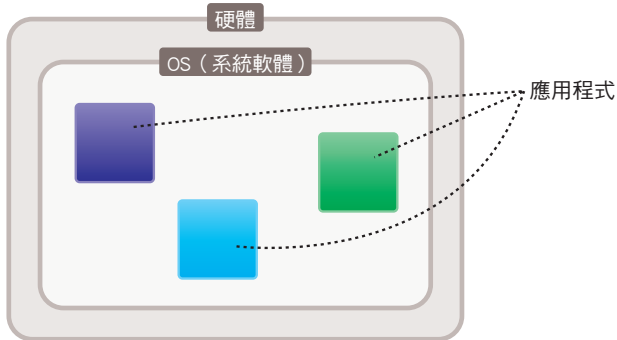
》》》 硬體與軟體

電腦、智慧型手機、電視遊樂器這些裝置都被稱為硬體，而且都是透過系統軟體（作業系統）控制。

不管是電腦還是智慧型手機，都有不同的軟體與 App 正在運作，而這些軟體與 App 都是在作業系統（OS）運作的應用程式。舉例來說，應該有很多人都會使用 Edge 或是 Safari 這類網頁瀏覽器，或是製作文件的 Word 以及試算表軟體 Excel，而網頁瀏覽器與辦公室軟體就是最具代表性的應用程式。

控制硬體的軟體是 OS，在 OS 運作的軟體是應用程式，這些軟硬體之間的關係如下。

圖 1-1-1 硬體、OS、應用程式



接著以智慧型手機為例，進一步說明。智慧型手機通常會搭載 iOS 或 Android 這類 OS。智慧型手機的用途除了打電話，還可以使用社群網站軟體，也可以利用照相機軟體拍照，或是利用電子計算機計算。社群網站軟體、照相機軟體、電子計算機都是在智慧型手機的 OS 運作的應用程式。

控制硬體基本處理的系統軟體，以及在系統軟體運作的各種應用程式，都是利用程式設計語言撰寫。

》》 程式就在日常生活的機器之中運作

剛剛雖然舉出了電腦或智慧型手機這類例子，但其實透過電子迴路與程式驅動的裝置還不只這些，例如電視、冷氣、冰箱、洗衣機、吸塵器這類家電，汽車、摩托車、電車、飛機這類交通工具，自動販賣機、銀行 ATM、超商的多媒體終端裝置，各種機器與機械都內建了電腦元件，這些產品也都利用各種電腦應用程式控制。

圖 1-1-2 程式在日常生活的機器之中運作



原來是這樣。我們在日常生活中使用了許多利用電腦程式控制的東西啊！



對啊！所以我們的生活可少不了電子迴路與程式喲！

什麼是程式？

接著具體說明電腦程式到底是什麼。

》》》 什麼是電腦程式？

電腦程式就像是**命令電腦進行處理的指令表**。接著讓我們以電腦遊戲為例，說明什麼是指令表。請大家想像一下，利用電腦鍵盤的方向鍵或是電視遊樂器的搖桿控制主角的遊戲。

當你按下左鍵，主角就會往左移動，按下右鍵，就會往右移動。這就是透過程式下達

- 建立管理主角座標的變數 x 與 y 。
- 按下左鍵時，讓變數 x 的值減少預先設定的量
- 按下右鍵時，讓變數 x 的值增加預先設定的量
- 於螢幕的 (x, y) 位置繪製主角

這類指令。

圖 1-2-1 向電腦下達指令，驅動遊戲的主角



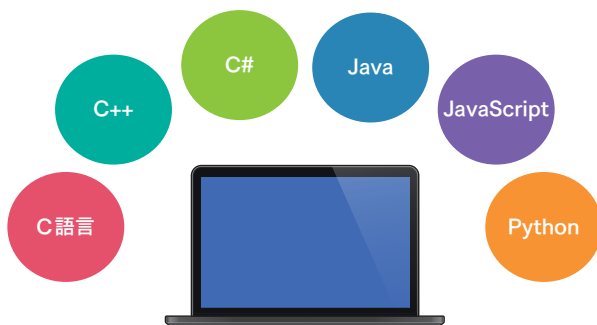
除了遊戲軟體與遊戲應用程式之外，所有的軟體或應用程式都是透過算式與指令組成的指令表（程式）運作。

電腦的應用程式稱為程式碼，有時也直接稱為程式。本書之後都統一稱為程式，而電腦遊戲則稱為「遊戲」。

》》》 各種程式設計語言

用於撰寫程式的程式設計語言中較有名的有 C、C++、C#、Java、Java-Script。

圖 1-2-2 各種程式設計語言



除了這些程式設計語言之外，可能還有人聽過 Swift、Perl、Ruby、VBA 這類程式設計語言。

本書介紹的 Python 是近年來人氣扶搖直上的程式設計語言。越來越多企業以 Python 開發公司內部系統，甚至有些基本資訊技術人員的考試也納入了 Python，對技術人員與學習資訊處理的人來說，Python 是越來越有機會接觸的程式設計語言。



C 語言、C++、Java 在許多系統軟體開發的領域應用，而 C# 則是用來開發 Windows 軟體的程式設計語言，有時也會搭配 Unity 這項工具開發智慧型手機的應用程式。

JavaScript 又是什麼樣的程式設計語言呢？



JavaScript 是於網頁瀏覽器後台運作的程式設計語言。例如在網頁顯示最新資訊或是更新圖片，都是由 JavaScript 進行。

程式真的是無所不在啊！



何謂演算法

接著說明什麼是演算法。

》》》 什麼是演算法

演算法就是解決問題的計算方法，或是解決問題的手段。在過去，演算法的意思是「筆算」，例如要用心算算出 78964×251 或 $98435 \div 736$ 是件很難的事，但只要學會筆算，只要中途沒算錯，就能算出這種多位數的乘法或除法。筆算可說是在計算大數字時的重要手法。

到了現在，演算法的意思是「解決問題的連續步驟」。舉例來說，在知名的數學演算法之中，有一個計算兩個自然數的最大公約數的「輾轉相除法」。

》》》 程式的演算法

在開發程式時，也很常使用演算法這個字眼。舉例來說，電腦程式的演算法就是以程式設計語言撰寫，用於解決問題的步驟。例如處理資料的演算法包含：

- 從多筆資料找出目標值的搜尋演算法
- 將隨機排列的數值依序排列的排序演算法

這類演算法。

圖 1-3-1 演算法範例 搜尋演算法

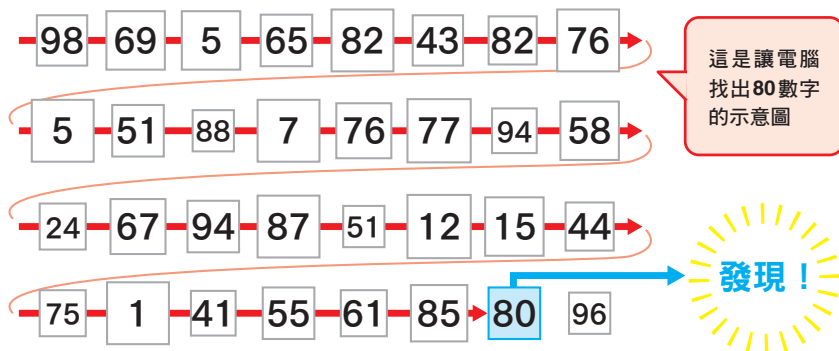
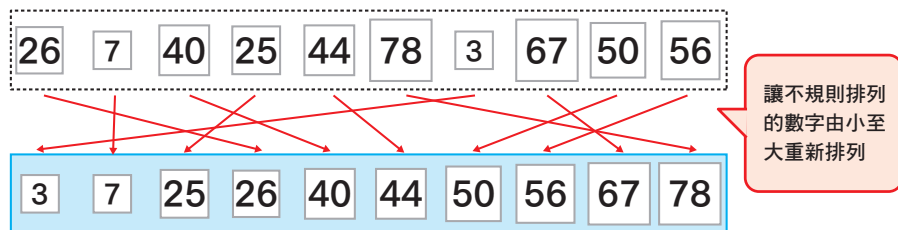


圖 1-3-2 演算法範例 排序演算法



搜尋演算法可用於搜尋文件之中的單字，排序演算法則可用於繪製電腦圖片的時候。搜尋單字或繪製圖片都只是其中一種例子，實際上，搜尋與排序這兩種演算法會應用於各式各樣的資料處理。

在遊戲開發的世界裡，最為有名的演算法就是判斷兩個物體是否接觸的碰撞偵測演算法，而這項演算法將於本書的附錄，也就是製作電子冰上曲棍球遊戲的時候說明。

>>> 解決遊戲開發的問題

讓我們將話題拉回 Lesson 1-2 的移動遊戲主角。假設有人給你一張主角的圖片，要求你「讓這個主角依照輸入的按鍵移動」，這時候如果你知道該怎麼根據輸入的按鍵計算座標，也知道在該座標的位置繪製圖片，就能寫出解決這個要求的程式。換言之，解決「讓主角依照輸入的按鍵移動」這個問題的步驟（演算法），就是「依照輸入的按鍵計算座標，以及在該座標的位置繪製圖片」。從這個例子可以知道，電腦程式的演算法比數學的演算法更加廣義。

原來如此，我之前還以為演算法很抽象，原來是這麼具體的步驟，不過我現在還似懂非懂就是了。



這次會在第 2 章學習程式設計的基礎，從第 3 章開始製作遊戲。我們在寫各種程式的時候，就會知道演算法到底是什麼，所以現在不用太著急囉。

原來如此，那我就放心了。



邊開發遊戲，邊學習演算法

接著透過遊戲的開發說明學習演算法與程式設計的意義。

》》》 透過遊戲開發學習的優點

本書會帶著大家一邊開發遊戲，一邊學習程式設計的技術與演算法。或許有人會覺得「為什麼要透過開發遊戲學習？」是因為透過遊戲開發學習演算法很有趣。

圖 1-4-1 快樂地學習



光是聽到演算法或程式設計這類字眼，可能有些人會覺得「很難」、「不知道該從哪裡開始」，但或許有許多人聽到開發遊戲會覺得「聽起來很難，但好像很有趣」、「如果學得會，想試試看」，對吧。

要製作很困難的遊戲當然需要更厲害的技術，這種程式設計的技術也不是一朝一夕就能學得會，不過，若是簡單的遊戲，就只需要學會程式設計的基技術。

本書會先帶著大家製作簡單的迷你遊戲，一邊熟悉程式設計的流程，一邊學習初階的演算法，再慢慢製作更困難的程式。最後則要挑戰翻牌配對遊戲的思考流程（人工智慧）。

快樂地學習能讓我們自然而然地學會演算法與程式設計的技術。大家都知道，讓人覺得快樂的事情才能持之以恆，而持續學習下去，程式設計的功力就會一步步提升。

》》》 初學者也能學得會的 Python

Python 的命令與語法都很單純，而且只要短短幾行程式就能完成處理。如果使用 Python，就能隨時寫幾行程式與確認執行過程。筆者的工作會用到 C、C++、C#、Java、JavaScript 以及其他的程式設計語言，但筆者確定在這些程式設計語言之中，Python 絕對是最適合初學者學習程式設計與演算法的程式設計語言。透過 Python 學會程式設計的基礎之後，也比較容易挑戰 C 語言或 Java 這類語言。在硬體與軟體都越來越複雜的資訊處理世界之中，許多人應該樂於見到 Python 這個容易學習的語言越來越受歡迎與普及。

真的是有趣就能持之以恆。
前輩也是樂在其中嗎？



我是從比較難的 Java 開始學，所以學得很痛苦。



原…原來是這樣啊（汗）
還好我的新人教育訓練是學 Python。



Python 的語法雖然簡單，
但要記的東西還是很多，
可不要掉以輕心喲！



了解了。



COLUMN

持之以恆，必有所成

筆者從小就很喜歡打電動，所以為了自己製作遊戲軟體而開始學習程式設計。一開始當然無法做出自己想要的遊戲。不過，當我一步步慢慢學之後，總算能做出簡單的迷你遊戲，而當我越學越久，也就能製作出更複雜的遊戲。在技術還不夠純熟的時候，我也會覺得自己怎麼連這種遊戲都寫不出來，但回想學習程式設計的過程之後，發現自己學得很快樂，這一切都是因為想要自己製作一個屬於自己的遊戲，我也覺得能抱著這樣的心情學習程式設計，真的很幸福。或許有些讀者學程式設計學得很辛苦，也或許有些讀者會覺得光是閱讀 Python 的入門書學不會，但請大家享受本書的內容。一如「持之以恆，必有所成」這句話，在學習的過程中，一定會有收穫，等到大家讀完本書，一定會覺得自己的技術成長了不少。

程式設計的準備①

— 顯示副檔名 —

接下來要完成撰寫程式的前置作業。第一步是顯示副檔名，才方便管理檔案，如果已經看得到副檔名，可直接跳過這一節，閱讀 Lesson 1-6 的內容。

》》》 何謂副檔名

副檔名就是接在檔案名稱後面，用來識別檔案類型的字串，檔案名稱與副檔名之間則是以點 (.) 間隔。

圖 1-5-1 檔案的副檔名

*****.py

檔案名稱 副檔名

例如，文字檔案的 txt、Word 文件的 docx 或 doc、圖片檔案的 bmp、png、jpeg 都是很常見的副檔名。

撰寫程式設計語言的程式的副檔名大致如下。

表 1-5-1 程式的副檔名種類

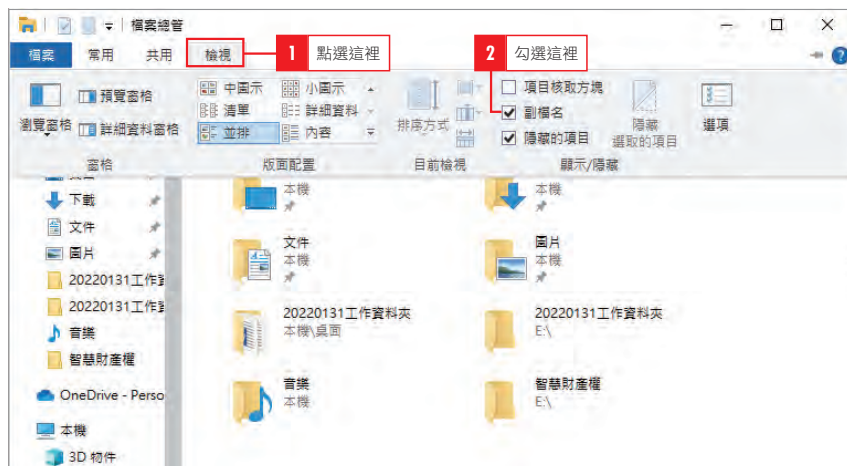
程式設計語言	副檔名
Python	py
C/C++	c、cpp
Java	java
JavaScript	js

使用 Windows 或 Mac 的讀者可分別透過下一頁的方式顯示副檔名。

》》》 在 Windows 的環境底下顯示副檔名

開啟資料夾，點選「檢視」，再勾選「副檔名」。

圖 1-5-2 在 Mac 的環境底下顯示副檔名



》》》 在 Mac 的環境底下顯示副檔名

選擇 Finder 的「偏好設定」，再於「進階」勾選「顯示所有檔案副檔名」。

圖 1-5-3 在 Mac 的環境底下顯示副檔名



要學習程式設計就必須顯示副檔名
啲！



程式設計的準備②

— 安裝 Python —

接著要安裝 Python。如果已經安裝了 Python，可直接跳到 Lesson 1-7。
在此說明在 Windows 與 Mac 環境安裝 Python 的方法。使用 Mac 的讀者可直接翻至第 25 頁再開始安裝。



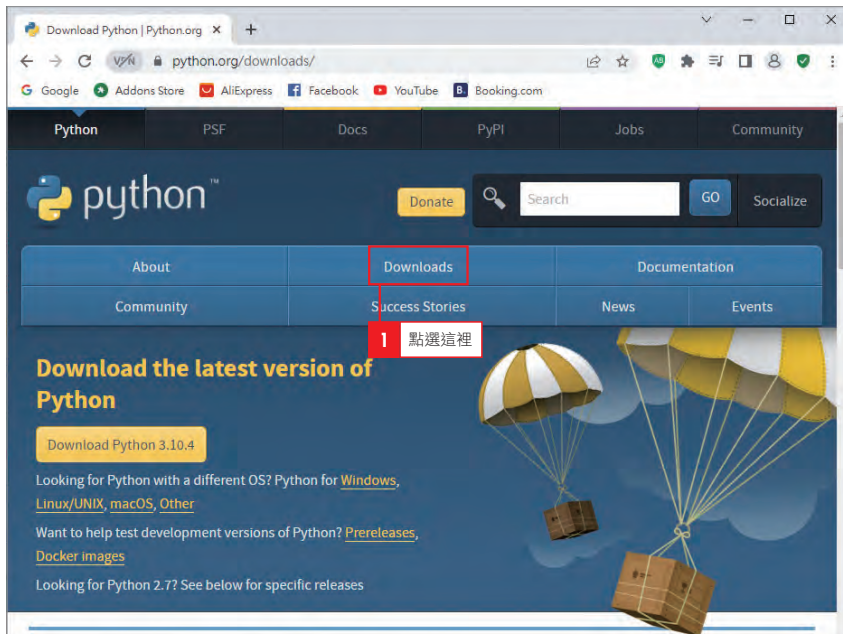
若從官方網站安裝 Python 就能立刻開始撰寫程式。

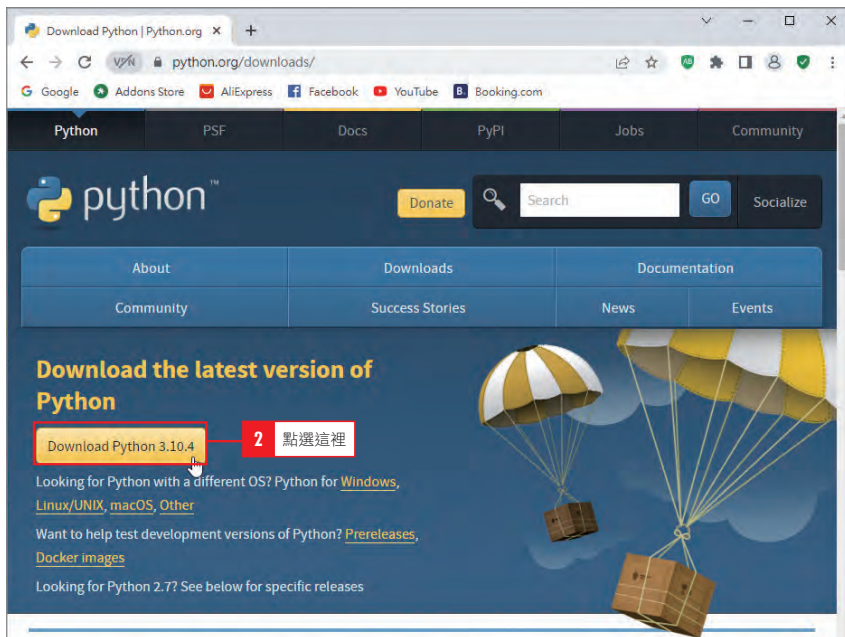
》》》 在 Windows 電腦安裝 Python

請先透過網頁瀏覽器瀏覽下列的網頁。

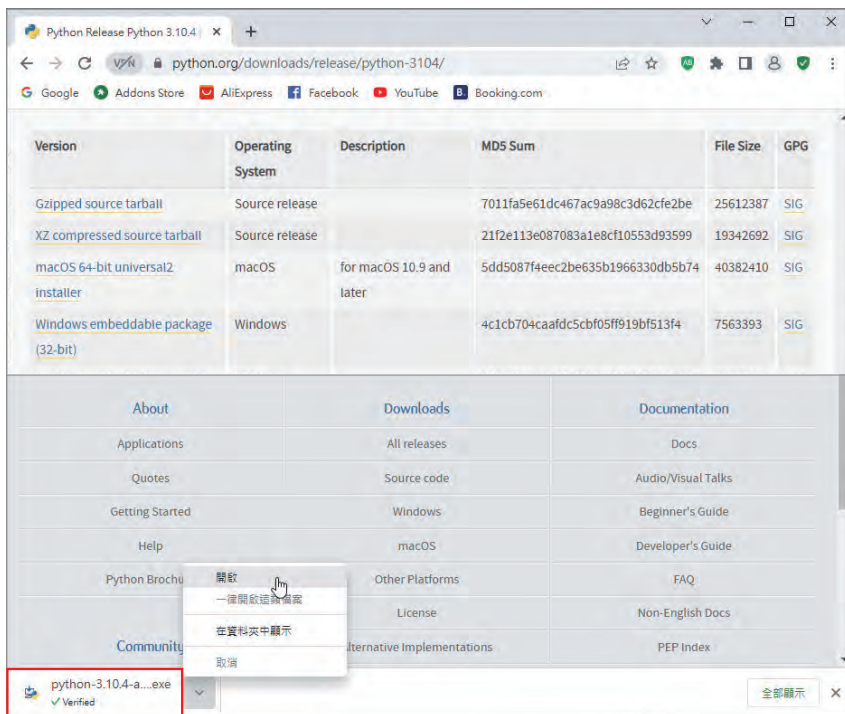
<https://www.python.org/>

點選「Downloads」，再點選「Windows」的「Python 3.*.*」按鈕。





點選「開啟」之後，就會開始安裝。

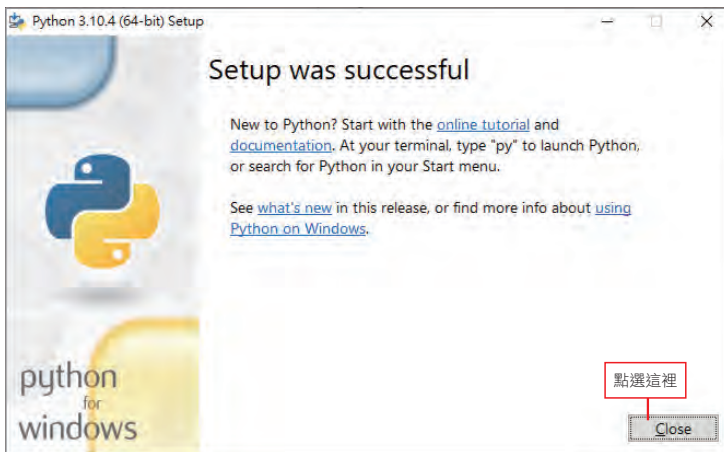


點選這裡，開始安裝。

勾選「Add Python 3.* to PATH」，再點選「Install Now」繼續安裝。



在「Setup was successful」畫面點選「Close」按鈕，完成安裝。

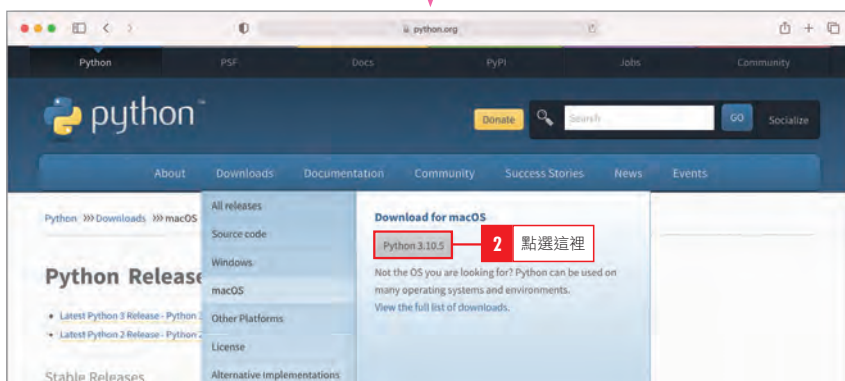
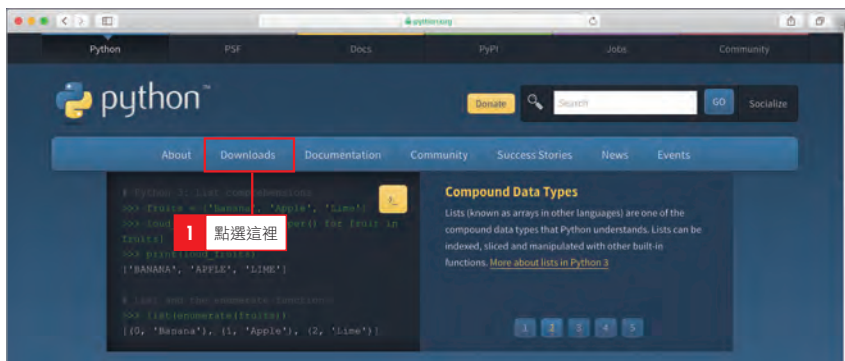


»» 在 Mac 環境安裝 Python

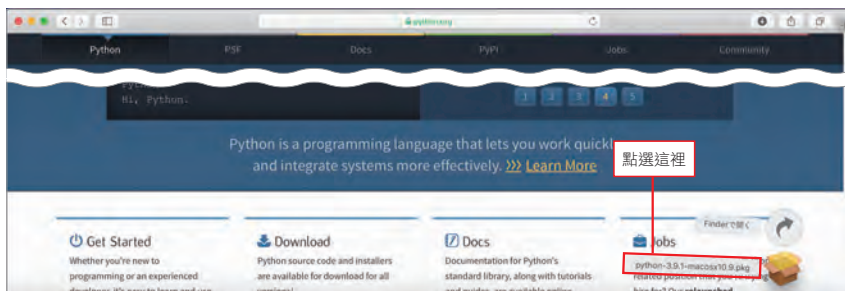
請利用網頁瀏覽器瀏覽下列的網址。

<https://www.python.org/>

請點選「Downloads」，再點選「macOS」的「Python 3.*.*」按鈕。



點選下載的「python-3.*.*-macosx***.pkg」。



點選「繼續」開始安裝。



點選「繼續」繼續安裝。



點選使用規範的「同意」，繼續安裝。



不需要自訂，繼續安裝即可。



看到「已成功完成安裝。」之後，點選「關閉」。

到此，安裝就完成了。

