

chapter 1

來一句 Python 文

/ 蔣宗哲、江政杰

學寫程式就像學一種語言，只是溝通的對象從人變成了電腦。本章將手把手帶領讀者初探寫程式是怎麼一回事，分為三小節，內容包括：環境建置與使用、資料運算（變數、資料型態和運算子）以及互動運算（內建函式之使用）。學習完本章後，讀者將在電腦上安裝好 **Python** 開發工具 **IDLE**，並可撰寫簡單的互動程式，從鍵盤輸入資料、進行簡易運作並在螢幕上顯示運算結果。

1-1 寫程式這回事

講到電腦，大家心裡想到的是甚麼呢？是很大台的超級電腦？一般桌上型電腦？還是工作使用的筆記型電腦呢？其實現代的電腦已經以各種不同的形態，廣泛的存在我們的身邊，最具有代表性的是你身上的智慧型手機了，手機內的各種應用程式幫助我們處理生活大小事，其他像平板、運動手環、穿戴裝置等，都是不同類型的電腦。

除了身邊用到的資訊裝置，現代社會中各行各業已經深刻地依賴電腦運作，尤其是近年來人工智慧的技術出現，讓資訊科技更快速地應用在社會運作的各個領域。大家或許已經發現，突然之間程式設計變得非常熱門，因為在可見的未來，人們將可透過程式設計的能力，直接控制身邊的資訊設備。透過程式設計的學習，能夠讓我們理解電腦實際運作的『思考』方式，在職場上更能夠增加跨領域的技能。

學習程式語言的目的是為了和電腦溝通，指揮電腦進行我們指定的運算流程。隨著人工智慧技術的進步，雖然人類已能使用自然語言指揮電腦（例：Siri），但其用途和規模仍有限。寫程式，指的是我們將需要電腦執行的運算流程，依循程式語言的文法規定撰寫出來。一般來說，我們會將程式存成純文字格式的檔案，這些程式檔案被稱為程式源碼 (source code)。不同的程式語言有不同的慣用副檔名，本書要談的 Python 程式語言，通常以 .py 作為副檔名 (如圖 1-1)。



圖 1-1 Python 程式源碼及文字內容示例

所謂學寫程式，其實是在學習如何運用程式語言來描述人類的思維，以期符合機器的運作方式，藉由機器的高效運算達成人類期望完成的工作。程式語言和一般我們認知的語言相同，都會有單字、句型、文法和語意等等需要學習。不過程式語言簡單多了，通常只要學會十來個單字，三五個句型，就能完成很多常見的工作。

簡單來說，寫程式包含三個基本步驟：撰寫程式源碼、翻譯成機器碼、以及執行機器碼。後面兩個步驟通常交由軟體來負責。我們可以使用任一種文字編輯軟體（如記事本也行）寫下程式源碼，再藉由程式開發工具來負責翻譯與執行。現今的程式語言都有整合性（整合編輯、翻譯和執行）的開發工具。這裡我們使用一種簡便的工具，稱為 IDLE。

安裝 Python IDLE 開發工具

讀者連上網址 <https://www.python.org/downloads/> 後，會看到一個下載的圖示，點擊圖示後，會在電腦中存下一個安裝檔。雙擊該安裝檔後，會跳出提示畫面。（此處以 Windows 安裝 Python 3.7.0 版本畫面為例，讀者若有較新的版本也無妨。）提示畫面中，請勾選最下面兩個小方塊，然後點擊 **Install Now** 即可。



圖 1-2 IDLE 整合開發工具下載頁面

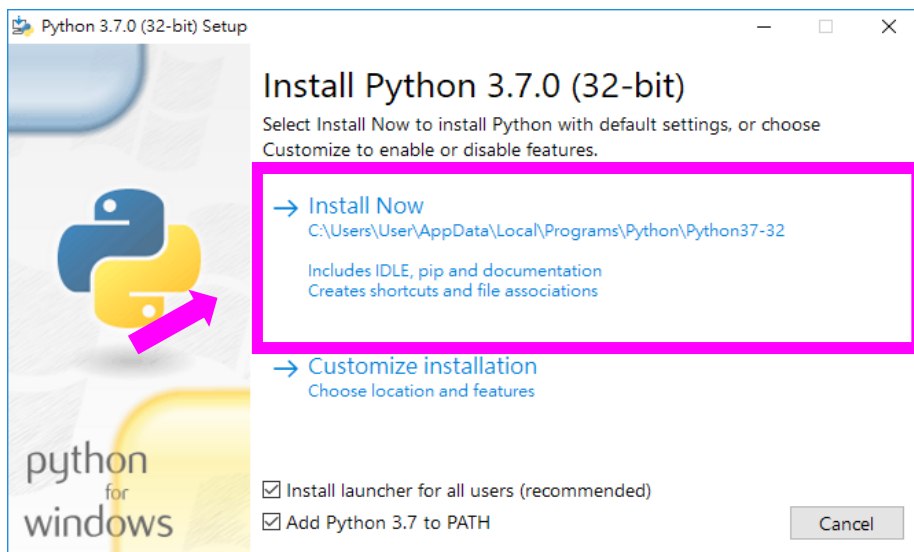


圖 1-3 IDLE 安裝畫面

安裝完畢後，找到 IDLE 軟體圖示，點擊以便執行。在 Windows 桌面的左下方有放大鏡，點擊放大鏡後，鍵入 idle 文字，應該會看到如下圖左邊的 IDLE 軟體圖示。點擊後，會開啟如下圖右邊的視窗，左上角有 Python ... Shell 文字，我們稱這個視窗為 Shell 視窗。

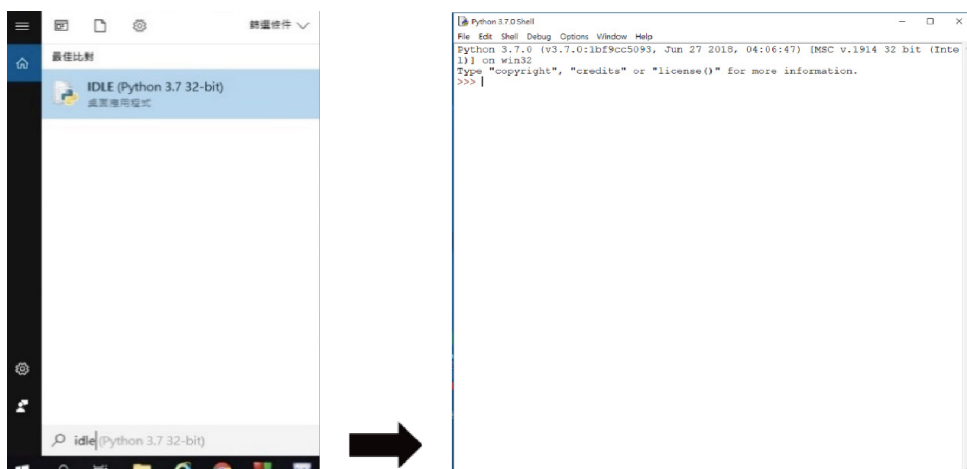


圖 1-4 Python 開發環境啟動範例

Hello World 程式

現在我們已經安裝好 Python 的程式開發環境了。前一節我們說過，程式源碼會以文字檔來儲存。我們首先開啟一個新檔案（這個動作就像平常使用文書軟體或簡報軟體一樣），請參見圖 1-5 的步驟 ❶。此時，我們會有一個新的空白視窗，我們稱此視窗為檔案視窗，在此編寫程式碼，如圖 1-5 的步驟 ❷。此處，我們先嘗試打上

```
print('Hello World')
```

這樣的文字。接著，儲存檔案，如步驟 ❸。此時，請指定一個目錄（例如桌面），然後把程式存為 _____.py，其中 _____ 可由讀者自行決定。儲存成功後，讀者應可以在指定的目錄中看到一個檔案，如步驟 ❹ 所示。接著，步驟 ❺，從功能列選擇 Run（執行程式的意思），再選擇 Run Module（或者按熱鍵 F5 一鍵搞定），此時若程式尚未存檔，會詢問是否存檔，如步驟 ❻。「確定」存檔後，便會在 Shell 視窗出現程式的執行結果了！

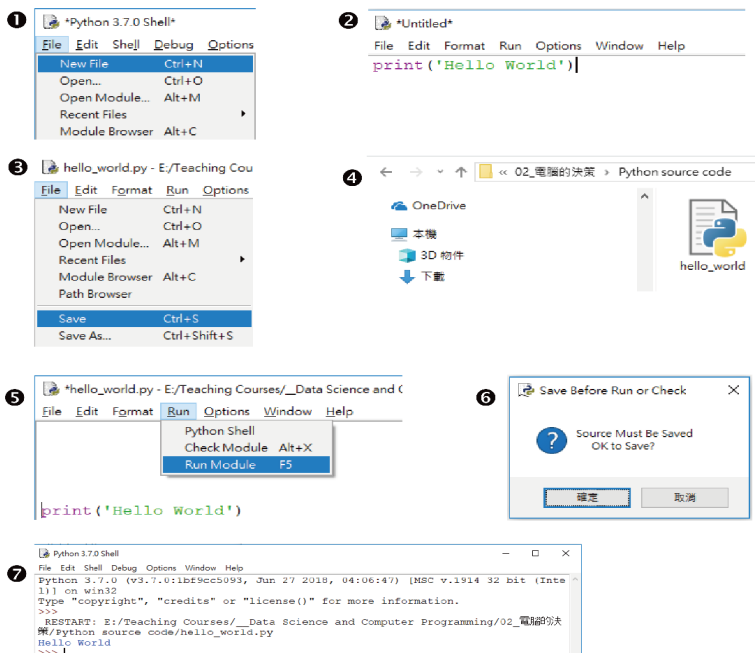


圖 1-5 Hello World 程式

隨堂練習

1.1.1 請撰寫一個程式，請寫個程式，幫你跟大家打招呼吧。

請顯示『大家好，我是 000』這裡的 000 請寫入你的名字。

執行結果：

大家好，我是 000

(參考解答)

```
01 print('大家好，我是 000')
```

1-2 資料運算

前一節讓 Python 跟大家打個招呼 say hello，引導讀者初步接觸 Python 的程式撰寫，現在我們來嘗試電腦最擅長的能力：運算。電腦最初的設計，是為了能夠快速進行繁複的科學運算，這也是電腦英文 **computer** 的由來。電腦面對大量繁複的運算時，不會累不會煩不會無聊不會抗議不會粗心，絕對是我們處理計算的好幫手。程式是直接控制電腦運作的媒介，所以學習程式首先會接觸的就是運算。

程式語言內可處理的運算種類很多，我們先來看看最常見的算數運算吧。表 1-1 列出六種運算範例、包括 Python 的運算子符號以及運算結果。舉例來說，在 Python 程式中，乘法是以星號 (*) 來表示；百分號 (%) 代表取餘數的運算，而連續兩個星號 (**) 則代表次方運算。

表 1-1：Python 的算術運算子

算術運算	Python 程式碼	運算結果
加	1+1	2
減	20-3	17
乘	3*7	21
除	100/8	12.5
取餘數	100%8	4 (100/8 = 12...4)
次方	4**2	16

我們可以直接在 **Shell** 視窗打入運算式，按下 **Enter** 鍵便可得到計算結果。如果要作的運算很多，我們可依前一節所學，將程式碼撰寫在一個源碼檔案中，然後再按下 **F5** 執行，最後在 **Shell** 視窗查看結果。

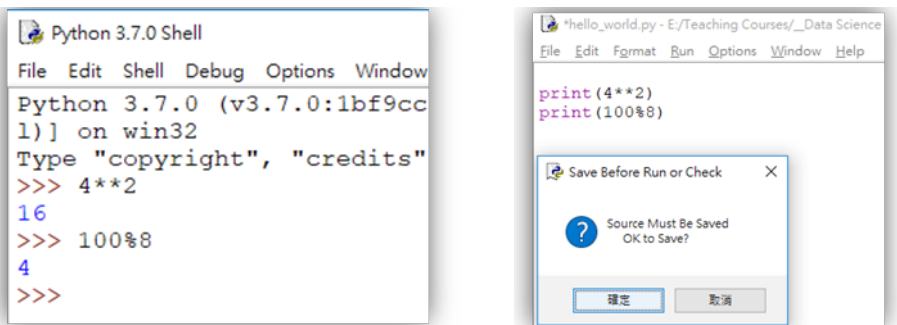


圖 1-6 撰寫 Python 程式進行算術運算

當我們同時使用多個算術運算，如 $1+10*3$ 同時用到加法和乘法運算時，我們需知道運算的順序。Python 程式語言訂定的順序符合我們的直覺，表 1-1 中的六個運算子，優先權最高的是次方，接著是乘、除、餘，最後是加減。同優先權的運算，則是由左至右計算。如果要改變既有的優先權，可以使用小括號來完成。注意，小時候我們學過的中括號 $[\]$ 和大括號 $\{ \}$ 在 Python 程式裡另有多用途，若有多層次，我們還是會持續使用小括號。

表 1-2：算術運算的優先次序

Python 程式碼	運算優先次序	運算結果
$1+10*3$	先算乘法，再算加法。	31
$20*2-4*3$	先算乘法，再算減法。	28
$3*4**2-1$	先算 $4**2$ ，再算 $3*16$ ，最後減 1。	47
$(1+2)*(3+7)$	先算加法（因為有小括號），再算乘法。	30
$((1+2)*3-1)*2$	先算 $1+2$ ，再算 $3*3-1$ ，最後乘以 2。	16

Python 的除法運算有兩種，一種是小數除法 (/)，另一種是整數除法 – 連續兩個斜線符號 (//)。整數除法較特別，計算結果是小於原除法結果的最大整數值（就是數學中的 floor 運算，一般記為 $\lfloor x \rfloor$ 。）另外，Python 中可將浮點數轉換成整數，以 int() 來操作，其中 int 取自整數 integer 的前三個字母。

表 1-3：Python 的除法運算

Python 程式碼	運算結果	說明
11/2	5.5	一般除法，得到小數結果
11//2	5	整數除法，得到整數結果 (較小的最靠近整數)
int(11/2)	5	強制轉型，得到整數結果 (無條件捨去小數部位)
-11/2	-5.5	一般除法，得到小數結果
-11//2	-6	整數除法，得到整數結果 (較小的最靠近整數)
int(-11/2)	-5	強制轉型，得到整數結果 (無條件捨去小數部位)

隨堂練習

1.2.1. 請使用 Python 執行下列的運算，執行前先用手計算看看，結果跟你想的是否相同？

- (1) 20+1/3
- (2) 20/3+1
- (3) 20/(3+1)
- (4) 20+1//3
- (5) 20//3+1
- (6) 20//(3+1)
- (7) 2*17%5*2
- (8) 17/2%5
- (9) -17%5
- (10) '12'*3

1-3 互動運算

前一節我們學會了以 Python 程式進行算術運算，但，運算的數值從哪裡來呢？我們只能計算固定的數值嗎？當然不是囉。本節我們將先介紹程式中最重要的元素 – 變數與型態，然後教大家如何讓程式獲得使用者的輸入，最後以一個簡單的互動程式作結尾。

變數

程式中的資料都存在記憶體中，現在的電腦多半有好幾 GB 的記憶體，也就是好幾十億個位元組的記憶體。一個整數大概花費 4 或 8 個位元組，所以我們的記憶體可以存放很多很多的整數資料。那麼，為了讓我們方便取用與指稱某一段記憶體中的數值，我們以一個有意義的名稱來代表，這就是變數的概念。

以下面的 Python 程式碼來說明。我們以 = 運算來生成一個變數並指派它內容。第 01 行的意思程式執行時會向電腦要求一（小）段記憶體空間，裡面存放整數值 10，並且將這段空間命名為 x。此後，當我們在程式中寫到 x 的時候，就代表著那一段記憶體。因此，程式第 02 行是利用 print() 函式來列印 x 所代表的記憶體的內容。由於第一行將 10 存入這段記憶體，所以第二行程式會列印出 10。第 03 行我們又定義了變數 y，令其代表 20。第 04 行示範 print() 可以一次印出多個變數、甚至是運算 x+y 的結果。多個列印結果之間預設以一個空白字元隔開。

01	x = 10
02	print(x)
03	y = 20
04	print(x, y, x+y)

執行結果：

10
10 20 30

資料型態

到目前為止，我們都在進行整數或小數型態的運算。在程式中，另一種常見的資料型態是文字型態。Python 中稱為字串 `str` 型態，其中 `str` 是字串 `string` 的前三個字母。在程式中，`10` 是整數型態、`10.0` 是小數型態，而 `'10'` 則是字串型態。Python 中以一對單引號或雙引號包起來的都視為字串型態。

了解資料型態對於撰寫程式十分重要，因為不同型態的資料可能允許不同的運算；即使允許同樣的運算，其運算的意義與結果也可能不同。表 1-4 比較了對整數和字串進行乘法運算與加法運算的結果。整數的部份就不贅述，字串對整數的乘法是複製的意思，如 `'12'*3` 的結果是 `'12'` 重複 3 次；字串對字串的加法是串接的意思，如 `'12'+'3'` 的結果是兩個字串串接起來，形成 `'123'`。

表 1-4：Python 的算術運算子

Python 程式碼	運算結果	說明
<code>12*3</code>	36	整數乘法，得到整數結果
<code>'12'*3</code>	<code>'121212'</code>	字串乘法，得到字串結果
<code>'12'*'2'</code>	失敗	字串不支援和字串相乘，翻譯失敗
<code>12+3</code>	15	整數加法，得到整數結果
<code>'12'+3</code>	失敗	字串不支援和整數相加，翻譯失敗
<code>'12'+'3'</code>	<code>'123'</code>	字串加法，得到字串結果

讀入資料

`print()` 函式可以用來輸出資料給程式的使用者看，那要如何讓使用者輸入資料呢？啊哈，輸入的英文是 `input`，Python 正是以 `input()` 函式來取得使用者的輸入資料。以下兩行程式，第 01 行以 `input()` 函式取得使用者輸入的文字，並以 `x` 指稱這段字串資料。第 02 行將 `x` 指稱的資料印出來。建議讀者立刻將這兩行程式鍵入檔案視窗中，按下 **F5**，體驗程式如何與使用者互動。按下 **F5** 後，**Shell** 視窗會有一個游標閃爍，此時讀者可嘗試輸入 **198**，應會看到程式回應了同樣的輸入文字 **198**。

01	<code>x = input()</code>
02	<code>print(x)</code>

執行結果 1：	執行結果 2：
198	Python is fun.
198	Python is fun.

當讀者按 **F5** 執行上述兩行程式時，可能會感到疑惑，因為 **Shell** 視窗中沒有任何提示訊息，讓人不知所措。我們可以在 `input()` 函式中增加一段提示訊息，這段訊息是文字型態，因此以一對單引號包起來。讀者可嘗試鍵入並執行下列程式碼，以了解 `input()` 和 `print()` 中列印文字的方式。

01	<code>x = input(' 請輸入一個整數值 > ')</code>
02	<code>print(' 你輸入的是 ', x)</code>

執行結果：
請輸入一個整數值 > 198 你輸入的是 198

相信有讀者已經迫不及待要對 `input()` 取得的使用者資料進行運算，但卻對運算結果感到疑惑。以下面程式為例，當使用者輸入 **198**，原先我們預期會得到 **396** ($198 \times 2 = 396$) 的結果，但是程式卻輸出 **198198**。聰明的讀者應該會想起前一頁我們提到字串與整數相除的運算。是的，此處 `x` 代表的是字串資料 `'198'` 而非整數資料 `198`。這是因為 `input()` 函式所帶回的資料是字串型態所致。

01	<code>x = input(' 請輸入一個整數值 > ')</code>
02	<code>print(' 此數乘以 2 的結果是 ', x*2)</code>

執行結果：

請輸入一個整數值 > 198
此數乘以 2 的結果是 198198

這該如何是好？不用擔心，稍早在表 1-3 我們曾提到 `int()` 可以將資料轉換成整數型態（同理，`str()` 可以把資料轉換成字串型態。），所以我們可以將程式修改如下。我們新增了第 02 行程式，這行的意思是取出 `x` 所代表的資料，將該資料（由字串型態）轉為整數型態，然後再令 `x` 代表這個新的資料。此例中，第 02 行會令 `x` 代表的資料由 `'198'` 變成 `198`。因此第 03 行的 `x*2` 便會進行整數運算，得到我們期待的結果。

01	<code>x = input(' 請輸入一個整數值 > ')</code>
02	<code>x = int(x)</code>
03	<code>print(' 此數乘以 2 的結果是 ', x*2)</code>

執行結果：

請輸入一個整數值 > 198
此數乘以 2 的結果是 396

隨堂練習

1.3.1 前一個程式跟大家打招呼後，你的名字到哪裡了呢？

延續前一個練習，請先用變數 `name` 放入你的名字再打招呼

請顯示『大家好，我是 000』這裡的 000 請寫入你的名字

執行結果：

大家好，我是 000

(參考解答)

```
01 name = '000'
02 print('大家好，我是 ', name)
```

1.3.2 寫個程式，用變數 `yy`, `mm`, `dd`, `day` 分別記錄年月日與星期幾，然後顯示出來。

執行結果：

今天是 2020 年 5 月 5 日 Tuesday

(參考解答)

```
01 yy = 2020
02 mm = 5
03 dd = 5
04 day = 'Tuesday'
05 print('今天是 ', yy, ' 年 ', mm, ' 月 ', dd, ' 日 ', day)
```

1.3.3 請撰寫一個程式，首先列印出程式功能「加法程式」，接著提示使用者輸入兩個整數值，最後列印出此二整數值的和。

執行結果：

加法程式
請輸入第一個整數值 > 33
請輸入第二個整數值 > 44
答案是 77

(參考解答)

```
01 print('加法程式')
02 x = input('請輸入第一個整數值 > ')
03 y = input('請輸入第二個整數值 > ')
04 total = int(x)+int(y)
05 print('答案是 ', total)
```