

# Chapter 2：程式基礎

## 本章重點

認識程式的物件、資料型別、變數、內建函式、匯入模組，這些是學程式最基本的概念。

## 準備材料



可上網的電腦

## 學習目標

1. 了解程式語言的相同與差異。
2. 認識程式最重要的基本名詞。

## 2-1. Hello, World! 最像英文的 Python

大學的資訊系有一門必修課，名稱為程式語言 (programming language)，主要內容是介紹各種程式的相同、差異與優劣比較。資訊系不會學習特定的語言，因為程式的語言太多了，每一種專業領域，例如：人工智慧、網際網路、資料庫等等，都有不同的程式來實作。但程式的語法邏輯都差不多。以下介紹前四大的程式，下面的程式都是在螢幕上顯示出字串 'Hello World!'。

Python	C
<pre># 印出 Hello World! 字串物件 print('Hello World!')</pre>	<pre>/* 印出 Hello World! 字串物件*/ #include &lt;stdio.h&gt; int main() {     printf("Hello, World!\n");     return 0; }</pre>
C++	Java
<pre>//印出 Hello World! 字串物件 #include &lt;iostream&gt; using namespace std; int main() {     cout &lt;&lt; "Hello World" &lt;&lt; endl;     return 0; }</pre>	<pre>//印出 Hello World! 字串物件 public class HelloWorld{     public static void main(String []args){         System.out.println("Hello World");     } }</pre>

四個程式中，藍色字是程式的主要功能：在螢幕上顯示一段文字的程式碼。由此可知語法結構都差不多，其中 Python 最像英文的語法。因此，先學好 Python，再轉換語法到各種程式語言，是一個不錯的入門學習。

## 2-2. 物件、資料型別、變數、內建函式、匯入模組

### 物件

一般寫英文句子時，會以名詞 + 動詞。Python 是以物件.方法來描述。

文章寫作	Python 編程	程式的解釋
車子	car	car 物件
車子發動	car.start()	car 物件的 start 方法

方法後面會加上括號()，有些方法需要加入額外的參數。

- 例如：car.go(100)，車子加速到 100。

若方法有多個參數，以逗點分隔。

- 例如：car.left(50, 30)，以 50 的速度，向左轉 30 度。

### 練習：操作字串物件的內建方法

在 IPython 中，輸入下列敘述，字串物件以雙引號 (") 或單引號 (') 來表示：(>>> 指的是在互動模式中，執行單行敘述。)

```
>>> 'Hello World!'.upper() —— 使用字串物件 'Hello World!' 的 upper()
'HELLO WORLD!'                方法，將字串轉成大寫
>>> 'Hello World!'.find('r') —— find() 方法尋找 'r' 的位置 (從 0 開始)
8
>>> 'Hello World!'.replace('r', 'u') —— replace() 方法將所有的 'r'
'Hello Would!'                 取代成 'u'
```

- 不同的物件會有不同的方法。例如：字串物件與整數物件 1, 2, 3, ...，字串物件有內建方法，而整數物件沒有任何的內建方法。
- 不同方法會回傳相同或不同的物件。例如：字串的 upper() 與 replace() 回傳字串，find() 回傳整數。

## 資料型別

除了字串 (str)，寫程式常有整數 (int) 與浮點數 (float) 物件，例如：111 與 11.1。

```
>>> 111 + 111  —— 整數物件相加，結果為整數
222
>>> '111' + '111' —— 字串物件串聯，結果為字串
'111111'
```

上述 + 的運算，因物件的資料不同而產生不同的結果。物件的種類，程式語言稱之為『物件型態』 (object type) 或『資料型態』 (data type)。

### 練習：要分清楚資料型別

兩個資料型別若不同，可能會導致程式錯誤。

```
>>> 111 + '111' —— 不同型別的資料相加，發生錯誤
Traceback (most recent call last):
  File "<ipython-input-6-4832c22160be>", line 1, in <module>
    111 + "111"
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

## 變數

『變數』 (variable) 就像是掛在物件的名牌，幫物件取名之後，讓我們方便識別物件與操作，其語法為：

```
變數名稱 = 物件
```

變數通常用來儲存暫時的資料，例如：計算成績的系統，會宣告多個變數儲存國文、英文、數學等科目的成績。例如：

```
>>> n1 = 123456789  ————— 將整數物件 123456789 指派給變數 n1
>>> n2 = 987654321  ————— 將整數物件 987654321 指派給變數 n2
>>> n1 + n2
1111111110
```

變數是物件的名牌，一定會有資料型態。可用 `type`（變數名稱）檢查其資料型態。例如：

```
>>> a = 123
>>> type(a)
int  —————  int  為整數型別
>>> b = '123'
>>> type(b)
str  —————  str  為字串型別
```

## 內建函式

『**函式**』（function）是一段預先寫好的程式，方便重複使用。而程式先將經常使用到的功能以函式的形式先寫好，稱為『**內建函式**』。

- 例如：`print()` 是最常用的顯示函數。
- 程式也可以『**自建函式**』。

```
>>> print('abc')  ————— 顯示字串物件
abc
>>> print('abc'.upper())  ——— 顯示字串物件.方法的執行結果
ABC
>>> print(111 + 111)  ————— 顯示整數物件運算的結果
222
```

## 匯入模組

內建函式不就越多越好？若內建函式無限制增加，會導致啟動速度越來越慢，執行時佔用的記憶體越來越多。『**模組**』 (module)，就是將同一類的函式打包成模組，預設不會啟用。需要時，再用**匯入** (import) 的方式啟用。程式預先寫好的稱為『**內建模組**』。

```
>>> import time —— 匯入時間相關的 time 模組
>>> time.sleep(3) —— 執行 time 模組的 sleep() 函式，暫停 3 秒
```

**TIP** 函式 sleep() 相對於模組 time，也稱為物件 time 的 sleep 方法。

而只匯入模組內特定的函式，會比較節省記憶體：

```
>>> from time import sleep —— 從 time 模組裡匯入 sleep() 函式
>>> sleep(3) —— 執行 sleep() 函式，暫停 3 秒
```

### 練習：匯入模組

**2-1** 暫停 3 秒後，印出 'Hello World!'。(在程式編輯窗格中編輯)

```
# 暫停 3 秒後，印出 Hello World!
from time import sleep

sleep(3)
print('Hello World!')
```

**TIP** 『#』為 python 的註解符號，程式執行時『#』後面的程式碼不會執行。

## 結論

各種程式語言的語法邏輯都差不多，就像人類語言的文法。

1. 程式的每一個東西都是**物件**，有些物件有其特定的操作方法。基本的物件有字串 (str)、整數 (int)、浮點數 (float)。
2. 基本物件有**資料型別**，型別不同，結果不同。甚至有時會錯誤。
3. 程式主要是操作物件，**變數**是物件的名牌而已，也方便程式設計師操作。
4. **內建函式**是經常用的函式，預先寫好的。
5. 將同一類的函式打包起來，稱為**模組**，適當的**匯入模組**，能精簡效能。

## 練習 2-2

1. 寫出以下的程式功能：每秒出現一行如下的星號。例如：第一行出現 1 個星號，第二行出現 2 個星號，依此類推。

```
*  
**  
***  
****  
*****
```