

程式語言與 Python 程式語言簡介

蔡尚融

2018-02-26

程式 (Program)

- 程式是由一連串的指令 (instruction) 構成，用以達成既定目標計算的工具。
- 一般程式由各類不盡相似的程式語言 (programming language) 編寫。
- 通用程式語言 (general-purpose programming language)，其基本指令涵蓋輸入 (input)、輸出 (output)、算術 (arithmetics)、條件執行 (conditional execution)、重複 (repetition)。
- 演算法 (algorithm) 是構成程式另一個重要的部份。

程式語言

- 依人類容易理解程度，可分為低階語言（low-level language）與高階語言（high-level language）。
- 低階語言包含機械語言（machine language）與組合語言（assembly language）。
 - ▶ 不同硬體架構（architecture）對應不同機械語言。
 - ▶ 電腦（computer）硬體只能執行機械語言。
- 高階語言優缺點：
 - ▶ 優點：花費較少時間撰寫，程式碼較簡潔，容易閱讀。
 - ▶ 缺點：需要額外處理程序，可能花費較長時間執行。
- 一般高階語言，分為兩種處理方式：
 - ▶ 由編譯器（compiler）先轉成機械碼程式，再另外執行。
 - ▶ 由直譯器（interpreter）翻譯成對應機械碼，並直接執行。

程式開發與使用的階段

1. 需求 (requirements)
2. 設計 (design)
3. 實作 (implementation)
4. 確認 (validation)
5. 發行 (release)
6. 修補 (patch)

除錯 (Debugging)

- 語法錯誤 (syntax error)

程式語法有誤，編譯器或解譯器無法分析語意。

例：`print('This is a text)` (錯)

`print('This is a text')` (對)

- 執行期錯誤 (runtime error)

語法正確，但是程式不能正常執行。

- 語意錯誤 (semantic error)

誤解程式實做目標，雖然語法正確，程式能正常執行，但是無法提供正確或要求的執行結果。

Python 程式語言

- Python 是直譯式物件導向（Object-oriented）語言。
- 目前可用的有 Python 2 與 Python 3 兩版本；課程以 Python 3 的語法為標準。
- 文字處理以 UTF-8 為預設編碼。
- 操作模式可分為，即時的互動模式（interactive mode）與檔案的腳本模式（script mode）。

互動模式

- 互動模式啟動方式為，在虛擬終端機中鍵入 `python3` 或 `python2` 執行 Python 解譯器。
- 螢幕顯示 `>>>` 提示 (prompt) 即為準備就緒。
- 鍵入 `exit()` 或 `quit()` 按下返回鍵 (Enter) 後，即可離開 Python 互動模式。
- 提供線上說明文件查詢。

互動模式下的說明文件

- 使用 `help()` 函式，查詢某些主題說明。

```
>>> help(print)
```

- 某些主題說明須引入指定模組。

```
import math
```

```
>>> help(math)
```

```
>>> help(math.cos)
```


腳本模式

- 用以儲存程式碼，以供重複利用。

- 通常使用副檔名.py。

- 第一行為

```
#!/usr/bin/env python
```

- 註解：在一行上，以 # 符號開始的程式碼不會被執行；通常用於說明程式運作等方面。

```
#!/usr/bin/env python
```

```
import math
```

```
# Compute sum of one and square root of two
```

```
1 + math.sqrt(2)
```

```
n = 5 # assign 5 to n
```

執行腳本

- Shell 中以 Python 直譯器引數輸入方式執行程式。

\$ python3 腳本檔案路徑

(註：\$ 為 Shell 提示符號，請不要鍵入。)

- 修改腳本檔案權限為可執行；用 `chmod` 指令修改權限。

隨堂練習：

1. 進入互動模式，並計算下列式：

1.1 $3 + 9 * 5 / 2$

```
>>> 3 + 9 * 5 / 2
```

```
25.5
```

1.2 $(3 + 9) * 5 / 2$

```
>>> (3 + 9) * 5 / 2
```

```
30.0
```

2. 結束互動模式。
3. 將題 1. 算式寫入一個名為 `ex_demo.py` 的腳本檔案。

以下列任一方式執行：

```
$ python3 ex_demo.py
```

或

```
$ chmod +x ex_demo.py
```

```
$ ./ex_demo.py
```