

## 12-2 程式語言簡介

**程式語言** (programming language) 是指使用者用來與電腦溝通之文字記號所形成的集合，也就是電腦能夠接受的語言，一旦我們找到解決某種問題的演算法，下一步便是以合適的程式語言將之表達為電腦能夠接受的形式，並藉此指揮電腦處理資料或解決問題。就好像中文、英文、法文或西班牙文...等一般，程式語言的種類繁多且各有其特定的文法（或稱語法）結構與適用範圍。雖然電腦其實只能理解某種由 0 和 1 組成的語言形式，但透過各種程式翻譯器，不同的程式語言最終都會被翻譯為相同形式，也就是稍後會介紹的機器語言。

### 12-2.1 低階語言

**低階語言** (low-level language) 指的是會隨著使用的 CPU 架構不同（如 X86 或 Mac）而有顯著的差異的語言，包括機器語言與組合語言，其優點是執行速度快，編寫程式較困難且程式碼不易理解。

#### **第一代語言：機器語言**

**機器語言** (machine language) 是電腦所能處理最直接的語言，不同處理器類型的電腦只適用一種特別為該處理器之指令集所設計的機器語言。所有的程式在執行前都必須被轉換成機器語言。

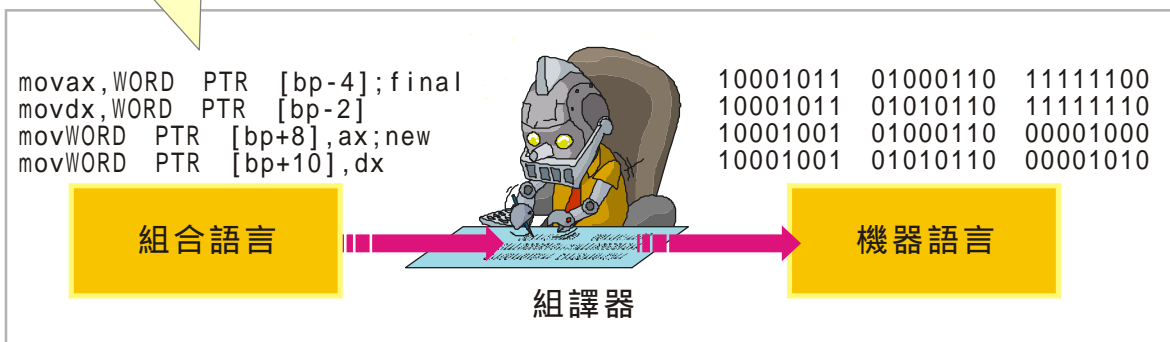
### ✓ 第二代語言：組合語言

由於機器語言對程式設計師來說並不方便，因此後來設計出一種使用英文簡寫來代表各種基本運算的語言。例如以 ADD 代表「加」，以 SUB 代表「減」；在機器語言中的「加」及「減」指令分別是由一長串的 0 與 1 所組成的數字。這些以英文簡寫所構成的語言叫做**組合語言**（assembly language）。

組合語言必須先被翻譯成機器語言後才能被電腦接受，擔任這個翻譯工作的程式叫做**組譯器**（assembler），如圖 12-2.1。

圖 12-2.1

組譯器運作示意圖。



### 12-2.2 高階語言

**高階語言**（high-level language）的語法已經接近人類日常生活用語，簡單易懂，在高階語言中，一個命令就可以代表數個組合語言中的命令。

### ✓ 第三代語言：程序式語言

**程序式語言**（procedural language）將重心放在解決問題的程式，設計師必須仔細思考各步驟的細節並進行控制，例如 FORTRAN、COBOL、BASIC、C、Pascal... 等都是程序式語言的代表例子。

## 延伸學習



### 常見的第三代語言

雖然程式語言已經發展到第五代，但是第三代語言（高階語言）的嚴謹架構及富有彈性的設計能力仍使它極受歡迎與重視，簡單介紹常見者如下表 A。

■ 表A 常見的第三代語言。

程式語言	簡介與說明
FORTRAN	處理數值計算的能力特別強，主要用於科學與工程應用。
COBOL	1960年制定的通用商業語言，語法非常類似英文句子。
ALGOL	演算法語言，在學術界、國際刊物常用來表達演算法。
BASIC 與 VB	BASIC採用直譯方式，是適合初學者的程式語言，後來的 Visual Basic 則融入視覺化開發環境與物件導向觀念。
PASCAL 與 Delphi	PASCAL 具有良好的結構化程式設計觀念，因此常被用來作為教學語言，後來則發展為具物件導向特性的 Delphi 語言。
C 與 C++	C 語言被廣泛地用來發展系統軟體或作業系統（如 Unix）。C 語言後來被加強成為 C++ 語言，增加了物件導向的特性。
Java	Java 源自於 C++，但是除了物件導向，它還提供分散式資料處理及跨平台的能力，是網際網路興起後相當熱門的程式語言，最大的特色為遠端使用者可透過瀏覽器執行 Java 程式。

另外，許多人開始學習運用簡單的語法與邏輯概念來控制或增強應用軟體的功能，網際網路的風行也帶動標記和文稿語言的發展，如表 B。

■ 表B 常見的應用軟體語言與標記語言。

類 型	簡介與說明	代表例子
應用軟體語言	應用軟體中提供的控制語言，藉以加強軟體功能。	VBA (內建於Office軟體) ActionScript (內建於Flash)
標記與文稿語言	與網頁設計相關的程式語言，多具備直譯特性。	HTML、XML、JavaScript、VB Script、Perl...

### ✓ 第四代語言：資料庫查詢語言

第四代語言 (4th Generation Language, 簡稱 4GL) 是使用者導向的語言, 它和程序式語言最大的不同, 在於它是以程式的「目的」為設計重心, 而非如第三代語言以「過程」為法則, 因此「要做什麼?」比「如何做?」更重要, 程式設計師只需寫出「要做什麼?」, 無需瞭解電腦是如何去執行的, 例如:

```
COUNT customer WHERE city EQUALS Taipei AND years EQUALS 2004
```

'計算2004年台北市的客戶數目

4GL 比第三代語言缺少了控制性及設計彈性, 執行速度也比較慢, 然而其易學易用的特性畢竟非第三代語言所能比擬, 因此至今在資料庫處理的領域裡, 4GL 仍然保有一片天空, 例如結構化查詢語言 (structured query language, SQL) 就是典型的代表實例。

### ✓ 第五代語言：人工智慧語言

第五代程式語言的發展目標為利用人工智慧技術, 使其接近人類使用的自然語言 (如英語), 也就是對於同一件事情可用多種不同的敘述方式來表達, 且能容忍少量的拼字及語法錯誤, 就有如我們在和電腦對談一般, 例如:

Tell Me the telephone number of KZ.

I don't know what you mean about KZ.

Kevin Zheng.

2987-6543.

目前人工智慧語言仍處於發展中的階段，現今開發的產品則多是以 LISP 或 Prolog 語言為基礎，專為特定領域設計的知識庫系統（如醫療、財經），或是智慧型的資料庫存取工具。

✓ 高階語言的翻譯程式

電腦不能直接接受以高階語言所寫成的程式，必須先轉換成機器語言後才能執行，而依轉換工具不同可分為直譯器 (interpreter) 及編譯器 (compiler) 二種，如圖 12-2.2。

圖 12-2.2  
直譯器及編譯器。

