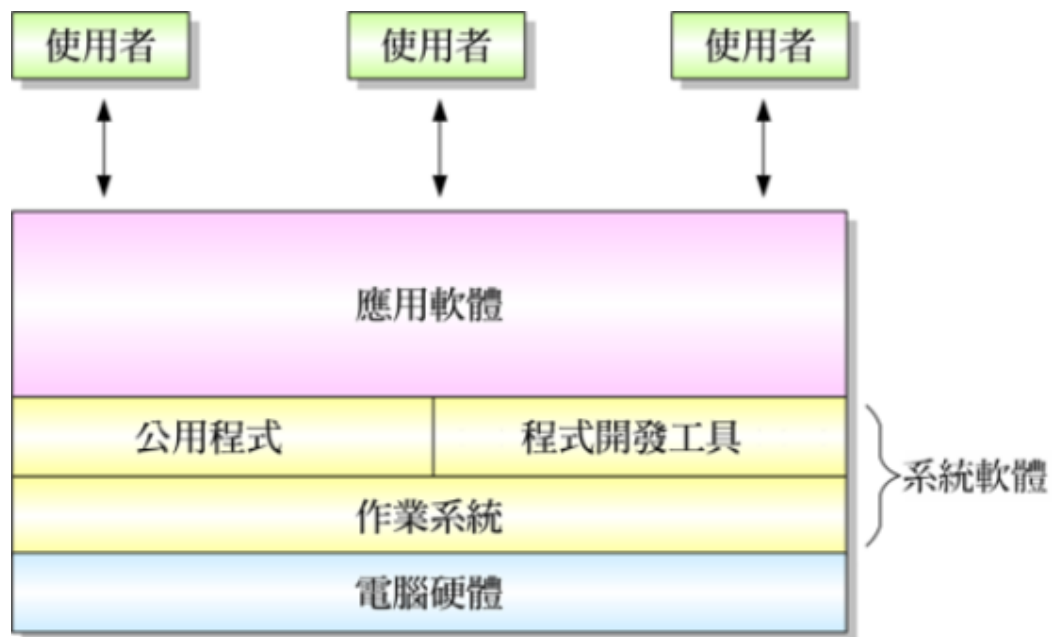


入門概論（一）：認識電腦

2-2 電腦軟體

軟體是許多指令的集合，可以指揮電腦硬體運作，用來解決我們的問題，而這些指令的集合就是程式。

一般將軟體分成系統軟體和應用軟體兩大類。



系統軟體 (system software)：管理電腦內各種硬體單元必備的程式，負責做硬體與軟體間溝通的橋樑，支援電腦運作的程式，包括作業系統 (operating system)、公用程式 (utility)、程式開發工具 (program development tool)。

- 作業系統是介於電腦硬體與應用軟體之間的程式，除了提供執行應用軟體的環境，還負責分配系統資源。
- 公用程式是用來管理電腦資源的程式。
- 程式開發工具是協助程式設計人員開發應用軟體的工具，包括文字編輯器等。

應用軟體 (application software)：針對特定事務或工作所撰寫的程式，是因應用戶需求而設計的程式，來完成特定的工作，目的是協助使用解決問題。



Windows作業系統屬於系統軟體



Word文書處理軟體屬於應用軟體

作業系統

作業系統是管理電腦硬體與軟體資源的系統軟體，同時也是電腦系統的核心與基石。作業系統層分為三層：



(1) 啟動層Bootstrap = 火星塞

開機程式是系統啟動後的第一個執行的程式，可將硬體（CPU與記憶體）初始化，讀進作業系統，直到可以迎接核心層的程度。

(2) 核心層Kernel

核心是一個電腦程式，進行的是應用軟體和電腦硬體的互動工作。核心層下管硬體層，任何程式要用硬體都得跟核心層申請。核心層上管程式層，負責監督、協調與控制作業系統中的處理程式，讓所有執行中的程式和平相處正常運作。

(3) 介面層Shell

一個接受並處理使用者的指令，讓使用者與系統互動的操作介面。

- 命令列介面(CLI, command-line interface)：用鍵盤在命令列上鍵入文字指令。

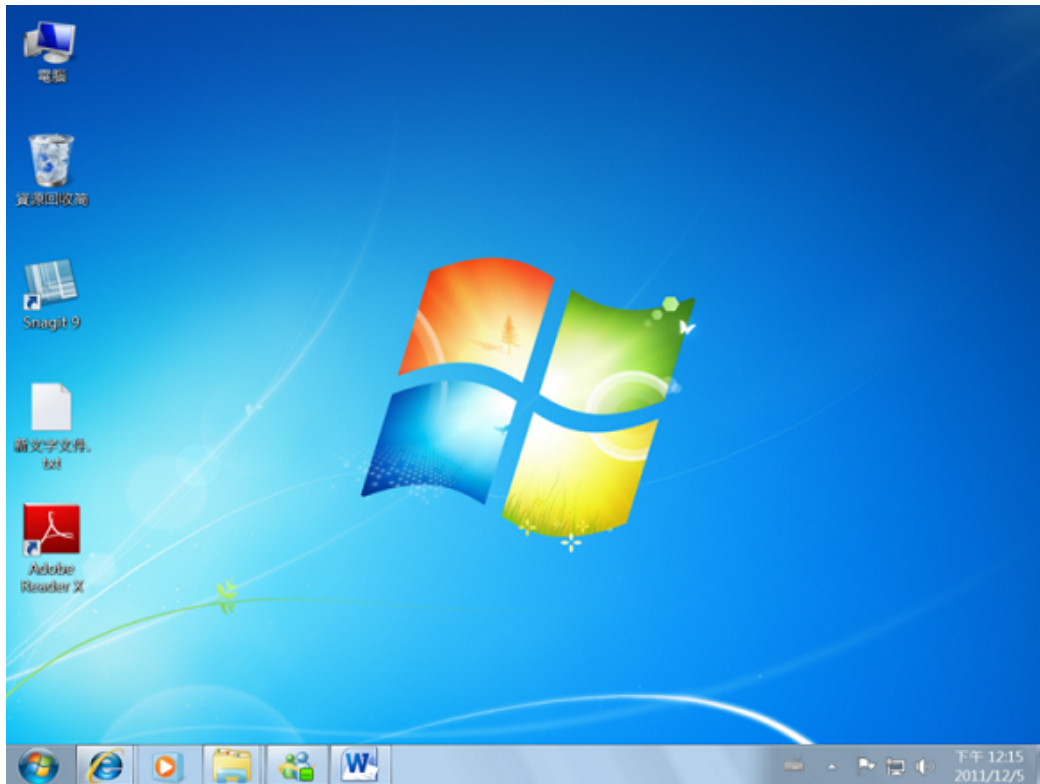
```
C:\>dir
磁碟區 C 中的磁碟是 SYSTEM
磁碟區序號: 58F0-41D5

目錄: C:\

2004/07/26  11:09a    <DIR>          WINNT
2004/07/26  11:11a    <DIR>          Documents and Settings
2004/07/26  11:12a    <DIR>          Program Files
2004/07/30  01:25p    <DIR>          TMP
2004/09/04  02:59p    <DIR>          WUTemp
2004/09/13  02:43p    <DIR>          php
2005/01/14  01:24a    <DIR>          downloads
                0 個檔案              0 位元組
                7 個目錄    48,616,669,184 位元組可用

C:\>
```

- 圖形化介面 (GUI, graphical user interface)：用滑鼠按下一圖形鈕。



常見的作業系統

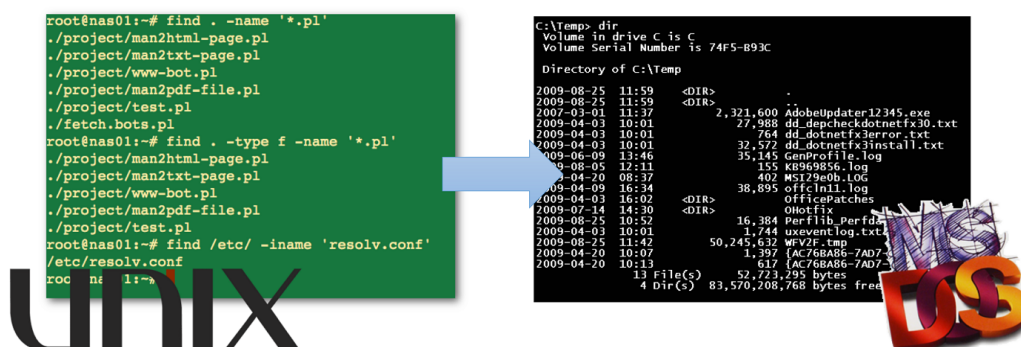
UNIX電腦作業系統是由美國AT&T公司的貝爾實驗室，在1969年開發成功的，具有多工、多用戶的特徵。所有作業系統或多或少衍伸自Unix系統。例如最受歡迎的Unix Shell是Bash (the Bourne Again Shell)，可在Linux系統上使用；Apple電腦的OSX系統為BSD (Berkeley Software Distribution)，是目前商業化最成功的Unix作業系統；Window電腦則是採Unix-like的Power Shell。

(1) Linux系統

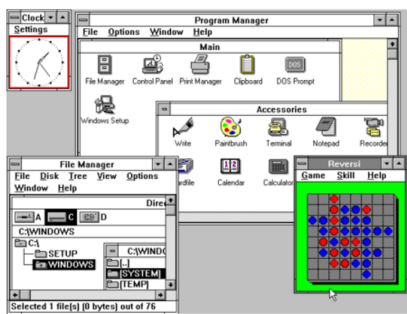
Linux是由芬蘭大學生林納斯·托華斯 (Linus Torvalds)，於1991年以UNIX為基礎，開發出來安裝在個人電腦上的作業系統。近年來開放原始碼系的Linux越來越受歡迎，是市佔率最高的系統。

(2) Windows系統

MS-DOS (Microsoft disk operating system)採用命令列使用者介面，是Microsoft公司於1975年針對IBM PC所推出的作業系統，使用者必須透過鍵盤輸入指定的指令集，才能指揮電腦完成工作，為Unix的仿作。



MS-Windows (Microsoft Windows)則是微軟在MS-DOS基礎上設計的圖形作業系統。1985年、1987年所推出的Windows 1.0和Windows 2.0順利轉型成為IBM相容 PC的標準圖形使用者介面系統，1990年推出的Windows 3.0獲得了空前迴響。



1990 MS-Windows 3.0



1995 MS-Windows 95

1995年推出的Windows 95不再包含MS-DOS，Windows從殼層程式轉變為真正的作業系統。之後Microsoft公司不斷推出新版的Windows作業系統，包括Windows Me、Windows XP、Windows Vista 和Windows 7/8/10。現在的Windows系統皆是建立於 Windows NT核心，可以在32位元和64位元的Intel和AMD的處理器上運行。

(3) MAC系統

macOS，稱「Mac OS X」或「OS X」，是一套執行於蘋果Macintosh系列電腦上的作業系統。Apple公司的創始人Steve Jobs意識到圖形使用者介面的重要性及未來前景，於1983年、1984年推出採用圖形化使用者介面的個人電腦Macintosh (麥金塔)。Mac OS指的就是安裝於Apple Macintosh電腦的作業系統，以卡內基美隆大學開發的 Mach做為核心，是第一個商用化成功的圖形化使用者介面系統，最終版本為1999年推出的Mac OS 9。之後Apple公司改以BSD UNIX為基礎推出OS X，從OS X 10.8開始在名字中去掉Mac，僅保留OSX和版本號。2016年年蘋果公司將OS X更更名為 macOS，現行的最新的系統版本是 macOS Mojave。

⚙️ 2001 ~ 現今 : Mac OS X
○————○

版本	代號
10.0	Cheetah
10.1	Puma
10.2	Jaguar
10.3	Panther
10.4	Tiger
10.5	Leopard
10.6	Snow Leopard
10.7	Lion
10.8	Mountain Lion
10.9	Mavericks
10.10	Yosemite
10.11	El Capitan
10.12	Sierra
10.13	High Sierra
10.14	Mojave

3. 電腦的思考

大家知道電腦是如何思考的嗎？二進位又是什麼呢？

我們常聽到一句話，電腦是零和一所組成的世界，二進位就是一種計算數量的方式。電腦只需要利用零和一這兩種數字，就可以處理很多的問題喔。我們操控電腦的方式，其實就是控制一些很小的電晶體開關來執行程式。

零和一就分別代表電晶體的開和關，每一對零和一組成一個位元，而位元也是電腦儲存空間的最小單位。所有電腦的輸入與輸出，不論是遊戲、文書處理、作業系統、網路影片，都是由非常多的二進位數字表示。

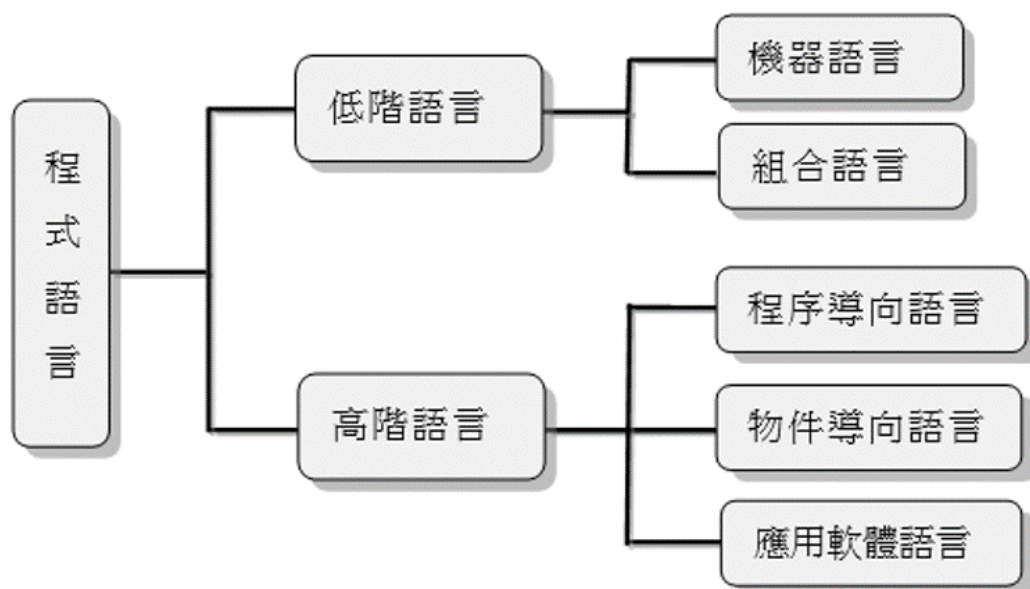
4. 程式語言

程式語言是人和電腦溝通的媒介，透過它命令電腦工作，達成我們想要完成的任務，這些指令的集合就是程式。也就是說，程式語言(program language)是用來撰寫程式的語言，是用來向電腦發出指令，讓程式設計師能夠準確地定義電腦所需要使用的資料，並精確地定義在不同情況下所應當採取的行動。

4-1 程式語言發展過程

程式語言依發展的過程可大致分為低階語言和高階語言。

- 高階與低階的區別是以接近自然語言（人說的話）或電腦語言（0與1）來區別的，並不是垂直劃分比出高下。



(1) 早期發展的程式語言是接近機器能辨識的符號，稱為低階語言，包括機器語言與組合語言。低階語言有程式執行速度快、占用記憶體少以及能發揮硬體效能等的優點。不過，低階語言學習門檻高，編寫程式要花費大量的時間，而且程式碼可讀性差不易維護。低階語言是屬於機器導向語言，和機器相依性高。



圖6-1.6 組譯器運作示意圖。

機器語言是用0、1二進位的機器指令，來編寫電腦能直接識別和執行的機器碼。由於不需經過翻譯就能直接執行，因此程式執行速度最快。機器語言有可以直接執行、占用記憶體少以及執行速度快的優點。但是使用機器語言編寫程式要花費大量的時間，而且程式碼可讀性差且難維護，所以除了電腦硬體的专业人員外已經少人學習。

● 機器語言：

```
10001011 01000110 11111100
10001011 01010110 11111110
10001001 01000110 00001000
10001001 01010110 00010000
```

● 組合語言：

```
mov ax,WORD PTR [bp-4];final
mov dx,WORD PTR [bp-2]
mov WORD PTR [bp+8],ax;new
mov WORD PTR [bp+10],dx
```

● 高階語言：

```
new = final;
```

圖6-1.7 機器語言、組合語言與高階語言。

為了增進與電腦之間的溝通速度，程式設計師們設計出了一種使用英文簡寫來代表各種基本運算的語言，是由字母和數字組成的助憶碼，取代0、1的機器語言符號。例如Add代表加、Sub代表減、Inc代表累加、Mov代表移動，各位看看ADD R1、R2、R3是不是比零和一看起來親切一些呢。這些以英文簡寫所構成的語言叫做組合語言。組合語言必須先用組譯器翻譯成機器語言後才能被電腦接受。

因為機器語言讀寫不易，因此後來發展出組合語言，用容易記憶的英文單字縮寫來取代零與一的組合，這些縮寫字串稱做輔助記憶碼或簡稱助憶碼，

(2) 由於低階語言不易學習，所以發展接近人類日常語言的程式語言，稱為高階語言，其語法簡單易懂。執行一行高階語言程式，就等於執行一長串的機器語言程式碼。高階語言具有容易學習、編寫程式較為快速，程式碼具有可讀性高、容易維護和可攜性高等的優點。但是，有程式執行速度稍慢、占用記憶體較多的缺點。例如：各位看看R1=R2+R3是不是又更

親切一些了呢。

表6-1.2 低階語言和高階語言的比較

比較項目	低階語言	高階語言
執行速度	快	慢
程式編寫難易	難	易
維護難易度	難	易
跨平台性	低	高
可攜性	低	高

- 程序導向語言是按照程式敘述的先後順序、指令邏輯和流程執行的程式語言。常用的程序導向語言有BASIC、C...等。
- 物件導向程式設計，是一種以物件為視角，利用多個物件來組成程式。常用的物件導向語言有Python、C++、Java、C#...等。
- 應用軟體語言是應用軟體專屬的程式語言，用來擴展該應用軟體的功能，常用的應用軟體語言有VBA、JavaScript。
- 查詢語言是以問題為導向，只描述問題不用撰寫解決問題的步驟，可以提高程式設計的產量。例如電子試算表-Excel、資料庫管理查詢系統-Oracle...等。

4-2 為什麼要用程式語言

不能直接用中文或英文嗎？不能！

不能用「中文」或「英文」嗎？



程式語言與自然語言的區別

程式語言是指使用者用來與電腦溝通之文字記號所形成的指令集合，也就是電腦能夠接受的語言。就好像人類的語言如中文、英文、法文或西班牙文等，程式語言的種類繁多，而且各有其特定的語法結構與不同的適用範圍。

- 自然語言，說的是人話、一種意思多種解讀
- 程式語言，說的不是人話、一種意思一種解讀，能順利和電腦溝通。



程式語言 = 限制多種說法



4-3 程式設計

什麼是程式設計？

程式設計就是把你希望電腦做的事情，用電腦語言寫出來，交給Compiler，翻成電腦能懂得形式，接著讓電腦根據這些指令很快地做完。

我們可以把程式設計想成是對電腦說話，叫它解決問題或做出東西，像是遊戲或應用程式。我們用來對電腦說話的語言就是程式語言，而給電腦的一整套指令稱為「程式」(program)或「程式碼」(code)。

為什麼要學程式設計？

程式設計是一種強大的工具，讓我們發揮創意來至做自己的應用程式、工具和遊戲。程式設計讓藝術家和Maker創作只有程式碼能完成的作品。學習程式設計能幫助我們瞭解周遭的數位世界。現代的世界是建立在程式碼的基礎上，從智慧型手機的應用程式、獻上購物網站到自動櫃員機(ATM)都是。因為電腦影響了幾乎所有產業，瞭解程式碼可以讓你為任何工作做好準備。

程式設計的一環是解決問題。寫程式時，我們要把一個問題分解成不同步驟，用數學和邏輯發展出解決方法，接著測試和調整來解決這個問題。程式設計的學習能吸引各年齡層的學生並幫助他們瞭解數學、科學、語言等領域。

影片觀念說明 - 何謂電腦語言與程式設計