

程式設計是甚麼？

臺北市立大學 [資訊科學系\(含碩士班\)](#)

賴阿福

CS TEAM

為什麼想學程式設計?why

- 請寫在右下角即時聊天區
- 目的? 動機? 好奇?想體驗? 未來?
- 人人須回答 不管他人觀點

臺北市立大學

- 原臺北市立教育大學源自於**1895年**建立的「**芝山巖學堂**」，迄今已有**118年**的歷史。**1945年**臺灣光復，稱為「**臺灣省立臺北女子師範學校**」；**1979年**更名為「**臺北市立師範專科學校**」；**1987年**升格為「**臺北市立師範學院**」；**2005年8月1日**改名為「**臺北市立教育大學**」。臺北市立大學於**2013年8月1日**正式成立，由臺北市立教育大學及臺北市立體育學院合併而成之綜合性大學。
- 臺北市中正區愛國西路**1號**
- 與北一女為鄰
- [資訊科學系\(含碩士班\)](#)

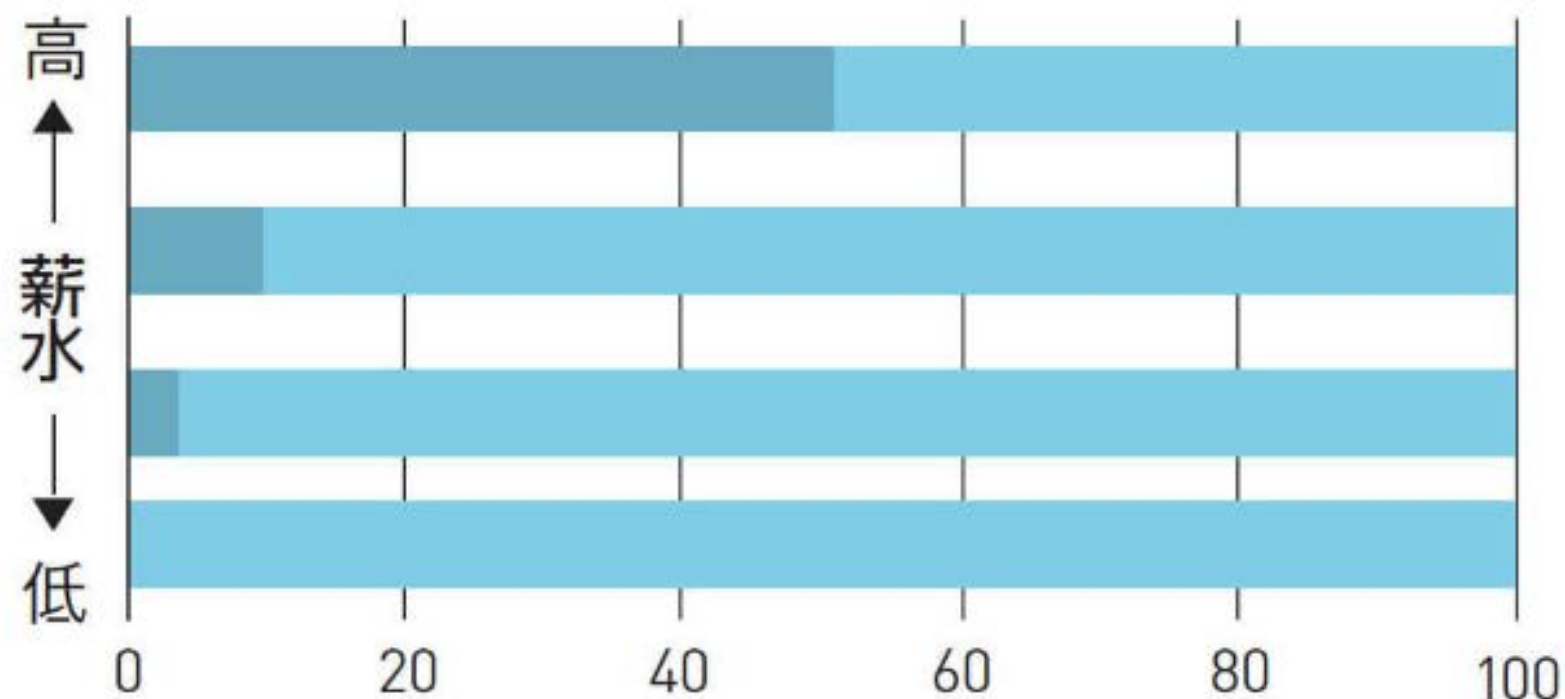


表2

美國近五成高薪職缺 要求懂程式

美國給薪等級與程式人才需求比例

應徵條件含懂程式語言比率(%) ■ 懂 ■ 不懂



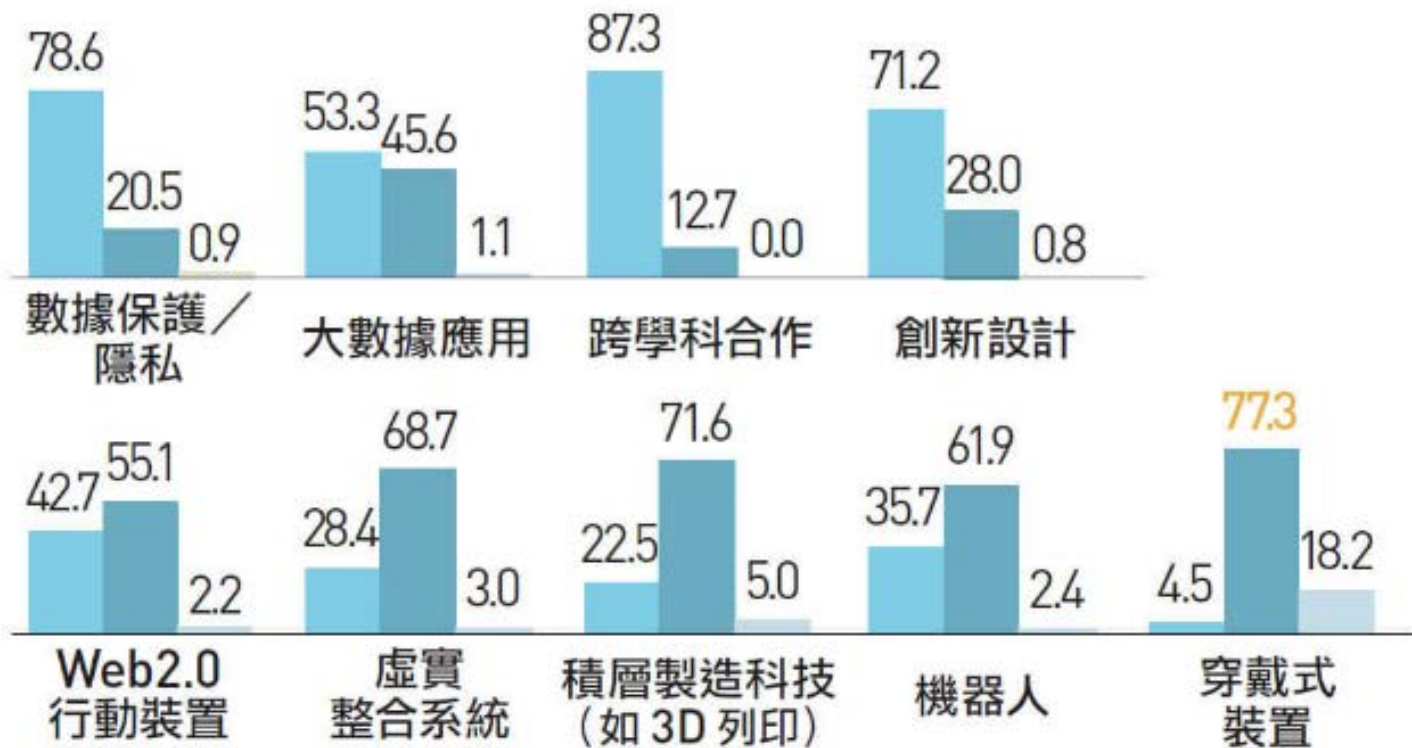
資料來源：經濟學人

表1

穿戴裝置技能 10年後最重要

認為該職能很重要的企業比例

■ 現在很重要 ■ 2025年時很重要 ■ 完全不重要



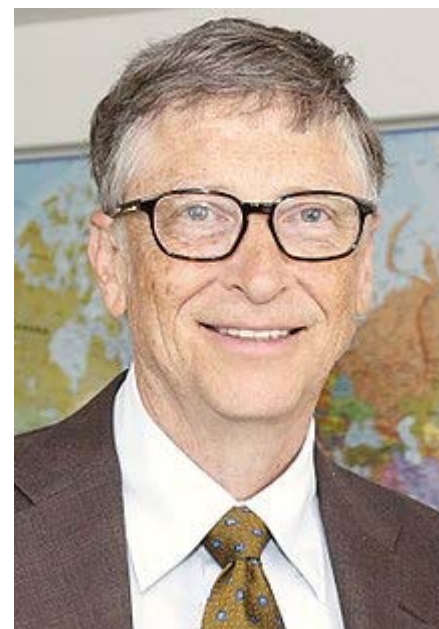
註：此為VDMA針對210家德國企業問卷調查結果。
資料來源：德國機械及製造商協會 (VDMA)

與程式設計軟體設計有極大關係

程式設計趨勢

- 台灣很多大學將程式設計列為必修
 - 不管哪一科系
- 教育部要求每一個大學6成以上的學生都要會程式設計
- 未來通過高中程式設計課程在大學可抵免
- 台灣舉辦運算思維檢定(Brebra)、程式設計先修檢定(APCS)

Who is Bill Gates?
Bill Gates 幾歲開始學程式?



英雄出少年

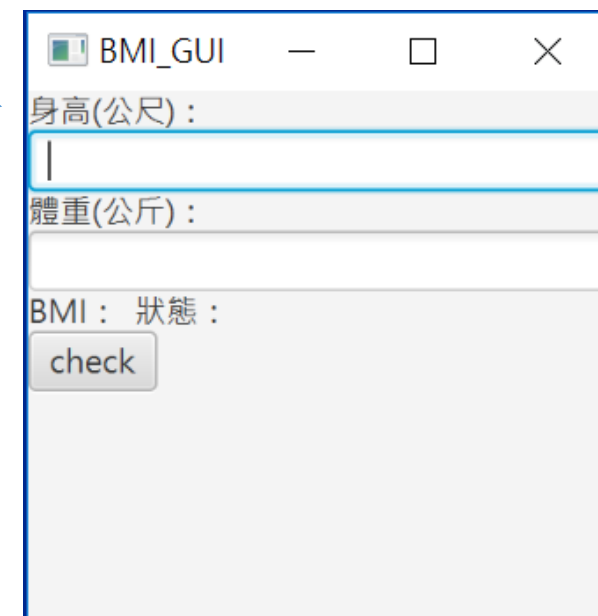
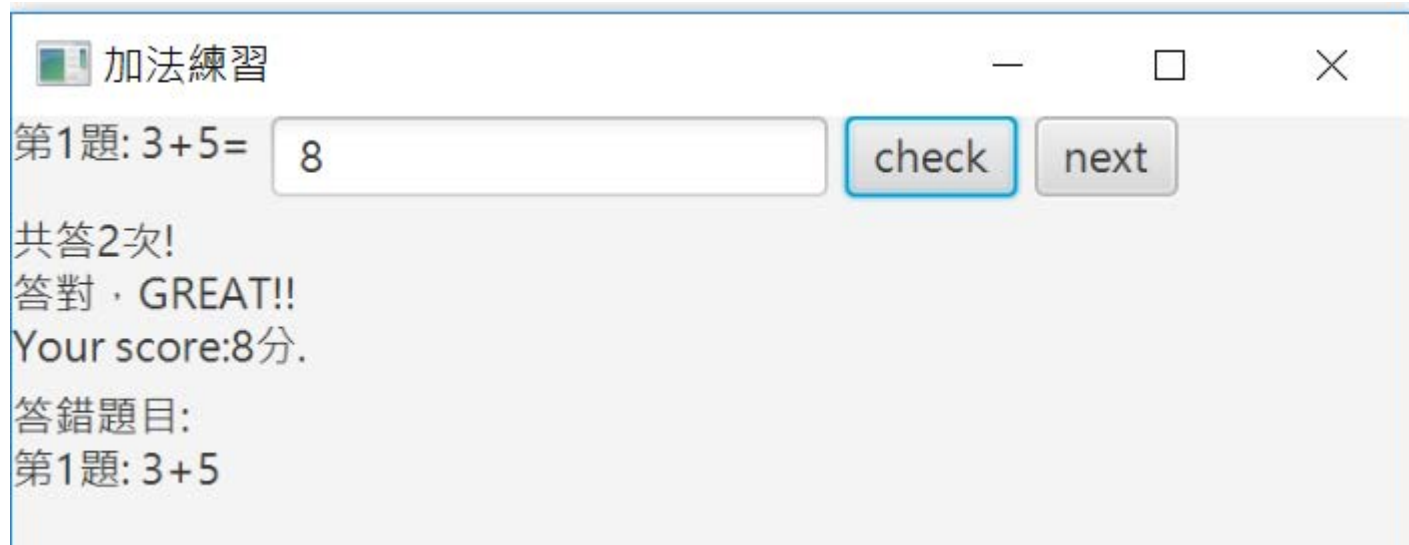
典範: 13歲的Bill Gates

- At 13, he enrolled in the [Lakeside School](#), a private preparatory school. When he was in the eighth grade, the Mothers' Club at the school used proceeds from Lakeside School's [rummage sale](#) to buy a [Teletype Model 33](#) ASR terminal and a block of computer time on a [General Electric](#) (GE) computer for the school's students.^[23] Gates took an interest in programming the GE system in **BASIC**, and was excused from math classes to pursue his interest. He wrote his first computer program on this machine: an implementation of **tic-tac-toe** that allowed users to play games against the computer. **Gates was fascinated by the machine and how it would always execute software code perfectly.** When he reflected back on that moment, he said, "There was just something neat about the machine."
 - https://en.wikipedia.org/wiki/Bill_Gates

你/妳就是下一個Bill Gates
你/妳就是未來的Steve Job

課程目標

- 學習Java語言:基本觀念、語法
- 學習Java程式設計基本技巧
 - 變數、陣列、if, for, while, switch, method, recursive method
 - 文字介面、圖形介面
- 能以Java程式設計解決基本問題



評量

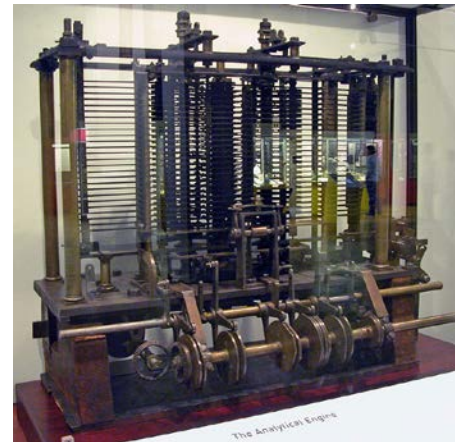
- 課堂出席及回答問題(搶答、問問題、填問卷)
- 線上學習表現: 自我學習(preview, review)
- 準時交習題(寫程式、設計歷程)
- 參加實作輔導
- 小專題製作
- 線上測驗
- 學科考試: 筆試

何謂指令、程式、語言？
為何學習程式？

電腦語言演化

撥動開關(Flipping Switches)

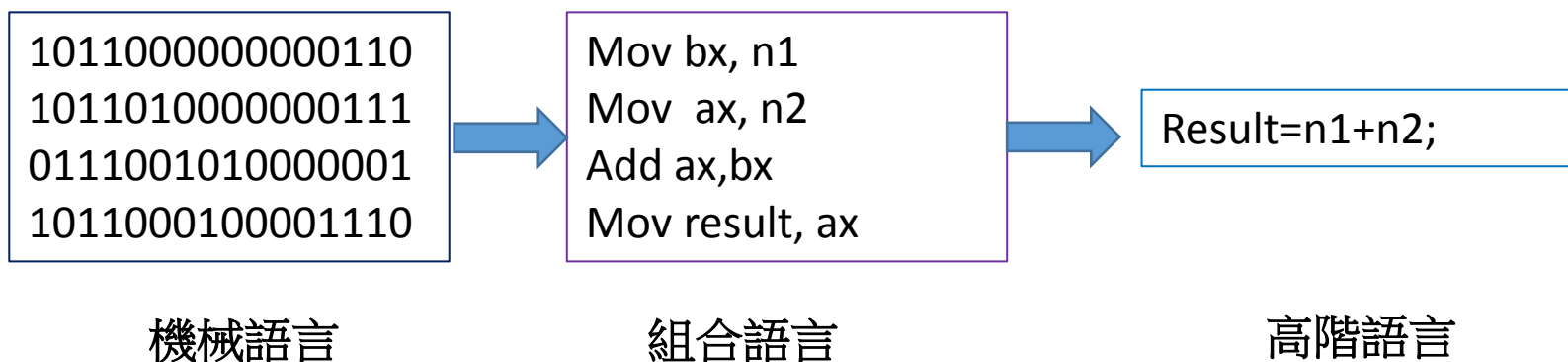
- 最早的電腦，無法用鍵盤輸入文字；而是要撥動電接電腦不同部分的開關。
- 如果你想要兩個數字相加，你就把數字部分的線路和加法的線路連起來，再和另外一個數字的線路連起來，最後把線路連到輸出結果的地區。
 - 早期電腦只有資料被儲存在記憶體，而程式是透過手工操作設定線路系統的開關
- **Stored program: a memory to store both data and instructions**
 - **von Neumann model and Princeton architecture**

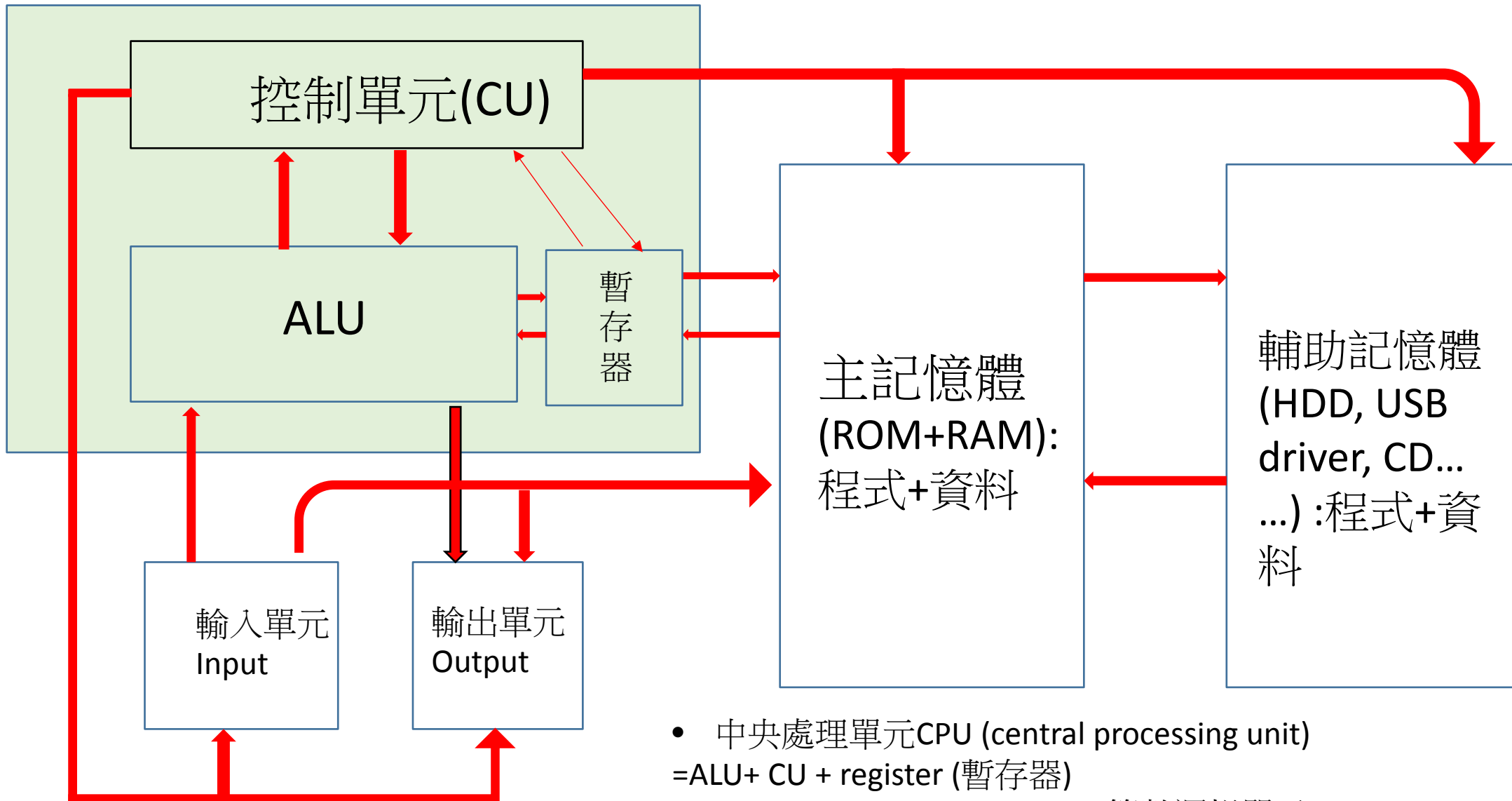


[Babbage's Analytical Engine](https://en.wikipedia.org/wiki/History_of_computing_hardware), the first general-purpose programmable computer (https://en.wikipedia.org/wiki/History_of_computing_hardware)

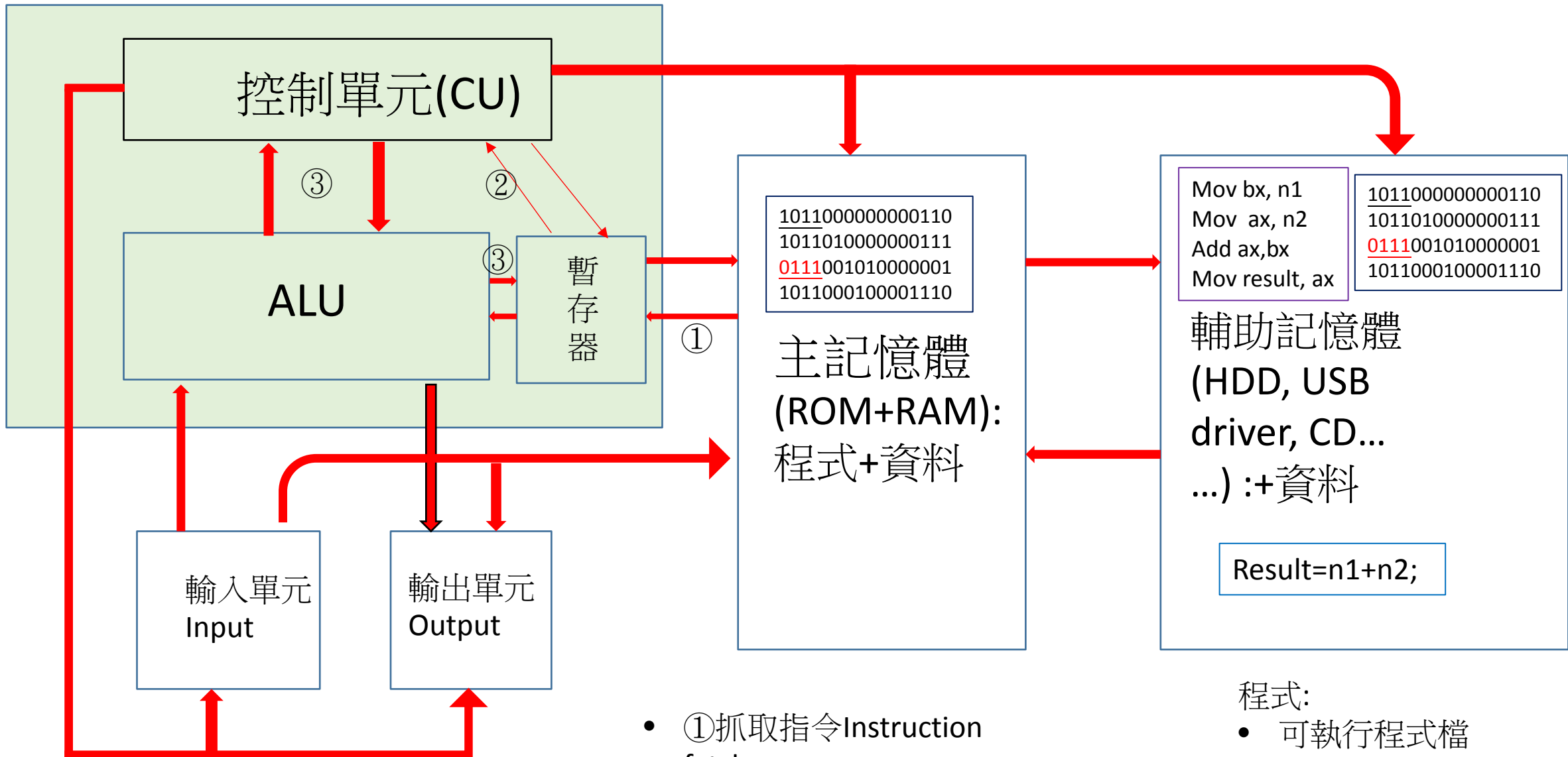
機械語言、組合語言

- (machine language):由機械指令(instruction)由0、1組成
 - 電腦聽(看)懂**機械語言**?
- 組合(中階/中介)語言(assembly language):組合語言指令(文字形式)與機械指令是一對一
 - 在設計程式時，你必須知道電腦處理的細節，且組合語言絕對不給你任何安全措施。你幾乎可以直接叫電腦做任何的事情，而且它樂於接受，甚至你可以要求它清除記憶體內的所有事情。另外，每一種電腦的組合語言都不一樣。所以你知道這一種機器的組合語言並不見得瞭解另一種機器之組合語言。**電腦聽(看)懂組合語言?**





- 中央處理單元CPU (central processing unit)
=ALU+ CU + register (暫存器)
ALU: Arithmetic logic unit (算數邏輯單元)
- Main memory
ROM: read only memory (唯讀記憶體)
RAM: Random access memory (隨機存取記憶體)



- ① 抓取指令 Instruction fetch
- ② 指令解碼 Decode
- ③ 執行指令 Execute
- 寫回 Write back



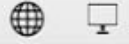





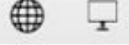
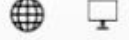
- 程式:
- 可執行程式檔
 - 來源程式檔

你認識那些單元?

- 中央處理單元CPU
- Main memory
 - ROM: read only memory (唯讀記憶體)
 - RAM: Random access memory (隨機存取記憶體)
- 輸入單元
- 輸出單元

高階語言

- 由文字所組成(指令以文字形式撰寫)
- 電腦聽(看)懂高階語言?
- 高階語言
 - Python、c、java、javascript、Visual basic、
- IEEE透過10種來源的12個指標分析，算出[2016年熱門程式語言排行榜](#)
 - <https://www.bnnext.com.tw/article/40380/BN-2016-07-28-180528-218>

Language Rank	Types	Spectrum Ranking
1. C		100.0
2. Java		98.1
3. Python		98.0
4. C++		95.9
5. R		87.9
6. C#		86.7
7. PHP		82.8
8. JavaScript		82.2
9. Ruby		74.5
10. Go		71.9

(圖片來源：[IEEE](#))

電腦語言(computer language)有哪些?

- 高階語言類型
 - 一般程式語言: C, C++, COBOL, FORTRAN, BASIC, PASCAL, DELPHI, Lua
 - 物件導向程式語言: JAVA, PYTHON, C++, C#
 - 網頁程式語言: HTML, PHP, JSP
 - 手機程式設計: Android(JAVA), IOS(Objective C, Swift)
 - 作業系統SHELL語言
 - 早期: COBOL (商)、Fortran(工程)
- 程式開發整合環境(IDE : integrated Development Environment)
 - Visual Basic
 - Dev cpp
 - Eclipse
 - NetBeans
 - XCode
 - App inventor2
 - Android studio

積木式程式設計(BLOCK PROGRAMMING)

- 積木是程式(**block**):程式由積木形式指令組成
- 常見積木式程式設計環境
 - MIT Scratch
 - APP inventor 2 、 code.org
 - webduino

什麼是程式(program) 程式設計(programming)、寫程式(coding)? 程式碼(code)?

- 什麼是程式(program) : 由一堆指令(instruction)或敘述所組成
- 程式設計(programming)、寫程式(coding)
- 程式碼(code) : code包含程式碼、資料碼
- **Coder**: 碼農
- 電腦聽(看)懂何種語言?
- 如何讓電腦聽(看)懂你所寫程式?

電腦(cpu)如何區分指令、資料?

- 在電腦內部指令、資料都是由**0,1**組成
- 機械指令由**0,1**組成
- 資料亦以(編碼),如英文使用**ASCII code**;
 - 資料有許多類型(Data type): **int, double,**
- 電腦(cpu)如何區分指令、資料?

總結

- **Java** 是最流行(程式設計者所喜愛採用)高階程式語言之一
- 機器語言:低階程式語言，只有特定機器能理解及執行
- 高階程式語言:接近人類語言形式，但電腦(機器)無法理解及直接執行
- 高階程式語言需經轉換/編譯(**compile**)為機器語言