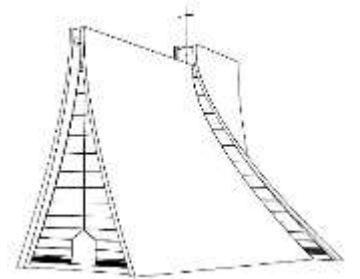




運算思維與程式設計

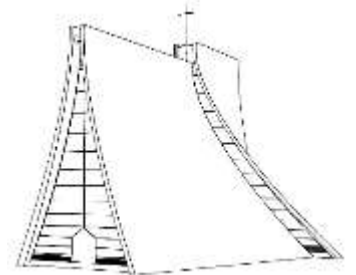
CH6-電腦與程式語言





電腦的組成

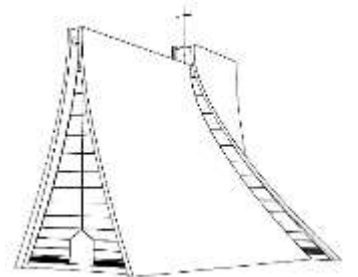
硬體與軟體分別為電腦的兩大構成要件：





認識硬體

1. 輸入單元
2. 輸出單元
3. 處理單元
4. 記憶單元





1. 輸入 / Input



眼睛



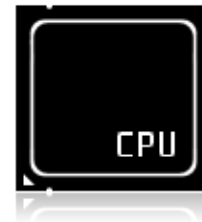
鍵盤

2. 處理 / Process



大腦

Center Processing Unit

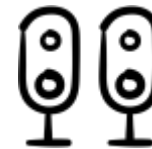


中央處理器

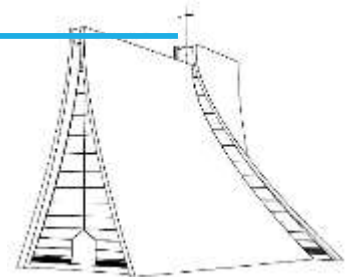
3. 輸出 / Output



嘴巴

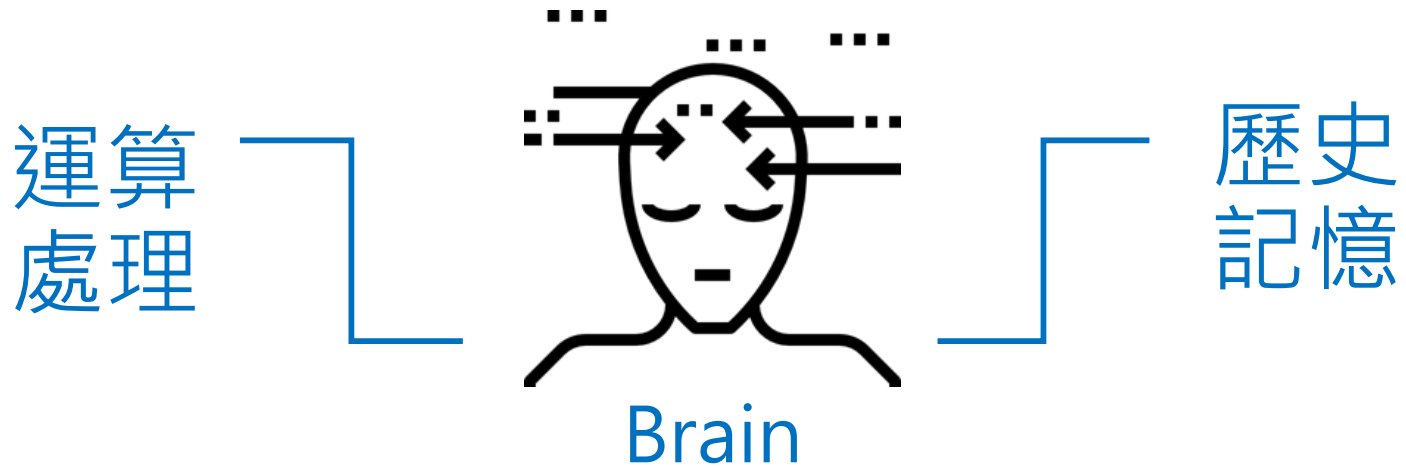


喇叭

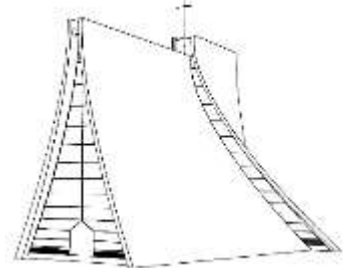




4. 記憶 / Memory

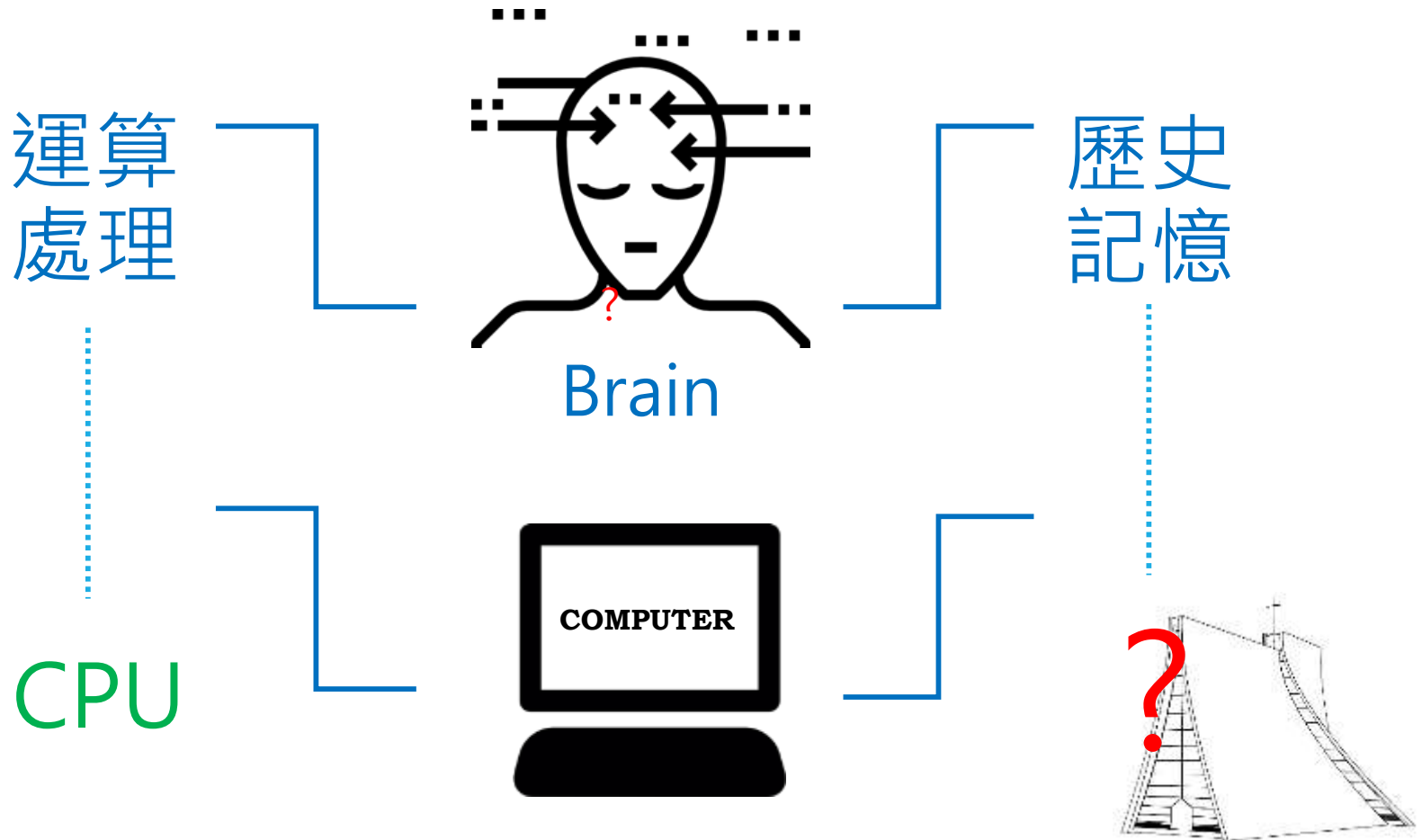


我們的大腦可以同時處理運算與記憶功能，但電腦就沒有那麼厲害。



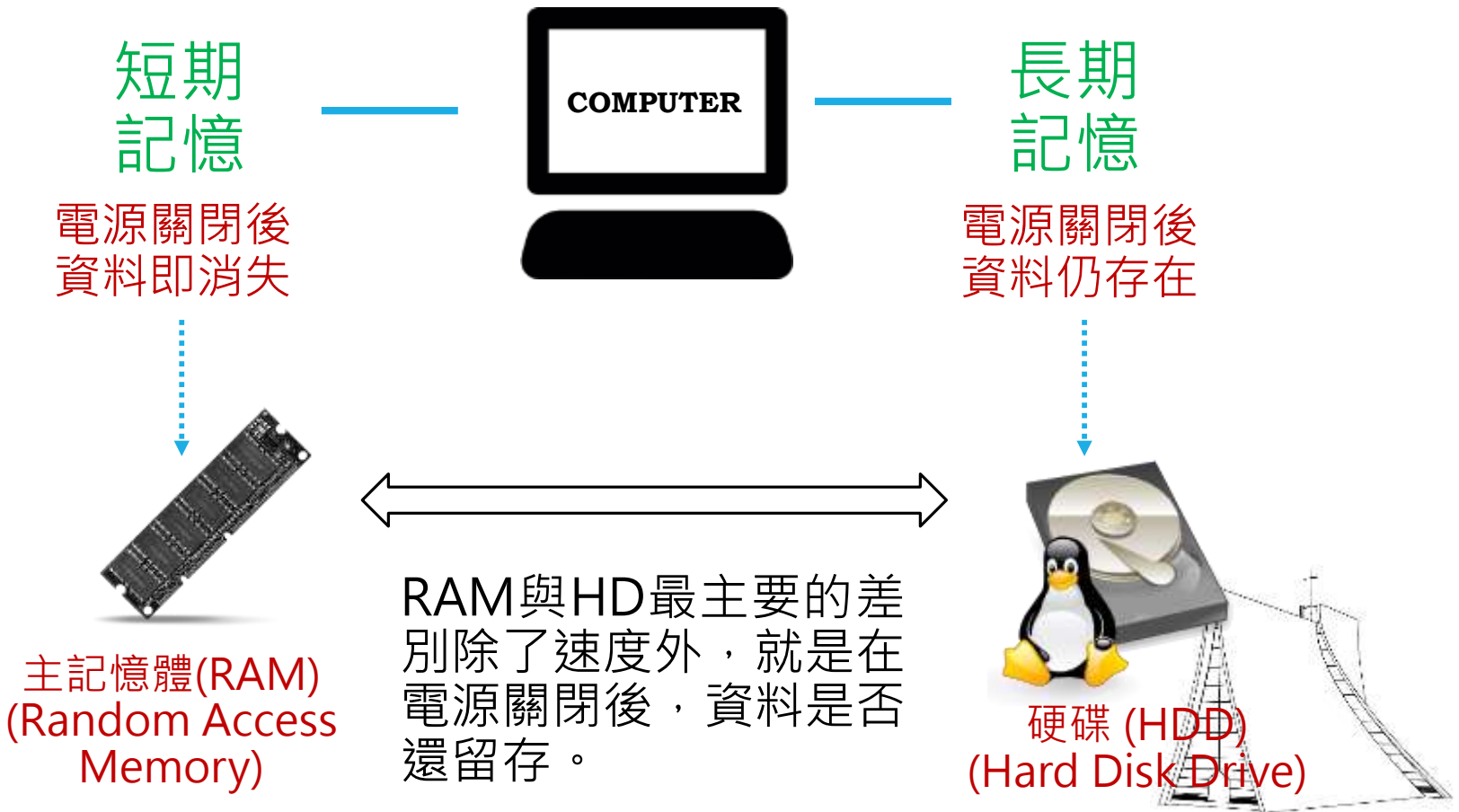


4. 記憶 / Memory





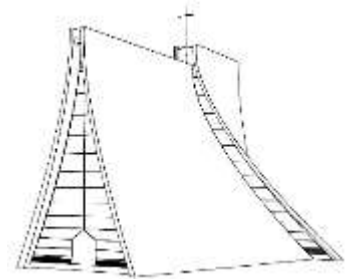
記憶種類





想想看，

除了硬碟外，還有哪些儲存設備？





硬碟 HDD



安全數位卡
SD Card



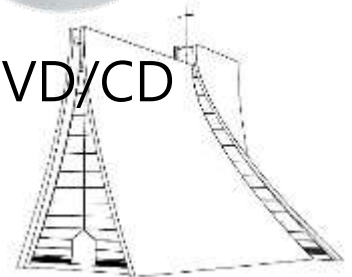
隨身碟



記憶體
RAM

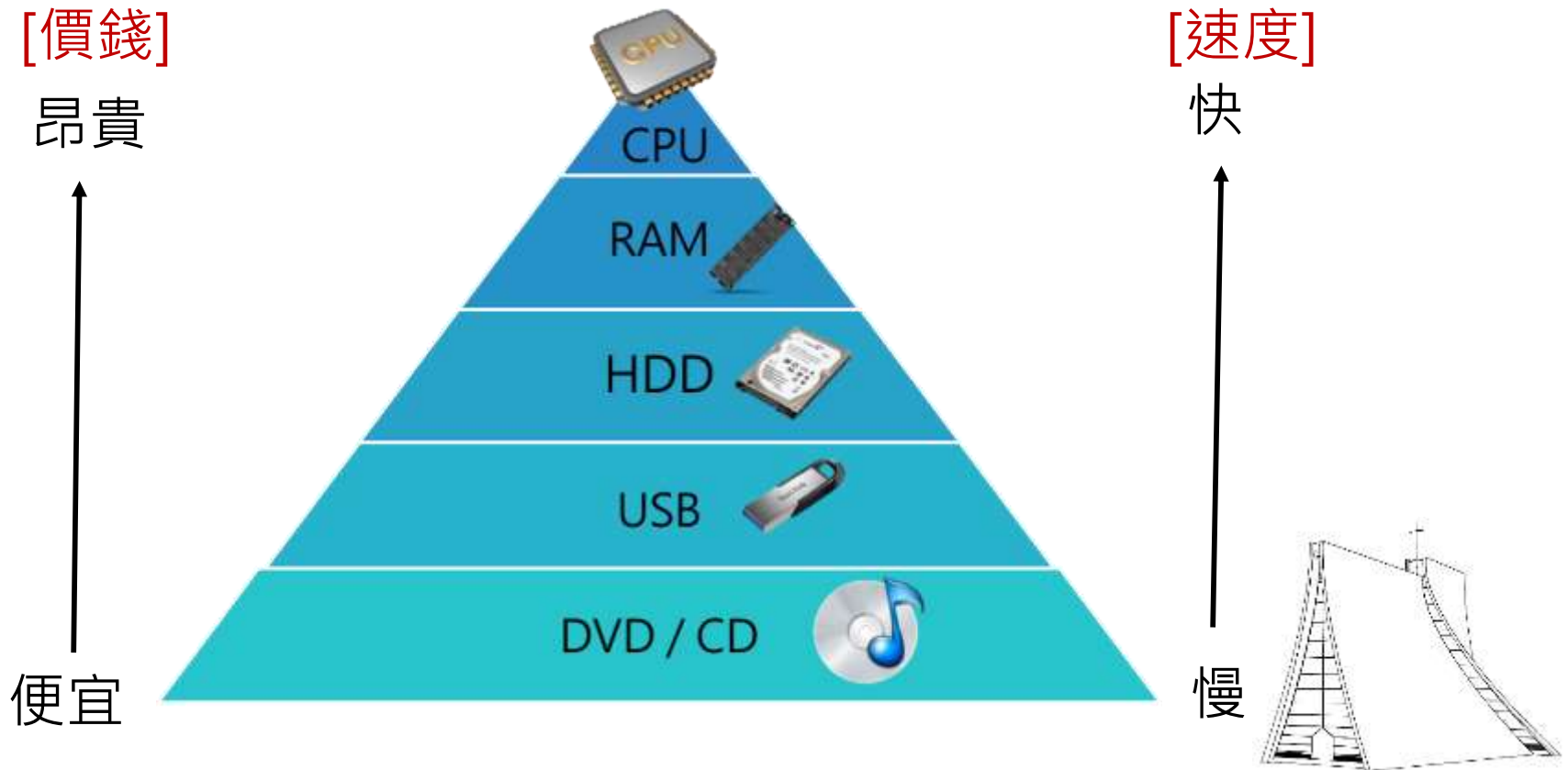


DVD/CD





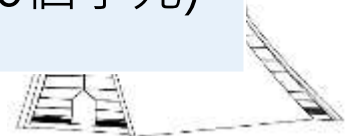
硬體速度比較





儲存單位

單位		相當於.....
位元 (bit)	0 或 1	是(True) 或 否(False)
位元組 (byte)	8 位元	鍵盤上任一字母, A, B,...
	例如： 0100 0001 → A 0100 0010 → B 0100 0011 → C	
千位元組 (Kilobyte, KB)	約1000個位元組	約兩段文字 (1000個字元)





單位		相當於.....
百萬位元組 (Megabyte, MB)	約1000個千位元組	約一分鐘的數位錄音檔
十億位元組 (Gigabyte, GB)	約1000個百萬位元組	將近4,500本書 (每本書平均200頁)
兆位元組 (Terabyte, TB)	約1000個十億位元組	約 230 部DVD電影
千兆位元組 (Petabyte, PB)	約1000個兆位元組	約 3.5 億張照片

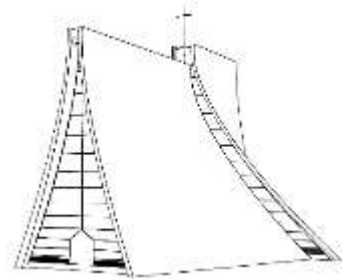




你可知道，我們的大腦相當於多少容量的儲存空間？

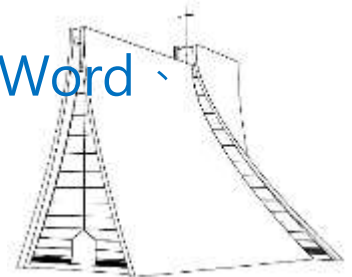
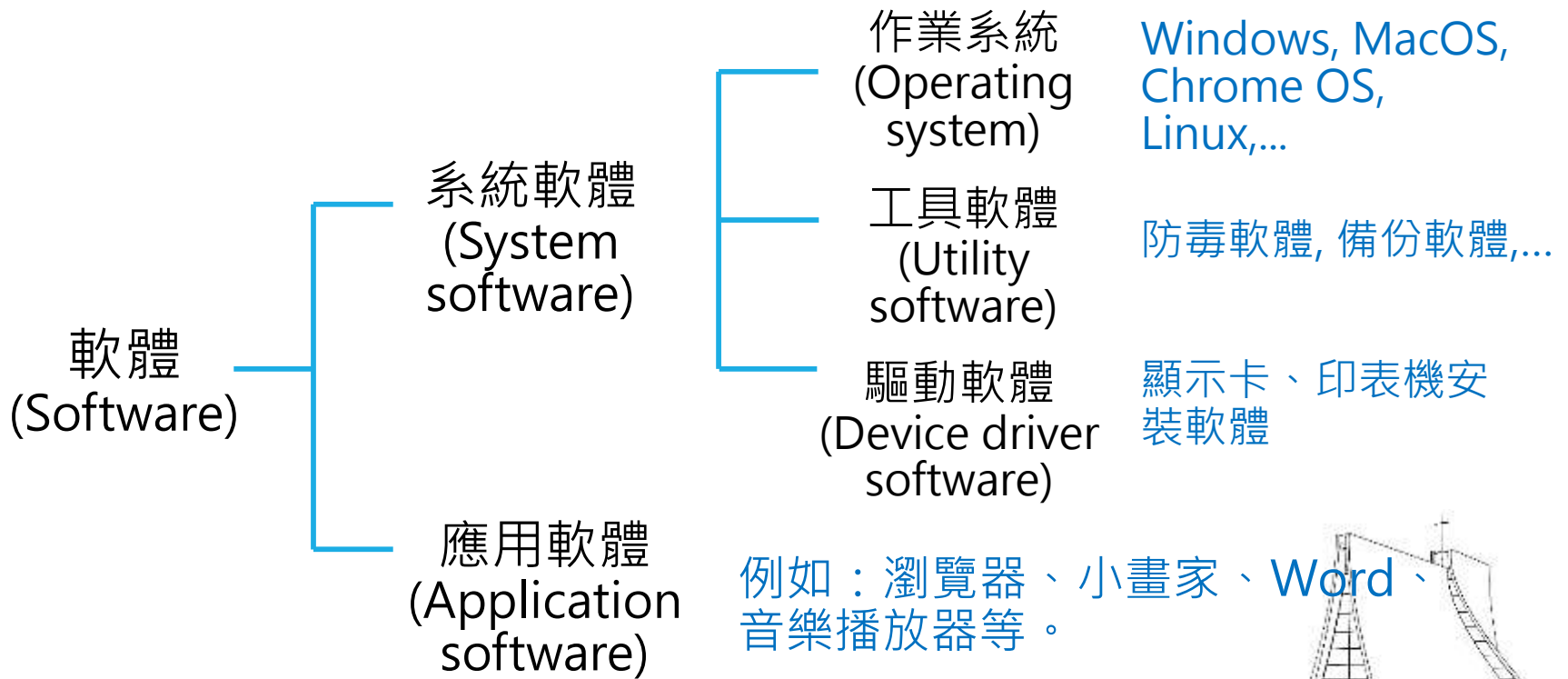


這是一項至今已經爭論多年的議題，有些神經科學家認為應該高達2.5PB，有些人則覺得大約1TB。





認識軟體



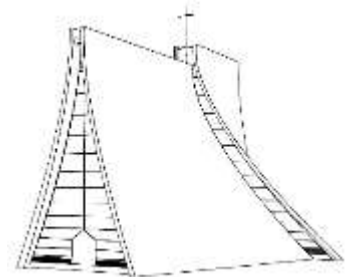


電腦與程式語言的關係

電腦的大腦是個二進位世界，只認得 0 與 1。所有與計算、資料有關的東西，都只能用這兩個字來編碼。

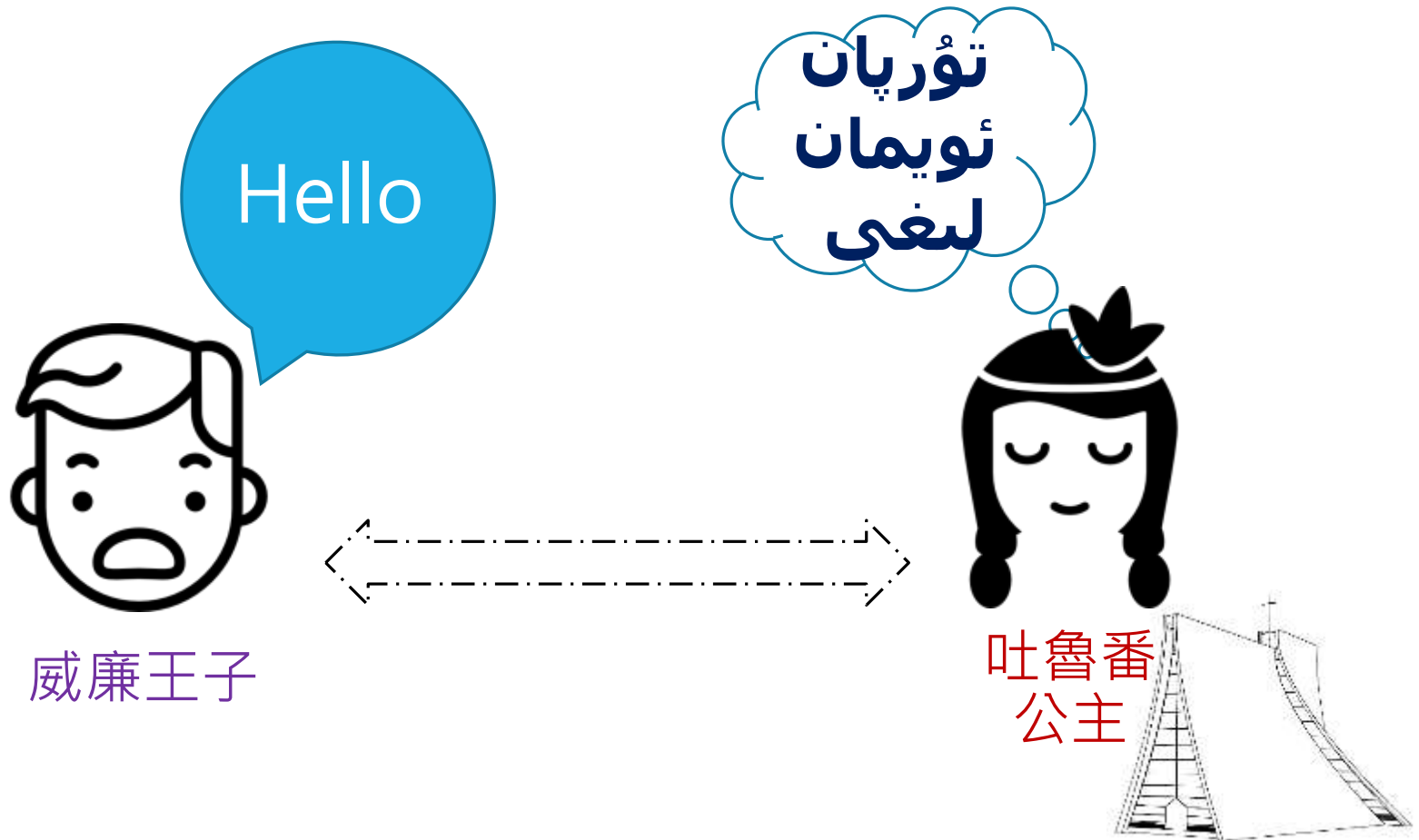
顯然這不是一般人能夠看得懂的內容。若將電腦擬人化，所幸我們可以透過程式語言來和電腦溝通，不用理會那些由 0 與 1 構成的天書。

電腦與程式語言如何溝通？





來自不同國家的人們，彼此該如何溝通？





精通多國語言的全能翻譯官，搭起威廉王子與吐魯番公主間有誼的橋樑！



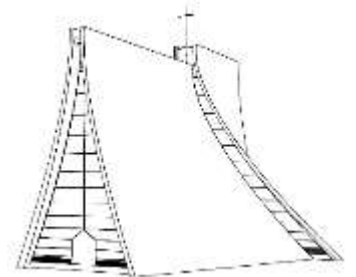
威廉王子

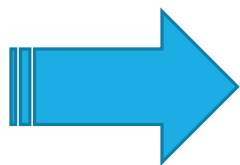


吐魯番
公主



全能翻譯官



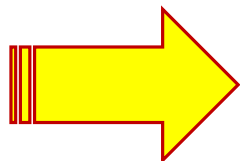


程式語言

Programming
Languages

高階語言

Java、Python、C 語言、
C++、JavaScript、Go
、ASP.NET.....

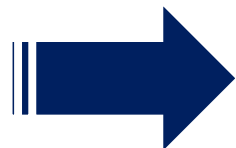


編譯器

Compiler

程式語言與電
腦間的翻譯官

是一種電腦程式，它會將
用某種程式語言寫成的原
始碼 (原始語言)，轉換成另
一種程式語言 (目標語言)。
此目標語言，有時概稱為
機器碼。

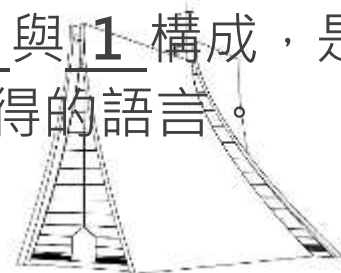


電 腦

Computer

低階語言

機器碼由 0 與 1 構成，是
電腦唯一認得的語言。

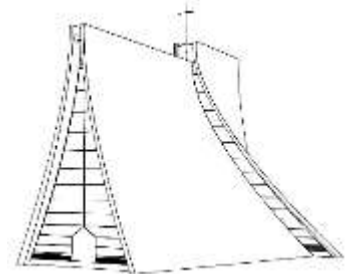




編譯器與直譯器

編譯器(Compiler)：在電腦的專業名詞中，我們寫好的程式也可稱為原始碼。編譯器會先將原始碼轉換成可執行碼，並另存檔案。執行時，先在電腦上載入這個編譯後的檔案至記憶體後，再啟動執行。

直譯器(Interpreter)：與編譯器不同，原始碼可直接在直譯器上被執行，無須先行編譯並另存執行碼檔案。

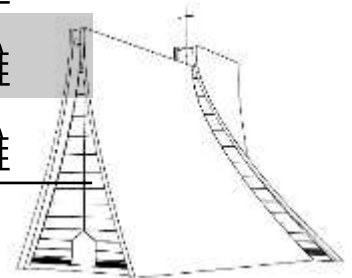




比較

目前流行的程式語言中，有些如C++, C#, Java 等，是採取編譯器；另外有些程式語言，如Python, R, Ruby等，則採取直譯器。編譯器與直譯器比較，直譯器顯然較為簡單、直接。但是由於下表諸多原因，許多程式語言採取編譯器方法：

	編譯器	直譯器
執行速度	較快	較慢
穩健性	較佳	較差
除錯	較易	較難
大型軟體開發	較易	較難



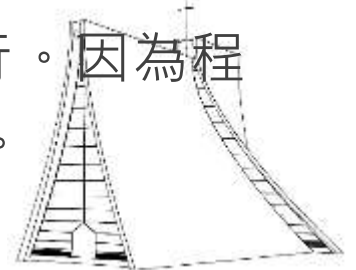


語法錯誤 VS 語意錯誤

程式撰寫過程不是一蹴可成，往往存在錯誤需經過多次的修改與測試，方才能夠正確無誤地執行。程式錯誤有兩種。

語法錯誤：也就是說不符合程式語言的文法或是字彙錯誤。此類錯誤編譯器或直譯器就能夠發現，程式不待執行前就可偵測出來；

語意錯誤：人類語言中雖然用對文法與詞彙，但是我們仍然可能因誤解而說錯話。在撰寫程式時，同樣會發生。儘管編譯器或直譯器認為程式正確，程式卻不見得可以正常執行。因為程式執行的過程是一個動態的世界，有許多例外會發生。

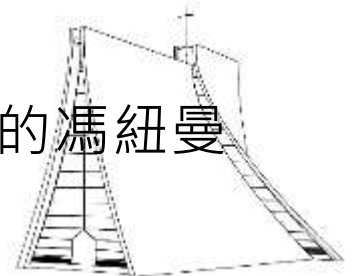




隨堂演練

1. 生活中哪些應用與電腦或資訊科技有關？
2. 除了課堂所提的之外，電腦還有哪些輸入/輸出設備？電腦硬體的組成要件有哪些？
3. 決定電腦效能優劣的硬體元件是？
4. 只認得「0」與「1」機器碼的電腦，該如何表達文字、數字、影像？
5. 什麼是「直譯器」(Interpreter)?
6. 電腦有了硬碟，為什麼還需要RAM？兩者都是儲存元件，為什麼不擇其一？
7. 你知道你正在用的電腦是延用70 多年前所發明的馮紐曼(Von Neumann)架構嗎？

<https://www.inside.com.tw/2017/06/05/von-neumann-architecture>



Thank You!