

2

資料的表達

Data Representation

2.1 資料的型別 *Data Types*

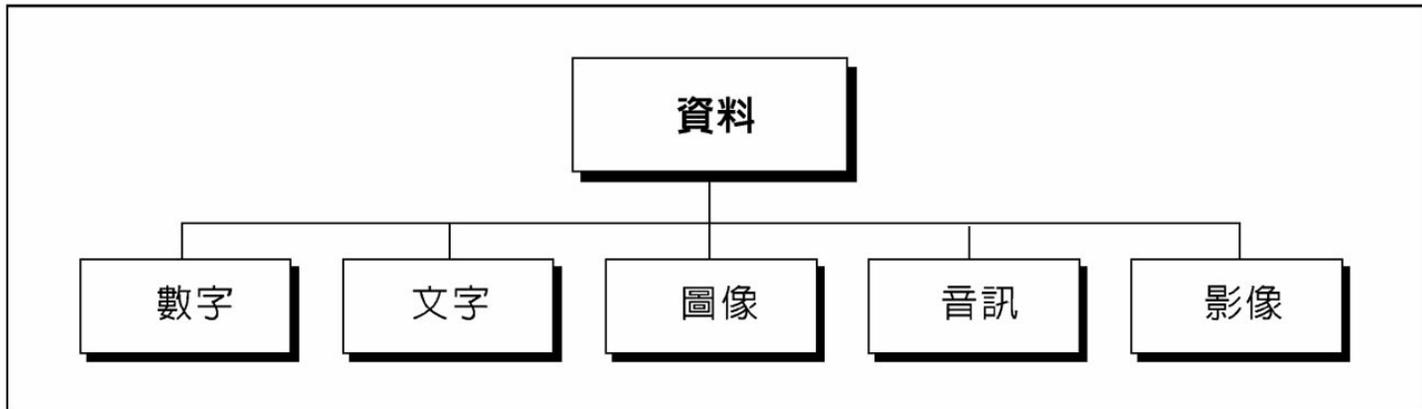


圖 2.1 不同的資料型別

2.2 電腦裡的資料 *Data Inside the Computer*

最有效率的方法是採用一個一致的資料表達方式。從電腦外而來的各種資料型別存入電腦時，都先要轉換成這個一致的資料表達方式，並且當資料從電腦取出的時候再被轉換回來。這個普遍的格式稱為位元樣式。

位元 BIT

爲了表達不同型別的資料可以使用位元樣式 (bit pattern)，這是一個一連串的位元，有時又稱為位元字串。圖 2.2 是由 16 個位元組成的位元樣式，它是許多 0 與 1 的組合。這意味著如果想要儲存一個由 16 個位元組成的位元樣式，就需要有 16 個電子開關。

位元樣式 BIT PATTERN

進一步討論位元樣式前，我們必須先定義**位元** (bit)。一個位元 (二進位數字) 是電腦裡可以被儲存的最小資料單元；可以是0 或1。一個位元可以表達一個設備兩個狀態中的任一種狀態。



1 0 0 0 1 0 1 0 1 0 1 1 1 1 1 1

圖 2.2 位元樣式

位元組 **BYTE**

傳統上，一個長度為8的位元樣式稱為一個**位元組**（byte）。這個專有名詞已經被用來計算記憶體或其他儲存設備的大小。

2.3 表達資料 *Representing Data*

文字 TEXT

任何語言裡的一段文字 (text) 是用來表示該語言中一個想法的一連串符號。例如，英文用26個符號 (A,B,C,...,Z) 表示大寫字母，用26個符號 (a,b,c,...,z) 表示小寫字母，用9個符號 (0,1,2,...,9) 表示數字符號 (不是數字；稍後你將明白其中的差異)，還有一些符號 (.,?,:;,...,!) 用來表示標點符號。其他如空白、換行和跳格等則是作為文字定位與增加可讀性。

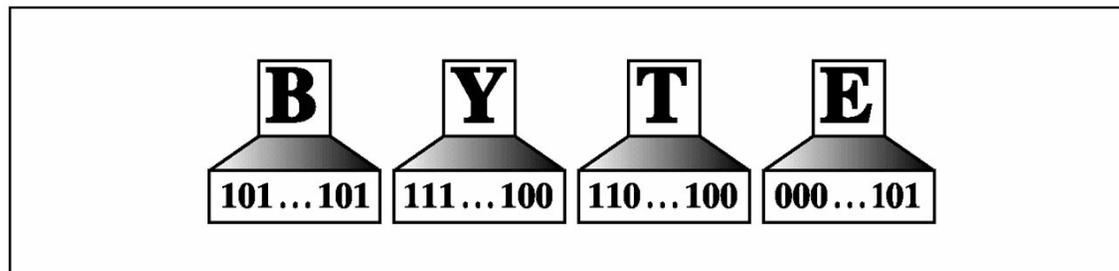


圖 2.4 用位元樣式來表達符號

雖然位元樣式的長度跟符號的數量有關，但是兩者之間並非呈現線性關係；而是對數關係。如果只需要表達兩個符號，那位元樣式的長度為 1 就足夠了 ($\log_2 2$ 是 1)。如果是四個符號，那長度就是 2 個位元 ($\log_2 4$ 是 2)。表 2.1 可以輕易地顯示它們之間的關係。

符號數量	位元樣式長度
2	1
4	2
8	3
16	4
...	...
128	7
256	8
...	...
65,536	16

表 2.1 符號數量與位元樣式長度

編碼法 CODES

美國資訊交換標準碼 美國國家標準局 (American National Standards Institute ; ANSI) 發展了一套編碼法稱作**美國資訊交換標準碼** (American Standard Code for Information Interchange ; ASCII) , 這套編碼中每個符號使用了7 個位元, 代表在這個編碼法中可以定義出128 (2^7) 個不同的符號。附錄A 是 ASCII 完整的位元樣式。圖 2.5 顯示了"BYTE" 這個英文字如何以ASCII 來表達。

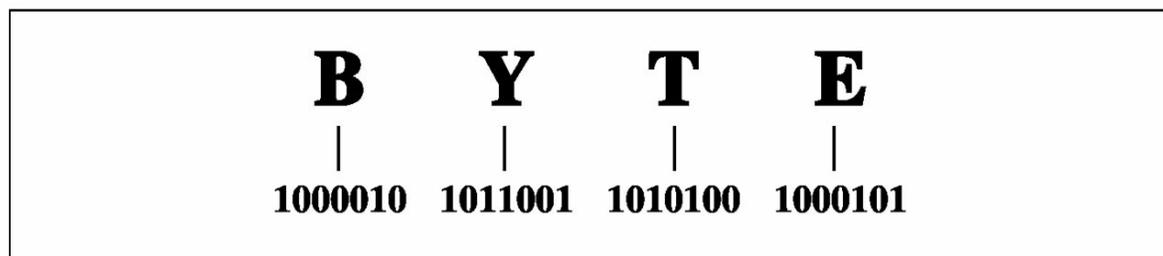


圖2.5 用美國資訊交換標準碼表達"BYTE" 這個英文字

擴充的美國資訊交換標準碼 爲了讓每一個樣式的長度剛好是一個位元組 (8 個位元)，ASCII 的位元樣式在最左邊增加了一個額外的 0。

延伸的二進位十進制交換碼 在早期的電腦中，IBM 發展了一個稱爲**延伸的二進位十進制交換碼** (Extended Binary Coded Decimal Interchange Code; EBCDIC)。這種交換碼使用 8 位元的樣式，所以它可以表達多達 256 個符號。

萬國碼 前述的編碼並沒有一個可以用來表達其他非英語系語言所屬的符號。因此，我們需要一個能夠容納更多的編碼法。一個由硬體和軟體廠商組合成的聯盟已經設計了一個使用 16 個位元、最多可以表達 65,536 (2^{16}) 個符號、稱爲**萬國碼** (Unicode) 的編碼法。

ISO 國際標準化組織 (International Organization for Standardization)，簡稱爲 ISO，已經設計了一個使用 32 位元樣式的編碼法，最多可以表達 4,294,967,296 (2^{32}) 個符號；肯定足夠表達現今世界上任何的符號。

數字 NUMBERS

一部電腦中，數字是用二進位系統（binary system）來表示的。在這個系統中，一個位元樣式（一個0和1的序列）表達一個數字。

圖像 IMAGES

在今日，圖像（images）在電腦中是以下列兩種方式之一來表達：位元圖繪圖法（bitmap graphic）或向量繪圖法（vector graphic）（圖 2.6）。

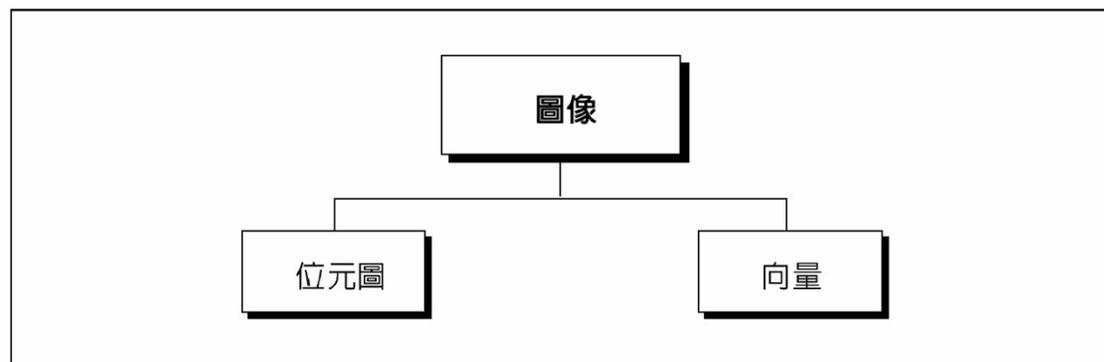


圖 2.6 圖像表達方法

位元圖繪圖法 BITMAP GRAPHIC

在這個方法中，一張圖像可劃分成一個**像素**（pixels）（**圖畫元素**；picture elements）矩陣，其中每一個像素就是一個小點。像素的多寡取決於一個叫做解析度的東西。

對於一張只有黑點與白點組成的圖像來說，一個單一位元的樣式已經足夠表達一個像素，一個0的樣式可用來表達黑色像素，而用1樣式來表達白色像素，如圖2.7所示。

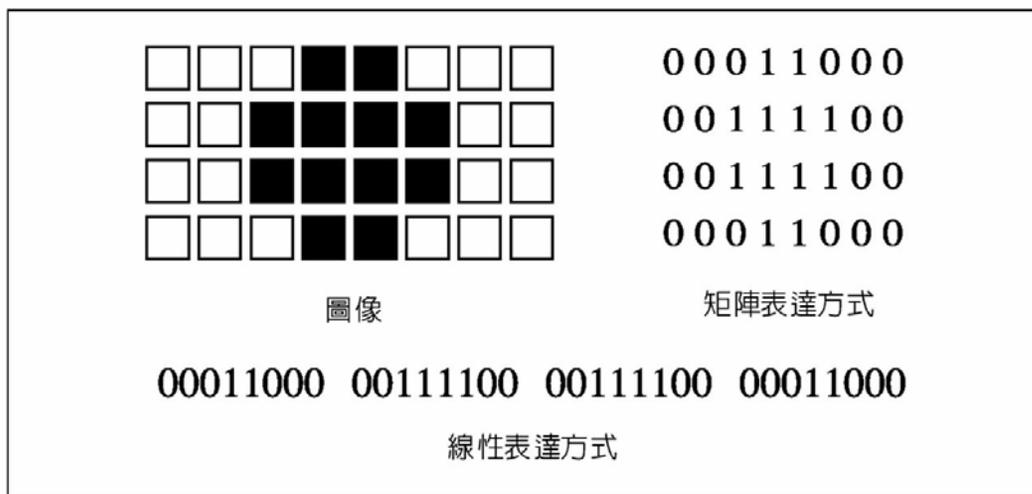


圖2.7 黑白圖像的位元圖表示法

如果一張圖像不是由純粹黑白兩種像素所組成，那麼就必須增加位元樣式的大小來表達它的灰階。例如，爲了顯示四個階層的灰階，可以使用兩個位元的樣式。一個黑色的像素可以用00 來表達，01 表達一個暗灰色像素，10 表達一個淺灰色像素，11 表達一個白色像素。

爲了表達彩色的圖像，每一個彩色的像素可分解成三個主要的顏色：紅色、綠色和藍色（RGB）。

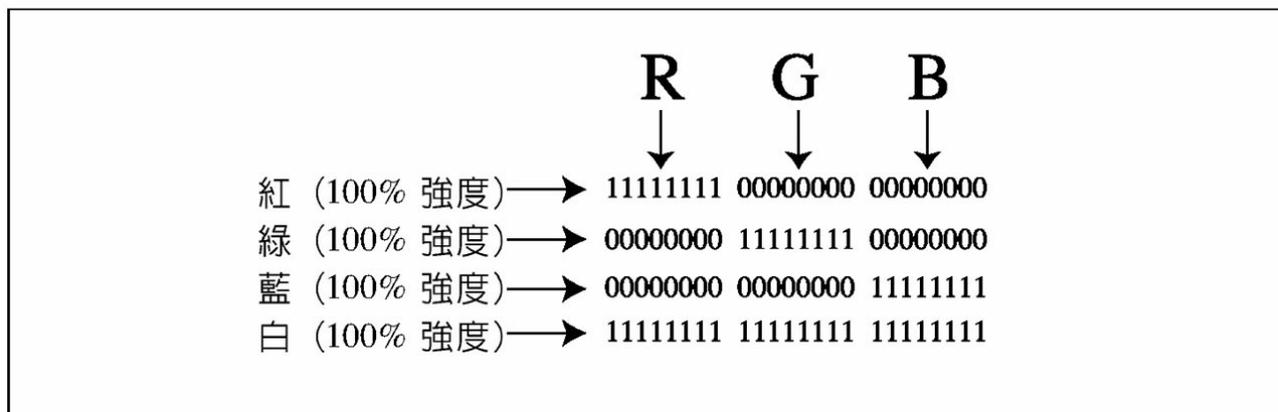


圖 2.8 彩色像素的表達方式

向量繪圖法 VECTOR GRAPHIC

位元圖繪圖法的問題在於，表達某一張特定圖像的精確位元樣式必須在電腦中被儲存起來。稍後，如果想要重新縮放這張圖片，就必須改變像素的數量，這將會產生鋸齒狀或粒狀的現象。然而，向量繪圖法並不儲存位元樣式，而是將一張圖像分解成曲線和直線的組合。每一條曲線或直線都是以一個數學方程式來表達。

音訊 AUDIO

音訊是一種聲音和音樂的表達方式。雖然儲存聲音和音樂並沒有一個標準，但是其構想就是將音訊轉換成**數位資料**（digital data），並使用位元樣式來儲存它們。

1. 在類比訊號中取樣。**取樣** (sampling) 指的是在等區間內測量訊號的數值。
2. 將取樣加以量化。**量化** (quantization) 指的是 (從一個集合中) 為取樣指定一個數值。舉例來說，一個取樣數值如果是 29.2，而這個集合是介於 0 到 63 的整數的話，就將這個取樣指定成數值 29。
3. 將量化的數值轉換成位元樣式。例如，數字 25 會被轉換成二進位樣式 00011001 (請參照第 3 章以瞭解數字如何轉換成樣式)。
4. 將二進位樣式儲存起來。

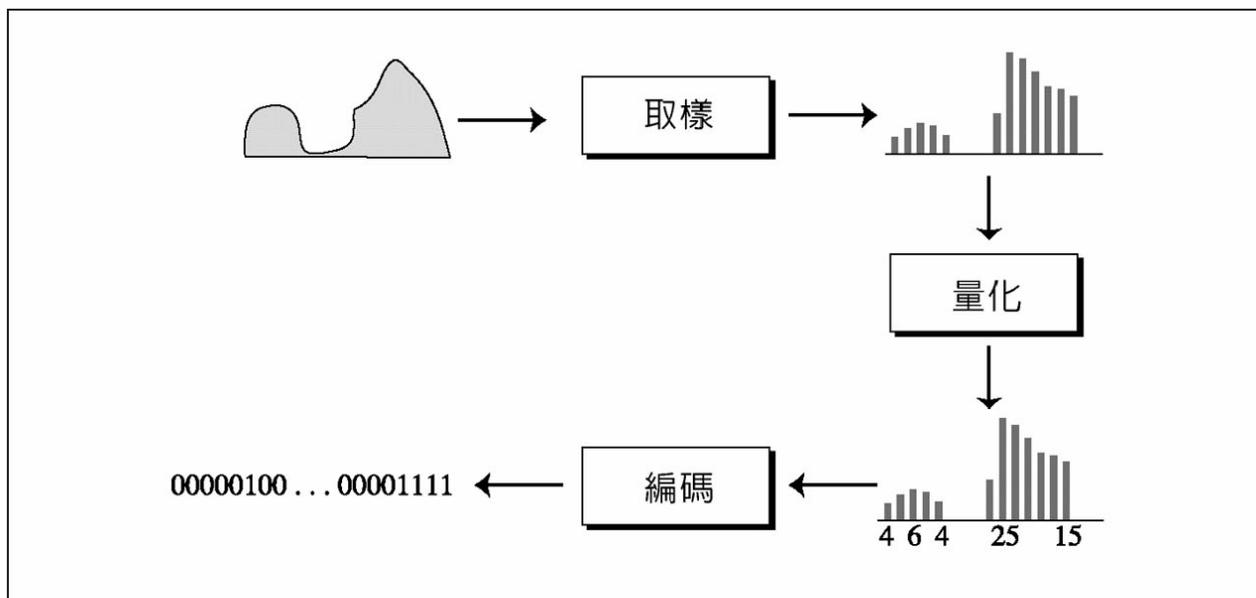


圖 2.9 音訊的表達方式

2.4 十六進位符號 *Hexadecimal Notation*

十六進位符號 (hexadecimal notation) 是以 16 為基礎 (hexadec 是希臘文中的 16)。這代表有 16 個符號 (十六進位數字) : 0,1,2,3,4,5,6,7,8,9, A,B,C,D,E 和 F 。

每一個十六進位數字可以表達四個位元，而四個位元可以表達一個十六進位數字。表 2.2 顯示了一個位元樣式和十六進位數字間的關係。

位元樣式	十六進位數字	位元樣式	十六進位數字
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

表 2.2 十六進位數字

轉換 CONVERSION

從一個位元樣式轉換到十六進位是藉由將樣式以四個為一組的方式分組後，找出每四個位元一組所對應的十六進位數值。而十六進位到位元樣式的轉換，則是將每一個十六進位數字轉換成其對應的四個位元的數值（圖 2.10）。

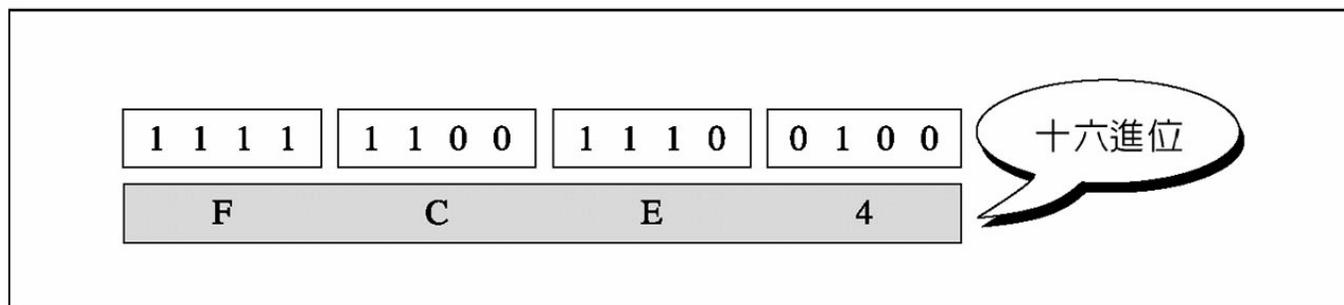


圖 2.10 二進位到十六進位和十六進位到二進位的轉換

2.5 八進位符號 *Octal Notation*

八進位符號是以8為基礎的（oct是希臘文中的8）。這表示我們有八個符號（八進位數字）：0,1,2,3,4,5,6和7。

每一個八進位數字可以用三個位元來表達，而三個位元可以用一個八進位數字來表達。表2.3顯示了一個位元樣式與八進位數字之間的關係。

位元樣式	八進位數字	位元樣式	八進位數字
000	0	100	4
001	1	101	5
010	2	110	6
011	3	111	7

表2.3 八進位數字

轉換 CONVERSION

從一個位元樣式轉換到八進位是藉由將樣式以三個為一組的方式分組後，找出每三個位元一組所對應的八進位數值。而八進位到位元樣式的轉換，則是將每一個八進位數字轉換成其對應的三個位元的數值（圖 2.11）。

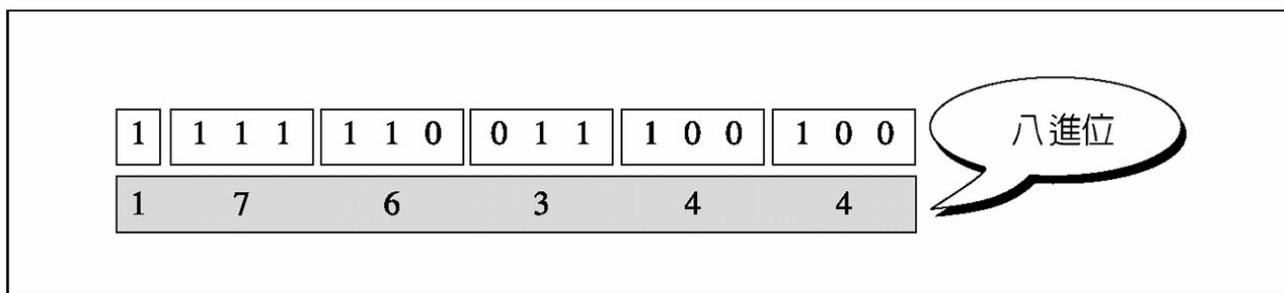


圖 2.11 二進位到八進位和八進位到二進位的轉換