讓電腦認識各種資料

在科幻電影裡面常常看到，闖進電腦的世界裡，就是一大堆的 0 和 1，但是就我們平常使用電腦的經驗，在螢幕上看到的是畫面，從喇叭聽到的是聲音，點點滑鼠、打打鍵盤，怎麼看都不見 0 和 1 的蹤影，究竟是怎麼一回事？如果電腦裡面真的是由一大堆的 0 和 1 組成，又是怎麼組成的呢？本章便是要帶您瞭解電腦裡面如何儲存各種資料的來龍去脈，首先我們從「數字」這種資料開始介紹。

第一節、如何只用 0 與 1 表示各種數字？

由於各行各業對數字計算的迫切需要，例如建築師要估量房屋的建造穩固與否、航海師要根據不同的海象決定航行角度…等等，數字這種資料在電腦中的出現比其他資料都還要早。而既然要利用電腦自動計算數字，在這之前得先讓電腦認識各種數字，也就是把數字用某種方式表示給電腦知道。

馮紐曼（John von Neumann, 1903-1957）在 1945 年與其他科學家聯名發表了一篇電腦設計的報告《EDVAC 的第一份草案》，明確闡述利用 0 與 1 表示各種數字的方法，以及採用二進制取代十進制運算的優點，成為現代電腦設計的基本原則。為什麼電腦要用只有 0 與 1 的二進制，而不用人類熟悉的十進制呢？光是 0 與 1 要怎麼表示像 2、3、4…等其它數字呢？在回答這些問題之前，我們先來看看在馮紐曼的那個時代，已經有哪些表示數字的方法。

圖 1-1：約翰．馮紐曼

在馮紐曼之前已經有許多科學家製作出十進制的計算機器，也就是我們所稱的電腦，但是並沒有人想到要讓電腦採用二進制計算。十九世紀，英國數學家巴貝奇（Charles Babbage, 1791-1871）利用標示有 0 到 9 的手工齒輪來做函數求值的數字運算。

歷史影片

差分機使用手工齒輪以十進制表示數值（0:47）

https://www.youtube.com/watch?v=LE2gECwwx8g

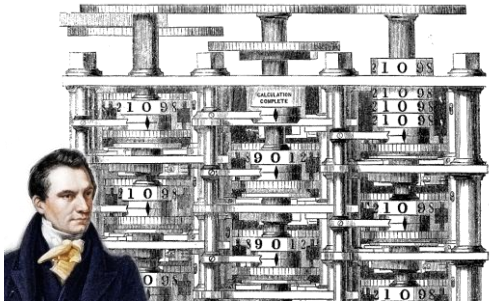


圖 1-2：查爾斯．巴貝奇與刻有 0～9 齒輪的計算機器

到了 1946 年，美國陸軍與賓州大學合力建造一台名為 ENIAC 的電腦，以九個一排的真空管來做計算。

歷史影片

ENIAC 使用真空管以十進制表示數值（0:58）

https://www.youtube.com/watch?v=315gJ-4o1fE

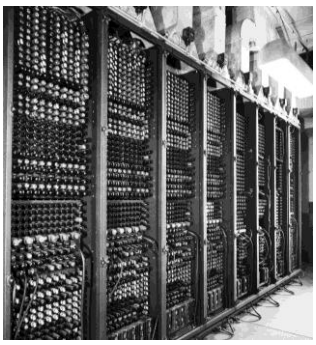


圖 1-3：ENIAC 電腦

真空管在此可以想作是具有控制電子流動功能的電燈泡，ENIAC 的真空管仿照齒輪有十個刻度分別表示數字 0～9，也是透過九個一排的方式來表示數字 0～9（如圖 1-4）。

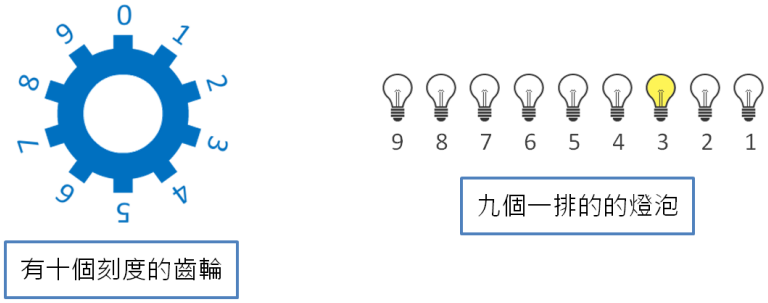


圖 1-4：齒輪與真空管的異同

當燈泡全暗的時候，表示數字 0，亮最右邊一顆燈泡，表示數字 1，亮右邊數來第二顆燈泡，表示數字 2，以此類推，每個位置的燈泡分別代表著數字 1、2、3、⋯、9，如表 1-1 所示，每排燈泡總共會有 0、1、2、⋯、9 共十種狀態。

表 1-1：以燈泡示意，利用十進制（共 9 顆燈泡）表示數字 0～9

燈泡									表示數字
9	8	7	6	5	4	3	2	1	0
9	8	7	6	5	4	3	2	1	1
9	8	7	6	5	4	3	2	1	2

									3
9	8	7	6	5	4	3	2	1	
									4
9	8	7	6	5	4	3	2	1	
									5
9	8	7	6	5	4	3	2	1	
									6
9	8	7	6	5	4	3	2	1	
									7
9	8	7	6	5	4	3	2	1	
									8
9	8	7	6	5	4	3	2	1	
									9
9	8	7	6	5	4	3	2	1	

採用通電的真空管代替機械齒輪換取了更快的計算速度，但還是稍嫌繁複，後來加入 ENIAC 建造計畫的馮紐曼，在當時他注意到真空管有個特性：僅有「**充滿電**」，或「**沒充電**」的兩種狀態。於是馮紐曼想把每排燈泡的狀態，從十種簡化到只有兩種，讓原本一排九顆燈泡，大幅縮減到一排一顆燈泡，暗的時候表示 0，亮的時候表示 1，如表 1-2 所示。

表 1-2：以燈泡示意，利用二進制（共 1 顆燈泡）表示數字 0、1

燈　　泡	表示數字
	0
	1

那麼，接下來的數字 2、3、4…等要怎麼辦呢？先來看看當十進制遇到數字 10、11、12…等數字的時候是怎麼做的：一排九顆燈泡在遇到數字 10 時，需要第二排燈泡來分別表示個位數與十位數，如表 1-3 在「十位數」那排，亮最右邊一顆燈泡，表示十位數字 1，在「個位數」那排全暗，表示個位數字 0，合起來看就是在表示數字 10；再接下來的數字 11，將「個位數」位置 1 的燈泡由暗轉亮，成為「十位數」亮位置 1、「個位數」亮位置 1，合起來表示數字 11。

表 1-3：以燈泡示意，利用十進制表示數字 10

燈泡									位數
9	8	7	6	5	4	3	2	1	個位數
9	8	7	6	5	4	3	2	1	十位數

同理，仿照十進制的做法，在一排一顆燈泡遇到數字 2 時，就是需要進位的時候了（這也是「二進制」的名稱由來），需要有第二排燈泡來表示不同的位數。在二進制裡面沒有個位數、十位數的稱呼，先暫且用 A 位數、B 位數說明，如表 1-4，數字 2 由 B 位數亮一顆燈泡、A 位數暗一顆燈泡合起來表示；再接下來的數字 3，將原本表示數字 2 時 A 位數由暗轉亮，成為 A、B 位數皆亮，合起來表示數字 3。

表 1-4：以燈泡示意，利用二進制表示數字 2

燈泡	位數
	A 位數
	B 位數

而 A 位數和 B 位數的意思，可以由觀察十進制的位數有個位數、十位數…等發現，不同的位數其實就是在表示十的次方數，例如個位數便是 10^0、十位數 10^1、百位數 10^2…，以此類推，在二進制裡面的 A 位數便是 2^0、B 位數便是 2^1，接下來的位數還有 2^2、2^3、2^4…等等。如此一來，儘管只有兩種狀態，也能經由 2^0、2^1、2^2、2^3…等位數來表示各種數字。以數字 0～9 為例，表示的方法如表 1-5（燈泡改為橫向顯示）。

表 1-5：以燈泡示意，利用二進制（共 4 顆燈泡）表示數字 0～9

燈泡				表示數字	說明
2^3	2^2	2^1	2^0	0	
2^3	2^2	2^1	2^0	1	$2^0 = 1$
2^3	2^2	2^1	2^0	2	$2^1 = 2$

2^3	2^2	2^1	2^0	3	$2^0 + 2^1 = 2 + 1 = 3$
2^3	2^2	2^1	2^0	4	$2^2 = 4$
2^3	2^2	2^1	2^0	5	$2^2 + 2^0 = 4 + 1 = 5$
2^3	2^2	2^1	2^0	6	$2^2 + 2^1 = 4 + 2 = 6$
2^3	2^2	2^1	2^0	7	$2^2 + 2^1 + 2^0 = 4 + 2 + 1 = 7$
2^3	2^2	2^1	2^0	8	$2^3 = 8$
2^3	2^2	2^1	2^0	9	$2^3 + 2^0 = 8 + 1 = 9$

而燈泡充滿電或沒充電的狀態，馮紐曼統一用「0」來表示沒充電、「1」來表示充滿電，所以數字 0 的燈泡「暗、暗、暗、暗」就可以用「0000_2」來表示，數字 9 的燈泡「亮、暗、暗、亮」就可以用「1001_2」來表示；其中那小小的「$_2$」表示這是個二進制的數，而在二進制中的每個 0 或每個 1 稱為一個位元（**bit**）。

原本在使用十進制時，每一顆燈泡表示一個特定的數，所以在表示任何數字時，每排總會有其他八顆燈泡是暗著沒在使用；現在使用二進制，每顆燈泡分別表示 2^0、2^1、2^2、2^3…等不同位數，在表示一個數字時，所有燈泡都會用得上，這讓同樣數量的燈泡可以表示更多的數字。舉例來說，同樣都是九顆燈泡，十進制只能表示 0～9 共十個數字，在二進制可以表示 0～511 共 512 個數字（詳見延伸學習「表示不同數字所需要的位數」）。

採用二進制的電腦，能配合真空管僅有兩種狀態的特性，而且狀態的數量被簡化，能加快運算的速度。儘管將電腦從十進制改用二進制無法徹底解決真空管容易燒壞的問題，但換用二進制不僅大幅降低真空管使用的數量以及找出燒壞真空管的時間，還對計算結果的正確性與速度有著功不可沒的影響，所以至今我們仍然使用著馮紐曼所提出的二進制電腦設計。

換你試試看

1. 在下面採用二進制四顆燈泡中，請你用螢光筆塗亮所需的燈泡位置與數量，來表示數字 10。

燈泡				
位數	2^3	2^2	2^1	2^0

2. 承上題，數字 10 用四顆燈泡即可表示，那麼數字 16 需要幾顆燈泡呢？為什麼？

_____

_____

3. 把亮暗燈泡換用 0 與 1 表示：請你將以下採用二進制四顆燈泡，改用「0」表示不亮的燈泡、「1」表示亮的燈泡。

燈泡				
位數	2^3	2^2	2^1	2^0

以 0 與 1 表示的結果：_____

4. 假設你現在手邊擁有五顆燈泡，並想利用這些燈泡來表示數字。如果採用十進制，最大能表示的數字是多少？又如果採用二進制，最大能表示的數字是多少？

_____

_____

_____

想一想

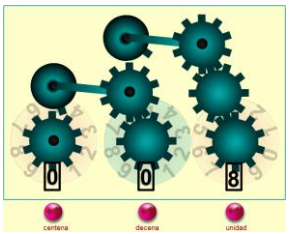
以現在的科技來說，製造精密的十進制電子元件在技術上已經不是問題，為什麼現在的電腦還是採用二進制？

延伸學習

- **利用齒輪計算數字**

  連結至：

  http://recursostic.educacion.es/descartes/web/matemagicas/pages/truc_mat/textes/pascaline.swf，點選個位、十位或百位的「+1」，觀看齒輪執行加法運算與進位的過程。



- **利用樂高製作的差分機**

  實機操作影片（*1:19*）見：

  http://www.youtube.com/watch?v=i_u3hpYMySk

- **更多的位元**

  位元（bit）是電腦中資料儲存的最小單位，不過由於資料量越來越大，為了計數方便，將每 8 個位元一數稱為位元組（**byte**）。隨著電腦的演進資料量倍增的情況，又出現了如 KB、MB、GB 等單位的出現，如下表所示。

單位	名稱	大小
KB	Kilo Byte	2^{10} Byte ≈ 10^3 Bytes
MB	Mega Byte	2^{20} Byte ≈ 10^6 Bytes ≈ 10^3 KB
GB	Giga Byte	2^{30} Byte ≈ 10^9 Bytes ≈ 10^6 KB ≈ 10^3 MB
TB	Tera byte	2^{40} Byte ≈ 10^{12} Bytes ≈ 10^9 KB ≈ 10^6 MB ≈ 10^3 GB

- **表示不同數字所需要的位元**

  在二進制要表示數字 511，從 2^0、2^1…一個個看，到 $2^8 = 256$ 仍僅能表示數字 0～255，還不夠表示數字 511，而 $2^9 = 512$，能表示 512 個數字，即從 0～511，足夠表示數字 511 了，所以需要 9 個位元才夠表示數字 511。

  換一個角度來說，9 個位元最大可以表示的數字即為 111111111（九個 1），而 111111111_2 的十進制值則為：$2^8 + 2^7 + 2^6 + \cdots + 2^1 + 2^0 = 256 + 128 + 64 + \cdots + 2 + 1 = 511$，所以 9 個位元最大表示數字為 511。

  或可以從下面的例子來找出位元與表示數字範圍的規律：

  - 1 位元所能表示最大的正整數是 1_{10}（表示 0、1）
  - 2 位元所能表示最大的正整數是 3_{10}（表示 0、1、2、3）
  - 3 位元所能表示最大的正整數是 7_{10}（表示 0、1、2、…、7）

  以此類推，N 位元所能表示最大的正整數是 2^N-1_{10}

- **生活應用：二進制手錶**

  你知道下面的手錶分別表示幾點幾分嗎？

（圖片來源：01THEONE）

表示時間	11:17	7:23	11:39
計算方式	時：8 + 2 + 1 = 11 分：16 + 1 = 17	時：4 + 2 + 1 = 7 分：16 + 4 + 2 + 1 = 23	時：8 + 2 + 1 = 11 分：32 + 4 + 2 + 1 = 39

- **十進制與二進制轉換器－Windows 小算盤**

  於 Windows 作業系統中操作【開始/所有程式/附屬應用程式/小算盤】，接著點選【檢視/程式設計師】，輸入十進制數字後，點選左方之「2 進位」即可轉換。

- **馮紐曼的小故事**

  在一次在宴會上，女主人向馮紐曼提出一個算題：有兩輛火車面對面行駛，相對速度為每小時三十哩，假設兩火車在一分鐘後相撞，此時有個時速六十哩的超快蒼蠅從一輛火車的車頭往另一輛飛，等它抵達另一輛火車車頭時往回飛，就這樣來回不停地飛，直到最後被壓扁為止，請問蒼蠅一共飛多遠？

  蒼蠅飛的總距離可以由來回飛的各個小片段加總起來而得，這是一個無窮級數的算法，雖不難，但有些繁瑣。再仔細看這問題，既然火車一分鐘後相撞，而時速六十哩的蒼蠅一分鐘可以飛一哩，所以答案當然就是一哩。

  女主人一問完問題，馮紐曼馬上回答：「一哩。」

  女主人驚訝地說：「數學家通常用無窮級數花幾分鐘來解，而忽略的速解的竅門。」

  馮紐曼說：「什麼竅門？我就是用無窮級數算的啊！」

  馮紐曼的心算速度之快，快到在他的有生之年，沒有一台電腦的速度超

過他的心算速度。 雖然如此，他還是熱衷於電腦的設計和發展。或許他知道，雖然電腦算不贏他，但是對旁人而言還是很重要的工具。而且，他當然知道，假以時日電腦的速度總會超過他的速度。

ENIAC 負責邏輯和程式設計的數學家 Goldstine 在火車站遇到馮紐曼。他們談話的氣氛，原本只是說說天氣很好，後來 Goldstine 提到 ENIAC 計畫，那種氣氛就變得像博士班口試一樣。本來 ENIAC 是個國防機密，但是馮紐曼參與了更機密的勾當：原子彈，所以輕易就參與了 ENIAC 的工作群。

馮紐曼對今日的電子計算機設計，有決定性的影響，但是他也有看錯的時候，例如他不認為像 FORTRAN 這樣的高階語言是必要的，他認為直接用機器碼來寫程式就好了。馮紐曼有一個博士班學生 Backus，因為用機器碼太煩了，著手設計組合語言。馮紐曼知道了以後大為惱火，認為他不該把寶貴的時間浪費在這種無用的工具上。

故事來源：

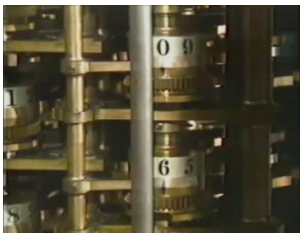
- 單維彰（2000）。計算機概論十六講擴編教材：John von Neumann。國立中央大學數學系。http://libai.math.ncu.edu.tw/bcc16/pool/3.02.shtml
- 趙坤茂（2004）。計算機概論摘錄。國立臺灣大學。http://www.csie.ntu.edu.tw/~kmchao/life/bcc.htm

教學補充資源

- 巴貝奇與愛達的故事

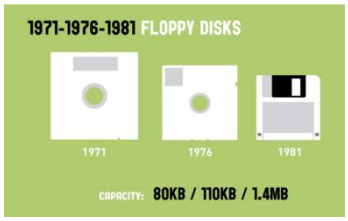
  巴貝奇設計差分機，以及與愛達相互討論出分析機的故事影片見：

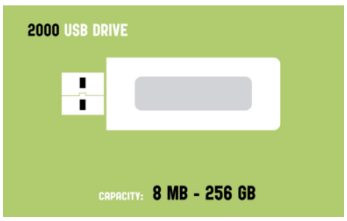
  http://www.youtube.com/watch?v=QVxbNZWLP60（*7:42*）



- 數位資料容量單位的演進

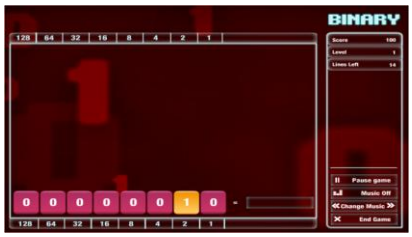
  影片見：https://www.youtube.com/watch?v=ah14LEFKe8Q（*1:44*）

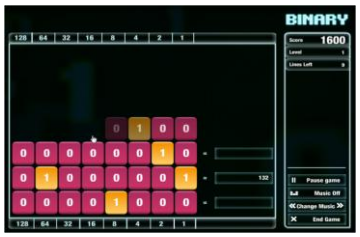




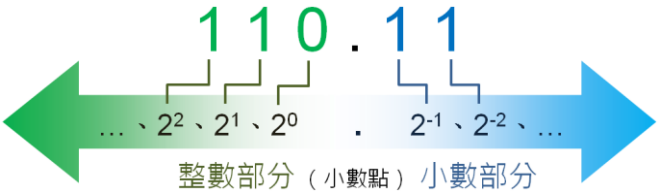
- 二進制消磚塊遊戲

  連結至：http://forums.cisco.com/CertCom/game/binary_game_page.htm，點選「Start」，即可開始二進制與十進制轉換的消磚塊遊戲。





- 小數的表示方法

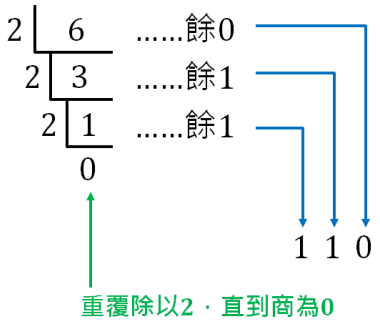


$(110.11)_2 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$

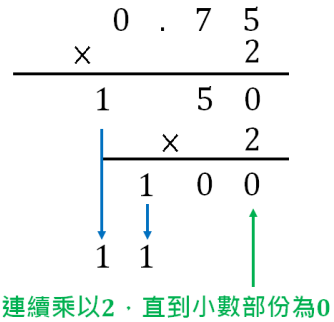
$= 4 + 2 + 0 + 0.5 + 0.25 = (6.75)_{10}$

- **快速轉換十進制與二進制的方法**

整數部分：



小數部分：



- **負數的表示方法**

以 9 個位元來說，原本表示正整數 0～511，可以拿其中一個位元用來表示該數的符號是正或負，接下來的 8 個位元再來表示數值。

詳細的轉換步驟如下：

1. 將十進制的數轉為二進制
2. 將二進制數的 0 與 1 互換（此步得到的結果為「一補數」）
3. 將 2 的結果加 1（此步得到的結果為「二補數」）

例如求-20 的二補數（以六位元為例）

1. $-20_{10} = \underline{0}10100_2$（補 0 到六位元）
2. 101011
3. $101011_2 + 1_2 = 101100_2$

至於上面提到的「一補數」是什麼呢？上例中將 010100 的 0、1 互換得到 101011，只將 0、1 互換而沒有再加 1 即為原二進制數的「一補數」。那為什麼不用一補數就好，還需要多一道步驟的二補數呢？這是因為然而由於一補數中存在多餘的負零(以六位元來說，000000 與 111111 都是在表示 0，但一個表示+0、一個表示-0)等問題，一補數並未像二補數一樣被廣泛應用。

- **參考解答：換你試試看**

1.

燈泡				
位數	2^3	2^2	2^1	2^0

2. 5 顆，24 只能表示 0～15，25 能表示 0～31，足夠表示數字 16。
3. 1100

4.

(1) 5，五顆燈泡全暗時表示 0，亮不同顆燈泡分別表示 1、2、3、4、5，五顆燈泡在十進制時能表示數字 0～5；

(2) 31，2^5—1 = 32—1 = 31，五顆燈泡在二進制時能表示數字 0～31。

※ 以五顆燈泡表示十進制數值的另解探討

回答結果	思路與原因
4	每顆燈泡分別表示 0、1、2、3、4，沒有利用全暗的燈泡表示 0，共可表示 0～4 的數。問題：最接近正解，但多耗費一顆燈泡來表示 0。
32	利用排列組合，挑選一顆、挑選兩顆、挑選三顆、挑選四顆與挑選五顆的情況分別表示不同數字，由 C5 取 0 + C5 取 1 + C5 取 2 + C5 取 3 + C5 取 4 + C5 取 5 = 32，包含 0 在內共可表示 0～31 的數（而非 32）。問題：此種表示方式，類似於文字編碼的表示法，將 0～31 視為 32 種不同的「符號」而非「數值」（沒有進制的概念，無法進行四則計算）。
15	每顆燈泡表示 1～5 的數，燈泡全暗時表示 0，再利用數值相加的方式，例如亮 5 與 1 兩顆燈泡表示 6，五顆全亮則表示 15，包含 0 在內可表示 0～15 的數。問題：同一個數會有不唯一的表示方式，增加編碼與解碼的複雜度，且同樣沒有進制的概念。
10	燈泡由右往左看時，分別表示 1、2、3、4、5，由左往右看時，分別表示 6、7、8、9、10，全暗時表示 0，最大表示 10，最小表示 0。問題：需要多一顆燈泡紀錄現在是要由右或由左看，且同樣沒有進制的概念。
98765 或 54321	每顆燈泡分別表示不連續的 90000、8000、700、60 與 5，最大（全亮）表示 98765、次大表示 98760…，最小（全暗）表示 0。問題：許多介於之間的數值無法表示。
99999 或 11111	一由每顆燈泡只能表示 0 或 9，可不連續的表示最大（全亮）99999，最小（全暗）0；二由 105-1 而來，受到二進制中能表示的最大數為 2N-1 的影響。問題：每顆燈泡只有亮或暗兩種狀態，同一顆燈泡無法表示 0～9 的十種狀態。
45	每顆燈泡各自表示 0～9，有 5 顆燈泡，5 * 9 = 45。問題：在 0 也是一種數值以及排列組合的學習上有迷思概念，如果每顆燈

	泡能有 0～9 的十種狀態，結果應為 10^5 = 10000，如果每顆燈泡有 1～9 的九種狀態，也應為 9^5 = 59045，但每顆燈泡同樣只有亮或暗兩種狀態，無法表示九或十種狀態。

- **參考解答：想一想**
  - 體積的觀點：如果還用十進制的話，需使用較多的電子元件，會佔用很多體積空間。
  - 技術的觀點：如果要改用十進制，不僅是電子元件要改，整台電腦的設計架構也都要修正。
  - 速度的觀點：使用兩種狀態的二進制最精簡，利用 shift 可以快速做乘除法運算，利用布林代數可以做邏輯判斷。

圖片來源

- 圖 1-1： von Neumann, John -- Kids Encyclopedia.
  http://kids.britannica.com/comptons/art-164722/John-von-Neumann
- 圖 1-2：研究者修改繪製自
  Bios (2011). Charles Babbage, pioneer of machine computing, 1833 - Big Other. Wordpress.
  http://bigother.com/2011/04/12/the-other-culture/charles-babbage-pioneer-of-machine-computing-1833/
  Georgi Dalakov (2013). The Differential Engine of Charles Babbage. History of Computers.
  http://history-computer.com/Babbage/DifferentialEngine.html
- 圖 1-3：ENIAC -- Historical Marker. http://explorepahistory.com/hmarker.php?markerId=1-A-2F3
- 圖 1-4：研究者自行繪製。燈泡之素材來源：
  Icon search engine and market place - Iconfinder
  https://www.iconfinder.com/

第二節、如何用 0 與 1 表示文字符號？

在能利用最簡化的二進制表示數字資料之後，1950 年代，文字、影像與音訊表示法分別在十分相近的時間點，從不同角度出發，各自開始制定所需的編碼標準（如圖 2-1）。

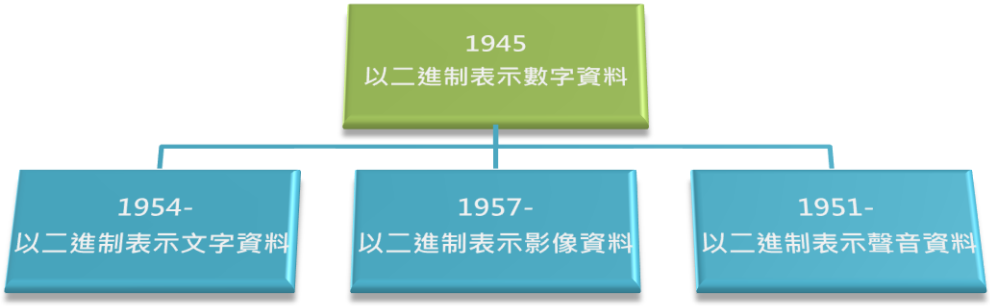


圖 2-1：數字表示法影響了文字、影像及音訊編碼的出現

其中利用二進制表示文字符號的「文字編碼」，1967 年美國國家標準局制定了 **ASCII 碼**，給予每一個英文字母及數字符號一個特定的編碼對照表，統一電腦與電腦之間互相傳遞文字資料的標準，影響了現代萬國碼的發展。不過，只是數字的 0 與 1 要怎麼表示像 A、B、C…的英文字母，甚至是中文字呢？本節要帶您學習文字在電腦中的表示方法，並回答這個問題。

在 ASCII 碼之前的文字編碼，最早可追溯到電報通信的摩斯電碼，利用「·（點）」與「－（劃）」來表示 26 個英文大寫字母以及 0～9 的數字（如）。

歷史影片

無線新聞的「News Today」摩斯密碼（1:05）

https://www.youtube.com/watch?v=u493fX2hYgU

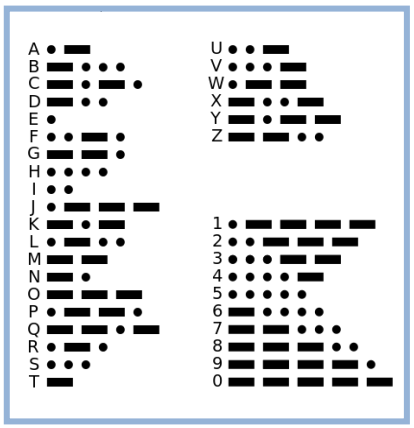


圖 2-2：摩斯電碼編碼表

摩斯電碼為了加快電報撥打與傳輸的速度，讓英文字母根據使用頻率採取不定長度的編碼表示，例如最常用的 E 只用一個符號「·」表示，少用的 Z 則用四個符號「－－··」表示。

摩斯電碼採用兩種符號表示各英文字母及數字，貼近二進制的特性，但在當時有個問題，例如接收到「・－」時，很難確定是在表示一個英文字母 A（・－），還是兩個英文字母 E（・）與 T（－），如圖 2-3。

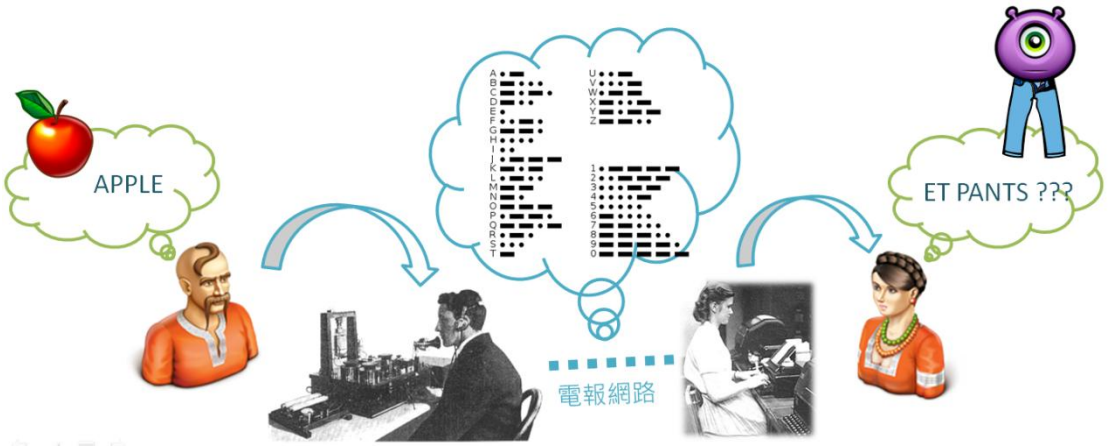


圖 2-3：摩斯電報傳輸出現誤解的情境示意圖

所以接下法國發明家博多（Émile Baudot, 1845-1903）在設計博多電報時，改進摩斯電碼容易傳輸錯誤的問題，統一將所有英文字母及數字符號通通採用五個位元的編碼表示，例如英文字母 A 的編碼為「●●○○○」、B 的編碼為「●○○●●」等（如圖 2-4）。

		1	2	3	4	5			1	2	3	4	5
A		●	●				Q	1	●	●	●		●
B	?	●			●	●	R	4		●		●	
C	(		●	●	●		S	'	●		●		
D	²	●			●		T	5					●
E	3	●					U	7	●	●	●		
F	½	●		●	●		V	)		●	●	●	●
G	¾		●		●	●	W	2	●	●			●
H	⅝			●		●	X	£	●		●	●	●
I	8		●	●			Y	6	●		●		●
J	⅞	●	●		●		Z	-	●				●
K	⅜	●	●	●	●		FIG SPACE	FIG SPACE	●	●		●	●
L	/		●			●	LTR SPACE	LTR SPACE			●		
M	,			●	●	●	LINE PAGE	LINE PAGE				●	
N	–			●	●		✳	✳	●	●	●	●	●
O	9				●	●	COL	COL		●			
P	0		●	●		●							

圖 2-4：博多電報編碼表

後來各個國家與各家公司在利用二進制替電腦設計文字編碼時，延續博多電報固定長度的編碼方式，分別制定出各自的編碼表。由於每套編碼表都是各字制定，總計當時光是英文、數字等西歐字符就存在六十多套編碼表，以同樣是英文字母 A 為例，美國 IBM 公司的 BCD 編碼是「010001」，歐洲電腦製造協會的 DEC 編碼是「010001」，蘇聯的 GOST 編碼則是「010000」，時常需要互相轉換而相當不便。

於是美國國家標準局邀請各方編碼專家召開會議，希望讓每一個英文字母及數字符號都有一個特定的編碼對照，而且可以廣泛通用在所有電腦上。經過多次的討論，終於在 1967 年提出了一套每台電腦都可以共通使用的編碼標準，也就是 ASCII 碼（如表 2-1）。在 ASCII 碼中，使用了七個位元來表示英文大小寫字母、數字及符號，另外還有一些如 Enter、Esc、Delete 等用於控制的不可見字元。其中英文字母的部分，大小寫按照字母順序有著各字的編碼，例如 A 的編碼為 100

0001，往後推一個 100 0010（$100\ 0001_2 + 1_2 = 100\ 0010_2$）即是 B 的編碼，反過來說，C 的編碼為 100 0011 往前推兩個（$100\ 0011_2 - 10_2 = 100\ 0001_2$）即為 A 的編碼），小寫則是接在大寫之後，從 a 的編碼 110 0001 開始，到 z 的 111 1010；而數字的編碼，前三碼為 011，後四碼即為該數字的二進制值，例如 3 的編碼 011 0011，將 0011_2 轉換為十進制即為 3。

表 2-1：ASCII 編碼表：可見字元及其編碼

字元	編碼	字元	編碼	字元	編碼	字元	編碼
(空白)	010 0000	8	011 1000	P	101 0000	h	110 1000
!	010 0001	9	011 1001	Q	101 0001	i	110 1001
"	010 0010	:	011 1010	R	101 0010	j	110 1010
#	010 0011	;	011 1011	S	101 0011	k	110 1011
$	010 0100	<	011 1100	T	101 0100	l	110 1100
%	010 0101	=	011 1101	U	101 0101	m	110 1101
&	010 0110	>	011 1110	V	101 0110	n	110 1110
'	010 0111	?	011 1111	W	101 0111	o	110 1111
(	010 1000	@	100 0000	X	101 1000	p	111 0000
)	010 1001	A	100 0001	Y	101 1001	q	111 0001
*	010 1010	B	100 0010	Z	101 1010	r	111 0010
+	010 1011	C	100 0011	[	101 1011	s	111 0011
,	010 1100	D	100 0100	\	101 1100	t	111 0100
-	010 1101	E	100 0101	]	101 1101	u	111 0101
.	010 1110	F	100 0110	^	101 1110	v	111 0110
/	010 1111	G	100 0111	_	101 1111	w	111 0111
0	011 0000	H	100 1000	`	110 0000	x	111 1000
1	011 0001	I	100 1001	a	110 0001	y	111 1001
2	011 0010	J	100 1010	b	110 0010	z	111 1010
3	011 0011	K	100 1011	c	110 0011	{	111 1011
4	011 0100	L	100 1100	d	110 0100	|	111 1100
5	011 0101	M	100 1101	e	110 0101	}	111 1101
6	011 0110	N	100 1110	f	110 0110	~	111 1110
7	011 0111	O	100 1111	g	110 0111		

而文字編碼的運作原理，在電腦內部便是以 0 與 1 的二進制來儲存與傳輸，當要顯示在螢幕上的時候，再利用該文字的 ASCII 碼去資料庫找出該文字的字型圖樣顯示出來。以英文字母 A 為例，ASCII 碼中 A 的編碼為「100 0001」，利用鍵盤輸入 A 時，電腦內部便是以「100 0001」來儲存或傳輸，要顯示於螢幕時，再

去資料庫中找出 A 的字型圖樣，然後在螢幕上顯示「A」（如圖 2-5）。

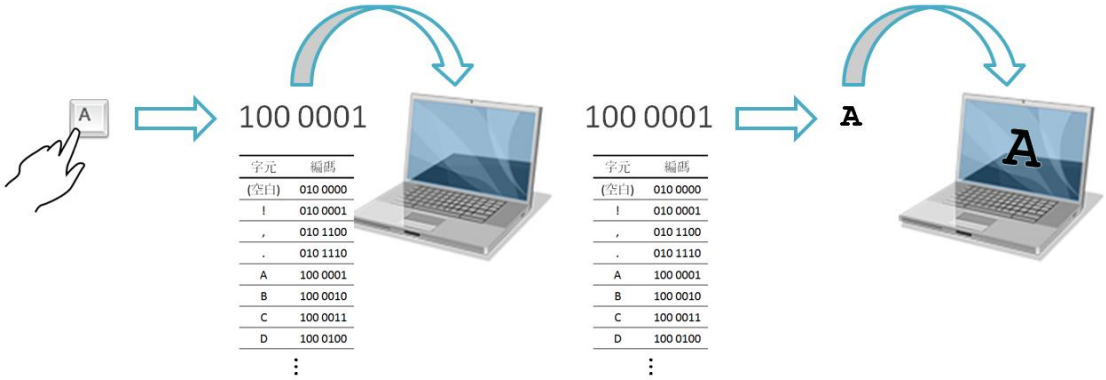


圖 2-5：文字編碼的運作原理

到了 1980 年代，美國、西歐以外的東亞、中歐等地區，也開始有使用電腦的需求；但在 ASCII 不足以表示各種文化的語言文字的情況下，各國又開始各自制定文字編碼，例如日本制定了日文編碼 Shift JIS、台灣制訂了正體中文編碼 BIG5（如表 2-2）、中國大陸制定了簡體中文編碼 GB2312、韓國制定了韓文編碼 KS X…等，再經歷過一次次制定地區統合方案後，能涵蓋世界上大部分文字符號的萬國碼（**Unicode**）終於成形。由多個國家政府和各大軟體商代表組成的 Unicode 聯盟，制定出包含十萬字以上的萬國碼（現今還在持續發展中），讓全世界的電腦都能遵循使用。

表 2-2：BIG5 編碼表：台灣常見前十姓氏之編碼

字元	編碼	字元	編碼
陳	1011 0011 1010 1111	王	1010 0100 1111 1101
林	1010 1010 0100 1100	吳	1010 0111 0110 0100
黃	1011 0110 1100 0000	劉	1011 1100 0100 0010
張	1011 0001 0110 1001	蔡	1011 1101 1011 0010
李	1010 0111 1111 0101	楊	1011 0111 1010 1000

文字編碼在生活中最常看到的實例就是網頁的編碼，一般來說正體中文網頁不外乎就是使用萬國碼和 BIG5 兩種，我們在瀏覽網頁時，需要使用與該網頁相同的編碼，才有辦法順利解讀，否則如果我們使用 BIG5 的編碼去觀看萬國碼的網頁時，就會出現亂碼的情況，這時候就要切換回正確的編碼才有辦法順利閱讀（如圖 2-6）。

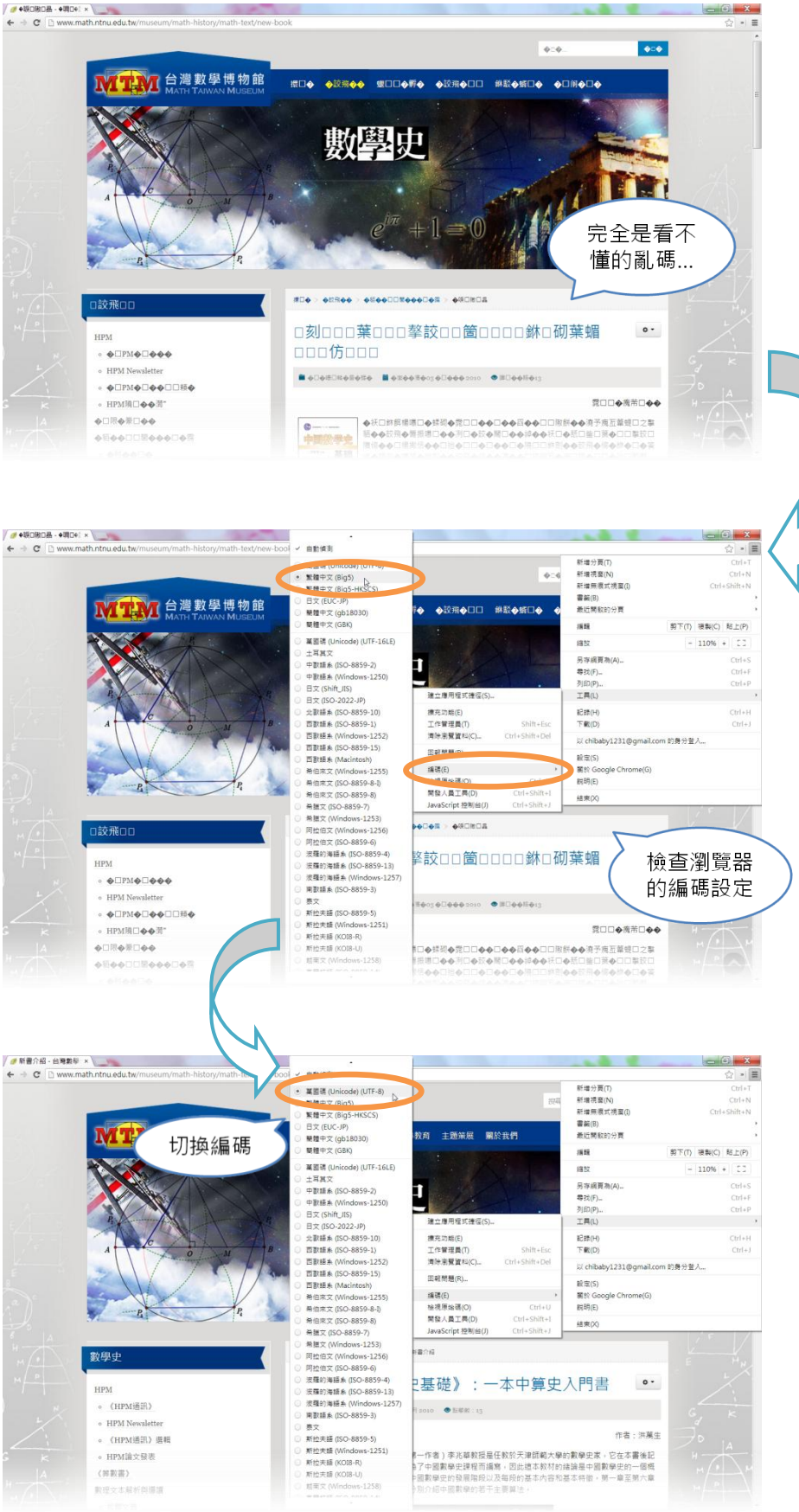


圖 2-6：切換網頁的文字編碼

換你試試看

1. 將英文句子「Just do it!」以 ASCII 碼表示：

2. 承上題，根據表 2-3 將你在第一題的答案轉換為更精簡的十六進制表示。
例如：將 A 的 100 0001 先補 0 成為 0100 0001，再將 0100 0001 分別轉換為 4 1。

表 2-3：二進制與十六進制轉換表

二進制	十六進制	二進制	十六進制	二進制	十六進制	二進制	十六進制
0000	0	0100	4	1000	8	1100	C
0001	1	0101	5	1001	9	1101	D
0010	2	0110	6	1010	A	1110	E
0011	3	0111	7	1011	B	1111	F

3. 將 ASCII 碼「1011001 1101111 1110101 0100000 1100001 1110010 1100101 0100000 1110100 1101000 1100101 0100000 1100010 1100101 1110011 1110100 0100001」解譯為英文句子：

4. 試以「ASCII 碼共使用七位元編碼」來說明為什麼 ASCII 碼不足以表示中文、日文、阿拉伯文等文字符號。（註：常用中文字約有五千個。）

想一想

文字編碼中的 ASCII 碼與萬國碼分別在統一哪些文字符號？為什麼要將文字編碼加以統一？

延伸學習

- 完整的 **ASCII** 編碼表

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

圖片來源：ASCII Table - Wikipedia

http://commons.wikimedia.org/wiki/File:ASCII-Table-wide.svg

- 以電腦科學家貝莫為首制定 **ASCII** 碼的故事

貝莫（Bob Bemer, 1920-2004）在 1956 年到 1962 年間為 IBM 公司服務時，親眼目睹了該公司九種不同的字元集所帶來的諸多麻煩，「他們都在談論電腦應該相互溝通，但都不肯放棄自己的標準。」貝莫回憶道：「於是我對他們講，你們不要互相責備而不顧別人。要知道，當時電腦界正在使用的符號系統絕非九種，用來表示字元的方式竟有六十種之多，這是真正的『巴別塔[註1]』！」

貝莫認為電腦是人造的，卻仍陷入巴別塔陰影的籠罩；然而，正因為電腦是人造的，人一定可以幫助它走出巴別塔。1960 年貝莫做了一個初步的設計，並向美國國家標準協會（ANSI）提交了一份關於制定通用電腦代碼的建議，根據這個建議，ANSI 組織了一個委員會來研製統一的標準。由於各委員都爭著要把自己的專用字元作為標準，利益之爭整整進行了兩年。貝莫艱難地做著說服工作：「這項工作非常瑣碎，但最終我和委員會主席在會議

[1] 據《聖經 創世記》第 11 章記載，人類當時聯合起來興建一座希望能通往天堂的高塔。為了阻止人類的計劃，上帝讓人類說不同的語言，使人類相互之間不能溝通，計劃因此失敗，人類自此各散東西。

室外握著手說：就是它了。」貝莫欣慰地發現，這個最終的結果與他最初的想法極為相似。

由於貝莫向 ANSI 提出標準電腦編碼的計畫、負責整合與協商，並設計出以往電腦未曾想過、對於現今卻很重要的字元符號（例如：Esc、\、…等），讓他獲得了「ASCII 之父」的美稱。

故事來源：Bob Bemer－百度百科 http://baike.baidu.com/view/1633904.htm

- **擴展為八位元的 ASCII 碼**

原本使用七位元表示 128（$2^7 = 128$）種文字符號的 ASCII，只能顯示 26 個基本拉丁字母、阿拉伯數目字和英式標點符號，因此在處理英語當中外來詞如 naïve、café、élite 等，所有重音符號都不得不去掉（這樣便違反拼寫規則），所以後來擴展為八位元的 EASCII（Extended ASCII），將各種重音符號、錢幣符號、KK 音標等字符納入，共可表示 256（$2^8 = 256$）種文字符號。

- **內碼與外碼**

每個中文字在電腦中儲存的二進制編碼就叫做內碼。由於中文字數龐大（鍵盤在製作時沒有辦法把所有中文字都列出來），且每個字之間不像英文字母具有明確的順序關係（很難從一個中文字的編碼去推估下個編碼是什麼字），要記住每個字的內碼並輸入實屬一件既耗時又非常不便的事，所以有了像是注音輸入法、倉頡輸入法、嘸蝦米輸入法等各種輸入法的出現，幫助我們用好記的方式輸入各個中文字。

以輸入「陳」這個文中字為例，我們在注音輸入法會打「ㄔㄣˊ」、在倉頡輸入法打「弓中木田」、在嘸蝦米輸入法打「B J Q S」，這些在輸入法裡面使用的編碼便是外碼。

教學補充資源

- **UTF-8、UTF-16 與 UTF-32**

萬國碼只是一個編碼方案而不是具體的儲存格式，現實中的 Unicode 會以各種 UTF（Unicode Transformation Format）形式存在，例如在網頁編碼中看到的 UTF-8，在資料庫編碼中看到的 UTF-16、UTF-32 等。其中 UTF-8 為了在網路上快速傳輸的需求，採用可節省空間的非固定長度進行編碼；UTF-16 與 UTF-32 則為了在資料庫中快速進行隨機存取的需求，採用固定長度編碼。

萬國碼的分類	使用位元數	備註
UTF-8 （不定長度）	8～32 位元 = 1～4 位元組	◆ 編碼方式： 1 byte 字元：與 ASCII 相同 2 bytes 字元：10 110 … 3 bytes 字元：10 1110 … 4 bytes 字元：10 11110 … ◆ 中文字大多是 3 bytes ◆ 網頁用萬國碼
UTF-16 / UTF-32 （固定長度）	16 位元 / 32 位元 = 2 位元組 / 4 位元組	◆ 資料庫用萬國碼

- **文字編碼的隨機存取**

隨機存取又稱為直接存取，相對於循序存取須順序尋找所需資訊，隨機存取可隨時跳至任何一處，存取需要的資訊。例如音樂 CD 便是隨機存取的實例，可以直接指定第 15 條歌播放，而不需要先快轉過前面 14 條歌。

在文字編碼中，根據字母使用頻率所設計的編碼表，每個字母與其他字母的編碼之間沒有關連，每次要做字母與編碼的對應，都必須從編碼表中一一找尋，屬於循序存取。而 ASCII 碼根據字母順序所設計的編碼表，每個字母與其他字母編碼具有可往前或往後推估的關連，例如 A 的編碼「100 0001」，下個字母 B 的編碼為 A 的編碼加 1 的結果「100 0010」，或計算「100 0011」為 A 的下 2 個字母 C，可以利用簡單的加減法快速得到該編碼所表示的文字符號，屬於隨機存取。

- 參考解答：換你試試看

		J	u	s	t	（空白）
1.	二進制	100 1010	111 0101	111 0011	111 0100	010 0000
2.	十六進制	4 A	7 5	7 3	7 4	2 0

	d	o	（空白）
二進制	110 0100	110 1111	010 0000
十六進制	6 4	6 F	2 0

	i	t	!
二進制	110 1001	111 0100	010 0001
十六進制	6 9	7 4	2 1

3. You are the best!
4. 常用中文字約有五千多個，以每個字需要一個編碼來說，7 位元的 ASCII 碼能表示 128 種文字符號，就算是擴充為 8 位元，最多也只能表示 256 種，所以要表示中文的五千多個常用字，至少需要使用 13 位元編碼，如果要表示更多中文字，以及更多國語言的文字符號，就需要使用更多位元來編碼（所以 BIG5 使用 16 位元編碼一萬三千多個中文字，萬國的 UTF-32 使用 32 位元編碼十萬多個各國文字）。

- 參考解答：想一想

ASCII 碼在統一英文大小寫字母以及常用數字符號的編碼，萬國碼則在統一全世界各國語言的文字符號。

如果文字編碼沒有統一，在傳遞訊息的時候，還需要多一道交換編碼的手續，而且還要交換成正確的編碼，才有辦法讀取文字訊息的內容，否則會呈現亂碼；將各種文字統一編碼標準，則可以免除交換編碼的手續，直接就可以順利讀取，讓文字訊息在交換或傳遞時更方便快速。

圖片來源

- 圖 2-1：研究者自行繪製。
- 圖 2-2：Morse code - Wikipedia

  http://en.wikipedia.org/wiki/Morse_code
- 圖 2-3：研究者自行繪製。使用電報的照片、人物圖示等素材來源：

  Patrick J. Kiger (2012). Life on Board: Greetings From the Titanic. National Geographic Channel.

  http://channel.nationalgeographic.com/channel/titanic-100-years/articles/life-on-board-greetin

gs-from-the-titanic/

Michelle L. Styer(1999). Women and Railroads During World War II. Milepost, the Journal of the Friends of the Museum.

http://www.rrmuseumpa.org/about/rrpeopleandsociety/women2.shtml

Icon search engine and market place - Iconfinder

https://www.iconfinder.com/

- 圖 2-4：Alan G Hobbs(1999). Five-unit codes. British Amateur Radio Teledata Group.

  http://www.nadcomm.com/fiveunit/fiveunits.htm

- 圖 2-5：研究者自行繪製。按鍵、手指與筆記型電腦之素材來源：

  Icon search engine and market place - Iconfinder

  https://www.iconfinder.com/

- 圖 2-6：研究者擷取修改自

  台灣數學博物館－國立臺灣師範大學數學系。

  http://www.math.ntnu.edu.tw/museum/

第三節、影像畫面如何用 0 與 1 表示？

利用二進制表示影像畫面的「影像編碼」，從現今的觀點又可分為點陣影像與向量影像（如圖 3-1）：前者是以不同顏色的小點來組合成一張圖片，後者則是利用點、線與各種幾何圖形來組合圖片。

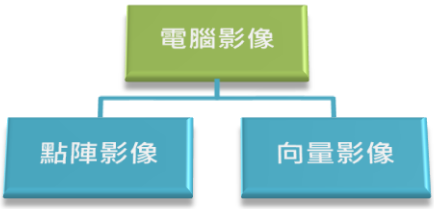


圖 3-1：影像編碼的分支圖

影像編碼的發展初期，點陣與向量兩者在十分接近的時間點，為了不同的需求與目標分頭並進。點陣影像方面，1957 年美國國家標準局定義出方形的像素點，成為現代數位照片、數位繪畫等點陣影像的編碼標準；向量影像方面，1963 年麻省理工學院的博士生薩森藍（Ivan Sutherland, 1938-）定義出點、線段、多邊形等向量元素，成為現代機械製圖、工程藍圖等向量影像的編碼標準。

究竟，要怎麼利用 0 與 1 來表示絢麗多彩的照片，或是精密複雜的機械設計圖呢？本節要帶您學習影像在電腦中的表示方法，並回答這個問題。首先我們先從源自於照片的點陣影像開始介紹。

點陣影像

在有像素點之前的影像編碼，1920 年代，傳輸文字訊息的電報曾利用如「＜」、「＞」、「＋」、「×」…等各種符號組合拼湊來表示一張照片（如圖 3-2）；接下來還有照片電報（也就是傳真機的前身）將黑白照片切割為許多小點，每個小點各自填滿黑色、灰色與白色，讓整齊排列的小點組合成一張照片（如圖 3-3）。

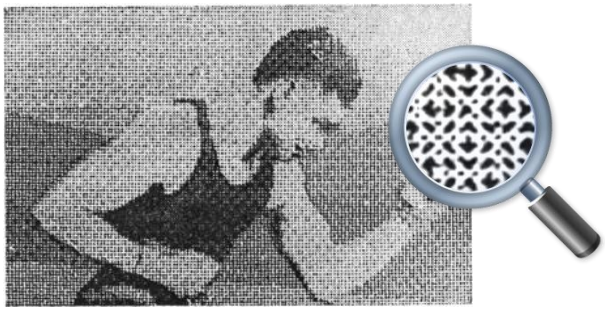


圖 3-2：利用特殊符號字樣構成一張照片

圖 3-3：利用深淺小點組合成一張照片

爾後到了 1957 年，美國國家標準局的研究團隊在替電腦設計影像編碼，讓照片可以用二進制表示時，延續照片電報將一張照片切割為許多小點的做法，定義數位照片的最小單位為正方形的像素點，每個像素點各自填滿深淺顏色來組合表示一張照片。那麼要如何把每個像素點填入不同顏色呢？該研究團隊的主持人柯爾基（Russell Kirsch, 1929-）想到：當一個像素點用二進制的一個位元來表示時，可以用 0 來表示黑、1 來表示白，如圖 3-5 的第一橫排儲存 1、1、1、1、1、1、1、1、1、1（共十個 1），第二橫排儲存 1、1、1、1、1、1、0、1、1、1（中間有一個 0），以此類推，便能組成一張有黑白兩色的照片；

圖 3-4：羅素．柯爾基

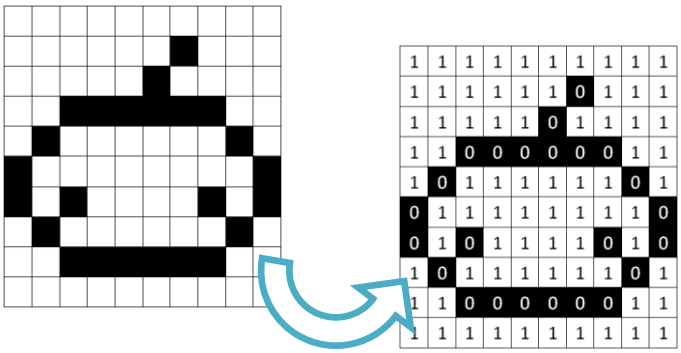


圖 3-5：以 0 與 1 表示黑白兩色的圖片

如果想要有更多的顏色，把一個像素點用兩個位元來表示，有 00 黑、01 深灰、10 淺灰及 11 白四種情況，如圖 3-6，第一橫排儲存 11、11、11、11、11、11、11、11、11、11（共十個 11），第二橫排儲存 11、11、11、11、11、11、01、11、11、11（中間有一個 01），以此類推到第十橫排，便組成一張由黑到白共有四種顏色的照片。

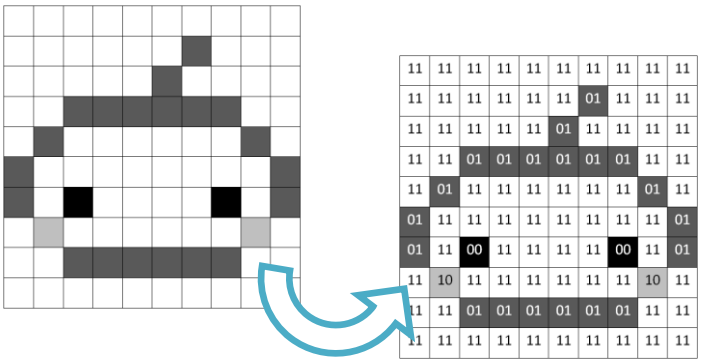


圖 3-6：以 00、01、10 與 11 表示由黑到白四色的圖片

該研究團隊第一張經電腦編碼儲存的照片(如圖 3-7)，正是將每個像素點用黑、深灰、淺灰與白這四個色階表示，並加以組合而成。在當時這張長度與寬度各只有 176 像素的數位照片，與現在動輒千萬畫素的數位照片相距甚遠，卻是電腦能儲存照片、處理照片的重要開端，影響了後來彩色數位影像的出現（使用更多位元來表示一個像素點，詳見延伸學習中「彩色的像素點」)，以及各種影像處理技術的發展（例如：把照片調亮、合成等）。

圖 3-7：以四個色階像素編碼的嬰兒照片

向量影像

除了利用點來組成一張圖，影像編碼還有沒有其他的可能性呢？在與像素點的出現十分接近的時間點，還有許多科學家試圖從雷達探測圖、工程與機械設計圖等角度思考其他影像編碼的方法。

1958 年美國空軍為了精確而有效率地顯示敵軍位置並預測其飛行路徑，參考了數學中使用方程式表示圖形－不會因放大或縮小而有任何偏差－的方式，使用線段做為影像編碼的最小單位，每條線段各有其起點與終點，拼接起來成為地圖的邊緣或是敵機的飛行路徑（如圖 3-8）。

歷史影片

SAGE 使用向量來表示地圖與路徑等電腦圖形（0:36）

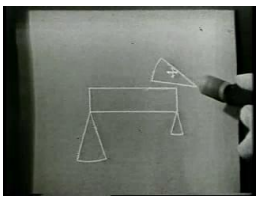
https://www.youtube.com/watch?v=LTbLkrvSM3c

圖 3-8：利用線段拼接雷達探測圖

而後在 1963 年，開發出史上第一台繪圖板 Sketchpad 的薩森藍(如圖 3-9)，由於從小看著父親繪製土木工程圖，意識到電腦圖形若只有線段還遠遠不夠，還需要圓形、三角形、矩形、多邊形，甚至將平面圖形加上厚度成為立體圖形等，更加具體而完整地設計出各種圖形的基本元素與屬性，能以精確作為基礎，組合出更為複雜而實用的各種圖形，開創了電腦輔助設計（Computer-Aided Design, CAD）以及使用者圖形介面（Graphical User Interface, GUI）的新領域。

歷史影片

Sketchpad 繪圖板使用向量來表示電腦圖形（1:23）



https://www.youtube.com/watch?v=dpKu1BJ5lRQ

圖 3-9：利用幾何圖形元素組成圖形

向量影像的編碼方式，是紀錄圖型的類型及其屬性等資訊。以向量圓形圖（如圖 3-10）與向量矩形圖的（如圖 3-11）為例，其中圖形的類別，圓形、矩形…等幾何圖形分別以 0000、0001…等編碼儲存，接下來屬性的部分，圓形有圓心坐標及半徑，以圓心坐標(2, 1)、半徑 4 來說，編碼結果如表 3-1，利用 0000 0010 0001 0100（共 16 位元）即可表示一個圓，而不必把整個圓的所有點都儲存下來；

表 3-1：向量圓形圖的編碼方式

圖形類別		屬性		
意義	圓形	圓心 X 坐標	圓心 Y 坐標	半徑長度
編碼	0000	0010	0001	0100

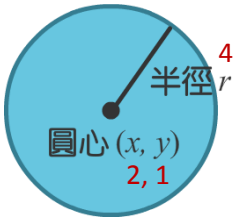


圖 3-10：向量圓形圖

矩形則有端點坐標、寬度及長度，以端點坐標(0, 0)、寬度 10、高度 6 來說，編碼結果如表 3-2，同樣利用 0001 0000 0000 1010 0110（共 20 位元）即可表示一個矩形，比起把 10×6＝60 個像素的資料全部儲存下來要精簡多了。

表 3-2：向量矩形圖的編碼方式

圖形類別		屬性			
意義	矩形	端點 X 坐標	端點 Y 坐標	寬度	長度
編碼	0001	0000	0000	1010	0110

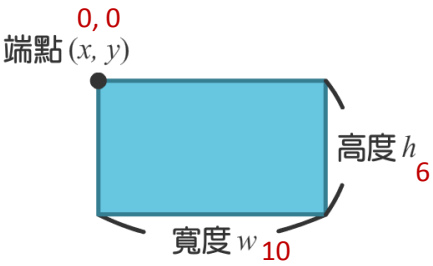


圖 3-11：向量矩形圖

如果要更進一步，讓向量圓形圖有輪廓或內部的色彩，只要增加更多的屬性來儲存就可以了。

小結

在影像編碼中，數位影像以像素點－或現今簡稱的像素（pixel）整齊排列成陣列的形式，便是「點陣影像」（如圖 3-12）；而以點、線段或多邊形組合而成的圖形，則屬於「向量影像」（如圖 3-13）。

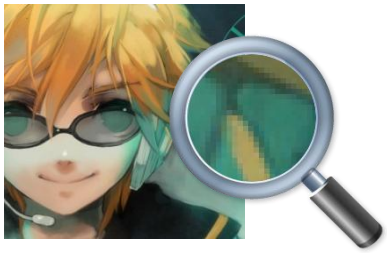


圖 3-12：點陣影像由像素組成

圖 3-13：向量影像由幾何元素組成

點陣與向量影像源起於不同的需求而有著各自的特性，對於同一圖形的點陣與向量影像（如圖 3-14），點陣影像能呈現如水彩的渲染效果，向量影像則能呈現工整的線條及漸層。

圖 3-14：同一圖形的點陣與向量影像

兩者最顯著的差異在於當點陣影像放大時，會出現鋸齒狀的失真情況，而向量影像則會隨著放大比率在螢幕上重新計算並繪製，保持原有的精準度不會失真（如圖 3-15）；所以當我們有不同需求，就需要選擇適合的編碼方式來儲存與處理各種影像。

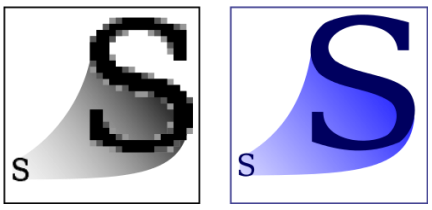
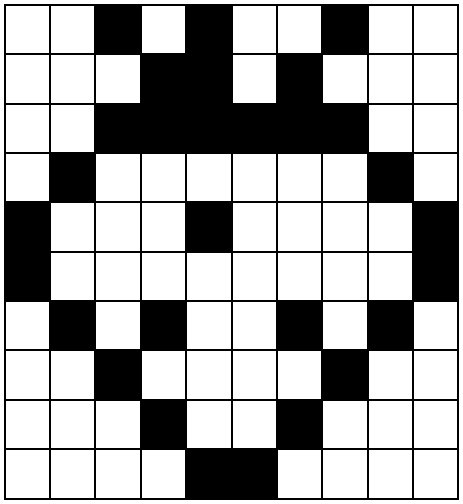


圖 3-15：放大點陣（左）與向量（右）影像之比較

換你試試看

1. 將下圖使用二進制（0 為黑、1 為白）來編碼表示：



2. 承上題，將該圖儲存進電腦中總要需要使用多少個位元？

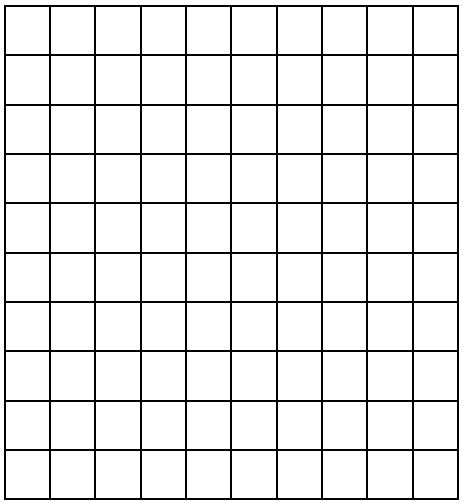
_____

3. 若要將圖 3-4（以兩位元表示一個像素）之圖形儲存進電腦中，又需要多少位元？

_____

4. 請畫出下列二進制影像編碼（0 為黑、1 為白）所表示之圖形：

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

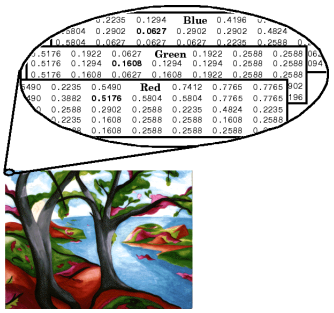


想一想

將像素定義為正方形的小格子，讓點陣影像在放大的時候會有不平滑的問題，有什麼方法可以改善或解決？

延伸學習

- **彩色的像素點**



彩色影像由紅（Red, R）、綠（Green, G）、藍（Blue, B）三色光以不同比例組合而成，在實際運用上將紅、綠、藍分別以 8 位元共同表示一個位元（即一個像素利用 24 位元儲存），即可達到全彩影像的效果。

以一個黃色的像素點來說明，黃色可以經由紅色加上綠色來組成，黃色的像素點在二進制的色碼即為紅色部分 1111 1111、綠色部分 1111 1111、藍色部分 0000 0000（黃色不需要由藍色組成），如下表所示；另外十六進制色碼則在簡化 0 與 1 的表示，將原本的 11111111 11111111 00000000 簡化為 FFFF00，這也是平常在網頁上看到如#FFFF00 色碼的由來。

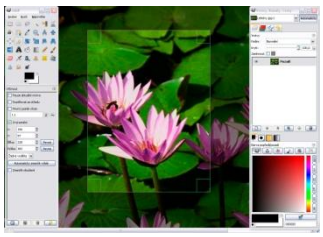
	紅 R	綠 G	藍 B
二進制色碼	1111 1111	1111 1111	0000 0000
十六進制色碼	F F	F F	0 0

註：電腦內仍是儲存二進制編碼，十六進制是為了提供設計師使用上比較便利。

- **向量影像名稱的由來**

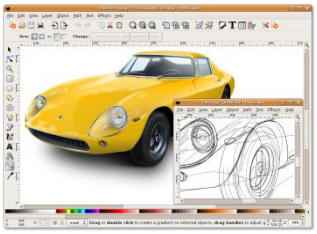
向量圖形（Vector Graphic）在當時是利用無線電取得戰機位置等資訊所繪製的圖形，其 vector 源自於動詞意義「以無線電導引」。

- **編輯點陣影像的自由軟體－GIMP**



點陣影像處理軟體 GIMP 可以做到包括照片潤飾、影像合成和建立影像等功能。軟體下載與 操作方式詳見：http://myshare.url.com.tw/note/325116

- **編輯向量影像的自由軟體－Inkscape**

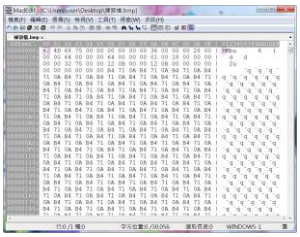


向量影像處理軟體 Inkscape 可以繪製符合 W3C 標準的 SVG 向量圖形。軟體下載與操作方式詳見：http://www.azofreeware.com/2011/01/inkscape-048.html

教學補充資源

- **可以將點陣影像開啟為十六進制編碼檢視的自由軟體－MadEdit**

  開啟 MadEdit 後開啟圖片檔，點選【檢視/十六進位模式】即可以十六進制檢視與編輯圖片檔。軟體下載位置：http://sourceforge.net/projects/madedit/



- **點陣與向量影像的更多比較**

  點陣與向量影像除了在放大時有著顯著差異，還有使用軟體、輸入電腦方法、應用範圍等其他不同之處，如下表之整理與說明：

影像類型	組成單位	適用之編輯軟體	輸入電腦的方法	應用範圍
點陣影像	像素	小畫家、Photoshop、PhotoImpact…等	掃描器、數位相機、數位錄影機、利用軟體繪製	照片、插畫、海報、網頁用影像等
向量影像	幾何元素	Illustrator、CorelDraw、AutoCAD…等	撰寫程式碼、利用軟體繪製	機械工程設計圖、插畫、海報、Flash 動畫等

- **現今常見的各種影像格式**

  數位影像除了根據點陣與向量分為兩類之外，還能根據不同用途而有許多格式類型，以下將常見的圖檔格式以表格說明：

格式	經過壓縮	色彩	支援透明效果	支援動畫效果	點陣影像	向量影像	用途
BMP	無	全彩			V		Windows 上的未壓縮圖檔
JPG	破壞式	全彩			V		網頁用照片或插畫
GIF	非破壞式	256 色	V	V	V		網頁用動畫
PNG	非破壞式	全彩	V		V		網頁用圖示或插圖
TIF	非破壞式	全彩	V		V		印刷用照片或插圖
SVG	無	全彩	V	V		V	網頁用插圖或動畫
PSD	Adobe Photoshop 軟體的原始檔格式。				V		原始檔
UFO	Corel PhotoImpact 軟體的原始檔格式。				V		原始檔

AI	Adobe Illustrator 軟體的原始檔格式。		V	原始檔
EPS	兼容點陣與向量圖形的原始檔格式，可用 Photoshop、Illustrator 或 CorelDraw 等軟體開啟編修。	V	V	原始檔

- **解析度面面觀**

  解析度是用來衡量數位影像清晰與否的指標。在拍攝或沖洗數位照片、購買螢幕常會遇到解析度的選擇與設定等問題，以下將常見的三種解析度以表格說明：

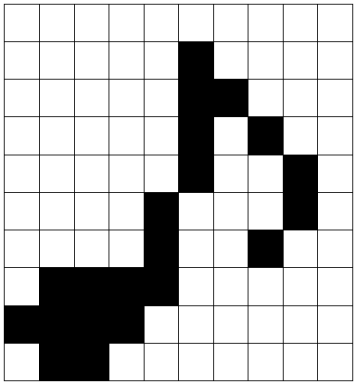
解析度種類	說明	計算單位	實例
像素解析度	數位影像中的像素數目。主要應用在網頁用影像或數位相機的畫素。	像素，通常以長×寬（像素）表示。	數位照片之尺寸 2592×1944，其畫素約為 500 萬畫素（2592 × 1944 ＝ 5,038,848）。
空間解析度	真實世界的長度單位與數位單位的轉換。主要應用在印表機、掃瞄機或印刷用的影像。	一英吋內的點數或像素數，有 dpi (dot per inch)與 ppi (pixel per inch)兩種單位。	要印刷的海報解析度為 300dpi。
螢幕解析度	螢幕畫面的顯示像素數。主要應用在表示螢幕顯示的大小。	像素，通常以長×寬（像素）表示。	螢幕設定解析度為 1280×960。

- **參考解答：換你試試看**

1.

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

2. 100（10 × 10 × 1＝100）
3. 200（10 × 10 × 2＝200）
4.



- 參考解答：想一想
  - 電腦螢幕的做法：把像素的尺寸縮小，小到肉眼幾乎分辨不出每個像素（對比過去嬰兒照片使用 5cm×5cm 很巨大的像素）。
  - 攝影的做法：先製造出更大尺寸的影像（設定較高的畫素），需要的時候再縮小。
  - 繪圖的做法：創建圖檔時，在同樣的尺寸內塞入更多的像素（設定較高的解析度）。
  - 作業系統的做法：在解碼像素的內容時，利用較大的圓形範圍互相交疊覆蓋，顯示平滑影像。

圖片來源

- 圖 3-1：研究者自行繪製。
- 圖 3-2、圖 3-3：研究者修改自
  John McVey (2005). telegraphic codes and message practice: elementary signs. Montserrat College of Art. http://www.jmcvey.net/cable/elements/pictures2.htm
- 圖 3-4：Steve Woodward (2007). Russell Kirsch: The man who taught computers to see. The Oregonian.
  http://www.oregonlive.com/living/index.ssf/2007/05/russell_kirsch_the_man_who_tau.html
- 圖 3-5、圖 3-6：研究者自行繪製。
- 圖 3-7：Russell A. Kirsch - Wikipedia. http://en.wikipedia.org/wiki/Russell_A._Kirsch
- 圖 3-8：Benj Edwards (2013). The Never-Before-Told Story of the World's First Computer Art (It's a Sexy Dame). The Atlantic.
  http://www.theatlantic.com/technology/archive/2013/01/the-never-before-told-story-of-the-worlds-first-computer-art-its-a-sexy-dame/267439/
- 圖 3-9：米粒（2013）。50 年前 Ivan Sutherland 說「要有圖」，Sketchpad 誕生了。優仕網共產檔－數位科技圖文。http://share.youthwant.com.tw/DF2004330.html
- 圖 3-10、圖 3-11：研究者自行繪製。
- 圖 3-12：研究者修改自
  じーた（2008）。【歌ってみた】メランコリア＠じーた。ニコニコ動画。
  http://www.nicovideo.jp/watch/sm3091922
- 圖 3-13：研究者修改自
  Flame Jony (2011). Unit 23 - computer Graphics: Task 2 - Types of Graphics. Blogger.
  http://unit23-jony.blogspot.tw/2011/10/task-2-types-of-graphics.html

- 圖 3-14：研究者修改自

  SANRIO(2009). ハローキティといっしょ！キャラクターーマール・キラザキ。SANRIO.

  http://www.with-kitty.com/kittyler05/

- 圖 3-15：可縮放向量圖形 (SVG)－維基百科。

  http://zh.wikipedia.org/wiki/%E5%8F%AF%E7%B8%AE%E6%94%BE%E5%90%91%E9%87%8F%E5%9C%96%E5%BD%A2

第四節、電腦裡的聲音長什麼樣子？

利用二進制表示音樂聲音的「音訊編碼」，從現今的觀點又可分為電子合成音與數位化音訊（如圖 4-1）：前者是以電腦模擬樂器的聲音，再根據樂譜演奏旋律；後者是將聲波轉換為數值的形式，再利用許許多多的數值來重現人聲或蟲鳴鳥叫等各種聲音。

圖 4-1：音訊編碼的分支圖

電子合成音由於編碼與解碼的手續較單純，曾是 1950-1980 年代音訊編碼的主流；數位化音訊則要到 1987 年，音效卡的出現結合了理論與實際，讓語音資料能有效率地以二進制來編碼表示，才逆轉電子合成音的地位而成為現今數位音訊的主流，並影響了後來音訊處理、語音辨識技術的發展，以及各種數位音訊格式的出現。

不過，利用樂器演奏的樂曲，或是人們講話的聲音，到底是怎麼利用 0 與 1 來表示的呢？本節便是要帶您學習聲音在電腦中的表示方法，並回答這個問題。首先我們從先出現、電腦比較容易模擬的電子合成音開始介紹。

電子合成音

音訊編碼中的電子合成音最早可以追溯到 1947 年，ENIAC 電腦為了除錯的需要，利用每隔一段時間發出一個隨機或持續不變的音階，來表示電腦正常運作或已經當機（當機時就無法再隨機取出一個音階，只能一直播出同樣的音階），這也是現在電腦開機時會發出短暫或持續的嗶聲來表示電腦是否正常運作的由來。三年後英國曼徹斯特大學建造的 Ferranti Mark 1 電腦，將樂譜中 Do、Re、Mi…等音階加以編碼（如圖 4-2），把原本 ENIAC 電腦無法操控的隨機音階改為能夠人為調整的特定音階，讓簡單旋律的兒歌、國歌等歌曲可以被電腦儲存與播放。

歷史影片

Ferranti Mark 1 利用音階與節拍表示電子合成樂（0:29）

https://www.youtube.com/watch?v=nx1gxl7duTo

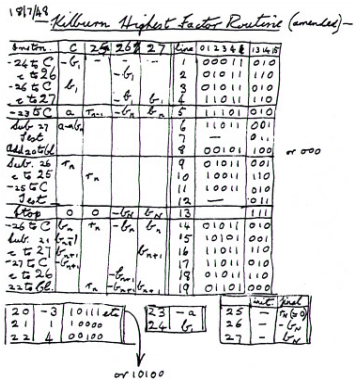


圖 4-2：Ferranti Mark 1 電腦各音階的二進制編碼

後來美國音響工程師史密斯（Dave Smith, 1947-）在 1981 年向音頻工程協會提出了 **MIDI**（Musical Instrument Digital Interface，樂器數位介面）的概念，將鋼琴、吉他、小喇叭、豎笛…等樂器的音色也加以編碼，開創了流行電音的新領域。

歷史影片

增加了樂器音色的 MIDI（0:55）

https://www.youtube.com/watch?v=R8yA1eQ3Y_U

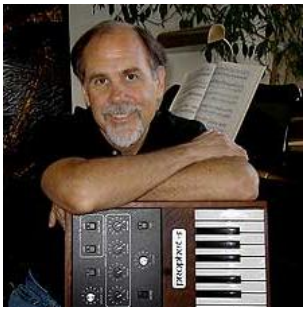
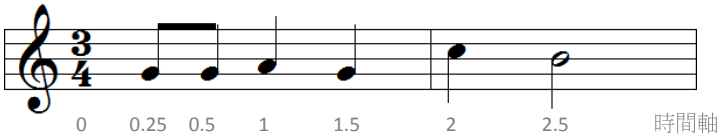


圖 4-3：戴夫．史密斯

MIDI 將樂譜加以編碼的方法，是紀錄時間軸、音階、節拍與音色等資訊。以下列《生日快樂歌》樂譜的前兩個小節為例：

演出樂器：鋼琴



時間從第 0 秒演奏到第 3 秒，分別記錄時間軸上第 0.25 秒、0.5 秒、…、2.5 秒的音階與節拍，音階從 Do、Re、…、Si 到高音 Do 共八種至少需要 4 個位元，節拍從全音符、二分音符到八分音符使用 4 個位元來表示，最後音色的部分從樂器編碼表查表得到鋼琴的編碼 0001，如表 4-1，由樂器演奏樂譜的電子合成音便能透過 0 與 1 來編碼表示。

表 4-1：電子合成音的編碼方式

	時間	音符		音色
意義	0.25 秒	音階 So: 5	節拍：八分音符	鋼琴
編碼	00.01	0101	1000	0001
意義	0.5 秒	音階 So: 5	節拍：八分音符	鋼琴
編碼	00.10	0101	1000	0001
意義	1 秒	音階 La: 6	節拍：四分音符	鋼琴
編碼	01.00	0110	0100	0001
意義	1.5 秒	音階 So: 5	節拍：四分音符	鋼琴
編碼	01.10	0101	0100	0001
意義	2 秒	音階 Hi Do: 8	節拍：四分音符	鋼琴
編碼	10.00	1000	0100	0001

意義	2.5 秒	音階 Si: 7	節拍：二分音符	鋼琴
編碼	10.10	0111	0010	0001

如果想要進階一點，讓這首樂曲加上低音伴奏的效果，就在同一個時間點多儲存一組音符資訊（例如：有兩筆時間為 00.01 的編碼資料，一筆是儲存主旋律的音符音色，另一筆則是儲存伴奏的）；而當一首樂曲的音階或節拍出現更多種類，就需要使用更多的位元來儲存（例如：現在節拍使用 4 位元儲存，要表示 16 分音符就不夠用，需要再多一個位元才有辦法表示）。

數位化音訊

然而人聲、磁帶錄音等音訊，雖然對人類來說同樣都是聲音，但是由於每個人都有自己獨特的音色，更別說各種動物的叫聲，或風吹草動的聲音，無法每種都利用電子合成的方式編碼模擬，所以有一些科學家試圖從聲波的物理特性以及電話通訊的角度思考音訊編碼的其他可能性。

在物理上來說，所有的聲音都是以聲波的方式傳遞，所以要將聲波儲存進電腦，必須先將聲波轉換為一個個的數值，才有辦法以二進制的 0 與 1 表示。如圖 4-4，人耳聽到的聲波屬於連續不間斷的類比訊號，而電腦接受的 0 與 1 屬於非 0 即 1、不連續的數位訊號。

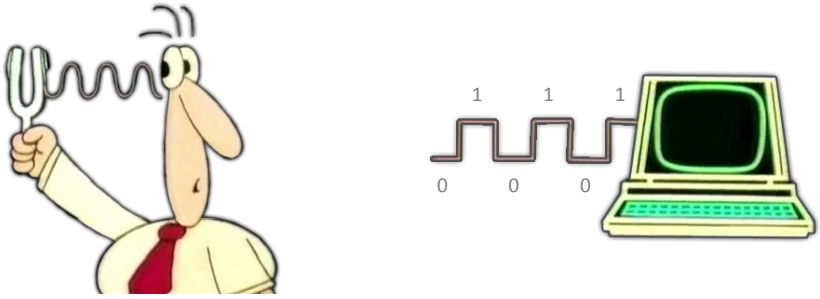


圖 4-4：類比訊號（左）與數位訊號（右）

至於把類比訊號轉換為數位訊號，也就是將聲波編碼的方法，早在 1937 年英國科學家李維（Alec Reeves, 1902-1971）為了減少長距離電話通信的雜訊，發現通訊科技中有一套 PCM（Pulse-Code Modulation，脈衝編碼調變）理論，可以將聲波擷取端點，再轉換為一個個數值，如此一來，原本抽象的聲音一旦轉換為數字，進一步要做編碼或傳遞等工作就顯得容易多了。

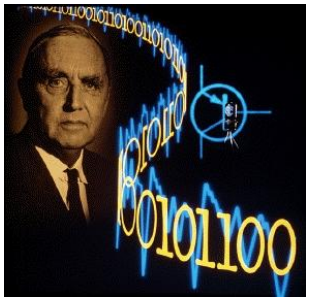


圖 4-5：亞歷克．李維

把 PCM 理論應用在聲音的步驟有三：取樣、量化及編碼。「取樣」指的是透過每隔一段時間對類比音訊進行測量（如圖 4-6），因為要根據聲波是隨著時間傳遞的特性，所以首先規劃出時間軸（橫軸），對相等時間間隔的聲波進行擷取；

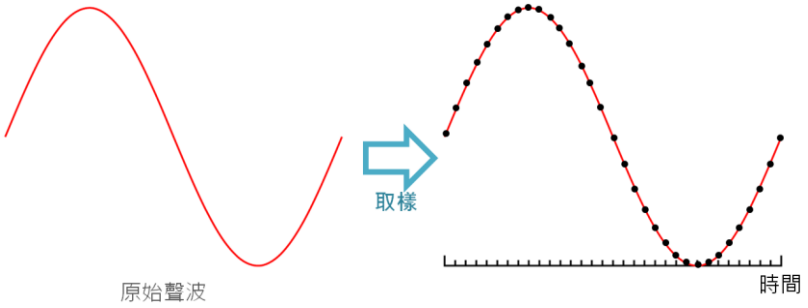


圖 4-6：應用 PCM 理論的步驟 1－取樣

接下來的「量化」（如圖 4-7），指的是把測量到的結果以數值表示，根據聲音有大聲、小聲也就是強弱的變化，規劃出音量軸（縱軸），對音量的程度進行紀錄；

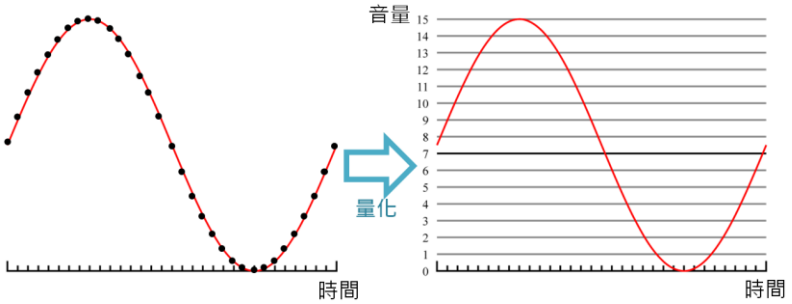
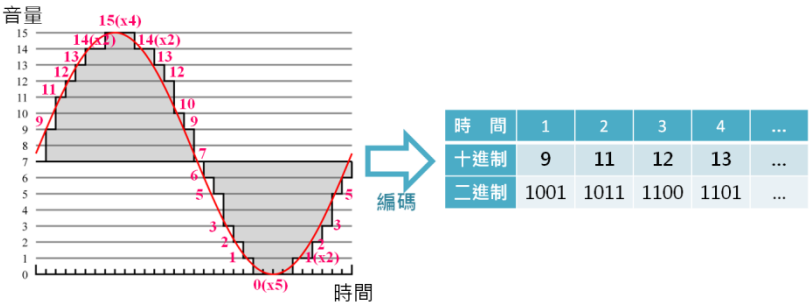


圖 4-7：應用 PCM 理論的步驟 2－量化

最後一步的「編碼」（如圖 4-8），為了因應電腦裡面採用 0 與 1 來儲存資料的特性，而將量化得到的數值轉換為二進制。



時　間	1	2	3	4	...
十進制	9	11	12	13	...
二進制	1001	1011	1100	1101	...

圖 4-8：應用 PCM 理論的步驟 3－編碼

而聲波從取樣、量化到編碼的整個過程，也就是讓聲波利用二進制的 0 與 1 來表示，便是對類比訊號進行「數位化」。

其中在取樣的步驟中，如果時間的間隔很大，也就是擷取的次數較少的時候，有些聲音的資料會被遺漏沒有被記錄到，導致取樣後的結果與原來的聲音有落差；反之，如果時間的間隔很小，即擷取的次數越多，便能越接近原來的聲音（如圖 4-9）。同理，在量化的步驟中，音量的種類只有劃分出 0～15 共 16 種，跟劃分出 0～255 共 256 種相比，後者能更真實的呈現原本的聲音。

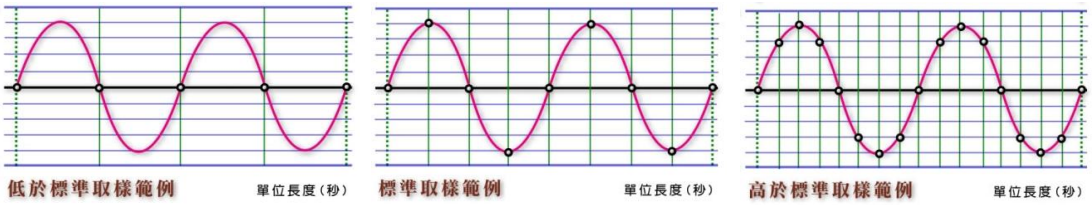


圖 4-9：取樣次數的多寡會影響儲存的音質是否接近原本的聲音

介紹到這邊，數位化音訊的編碼方式已經不是問題，但為什麼數位化音訊還遲遲不出現，讓電子合成音在音訊編碼中獨占鰲頭呢？這是因為數位化音訊不像電子合成音的音階、節拍、音色等編碼資料皆可各自對應到一個聲音特性，編碼與解碼的手續較單純，在硬體方面所需的門檻相對較低；數位化音訊在編碼的時候要將聲波準確而快速地進行取樣與量化，而在解碼的時候則要將編碼好的資料再轉換為不同的頻率與音色播放出來，需要電腦硬體的配合才有辦法實現。

到了 1987 年，能轉換類比與數位音訊的音效卡（如圖 4-10）終於問世，讓人聲、磁帶錄音等聲音資料能有效率地以二進制編碼表示，才逆轉電子合成音的地位而成為現今音訊編碼的主流，並影響了後來音訊處理（例如：把聲音快轉、調高音調）、語音辨識（例如：Siri 能瞭解使用者在說什麼）技術的發展，以及像是 MP3 等各種數位音訊格式的出現（詳見延伸學習中「現今的各種音訊格式」）。

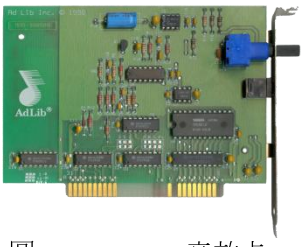


圖 4-10：AdLib 音效卡

小結

在音訊編碼中，源自利用樂器演奏音樂，仿照樂譜以音階、音色、節拍等資訊組成，是「電子合成音」；而源自人聲語音的需要，將聲波經過取樣、量化及編碼處理，則屬於「數位化音訊」。所以當我們有創作電子音樂，或是錄製講話、唱歌聲音的不同需要，就要選擇適合的編碼方式來儲存與處理各種音訊。

換你試試看

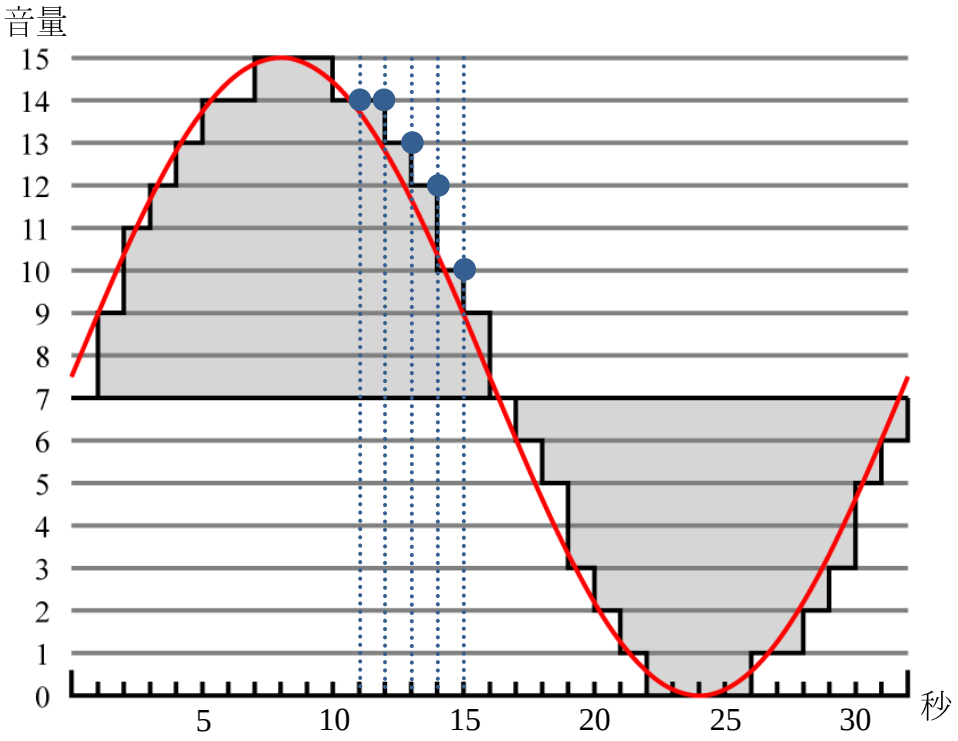
1. 將二進制音訊編碼「011 010 001 010 011 011 011」根據表 4-2 改以簡譜表示：

表 4-2：音階編碼表

音階	Do	Re	Mi	Fa	So	La	Si
編碼	001	010	011	100	101	110	111
簡譜	1	2	3	4	5	6	7

_____

2. 下圖是經過取樣處理的類比音訊，請將第 11 至第 15 秒轉換為量化數值：



	第 11 秒	第 12 秒	第 13 秒	第 14 秒	第 15 秒
量化結果					

3. 承上題，將量化結果編碼為二進制：

	第 11 秒	第 12 秒	第 13 秒	第 14 秒	第 15 秒
編碼結果					

4. 承上題，若要將編碼完成的第 11 至 15 秒音訊資料儲存進電腦中，共需要多少位元？

_____

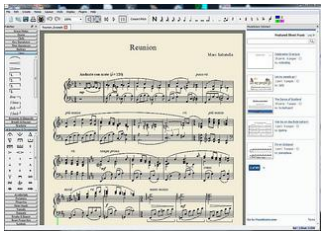
想一想

請利用影像與音訊編碼，設計一套影片的編碼方式。

提示：同時具有聲光效果的影片，主要結合了影像與音訊兩種編碼而構成。在影像部分，你會採用點陣或向量影像？在音訊部分，你會採用電子合成音或數位化音訊？

延伸學習

- **編輯電子合成音的自由軟體－MuseScore**



MIDI 創作軟體 MuseScore，可用滑鼠、鍵盤或 MIDI 鍵盤輸入音符，編輯完成的樂譜可以匯出成 mid、wav、flac、ogg 等音樂檔案格式。軟體下載與操作方式詳見：
http://blog.xuite.net/yh96301/blog/59069482

- **編輯數位化音訊的免費軟體－mp3DirectCut**

MP3 剪接與編修軟體 mp3DirectCut，可以擷取與結合 MP3 音訊，或調整音量大小做出淡入淡出的效果。軟體下載與操作方式詳見：
http://steachs.com/archives/1612

- **現今常見的各種音訊格式**

數位音訊除了根據電子合成音與數位化音訊分為兩類之外，還能根據不同用途而有許多格式類型，以下將常見的音訊格式以表格說明：

類型	格式	經過壓縮	用途
電子合成樂	MID	無	網頁、傳輸用的電子合成樂
數位化音訊	WAV	無	Windows 用的未壓縮音訊
	WMA	破壞式	網頁、傳輸用的數位化音訊
	CDA	無	CD 用的高品質音訊
	MP3	破壞式	網頁、傳輸用的高壓縮音訊
	OGG	非破壞式	網頁、傳輸用開放音訊格式
	FLAC	非破壞式	網頁、傳輸用的高品質音訊

教學補充資源

- **MIDI 檔案格式**

  現在的 General MIDI 規範，音色的編碼多達 128 種，包含打擊樂器、弦樂器、銅管樂器、民族樂器，以及一些特殊音效，另外還支援同時發出 24 個聲音來達到多聲部的效果。而為了避免無謂的空間浪費，音色的編碼可以音軌為單位統一記錄，時間軸也可利用一個四分音符所佔用的基本時間來統一計算。

  參考資源：

  - MIDI 文件格式分析－Midifan：
    http://www.midifan.com/modulearticle-detailview-901.htm
  - 標準 MIDI 檔案格式規格書－MIDI 產業協會：
    http://www.music.mcgill.ca/~ich/classes/mumt306/midiformat.pdf

- **取樣頻率與取樣量**

  為了要確保經過 PCM 處理的數位音訊不會與原始音訊相差太多，在取樣與量化分別有「取樣頻率」與「取樣量」可以作為音質好壞的評估標準。取樣頻率的單位為赫茲（Hz），取樣量的單位為位元（bit），一般音樂 CD 的取樣頻率為 44.1KHz，取樣量為 16 位元。取樣量又稱為音訊解析度或位元深度，當使用 4 位元的取樣量時，讓聲音大小共有 2^4=16 種變化情形；使用 8 位元的取樣量，則有 2^8=256 種變化情形可以儲存。

- **電子合成音與數位化音訊檔案大小的比較**

  兩種編碼方式同樣在儲存持續 1 秒的音階 Do，電子合成音僅需儲存時間的長度以及該音階的編碼，數位化音訊則是要把一秒鐘振動 523 次（即 523Hz）的聲波資料全部記錄下來，檔案大小相對比電子合成音大上許多。

- **音訊資料的壓縮**

  由於數位化音訊在記錄聲波資料時，為了要盡可能貼近原始的聲音，常會讓檔案大小非常的大，若要在網路上傳輸該檔案便會非常的慢。為了要解決這樣的問題，而有了「資料壓縮」研究領域的出現，例如將人耳聽不到的頻率捨棄不存，或是利用霍夫曼編碼等壓縮演算法，在維持合理的音訊品質下，大幅縮減數位化音訊的檔案大小。

- 參考解答：換你試試看
  1. 3 2 1 2 3 3 3
  2. 14 14 13 12 10
  3. 1110 1110 1101 1100 1010
  4. 20（每秒 4 位元，5 秒共使用 4×5＝20 位元）

- 參考解答：想一想

  以 Youtube 上看到的影片來說，影像部分是採用點陣影像，能表現真實的照片場景，聲音部分採用數位化音訊，能呈現細膩多變的人聲、風聲等各種聲音；以 Flash 動畫來說，為了要精細地呈現各種圖案，影像部分採用向量影像，而聲音同樣採用數位化音訊；電子合成音主要用在電玩的背景音樂或歌唱伴奏等音樂創作領域。

圖片來源

- 圖 4-1：研究者自行繪製。
- 圖 4-2：Jonathan Fildes (2008). 'Oldest' computer music unveiled. Science and technology reporter, BBC News. http://news.bbc.co.uk/2/hi/technology/7458479.stm
- 圖 4-3：George Petersen (2010). 2010 TECnology Hall of Fame - Audio Innovations that Changed the (Pro Audio) World. NAMM Foundation. http://tecfoundation.com/hof/10techofdetails.html
- 圖 4-4：Bits and Bytes TVO(2012). Episode 10: Computer Music. Youtube. https://www.youtube.com/watch?v=xF76MmVkjpA
- 圖 4-5：David Robertson(2010). The Curious Life and Complex Legacy of Alec Reeves. Cafe Scientifique. http://goforich.co.uk/cafescientifique/previous/previousreeves.htm
- 圖 4-6、圖 4-7、圖 4-8：研究者修改繪製自
  脈衝編碼調變－維基百科
  http://zh.wikipedia.org/wiki/%E8%84%88%E8%A1%9D%E7%B7%A8%E7%A2%BC%E8%AA%BF%E8%AE%8A
- 圖 4-9：賴建宇（2013）。認識數位音訊。Yamaha 教育訓練。
  http://www.midi.twmail.net/ltu01.htm
- 圖 4-10：AdLib－維基百科 http://zh.wikipedia.org/wiki/AdLib