

CS T_T O CANPLUGGED

Computer Science for Taiwan Teens

偷 插電的資訊科學

【教師手冊】



謝宗翔／教材編著

顏國雄／程式實作

中華民國軟體自由協會 出版

謹以此書獻給

我的家人

澤銘

TPET 教育撲浪客

以及

所有關心教育的你們

前 言

Computer Science for Taiwan Teens (簡稱 CS T_T)，旨在提供國中學生一套符合運算思維的資訊教材，透過「偷插電」的設計實現最小成本以及最適於不同軟硬體環境的教學活動。

不插電 (Unplugged) 教學，指的是不需要電腦，只要紙筆或簡單的教具即可進行的學習方式。

插電 (Plugged) 教學，指的是透過電腦的操作，學習資訊科學的知識，可能是應用軟體、可能是網路操作、也可能是用程式模擬知識的內容。

本教材中文名稱為「偷插電的資訊科學」，主要提供於國中任教的資訊老師，一套在一般教室或電腦教室都能實施的資訊教材，不受空間或外在環境的限制，老師可以根據自己的情況選擇上課的地點。(亦即可自由選擇插電或不插電的教學方式)

為了讓教師在實施教學時能有所依循，特編撰教師手冊。本冊的教學內容旨在提供國中七年級基本的資訊科學知識。另有國中八年級的教材，內容則是著重在運算思維的發展概念，將於後續日程提供教師參考。

此外我們同時提供延伸的教學內容，學生除了可以用紙筆推演教材所提供的概念，亦能應用這些概念在應用軟體的操作，或是與實際的生活經驗連結。

所有教材的補充說明與推廣活動、工作坊訊息，皆以官方網站與 Facebook 粉絲專頁為主，敬請大家瀏覽與愛護，並不吝給我們指教。官方網站的網址為：

<http://cstt.slat.org>

本書所有教材均以創用 CC 4.0 姓名標示 - 非商業性 - 相同方式分享釋出。授權相關資訊請參考以下網址：

https://creativecommons.org/licenses/by-nc-sa/4.0/deed.zh_TW

本教材內的教學活動的內容與發想，出自「不插電的資訊科學」一書 (Computer Science Unplugged, <http://csunplugged.org/>)。

謝宗翔

教學進度參考範例

【上學期】

週次	主題	內容	不插電	插電	學習表現
1	電腦教室規則說明	教室規則、google classroom 帳號登入並修改密碼	○		
2	二進位	翻牌程式說明、操作，回家作業(二與十進位的互換)	○	○	* 資 t-IV-1 能了解資訊系統的基本組成架構與運算原理。 * 資 t-IV-4 能應用運算思維解析問題。
3		進階常識，儲存單位，實例：手機容量需求的計算	○		* 資 p-IV-1 能選用適當的資訊科技組織思維，並進行有效的表達。
4		活動：傳遞祕密訊息，google 翻譯的說明與體驗	○	○	* 資 a-IV-4 能具備探索資訊科技之興趣。
5	用數色表示顏色	概念說明，手寫與程式操作	○	○	* 資 t-IV-1 能了解資訊系統的基本組成架構與運算原理。 * 資 t-IV-4 能應用運算思維解析問題。
6		給題目，截圖後透過 google 繪圖、硬碟繳交作業(練習)		○	
7	段考週				
8	用數色表示顏色	給題目，截圖後透過 google 繪圖、硬碟繳交作業(評分)，回家作業(紙本，彩色圖案)	○	○	* 資 c-IV-1 能熟悉資訊科技共創工具的使用方法。 * 資 c-IV-2 能選用適當的資訊科技與他人合作完成作品。 * 資 c-IV-3 能應用資訊科技與他人合作進行數位創作。
9	資訊理論	猜測的概念，猜數字遊戲，珠璣妙算規則說明	○		* 資 t-IV-4 能應用運算思維解析問題。 * 資 p-IV-1 能選用適當的資訊科技組織思維，並進行有效的表達。
10		珠璣妙算遊戲，遊戲策略說明		○	* 資 p-IV-2 能利用資訊科技與他人進行有效的互動。
11		決策樹說明與繪製	○		

【上學期】

週次	主題	內容	不插電	插電	學習表現
12		卡片翻轉魔術活動，概念說明	<input type="radio"/>	<input type="radio"/>	* 資 a-IV-1 能落實康健的數位使用習慣與態度。
13	錯誤的發現與修正	書碼與條碼概念，ISBN 檢查碼的運算(紙筆、小算盤輔助)，鍵盤數字區的說明與鍵位練習	<input type="radio"/>	<input type="radio"/>	* 資 a-IV-2 能了解資訊科技相關之法律、倫理及社會議題，以保護自己與尊重他人。 * 資 a-IV-3 能了解資訊科技相關行業之進路與生涯發展。 * 資 a-IV-4 能具備探索資訊科技之興趣。
14	段考週				
15		以 google 試算表的公式、函數功能作 ISBN-10 與 ISBN-13 檢查碼的產生(練習)		<input type="radio"/>	* 資 t-IV-3 能設計資訊作品以解決生活問題。 * 資 t-IV-4 能應用運算思維解析問題。 * 資 c-IV-1 能熟悉資訊科技共創工具的使用方法。
16	錯誤的發現與修正	以 google 試算表的公式、函數功能作 ISBN-10 與 ISBN-13 檢查碼的產生(作業)		<input type="radio"/>	* 資 c-IV-2 能選用適當的資訊科技與他人合作完成作品。 * 資 c-IV-3 能應用資訊科技與他人合作進行數位創作。
17		以 google 試算表的公式、函數功能作身份證字號產生器(練習)		<input type="radio"/>	
18		以 google 試算表的公式、函數功能作身份證字號產生器(評分)		<input type="radio"/>	
19	電腦科測驗	紙筆測驗	<input type="radio"/>		
20	電腦科測驗	Google 問卷(心得分享)		<input type="radio"/>	
21	段考週				

【下學期】

週次	主題	內容	不插電	插電	學習表現
1	搜索演算法	傳統海戰棋	○		* 資 t-IV-1 能了解資訊系統的基本組成架構與運算原理。 * 資 t-IV-4 能應用運算思維解析問題。 * 資 a-IV-4 能具備探索資訊科技之興趣。
2		程式海戰棋體驗、概念說明		○	
3		搜索演算法說明與講解		○	
4		測驗	○		
5	排序演算法	紙杯排序遊戲	○		* 資 t-IV-1 能了解資訊系統的基本組成架構與運算原理。 * 資 t-IV-4 能應用運算思維解析問題。
6		天平排序程式體驗、概念說明		○	
7	段考週				
8	排序演算法	排序演算法說明與講解	○		* 資 a-IV-4 能具備探索資訊科技之興趣。 * 資 p-IV-1 能選用適當的資訊科技組織思維，並進行有效的表達。
9		測驗	○		
10	文字壓縮	概念程式體驗，問題討論		○	* 資 t-IV-1 能了解資訊系統的基本組成架構與運算原理。 * 資 t-IV-4 能應用運算思維解析問題。 * 資 p-IV-1 能選用適當的資訊科技組織思維，並進行有效的表達。
11		再次操作概念程式，發下紙本作業	○	○	
12		測驗	○		
13	網路通訊協定	戶外活動體驗	○		

【下學期】

週次	主題	內容	不插電	插電	學習表現
14	段考週				
15	網路通訊協定	概念程式操作體驗、說明、討論		○	* 資 t-IV-2 能熟悉資訊系統之使用與簡易故障排除。 * 資 t-IV-4 能應用運算思維解析問題。 * 資 c-IV-2 能選用適當的資訊科技與他人合作完成作品。 * 資 p-IV-2 能利用資訊科技與他人進行有效的互動。
16		測驗	○		
17	文字加密	三種加密概念說明，紙筆操作	○		* 資 t-IV-3 能設計資訊作品以解決生活問題。 * 資 t-IV-4 能應用運算思維解析問題。 * 資 p-IV-1 能選用適當的資訊科技組織思維，並進行有效的表達。 * 資 a-IV-2 能了解資訊科技相關之法律、倫理及社會議題，以保護自己與尊重他人。 * 資 a-IV-4 能具備探索資訊科技之興趣。
18		加密程式操作，發回家作業(紙本)	○	○	
19		測驗			
20	期末測驗	紙筆測驗或心得分享	○		
21	段考週				

【課前準備】

1. 附錄的程式檔案只需要網路瀏覽器軟體，直接開啟資料夾中的 html 檔即可直接執行，手機、平板電腦與個人電腦皆可操作。
2. 若要將附錄的程式檔放置在網路上，讓同學可以直接透過點擊連結進行操作，你會需要一個可讀取網頁檔案的網路空間。(推薦安裝 AppServ 環境，或是請學校的資訊組幫忙規劃)
3. 部份章節需要進行應用軟體的操作，若不想另外安裝軟體，推薦使用 Google for Education，除了方便開設學生帳號外，還能設定公用網路硬碟，方便讓學生繳交已完成的作業檔案。
4. 若你決定提供學生每人一個專用帳號，建議於學期初安排一節課的時間，讓他們練習以個人帳號、密碼至少登入過一次。視你的學生之狀況，決定是否讓他們更改預設密碼（記得請他們把新密碼抄下來，並替他們檢查抄寫是否正確）。如果時間允許的話，甚至可以請學生登出帳號後，再重新登入一次，確定他們真的學會這個動作了，日後可以避免掉許多重設密碼的時間。
5. 每一個章節的活動結束後，老師可以自行設計測驗問題，此部份讓教師自由發揮創意，本教師手冊不另提供。
6. 本書所使用之程式與參考檔案，請參考軟體自由協會線上書城之相關資訊：
<https://sites.google.com/view/slat-bookstore/>

目 錄

前言	i
教學進度參考範例	iii
第一章 二進位的概念	1
第二章 用數字表示圖像	13
第三章 資訊理論	23
第四章 錯誤的發現與修正	29
第五章 搜尋演算法	43
第六章 排序演算法	55
第七章 文字壓縮	63
第八章 網路通訊	69
第九章 資料加密	85

第一章 二進位的概念

教學時數：3 節

事前準備：

1. 不插電：教師示範使用附錄 1-1 的大張卡片，學生練習則使用附錄 1-2 的小張卡片（一人一份）
2. 插電：使用附錄程式檔，開啟資料夾名稱為「numbers」的 `binary_numbers.html`

活動流程——十進位數字與二進位數字的轉換

1. 觀察：請學生按大小順序排列卡片，點數最大放在左手邊、點數最小的放在右手邊（插電時可省略這步驗）。接著觀察卡片上方點數的規律，詢問學生看出什麼了？（由最小的點數開始，每張牌的點數都是將上一張牌乘以二所得到的結果）
2. 湊點數：任意指定 1~31 之間的數字，請學生依照「蓋牌」、「打開」的方式，用這五張牌湊出符合的點數，可讓學生多練習幾次。例如指定 9 點，則結果為：



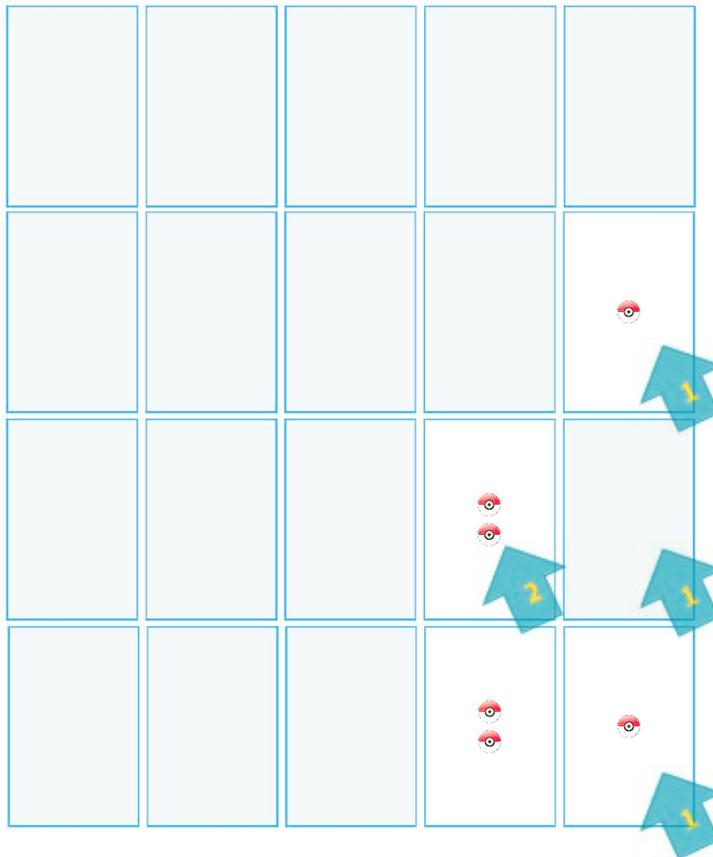
3. 告訴學生，每張牌有「蓋牌」、「打開」的兩種時候，為了方便說明，現在我們改用「0」取代蓋牌、用「1」取代打開的狀態。任意指定一種翻牌狀態，請學生試著由左至右使用 0 與 1 念一邊。例如下圖的狀態念為「10110」：

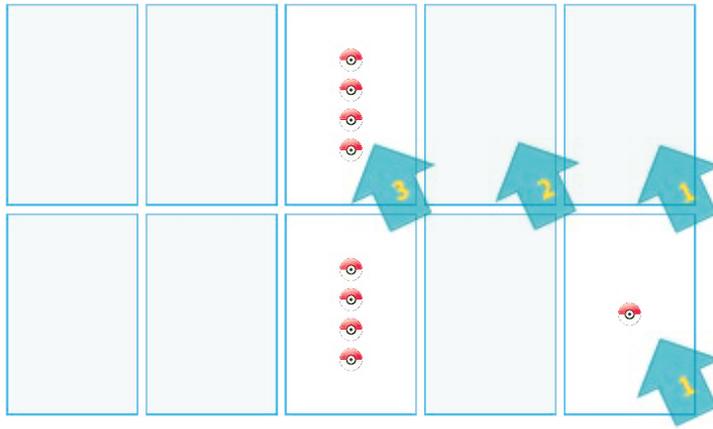


- 任意出題，指定 0 與 1 的狀態，請學生回答有幾個點數，此為二進位轉十進位的練習。
- 任意出題，請學生使用卡片湊出指定點數之後，寫下 0 與 1 的狀態，此為十進位轉二進位的練習。也可以指定幾組學生常見到的數字組合，如生日、電話 … 等，用來練習十進位與二進位的轉換。

活動流程二 — 數數兒，進位的概念

- 請學生回答，這五張牌所能表達的點數，最多與最少的時候各為幾個點？（最多 31 點、最少 0 點）
- 從小我們就會數數兒，接下來我們要練習一邊翻牌，一邊從 0 數到 31。
- 翻牌的規則是，一律要從最右邊依序開始操作「打開」或「蓋上」的動作，不可以跳著翻牌。接著由老師示範由 0 數到 5 的動作，箭頭上的數字代表翻面的順序，示範後請學生接著由 6 往下數，直到全部的卡片都打開為止：





延伸學習一

1. 想一下，過去我們在學十進位數字時，當我們把一個 0 放到一個數字右邊，就等於把那個數子乘上 10，例如：

$$9 \rightarrow 90$$

$$30 \rightarrow 300$$

2. 在二進位數字的世界中，當我們把 0 放到一個二進位數字的右邊會發生什麼事呢？學生已經學過二進位轉換的概念，所以可以知道？處應該代表數字 18，由此可看出點數增加為原來的 2 倍。

$$1001 \rightarrow 10010$$

(9)

(?)

3. 之前我們使用的每一張卡片都可以代表電腦裡面的一個「位元」（英文稱為 bit）。對於電腦來說，每一個數字、字母或是符號假設都給予一個十進位的編號，這個編號轉換成二進位後，如果用 5 張卡片表示，就說它們各佔了 5 個「位元」的空間。
4. 帶著學生認識鍵盤上所有的字母、數字、符號及功能鍵。接著請學生觀察鍵盤，數數看電腦總共有幾個字元（包含英文、數字及所有看得到的標點符號）要表示。算算看電腦總共需要幾個位元才能儲存所有的字元。
5. 現代的電腦都使用「美國資訊交換標準代碼」（American Standard Code for Information，又稱為 ASCII 碼）來表示每一個字元。但在一些不說英文的國家則必須使用更長的編碼才足以儲存會用到的所有字母。
6. 一個位元能表達出來的東西，不是 0 就是 1，只能儲存 2 種字元。若是將 8 個位元聚集在一起，就可以表達從 0 到 255 總共 256 種字元。通常我們把 8 個位元稱作一個「位元組」（byte）。

7. 電腦的速度取決於它一次可以處理幾個位元的訊息。舉例來說，一個 32 位元的電腦在一次操作中可以處理 32 個位元；但 16 位元的電腦則必須將 32 位元的數字分成兩個 16 位元，因此速度慢多了（但比較便宜）。

延伸學習二

1. 認識電腦的儲存單位：

- bit (位元)
- byte (位元組) = 8 bits
- KB (KiloByte) = 1024 bytes
- MB (MegaByte) = 1024 KB
- GB (GigaByte) = 1024 MB
- TB (TeraByte) = 1024 GB

2. 除了 bit 與 byte 之外，隨著電腦儲存資料的空間需求越來越大，也就跟著需要不同的儲存單位。就像是錢幣與紙鈔，不同的幣值適用的時機也不一樣。買一個價值 100 元的東西，除了直接給予一張 100 元的鈔票外，當然也能選擇給予 100 個一元，只是通常不會選擇攜帶這麼多的硬幣。對於電腦來說，平均一首 mp3 歌曲的大小可以用 5MB 來表示，改用 $5 \times 1024 = 5120$ KB 雖然意思相同，但是總會讓人覺得有點太過多餘且不實用。

3. 與生活的實際例子結合，手機的售價隨著容量而增加，我們應該隨著自己的使用狀況來決定該買何種規格，才能節省金錢的使用。以一般性舉例，每張照片以 1MB、每首 MP3 以 5MB 估算，程式 APP 則以 500MB 為估計值。假設手機內儲存了 10000 張照片、2000 首 MP3 以及 100 個 APP，則總共佔去 70000 MB，大約 68GB 左右的空間。請學生思考自己的使用習慣，回答適合什麼容量的手機？

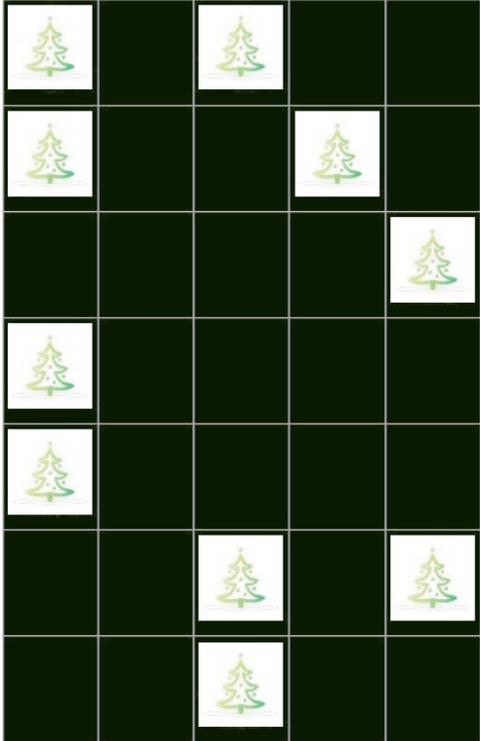
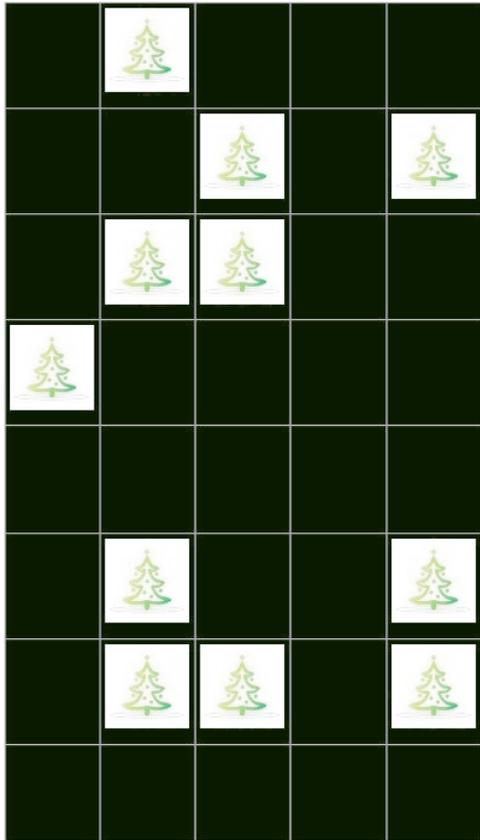
應用 — 傳遞秘密訊息

模擬情境：謝 KK 被困在百貨公司的最上層了！聖誕節就快要到了，他想帶著他的禮物回家。他該怎麼辦？謝 KK 試過大聲求救，但是附近根本沒有人。不過他可以看到對街有一些電腦工程師深夜裡還在工作。他該怎麼引起他們的注意呢？謝 KK 看了一下他的四周，然後靈光一現：他可以用聖誕樹上面的燈傳送訊息給對街正在工作的女士！他找到了所有的燈泡，然後插上電源，這樣他就可以關關這些燈泡。

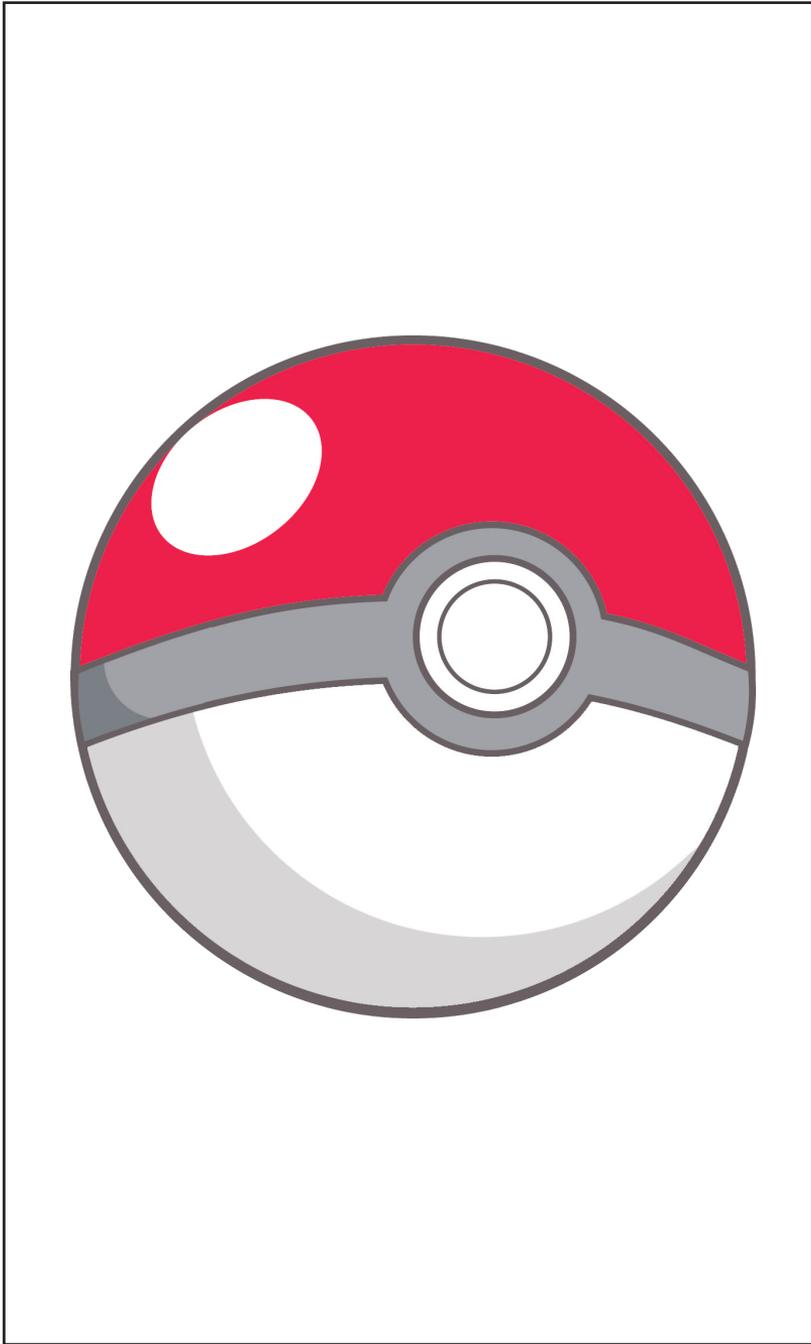
他擬了一則簡單的二進位代碼，確定對街的女士可以懂他的意思。你能幫助他嗎？

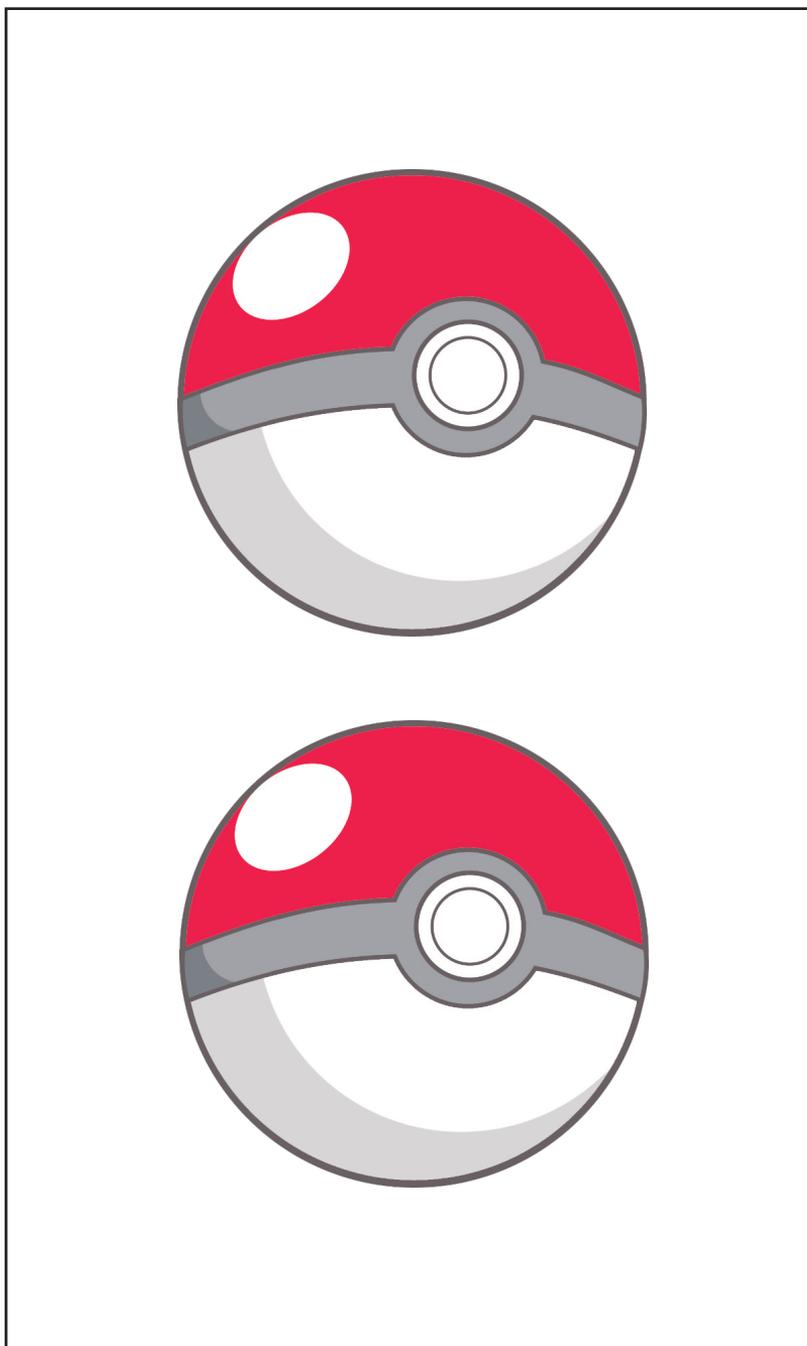
1. 先從以下的二進位燈泡訊息，得到對應的英文句子。(註：全黑代表空格)
2. 開啟 google 翻譯 <https://translate.google.com.tw/>，輸入剛剛得到的句子，看看中文的意思是什麼？(正確答案：help im trapped，翻譯為：救我被困)
(講解時可示範功能，包含語音、多國語言及整篇翻譯。先確認左邊語言選擇的是「英文」、右邊語言則選擇「中文(繁體)」)

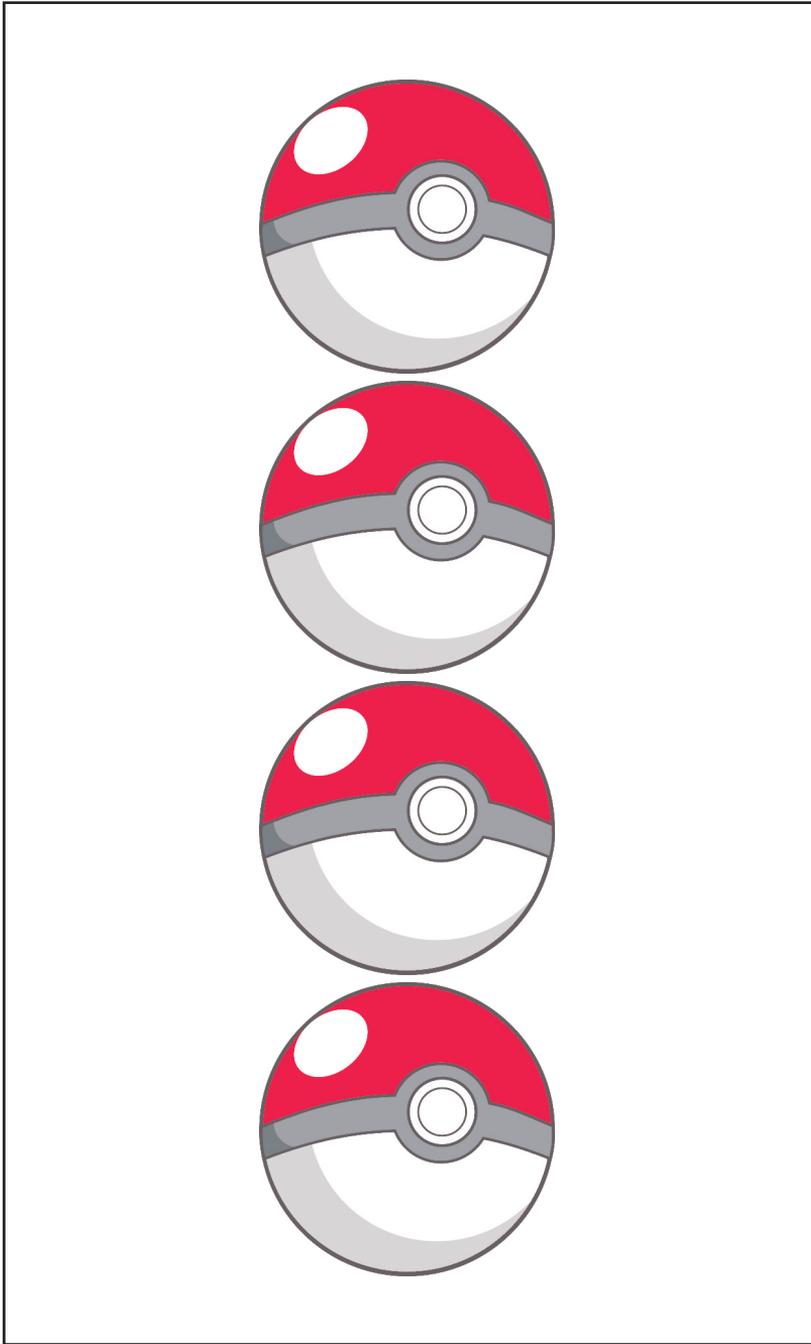
1	2	3	4	5	6	7	8	9	10	11	12	13
a	b	c	d	e	f	g	h	i	j	k	l	m
14	15	16	17	18	19	20	21	22	23	24	25	26
n	o	p	q	r	s	t	u	v	w	x	y	z

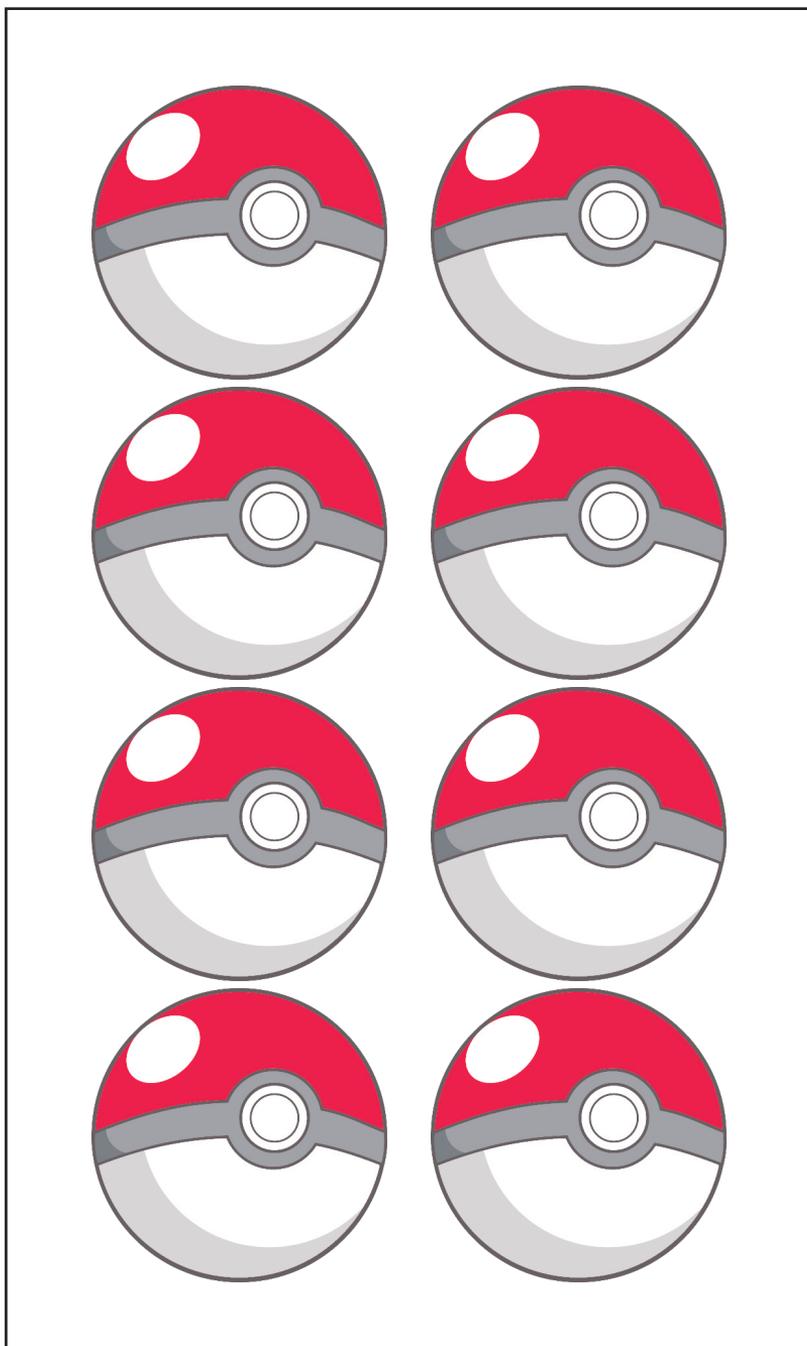


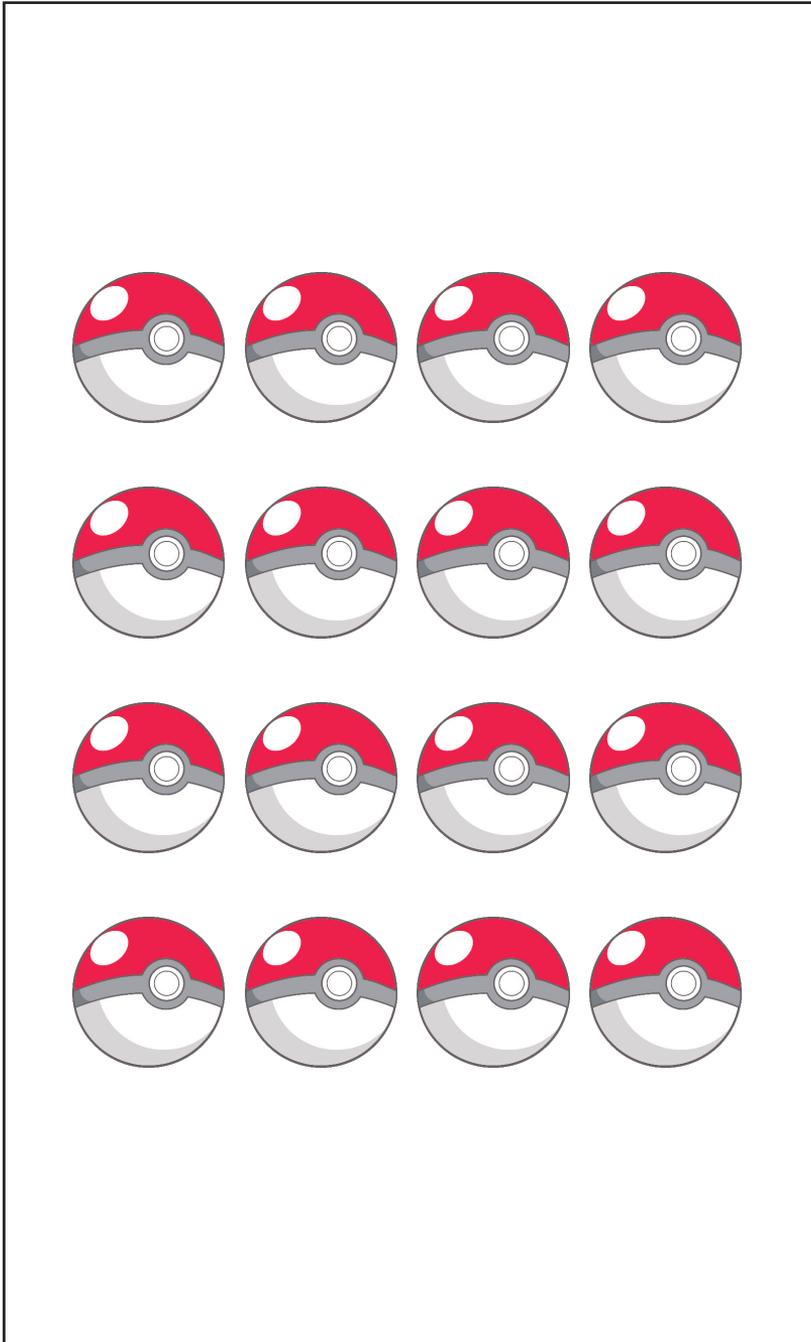
附錄 1-1：教師示範用卡片



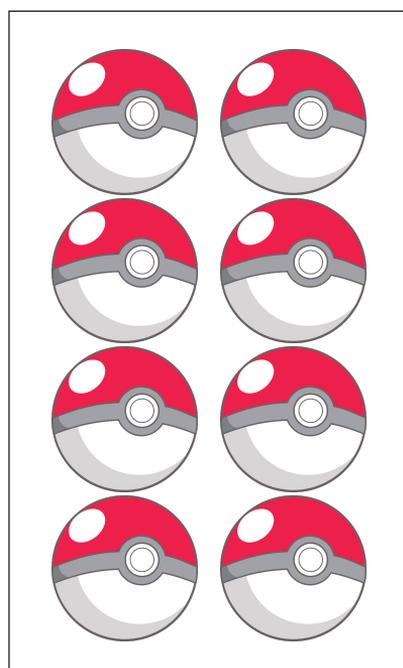
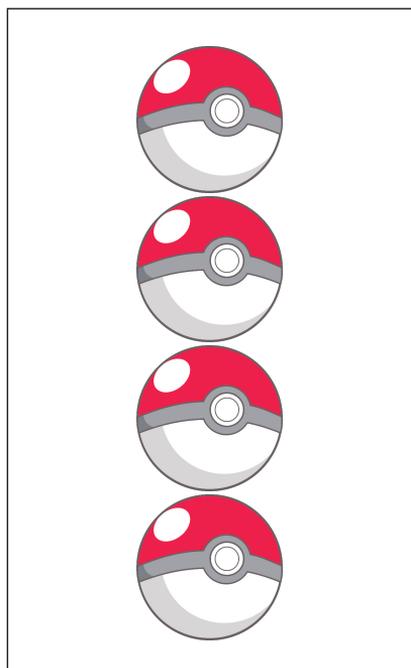
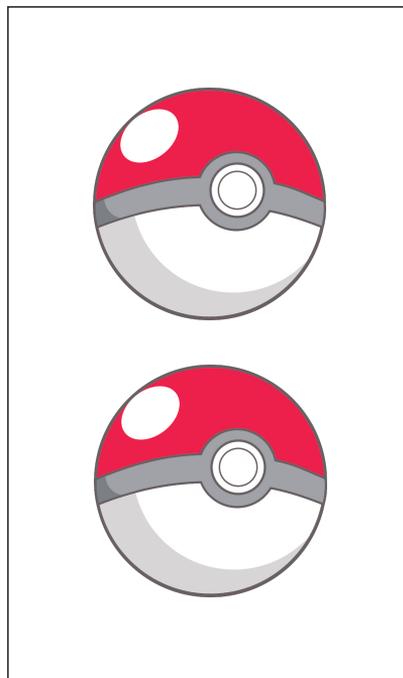
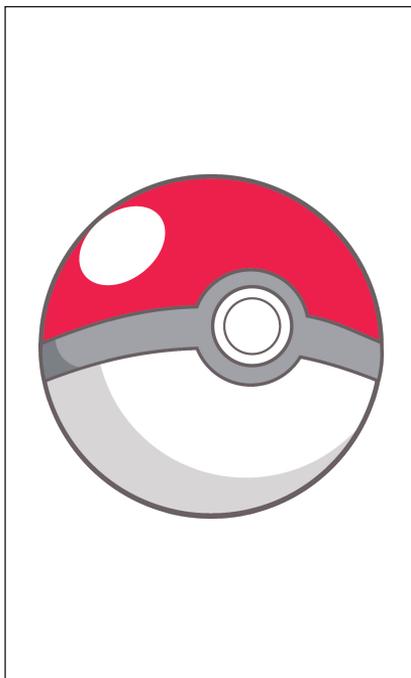


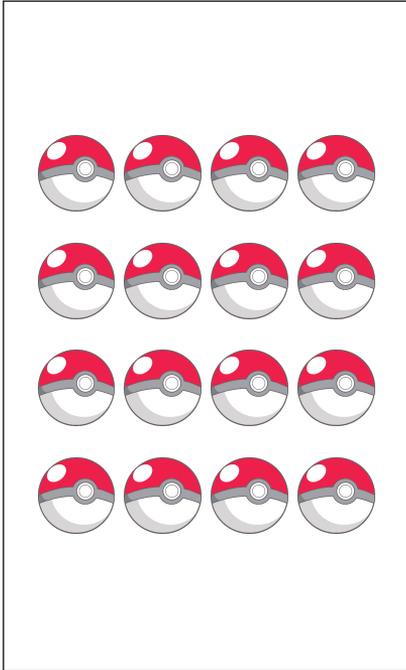






附錄 1-2：學生操作用卡片





第二章 用數字表示圖像

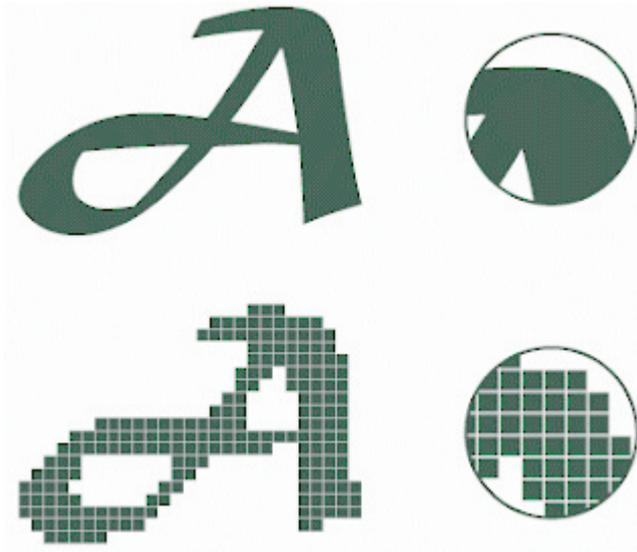
教學時數：3 節

事前準備：

1. 不插電：已編碼的紙本作業在附錄 2-1，另有如附錄 2-2 的空白表格可自行編碼使用。
2. 插電：使用附錄程式檔，資料夾名稱為「html5_colour_by_numbers」。

活動流程一 — 基本概念與黑白圖形的編碼

1. 電腦螢幕被分割成許許多多的小格子，稱為像素 (pixel)。在黑白的畫面下，每個像素不是黑色就是白色。如下列圖形所示：



但是在彩色的畫面下，每個像素都會有自己的顏色。如下列圖形所示：



- 若在電腦教室，可以示範在任意畫面下按鍵盤「PrtSc」螢幕截圖鍵，在小畫家等繪圖軟體之中貼上，接著不斷將顯示比例放大，讓學生實際看到所有的字體與圖案皆是由像素格所組合而成。
- 黑白圖形編碼範例：底下的字母 " a " 已被放大來顯示像素。當電腦儲存圖片時，它要儲存的就是哪些像素是黑的，哪些像素是白的。

	■	■	■		1, 3, 1
				■	4, 1
	■	■	■	■	1, 4
■				■	0, 1, 3, 1
■				■	0, 1, 3, 1
	■	■	■	■	1, 4

第一行由 1 個白色像素，3 個黑色像素，再加 1 個白色像素所組成。因此，第一行被表示為 1, 3, 1。(第一個號碼必須是白色像素的數目。如果第一個像素是黑的，那該行會從 0 開始)

- 展示以下圖形，請學生寫下編碼：

■					
■					
■	■	■	■		
■				■	
■				■	
■	■	■	■		

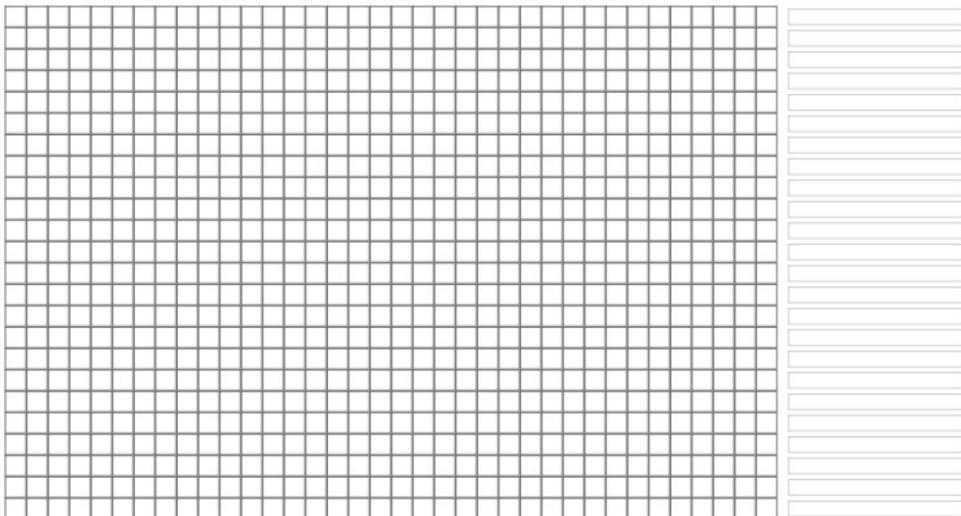
	■	■	■		
■				■	
■					
■				■	
	■	■	■		

正確答案為：

█					0, 1, 4
█					0, 1, 4
█	█	█	█		0, 4, 1
█				█	0, 1, 3, 1
█				█	0, 1, 3, 1
█	█	█	█		0, 4, 1
	█	█	█		1, 3, 1
█				█	0, 1, 3, 1
█					0, 1, 4
█					0, 1, 4
█				█	0, 1, 3, 1
	█	█	█		1, 3, 1

5. 打開「colour_by_numbers」資料夾中的 colour_by_numbers.html，在進行前提示學生先點擊畫面下色綠色的數字，將版面更改為 5 欄、6 行。

Colour by Numbers



36 X 24

請輸入欄數

請輸入行數

避免此頁面產生更多對話框

確定

取消

確定

取消

6. 請學生按以下指定的像素編碼，完成圖案。可以請學生用切換瀏覽器分頁、將題目抄下，或是點擊程式版面右方的空格輸入題目，以利對照完成之用。

0, 5
3, 1, 1
2, 1, 2
1, 1, 3
0, 1, 4
0, 5

0, 1, 3, 1
0, 2, 1, 2
0, 1, 1, 1, 1, 1
0, 1, 1, 1, 1, 1
0, 1, 3, 1
0, 1, 3, 1

正確答案為：

Colour by Numbers

0,5
3,1,1
2,1,2
1,1,3
0,1,4
0,5

5 X 6

Colour by Numbers

0,1,3,1
0,2,1,2
0,1,1,1,1,1
0,1,1,1,1,1
0,1,3,1
0,1,3,1

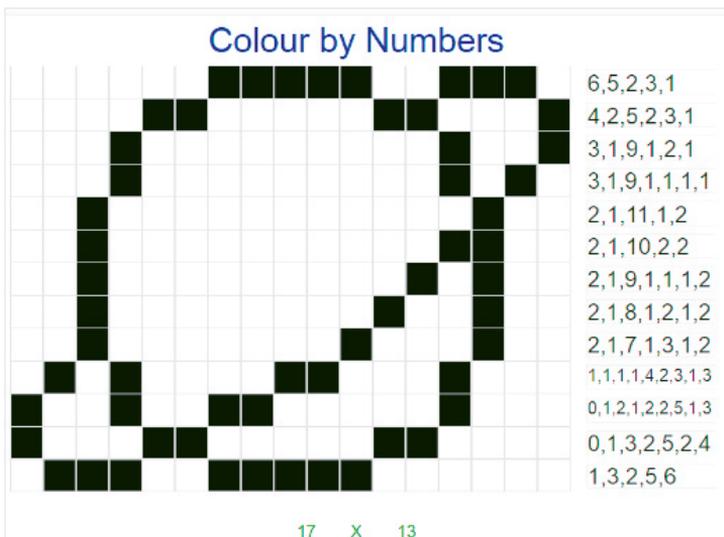
5 X 6

7. 填圖作業，請學生按以下步驟操作，於課堂中完成。

- i. 按指定的題目編碼，設定程式版面為 17 x 13，然後完成圖形。
- ii. 在顯示圖形的畫面，按下鍵盤的「PrtScr」（螢幕列印鍵）。
- iii. 開啟公用 Google 網路硬碟的連結（參考「課前準備」頁面），使用自己的帳號密碼登入。
- iv. 按下左上角「新增」按鈕，選擇「Google 繪圖」。
- v. 在編輯畫面直接按組合鍵「Ctrl + V」，將截圖貼上。
- vi. 點擊左上角「無標題繪圖」處，將檔名改為「班級 + 座號」的 5 位數字組合（也就是學生自己的帳號，不含 domain name）。
- vii. 點擊左上角「開啟 Google 雲端硬碟」的圖案，完成檔案的儲存，至此即完成繳交作業。

6, 5, 2, 3, 1
4, 2, 5, 2, 3, 1
3, 1, 9, 1, 2, 1
3, 1, 9, 1, 1, 1, 1
2, 1, 11, 1, 2
2, 1, 10, 2, 2
2, 1, 9, 1, 1, 1, 2
2, 1, 8, 1, 2, 1, 2
2, 1, 7, 1, 3, 1, 2
1, 1, 1, 1, 4, 2, 3, 1, 3
0, 1, 2, 1, 2, 2, 5, 1, 3
0, 1, 3, 2, 5, 2, 4
1, 3, 2, 5, 6

正確答案為：



延伸結論

1. 傳真機實際上就只是一台簡單的電腦，掃描黑白頁面成大約 1000×2000 像素，並使用數據機傳輸給另一台傳真機，最後將像素印在紙上。傳真的圖像常常有著部份空白（邊緣）或黑色像素（垂直的線）。彩色圖片也會有許多重覆部份。為了節省儲存空間，開發人員會使用很多種壓縮技術。而在這活動中使用的方法稱為變動長度編碼法（RLE, run-length encoding，又稱為游程編碼），是一種有效的壓縮圖片的方式。如果我們不壓縮圖片，那可能會花很多的時間來傳送它，而且也需要更多的儲存空間。這樣會讓傳真或將圖放在網路上變得成本太高而不可行。舉例來說，傳真圖片通常被壓縮至原大小的七分之一左右。也就是說，如果不壓縮，光傳輸就會多出七倍的時間。相片或圖片常被壓縮至原大小的十分之一或甚至百分之一（使用像是 JPEG、GIF 或 PNG 等技術）。這樣一來就可以在磁碟中儲存更多圖片，而這也表示在網站瀏覽時可以花更少的時間。
2. 對於學生關心的照相問題，在選擇相機時最常注意到的規格標示就是「畫素」了。畫素的多寡與照片的成果好壞沒有絕對的關係，影響最大的在於「輸出」（也就是沖洗成照片）的時候。將畫素較少的檔案輸出成大尺寸的照片，就像我們把一張圖片的顯示比例一直放大，將可能看到許多的畫面鋸齒，反之畫素較多的檔案就能輸出為更大張的照片。
3. 既然談到拍照，不如跟學生聊聊拍照的話題。請他們反過來跟老師分享自拍的訣竅，或是常常使用什麼樣的拍照 APP，可能會得到許多新奇的答案喔！

第三章 資訊理論

教學時數：3 節

事前準備：

1. 不插電：請學生準備空白紙與文具用品即可。
2. 插電：使用下方連結進行珠璣妙算 Master Mind 遊戲，若連結失效可搜尋其他網頁替代。（注意頁面是否會出現不適宜的廣告）

<http://www.archimedes-lab.org/mastermind.html>

活動流程一 — 資訊的概念（問答與討論）

透過以下問答來討論何謂資訊 (Information)？

1. 我們如何得知在一本書中有多少的資訊？首先閱讀目錄與推薦序，可以先大略知道這本書大概在談些什麼內容。
2. 書中的頁數和字數是重要的因素嗎？比起內容，頁數與字數不是最重要的事情。
3. 一本書會比另外一本有更多的資訊嗎？取決於「實用性」，教科書相對於漫畫書，因為牽涉到考試用途，所以相對被認為擁有比較多的資訊。若是將漫畫書視為考試科目，那裡面所談的任何事情就因此被視為資訊。
4. 一本非常無聊或是特別有趣的書，哪個包含的資訊量較多？如上一個問答，趣味性與實用性並不相等。
5. 一本 400 頁但都是廢話的書，會比一本電話簿含有更多的資訊嗎？如果常常需要查詢電話簿的內容，則這本電話簿相對含有更多的資訊，但若平時並沒有使用電話簿的習慣，那裡面的內容就不被視為資訊了。

通常我們藉由某段訊息 (或者某本書) 「讓人驚訝的程度」來測量資訊。

例如：一個總是走路到學校的朋友告訴你：「我今天走路到學校」，你一點都不會感到驚訝，因為你早就知道了。

但是如果你的朋友告訴你：「我今天坐著直升機到學校」。這樣的訊息會讓你很驚訝，接下來他告訴你的事情就會包含非常多的資訊。

如何測量某段訊息的「驚訝值」？

看看要猜到那個資訊會有多困難。

如果你的朋友告訴你：「猜猜我今天是如何到學校的？」而答案是他走路到學校，這個時候你會有比較高的機會在第一次就猜到正確答案。但如果答案是坐直升機，甚至太空船，那麼有可能得猜好幾次才能猜到答案。

因此，某個訊息所包含的資訊量，是藉由是否容易被猜到來衡量的。

活動流程二 — 猜測的策略，終極密碼（猜數字遊戲）

由一個自願者擔任出題者，在 1~100 之間選擇一個數字作為題目。其他人依需猜測數字，每猜測一次且並未正確猜中題目時，出題者要回答題目在剩下的數字範圍區間，下一人再接著從這個範圍區間之中猜測，直到有人猜中數字為止。

例如出題者選擇「37」為題目，第一位猜測「54」，則出題者回答「1 到 54」。第二位猜測「22」，則出題者回答「22 到 54」，依此類推。

有時候終極密碼的遊戲會在聚餐進行到尾聲的時候進行。由與會者輪流擔任出題者，其他人進行猜測，不小心猜到題目數字者必須吃掉指定的剩菜。在教室以活動方式進行，因為不適合使用處罰手段，所以改變遊戲方式，請學生以「儘快猜到題目」為目標。使用範圍 1~100 進行幾回合的遊戲，請學生思考應該使用什麼策略才能更快地猜中題目數字。（提示：每次將範圍縮小一半）

接著進行以下問題討論，在出題方式改變之下，猜測的策略是否仍然一樣，還是變得不可猜測了呢？

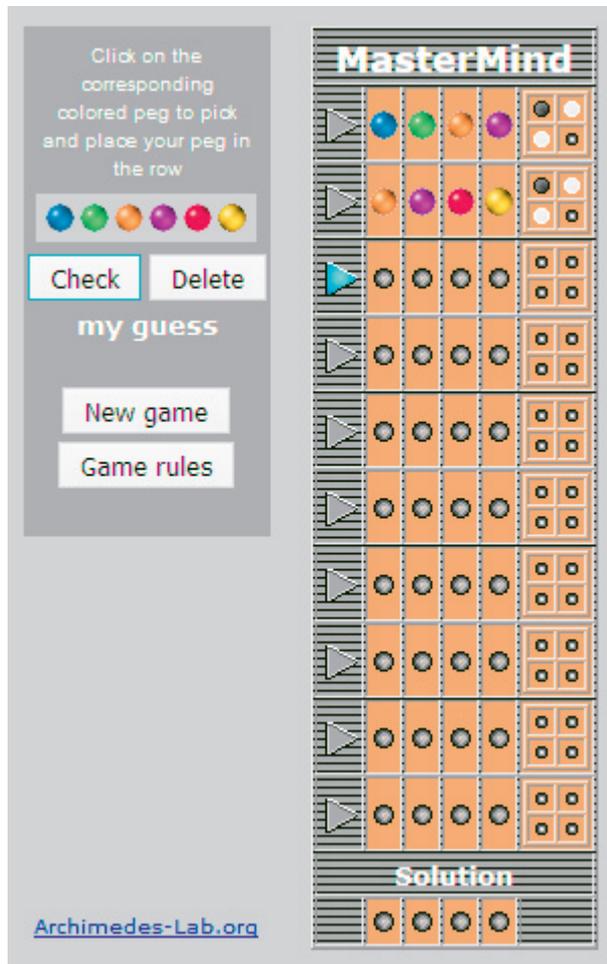
1. 猜 1 到 1000 之間的數字
2. 猜 1 到 100 萬之間的數字
3. 任何一個整數
4. 有特定模式的一串六個數字，但模式只有出題者自己知道（回答時必須從第一個到最後一個依序回答），例如 2、4、6、8、10、12

對於前兩個題目來說，可以同樣使用每次將範圍縮小一半的策略來進行，並不會多花費多少時間。但後面兩個題目因為沒有一個明確的範圍可作為線索（整數包含負整數、0、以及整數，無法界定極大與極小的範圍。特定模式的數字，由於模式並未公開說明，範圍同樣無法定義），因此是無從猜測起的。

活動流程三 — 珠璣妙算 Master Mind

使用網路瀏覽器開啟遊戲連結，讓學生遊玩並探索猜測的策略。使用以下網址為例：<http://www.archimedes-lab.org/mastermind.html>

1. 總共有 6 個顏色的球，依你的猜測依序點擊。
2. 按下「Delete」可重新輸入猜測、按下「Check」後檢查你的答案。
3. 顏色、位置皆正確，右方顯示一顆「黑球」；顏色正確但位置錯誤，則顯示一顆「白球」。
4. 反覆猜測，在十次猜測機會中得到四顆球的正確顏色與位置。
5. 注意同樣顏色的球可以重覆出現。



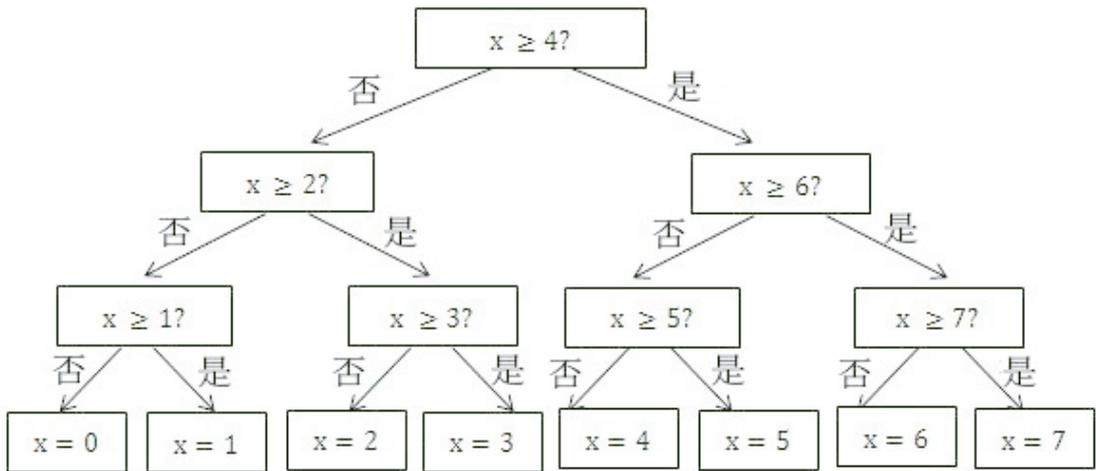
活動流程四 — 決策樹的繪製

先前提到的猜數字遊戲，假設題目訂為 0 到 7 之間的某個數字，若我們想嘗試用最少的猜測次數來猜到答案，可以使用畫圖的方式來展現猜測的策略。

猜測的訣竅是每次都將猜測的範圍縮小一半，所以第一次詢問出題者，就問他題目有沒有大於或等於 4？

出題者若回答「否」，那代表題目可能的範圍只剩下 0、1、2 或 3 這四個數字，範圍馬上縮小了。

依此類推，我們把詢問的過程畫成下面的樹狀圖：



空格內為提問，箭頭方向則是對於提問的回應。依據每次縮小一半範圍的猜測策略，第一次詢問題目是否大於或等於 4？若出題者回答「是」，則剩餘的可猜測範圍就是 4、5、6、7 四個數字。第一次詢問題目是否大於等於 6？若出題者回答「否」，則可猜測範圍就剩下 4 或 5 兩個數字。最後只要再詢問題目是否大於或等於 5，就能確定正確答案為何了。

說明完決策樹的畫法後，接著與學生進行以下問答：

1. 要猜到數字 5 需要做哪些決定（是／不是）？（按「是」、「否」、「否」的順序進行即可）
2. 你需要多少決定（是／不是）才能猜測出任何數字？（3 次）
3. 在樹最下方的數字下方，寫下這 8 個數字的二進位表示法。提示學生，依照之前學習二進位數字的方法，設定使用的卡片為 3 張。
4. 仔細觀察決策樹，如果路徑「不是」= 0、「是」= 1，你發現了什麼？（路徑走法與數字的二進位表示法是一致的，例如數字 6 的二進位數字表示為 110，在決策樹上的走法為「是」、「是」、「否」，即對應為「1」、「1」、「0」)

請學生在紙上畫出「猜測數字範圍 0~15」的猜測決策樹。（給老師的提示：樹的大小會比 0~7 的決策樹大一倍）

延伸結論

猜測的方法能被應用來建造一個預測使用者輸入的電腦介面。這對於打字有困難的身心障礙人士相當有幫助。電腦可以建議他們接下來可能要輸入什麼，而他們只需要選擇他們想要的字即可。這樣的系統也以不同的形式被應用在手機打字的介面上。

第四章 錯誤的發現與修正

教學時數：6 節

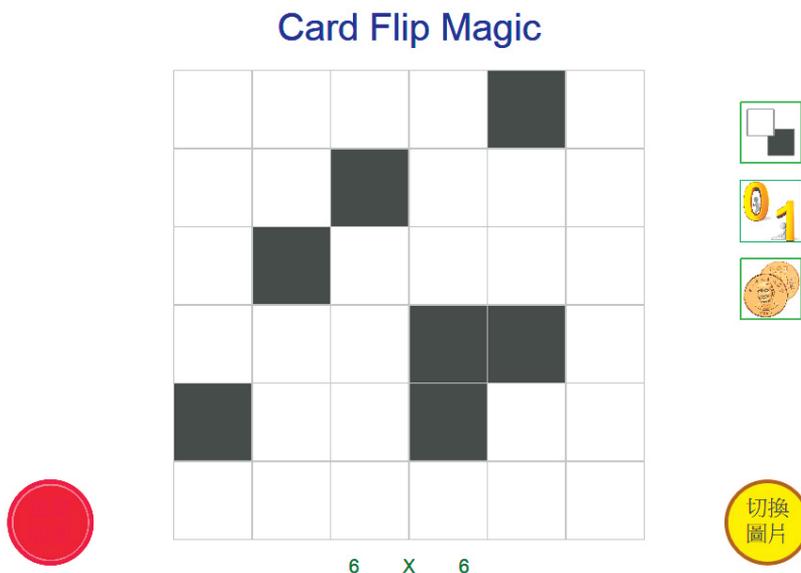
事前準備：

1. 不插電：請學生準備空白紙與文具用品即可。若要以不插電的方法演示卡片翻轉魔術，可以準備正方形紙卡 36 片（正面白色、反面黑色，想吸附在黑板或白板上，則選擇以軟性磁性加工），或是準備 36 個硬幣亦可。
2. 插電：附錄程式檔，資料夾名稱為「card_flip_magic」，以及使用 google 試算表進行專題作業。教師亦可選用已安裝好的其他試算表軟體。

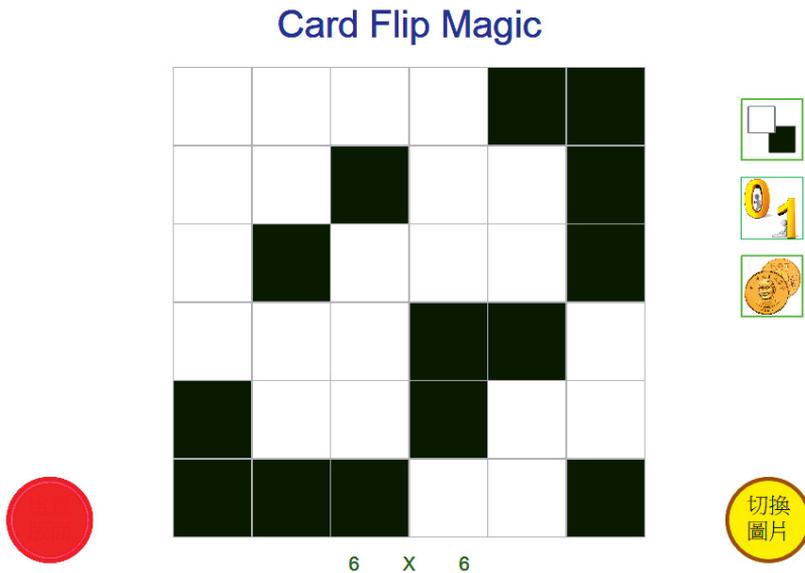
活動流程一 — 卡片翻轉魔術

當資料儲存在磁碟或是要從一台電腦傳送到另一台時，我們會希望在傳送過程中，資料不會因為任何原因而被改變。然而，有時候還是會發生一些意外性錯誤，使資料毀損或改變。在以下活動中，我們會使用一個魔術般的手法，在資料受損而被改變時能夠發現並修正它。

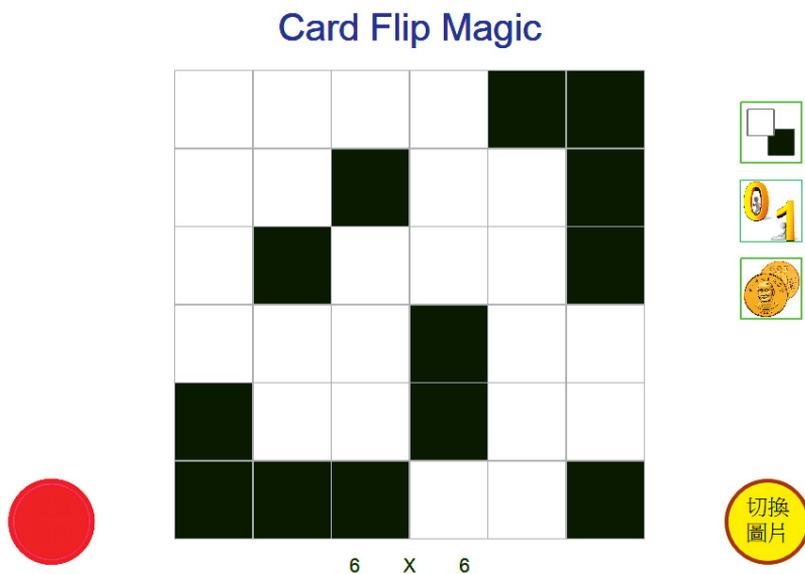
1. 開啟「card_flip_magic」資料夾中的 card_flip_magic.html，以鎖定螢幕或投影方式展示給全班看。
2. 找一位自願的同學，指定只能使用畫面上 5 x 5 的區域，可以隨意指定這個範圍的空格為黑色或白色。假定結果如下：



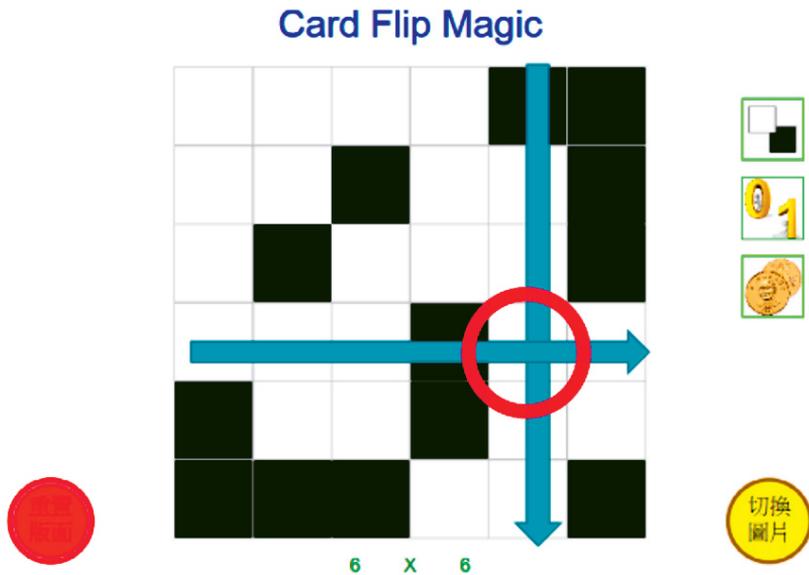
3. 告訴學生，最右邊與最底下這排，會由老師「任意」增加一些黑色的格子，目的是讓畫面看起來更複雜（實際上是有規則的，讓每個橫排與直行的黑色格子都是偶數）。因此依據上圖，加上黑色格子的結果如下：



4. 再找一位自願的同學在版面上動手腳，任意選擇一個方格，讓它由黑色變成白色、或是由白色變成黑色。提醒學生要配合規則，只能夠挑選「一個格子」動手腳。老師可以蒙上眼睛、背對螢幕或是離開教室，證明自己沒有偷看的機會。（對了，要提醒學生，動完手腳之後記得把滑鼠移開喔！）假設下圖是已被動過手腳的版面：

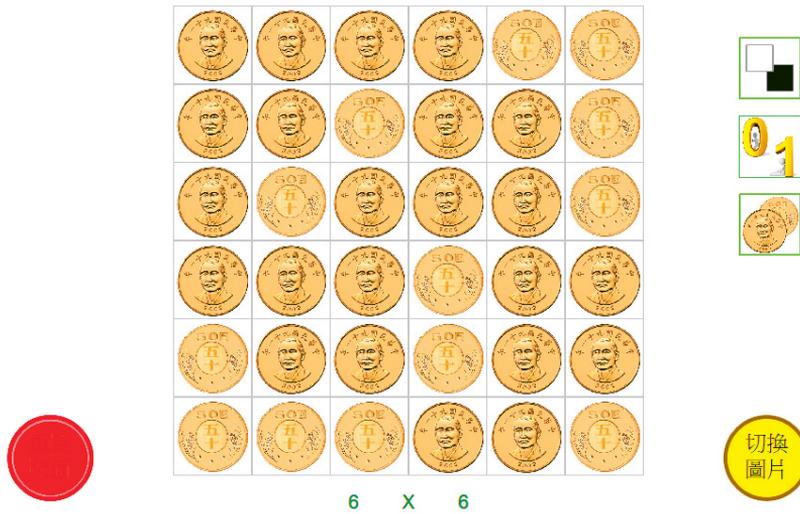


5. 老師只要快速地看過版面一眼，就可以很快地找出被翻面的那個空格。先請學生猜猜老師是如何做到的？老師所增加的黑色空格稱為同位卡 Parity Card，可以透過尋找奇數個黑色格子的直排、橫列交叉點，快速的定位要找的正確位置。



6. 進行討論，如果同學不按照約定，同時改變了兩個以上的空格顏色，還能找出來被翻動過的卡片嗎？答案是不行的，老師可以實際操作一遍示範給同學看看。
7. 點擊畫面下方 6 x 6 的數字，嘗試更改擴大卡版面（例如變成 9 x 9），用同位卡的操作讓它變成 10 x 10，再重新操作這個魔術一次。
8. 教師可以拖曳右邊的硬幣圖案到「切換圖片」按鈕上，告訴學生這個魔術也可以用硬幣或是任何一種有正、反兩面的東西來操作。
9. 若教師想做更深入的解說，可以拖曳右邊「01」的圖案到「切換圖片」按鈕上，以 0 與 1 解釋同位卡的概念。

Card Flip Magic



活動流程二 — 書碼與條碼

已發行的書本上，會有一組 10 或 13 位數的數字，通常會印在書背。最後一位的數字就是檢查碼，就像卡片翻轉魔術中的同位卡片一樣。

這表示如果你訂購了一本使用了 ISBN（International Standard Book Number，國際標準書號）的書，網站能幫你檢查書號有沒有錯誤。只需要依據規則檢查同位檢查碼就好了，這樣就不會等了老半天，卻拿到錯誤的書本。

食品雜貨所使用的條碼，用的是不同的公式。如果條碼讀錯了，掃描器會發出嗶嗶的聲音，櫃檯的結帳人員就會重新掃描一次條碼。此外，檢查碼的概念還會應用在銀行帳號、身份證字號、火車等大眾交通工具等。



以下請學生按照 ISBN 的檢查碼規則，進行紙筆的運算操作。學生會很不喜歡這個部份，但是老師一定要耐著性子讓他們完成，這關係到後續的試算表操作作業。

ISBN-10 的檢查碼

條碼為 0-13-911991-4 的書，依據以下公式，進行檢查碼的驗證：

1. 第一步為相乘、加總。第一碼乘以 10、第二碼乘以 9 … 依此類推第九碼乘以 2，然後把乘積相加：

$$(0 \times 10) + (1 \times 9) + (3 \times 8) + (9 \times 7) + (1 \times 6) + (1 \times 5) + (9 \times 4) + (9 \times 3) + (1 \times 2) = 172$$

2. 第二步為除以 11 取餘數

$$172 \div 11 = 15 \text{ 餘 } 7$$

3. 第三步為餘數的判斷。如果餘數剛好是 0（整除），則檢查碼就是 0。如果不是的話，就用 11 減掉餘數，得到的值就是檢查碼。

$$11 - 7 = 4$$

4. 最後檢查該 ISBN 書號的最後一碼，跟算出來的檢查碼是否一致。必需是一致的，才代表這個 ISBN 碼正確。

請學生練習找出「986-470-526-□」空格內的數字。（答案為 1）

ISBN-13 的檢查碼

條碼為 978-986-181-728-6 的書，依據以下公式，進行檢查碼的驗證

1. 第一步同樣為相乘、加總。第一碼乘以 1、第二碼乘以 3… 依此類推交替乘以 1 與 3，然後把乘積相加：

$$(9 \times 1) + (7 \times 3) + (8 \times 1) + (9 \times 3) + (8 \times 1) + (6 \times 3) + (1 \times 1) + (8 \times 3) + (1 \times 1) + (7 \times 3) + (2 \times 1) + (8 \times 3) = 164$$

2. 第二步為除以 10 取餘數：

$$164 \div 10 = 16 \text{ 餘 } 4$$

3. 第三步為餘數的判斷。如果餘數剛好是 0（整除），則檢查碼就是 0。如果不是的話，就用 10 減掉餘數，得到的值就是檢查碼。

$$10 - 4 = 6$$

4. 最後檢查該 ISBN 書號的最後一碼，跟算出來的檢查碼是否一致。必需是一致的，才代表這個 ISBN 碼正確。

請學生練習找出「978-986-470-666-□」空格內的數字。（答案為 2）

再請學生試著找出以下兩題各三個選項中，正確的那一個。這個時候學生會開始央求老師別再讓他們算數學了，老師可以開放讓學生使用電腦內建的小算盤軟體，順便告訴學生電腦所使用的乘除符號，跟平常的書寫習慣不同，乘號是「*」、除號是「/」。使用小算盤只能輔助計算相乘加總的部份，但是取餘數的時候就沒有這麼方便（算出來可能是小數，代表「商」），最後判斷餘數的部份則是不需要使用到小算盤，直接用心算即可。告訴學生，之後我們會使用電腦來幫我們做計算，只要設計過一次，日後直接輸入數字馬上就可以得到結果。

ISBN-10

- (A) 986-929-945-8
- (B) 986-235-536-0
- (C) 986-235-543-3

ISBN-13

- (A) 978-986-477-101-3
- (B) 978-957-136-792-7
- (C) 978-986-931-045-9

延伸應用：

使用網路書店。按照老師的指示，打開任一網路書店的網頁，找到任一書本的 ISBN 碼後，利用公式實際檢查該書本的 ISBN 碼是否正確。透過網路書店的進階搜尋功能，輸入在剛剛的練習題中得到的正確 ISBN-13 碼，看看找到的是什麼書。

結論：

想像你將 10 塊錢現金存入銀行的帳戶。出納員整理出所有存款的量，並傳送到中央電腦。假設在傳送的過程中，線路發生了一些干擾，結果數字從 10 塊錢變成 1000 塊。雖然你會覺得開心，但是這對出納員來說是一個天大的災難。

在傳送資料時檢查錯誤是非常重要的。所以接收端的電腦需要檢查傳過來的資料沒有因為某些干擾造成損毀。有時候發生傳送錯誤時，原始資料可以再傳送一次，但有些情況則是不行，比方說磁碟因為曝露在磁力或電力輻射或是高溫下，或是任何其它的物理性破壞而造成資料損毀。如果資料是從深遠的太空偵測器傳送過來，那麼當錯誤發生時，等待重新傳送將會需要非常長的一段時間。(若在木星最靠近地球時，傳送訊息至地球大約需要半個小時)

因此，當資料損毀時，我們需要有辦法發現 (錯誤檢查)，並且能重建原始的資料 (錯誤修正)。在卡片魔術活動時所用的技巧，藉由加入同位檢查位元到每一列與每一行，這樣我們不只能發現錯誤，還能找出錯誤的位置。這個出問題的位元會被改回去，這就是錯誤修正。

當然，電腦使用的錯誤控制系統通常會複雜得多，以檢查和修正多重錯誤。電腦中的硬碟裡，有很大量的空間是被用來做錯誤修正的。這樣即使部份的硬碟壞了，也還是可以工作。

活動流程三 — 以試算表作 ISBN 驗證碼的檢查與產生

ISBN-10 檢查碼的產生

1. 以 Google 帳戶登入公用網路硬碟。
2. 按下左上角的「新增」按鈕，點選「Google 試算表」，開啟一個空白的文件。
3. 在第一橫列上，從 B1 開始依序在每一欄輸入前 9 個號碼 013911991 (一個數字一個欄位)
4. 運算 1：按以下公式算出結果，置於儲存格 B3

=B1*10+C1*9+D1*8+E1*7+F1*6+G1*5+H1*4+I1*3+J1*2

A	B	C	D	E	F	G	H	I	J
ISBN-10	0	1	3	9	1	1	9	9	1
運算 1	=B1*10+C1*9+D1*8+E1*7+F1*6+G1*5+H1*4+I1*3+J1*2								

5. 運算 2：使用函數 MOD，取除以 11 的餘數，運算結果置於儲存格 B4

fx | =MOD(B3,11)

	A	B
1	ISBN-10	0
2		
3	運算1	7 × 172
4	運算2	=MOD(B3,11)

6. 運算 3：使用函數 IF，若餘數為 0，則檢查碼 = 0；否則檢查碼為 11 減掉該餘數。運算結果置於 B5，B5 之數值即為第 10 碼檢查碼。

fx | =IF(B4=0,0,11-B4)

	A	B	C	D
1	ISBN-10	0	1	3
2				
3	運算1	172		
4	運算2	4 × 7		
5	運算3	=IF(B4=0,0,11-B4)		
6		IF(邏輯運算式, 是_true_的值, 是_false_ ^ X 的值)		
7		範例		
8		IF(A2 = "foo", "A2 是 foo", "A2 不是 foo")		
9		摘要		
10		邏輯運算式是「TRUE」時傳回一個值，是「FALSE」時傳回另一個值。		
11		邏輯運算式		
12		運算式或某儲存格的參照，這個儲存格必須包含代表某個邏輯值(亦即「TRUE」或「FALSE」)的運算式。		
13		是_true_的值		
14		「邏輯運算式」是「TRUE」時傳回的函式值。		
15		是_false_的值		
16		「邏輯運算式」是「FALSE」時傳回的函式值。		
17		進一步瞭解「IF」		
18				
19				

7. 實際在試算表中輸入幾個題目，取得正確的檢查碼：

- (1) 986-470-526-□
- (2) 986-929-945-□
- (3) 986-235-536-□
- (4) 986-235-543-□

8. 操作完畢後，更改左上方檔案名稱為「帳號 -ISBN10」，按下左上角綠色按鈕以儲存退出。

ISBN-13 檢查碼的產生

1. 以 Google 帳戶登入公用網路硬碟。
2. 按下左上角的「新增」按鈕，點選「Google 試算表」，開啟一個空白的文件。
3. 在第一橫列上，從 B1 開始依序在每一欄輸入前 12 個號碼 978 986 181 728 (一個數字一個欄位)
4. 運算 1：按以下公式算出結果，置於儲存格 B3

ISBN-13		9	7	8	9
運算 1	=B1*1+C1*3+D1*1+E1*3+F1*1+G1*3+H1*1+I1*3+J1*1+K1*3+L1*1+M1*3				

5. 運算 2：使用函數 MOD，取除以 10 的餘數，運算結果置於儲存格 B4

	A	B	C	D
1	ISBN-13	9	7	8
2				
3	運算 1	4	164	
4	運算 2	=MOD(B3,10)		
5	運算 3	MOD(被除數, 除數)		
6				
7				
8				
10				
11				
12				
13				
14				

範例
MOD(10, 4)

摘要
傳回模數運算子的結果 (相除後剩下的餘數)。

被除數
為求餘數而必須被除的數字。

除數
除以被除數的數字。

[進一步瞭解「MOD」](#)

6. 運算 3：使用函數 IF，若餘數為 0，則檢查碼 = 0；否則檢查碼為 11 減掉該餘數。運算結果置於 B5，B5 之數值即為第 10 碼檢查碼。

	A	B	C	D
1	ISBN-13	9	7	8
2				
3	運算1	164		
4	運算2	6 × 4		
5	運算3	=IF(B4=0,0,10-B4)		
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				

fx | =IF(B4=0,0,10-B4)

IF(邏輯運算式, 是_true_的值, 是_false_ ^ × 的值)

範例
IF(A2 = "foo", "A2 是 foo", "A2 不是 foo")

摘要
邏輯運算式是「TRUE」時傳回一個值，是「FALSE」時傳回另一個值。

邏輯運算式
運算式或某儲存格的參照，這個儲存格必須包含代表某個邏輯值(亦即「TRUE」或「FALSE」)的運算式。

是_true_的值
「邏輯運算式」是「TRUE」時傳回的函式值。

是_false_的值
「邏輯運算式」是「FALSE」時傳回的函式值。

[進一步瞭解「IF」](#)

7. 實際在試算表中輸入幾個題目，取得正確的檢查碼
- (1) 978-986-470-666-
 - (2) 978-986-477-101-
 - (3) 978-957-136-792-
 - (4) 978-986-931-045-
8. 試操作完畢後，更改左上方檔案名稱為「帳號-ISBN13」，按下左上角綠色按鈕以儲存退出。

活動流程四 — 以試算表製作身份證字號產生器

學生喜歡在感興趣的網站上註冊帳號，往往需要輸入許多的個人資料才能通過驗證，若是不想透露自己的身份證字號，可能選擇放棄註冊或是直接去網路下載已經製作好的身份證字號產生器程式，但是這個動作可能會有感染電腦病毒或木馬程式的風險。只要使用試算表軟體，就能製作一個自己的身份證字號產生器了！

1. 首先帶學生認識身份證字號的英文字母代碼，實際上每個字母都代表一個二位數的數字。(注意一下，表格中有些英文字母不見了，是為什麼呢？)
(參考規則：[https://zh.wikipedia.org/wiki/ 中華民國國民身分證](https://zh.wikipedia.org/wiki/中華民國國民身分證))

字母	轉換字元	縣市	字母	轉換字元	縣市
A	10	台北市	M	21	南投縣
B	11	台中市	N	22	彰化縣
C	12	基隆市	O	35	新竹市
D	13	台南市	P	23	雲林縣
E	14	高雄市	Q	24	嘉義縣
F	15	新北市	T	27	屏東縣
G	16	宜蘭縣	U	28	花蓮縣
H	17	桃園市	V	29	台東縣
I	34	嘉義市	W	32	金門縣
J	18	新竹縣	X	30	澎湖縣
K	19	苗栗縣	Z	33	連江縣

2. 與先前一樣的操作方法，先在公用網路硬碟中新增一個試算表的空白檔案。找一個地方輸入英文字母與轉換字元的對照，縣市的中文字部份可以視進度決定要不要請學生輸入。

	A	B	C
1	試算表版身份證字號產生器		
2			
3	縣市代碼對照區		
4	英文字母	縣市	代碼
5	A	台北市	10
6	B	台中市	11
7	C	基隆市	12
8	D	台南市	13
9	E	高雄市	14
10	F	新北市	15
11	G	宜蘭縣	16
12	H	桃園市	17
13	I	嘉義市	34
14	J	新竹縣	18
15	K	苗栗縣	19
16	M	南投縣	21
17	N	彰化縣	22
18	O	新竹市	35
19	P	雲林縣	23
20	Q	嘉義縣	24
21	T	屏東縣	27

3. 決定一個區域為數字區，每個英文字母與數字佔一欄，第九碼暫時留白。

E	F	G	H	I	J	K
輸入區(第9碼自動產生, 不需輸入)						
英文代號	第1碼	第2碼	第3碼	第4碼	第5碼	第6碼
A	1	2	3	4	5	6
計算區						
英文代碼轉換	10					
加權總合	121					
驗證碼	9					
結果輸出區						
	A123456789					

4. 英文代號與代碼的轉換，透過函數 LOOKUP 達成。以此範例的儲存格編號，來源設為 E5、搜尋範圍 A5:A26、對照範圍 C5:C26（冒號代表一個連續的儲存格範圍）

=LOOKUP(來源, 搜尋範圍, 對照範圍)

5. 接下來先把轉換後的兩位數拆成分開的兩個數字，使用 INT 函數取得十位數、使用 MOD 函數取得個位數。可以使用兩個儲存格分別輸入函數。

=INT(代碼儲存格編號 /10)

將數字除以 10 之後所得的小數，無條件捨去小數部份

=MOD(代碼儲存格編號,10)

將數字除以 10 之後取餘數，即為該數字的個位數字

英文字母轉換後的第一個數字	1	→	f_x	=INT(F8/10)
英文字母轉換後的第二個數字	6	→	f_x	=MOD(F8,10)

6. 計算加權總合，公式是將每個數字，依序乘以「1、9、8、7、6、5、4、3、2、1、1」再相加。其中第一碼為上一步驟中，將英文字母轉換為二位數代碼的十位數字；第二碼為二位數代碼的個位數字；其餘則為身份證數字部份的第一碼至第八碼。

f_x | =J8*1+J9*9+F5*8+G5*7+H5*6+I5*5+J5*4+K5*3+L5*2+M5*1

7. 取加權總合除以 10 的餘數，使用函數 MOD 達成

=MOD(加權總合儲存格編號, 10)

8. 判斷驗證碼 (第九碼)，使用 IF 函數達成，若上一步驟所算出的餘數為 0，則驗證碼為 0，否則驗證碼為 10 減去該餘數。

=IF(餘數儲存格編號 = 0, 0, 10- 餘數儲存格編號)

fx | =IF(J12=0,0,10-J12)

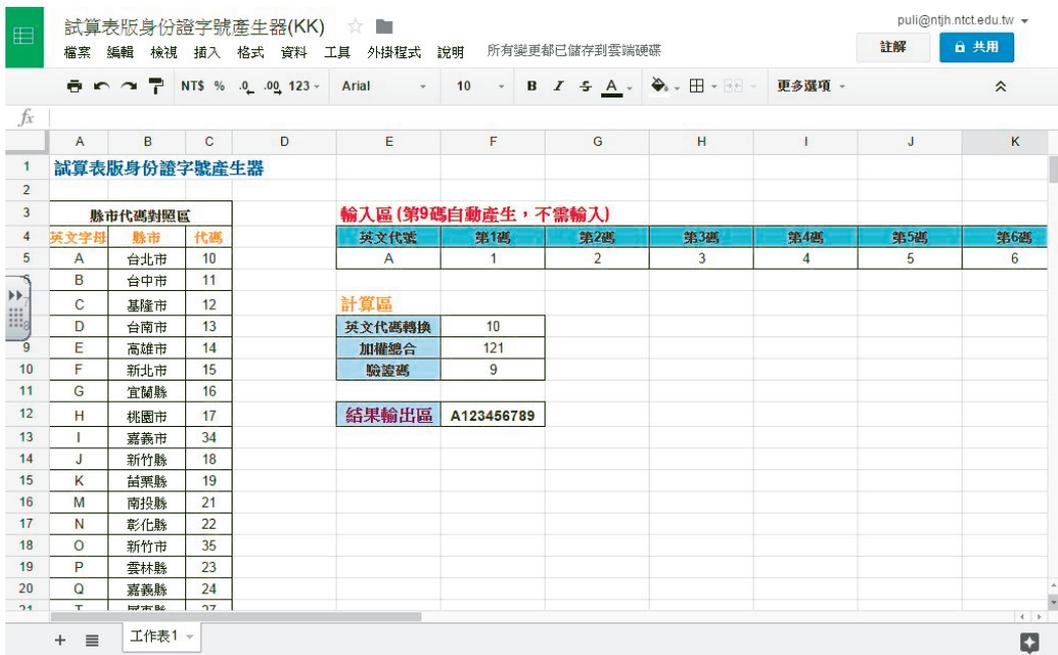
9. 在輸入區為本留白的「第九碼」欄位，指定參照計算驗證碼的那個儲存格。(亦可直接在該欄位進行 IF 函數的判斷)

10. 合併輸入區的所有欄位，從英文字母開始一直到第九碼，使用函數 CONCATENATE 達成。(註：在 Google 試算表中，可以使用「冒號」一次標記範圍。若要以 Microsoft Excel 或 LibreOffice Calc 操作這個函數的功能，無法以冒號一次框選整個範圍的儲存格，因此必需分別輸入每個儲存格的編號，再以逗點隔開)

=CONCATENATE(輸入區儲存格範圍)

fx | =concatenate(E5:N5)

11. 最後的完成畫面範例如下圖。視學生操作熟練度調整講解與操作進度，有時候一節課只能完成一個函數的操作，這是正常的。



(註：這個檔案也包含於本教材提供的壓縮包內)

第五章 搜尋演算法

教學時數：4 節

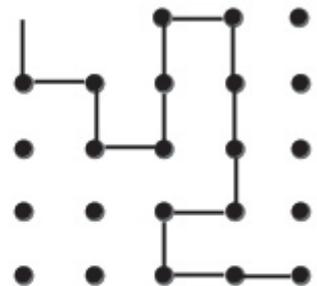
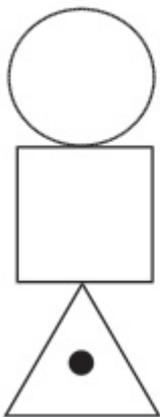
事前準備：

1. 不插電：列印附錄 5-1 的海戰棋表格，一張 A4 包含兩個一模一樣的盤面，可以進行兩回合的遊戲。
2. 插電：附錄程式檔，資料夾名稱為「html5_battleships」。

活動流程一 — 何謂演算法

演算法是一套事先被設計好的步驟，無論什麼人、什麼時間、在什麼地點，只要照著這套步驟來執行，都能得到相同的結果。像是銀行的 ATM 提款機，從插卡開始，輸入密碼、選擇提款功能、輸入金額的步驟都是事先被設計好的，同樣輸入提款一萬元，如果不同的人來操作會拿到不同的金額，那就是一場災難了！對於人類來說，設計一套步驟也許不難，但是執行時可就不是這麼容易了，透過以下活動來演示這個概念。

1. 找一個自願的同學到台前，展示一張圖片給他看，請他先思考一下之後，請全班同學按照他的指令將這張圖畫出來。畫完之後你會發現底下的同學畫出來的圖案可能會是千奇百怪的。
2. 進行討論，剛剛負責提供指令的同學，說的話是否不夠精確？有什麼地方可以再修正的？為什麼我們總是無法像電腦一樣，每個人按照指令執行都能得到相同的結果？（以下是幾個圖形的範例，老師可以自行增加修改）



活動流程二 — 紙本海戰棋

1. 發下海戰棋的遊戲盤，每個同學一張。告訴他們接下來這個遊戲要分組進行，請他們兩人為一組，找好對手之後搬動椅子坐在桌子的對側。(若班級人數為奇數，可以讓其中一組有三個人，其中一人擔任裁判，監督遊戲進行是否順利或違反規則)
2. 雙方輪流擔任攻擊與防守方，目的是率先擊破對方艦隊的所有船隻。擔任攻擊方時，將過程記錄在右手邊的「炮擊盤」，擔任防守方時，將過程記錄在左手邊的「佈局盤」。所以無輪現在自己輪到攻擊或是防守，都必須有記錄的動作。

玩家 1 (佈局盤)

	A	B	C	D	E	F
1						
2						
3						
4						
5						
6						

玩家 1 (炮擊盤)

	A	B	C	D	E	F
1						
2						
3						
4						
5						
6						

3. 「佈局階段」：遊戲開始前，雙方在版面上放置共計四艘的船艦，分別是佔五格的航空母艦（以字母 A 表示）、佔四格的戰艦（以字母 B 表示）、佔三格的巡洋艦（以字母 C 表示）、以及佔兩格的巡邏艇（以字母 D 表示）。請同學按照自己的想法，將這些船放置在佈局盤中。(記得別被對面的人看到了) 假設玩家 1、2 佈局的結果如下：

玩家 1 (佈局盤)

	A	B	C	D	E	F
1	A				D	D
2	A					
3	A		B	B	B	B
4	A		C			
5	A		C			
6			C			

玩家 2 (佈局盤)

	A	B	C	D	E	F
1		A	A	A	A	A
2		B				
3		B				
4		B		C	C	C
5		B				D
6						D

4. 「炮擊階段」：雙方玩家輪流進行炮擊，炮擊時喊出一個位址作為炮擊點。
 若**防守方**棋盤上所對應的位置**放有船艦**，則需**說出被擊中的船艦代號**並在佈局盤上**被炮擊的位置打 X**。**攻擊方**則根據剛剛宣告的位置，在炮擊盤上**記錄擊中的代號**；

若**防守方**棋盤上所對應的位置**沒有船艦**，則說「**M I S S !**」，並在佈局盤上**被炮擊的位置打 O**。**攻擊方**則根據剛剛宣告的位置，在炮擊盤上**記錄為「X」**。

5. 若其中一方的佈局盤上，所有的英文字母都劃上 X，代表他的所有船艦皆被擊沉，則遊戲結束。以下為可能的遊戲進行過程範例：

【回合一 – 玩家 1 炮擊位置 3 B，結果為 M I S S，進行後棋盤記錄如下】

玩家 1 (佈局盤)

	A	B	C	D	E	F
1	A				D	D
2	A					
3	A		B	B	B	B
4	A		C			
5	A		C			
6			C			

玩家 2 (佈局盤)

	A	B	C	D	E	F
1		A	A	A	A	A
2	B					
3	B	O				
4	B		C	C	C	
5	B					D
6						D

玩家 1 (炮擊盤)

	A	B	C	D	E	F
1						
2						
3		X				
4						
5						
6						

玩家 2 (炮擊盤)

	A	B	C	D	E	F
1						
2						
3						
4						
5						
6						

【回合二 - 玩家 2 炮擊位置 4 C，結果擊中船艦代號 C，進行後棋盤記錄如下】

玩家 1 (佈局盤)

	A	B	C	D	E	F
1	A				D	D
2	A					
3	A		B	B	B	B
4	A		✘			
5	A		C			
6			C			

玩家 2 (佈局盤)

	A	B	C	D	E	F
1		A	A	A	A	A
2	B					
3	B	○				
4	B		C	C	C	
5	B					D
6						D

玩家 1 (炮擊盤)

	A	B	C	D	E	F
1						
2						
3		✘				
4						
5						
6						

玩家 2 (炮擊盤)

	A	B	C	D	E	F
1						
2						
3						
4			C			
5						
6						

【回合三 - 玩家 1 炮擊位置 1 D，結果擊中船艦代號 A，進行後棋盤記錄如下】

玩家 1 (佈局盤)

	A	B	C	D	E	F
1	A				D	D
2	A					
3	A		B	B	B	B
4	A		✖			
5	A		C			
6			C			

玩家 2 (佈局盤)

	A	B	C	D	E	F
1		A	A	✖	A	A
2	B					
3	B	○				
4	B		C	C	C	
5	B					D
6						D

玩家 1 (炮擊盤)

	A	B	C	D	E	F
1				A		
2						
3		✖				
4						
5						
6						

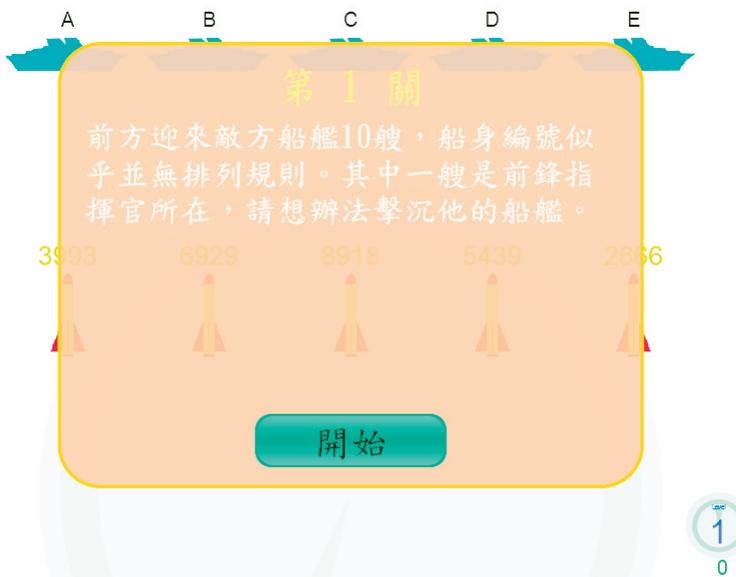
玩家 2 (炮擊盤)

	A	B	C	D	E	F
1						
2						
3						
4			C			
5						
6						

- 先讓同學進行幾分鐘的遊戲，接著老師要喊出「暫停」，向同學提示遊戲的進行訣竅。首先遊戲的關鍵在於一開始的佈局動作，較好的策略應該是讓船艦不依特定規則放置，儘量是分散、不集中於一處的。另外遊戲進入到幾個回合之後，因為有些大型船是被放在連續的格子之中，所以可以根據炮擊盤上的狀況，直接找出他接下來的位置。
- 在課堂進行一回合的遊戲，紙張下方可以讓學生在下課時間再找一位對手玩一次對戰，在上面寫下自己的姓名之後，將兩人的紙張裝訂在一次，作為紙本的回家作業。

活動流程三 — 程式海戰棋

1. 開啟附錄程式檔「html5_battleships」資料夾的 battleships.html
2. 與紙本海戰棋不同，這次的對手是電腦。電腦派出了艦隊意圖進行侵略，我們手上擁有可以攻擊船艦的飛彈。電腦的每艘船艦上都有一個編號，一個飛彈對應其中一艘。比起將所有船艦擊沉，我們的目的是針對電腦的指揮官所在的那艘船，只要成功擊中就算完成這次的攻擊回合。遊戲總共有六關，每關開始都有遊戲提示，請閱讀完提示之後按下開始，畫面右下角有一個計時器，看看同學需要多久的時間才能全部完成。(遊戲有炮擊音效，怕吵的老師記得請同學先將擴音喇叭關掉)



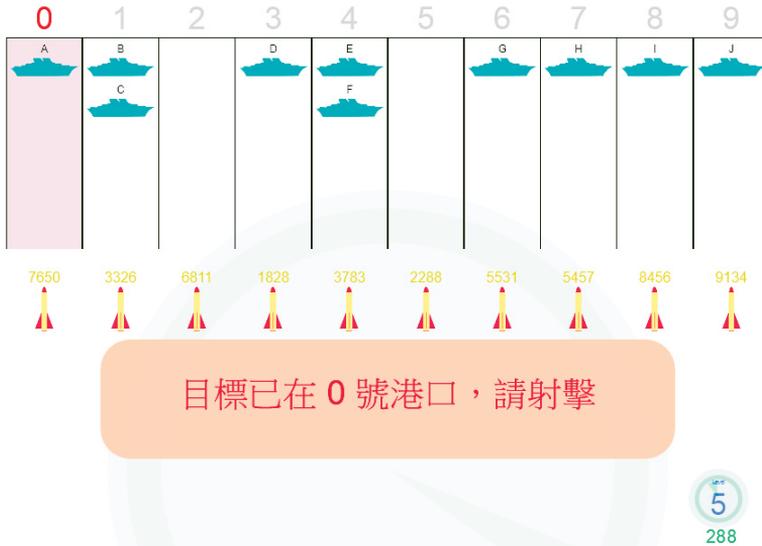
3. 為了不讓飛彈能連續發射，遊戲設計了一個飛彈冷卻的機制，必須在擊中提示逐漸淡出後才能再發射下一顆飛彈。老師先讓學生自己去嘗試與探索過關的方式與技巧，讓學生進行幾回合或是幾分鐘的遊戲之後再暫停活動，進行搜索策略的提示。



- 4. 遊戲的第 1、2 關，船隻排列的方式為不規則，我們能運用的策略就只能是亂猜了。運氣好的話幾次就猜中，運氣不好可能猜到最後一艘船才成功過關。
- 5. 遊戲的第 3、4 關，多了情報員的提示。由於船艦已經按照大小順序排列，每次攻擊失敗後都會提示目標在更小或更大的編號，如此就算遭遇 26 艘船，最多在 5 次攻擊內就能過關。



6. 遊戲的第 5、6 關，船艦按照特定規則停泊在港口之中，直接縮小了攻擊的範圍。只要有耐心按照規則計算編號，就能快速地鎖定攻擊目標。以下圖為例，目標停泊在 0 號港口，按照提示的規則，只要將飛彈上的四個號碼相加，看結果的個位數字就能知道對應的港口編號。因此我們要找的目標即為「2288」。



活動說明 — 搜尋演算法

依照程式海戰棋的不同關卡與策略，我們大致應用到幾個基本的演算法，對學生說明之後，老師可以安排一次測驗，以問答題的方式讓學生依自己的認知回答這幾種演算法的實行方式，不一定是背下所有的文字，也可以圖解或舉例說明：

1. 線性搜尋法 (linear search 或稱 sequential search) :
編號隨機排列，搜尋時除了憑空猜測之外，只要一個接一個走訪所有的位置，一定可以找到目標。
2. 二分搜尋法 (binary search) :
先將編號按照大小順序排列，每次猜測時都將目標範圍縮小一半，如此可快速鎖定範圍。
3. 雜湊搜尋法 (hashing search) :
編號已按照特定規則進行分組，只要知道分組的位置，就能馬上縮小搜尋的範圍。

二分搜尋法比線性搜尋法快，但線性搜尋法不需要照順序排列編號的時間。

雜湊搜尋法通常來說會比其他兩種快，但也有可能會變得非常慢。在最糟的情況下，如果所有船都同在一個船泊，那就會變得和線性搜尋法一樣慢了。

電腦儲存了大量資訊，它們必須要能夠快速對其進行搜尋篩選。而搜尋引擎所面對的世界上最大搜尋問題之一，就是他們必須要在極短時間內搜尋數十億個網頁。

而電腦被要求尋找的資料，像是文字、條碼編號或是作者名稱等，則被稱為「搜尋關鍵字」。

電腦可以非常快速的處理資訊，而你可能會認為為了找到想要的資訊，它們必須從頭開始，直到找到想找的目標為止。這是我們在前兩關進行「線性搜尋」中所使用的方法。但這種方法即使對電腦而言也過於緩慢了。

舉個例子，假設一間超級市場的貨架上有一萬種不同產品，當櫃檯掃描一個條碼時，電腦必須最多搜尋到第一萬次才能找到產品的名稱與價格。這樣即使檢查每筆條碼只花千分之一秒，檢查完所有條碼也要花 10 秒。想像一下，要花多少時間才能掃描完一個家庭會採買的林林總總的雜貨。

二分搜尋法看起來是比較好的方式。二分搜尋法中，先將每筆數字照順序排列，然後檢查列表中間的項目，就可以知道要找的關鍵字是在列表前半還是後半。接著重複該動作，直到找到要搜尋的項目為止。舉剛才超級市場的例子，一萬個項目最多需要進行十四次檢查，也就是大約兩百分之一秒，幾乎是一瞬間即可完成搜尋。

第三個尋找資料的方法叫做雜湊。在這種方法中，關鍵字本身會直接指示到哪裡找到這個資訊。舉個例子，假設要搜尋電話號碼，你可以將所有數字加起來，然後除以 11 取餘數。這個做法得到的雜湊關鍵字有點類似先前在錯誤檢查、修正的單元中所提到的同位檢查數字。用這種方法，通常電腦可以很直接地找到要搜尋的資訊。不過還是有小部份的可能，在同一個位置有好幾筆資料，而電腦必須在這幾筆裡面再繼續搜尋。

附註 — 程式海戰棋的自訂項目

程式之中的遊戲提示、船身編號位數、擊中提示訊自、圖片與音效都是可以修改的，以記事本軟體開啟「battleships_conf.js」檔案，修改完成之後，再開啟「cache.appcache」，修改「Updated on:」後方的時間，就能讓修改部份生效。

附錄 5-1 — 海戰棋

姓名：_____

【佈局盤】

	A	B	C	D	E	F
1						
2						
3						
4						
5						
6						

佈局代號：

航空母艦 - 五格A 戰艦 - 四格B
巡洋艦 - 三格C 巡邏艇 - 兩格D

【防守階段】

對方炮擊成功：被炮擊位置劃X
對方炮擊失敗：被炮擊位置劃○

【炮擊盤】

	A	B	C	D	E	F
1						
2						
3						
4						
5						
6						

【攻擊階段】

成功擊中船艦：在炮擊位置寫下該船艦的代號
炮擊失敗：在炮擊位置劃X

姓名：_____

【佈局盤】

	A	B	C	D	E	F
1						
2						
3						
4						
5						
6						

佈局代號：

航空母艦 - 五格A 戰艦 - 四格B
巡洋艦 - 三格C 巡邏艇 - 兩格D

【防守階段】

對方炮擊成功：被炮擊位置劃X
對方炮擊失敗：被炮擊位置劃○

【炮擊盤】

	A	B	C	D	E	F
1						
2						
3						
4						
5						
6						

【攻擊階段】

成功擊中船艦：在炮擊位置寫下該船艦的代號
炮擊失敗：在炮擊位置劃X

第六章 排序演算法

教學時數：4 節

事前準備：

1. 簽字筆一支，不透明的紙杯數個，以及對應大小的杯蓋，有刻度的水壺數個。
2. 插電：附錄程式檔，資料夾名稱為「html5_sort_scale」。

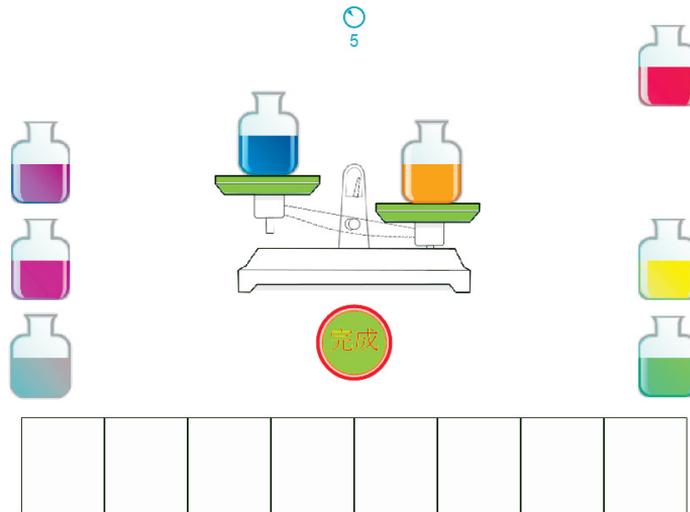
活動流程一 — 杯子的排序遊戲

1. 展示老師準備好的紙杯，問問同學這樣一杯的容量大約是幾 c.c.。接著在講桌上排列六到八個杯子，在第一個杯子裝滿溫開水，詢問同學接下來若要依序遞減杯子裡面的水，每次應該減去多少，才能讓每個杯子都有裝水，而且能感受到重量的差異？
2. 確認好每杯應該裝的水量之後，請同學協助在杯中倒水，倒完之後蓋上杯蓋，然後在杯底以簽字筆寫下每杯的水量。
3. 隨意將所有杯子排列成橫排。找一個自願者到前方，請他想辦法將前方的杯子按照輕重的次序排列，提醒他不能夠直接把杯子拿起來偷看答案。接著替他計時，看看需要多久才能完成，或是能不能排出正確的次序。
4. 為了增加趣味性，按班級人數請學生平均分成幾組，每組要先討論排水杯的策略，如何才能又快又正確。接著推派一個代表，以及一個待會要接受遊戲懲罰的同學。
5. 接著請各組的代表依次上台進行排水杯的動作，老師替這些同學計時並記錄時間與正確與否。最後依照完成的時間，最少者為勝方，其餘組別列為負方。若是排序結果是錯誤的，則一律視為負方。
6. 請負方組別的代表到前方，將事先設定好容量的溫開水喝掉。(為了身體健康，一律盛裝溫開水，容量在 500 c.c. 之內為宜，若遊戲可能進行好幾回合，也不要讓同一人一直擔任喝水的角色)

活動流程二 — 天平的排序遊戲

1. 開啟附錄程式檔「html5_sort_scale」資料夾中的 sort_scale.html
2. 畫面正中央有一個天平，旁邊八個瓶子分別設定了不同的重量，利用天平為工具，將瓶子排列到下方的空格中，按下完成即可知道排序的結果。請同學探索，按照自己的策略想辦法得到正確答案。為了達成探索的目的，不妨開放同學自由進行討論。

3. 遊戲同樣有計時的機制，老師也可以在探索活動之後分組進行，請同學分組進行競賽。



活動流程二 — 天平的排序遊戲

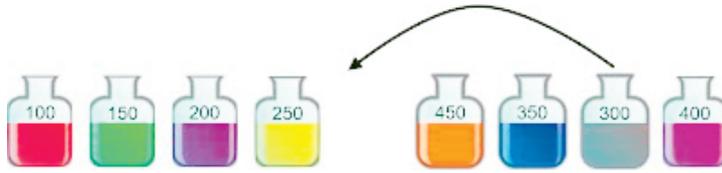
1. 電腦時常被使用來把資料依序排列。舉例來說，把名字依字母順序排列、依日期排序電子郵件或行程，或是依數量多寡排列物品等等。排序除了可使得我們在找東西時更快速之外，還有許多顯著的好處。例如把全班的成績依高低排序，最低分與最高分就很明顯了。
但若使用錯誤的方法，即使有設備很好、很快速的電腦，將大量的資料正確排序可能還是會花很多時間。還好，有幾種快速的演算法非常適合使用於排序。
2. 【討論問題】
 - i. 什麼樣的資料會需要排序？
 - ii. 為什麼排序很重要？
 - iii. 如果資料沒有排序會怎麼樣？

對於順序的排列，如果有更好的方法，可以有效地減少完成所需的時間，以下介紹幾種不同的排序演算法，在每學會一種演算法之後，請同學按這個演算法重新操作一次程式。

選擇排序法 (Selection sort)

先找到最輕的容器並擺在一邊。接下來，再從剩下的容器中挑出最輕的，並也擺到一邊。重復此動作直到所有的容器都被擺到同一邊。

在還沒有進行提示之前，先讓學生自行探索，試著用自己的模式排出正確順序來。操作幾分鐘之後暫停學生的動作，詢問剛剛他們使用的策略是什麼，大部份的學生一開始採取的方式都會是選擇排序法。講解完選擇排序的原理，讓學生在已經知道策略為何的情況下再操作一次看看。



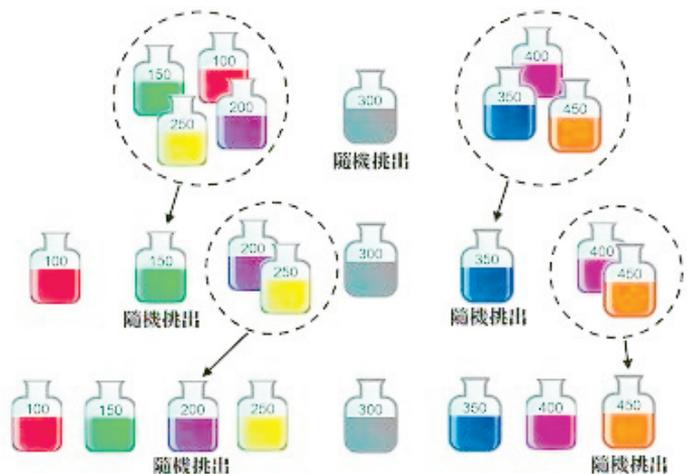
快速排序法 (Quick sort)

從全部的物件中隨機挑出一個物件，把它放在天平的一端。然後將它與其他剩餘的物件一一比較，將較輕的放在左側，剛才選出的物件放在中間，較重的則分放在右側（有可能在結束比較時，兩邊物件的數量不一樣）。

選擇其中一側，將該側所有物件重複上面的步驟。再將另一側的物件也做一樣的處理（記得一開始選擇的物件要保持在中間）。

剩餘的物件群一直重複這些步驟，直到每一群都只剩一個物件。一旦所有物件群都被分到只剩一個物件，所有物件便會被分成由最輕到最重。

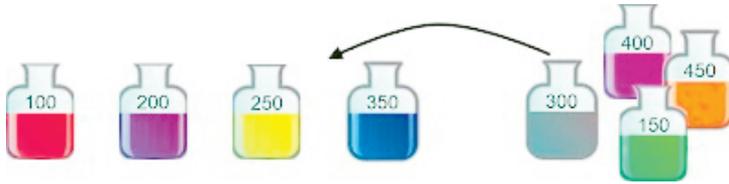
單純使用輔助程式進行快速排序法，其實並不方便，必須同時使用紙筆協助記錄分組的動作。教師先透過白板講解快速排序的原理，一邊透過程式的協助找出瓶子的輕重，一邊記錄分組的過程。講解完之後請學生依照剛剛的示範進行排序的操作。



插入排序法 (Insertion sort)

從尚未排序的物件群中挑出一個物件，把它插入已排序物件群中正確的位置。隨著一次次的動作，未排序物件群的規模會越來越小，而已排序物件群的規模則會越來越大，直到所有物件都排序完成。玩撲克牌時就常常利用這種方式來理牌。

要操作插入排序法的過程，必須事先就知道所有瓶子的重量，因此這部份不適合使用程式進行排序，教師只需要講解原理即可。



氣泡排序法 (Bubble sort)

一次又一次從頭到尾檢查整個物件群，如果兩個相鄰的物件順序不對，就把它們交換，整個物件群檢查完之後，再重頭開始檢查一次。如果沒有任何物件在檢查名單時被交換，就表示排序完成了。這個方法並不是很有效率，但是對某些人來說這種方法較容易理解。

以不插電的方式進行的排水杯遊戲，大部份學生使用的方式即是氣泡排序法。講解完氣泡排序的原理，由教師先進行操作的示範。先將所有瓶子隨意放到畫面下方的空格處，再開始進行兩個兩個的重量比較。示範之後再請學生按此模式進行操作練習。



合併排序法 (Merge sort)

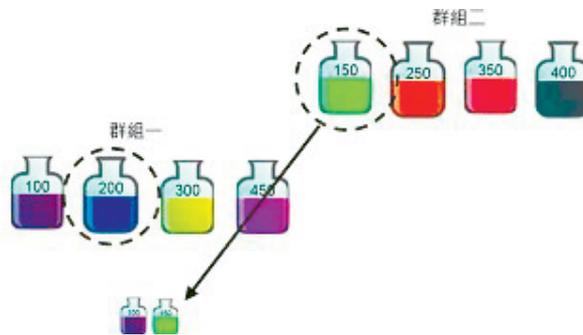
首先，將名單隨機分成大小相同的兩個群組（奇數個時則分成大小相近的兩份）。兩個群組都要分別做排序，最後再將兩個群組合併。

合併群組的方法：

比較兩個群組最前面的物件，把較小的那個挑出來放入排序群組中。在下圖中，兩個群組中最輕的物件分別是 100 和 150 公克，所以下先把 100 公克的物件挑出來放入下方的第一個位置。

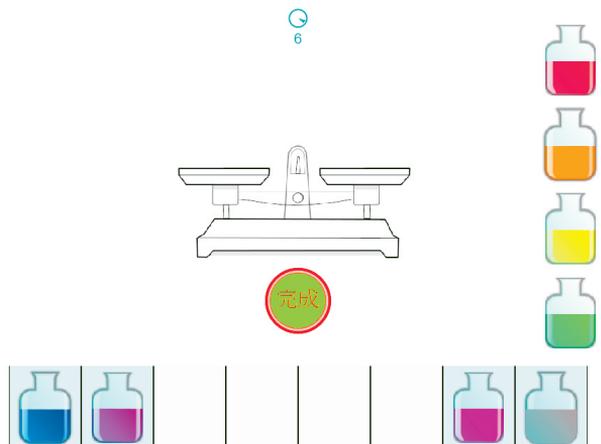


第二回合比較兩個群組中最輕的物件，這次是 200 和 150 公克，這次則是把 150 公克挑出來，放入下方的第二個位置。

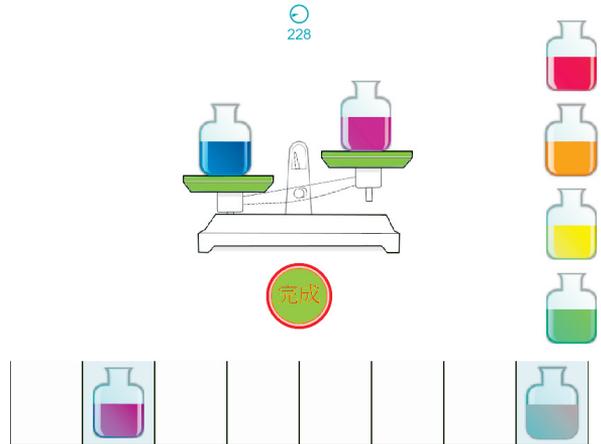


最開始分成兩個群組（各四個瓶子）以後的排序，同樣採用合併排序法處理。教師可以示範四個瓶子如何透過合併排序法做排序，將四個瓶子分成兩組（各兩瓶），放在畫面下方的空格內，先分別比較出兩組之中瓶子的輕重順序，再進行合併。過程如下所示：

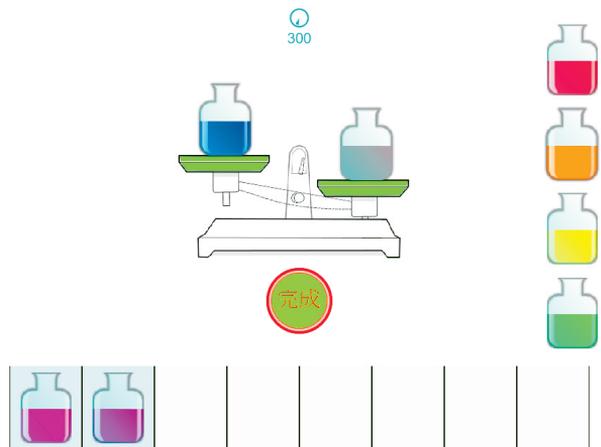
1. 先將畫面左方四個瓶子任意分成兩組，並針對兩組中的瓶子進行排序。按圖所示左手邊是藍色瓶比紫色瓶輕、右手邊的粉紅色瓶比灰色瓶輕。



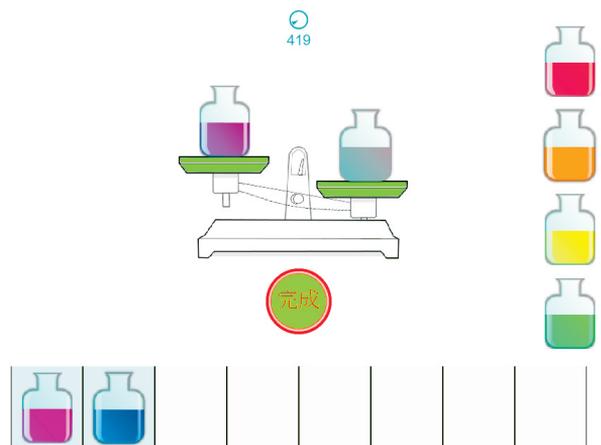
2. 從兩組中拿出最輕的兩個瓶子進行比較（藍色與粉紅色），因為粉紅色瓶的重量較輕，因此把它排在最左邊的位置。



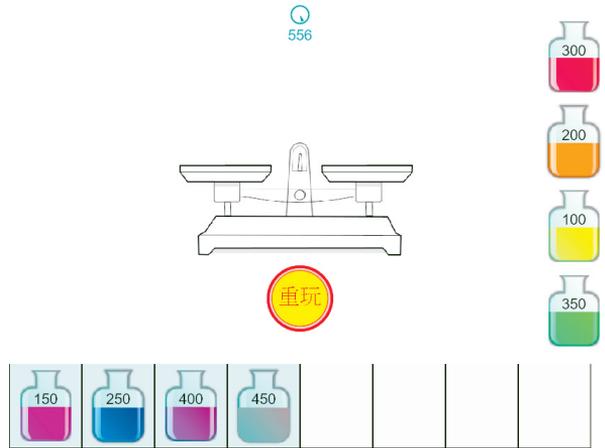
3. 藍色瓶留在天平上，與右手邊這一組剩下的灰色瓶比較重量。因為藍色瓶比較輕，因此把它排在粉紅色的右邊。



4. 最後一回合，比較紫色與灰色瓶的重量。因為紫色較輕，所以重量順序就確定為「粉紅、藍、紫、灰」了，如此則完成四個瓶子的合併排序過程。



5. 如果要針對右手邊四個瓶子進行排序，跟剛剛的操作方式是一樣的。兩組四個瓶子都進行完排序，再依次合併即可操作完成。



快速排序與合併排序都屬於分治法 (Divide and Conquer) 的一種操作方式，這是目前最好的方法之一。

在已排序的清單中尋找資訊會容易很多。電話簿、字典和書籍索引等都是用字母來排序。試想如果它們沒有經過這樣的排序，我們要找相關資訊時會有多不方便。若是將一串數字（例如一些支出）由小到大進行排列，就可以輕易的在排序的兩端找到最大和最小的數字，如果有數字重複的話那也是一目了然，因為它們一定會被排在一起。

實際上電腦在執行時花了很多的時間在把資料排序。所以，找到快速又有效率的排序方法一直是資訊科學家的重要工作之一。一些比較慢的方法如插入排序法、選擇排序法和泡沫排序法、合併排序法這些速度較快的排序法。舉例來說：如果有十萬筆資料要排序，快速排序法大概比選擇排序法要快上 2000 倍。若資料總數上升至一百萬筆，那麼兩種排序法所花的時間會差上 20000 倍。

電腦常常要同時處理上百萬筆資料（很多網站擁有上百萬的訪客量，甚至連一張從便宜的照相機所拍的照片也有超過百萬的像素要處理）。選擇不同的演算法的差異可能是要花費一秒鐘還是五小時來完成一個完全相同的任務。不僅僅是時間上的延遲令人無法忍受，還有其他的成本，比方說要耗上 20000 倍的電力（如此浪費電力不僅有違環保理念，而且也會減少可攜式裝置的電池壽命）。所以，選擇合適的演算法才能有好的效果。

快速排序法使用到一種方法叫做分治法。在快速排序法中，我們不斷的把物件群組分成較小的兩個群組，然後再對這些較小的部份執行快速排序法。最後，

整個物件群組被反覆的分割，直到它們小到可以被簡易的排序。在快速排序法中，物件群被分割到只剩下一個物件，那還需要排序嗎？儘管這些步驟看似十分複雜，但是在實際運行中，快速排序法卻戲劇性地比其他方法快速。分治法所使用的這種觀念稱為遞迴（Recursion），就是演算法裡會反覆呼叫自己來解決問題。這聽起來很奇怪，但實作起來卻意外的管用。

延伸活動 — 試算表軟體中，排序與分析的應用

在一般試算表中的教學與應用中，資料排序與分析是最常被提到的部份。老師可按照教學現場的情況，利用成績單展示排序的效果，或是導入以函數成績單設計的教學，讓同學知道如何計算成績的總分、平均、排序或排名是如何被達成的。

第七章 文字壓縮

教學時數：3 節

事前準備：

1. 不插電：附錄 7-1，文字壓縮的紙本挑戰題目。
2. 插電：附錄程式檔，資料夾名稱為「html5_text_compression」。

活動流程一 — 文字壓縮的概念

因為電腦可以儲存的資訊空間是有限的，所以要盡可能用有效率的方法來呈現資訊，這個概念就叫做「壓縮」。

藉由對資料做編碼，然後要取出資料時再解碼的方式，電腦可以儲存更多資料，或是更快地在網路傳遞資料。

1. 開啟資料夾「html5_text_compression」的 text_compression.html
2. 操作步驟：
 - i. 按下「開始選取保留目標」，開始找到文章中重覆出現的單字。以滑鼠點擊或是拖曳選取其中一個單字（或是一整個句子），我們稱為保留標的。選取後按下「保留標的選取完成」。（選取保留標的時，不能同時跨越好幾個句子，程式會顯示錯誤訊息）

保留標的選取完成

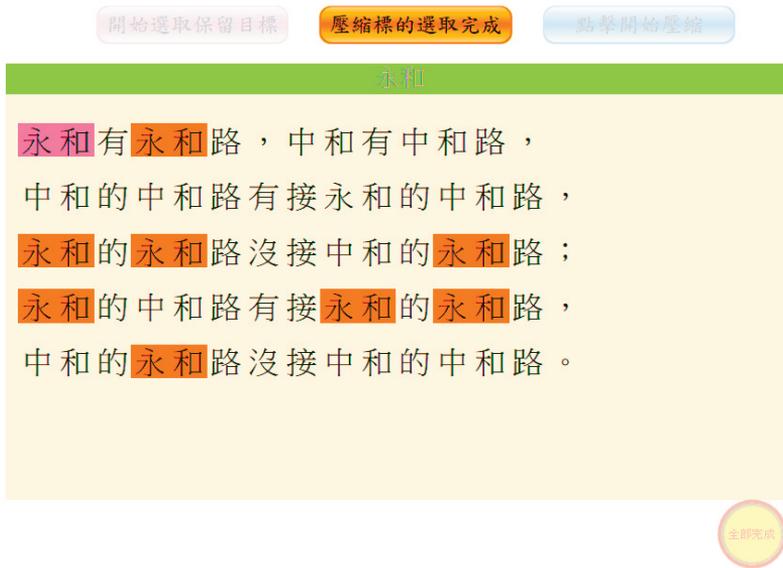
開始選取壓縮標的

點擊開始壓縮

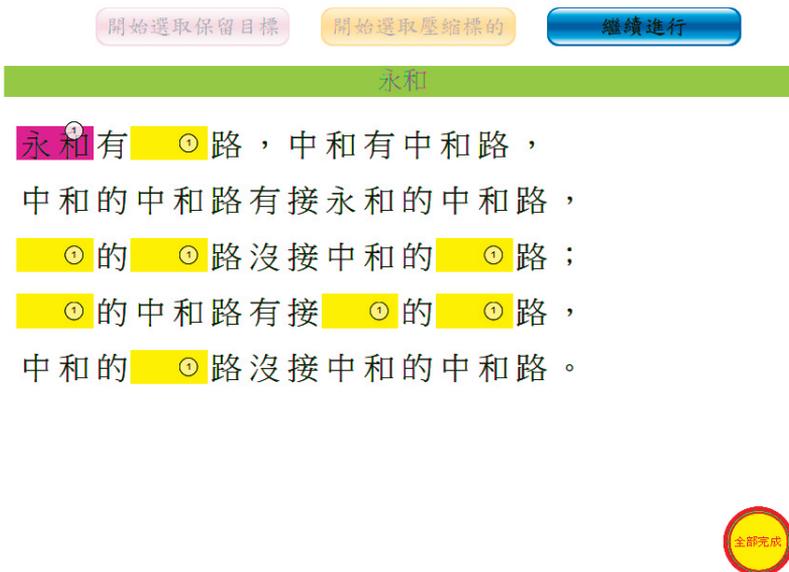
永和 有永和路，中和有中和路，
中和的中和路有接永和的中和路，
永和的永和路沒接中和的永和路；
永和的中和路有接永和的永和路，
中和的永和路沒接中和的中和路。



- ii. 按下「開始選取壓縮標的」按鈕，這次是要選取其他重覆出現的單字（或是一整個句子），我們稱為壓縮標的。找到所有與保留標的相同的地方之後，按下「壓縮標的選取完成」。



- iii. 最後按下「開始壓縮」按鈕，程式會將所有重覆的單字（或句子）清除，只留下第一步驟的保留標的，上方會出現一個編號。壓縮標的消失後則用數字標示它所對應的標的位置。



- iv. 按下「繼續進行」重複以上步驟，一直進行到無法再找到保留或壓縮標的為止。按下「全部完成」，看看總共用了幾個壓縮的步驟、成功壓縮了幾個字元。（註：一個中文字要算一個或兩個字元？這個深入的問題在課堂中暫不列入討論）

開始選取保留目標 開始選取壓縮標的 繼續進行

永和

永和¹有 路，中和有中和路，
中和的中和路有接永和的中和路，
的 路沒接中和的 路；
的中和路有接 的 路，
中和的 路沒接中和的中和路。

全部完成

- v. 同學在進行壓縮步驟時，可能採取不同的策略。把所有的「永和」一次壓縮起來，與把「永和」及「永和路」視為不同的保留標的，哪一種的方式可以壓縮更多的字元呢？請同學多方進行嘗試，並進行課堂的討論。
- vi. 程式內建提供四個關卡，包含英文與中文。按下瀏覽器的重新整理即可選擇不同的題目。

3. 進行以下問題討論：

- 一、為什麼選取標的時，要被限制在至少選取兩個字元以上？
- 二、尋找壓縮標的時，如果漏掉了其中幾個，對於結果有什麼影響？
- 三、要優先標記一整個句子，還是以較短的單字為優先？

結論與說明

電腦的儲存容量正以令人難以置信的速度增長。在過去 25 年裡，傳統電腦的儲存量已經至少增長了數百萬倍。雖然如此，我們打算存入電腦中的東西也跟著增加很多很多。只要空間足夠，電腦可以儲存整本書，甚至一整個圖書館的資料，當然現在的電腦儲存許多音樂與電影再就不是問題了。

大量的資料在網路網路上是個大問題，因為資料越大就需要更長的時間才能完

成下載的動作。此外，我們一直在嘗試將電腦越做越小（像是手機、手錶與許多的穿戴裝置），但是我們卻希望在上面儲存越多的資訊。

要解決儲存空間或傳輸大量資料的問題，除了買更多如硬碟、快閃磁碟之類的儲存空間，或是加快網路的傳輸速度之外，還有一個解決這個問題的方法，就是我們可以透過壓縮使得資料佔用的空間變少。壓縮和解壓縮的這個過程通常會交給電腦自動完成。我們會注意到的可能只有磁碟因此增加了更多的儲存空間，或是網頁的顯示速度因此變快了。

近年來，有許多壓縮的方法被發明了。在以上的活動中，我們操作的概念是用一個指標指向前面出現過的區塊的做法，被稱為 LZ 編碼（Ziv-Lempel Coding，或 LZ Coding）。這個方法可以方便地用於任何語言，並可以輕易地減少一半的資料量。它有時被稱為在個人電腦上的「拉鍊」（zip），並且也用在 GIF 和 PNG 這兩個圖檔格式。此外，也被用於高速的數據機。數據機利用壓縮減少透過電話線傳送的資料量，所以速度就快得多了。

活動流程二 — 紙本作業的挑戰

發下附錄 7-1 的英文文章給學生，提示同學至少有 1633 個字元可以被壓縮（每次壓縮至少要選取兩個字母），請他們以不同顏色的筆或是不同的標記方式，儘可能嘗試壓縮出更多的字。（如果學生不喜歡英文的題目，可以自行搜尋中文的長篇文章，儘量選擇出現重覆字詞的文章）

程式的自訂項目

程式中的題庫，放在 `text_compression_config.js` 檔案中，以記事本開啟

每一則題庫包括兩部份（欄位）：選單及短文

1. 中括號的左邊括號下第一行為選單的標題
2. `function()` 的下一行為短文內容

中括號的結束與下一個中括號之間，為了分開兩則不同的題目，所以中間必須加上一個逗號，在新增題目時別忘了。

因為版本範圍的限制，題目長度請做適度的調整。

附錄 7-1 紙本的壓縮挑戰題

Once upon a time, long, long ago, three little pigs set out to make their fortunes. The first little pig wasn't very clever, and decided to build his house out of straw, because it was cheap. The second little pig wasn't very clever either, and decided to build his house out of sticks, for the "natural" look that was so very much in fashion, even in those days. The third little pig was much smarter than his two brothers, and bought a load of bricks in a nearby town, with which to construct a sturdy but comfortable country home.

Not long after his housewarming party, the first little pig was curled up in a chair reading a book, when there came a knock at the door. It was the big bad wolf, naturally.

"Little pig, little pig, let me come in!" cried the wolf.

"Not by the hair on my chinny-chin-chin!" squealed the first little pig.

"Then I'll huff, and I'll puff, and I'll blow your house down!" roared the wolf, and he did huff, and he did puff, and the house soon collapsed. The first little pig ran as fast as he could to the house of sticks, and was soon safe inside. But it wasn't long before the wolf came calling again.

"Little pig, little pig, let me come in!" cried the wolf.

"Not by the hair on my chinny-chin-chin!" squealed the second little pig.

"Then I'll huff, and I'll puff, and I'll blow your house down!" roared the wolf, and he did huff, and he did puff, and the house was soon so much firewood. The two terrified little pigs ran all the way to their brother's brick house, but the wolf was hot on their heels, and soon he was on the doorstep.

"Little pig, little pig, let me come in!" cried the wolf.

"Not by the hair on my chinny-chin-chin!" squealed the third little pig.

"Then I'll huff, and I'll puff, and I'll blow your house down!" roared the wolf, and he huffed, and he puffed, and he huffed some more, but of course, the house was built of brick, and the wolf was soon out of breath. Then he had an idea. The chimney! He clambered up a handy oak tree onto the roof, only to find that there was no chimney, because the third little pig, being conscious of the environment, had installed electric heating. In his frustration, the wolf slipped and fell off the roof, breaking his left leg, and severely injuring his pride. As he limped away, the pigs laughed, and remarked how much more sensible it was to live in the city, where the only wolves were in the zoo. And so that is what they did, and of course they all lived happily ever after.

第八章 網路通訊

教學時數：3 節

事前準備：

1. 不插電：附錄 8-1，水果遊戲的卡片，一個小組六個人使用一份。在網路通訊協定的程式遊戲中，需要給發送者隨機的訊息，可以選擇使用小紙條，或是再封入信封之中，以增加保密性。
2. 插電：附錄程式檔，資料夾名稱為「html5_network_protocol」。

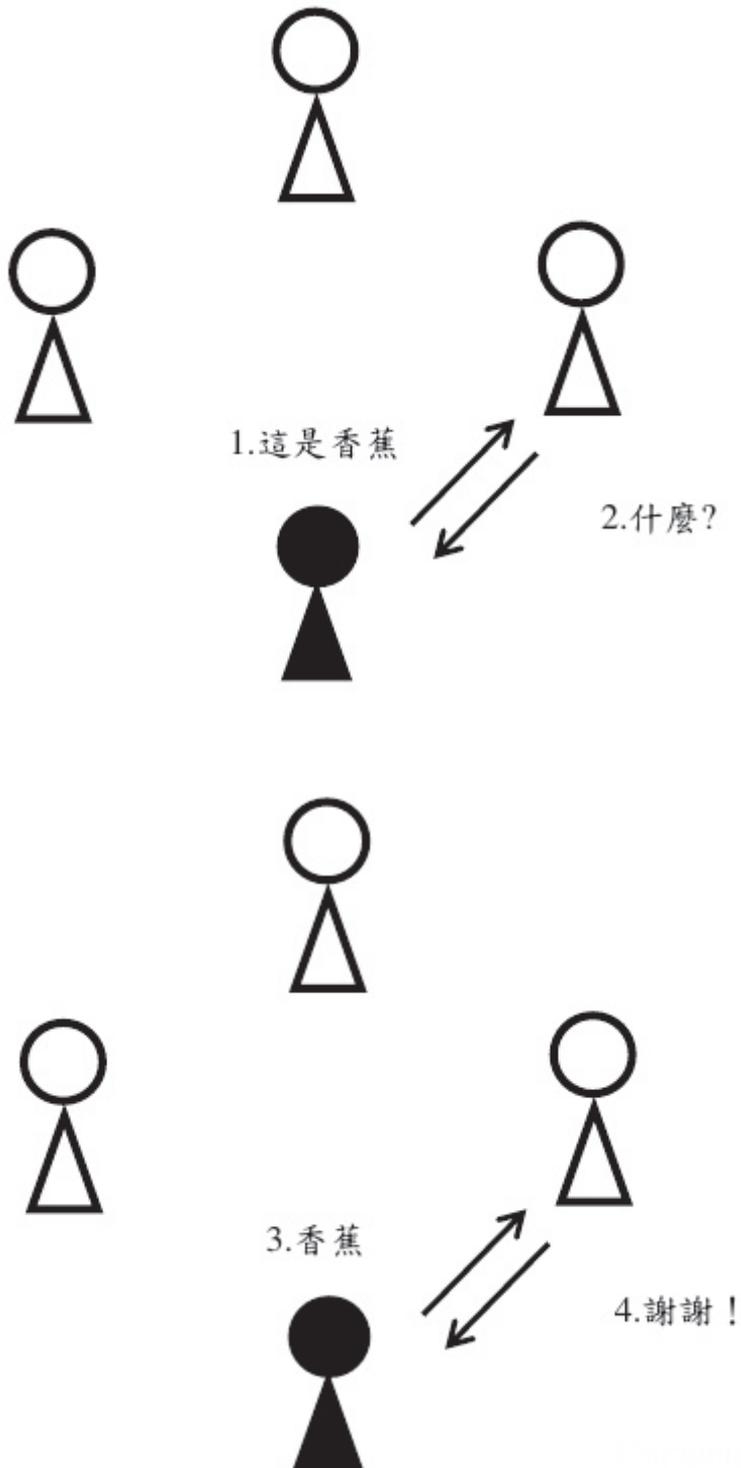
活動流程——水果遊戲，網路中的路由與死結

當有很多人同時使用同一個資源（例如很多車擠在道路上，或很多訊息要透過網路傳送），有可能會造成「死結」而影響過程的進行。因此，為了避免這種狀況發生，就需要良好的協同作業。進行完兩種遊戲後，與同學進行討論，如何才能讓過程更順利，哪個位置的人比較容易處理訊息，又哪個位置的人比較複雜卻最為關鍵？

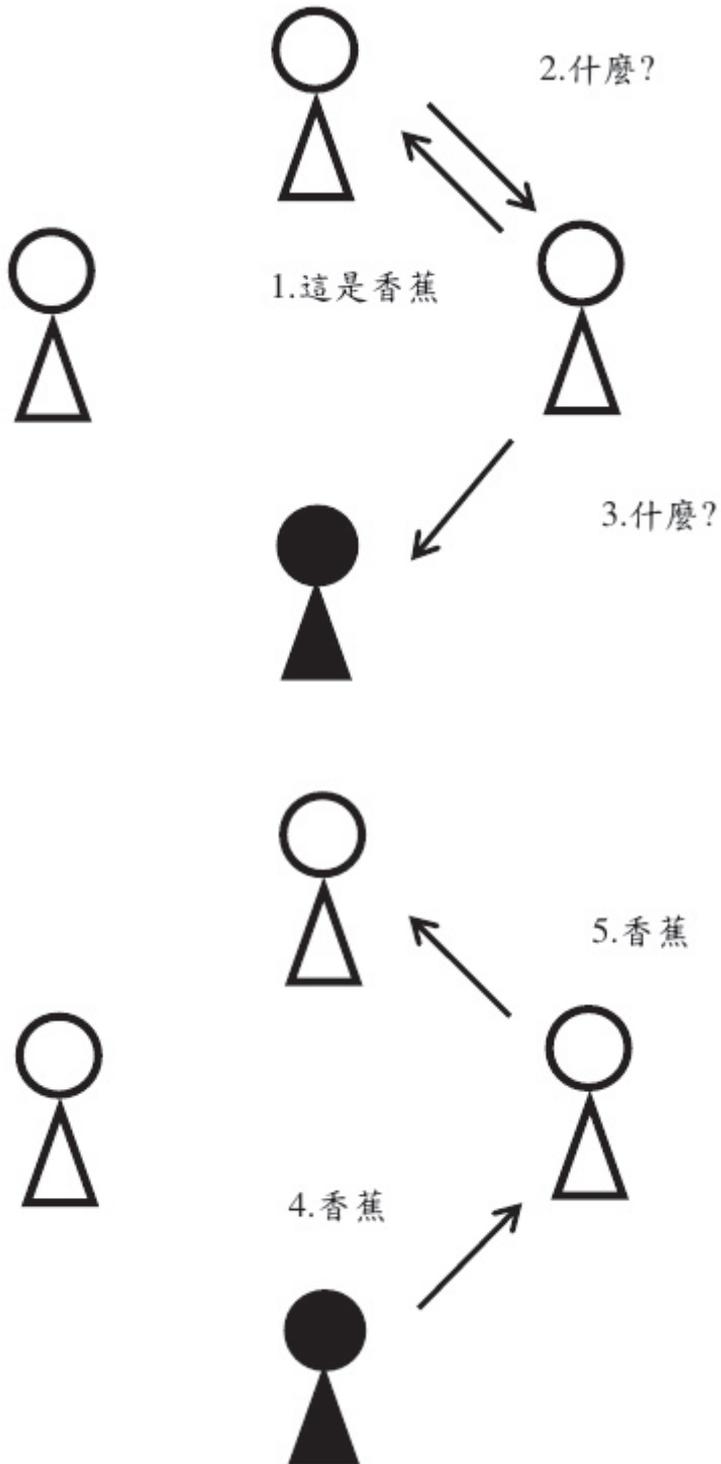
【單向水果遊戲】

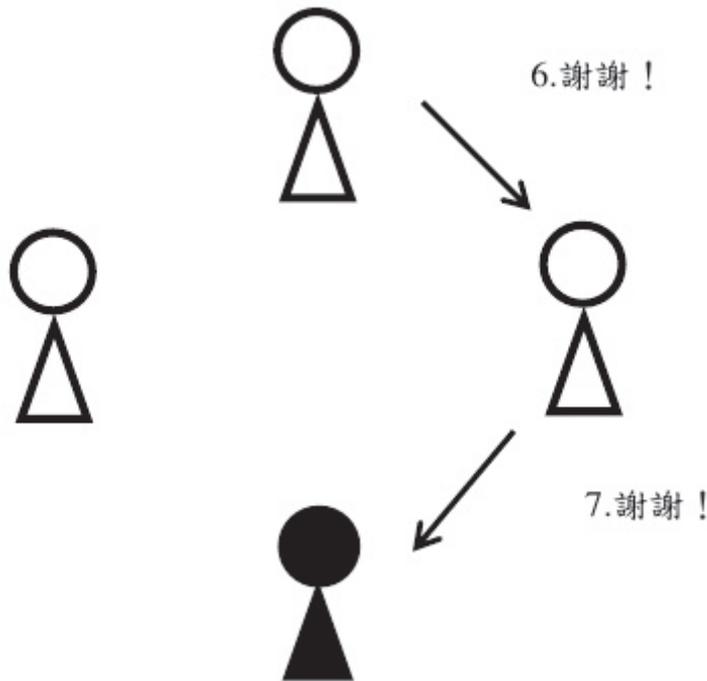
1. 請同學分組，每組大約 7 到 8 人，圍成一個圓圈就坐。
2. 指定其中一個同學為起點，並先設定好傳遞水果的方向為順時針或逆時針。然後請起點同學任意挑選一個水果的種類，假定被選中的是「香蕉」。
3. 起點同學朝著指定方向的第一位同學，說：「這是香蕉。」
第一位同學詢問起點：「什麼？」
起點回答：「香蕉。」
第二位回應起點：「謝謝！」，至此完成第一次的傳遞。
4. 第二位同學向第三位同學說：「這是香蕉。」
第三位詢問第二位：「什麼？」、第二位詢問起點：「什麼？」
起點回答：「香蕉。」 第二位回答第三位：「香蕉。」
第三位回應第二位：「謝謝！」、第二位回應起點：「謝謝！」，至此完成第二次的傳遞。
每一回合的「什麼」訊息都必需回到起點，「謝謝」訊息也是。
5. 依此類推，將水果傳遞回起點同學即算完成遊戲。中途如果有同學說錯指定的句子，或是因為不清楚規則而發生停頓，則必需回到起點重新開始。遊戲過程如下，以四人一組為例：

【第一回合】



【第二回合】

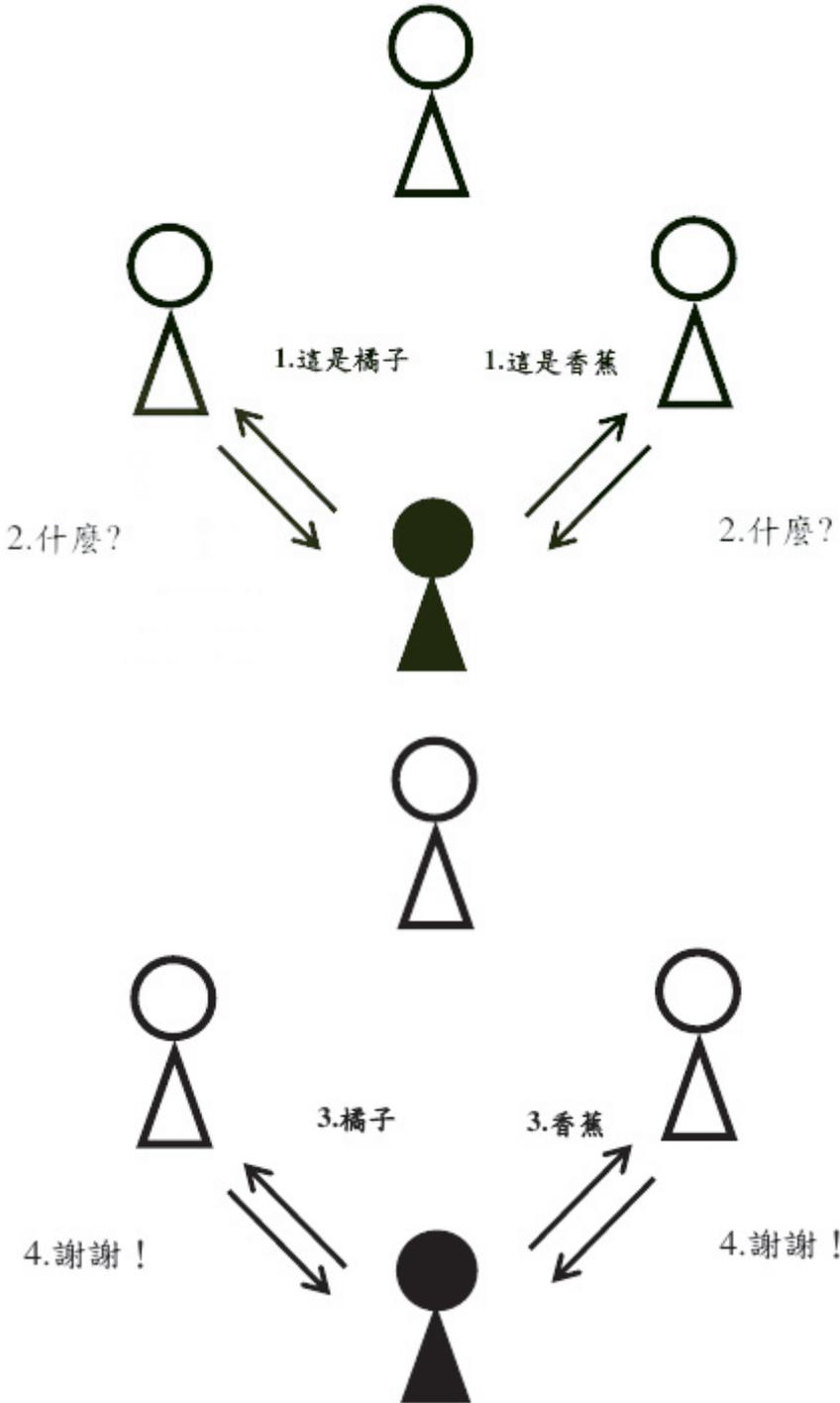




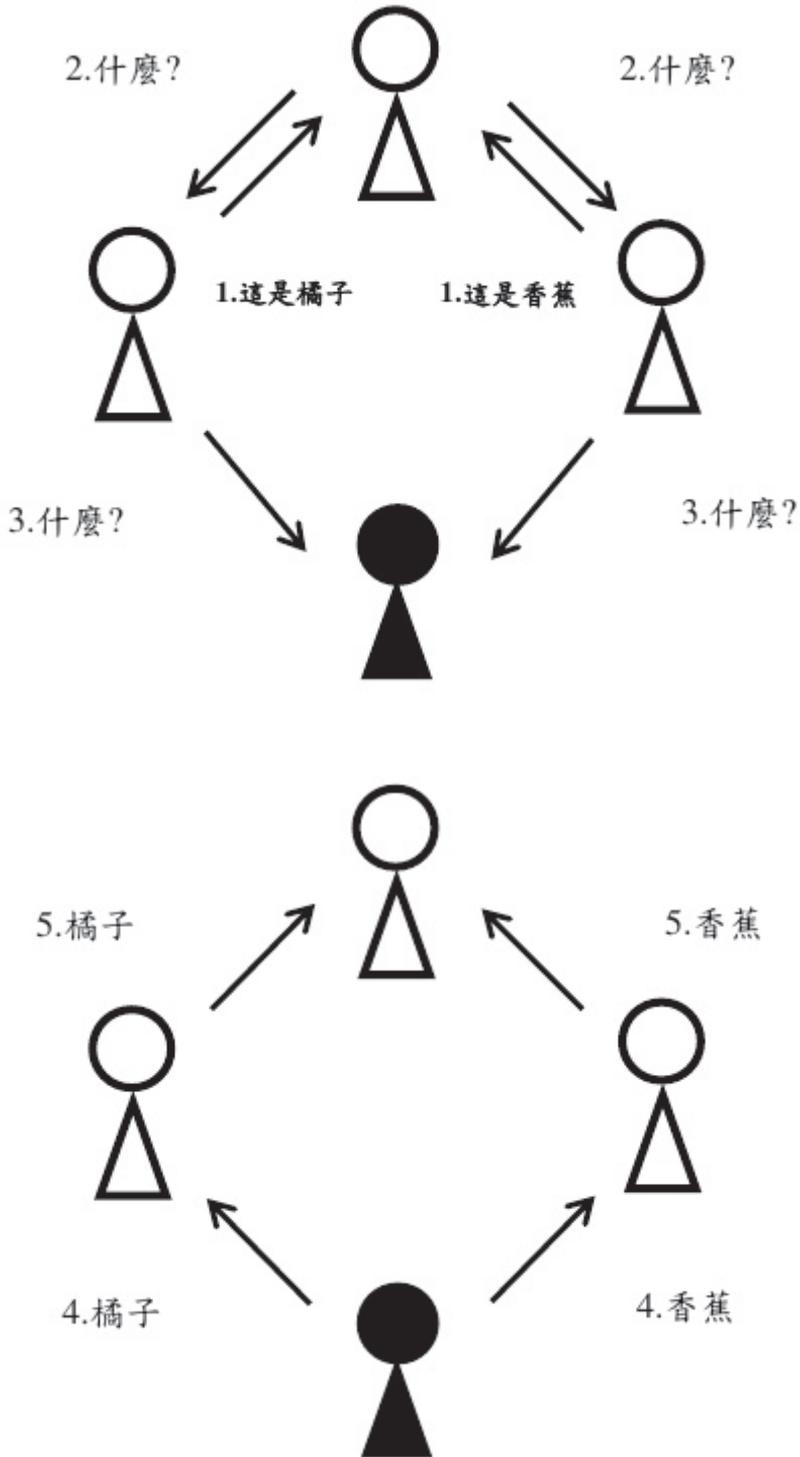
【雙向水果遊戲】

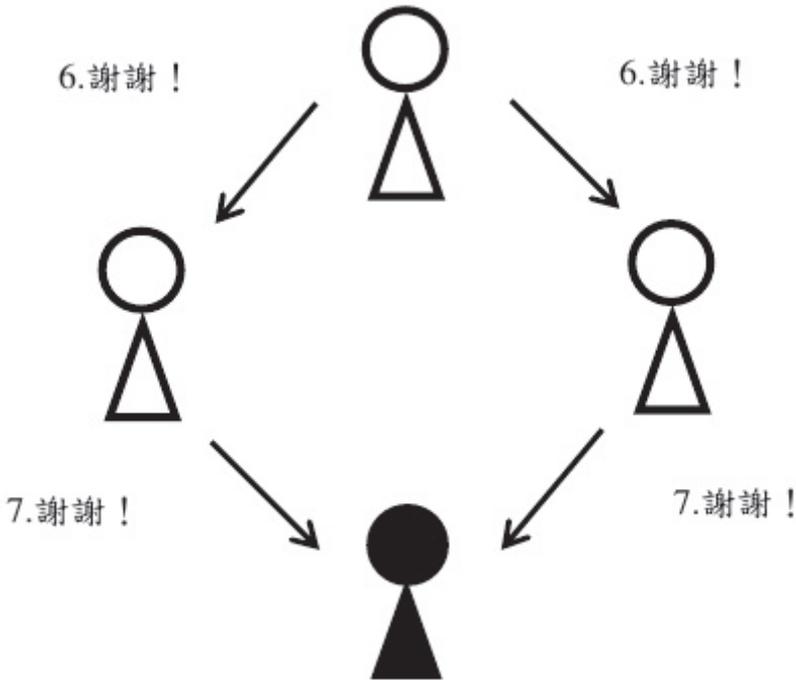
1. 同樣指定一人為起點。這次起點必須同時向左右邊傳遞各一種水果，假設向右手邊（接下來以 R1 簡稱）傳遞「香蕉」、向左手邊（接下來以 L1 簡稱）傳遞「橘子」，則 R1、L1 兩人要分別詢問起點「什麼？」，起點再分別向 R1 回應「香蕉」、L1 回應「橘子」，最後 R1、L1 兩個人再向起點說「謝謝！」，完成第一回合。
2. 第二回合，原本起點右手邊的人（R1）因為得到了「香蕉」訊息，所以接著向他右手邊的人（以下簡稱 R2）傳遞「香蕉」，接著 R2 詢問 R1「什麼？」、R1 詢問起點「什麼？」，起點回應 R1「香蕉」、R1 回應 R2「香蕉」，R2 再向 R1 說「謝謝！」、R1 向起點說「謝謝！」完成右手側的回合。在此同時，起點左手邊的 L1、L2 兩人也同時進行「橘子」的傳遞，也就是說起點在這一回合會同時接到來自右手與左手邊的詢問，以及右手與左手邊傳遞回來的「謝謝！」
3. 雙向的水果傳遞，最容易出錯的地方在會有人同時接到兩個方向傳遞過來的水果，在往前詢問「什麼？」與傳回「謝謝！」這兩個訊息的時候，會不小心搞錯方向。若是出現錯誤，同樣必須從起點開始，老師應適時介入提供協助。

4. 遊戲在兩個水果傳回起點處後即結束。以下為遊戲過程範例，同樣以四人為例：
【第一回合】



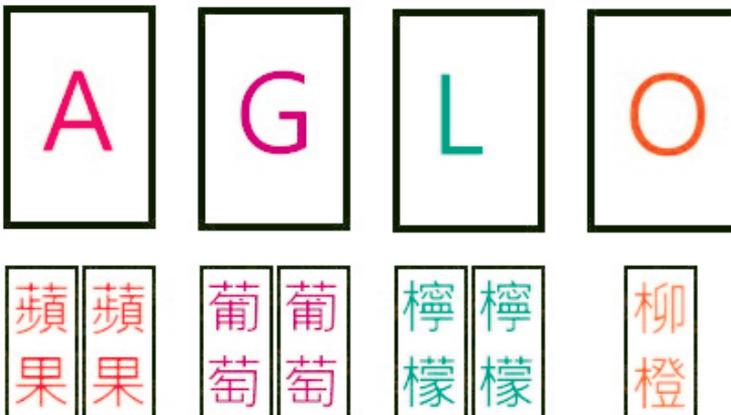
【第二回合】



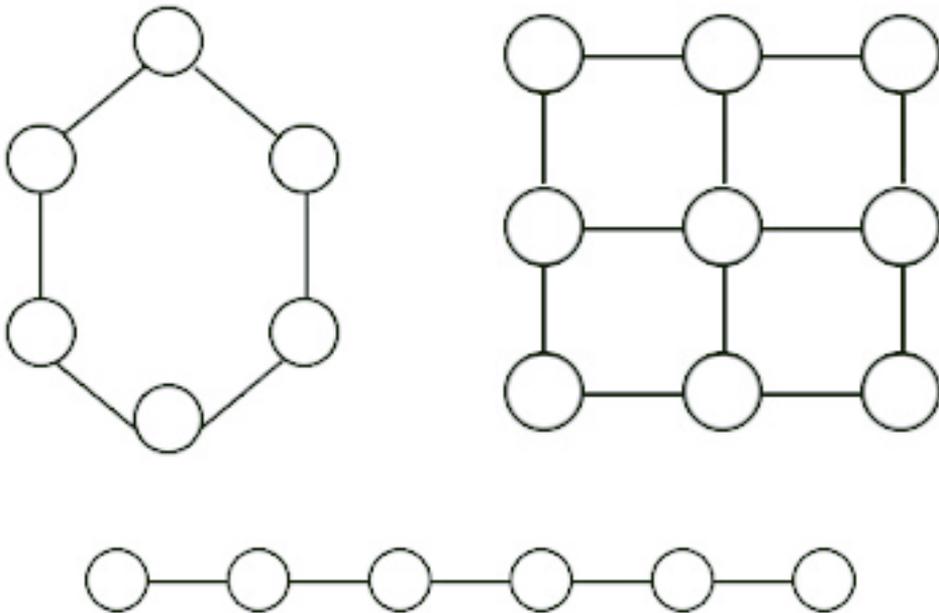


活動流程二 — 卡片式水果遊戲，網路中的路由與死結

1. 為同學進行分組，每組至少四人、最多六人。影印附錄 8-1 的卡片並裁切，每個人擁有同樣的中文水果牌兩張，以及對應水果牌第一個英文字母的標籤卡一張。（例如鳳梨對應英文字母 P、奇異果對應英文字母 K）
2. 讓同一組的同學坐成一個圓圈。
3. 將中文水果牌隨意洗牌並發下給同學，每人左右手各持一張，然後指定其中一人，讓他只能持有一張中文水果牌。



4. 開始遊戲前，每個人將自己的英文標籤卡放在前面。
5. 傳遞手上的中文水果牌必須按照以下兩條規則
 - i. 一隻手只能拿一張牌
 - ii. 只有在旁邊的人手上空著的時候才能傳遞中文水果牌。(傳遞手上的哪一張牌，可以自由選擇)
6. 遊戲目標是每個人都能拿到與自己標籤對應的所有水果牌。
7. 與學生進行討論，思考該如何讓遊戲可以順利的進行。(有些人拿到自己的兩張水果牌之後，就不肯再將手上的牌傳遞出去了，這種行為即會造成死結狀況)
8. 改變同學的坐法與分組人數，看看需要什麼不同的策略才能完成遊戲。以下為幾種坐法的參考方式：



路由和死結是在許多網路中都會遇到的問題，像是道路系統、電話或電腦網路系統等等。工程師們往往要花上大量的時間，試圖搞清楚該如何解決這些問題，以及如何設計網路，好讓這些問題更容易被解決。

路由，擁塞和死結可以在不同的網路裡造成許多令人沮喪的問題。想想看，有沒有遇過大塞車？或是有沒有看過在小巷子裡幾台車進退不得，每台車都無法動彈的狀況？

有時，電腦會在商業活動中「當機」（像是銀行）。這個問題是因為通訊網路的「死結」所引起的。如何設計網路，讓路由的選擇更容易、更高效和盡可能避免阻塞的狀況，是許多工程師所面臨的棘手問題。

有時不止一個人想要在同一時間取得相同的資料。如果一些資料（例如某個客戶的銀行戶頭餘額）更在更新，那在更新過程中把它「鎖定」是很重要的事。如果沒有把它鎖定，別人可能也會在同一時間進行更新，那戶頭的餘額可能就會出錯。但是，如果鎖定的項目被另一個鎖定的項目所干擾，這時就可能發生死結。

在電腦的設計中，一項最令人振奮的發展是平行計算的發明。我們可能把數百個或數千個類似個人電腦的處理器在網路中組合，就變成相當於一台功能強大的電腦。而這些電腦在平行運作時，想像一下，就好像這些電腦同極快的速度在玩水果遊戲一樣。

活動流程三 — 網路通訊協定

電腦通過訊息在網路網路上傳遞來彼此溝通。然而，網際網路有時卻不是很可靠，可能會遭遇到延遲甚至是遺失訊息的機會。因此，我們可以透過在這些要被傳遞的訊息中做點什麼處置，來確保訊息可能成功地被發送與接收。這些處置就被叫做「協定」。

開啟程式資料夾「html5_network_protocol」內的 network_protocols.html 檔案。

【基本玩法】

1. 老師將同學分為兩人一組，指定一人為發送者，另一人為接收者。
2. 老師給發送者一張隨機的訊息，上面寫有數個字元的英文字母（並非有意義的單字或句子，免得容易被猜中答案）。可以使用以下網址所提供的亂數字串產生器，若連結失效就只能自行決定內容了。

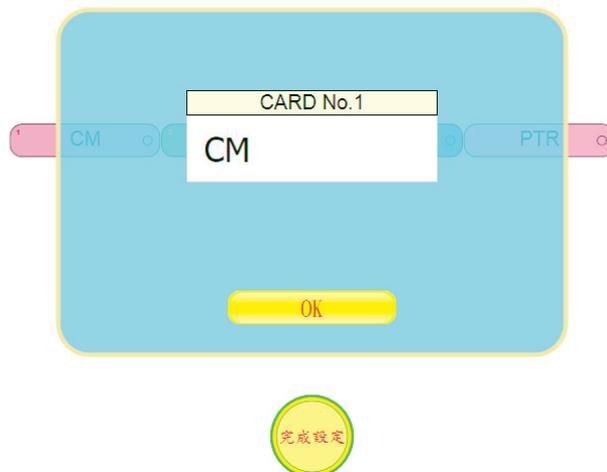
<http://cowwu.myweb.hinet.net/note/js/RandomStr.htm>

3. 第一回合，發送者指定程式中的網路環境為「可靠網路」。



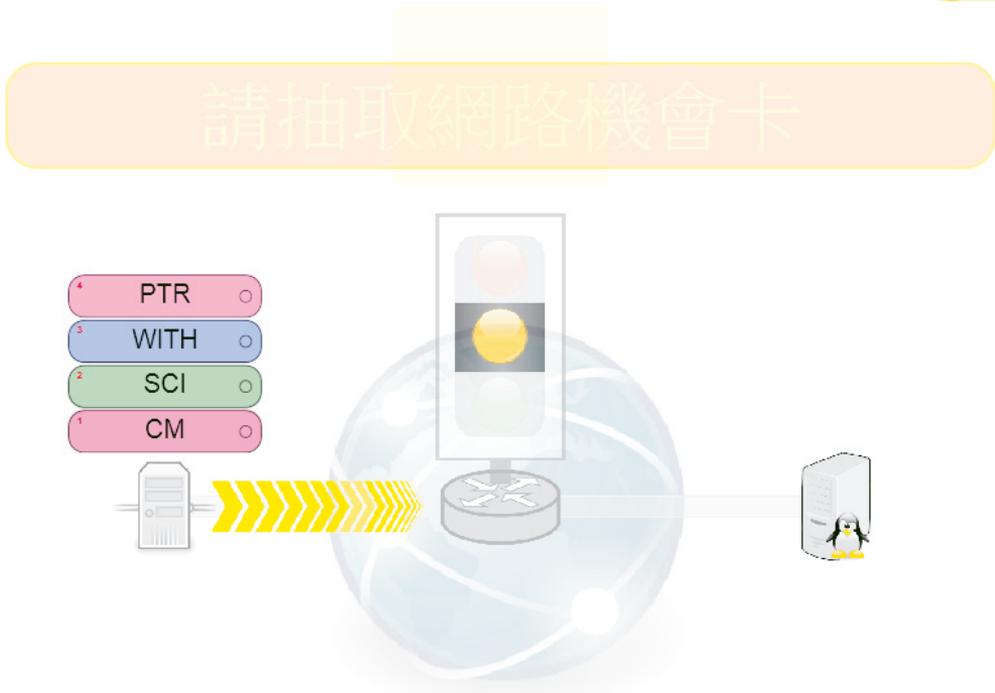
4. 按照手上的訊息，決定總共需要幾張訊息卡片，調整畫面下方的顯示數量（最多 20 張訊息卡片，每張至多只能輸入 5 個字元）。
5. 按下「完成設定」，畫面按照剛剛的設定，出現對應數量的卡片。用滑鼠點擊卡片後，就能編輯訊息卡的內容。讓同學自由決定每張卡片的內容，每張卡片都編輯完成之後，再按下畫面下方的「完成設定」按鈕，進行訊息發送的程序。

輸入卡片訊息



6. 按下「開始傳送」依序發送訊息卡。在傳送過程中，會通過畫面中央的「網路機會」（或稱為「路由」），將隨機決定這張訊息卡會被「順利傳送」（綠燈）、「被迫等待」（黃燈）或是「被丟棄」（紅燈）。進入等待區的訊息卡片，會在所有卡片都傳送完成之後，再會一次性地送往目的地。

ACK: OFF



7. 發送者傳送完所有訊息卡後，請接收者過來抄寫最後程式接收到的訊息，並試圖拼湊出正確的結果。對照發送者手上的原始訊息以決定接收者的答案正確與否。
8. 發送與接收者交換，老師重新發下隨機的訊息，分別指定網路環境為「延遲網路」與「不可靠網路」，分別再進行兩回合的遊戲。
9. 若訊息卡片因為延遲而不按照順序到達目的地，將會造成拼湊訊息的錯誤，與同學進行討論，該如何避免這個狀況呢？（提示：在編輯訊息卡片時在最前面加上編號，雖然會多佔去一個字元的空間，卻是值得的）
10. 在「不可靠網路」的情況下進行遊戲，封包常常會被迫丟失，將無法得到完整的訊息，又該如何避免這個情況？（提示：在以下進階玩法之中，加入 ACK 的確認機制即可）

【進階玩法】

1. 回到程式的主畫面，指定程式上方「ACK signal」狀態為「ON」，再按上述基本玩法的流程進行遊戲。
2. 「ACK signal」的用途：接收端在接收到訊息卡片後，會回傳一張稱為 ACK 的確認卡片回去，發送端在接到 ACK 後，才能再發送下一張訊息卡。
3. 訊息卡若在傳送時進入等待區或被丟棄，因為沒有收到 ACK 確認，在計時一段時間後必須重新傳送一次這張訊息卡。(如此就可避免基本玩法中，因為訊息丟失而造成拼湊不完整的情形發生)
4. ACK 的傳送過程中，一樣會通過網路機會路由，若抽到「被迫等待」或是「被丟棄」，則必須重新傳送一次 ACK。
5. 同樣讓同學在三種不同狀態的網路環境中進行遊戲。
6. 與同學討論 ACK 功能存在的意義，表面上看起來讓整個傳送的過程更為複雜與費時，但它是否為必要的設計呢？

結論

在網路上資料是被分割成許多封包來傳送。但是這些封包傳輸時經過的管道不一定一直都是穩定的。個別的封包有時會受損、遺失或不照順序送達。

在「網路通訊協定遊戲」中，每張訊息卡片就被稱為一個封包，而在訊息卡片中被輸入的文字就是資料。在遊戲過程中，由於網路狀況的不同，封包也許會不按順序抵達接收端。因此，我們會嘗試在每個封包中加入被稱為「標頭資訊」的內容，讓接收者能藉由安排正確的封包順序，來還原原本的訊息內容。但是，因為每個封包(訊息卡)的大小是有限制的，標頭資訊的多寡會影響每張訊息卡能被放入的資料量，所以發送者必須衡量標頭資訊與內容資訊的平衡點。

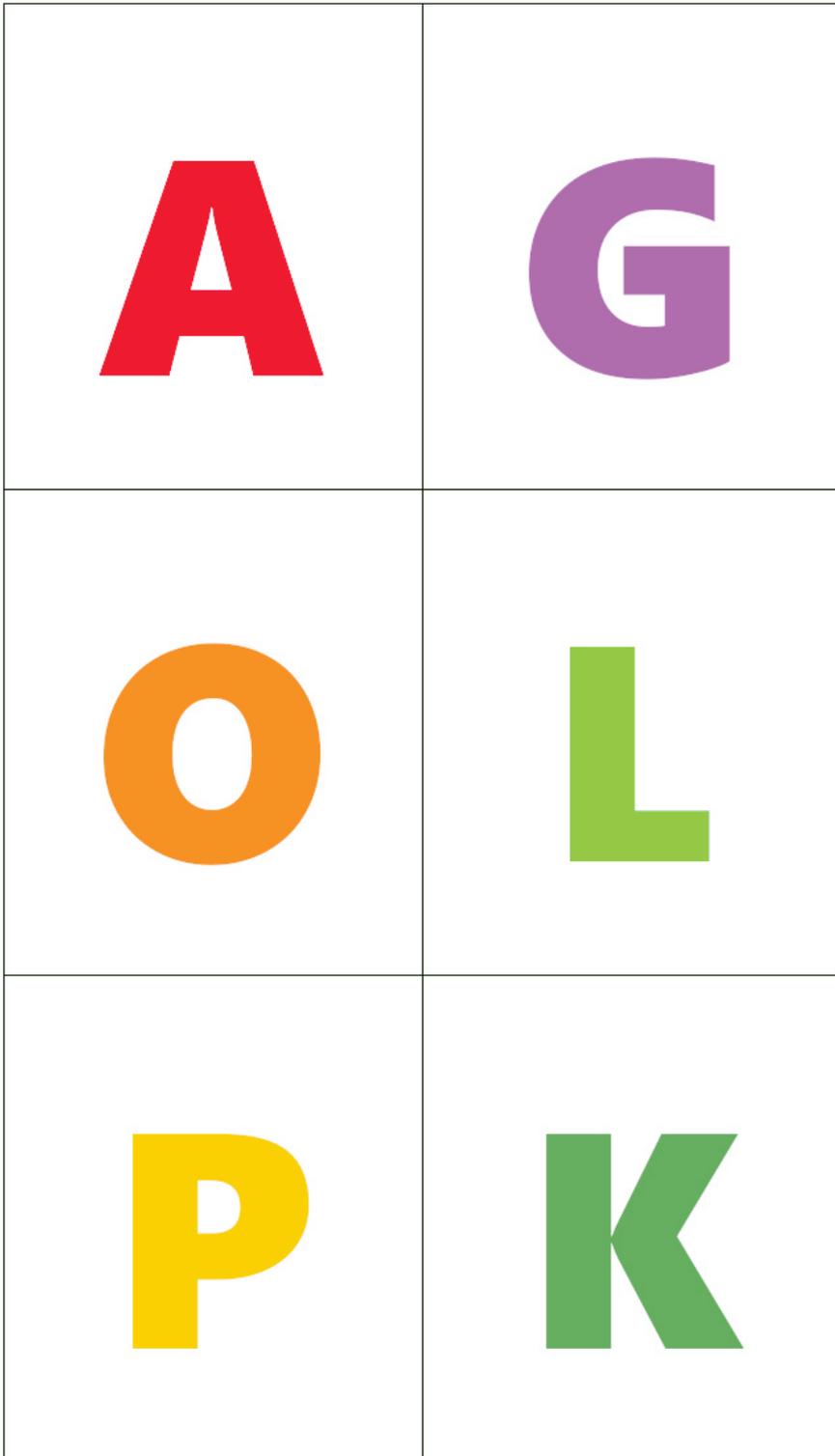
我們會發現，在開始傳送前，發送者跟接收者若能事先溝通出一個交換資料的方法，包括標頭資訊的編碼方式，與遇到突發狀況時的處理方法，這樣就能讓訊息傳遞的過程更為順利。真正的網際網路通訊協定有很多種，像是 TCP、UDP 等，協定本身的設計就會考慮平衡種種的因素，來建立一個可信賴且有效率的資料傳輸協定。

程式的自訂項目

程式資料夾中的 `text_compression_config.js` 檔案，提供訊息的自訂，以下這段程式碼可修改三種網路環境中，抽取到「傳送」、「延遲」與「丟棄的機會」。Level 0 為「可靠網路」環境，Level 1 為「延遲網路」，Level 2 為「不可靠網路」，注意三種機會的比例。

```
cfg_levelActionConf = new Array(  
    [0, 8, 27],    //level 0 drop:0 , delay:5 , deliver:30  
    [0, 17, 18],  //level 1 drop:0 , delay:17, deliver:18  
    [8, 13, 14],  //level 2 drop:8 , delay:13, deliver:14  
);
```

附錄 8-1 卡片式水果遊戲



蘋果	柳橙	鳳梨
蘋果	柳橙	鳳梨
葡萄	檸檬	奇異果
葡萄	檸檬	奇異果

第九章 資料加密

教學時數：3 節

事前準備：

1. 不插電：附錄 9-1，基本替換法的字母對照表，有已經寫好以及空白的兩種版本供老師選用。附錄 9-2 則是進階替換法所需要使用的 Tabula Recta 表格。
2. 插電：附錄程式檔，資料夾名稱為「math_flashcards」。

活動流程一 — 基本訊息加密法的介紹

你一定聽過間諜和特務使用隱密的代碼或隱形墨水寫下並交換訊息之類的故事。那就是「密碼學」這門學問的源起。密碼學是一門傳遞訊息的藝術。在二戰期間，英國人建造了專門破解密碼的電子機器，並且用這個機器來破解敵軍的軍事機密訊息。接著，電腦的出現改變了一切，並且使密碼學進入了一個新的時代。在過去完全無法想像的大量計算，現在可以透過配置完善的電腦來進行。當人們開始互相分享電腦系統時，密碼出現了新的用法。而當電腦連在網路上時，人們就必須開始保護自己的資訊，以免被其他人竊取。當電子郵件出現時，確定郵件內容有沒有被改過，是不是真的是由發信者所發出，還是別人偽造發信者的身份所發的，就變得很重要了。現在電腦可以使用網路銀行的功能，也可以透過電腦買賣商品，在電腦網路上我們就非常需要訂單與金流的安全管道。而恐怖份子攻擊電腦系統所帶來的威脅日益增加，也使得電腦安全越來越重要。

講到密碼學，可能讓你想到儲存密碼的電腦，和弄亂訊息的文字使得敵人無法解讀之類的事。但實際上並不是這樣。現代電腦系統不會儲存密碼，因為如果系統這樣做，任何設法取得密碼的人就可以突破系統內所有的安全機制，那會是一個大災難：他們可以盜領帳戶內的錢、假裝是其他人傳送訊息、讀取所有人的機密資料、甚至命令軍隊推翻政府。

當電腦逐步入侵我們的日常生活，密碼學的應用也變得更加重要。大多數的人根本不知道現代的加密協定是怎麼一回事。結果就是當大型機構 - 包括政府與商業機構 - 建立了涉及個人資訊的系統，會變成由這些機構來做重要決定，包括如何處理事物、要蒐集什麼資訊、什麼東西可以開放給誰等等。如果人們對現代

科技產生的可能性有更深的了解的話，他們就能主動參加更多像這樣的決策，而整個社會也會建立起一個不同的資訊基礎。

密碼學的理論本質一般來說被認為是非常先進的。但這些想法本身並不困難。難以理解的是它的技術性，而不是它的觀念。在現實的狀況中，例如電子商務，這些技術都用軟體包裝好了，因此加密技術就變得容易使用。但是了解這些技術的基礎想法，知道這些軟體能做什麼也是很重要的事。

政府對密碼系統有很大的興趣，不只是因為要維護官方通訊的安全性，也是擔心從事非法活動的人們，像是毒品走私和恐怖活動等等，也會使用加密通訊。如果這樣的人使用加密，那竊聽就會變得沒用，除非知道怎麼解密。

密碼學的觀念並不只是保密訊息而已。另外還有許多應用，像是保證訊息真的是傳送者所發出 - 這種機制稱為「認證」，沒有這個機制，就不會有現在的電子商務。還有用在電子投票，確保其他人 - 包括營運該系統的人 - 不會知道投票者投給誰，同時還能避免一個人投兩票以上。

加密技術對於資訊安全來說是個關鍵。而現代的加密技術的關鍵，則是訊息的發送者只需要使用某些公開的資訊，就可以將一段訊息鎖住，只有特定的人才能解鎖打開。

這就如同每個人都買了一個掛鎖，在上面寫上他們的名字，然後把它們全部放在相同的桌上供其他人使用。當然每個人自己都有自己鎖的鑰匙，但掛鎖是開著的，鎖上它們不需要任何鑰匙。如果你要寄給某人一則機密的訊息，你只需要把訊息放入一個盒子，拿鎖把它鎖起來，然後寄給收件者。即使盒子落入別人之手，因為沒有鑰匙，所以也無法打開。有了這樣的機制，大家就不需要事先溝通解鎖的密碼為何。在數位的世界中，我們不是直接拿起掛鎖鎖住盒子，而是去複製這個掛鎖，並使用複製後的掛鎖，把原始的掛鎖留在桌上。

這個使用掛鎖的機制，我們稱為「公開金鑰加密」。除此之外還有其他的加密方式，以下介紹幾種基本的方法：

【換位法】

1. 加密訊息的產生

假設欲傳送的訊息為「MERRYXMAS」，假設訊息傳送者與接收者事先約定使用「換位 4」加密，則依序在每個字母底下加上數字。編號設定到 4 之後，則下個字母再重新由 1 開始設定號碼。

M	E	R	R	Y	X	M	A	S
1	2	3	4	1	2	3	4	1

接著先將所有編號為 1 的字母依序排列，接著排列編號 2 的字母，依些類推。排列的結果如下，如此則產生了加密後的訊息「MYSEXRMA」。

M	Y	S	E	X	R	M	R	A
1	1	1	2	2	3	3	4	4

2. 已加密訊息的還原

假設訊息傳送者與接收者事先約定使用「換位 3」加密，接收者得到的訊息如下：

M	R	M	E	Y	A	R	X	S
1	1	1	2	2	2	3	3	3

先取字母編號第一個 1、第一個 2、第一個 3 等三個字母，接下來取第二個 1、第二個 2 及第二個 3 等三個字母，依此類推。如此則得到還原過後的訊息為「MERRYXMAS」。

M	E	R	R	Y	X	M	A	S
1	2	3	1	2	3	1	2	3

【替換法】

1. 加密訊息的產生

首先與同學約定好英文字母替換對照表如附錄 9-1，可以選擇列印張貼或是統一投影讓同學觀看（當然老師可以選擇重新定義不同的對照表）：

原始	A	B	C	D	E	F	G	H	I
替換	O	P	E	M	X	A	I	U	V
原始	J	K	L	M	N	O	P	Q	R
替換	D	F	Q	B	Y	L	J	T	W
原始	S	T	U	V	W	X	Y	Z	
替換	N	C	H	K	G	R	Z	S	■

假設欲傳送的訊息為「MERRYXMAS」，按照表格所對應的替換字母如下，因此傳遞時會傳送訊息「BXWWZRBN」。

M	E	R	R	Y	X	M	A	S
B	X	W	W	Z	R	B	O	N

2. 已加密訊息的還原

按照附錄 9-1 的對照表，若接收到的訊息為「BXWWZRBN」，按照表格所對應的原始字母還原為「MERRYXMAS」。

M	E	R	R	Y	X	M	A	S
B	X	W	W	Z	R	B	O	N

活動流程二 — 進階訊息加密法的介紹

【表格法 - Tabula Recta】

將附錄 9-2 的表格列印出來，每個同學手上持有一份，以方便進行對照之用。

1. 加密訊息的產生

在進行加密前，必須先約定好一個「金鑰」，假設金鑰內容為「NTCT」，預傳送的訊息為「HELP」。

則加密訊息的

第一個字母為第 H 行、第 N 列的「U」

(註：先看列、再看行的結果也是一樣的)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O

依此類推：

- 第二個字母為第 E 行、第 T 列的「X」
- 第三個字母為第 L 行、第 C 列的「N」
- 第四個字母為第 P 行、第 T 列的「I」

如此則得到加密訊息為「UXNI」

(註：金鑰長度可以比原始訊息更短，加密時依序取用金鑰的字母，用完時則從頭開始使用第一個字母)

2. 已加密訊息的還原

約定金鑰的字母，代表「行」的位置，還原時要在這行找到加密後的訊息字母，該字母所在的「列」則代表原始的訊息。(進行還原時)

假設事先約定密鑰為「NTCT」，接收到的加密訊息為「UXNI」。

在第 N 行找到字母「U」，它在第 H 列，所以原始訊息的第一個字母為「H」

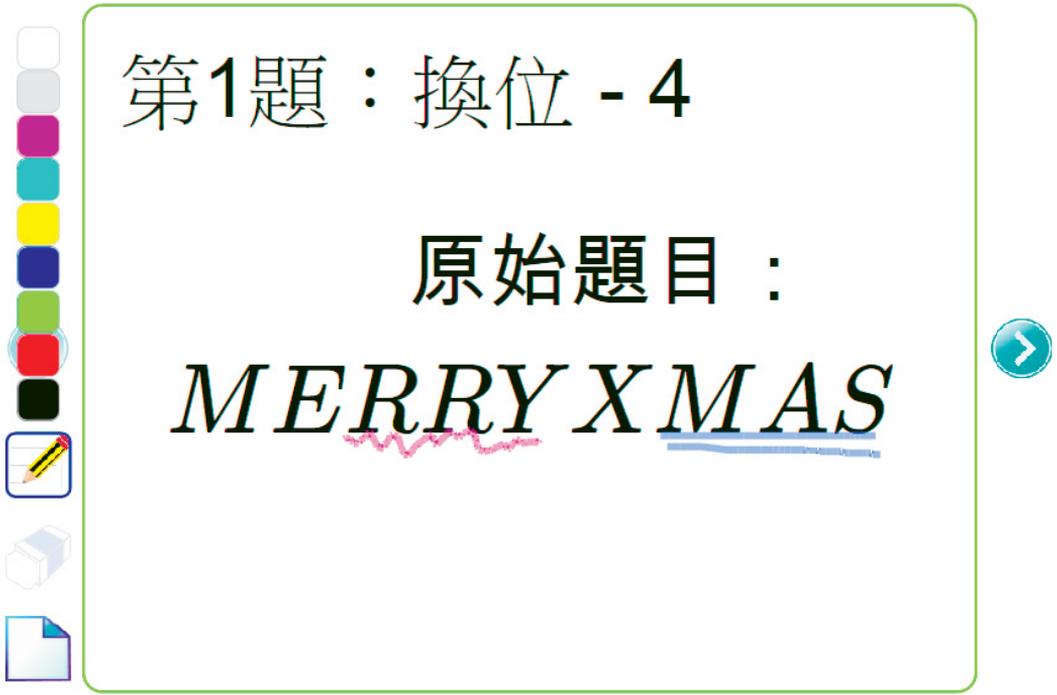
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O

在第 T 行找到字母「X」，它在第 E 列，所以原始訊息的第一個字母為「E」
 在第 C 行找到字母「N」，它在第 L 列，所以原始訊息的第一個字母為「L」
 在第 T 行找到字母「I」，它在第 P 列，所以原始訊息的第一個字母為「P」

如此則得到加密訊息為「HELP」。

輔助程式 — FLASHCARD 的使用與自訂方法

程式資料夾「math_flashcards」的 math_flashcards.html，是一個可以自訂卡片正、反面訊息的教學輔助程式。同時附帶有畫筆加註、橡皮擦與劃記清除的功能，只要完成自訂，亦可應用在任何學科的教學。



The screenshot shows a digital flashcard interface. On the left side, there is a vertical toolbar with icons for selecting colors (white, grey, pink, cyan, yellow, blue, green, red, black), a drawing tool (pencil on a notepad), a folder icon, and a document icon. The main area of the flashcard is enclosed in a green rounded rectangle and contains the following text:

第1題：換位 - 4

原始題目：

MERRY XMAS

The word "MERRY XMAS" is displayed in a stylized, italicized font. The word "MERRY" has a pink wavy underline, and the word "XMAS" has a blue straight underline. To the right of the main text area, there is a blue circular button with a white right-pointing arrow.

2014.01.17.f by gsy/zn

以記事本軟體開啟 math_flashcards_set.js

如果有 back slash , 因為 escape 的關係 , 必須使用兩個 \\
想強制換行的話 , 將內容頭尾加上 \\[\\]
換行的地方用四個反斜線 \\\\

例如

```
'第 2 題：換位 - 3 \\[ 原始題目：\\\\CANPLUGGED\\']  
  ; 加密後：\\[CPGDALGNUE\\']
```

該段設定是將「原始題目：CANPLUGGED」強制在「換位 - 3」之後換行，原始題目後方的「：」後換行再顯示「CANPLUGGED」



第2題：換位 - 3

原始題目：

CANPLUGGED

The screenshot shows a flashcard with a green border. On the left side, there are four icons: a left arrow, a pencil, a cube, and a document. On the right side, there is a right arrow. The text on the card is centered and reads: '第2題：換位 - 3' followed by '原始題目：' and then 'CANPLUGGED' in a large, italicized, black serif font.

2014.01.17.f by gsj/an

附錄 9-1 基本替換法表格

原始	A	B	C	D	E	F	G	H	I
替換	O	P	E	M	X	A	I	U	V
原始	J	K	L	M	N	O	P	Q	R
替換	D	F	Q	B	Y	L	J	T	W
原始	S	T	U	V	W	X	Y	Z	
替換	N	C	H	K	G	R	Z	S	

原始	A	B	C	D	E	F	G	H	I
替換									
原始	J	K	L	M	N	O	P	Q	R
替換									
原始	S	T	U	V	W	X	Y	Z	
替換									

附錄 9-2 進階替換法表格 Tabula Recta

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

國家圖書館出版品預行編目資料

偷插電的資訊科學教師手冊 / 謝宗翔編著 ; 顏國雄程式實作 . -- 初版 . --
新北市 : 軟體自由協會 , 2017.04
面 ; 公分
ISBN 978-986-92929-1-7(平裝)
1. 電腦教育 2. 資訊教育 3. 中等教育
524.375 106006173

Computer Science for Taiwan Teens

偷插電的資訊科學 教師手冊

教材編著：謝宗翔

程式實作：顏國雄

發行人兼總編輯：翁佳驥

出版：中華民國軟體自由協會

地址：22170 新北市汐止區汐萬路三段 199 巷 26 弄 4 號

電話：(02) 5593-3701

傳真：(02) 7741-7371

電子郵件信箱：service@slat.org

印刷：堯棠印藝有限公司

地址：新北市鶯歌區尖山路 282 號 8 樓

初版：2017 年 4 月

授權：CC-BY-NC-SA 4.0

https://creativecommons.org/licenses/by-nc-sa/4.0/deed.zh_TW