

這是 2014 / 11 / 26 逢甲系程(CH05)的補充資料，這裡只有給”重要”概念和片段程式碼，如果還是不熟悉，一定要找書來看，或是找人問，最好最好要實際寫一遍程式試試，加深印象

阿德

陣列

基本觀念：

```
int numbers[6];
```

宣告長度為 6 的 int 陣列。

在記憶體裡會長這樣：

0x1000	??	numbers[0]
0x1004	??	numbers[1]
0x1008	??	numbers[2]
0x100C	??	numbers[3]
0x1010	??	numbers[4]
0x1014	??	numbers[5]

(int 佔 4 個 byte)

(記憶體位置不一定從 0x1000 開始)

陣列邊界檢查：

```
int numbers[6];
```

```
numbers[6] = 100; //<- 可以執行，但不知道何時會發生問題
```

雖然 numbers[6] 已經超出陣列邊界(請參考上圖)，但電腦只會乖乖執行，有時會出問題，有時卻不會，寫程式時要特別注意，如果**放任不管**，會導致除錯很困難。

(在一些語言(像是 Java)會去檢查索引*是否超出邊界，但電腦就要花一點點點點時間檢查，早期電腦沒有這麼多的運算能力，或許這是 C 語言沒有邊界檢查的原因)

*索引：是用在中括號的值，代表編號。

二維陣列：

看下方的程式碼：

```
//宣告一個int二維陣列，有2列(row)、3行(col)
```

```
int arr[2][3] = {  
    {0, 1, 2},  
    {3, 4, 5}  
};
```

```
int number1 = arr[0][1]; // number1 會等於 1
```

在記憶體裡會長這樣：

int arr[2][3]		col		
		0	1	2
row	0	0	1	2
	1	3	4	5

0x2000	0	arr[0][0]
0x2004	1	arr[0][1]
0x2008	2	arr[0][2]
0x200C	3	arr[1][0]
0x2010	4	arr[1][1]
0x2014	5	arr[1][2]

二維陣列邊界：

再看看下列程式碼：

```
int number2 = arr[0][3];
```

你覺得 number2 會抓到甚麼？

答案是：3 (也就是 arr[1][0])
(想想看，等教完指標再解釋)

字串

電腦要怎麼存一句話，像是「Hello, world!」之類的？

.....

其實也只能一個一個字元儲存

請看下列程式碼：

```
char str[] = {'h', 'e', 'l', 'l', 'o', ',', ' ', 'w', 'o', 'r', 'l', 'd', '!', '\0'};
```

註：宣告陣列時，如果有直接初始化，[]不用填大小，編譯器會幫你算 (如果算不出來，你就要跟編譯器說陣列大小啦)

這樣寫很麻煩，所以...

```
char str[] = "Hello, world!"; //不用加'\0'，編譯器會幫忙加
```

這樣寫也可以

有沒有注意到用黃色標記的部分，'\0'，在記憶體會存成 0，代表字串的結尾，所有 C 語言的函式在處理字串時，都是利用這個事實，所以忘了加結尾 0 ('\0')，會造成一些函式無法正常運作。

常見的字串處理函式：

(註 1：我只有介紹一點點，抱歉^_^)

(註 2：下列介紹的函式需要 include "string.h")

(參考資料：<http://www.cplusplus.com/reference/cstring/>)

strlen：計算字串長度

```
size_t strlen(const char * str);
```

```
char str[] = "Hello, world!";  
printf("%d", strlen(str));
```

會印出 13

strcpy：複製字串

```
char *strcpy(char *destination, const char *source);
```

```
char str[] = "Hello, world!";  
char str2[40];  
strcpy(str2, str); //將str複製到str2  
//要注意目的地空間要足夠
```

讀取字串：

```
char str[12];  
scanf("%s", str);
```

這樣就可以從鍵盤讀字串了。

如果輸入長度超過 11 (> 11) 的字串時，會發生什麼事？

嗯.....有時會直接當掉，有時會程式會繼續執行，卻準備跳下懸崖^[1]

為了防止錯誤發生，我們可以：

```
scanf("%11s", str);
```

這行只會先讀入 11 個字

補充結束

[1] 編程創藝：編寫出無暇的程式碼 p.156