寫點科普，請給指教。

瞭解各產業領域的運行規則，保有思考與觀察力的敏銳。

MENU

C語言◘: 超好懂的指標，初學者請進～

2017-05-15

指標（pointer）這個功能在 C 語言中有著非常重要的地位。

C 語言中特有的指標，可以透過記憶體映射的方式直接控制硬體，這也是為什麼 C 語言在硬體系統特別強大的原因，包括資料結構（陣列/字串/鏈結串列）、系統程式（編譯器/作業系統）、演算法，都會進一步使用到。

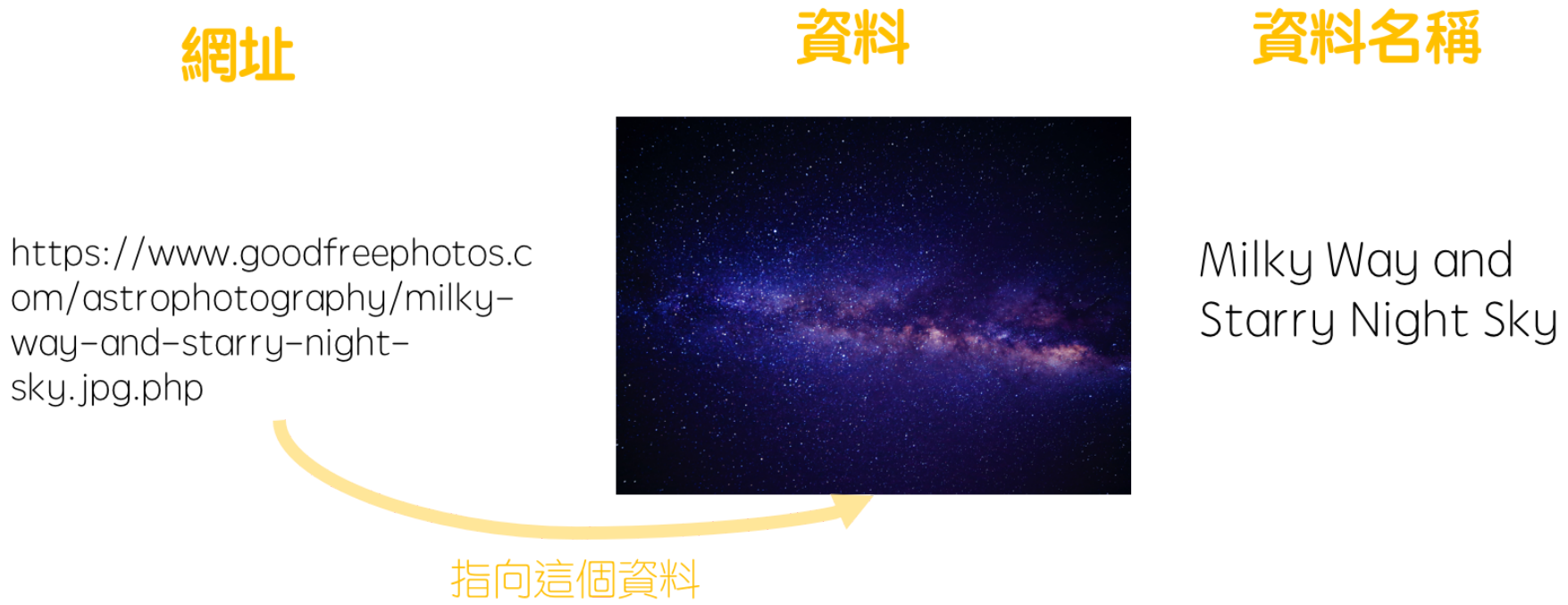
但對於初學者來說，一開始無法釐清指標、導致後續指來指去，指到最後往往變成噩夢。

今天我們的目標，就是從頭開始把指標介紹的非常簡單。

什麼是指標（POINTER）

這張照片叫「Milky Way and Starry Night Sky」，是一張看著便賞心悅目的美麗星空圖，當桌布做美工都適合。

如果你想要這張照片的話，最直接的辦法就是我給你這張照片的網址「https://www.goodfreephotos.com/astrophotography/milky-way-and-starry-night-sky.jpg.php」，讓你在網址上拿到這張照片。



這和指標有什麼關係呢？概念很類似噢！只要我給你這個網址，你就可以藉由這個網址拿到這張照片，也就是說網址是指向這個資料的「指標」。

哦～原來資料的地址就是指標嘛，是不是非常簡單的概念呢？

等等，那 C 語言中的指標是長什麼樣子？

讓我們來看看這段程式碼：

```
1 void main(){
2   int a = 15;
3   int b = 2;
4   int c = 39;
5   int d = 180;
6   int e = 67;
7 }
```

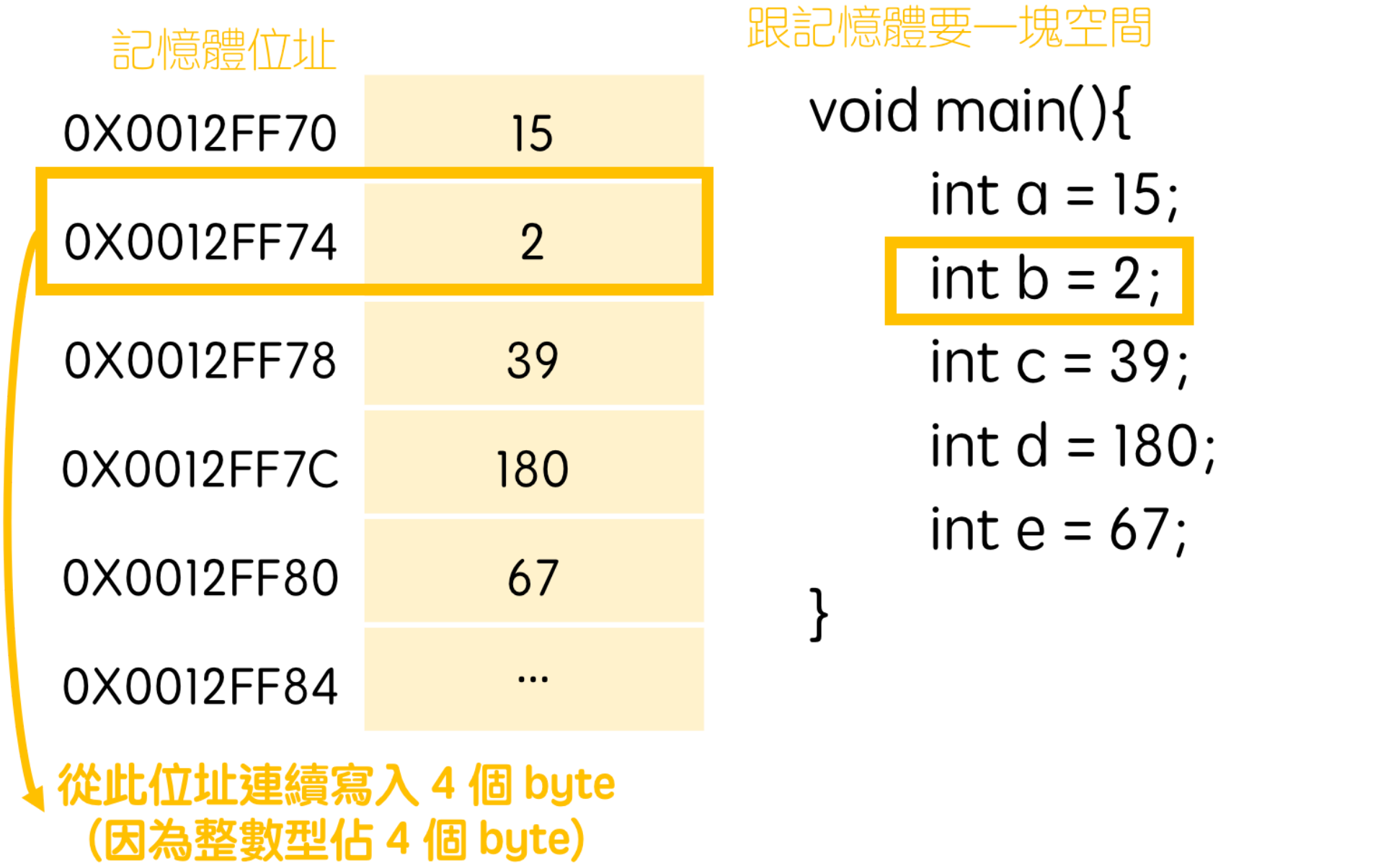
在這個程式碼中，我們宣告了五個變數。

以 int b = 2 為例，在宣告好變數、程式開始執行後，會去記憶體中要一塊儲存空間，然後把 2 這個資料放進去。

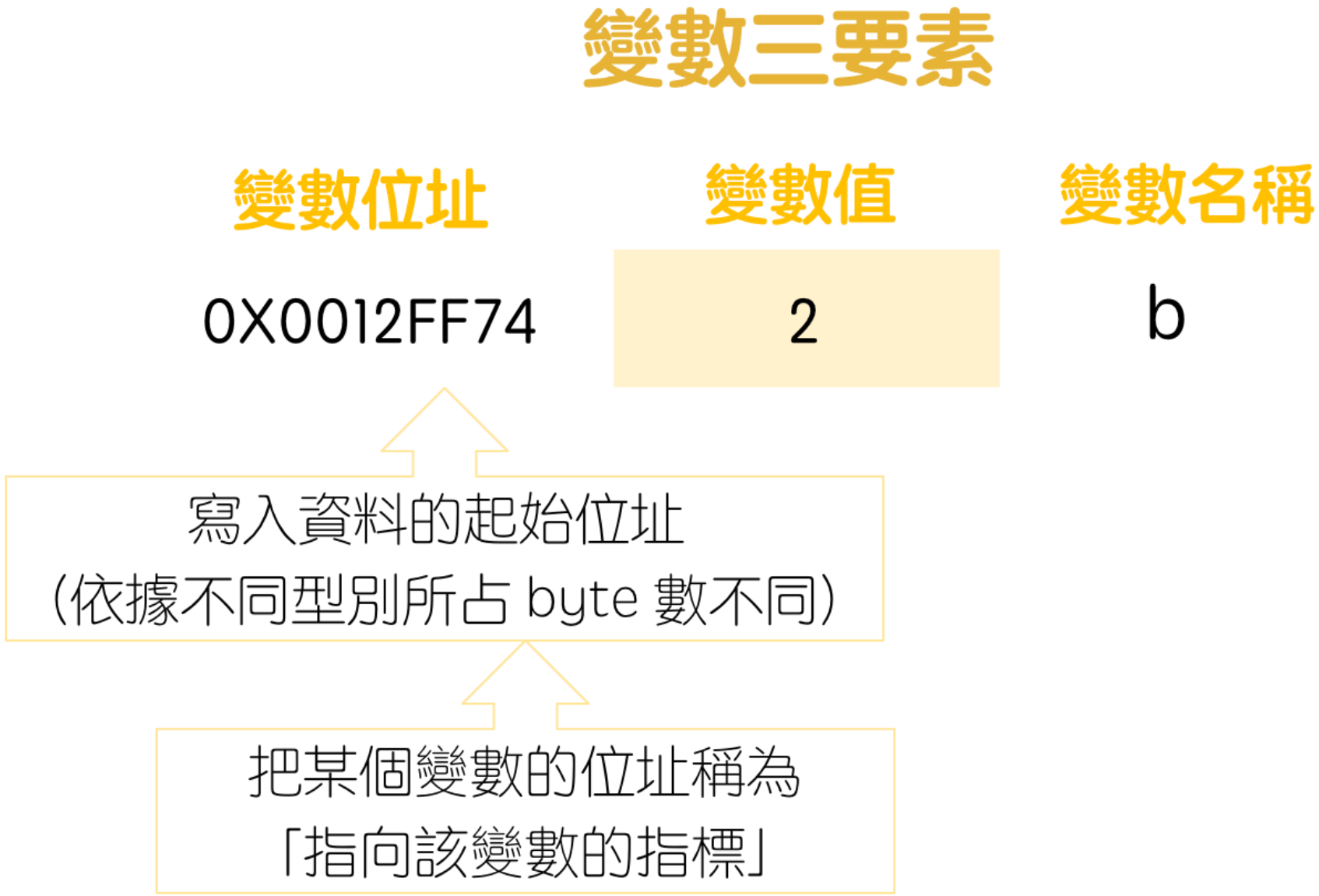
還記得我們說過記憶體就像一個大櫃子、每個格子都有相對應的地址嗎？這個 2 的地址就是在記憶體中的某一個地方。

但我們說過，記憶體中一個格子的大小是 1 個 byte，而一個 int （整數型）的大小就占了 4 個 byte，所以這邊寫的地址，是 2 這個整數所占的這一塊記憶體空間的起始地址。

從起始地址開始起算、共佔了 4 個格子，也就是 4 byte。



也就是說，當我們宣告一個變數時，總共會有三個要素：



程式會向記憶體要一塊空間來儲存變數值，所以這個儲存空間有一個起始位址。再加上這個變數的名稱（b）與變數值（2）。

通常我們會把變數的位址，稱為「指向該變數的指標」。

在這邊需要特別注意的是，我講的是「指標」，而不是「指標變數」；這兩個是不同的東西。

拿一開始的照片作為對照例子，是不是很清楚呢？

變數三要素

變數位址	變數值	變數名稱
0X0012FF74	2	b

https://www.goodfreephotos.com/astrophotography/milky-way-and-starry-night-sky.jpg.php

Milky Way and Starry Night Sky

但我們在宣告一個變數，比如宣告 b 的時候，要怎麼知道到底是跟記憶體要了哪一塊地址放這個 b 呢？怎麼樣才能看到這個變數的位址？

這邊就要介紹一下在 C 語言裡面，有個運算符號是用來「取址」，就是「&」。

怎麼做呢？請參考以下程式：

```c
#include <stdio.h>

int main(void) {
    int b = 2;

    printf("變數 b 的值：%d\n", b);
    printf("變數 b 的記憶體位址：%p\n", &b); //%p為印出地址的16進位表示法

    return 0;
}
```

看一下在 Terminal 中編譯完的結果：

```
sixde-Air:programming sixstockfeelair$ g++ address.cpp -o address
sixde-Air:programming sixstockfeelair$ ./address
變數 b 的值：2
變數 b 的記憶體位址：0x7fff54a109c8
```

這個程式中，宣告了一個 `int` 整數變數 b，並藉由印出「&b」的值，知道 b 所在的記憶體位址是 0x7fff54a109c8（16進位表示法）。

從 0x7fff54a109c8 開始的 4 個 byte 都是 b 所配置到的記憶體空間，儲存了 2 這個值。

這時你一定想問：有地址要做什麼？我們宣告完變數後也用得好好的，沒事要它的地址幹嘛？

事實上我們拿到一個地址，都是為了要去到這個地址上、以抓取上面的變數。（知道了朋友的地址，目的不就是要去拜訪他嗎？或就像我們利用網址去抓照片一樣）

但問題來了，怎麼樣能利用一個變數的地址、去拿到這個變數呢？直接把地址寫出來然後執行嗎？

答案是不行的。我們要利用 C 語言中的另一個運算元「*」來做這件事。

```c
#include <stdio.h>

int main(void) {
    int b = 2;

    printf("變數 b 的值：%d\n", b);
    printf("變數 b 的值：%d\n", *&b);//從這個地址中取出變數b的值

    return 0;
}
```

來看看 Output 的結果：

```
cdsixde-Air:~ sixstockfeelair$ cd desktop/programming
sixde-Air:programming sixstockfeelair$ ./address
變數 b 的值：2
變數 b 的值：2
```

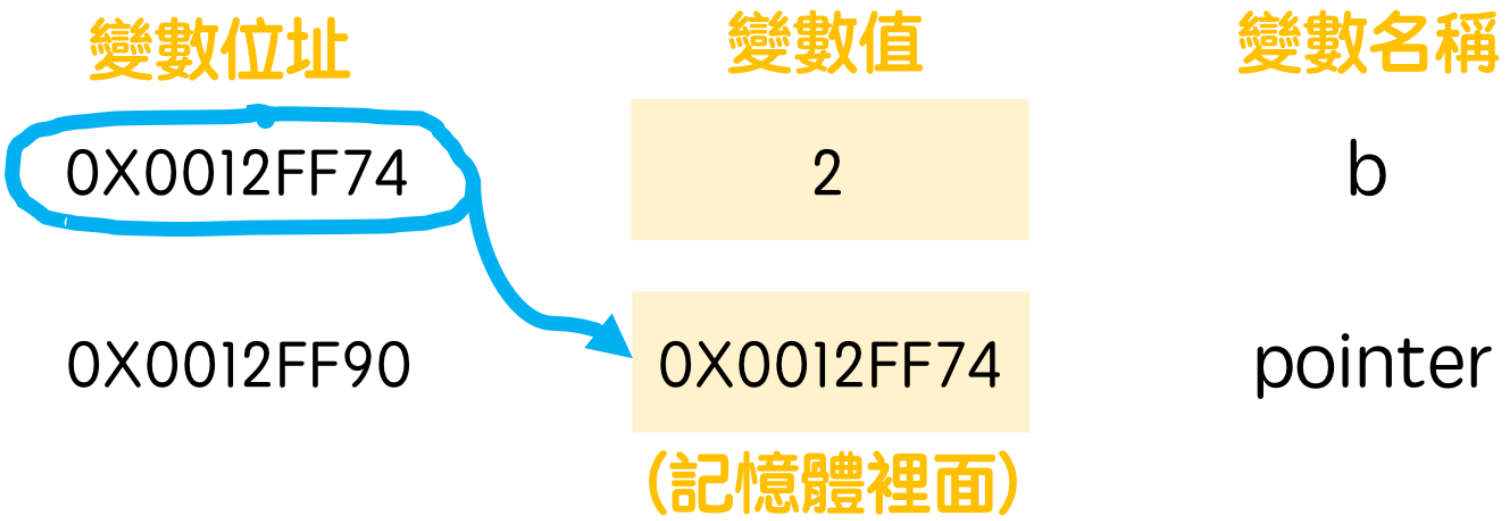
奇怪，明明印出 b 變數的值，和「從 b 這個位址中取值」印出 *&b，還不是一樣的答案？這樣的話為什麼要寫那麼多？

你說的沒錯。事實上，「*&b」和「b 」的意義是等價的。

指標變數（POINTER VARIABLE）

講完了什麼叫指標後，接下來讓我們看看「指標變數」。

還記得我們說過，指標 (Pointer) 就是某變數的位址。而這邊的指標變數 (Pointer Variable)，則是用來存放指標的變數。



指標 (Pointer)：某變數的位址

指標變數 (Pointer Variable)：用來存放指標的變數

宣告一個指標變數，和一般宣告變數一樣，
是跟記憶體要一個區域、存放這個變數的值。
只是這個變數的型別是指標。

案例中的 pointer 就是一個指標變數。變數都是用來存放「值」的，而整數型變數 int 就是存整數、字元型變數 char 就是存字元。所以這個指標變數就是用來存「地址」的變數。

也就是說，宣告一個指標變數，和一般宣告變數一樣， 是跟記憶體要一個區域、存放這個變數的值。只是這個變數的型別是指標。

另外，由於 pointer 中存的地址是變數 b 的值，因此我們又把 pointer 稱為「變數 b 的指標變數」。

這些概念非常的簡單，只是一定要弄清楚。

接下來你可能會問：要怎麼去宣告一個指標變數呢？

我們可以採用「*」這個運算符號。

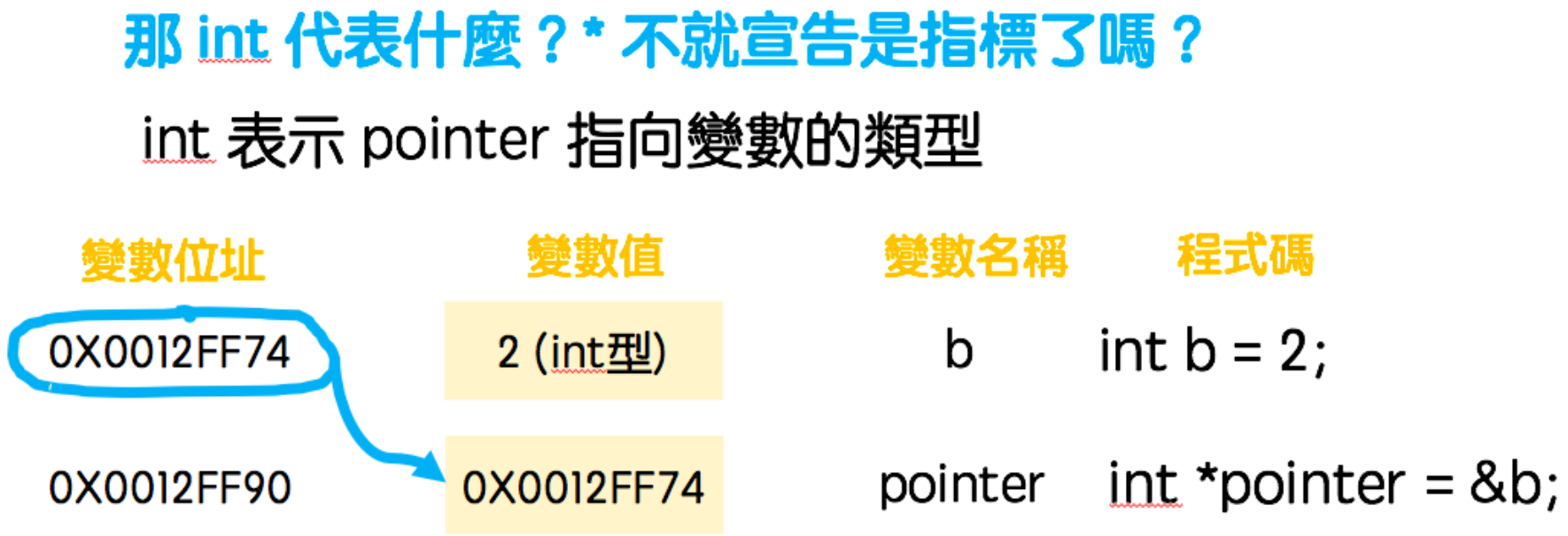
```
1 int* pointer
```

pointer 表示這個變數的名稱，而 * 表示 pointer 這個變數是個指標。

等等，那 int 代表什麼意思？星星符號運算元 * 不就說是個指標了嗎？

int 代表這個指標變數指向的變數的類型。… 你在說什麼在繞口令嗎？

直接看下圖吧！



具體來說的程式碼長這樣：

```c
int b;
//跟記憶體要一塊區域稱為b,這塊區域專門放int型變數值

b = 2;
//把2這個值給變數b

int* pointer;
//跟記憶體要一塊區域稱為pointer,這塊區域專門放指向int型變數的指標（地址）

pointer = &b;
//把變數b的地址值給pointer，注意不能寫成 pointer = b;
```

還記得「&」代表「把這個變數的地址取出來」的意思嗎？要注意，這邊絕對不能寫成 pointer = b; ，因為 pointer 是專門存放地址的變數。

如此一來，我們就稱「指標變數 pointer 指向了變數 b」，是不是很好懂呢！

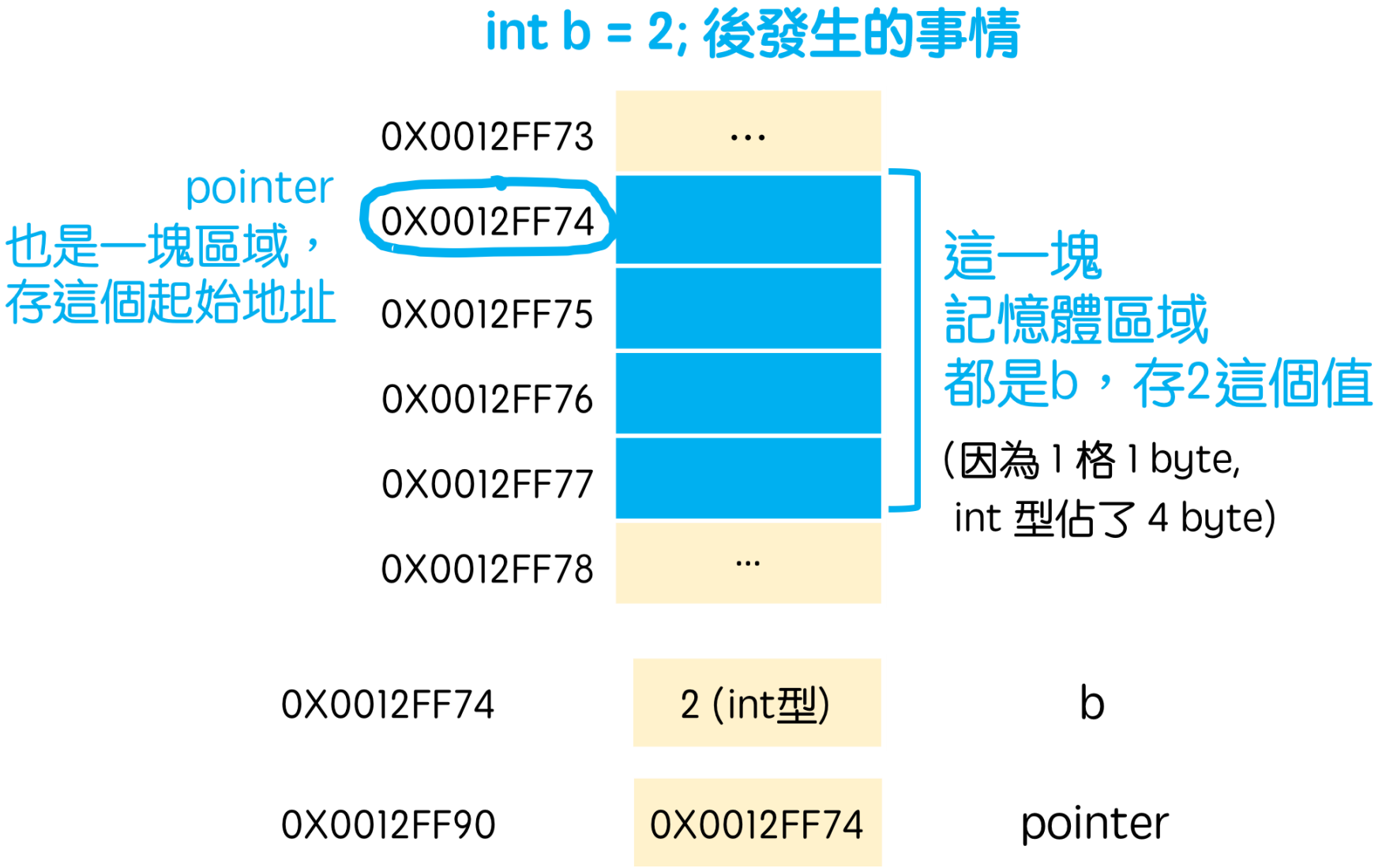
很多書上寫的有些模糊，初學者又還搞不清楚時，就會導致指標、指標變數、指標變數宣告等產生混亂。

接下來還想問個問題：能不能利用 pointer 去拿到它指向的 b 這個變數呢？當然可以。這邊同樣要利用到 * 這個運算符號。

當我們跑完這個程式碼之後：

```
1 int b = 2;
2 int* pointer = &b;
```

會發生這件事：



也就是說變數 b 在記憶體中對應了一塊儲存空間，而這塊儲存空間總有一個起始的地址。所以 pointer 對應到的就是這個起始地址。

在這種狀況下，就可以用「*pointer」來拿到這個變數。

這裡的「*」，和宣告指標變數的 int* pointer 的意義不太一樣。反而是和「&」相對應——「&」代表「取出地址」、「*」代表「取出內容」。

等等，那所謂的「*pointer 取出的內容」指的到底是變數 b、還是變數 b 的值 2？

這兩個是不同的東西喔！變數 b 是這塊區域，2 是值。

答案是：*pointer 代表的就是變數 b。所以我們可以把 *pointer 當作變數 b 來使用。

直接看程式碼吧：

```c
#include <stdio.h>

int main(void) {
    int b = 2;
    int* pointer = &b;

    printf("變數 b 的值：%d\n", b);
    printf("變數 b 的地址：%p\n", &b);
    printf("pointer 的值：%p\n", pointer);
    printf("\n"); //換行

    *pointer = 100;

    printf("*pointer 的值：%d\n", *pointer);
    printf("變數 b 的值：%d\n", b);
    printf("變數 pointer 的地址：%p\n", &pointer);

```

```
18      return 0;
19  }
```

執行這段程式碼的結果：

```
sixde-Air:programming sixstockfeelair$ ./address
變數 b 的值：2
變數 b 的地址：0x7fff551b49c8
pointer 的值：0x7fff551b49c8

*pointer 的值：100
變數 b 的值：100
變數 pointer 的地址：0x7fff551b49c0
```

我們可以看到一開始的變數 b 的值被設定為 2，所以印出來也會是 2。然後用「&b」取出變數 b 的地址為「0x7fff551b49c8」。

由於 &b 的值被賦予給 pointer，所以把 pointer 印出來後同樣也是「0x7fff551b49c8」。

由於我們說過 *pointer 就是其指向的變數 b，所以在這邊我們試著把 *pointer 中的值改成 100，然後印看看原有的變數 b 會不會跟著改變。

發現會欸！2 被改成 100 了！

最後，由於存放 pointer 這個變數的地址，和變數 b 的地址不一樣，所以利用「&pointer」後，可發現地址「0x7fff551b49c0」和變數 b 的地址果然不一樣。

試著寫個小程式玩玩看：

```c
#include <stdio.h>

int main(void) {
    int a, b, temp;
    int *p1, *p2;
    printf("請輸入 a 的值：");
    scanf("%d", &a);
    printf("請輸入 b 的值：");
    scanf("%d", &b);

    p1 = &a;
    p2 = &b;

    if(*p1 < *p2){
      temp = *p1;
      *p1 = *p2;
      *p2 = temp;
    }
    printf("*p1的值：%d\n", *p1);
    printf("*p2的值：%d\n", *p2);
}
```

讓使用者能自行輸入 a 和 b 變數的值，然後再把該變數賦值給 *p1 和 *p2 兩個 pointer。

最後比較一下，如果 *p1 值小於 *p2 值，就把兩數交換。

```
sixde-Air:iteration sixstockfeelair$ ./pointer
請輸入 a 的值：3
請輸入 b 的值：7
*p1的值：7
*p2的值：3
```

今天的指標與指標變數教學就到這邊，下一篇會來介紹指標與陣列的相互應用。感謝收看～

(來源：xkcd.com)

念金融的一聽到「指標」，就會反射性想到「市場指標」之類的折線圖或指數；結果現在看指標太久，下意識就想到一堆記憶體地址…

#兩種領域間的腦內打架 #Lynn編崩潰中

相關

形式語言：現代工程師可以用程式語言、不需要用紙帶打洞來寫程式都要感謝它

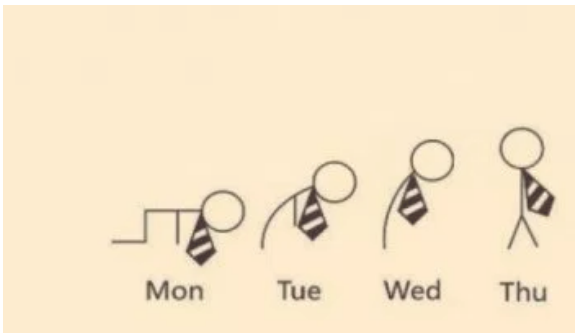
2017-06-20

在「科學」中

C語言入門: 宣告,賦值,PRINTF

2017-05-06

在「C/C++」中



C 語言：結構變數與指標

2017-05-30

在「C/C++」中

Posted in: C/C++, 程式 Tagged: C/C++, pointer, pointer variable, 指標, 指標變數, 記憶體

← 書評：打造超級 IP——新時代數位行銷的芭樂詞，看得我心都累了 你有聽過台灣納稅人的血淚WiMax嗎？中離王Intel與榮景不再的高通—4G 篇 →

4 THOUGHTS ON “C語言◘: 超好懂的指標，初學者請進～”

PALA 2017-10-03 at 15:07:34 回覆 →

看哈佛CS 50課程查資料查到這邊，感謝分享！

DARREN 2017-11-03 at 10:16:55 回覆 →

謝謝 寫得淺顯易懂，感謝您的分享

CHARLES 2017-11-04 at 11:59:17 回覆 →

這篇寫的好棒，淺顯易懂，謝謝分享 🙂

KEVIN 2017-11-12 at 12:06:33 回覆 →

這篇寫的超棒，非常好懂。

發表迴響

你的電子郵件位址並不會被公開。 必要欄位標記為 *

迴響

名稱 *

電子郵件 *

個人網站

張貼迴響

用電子郵件通知我後續的迴響。

新文章使用電子郵件通知我。

累計瀏覽人次 454,325
累計贊助人次 238
累計訂閱人次 943

寫點科普，請…
1.2 萬 按讚次數

- 【贊助這個寫作計畫】一杯咖啡的價格，能用來購買書籍圖庫、或之後的網站改版，以為讀者提供品質更加優良的文章與閱讀環境。 -

- 使用電子郵件訂閱 -

輸入EMAIL，即可訂閱網站的新文章，在發文時接收EMAIL通知。

輸入EMAIL

訂閱我吧

- 熱門文章 -

C語言: 超好懂的指標，初學者請進～

DRAM、NAND Flash 最近貴到炸，你還搞不懂記憶體的差異嗎？

晶圓代工爭霸戰：半導體知識（前傳）

C 語言：鏈結串列(Linked List)的建立與刪除

機器學習的機器是怎麼從資料中「學」到東西的？超簡單機器學習名詞入門篇！

2017 年深度知識專題總覽

一看就懂的 IC 產業結構與競爭關係

- 標籤雲 -

ARM C/C++ CAPM CDMA DRAM Google Hinton IBM Intel IPO Neural Network SEO SoC晶片 struct UI設計 三星 台積電 喬姆斯基 圖靈 圖靈機 基金 報酬 大數據 形式語言 指標 排序法 數位行銷 數本原 標準差 機器學習 深度學習 結構 聯發科 股票 芝諾悖論 處理器 記憶體 設計 邏輯電路 雲端運算 電晶體 類神經網路 風險 風險溢酬 高通