



2017

程式設計  
加強班

# 程式設計與實習(一)

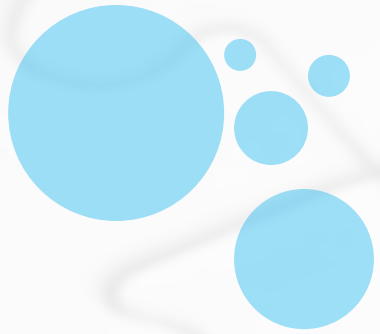
BY 孫茂勛

Email:JOHN85051232@GMAIL.COM



## 警告

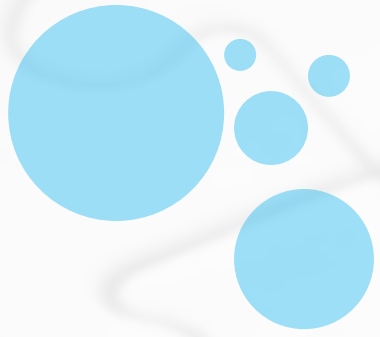
以下部份很抽象  
可說是C語言最難懂的部分  
回去請多練習、上網查資料或多詢問



# Pointer

學習指標前要知道的幾件事：

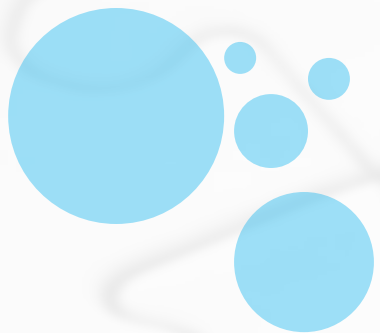
- ❖ 指標跟記憶體有關係
- ❖ 只要打錯整個程式就很容易當掉，所以不要緊張
- ❖ C 語言的特色之一



# 想想第一節加強班說過什麼

🔷 “變數宣告是跟電腦要一塊記憶體空間儲存資料”

🔷 現在我們要深入探討記憶體與資料之間的關係



# Pointer

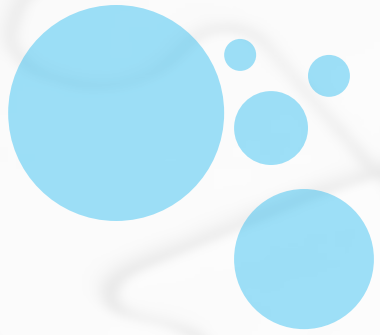
我們先來看一下變數的宣告：

```
int a = 10;
```

這一句對電腦代表甚麼意思呢？

a	
記憶體	0x000001
值	10

讓電腦生出一個4Byte大小的  
記憶體空間  
然後把10這個資料存進去



# Pointer

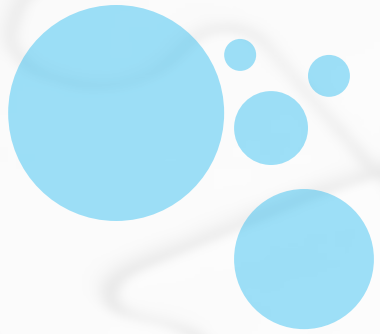
再來看一下變數的指定：

`a = 20;`

這一句對電腦代表甚麼意思呢？

記憶體	0x000001
值	20

讓電腦把0x000001這個位置的資料改成20



# Pointer

❖ 那，有沒有一種方法可以**直接把記憶體位址的資料進行更改**呢？

❖ 可以達成這種效果的就是指標(Pointer)

# 看一下變數的記憶體位置-1

📦 %p : 將資料以16進位的方式輸出

```
#include <stdio.h>
#include <stdlib.h>
/*pointer*/
```

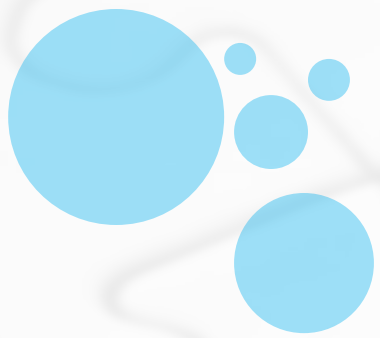
```
int main()
{
    int a = 10;
    float b = 1.5;
    printf( "%p\n", &a);
    printf( "%p\n", &b);
    return 0;
}
```

0043FD08

0043FCFC

請按任意鍵繼續 . . .



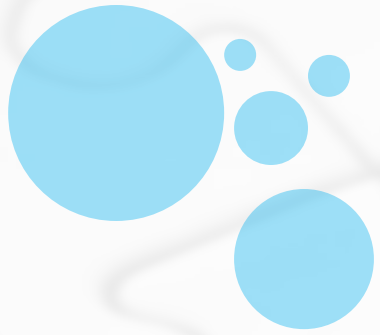


## 看一下變數的記憶體位置-2

🔍 看一下陣列每個元素的記憶體位址

```
#include <stdio.h>
#include <stdlib.h>
|
int main()
{
    int d[10] = {0};
    for(int i = 0 ; i < 10 ; ++i
        printf("%p\n",&d[i]);
    }
    return 0;
}
```

```
1A6FFD88
1A6FFD8C
1A6FFD90
1A6FFD94
1A6FFD98
1A6FFD9C
1A6FFDA0
1A6FFDA4
1A6FFDA8
1A6FFDAC
請按任意鍵繼續 . . .
```

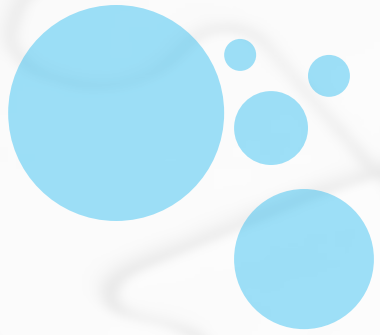


## 看一下變數的記憶體位置-3

- ◆ char array?
- ◆ float array?
- ◆ long long array?
- ◆ double array?

◆ 觀察上述情形的記憶體位址，有沒有發現位址是連續的

◆ 還記得之前說過陣列是跟電腦要一塊連續的記憶體空間嗎？

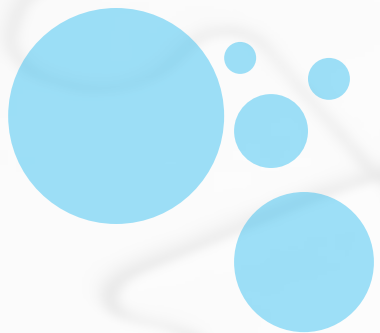


## 看一下變數的記憶體位置-4

int arr[10] = {0};

arr[0]的記憶體位址如果是0x000001

Q : arr[3]的記憶體位址是多少?



# Pointer

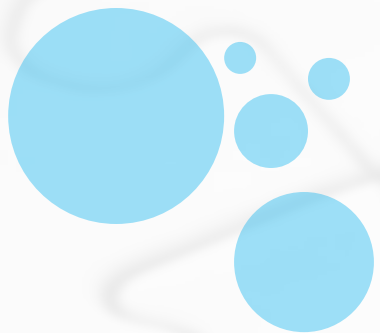
今天有一種變數可以儲存某個記憶體位置，叫做指標變數

Ex : `int *b = &a;` //宣告b是個指標變數，儲存(指向)a變數的記憶體位置

	a	b
記憶體	0x000001	0x000005
值	10	0x000001

變數b有自己的記憶體位址，但儲存的值是a的記憶體位址。

知道a變數的記憶體位址，我們就可以透過b變數去修改a的值



# Pointer

指標的資料型態：與變數相同，有int/float/char....

指標的宣告：

```
int i = 10;
```

```
int *ptr = &i;
```

或

```
int i = 10;
```

```
int *ptr = null;
```

...

```
ptr = &i;
```

Int i

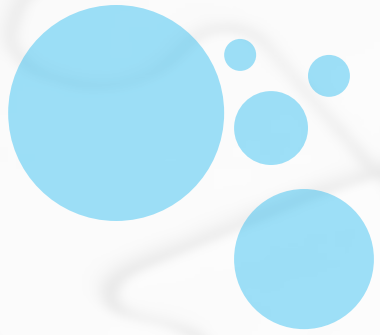
10

記憶體位置:0022FF54

Int \*ptr

0022FF54

記憶體位置:2622FF54

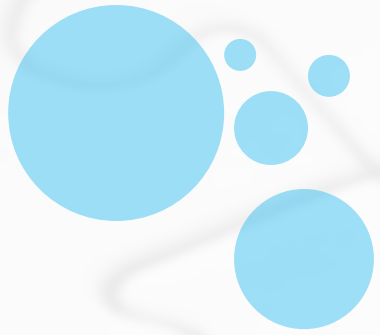


# Pointer

常見問題:

❖ `int* prt1, ptr2;`

❖ 這樣的宣告只有ptr1才是指標，ptr2是一個整數變數

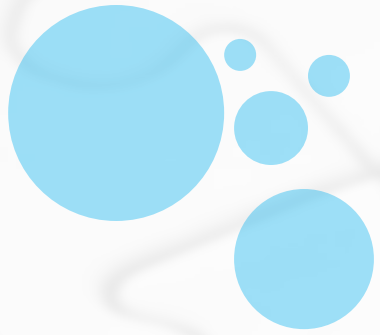


# Pointer

❖ 常見問題:

```
int *ptr;  
*ptr = 10; //error
```

- ❖ 沒有給 \*ptr 一個初始的位置，直接給值的話，通常會出現記憶體區段錯誤。
- ❖ 如果宣告完指標沒有要馬上指定變數的話，可以先設初始值為NULL，表示沒有指向任何變數。



# Pointer

Pointer (指標)中兩個重要的運算子：

◆取值運算子\*：用來取得指標變數的值。

◆取址運算子&：取出某個變數的記憶體位址。

◆Ex：scanf( "%d" ,&a); 是將資料寫入變數a的記憶體位址，所以前面要加&





# \*(取值)在宣告和使用時的意義不同

Ex:

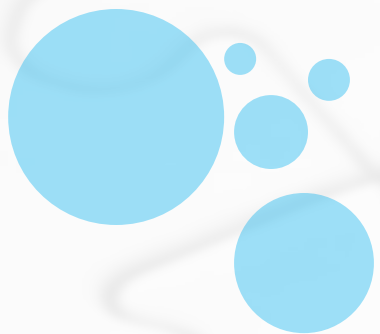
```
int b = 10;
```

```
int *ptr = &b;
```

```
printf( "%d\n" ,*ptr);
```

在宣告指標時的`int *ptr = &b;` 此時的`*`並不是取值的意思，而是跟電腦說這是一個指標變數

`printf`的`*`則是將`ptr`變數內存的記憶體位址的值取出



# Pointer

❖ 以下的程式碼執行結果？

```
int a = 10;  
int *ptr = &a; //宣告ptr是一個指標變數，指向a的記憶體位址  
printf( "%p\n", &*ptr);  
printf( "%p\n", *&ptr);
```

❖ 取值 ( \* ) 與取址 ( & ) 是互補的(可以互相抵銷)

期中考完後再來玩玩透過記憶體修改小遊戲





**THANK YOU**