



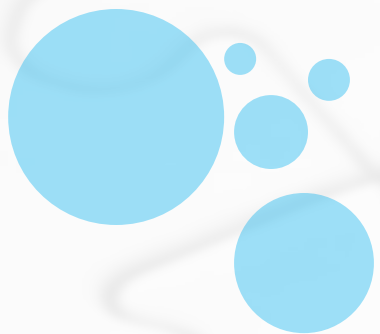
2017

程式設計
加強班

程式設計與實習(一)

BY 孫茂勛

Email:JOHN85051232@GMAIL.COM



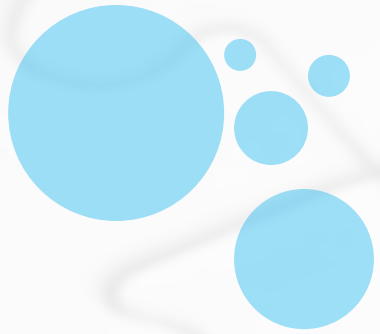
Pointer

- ❖ 陣列是一連串的記憶體空間。
- ❖ 且陣列名稱本身就是一個指向陣列開頭的位址。

```
int arr[5] = {1,2,3,4,5};  
printf("arr的位址 = %p \n",arr); //陣列名字是陣列起始的記憶體位址  
printf("arr[0]的位址 = %p \n",&arr[0]);
```

	0x0000	0x0004	0x0008	0x000C
Int arr[4]	Arr[0]	Arr[1]	Arr[2]	Arr[3]

所以也可以透過移動指標指向不同的位址來存取陣列



Pointer

透過移動指標指向不同的位址來存取陣列

指標移動：以一個資料型態(+1)為單位

❖ int型態的指標+1 = +4byte

❖ Char型態的指標+1 = +1byte

Int *Ptr = arr

Ptr

Ptr+1

Ptr+2

Ptr+3

0x0000

0x0004

0x0008

0x000C

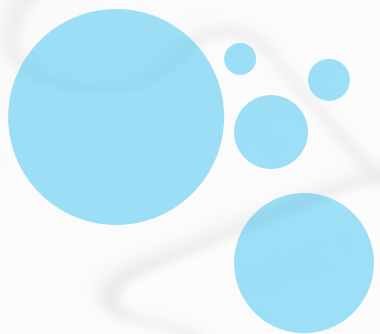
Int arr[4]

Arr[0]

Arr[1]

Arr[2]

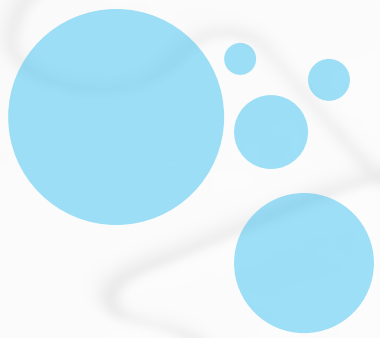
Arr[3]



Pointer

練習

```
int arr[5] = {1,2,3,4,5};  
int *ptr = arr; // ptr指向arr陣列的起始位址|  
printf("arr[0] = %d \n", arr[0]);  
printf("arr[0] = %d \n", *(ptr+0)); // 注意要先移動單位在作*運算  
  
printf("arr[1] = %d \n", arr[1]);  
printf("arr[1] = %d \n", *(ptr+1)); // 注意要先移動單位在作*運算
```



Pointer

練習 – 迴圈版本

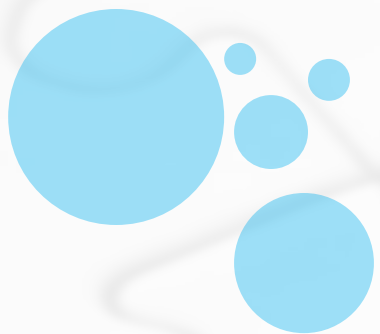
```
int arr[5] = {1,2,3,4,5};  
int *ptr = arr; // ptr 指向陣列 arr 的起始位址  
for(int i = 0 ; i < 5 ; ++i)  
{  
    printf("arr[%d] = %d\n", i, arr[i]);  
    printf("ptr 第 %d 的值 = %d \n", i, *(ptr+i))  
}
```

```
arr[0] = 1  
ptr 第 0 的值 = 1  
arr[1] = 2  
ptr 第 1 的值 = 2  
arr[2] = 3  
ptr 第 2 的值 = 3  
arr[3] = 4  
ptr 第 3 的值 = 4  
arr[4] = 5  
ptr 第 4 的值 = 5  
請按任意鍵繼續 . . .
```



參考資料

- ✦ http://edisonshih.pixnet.net/blog/post/27961535-01_%E5%88%9D%E5%AD%B8%E6%8C%87%E6%A8%99%E8%AB%8B%E9%80%B2---%E6%8C%87%E6%A8%99%E8%88%87%E4%BD%8D%E5%9D%80
- ✦ <http://openhome.cc/Gossip/CGossip/Pointer.html>



Const

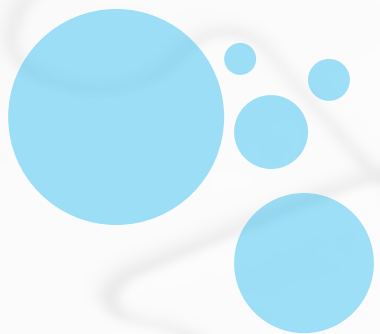
- ❖ 用來固定內容，使得無法被更改。

- ❖ 以int型態為例子，Ex：

```
const int var = 10; //固定變數var的值
```

```
var = 20; //error，因為加了const的變數是不能再被更改的
```

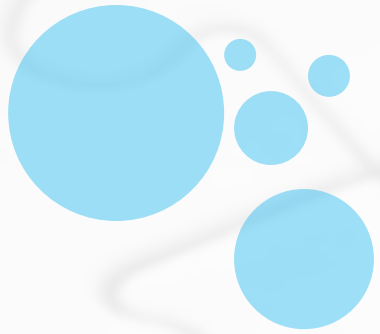
- ❖ Const用於指標可固定目標的值，或指向的記憶體位址。



Const

```
int a = 10;  
int b = 20;  
const int *ptr = &a;  
*ptr = 20; //error, const int的變數ptr無法更改內容  
ptr = &b; //const int的變數可以更改指向的記憶體位址
```

```
int a = 10;  
int b = 20;  
int *const ptr = &a;  
*ptr = 20; //int *const的變數ptr可以更改內容  
ptr = &b; //error, int *const的變數ptr無法更改指向的記憶體位址
```

Const

總結：

🔲 用於一般變數：

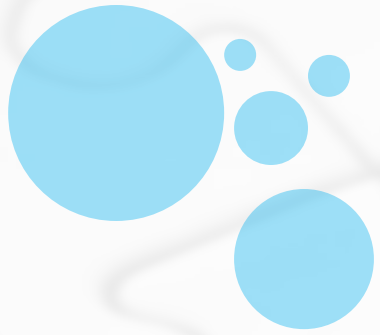
Const int：該變數的內容無法被更改

🔲 用於指標變數：

Const int*：該指標指向位址的內容無法被更改

Int *const：該指標指向的位址無法被更改

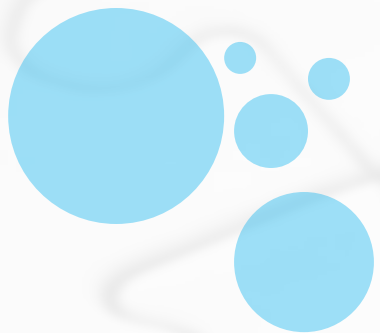
Q：Const int *const？



String

Q : Char只能印出一個字元

今天如果要印出很多個字元（一串文字）？



String

字串 (String)

陣列的一種：一維的字元陣列(Char Array)

宣告跟一維陣列一樣

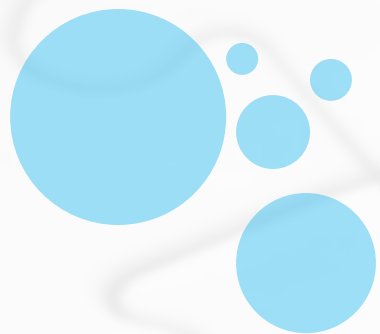
每個字串最後面都有一個 `'\0'` 的字元，代表字串結尾

Ex: `Char str[6] = "hello";`

C會自動加在結尾，所以記得宣告陣列大小一定要多保留一格給`'\0'`

str

Index	0	1	2	3	4	5
value	'h'	'e'	'l'	'l'	'o'	'\0'



宣告

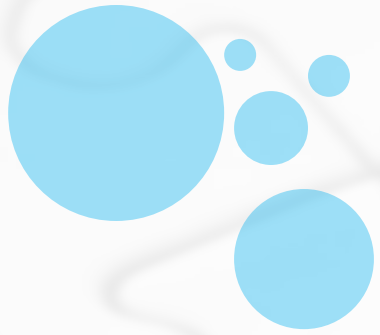
String

```
//宣告一個長度上限為20的字元陣列，並給定初始值
//1個char占一格陣列單位
char str[10] = "hello";
printf("%s\n",str); //字串用%s，參數給字串的記憶體位址開頭

char str1[5] = "hellohello"; //error, str1只能放5個字元，初始值長度超過

char str2[] = "hello"; //根據後面的長度配置陣列大小
printf("%s\n",str2);

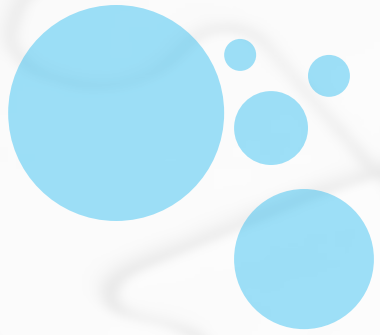
char *str3 = "hello"; //用指標的方式宣告字元陣列，此時會根據後面的長度配置陣列大小
printf("%s\n",str3);
```



String I/O

```
printf( "%s" ,str); //字元陣列，給陣列開始的記憶體位址  
scanf( "%s" ,str); //你也可以寫成&str[0]  
gets(str);  
puts(str);  
...
```

Q：只需要給予字串的开頭記憶體位址，why？

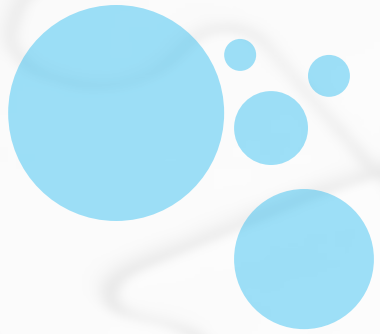


String

Char *str = "helloworld" ;

❖ 内容可以直接修改

```
char *a = "hello";  
printf("%s\n", a);  
a = "test";  
printf("%s\n", a);
```



String

Char str[] = "helloworld" ;

- ❖ 內容無法直接修改，只能一個字元一個字元做更改。
- ❖ 或是透過strcpy()修改

```
char a[] = "hello";  
printf("%s\n", a);  
a = "test";  
printf("%s\n", a);
```



THANK YOU