

Chapter8 陣列

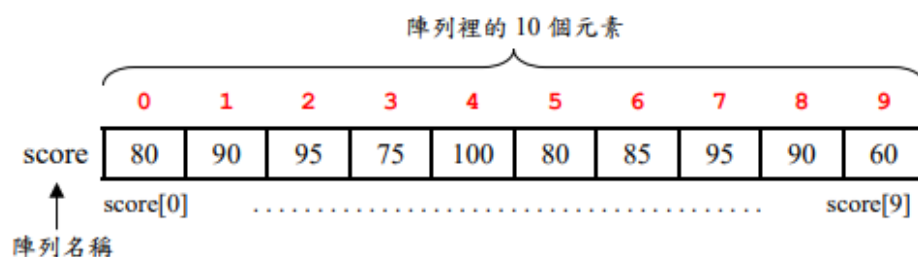
一、一維陣列

1. 宣告形式：

型態 陣列名稱[個數];

2. 範例：

```
int score[10] = {80,90,95,75,100,80,85,95,90,60};
```



每個元素佔一個 int 型態大小(4 bytes)。

sizeof(score) → 4 * 10 = 40 bytes

3. 初始值的設定方式：

(1) 將所有元素的值設成 1：

```
int arr_init1[3] = {1};
```

(2) 不指定大小，大小會依照後面元素個數來決定

```
int arr_init2[] = {1, 2, 3};
```

(3) 指定特定元素，其他未被指定元素會被設定 0 (此為 C99 新增)

```
int arr_init3[] = {[2] = 2};
```

(4) 初值個數少於宣告元素的個數十，剩餘的空間填 0

```
int arr_init4[10] = {1,2,3,4,5};
```

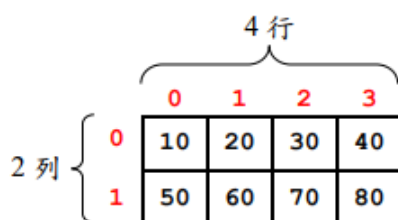
二、二維陣列

1. 宣告形式：

型態 陣列名稱[列數][行數]; (圖形化：直行橫列)

2. 範例：

```
int sale[2][4] = {{10,20,30,40}, {50,60,70,80}};
```



sale[0][0] → 10

sale[0][1] → 20

...

sale[1][0] → 50

sale[1][1] → 60

...

陣列中的元素個數 2*4=8

sizeof(sale) → 4*(2*4)=32bytes

3. 初值的設定方式：

```
int temp[][4] = { {1,2,3,4},{5,6,7,8},{9,10,11,12}};  
/* 未定長度之二維陣列的初值設定 */
```

多維陣列的宣告及初值設定方式依此類推。

三、陣列與函數

陣列可以當作引數傳遞到函數中，此時函數接收到的是陣列的「位址」，而非陣列的值。事實上，傳遞至函數中的也就是指向陣列位址的指標 (Pointer)。

```
#include <stdio.h>  
#define SIZE 5  
  
void print_matrix(int A[]); /* 函式原型 */  
  
int main(void)  
{  
    int data[SIZE] = {23,25,28,29,27};  
    printf_matrix(data);  
    return 0;  
}  
  
void print_matrix(int A[])  
{  
    int i;  
    for (i=0; i<SIZE; i++)  
        printf("%d ", A[i]);  
    printf("\n");  
    return;  
}
```

Memory

In main()

data

0	23	0253FDB8
1	25	
2	28	
3	29	
4	27	

In print_matrix()

A 為指標常數 0253FDB8