



# 一維陣列

• 宣告

```
int a[100];
int b[3]={1,2,3};
```

• 存取

```
int b[3]={1,2,3};
std::cout<<b[0]<<"\n";
std::cin>>b[1];
b[2]=b[2]*2;
```

b[0]	b[1]	b[2]		
1	2	3		



# 二維陣列

• a[3] a[2][3]

a[0] a[1] a[2]

	column1	column2	column3
row1	a[0][0]	a[0][1]	a[0][2]
row2	a[1][0]	a[1][1]	a[1][2]



```
int a[50][50];
int b[3][3]={{1,2,3},{4,5,6},{7,8,9}};
```

### • 存取

```
int b[3][3]={{1,2,3},{4,5,6},{7,8,9}};
std::cout<<b[0][1]<<"\n";
std::cin>>b[2][3];
```

```
#include<iostream>
 2 □ int main(){
        int a[3][5];
        //給值
        for(int i=0;i<3;i++){</pre>
             for(int j=0;j<5;j++){
 6 🖨
                 a[i][j]=i+j;
8
9
10
        //輸出陣列
        for(int i=0;i<3;i++){
11 🖨
             for(int j=0;j<5;j++){
12 🖨
                 std::cout<<a[i][j]<<' ';
13
14
15
             std::cout<<"\n";
16
17
        return 0;
18
10
```

0 1 2 3 4 1 2 3 4 5 2 3 4 5 6



# 多維陣列

- 如果在後面多加幾組中括號,可形成更高維度的陣列
- Ex: a [2][3][4][5]



# 延伸內容



### 記憶體配置

- 一維陣列在記憶體中是一大塊連續的記憶體
- 形如 a [ m ] [ n ] 的二維陣列可以想像成是有m個有n個元素的一維陣列,在記憶體中的先擺放完一個再放另一個,如圖

a[0][	a[0][1]	•••	a[0][n]	a[1][0]	•••	a[m][n-1]	a[m][n]
-------	---------	-----	---------	---------	-----	-----------	---------

## 記憶體配置

- 同理,三維的陣列可以想像成多個二維陣列的結合,在記憶體中先擺完一個二維陣列再放另一個,如圖
- 以此類推...

C	1[0][0]	a[0][0][1]	• • •	a[0][0][n]	a[0][1][0]	•••	a[0][m][n-1]	a[0][m][n]
	a[1][0][0	] a[1][0][1]	]	a[1][0][n]	] a[1][1][0	]	a[1][m][n-1	] a[1][m][n]