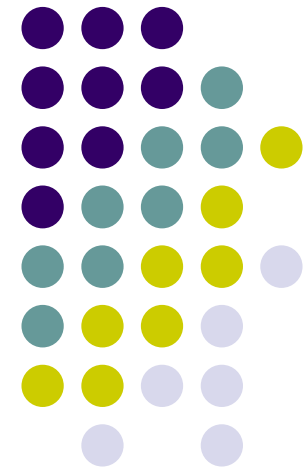


# 第六章 陣列

認識陣列與一般資料型態的不同

認識一維與二維陣列

學習陣列的應用





# 一維陣列

- 一維陣列（1-dimensional array）可以存放多個相同資料型態的資料。
- 使用陣列必須經過兩個步驟：
  - (1) 宣告陣列
  - (2) 配置記憶體給該陣列
- 一維陣列的宣告與配置記憶體格式：

## 一維陣列的宣告與配置記憶體

```
資料型態 陣列名稱[];           // 宣告一維陣列  
陣列名稱 = new 資料型態[個數]; // 配置記憶體給陣列
```



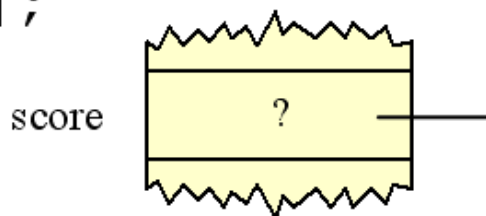
# 一維陣列的宣告及使用 (1/3)

- 下面的範例是一維陣列的宣告及記憶體配置：

```
01  int score[];           // 宣告整數陣列 score
02  score=new int[4];       // 配置可存放 4 個整數的記憶體空間，以供陣列 score 使用
```

- 執行完第1行後，編譯器會配置一塊記憶體給它：

```
int score[];
```

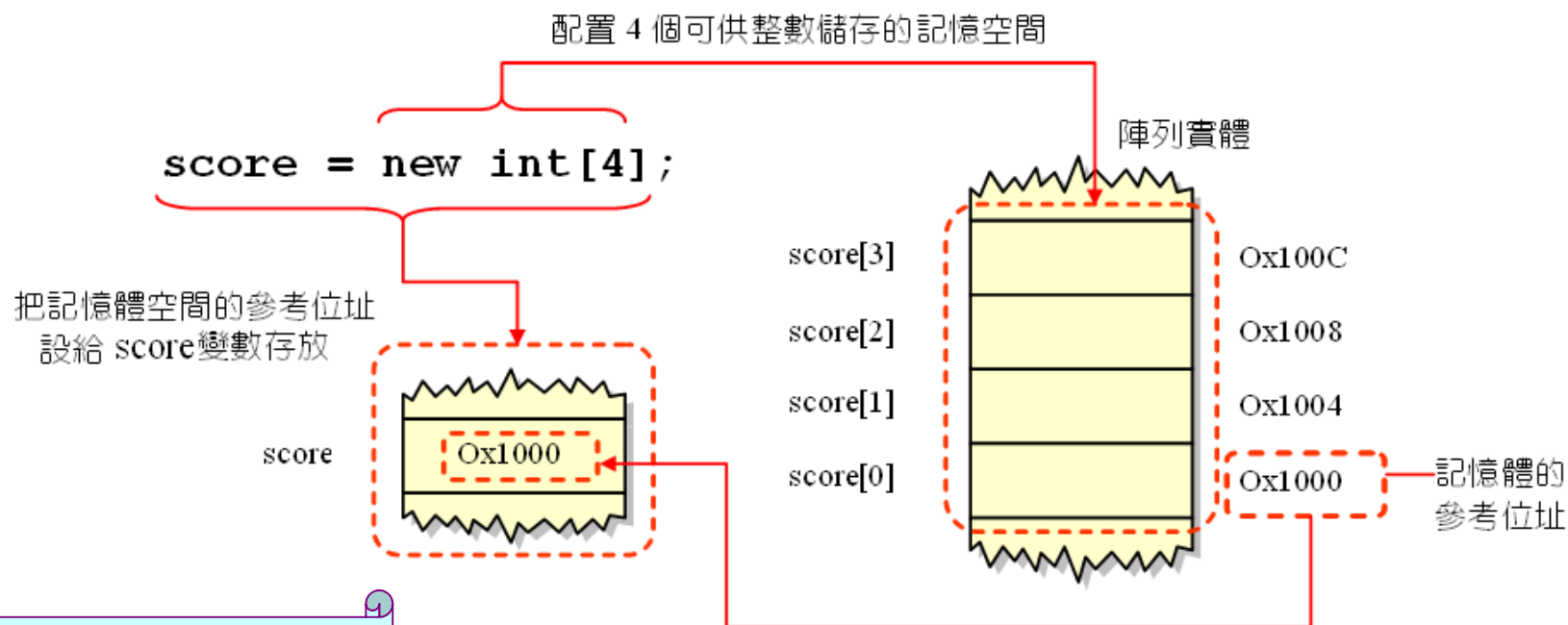


score 尚未指向陣列實體的位址，所以 score 的內容為未知



# 一維陣列的宣告及使用 (2/3)

- 第2行是記憶體配置的動作：



陣列是屬於非基本資料型態，因此score儲存的是陣列實體的參考位址



# 一維陣列的宣告及使用 (3/3)

- 宣告一維陣列的另一種寫法：

宣告陣列的同時便配置記憶體

資料型態 陣列名稱[] = **new** 資料型態 [個數];

- 一維陣列的宣告範例：

```
int score[] = new int[4];
```

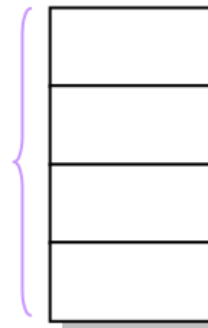
宣告一個整數陣列score，同時配置一塊可存放4個整數的連續記憶體空間

```
int score[] = new int[4];
```

將陣列score化為圖形表示

陣列裡元素的個數

score ← 陣列名稱



每一格代表一個元素，每個元素皆可存放整數型態的數值



# 陣列的另一種宣告方式

- 稍早是以下列語法宣告score陣列：

```
int score[];           // 宣告score陣列為整數型態
```

- 還可以用另一種語法宣告：

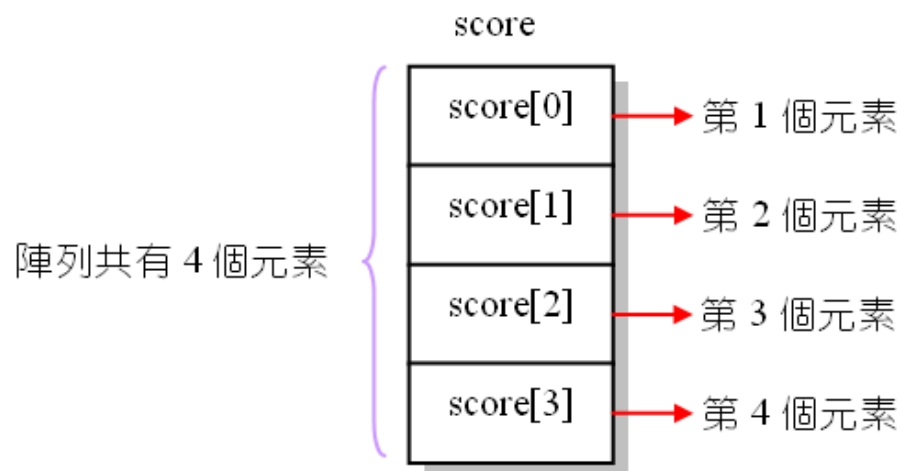
```
int[] score;           // 宣告score變數，其型態為整數陣列
```



# 陣列元素的表示方法

- 要存取陣列裡的元素，可以利用索引值（index）
- 陣列索引值的編號是由0開始
- 下圖為score陣列中元素的表示法及排列方式：

```
int score[]=new int[4];
```





# 一維陣列的使用範例

- 下面的程式是一維陣列的使用範例：

```
01 // app6_1, 一維陣列
02 public class app6_1
03 {
04     public static void main(String args[])
05     {
06         int i;
07         int a[];          // 宣告整數陣列 a
08         a=new int[3];      // 配置可存放 3 個整數的記憶體空間供整數陣列 a 使用
09         a[0]=9;            // 設定第一個元素的值為 9
10         a[1]=6;            // 設定第二個元素的值為 6
11
12         for(i=0; i<a.length; i++)          // 印出陣列的內容
13             System.out.print("a["+i+"]="+a[i]+",\t");
14
15         System.out.println("\nLength of array = "+a.length); // 印出陣列長度
16     }
17 }
```

```
/* app6_1 OUTPUT-----
a[0]=9, a[1]=6, a[2]=0,
Length of array = 3
-----*/
```





# 陣列的長度

- 取得陣列元素的個數（陣列長度）的格式：

陣列長度的取得

陣列名稱.length

- 如下面的程式片段：

```
a.length           // 印出陣列的長度
```



# 陣列初值的設定 (1/2)

- 在宣告時就給與陣列初值的格式：

## 陣列初值的設定

資料型態 陣列名稱[]={初值1，初值2，...，初值n}；

- 以上面的格式宣告時，會視初值的個數來決定陣列的長度，如下面的範例：

```
int day[]={31,28,31,30,31,30,31,31,30,31,30,31};    // 宣告並設定初值
```

陣列元素有12個，day[0]為31，  
day[1]為28，...，day[11]為31



## 陣列初值的設定 (2/2)

- app6\_2是一維陣列設定初值的範例：

```
01 // app6_2, 一維陣列的設值
02 public class app6_2
03 {
04     public static void main(String args[])
05     {
06         int sum=0;
07         int a[]={62,7,12,3,8,47}; // 宣告整數陣列 a, 並設定初值
08
09         for(int i=0;i<a.length;i++) // 計算陣列元素的和
10             sum+=a[i];
11
12         System.out.println("Average "+(float)sum/a.length);
13     }
14 }
```

**/\* app6\_2 OUTPUT----**  
Average 23.166666  
**-----\*/**



## 簡單的範例

- 下面的程式可以找出陣列裡元素的最大值及最小值：

```
01 // app6_3, 比較陣列元素值的大小
02 public class app6_3
03 {
04     public static void main(String args[])
05     {
06         int i,min,max;
07         int a[]={74,48,30,17,62}; // 宣告整數陣列 a,並設定初值
08
09         min=max=a[0];
10         System.out.print("Elements in array a are ");
11         for(i=0;i<a.length;i++)
12         {
13             System.out.print(a[i]+" ");
14             if(a[i]>max) // find Maximum
15                 max=a[i];
16             if(a[i]<min) // find Minimum
17                 min=a[i];
18         }
19         System.out.println("\nMaximum is "+max); // 印出最大值
20         System.out.println("Minimum is "+min); // 印出最小值
21     }
22 }
```

**/\* app6\_3 OUTPUT-----**  
Elements in array a are 74 48 30 17 62  
Maximum is 74  
Minimum is 17  
**-----\*/**

# 二維陣列的宣告

## 6.2 二維陣列



- 二維陣列的宣告與配置記憶體空間的格式：

### 二維陣列的宣告格式

```
資料型態 陣列名稱[][];  
陣列名稱=new 資料型態[列的個數][行的個數];
```

- 如下面的範例：

```
int score[][];          // 宣告整數陣列 score  
score=new int[4][3];    // 配置一塊記憶體空間，以供 4 列 3 行的整數陣列 score 使用
```

- 以較為簡潔的方式來宣告陣列：

### 二維陣列的宣告格式

```
資料型態 陣列名稱[][]=new 資料型態[列的個數][行的個數];
```

- 下面是二維陣列的宣告範例：

```
int score[][]=new int[4][3]; // 宣告整數陣列 score，同時配置一塊記憶體空間
```



## 二維陣列的實例 (1/2)

- 下表為某汽車銷售公司的車輛銷售量：

業務員	2009 年銷售量			
	第一季	第二季	第三季	第四季
1	32	35	26	30
2	34	30	33	31

陣列 sale

	第 1 行	第 2 行	第 3 行	第 4 行
第 1 列	(0,0) 32	(0,1) 35	(0,2) 26	(0,3) 30
第 2 列	(1,0) 34	(1,1) 30	(1,2) 33	(1,3) 31

每一格代表一個元素，每個元素皆為 `int` 型態

- 上面的資料可用二維陣列儲存，宣告方式為

```
int sale[2][4];
```

```
// 宣告一個 2 列 4 行的整數陣列 sale
```



## 二維陣列的實例 (2/2)

- 二維陣列的宣告與配置記憶空間的格式：

### 二維陣列初值的設定格式

資料型態 陣列名稱 [ ] [ ] = { { 第1列初值 },  
{ 第2列初值 },  
{ ... },  
{ 第n列初值 } } ;

```
int sale[][]={{32,35,26,30},  
              {34,30,33,31}};
```

2×4的陣列是由2個具有4個  
元素的一維陣列所組成

```
int sale[2][4]={{32,35,26,30},{34,30,33,31}};
```

2×4的陣列

一維陣列，  
有4個元素

一維陣列，  
有4個元素



## 每列的元素個數不同的二維陣列

- `matx[]`為每列元素個數不同的二維陣列：

```
int matx[][]={{31,12,14,11},          // 每列元素個數不同的二維陣列
               {33,34,30},
               {12,81,32,14,17}};
```

- 宣告每列元素個數不同的二維陣列，但不設定初值：

```
int matx[][]=new int[3][];           // 宣告二維陣列，並指定列數
matx[0] = new int[4];                 // 指定第一列有 4 個元素
matx[1] = new int[3];                 // 指定第二列有 3 個元素
matx[2] = new int[5];                 // 指定第三列有 5 個元素
```





# 取得列數與特定列之元素的個數

- 取得二維陣列的列數與特定列之元素的個數語法

取得二維陣列的列數與特定列之元素的個數

陣列名稱.length	// 取得陣列的列數
陣列名稱[列的索引值].length	// 取得特定列元素的個數

- 如下面的程式片段：

matx.length	// 取得陣列 matx 的列數，其值為 3
matx[0].length	// 取得陣列 matx 的第 1 列元素的個數，其值為 4
matx[2].length	// 取得陣列 matx 的第 3 列元素的個數，其值為 5



# 二維陣列元素的引用及存取

- 下面是二維陣列的完整使用範例：

```
01 // app6_4, 二維陣列的輸入輸出
02 public class app6_4
03 {
04     public static void main(String args[])
05     {
06         int i,j,sum=0;
07         int sale[][]={{32,35,26,30},{34,30,33,31}}; // 宣告陣列並設定初值
08
09         for(i=0;i<sale.length;i++) // 輸出銷售量並計算總銷售量
10         {
11             System.out.print("業務員"+(i+1)+"的業績分別為 ");
12             for(j=0;j<sale[i].length;j++)
13             {
14                 System.out.print(sale[i][j]+" ");
15                 sum+=sale[i][j];
16             }
17             System.out.println(); // 列印換行
18         }
19         System.out.println("\n 總銷售量為"+sum+"部車");
20     }
21 }
```

```
/* app6_4 OUTPUT-----
業務員 1 的業績分別為 32 35 26 30
業務員 2 的業績分別為 34 30 33 31

總銷售量為 251 部車
-----*/
```



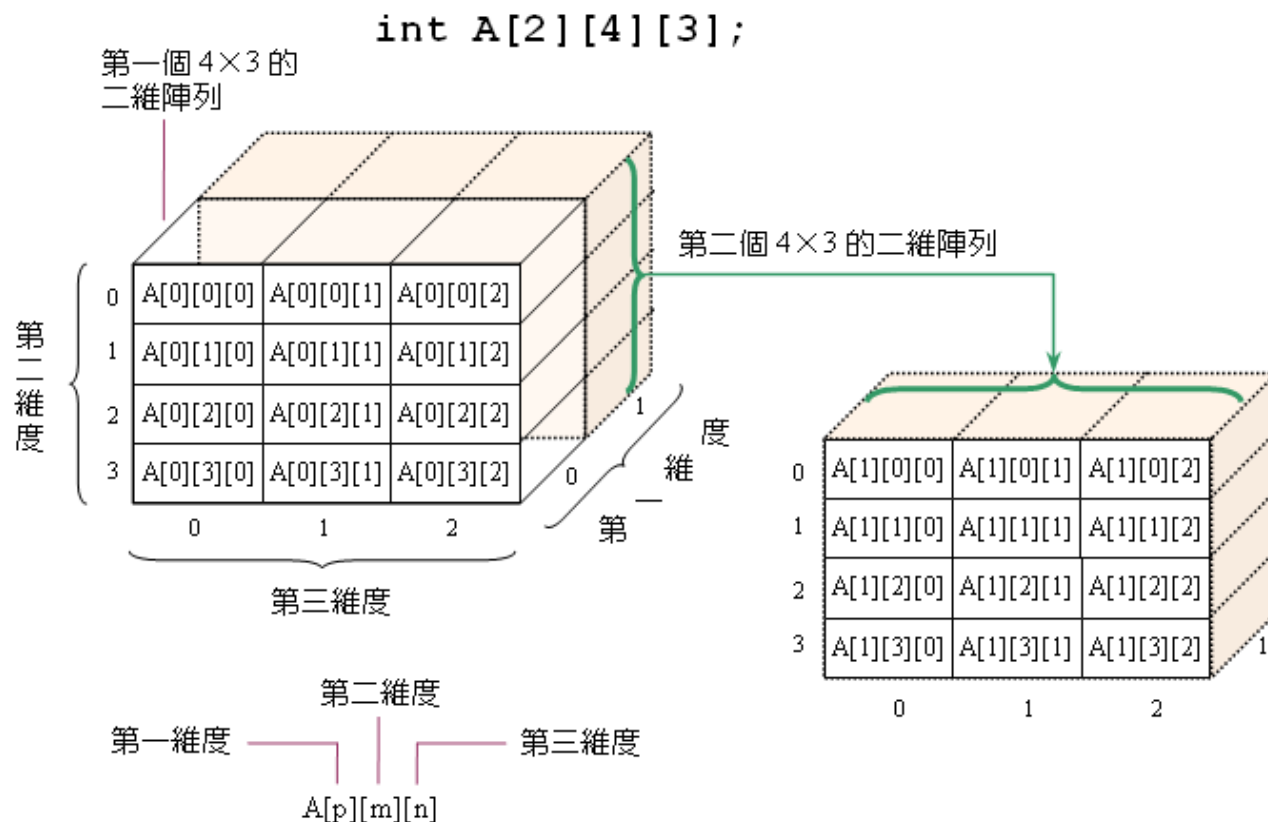
# 三維陣列

- 三維陣列的宣告範例：

```
int A[2][4][3]; // 宣告 2×4×3 整數陣列 A
```

2×4×3的三維陣列可看成是由2個4×3的二維陣列所組成

也就是兩組4個橫列，3個直行的積木併在一起，組成一個立方體





# 找出最大值的範例 (1/3)

```

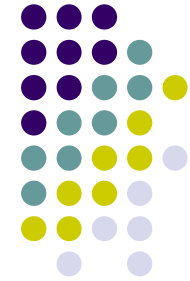
01 // app6_5,
02 public class app6_5
03 {
04     public static void main(String args[])
05     {
06         int A[][][]={{21,32,65},
07                       {78,94,76},
08                       {79,44,65},
09                       {89,54,73}},
10         {{32,56,89},
11          {43,23,32},
12          {32,56,78},
13          {94,78,45}}};
14
15         int i,j,k,max=A[0][0][0]; // 設定 max 為 A 陣列的第一個元素
16
17         for(i=0;i<A.length;i++) // 外層迴圈
18             for(j=0;j<A[i].length;j++) // 中層迴圈
19                 for(k=0;k<A[i][j].length;k++) // 內層迴圈
20                     if(max<A[i][j][k])
21                         max=A[i][j][k];
22
23         System.out.println("max="+max); // 印出陣列的最大值
24     }
25 }

```

**/\* app6\_5 OUTPUT---**  
max=94  
**-----\*/**

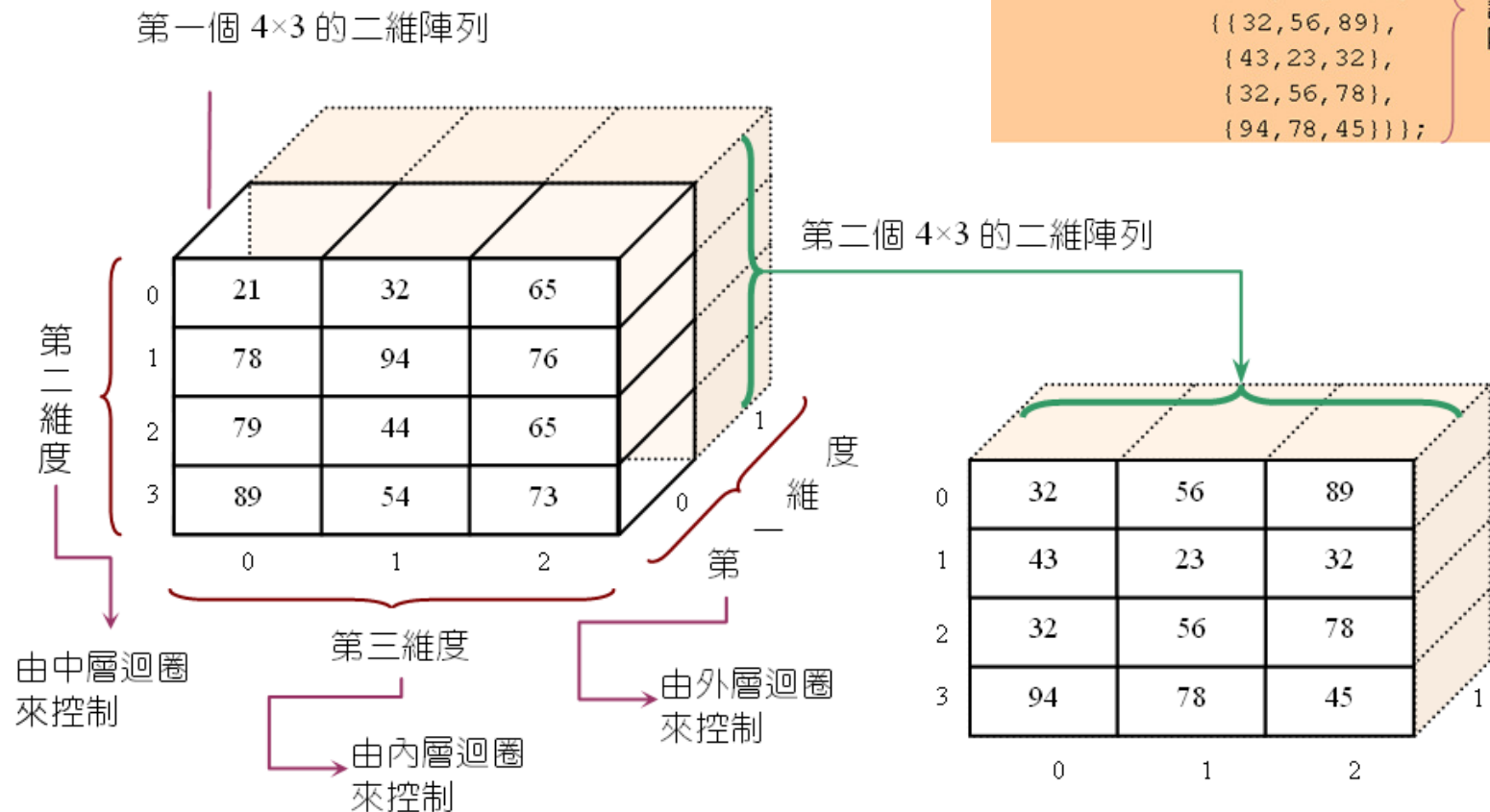
設定 2×4×3  
陣列的初值

利用三個 for 迴圈找出陣列的最大值



# 找出最大值的範例 (2/3)

- app6\_5中三維陣列A的示意圖：



```
int A[][][]={{ {21,32,65},
                {78,94,76},
                {79,44,65},
                {89,54,73}},
              {{32,56,89},
                {43,23,32},
                {32,56,78},
                {94,78,45}}};
```

設定 2×4×3 陣列的初值



## 找出最大值 (3/3)

- $2 \times 4 \times 3$  的三維陣列可以寫成

$2 \times 4 \times 3$  的三維陣列 = {  $4 \times 3$  的二維陣列 ,  $4 \times 3$  的二維陣列 }

- 陣列A初值的設定便可用下圖來表示：

