第六章
陣 列

本章學習目標

- 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

6.1 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。

6.1.1 一維陣列的宣告與記憶體的配置

要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：

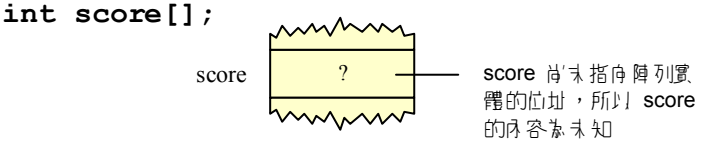
格式 6.1.1
一維陣列的宣告與配置記憶體

```
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
```

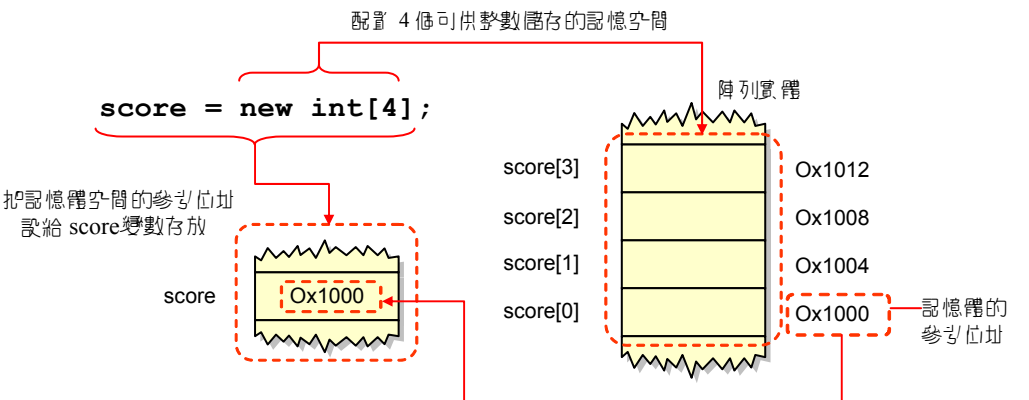
下面的範例說明一維陣列宣告的方式，以及如何配置記憶體給它：

```
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
```

執行完第一行後，編譯器會配置一塊記憶體給它，用來儲存指向陣列實體的位址，如下圖所示：



第二行的記憶體配置動作可用下圖來表示：



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。

格式 6.1.1 可以用較簡潔的方式來表示：

```
資料型態 陣列名稱[]=new 資料型態[個數];
```

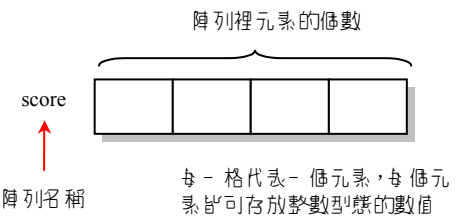
格式 6.1.2
宣告陣列的同時便配置記憶體

下面是宣告整數陣列 score，並配置記憶空間的範例：

```
int score[]=new int[4];    /* 宣告一個整數陣列 score，同時配置
                              一塊可存放 4 個整數的記憶體空間，
                              以供該陣列使用 */
```

將陣列 score 化為圖形表示：

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



6.1.2 陣列的宣告 – 標準寫法

Java 也允許我們把 score 陣列宣告成

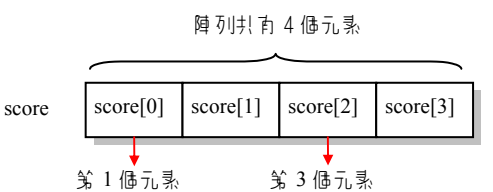
```
int[] score;   // 宣告 score 變數，其型態為整數陣列
```

6.1.3 陣列元素的表示方法

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];
```



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]=5;           // 設定第一個元素的值為5
10        a[1]=8;           // 設定第二個元素的值為8
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]=5, a[1]=8, a[2]=0,
Length of array = 3
----------------------*/
```

取得数组长度的格式：

格式 6.1.3
数组长度的取得

```
数组名称.length
```

如下面的程序片段：

```
a.length    // 即为数组的长度
```

6.1.4 陣列初值的設定

在宣告時就給與陣列初值的格式：

```
資料型態 陣列名稱[]={初值 1，初值 2，…，初值 n};
```

格式 6.1.4
陣列初值的設定

用此格式來宣告陣列並設定初值時，不需要將陣列元素的個數列出，如下面的範例：

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

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[]={15,6,8,7,12,7};      // 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 9.166667
--------------------*/
```

6.1.5 簡單的範例：找出陣列元素的最大值與最小值

下面的程式碼說明如何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)       // 判斷最大值
15                max=a[i];
16           if(a[i]<min)       // 判斷最小值
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
----------------------------------------*/
```

6.2 二維陣列

6.2.1 二維陣列的宣告與配置記憶體

二維陣列的宣告與配置記憶空間的格式：

```
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];
```

格式 6.2.1
二維陣列的宣告格式

如下面的範例：

```
int score[][];
score=new int[4][3];
```

以較為簡潔的方式來宣告陣列：

格式 6.2.2
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];
```

以上述的寫法，在宣告的同時即配置一塊記憶體空間，以供該陣列使用：

```
int score[][]=new int[4][3];
```

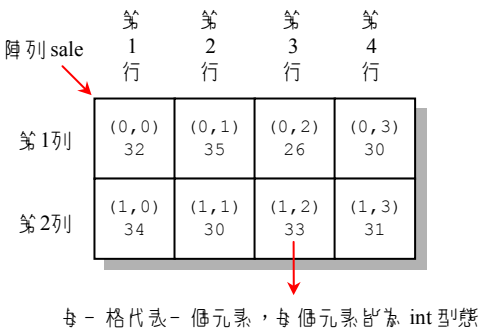
下表為某汽車銷售公司有兩個業務員，於 2005 年的每輛銷售量：

業務員	2005 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

上面的資料可用一個 2 列 4 行的二維陣列來儲存，也就是把陣列宣告為

```
int sale[2][4];      // 宣告一個 2 列 4 行的整數陣列 sale
```

把 sale 陣列化為圖形表示：



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 }};
```

格式 6.2.3
二維陣列初值的設定格式

在宣告時就設定陣列的初值，如下面的範例：

```
int sale[][]={{30,35,26,32},          // 二維陣列的初值設定
              {33,34,30,29}};
```

每列的元素個數不同的二維陣列

二維陣列中每列的元素的個數可以不相同，如下面的範例：

```
int matx[][]={{31,12,14,11},
              {33,34,30},
              {12,81,32,14,17}};
```

想先宣告每列元素不相等之二維陣列，可以參考下面的寫法：

```
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
```

取得二維陣列的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之語法如下：

```
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
```

格式 6.2.4
取得二維陣列的列數與特定列之元素的個數

如下面的程式片段：

```
matx.length
matx[0].length
matx[2].length
```

6.2.2 二維陣列元素的引用及存取

下面的範例可列印出兩個業務員的銷售業績，並計算其輛的總銷售量：

```
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
------------------------*/
```

6.3 多維陣列

下面的範例說明如何在三維陣列裡，找出所有元素的最大值：

```
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];
```

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

```
22
23          System.out.println("max="+max);    // 印出陣列的最大值
24      }
25  }
```

```
/* app6_5 OUTPUT---
max=94
--------------------*/
```

-The End-