

# Java

## 二維陣列

# 這是一維陣列

若將大樓內不論住在哪一樓的所有住戶都依序放到一個陣列裡，在此陣列內已經無法分辨裡面住戶是住在哪一層樓時，就將這個如同一條線性空間的陣列稱為**一維陣列**。



01	02	03	04	05	06	07	08	09	10
----	----	----	----	----	----	----	----	----	----

as[0] as[1] as[2] as[3] as[4] as[5] as[6] as[7] as[8] as[9]

# 一維陣列的缺點

- 一維陣列只能提供一個連續空間存放大量資料。
- 當資料有空間性時，一維陣列無法明確傳達資料的空間性。
- 例如：
  - 大樓住戶無法用一維陣列紀錄那些住戶住在哪一樓。
  - 用一維陣列紀錄的表格資料無法傳達表格內的行與列特性。

改善一維陣列的缺點 ⇒ **二維陣列**

# 二維陣列

- 一維陣列就像是只有 **1 列** 的表格。

--	--	--	--	--	--

- **二維陣列**就是有多列的表格。


# 二維陣列結構

使用二維陣列時，就不再侷限於線性空間，而是改用平面空間儲存，如此一來，就可以保留住戶的居住樓層資訊。



07	08	09	10
03	04	05	06
01	02		

# 宣告二維陣列

## 一維陣列變數宣告

```
// 第一種方法:資料型態 陣列名稱[];  
int as1[];  
long bs1[];  
  
// 第二種方法:資料型態[] 陣列名稱;  
int[] as2;  
long[] bs2;
```

宣告二維陣列的方式與一維陣列非常相似，只要在 [] 後面再加一個 [] 就可以了！

## 二維陣列變數宣告

```
// 第一種方法:資料型態 陣列名稱[][];  
int as1[][];  
long bs1[][];  
  
// 第二種方法:資料型態[][] 陣列名稱;  
int[][] as2;  
long[][] bs2;  
  
// 第三種方法:資料型態[] 陣列名稱[];  
int[] as2[];  
long[] bs2[];
```

# 定義二維陣列：三種方式

- 用初始值方式定義

```
int as1[][] = new int[][]{{11, -2}, {-3}, {24, 5}};
```

- 一次決定列與行數量

```
int as2[][] = new int[3][4];
```

- 先決定列，再逐一決定行

```
int as3[][] = new int[3][];  
as3[0] = new int[2];  
as3[1] = new int[5];  
as3[2] = new int[3];
```

一維陣列與二維陣列的定義方式同樣是相似的，只要在行數的定義外加上列數的定義，就可以成功定義二維陣列了！



# 取得二維陣列元素值



```
import java.util.Arrays;  
class web{  
    public static void main(String[] args) {
```

一維  
陣列

```
int as1[] = new int[]{1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  
System.out.println(as1[5]);
```

二維  
陣列

```
int as2[][] = new int[][]{{1, 2}, {3, 4, 5, 6}, {7, 8, 9, 10}};  
System.out.println(as2[1][3]);
```

[二樓][第四間]  
的住戶

變數名稱[第幾列][第幾行]

一維陣列是線性的，  
所以只要提供一個元  
素索引值，就可以存  
取對應的陣列元素。

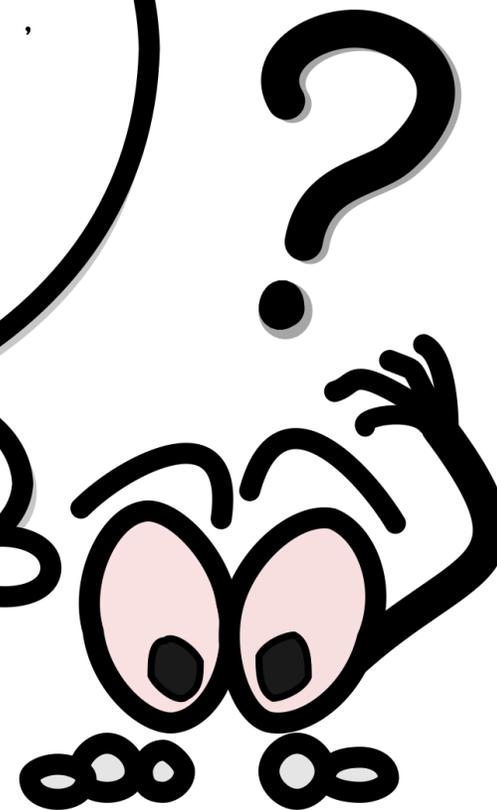
二維陣列有區分  
列跟行，所以需要  
提供兩個元素  
索引值分別對應  
列跟行。

# 用二維陣列解決問題

請用二維陣列記錄一個如圖所示的表格資料，  
表格共有五列，第1列有1行，第2列有2行，  
第3列有3行，第4列有4行，第5列有5行，  
每一格裡面的值會記錄到目前為止有幾格。

1				
2	3			
4	5	6		
7	8	9	10	
11	12	13	14	15

問題



# 二維陣列解決問題：程式碼

二維陣列類似表格的結構，所以二維陣列的存取常常會搭配巢狀迴圈使用喔！

```
import java.util.Arrays;
class web{
    public static void main(String[] args) {
        int as[][] = new int[5][];
        int sum = 1;
        for (int i=0; i<5; i++) {
            as[i] = new int[i+1];
            for (int j=0; j<=i; j++) {
                as[i][j] = sum;
                sum++;
            }
        }
        for (int i=0; i<5; i++) {
            System.out.println(Arrays.toString(as[i]));
        }
    }
}
```

因為前面陣列定義時只有先定義列數，所以在要設定各個陣列元素值之前別忘在要在定義行數喔！

執行結果

```
[1]
[2, 3]
[4, 5, 6]
[7, 8, 9, 10]
[11, 12, 13, 14, 15]
```

# JAVA

## 延伸的概念

# 概念1：二維陣列的 for (1)

```
import java.util.Arrays;  
class web{  
    public static void main(String[] args) {  
        int as1[] = new int[]{11, -2, -3, 24, 5};  
  
        for (int i : as1) {  
            System.out.println(i);  
        }  
    }  
}
```

## for (變數i : 陣列A)

將陣列A內的每個陣列元素依序放到變數i內，透過變數i就可以依序取得陣列A內包含的所有元素值。



還記得一維陣列  
有一個特殊的  
for 迴圈結構嗎？

執行結果

```
11  
-2  
-3  
24  
5
```

# 概念1：二維陣列的 for (2)

```
import java.util.Arrays;
class web{
    public static void main(String[] args) {
        int as[][] = new int[5][];
        int sum = 1;
        for (int i=0; i<5; i++) {
            as[i] = new int[i+1];
            for (int j=0; j<=i; j++) {
                as[i][j] = sum;
                sum++;
            }
        }
        for (int[] iArr : as) {
            for (int z : iArr) {
                System.out.printf("%d ", z);
            }
            System.out.println("");
        }
    }
}
```



二維陣列同樣可以使用特殊的 for 迴圈，但要注意二維陣列的外層迴圈是取**每一列**，每一列本身還是一維陣列結構，所以二維陣列的外層迴圈變數要使用**一維陣列變數**去紀錄。

執行結果

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```