

# Java

## 一維陣列

## 先思考一個問題

要如何從鍵盤輸入5個整數，  
然後印出輸入的第4個整數？

問題



程式碼如下：

```
import java.util.Scanner;
class web{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int a, b, c, d, e;
        a = sc.nextInt();
        b = sc.nextInt();
        c = sc.nextInt();
        d = sc.nextInt();
        e = sc.nextInt();
        System.out.println(d);
    }
}
```



## 再思考一個問題

要如何從鍵盤輸入100個整數，  
然後印出指定的輸入整數，  
如第44個整數？

問題



# 變數宣告完都累了

```
import java.util.Scanner;
class web{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int a1, a2, a3, a4, a5, a6, a7, a8, a9, a10;
        int a11, a12, a13, a14, a15, a16, a17, a18, a19, a20;
        int a21, a22, a23, a24, a25, a26, a27, a28, a29, a30;
        int a31, a32, a33, a34, a35, a36, a37, a38, a39, a40;
        int a41, a42, a43, a44, a45, a46, a47, a48, a49, a50;
        int a51, a52, a53, a54, a55, a56, a57, a58, a59, a60;
        int a61, a62, a63, a64, a65, a66, a67, a68, a69, a70;
        int a71, a72, a73, a74, a75, a76, a77, a78, a79, a80;
        int a81, a82, a83, a84, a85, a86, a87, a88, a89, a90;
        int a91, a92, a93, a94, a95, a96, a97, a98, a99, a100;
    }
}
```



如果是輸入10000個

我猜光是指定變數就煩死你....

問題



# 使用 陣列 解決

```
import java.util.Scanner;
class web{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int nums[];
        nums = new int[100];
        for (int i=0; i<100; i++) {
            nums[i] = sc.nextInt();
        }
        System.out.println(nums[44]);
    }
}
```

定義陣列變數

宣告陣列變數

指派陣列變數內容

輸出陣列變數  
指定內容



陣列的操作手冊



**JAVA**



什麼是陣列



如何宣告陣  
列



如何定義陣  
列



如何用陣列  
解決問題？



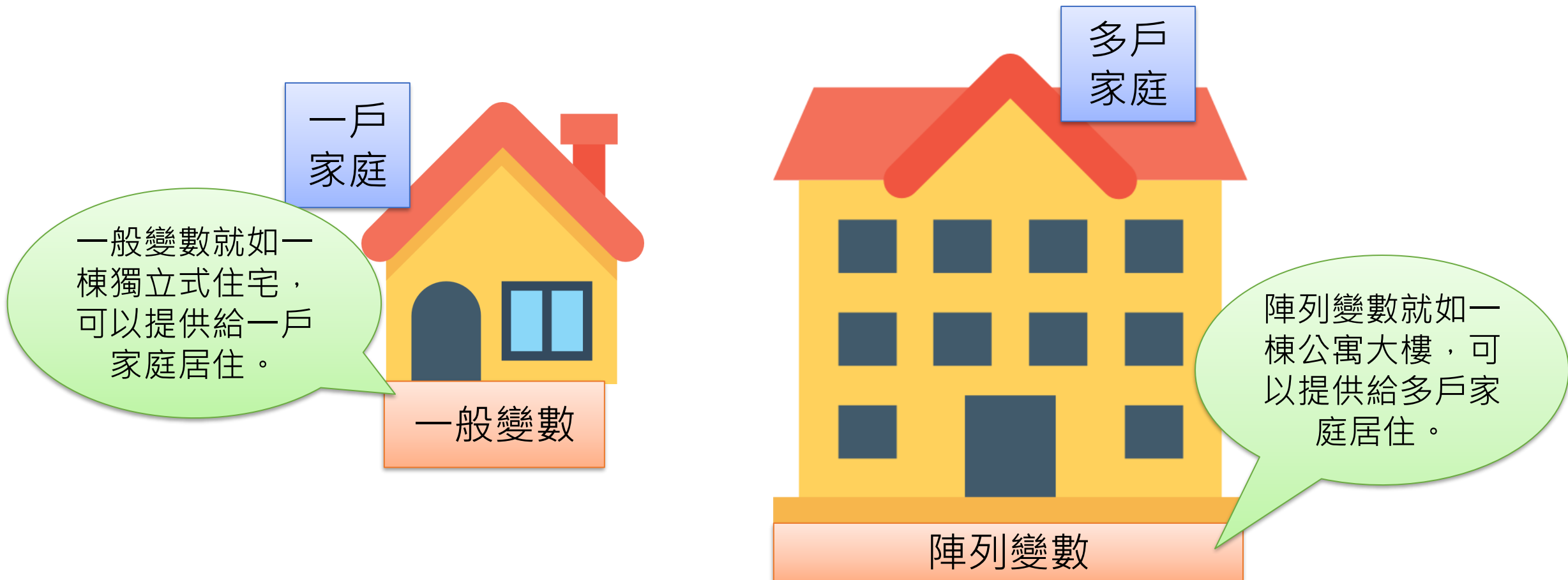
如何指派陣  
列內容



如何取得陣  
列元素值

# 陣列是什麼：存放大量資料

- 陣列是可以存放大量相同資料型態的組合空間。



# 陣列是什麼：存放**相同資料型態**

- 陣列是可以存放**大量相同資料型態**的組合空間。



一棟公寓會有多戶家庭，這些家庭會有相同的某樣性質，可能是在這附近工作的家庭，或是在這附近就學的學生群。

陣列變數也存在相似的情況，一個陣列變數內只會存放有同樣資料型態的資料！

# 陣列是什麼：包含多個**陣列元素**的集合

- 陣列內的每個空間稱為**陣列元素**。
- 一個一般變數可以看作是一個陣列元素，陣列就是這些陣列元素的大門，也就是這些元素的管理中心。



大樓裡面的住戶都要透過這個大門進出。

陣列裡面的元素都要透過陣列變數存取。

什麼是陣列



如何宣告陣  
列



如何定義陣  
列



如何用陣列  
解決問題？



如何指派陣  
列內容



如何取得陣  
列元素值

# 宣告陣列：先決定資料型態

- 前面介紹陣列時有提過，陣列是儲存相同資料型態，因此在宣告陣列時，要先決定要使用哪種資料型態！
- 陣列可以使用的基本資料型態與一般變數可以使用的基本資料型態一樣，共有 8 種。

資料型態	名稱
<b>byte</b>	位元整數
<b>short</b>	短整數
<b>int</b>	整數
<b>long</b>	長整數
<b>float</b>	單精度浮點數
<b>double</b>	雙精度浮點數
<b>char</b>	字元
<b>boolean</b>	布林

# 宣告陣列：再依照指定格式宣告

## 一般變數宣告

```
int a;  
long b;
```

## 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;
```

宣告陣列與宣告一般變數相當相似。只要宣告時在資料型態後面加上[]，或是在變數名稱後面加上[]就可以宣告陣列變數了！

什麼是陣列



如何宣告陣  
列



如何定義陣  
列



如何用陣列  
解決問題？



如何指派陣  
列內容



如何取得陣  
列元素值



# 定義陣列：需要設定空間大小

一般變數宣告後就可以立刻指派值

```
int a = 5;
```

```
int b;
```

```
b = 4;
```

陣列變數？

- 陣列如公寓大樓一樣，是存放大量相同資料型態的組合空間
- 公寓大樓可大可小，在蓋大樓前要先決定大樓要提供給幾戶人家居住，才能畫出設計圖。
- 陣列的組合空間同樣也是可大可小，在指派值之前要先決定需要多少空間放資料，也就是可以存放多少個陣列元素。

# 定義陣列：兩種空間設定方法(1)

- 第一種：設定陣列元素個數

定義空間時，前面一定要加上 **new**，**new** 的作用類似於跟 Java 系統說：請給我新的此資料型態的空間。

```
int as1[];  
as1 = new int[3];
```

格式

**new** 資料型態[元素個數]

資料型態後面接中括號，並在中括號內指定元素個數 N，就可以產生一個可以容納 N 個此資料型態的陣列空間。

# 定義陣列：兩種空間設定方法(2)

- 第二種：將陣列要存放的資料直接指派給陣列

不論是哪種  
定義方式，  
前面一定要  
加上 **new**。

```
int as2[];  
as2 = new int[]{1, 2, 3};
```

格式

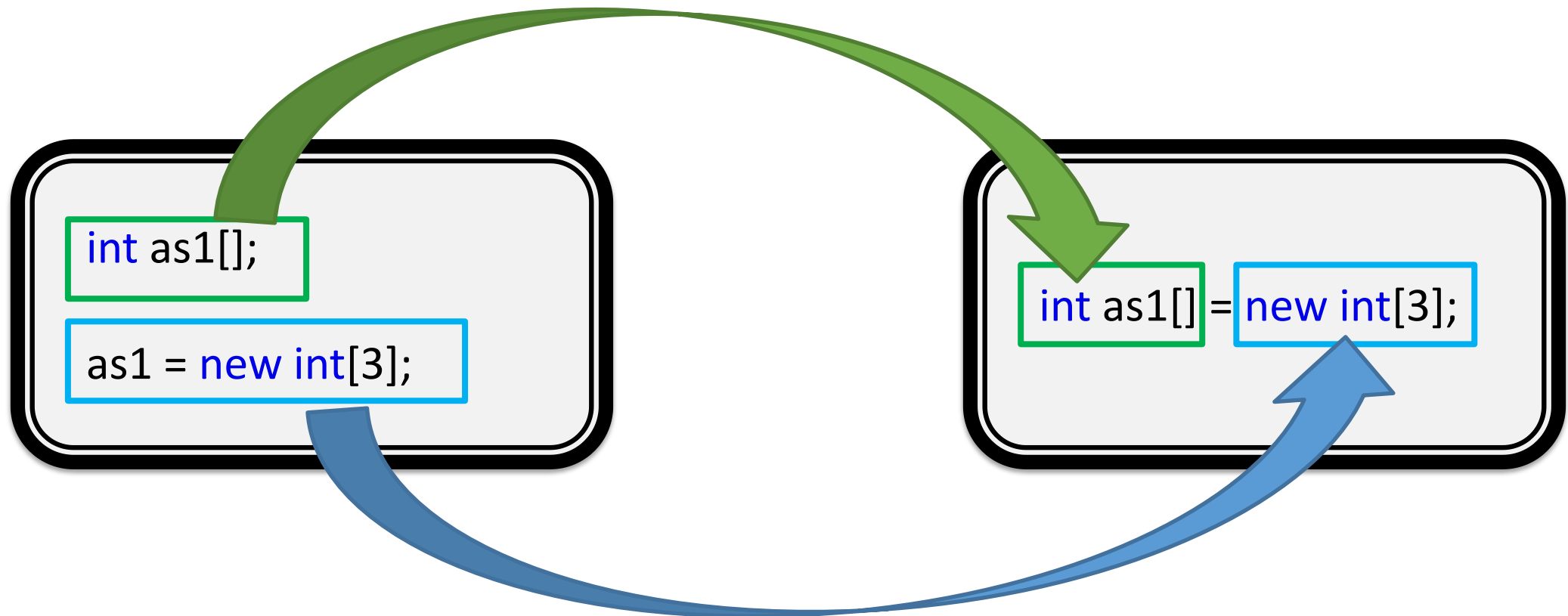
**new** 資料型態[] {陣列內容}

當陣列已經明確知道要放那些資料時，可以直接將資料用大括號包住並接在中括號後面。

使用這種方式時，中括號內就不用填入陣列元素個數，Java 可以自動根據大括號內設定的數量決定陣列元素個數。

# 定義陣列：宣告跟定義變一行(1)

- 定義陣列可以在宣告陣列變數時就一併處理。
  - 第一種：設定陣列元素個數



# 定義陣列：宣告跟定義變一行(2)

- 定義陣列可以在宣告陣列變數時就一併處理。
  - 第二種：將陣列要存放的資料直接指派給陣列

```
int as2[];
```

```
as2 = new int[]{1, 2, 3};
```

第二種定義方式  
有兩種方法可以  
同時宣告與定義  
陣列變數

```
int as2[] = new int[]{1, 2, 3};
```

```
int as2[] = {1, 2, 3};
```

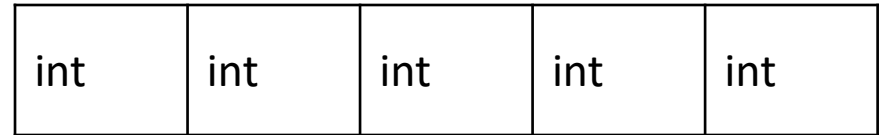
宣告陣列時若已經知道要放那些資料，就可以直接在宣告的變數後面放入要指派的值，就不用加上 **new** 資料型態[]

# 定義陣列：產生可以存放指定數量的空間

```
int as1[] = new int[5];
```

產生一個可以存放連續5個整數資料的空間

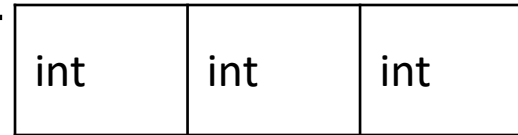
as1



```
int as2[] = new int[]{1, 2, 3};
```

產生一個可以存放連續3個整數資料的空間

as2



什麼是陣列



如何宣告陣  
列



如何定義陣  
列



如何用陣列  
解決問題？



如何指派陣  
列內容



如何取得陣  
列元素值

# 取得陣列元素值

- 大樓可以透過門牌號碼找到各個住戶
- 陣列也可以透過編號找到各個陣列元素





# 取得陣列元素值：編號

每個陣列元素都有自己的編號，一個陣列裡面的陣列元素編號會從 **0** 開始，依序遞增。這個元素編號就稱為**索引值**。

```
int as1[] = new int[5];
```



**as1**

<b>[0]</b>	<b>[1]</b>	<b>[2]</b>	<b>[3]</b>	<b>[4]</b>
int	int	int	int	int
<b>as[0]</b>	<b>as[1]</b>	<b>as[2]</b>	<b>as[3]</b>	<b>as[4]</b>

每個陣列元素可以透過陣列名稱[索引值]的方式存取對應的陣列元素。

# 取得陣列元素值：程式碼

記得陣列元素編號是從 0 開始，跟我們一般從 1 開始編號的習慣不一樣喔！

```
class web{  
    public static void main(String[] args) {  
        int as1[] = new int[]{6, 2, 3};  
  
        System.out.println(as1[0]);  
        System.out.println(as1[1]);  
        System.out.println(as1[2]);  
    }  
}
```

執行結果

6  
2  
3

什麼是陣列



如何宣告陣  
列



如何定義陣  
列



如何用陣列  
解決問題？



如何指派陣  
列內容



如何取得陣  
列元素值

# 指派陣列內容：兩種方式

- 第一種：在定義陣列變數時就給值

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

- 第二種：使用陣列元素索引值動態給值

```
陣列名稱[索引值] = 值定值;
```

# 指派陣列內容

透過陣列名稱[索引值]，再加上一個變數的指派方式，就可以動態的更改指定的陣列元素值。

```
class web{  
    public static void main(String[] args) {  
        // 第一種：在定義陣列變數時就給值  
        int as1[] = new int[]{6, 2, 3};  
        // 第二種：使用陣列元素索引值動態給值  
        int as2[] = new int[5];  
        as2[0] = 2;  
        as2[1] = 1;  
        as2[2] = 33;  
        as2[3] = 12;  
        as2[4] = 5;  
        System.out.println(as2[0]);  
        System.out.println(as2[2]);  
        System.out.println(as2[3]);  
    }  
}
```

執行結果

2  
33  
12

什麼是陣列



如何宣告陣  
列



如何定義陣  
列



如何用陣列  
解決問題？



如何指派陣  
列內容



如何取得陣  
列元素值

## 先回顧前面的問題

要如何從鍵盤輸入100個整數，  
然後印出指定的輸入整數，  
如第44個整數？

問題



# 用陣列解決問題

只要記得：

1. 定義變數時加上 []
  2. 多一個宣告變數的步驟
  3. 根據索引值存取陣列元素
- 記得這三個差異點後，陣列變數就可以像一般變數使用了！



```
import java.util.Scanner;
class web{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int nums[];
        nums = new int[100];
        for (int i=0; i<100; i++) {
            nums[i] = sc.nextInt();
        }
        System.out.println(nums[44]);
    }
}
```

定義陣列變數

宣告陣列變數

指派陣列變數內容

輸出陣列變數  
指定內容





## 用陣列解決另一個問題

要如何從鍵盤輸入100個整數，  
然後將其反向輸出？

問題



# 還是用陣列解決問題

```
import java.util.Scanner;
class web{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int i, num = 100;
        int as[];
        as = new int[100];
        for (i = 0; i < num; i++) {
            as[i] = sc.nextInt();
        }
        i = num - 1;
        while (i >= 0) {
            System.out.printf("as[%d]: %d\n", i, as[i]);
            i--;
        }
    }
}
```



陣列最大的特質就是會存放大量資料，因此陣列大多會搭配迴圈結構，才可一一存取每個陣列元素。

執行結果

```
as[99]: 100
as[98]: 99
as[97]: 98
as[96]: 97
as[95]: 96
as[94]: 95
```

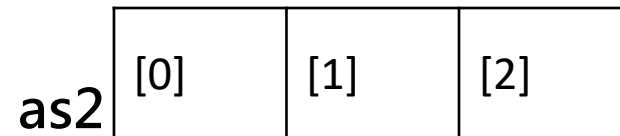
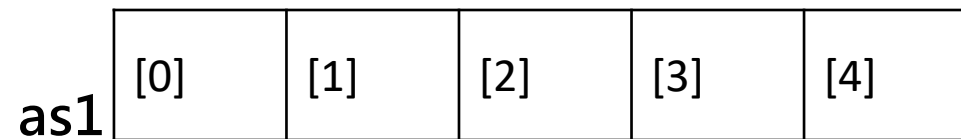
陣列的工具箱

**JAVA**



# 陣列有問題？

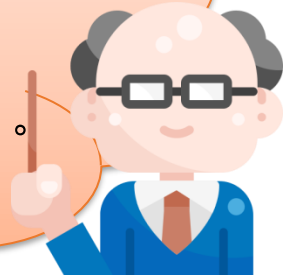
- 陣列使用時，因為存放大量資料，所以常常會有各種困擾：
  - 這個陣列到底可以放多少陣列元素？
  - 如果是整數陣列，能不能讓陣列自動排序，數字由小到大呢？
  - 有沒有簡單的方法，讓陣列元素都填上統一的值呢？
  - 要怎麼快速地知道這兩個陣列是不是一樣呢？
  - 能不能一次性的顯示整個陣列內容？



# 陣列小工具：.length

```
class web{  
    public static void main(String[] args) {  
        int as[] = new int[]{3, 11, 5, 1, 2};  
        int size = as.length;  
  
        for (int i=0; i<size; i++) {  
            System.out.println(as[i]);  
        }  
    }  
}
```

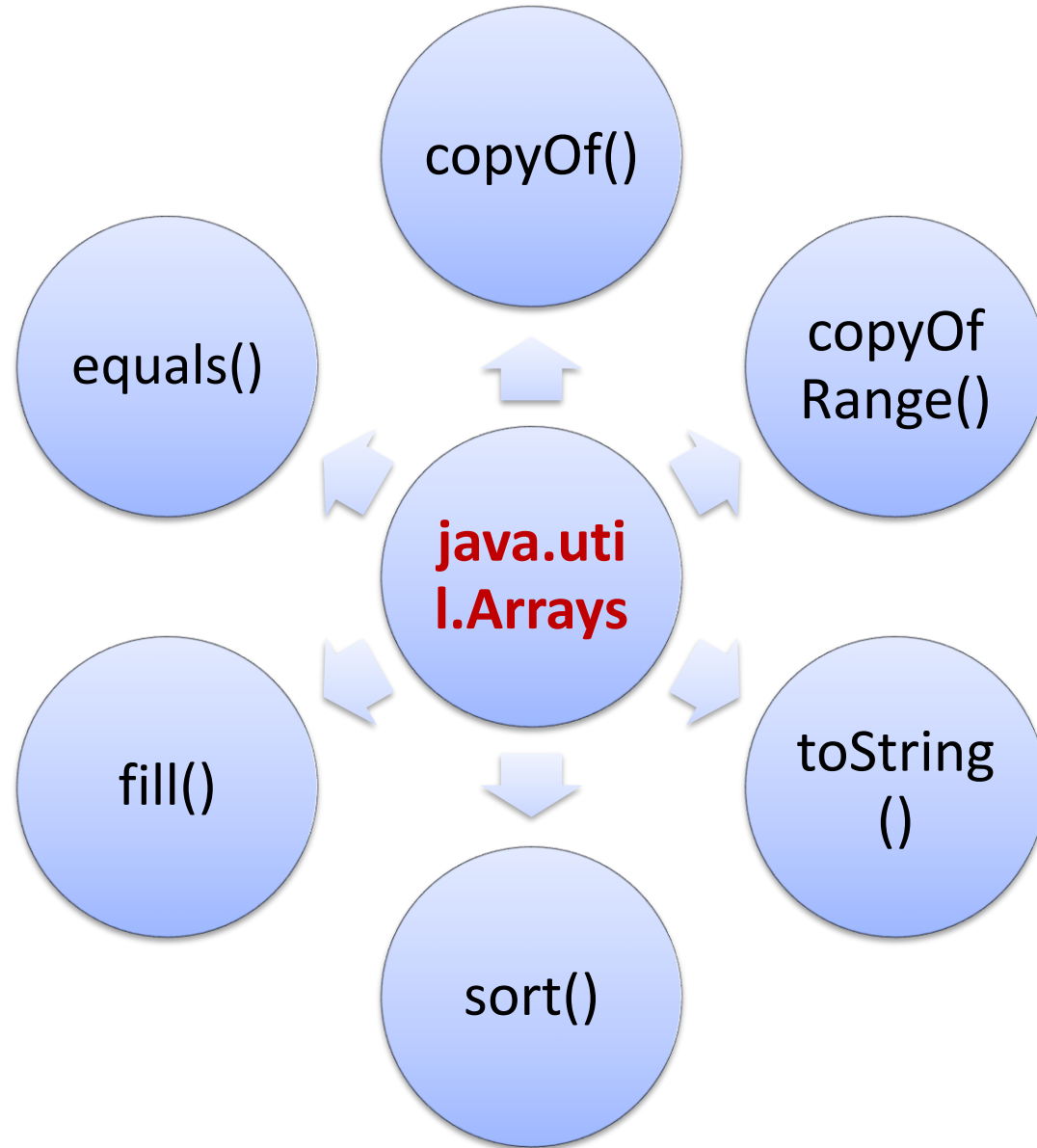
陣列.length 可以知道這個陣列的元素數量，就不需要特定去記得每個陣列的長度。



執行結果

```
3  
11  
5  
1  
2
```

不只有小工具，甚至有個工具箱：`java.util.Arrays`



Java 有特別準備一個工具箱，用來輔助陣列的操作！

# 陣列工具箱 (1)

**Array.copyOf(陣列A, 元素個數N)**

複製陣列A的前N個陣列元素到另一個陣列。

**Array.copyOfRange(陣列A, 索引值x, 索引值y)**

複製A[x]到A[y]內的所有陣列元素(不包含A[y])到另一個陣列。

```
import java.util.Arrays;
class web{
    public static void main(String[] args) {
        int as1[] = new int[]{3, 11, 5, 1, 2};
        int as2[] = Arrays.copyOf(as1, as1.length-1);
        int as3[] = Arrays.copyOfRange(as1, 1, as1.length-2);

        System.out.println(Arrays.toString(as1));
        System.out.println(Arrays.toString(as2));
        System.out.println(Arrays.toString(as3));
    }
}
```

# 陣列工具箱 (2)

## Array.toString (陣列A)

用逗點將陣列A內的所有元素串接在一起產生一個字串資料，並在字串前後用中括號包住。

```
import java.util.Arrays;
class web{
    public static void main(String[] args) {
        int as1[] = new int[]{3, 11, 5, 1, 2};
        int as2[] = Arrays.copyOf(as1, as1.length-1);
        int as3[] = Arrays.copyOfRange(as1, 1, as1.length-2);

        System.out.println(Arrays.toString(as1));
        System.out.println(Arrays.toString(as2));
        System.out.println(Arrays.toString(as3));
    }
}
```

執行結果

```
[3, 11, 5, 1, 2]
[3, 11, 5, 1]
[11, 5]
```



# 陣列工具箱 (3)

```
import java.util.Arrays;
class web{
    public static void main(String[] args) {
        int as1[] = new int[]{3, 11, 5, 1, 2};
        int as2[] = new int[]{3, 1, 5, 1, 2};

        System.out.printf("is as1 equals to as2: %b\n", Arrays.equals(as1, as2));
        as2[1] = 11;
        System.out.printf("is as1 equals to as2: %b\n", Arrays.equals(as1, as2));
    }
}
```

## Array.equals (陣列A, 陣列B)

判斷 陣列A 與 陣列B 內容是否一樣。

Arrays.equals(as1, as2);

Arrays.equals(as1, as2);

執行結果

```
is as1 equals to as2: false
is as1 equals to as2: true
```

# 陣列工具箱 (4)

```
import java.util.Arrays;
class web{
    public static void main(String[] args) {
        int as1[] = new int[]{1, 2, 3, 4, 5};

        System.out.println(Arrays.toString(as1));

        Arrays.fill(as1, 10);

        System.out.println(Arrays.toString(as1));
    }
}
```

**Array.fill (陣列A, 指派值x)**

將 陣列A 內的每個陣列元素都設定成 x。

執行結果

```
[1, 2, 3, 4, 5]
[10, 10, 10, 10, 10]
```

# 陣列工具箱 (5)

```
import java.util.Arrays;
class web{
    public static void main(String[] args) {
        int as1[] = new int[]{11, -2, -3, 24, 5, 2, -10};

        System.out.println(Arrays.toString(as1));
        Arrays.sort(as1, 0, 2);
        System.out.println(Arrays.toString(as1));
        Arrays.sort(as1, 3, 5);
        System.out.println(Arrays.toString(as1));
        Arrays.sort(as1);
        System.out.println(Arrays.toString(as1));
    }
}
```

**Array.sort (陣列A,索引值x, 索引值y)**

A[x]到A[y]內的所有陣列元素(不包含A[y])由小到大排序。

若不指定 x 與 y，就會將整個陣列內的陣列元素由小到大排序

執行結果

```
[11, -2, -3, 24, 5, 2, -10]
[-2, 11, -3, 24, 5, 2, -10]
[-2, 11, -3, 5, 24, 2, -10]
[-10, -3, -2, 2, 5, 11, 24]
```

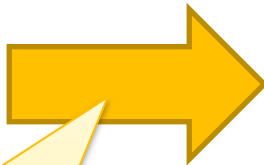
# JAVA

## 延伸的概念

# 概念1：特殊的 for 迴圈(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=0; i<as1.length; i++) {
            System.out.printf("%d, ", as1[i]);
        }
    }
}
```



```
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.printf("%d, ", i);
        }
    }
}
```

遇到資料是陣列時，可以將原始的 for 迴圈結構簡化成：  
**for (變數i : 陣列A)。**

# 概念1：特殊的 for 迴圈(2)

```
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。

執行結果

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