

¿Qué es un array?

Un array es una estructura que nos permite almacenar una lista de datos. Es como si tuviéramos una lista de variables, todas del mismo tipo.

No puedo tener en un array mezclados datos de tipo String con tipo Integer. Todos los datos que yo tengo en un array son siempre del mismo tipo.

¿Cómo se declara un array para utilizarlo luego?

Al igual que las variables, un array tiene que ser declarado. Se hace utilizando los corchetes []

Sería así:

```
TipoDeDato[] nombreArray;
```

Por ejemplo, si quiero declarar un array para almacenar String y que se llame "listaPalabras", lo haría así:

```
String[] listaPalabras;
```

Para poder empezar a utilizar el array es obligatorio inicializarlo indicándole cuál va a ser su tamaño. El tamaño de un array no puede cambiar una vez que esté inicializado.

Para inicializarlo, podemos hacer de dos maneras:

- a) Lo podemos inicializar con la palabra "new" y le indicamos el tamaño que va a tener la lista. El array estará vacío al principio. En este ejemplo lo inicializamos con tamaño 6:

```
listaPalabras = new String[6];
```

(Esta es la forma más habitual)

- b) Lo podemos inicializar con la palabra "new" y le indicamos la lista de elementos que va a tener inicialmente. El tamaño del array será igual al total de elementos que hemos metido. En este ejemplo, lo inicializamos con 2 palabras y el tamaño definitivo será 2:

```
listaPalabras = new String[]{"candado", "babosa"};
```

La declaración y la inicialización las podemos hacer en la misma línea:

```
Integer[] listaNumeros = new Integer[]{4,89,-9,0,78};  
String[] listaPalabras = new String[6];
```

¿Cómo se cambian los datos que tiene dentro un array?

Podemos cambiar los datos de un array igual que cambiamos el valor de una variable. Lo único que tenemos que indicar es en qué posición del array queremos meter el dato. Esto se hace con un número. IMPORTANTE: la posición inicial de un array será siempre 0. Y la última posición será el tamaño total menos 1.

Ejemplo:

```
String[] listaPalabras = new String[6];  
listaPalabras[0] = "Hola"; // Pongo hola en la primera posición  
listaPalabras[5] = "Adiós"; // Pongo adiós en la última posición  
listaPalabras[6] = "Error"; // Esto daría un error !!!
```

Los valores se pueden cambiar siempre que se quiera:

```
String[] listaPalabras = new String[6];  
listaPalabras[0] = "Hola";  
listaPalabras[0] = "Adiós";  
listaPalabras[0] = "Tercero";  
// Después de esto, el valor de la primera posición será "Tercero"
```

¿Cómo se obtienen los datos que tiene dentro un array?

Se obtienen los datos de un array igual que los de una variable. Lo único que tenemos que indicar es de qué posición del array queremos leer el dato. Esto se hace con un número igual que en el caso anterior.

Ejemplo:

```
String[] listaPalabras = new String[6];  
listaPalabras[0] = "Hola";  
listaPalabras[5] = "Adiós";  
  
String palabra = listaPalabras[0];  
System.out.println(palabra);  
System.out.println(listaPalabras[5]);
```

¿Cómo puedo convertir una cadena en un array?

Las cadenas (String) tienen un método llamado "split" que divide la cadena en trocitos. Cada trozo, se mete dentro un array. La forma de dividir la cadena es según se le indique al método split por parámetro, ya que le debemos pasar por dónde queremos que divida.

Mira los siguientes ejemplos:

Ejemplo1

```
String texto = "Paula-María-Lucía-Lola";

// Queremos dividir el texto separando las palabras por los guiones que existan
// Esto nos devolverá un array de tamaño 4
String[] nombres = texto.split("-");

// Al imprimir esto se imprimirán los 4 nombres sin los guiones
for (int i = 0; i < nombres.length; i++) {
    System.out.println(nombres[i]);
}
```

Ejemplo2

```
String frase = "Me gusta cuando callas porque estás como ausente";

// Si queremos obtener todas las palabras, podemos dividir el texto por el espacio " "
// Tendremos un array de tamaño 8
String[] palabras = frase.split(" ");

// Al imprimir esto se imprimirán las 8 palabras de la frase
for (int i = 0; i < palabras.length; i++) {
    System.out.println(palabras[i]);
}
```

Ejemplo3

```
String nombre = "Napoleón";

// Si al split le pasamos una cadena vacía, nos devolverá un array con todas las letras
String[] letras = nombre.split("");

// Al imprimir esto se imprimirán las 8 letras de la palabra Napoleón
for (int i = 0; i < letras.length; i++) {
    System.out.println(letras[i]);
}
```

¿Qué es un bucle foreach?

Es un bucle que nos permite recorrer todos los elementos de un array de forma sencilla. La forma en la que se escribe es la siguiente:

```
for(TipoDeDato nombreVariable : nombreArray){  
    // aquí iría lo que queremos hacer con cada elemento del array  
}
```

Por ejemplo, para imprimir todos los elementos de un array de enteros, se haría así:

```
Integer[] números = new Integer[]{3, 3, 2, 5};  
for(Integer numero : numeros){  
    System.out.println(numero);  
}
```

AVANZADO

¿Qué es un array multidimensional?

Es un array que no tiene sólo una dimensión como los que hemos visto hasta ahora. Es decir, es un array de arrays. Por ejemplo, si tengo un array de 2 dimensiones, es como si fuera una tabla: una lista de listas. Si tengo un array de 3 dimensiones, es como si fuera un cubo: una lista de listas de listas.

Otra forma de verlo es que si tengo un array de 2 dimensiones significa que tengo un array que dentro de cada posición lo que hay es otro array.

Se declaran y usan exactamente igual que los arrays de una sólo dimensión, simplemente vamos añadiendo corchetes para cada dimensión adicional. Mira estos ejemplos:

```
// Declaro un array de dos dimensiones de 4 x 3 de tipo Integer;
Integer[][] tablaNumeros = new Integer[4][3];

// Escribo el número 7 en la posición 3, 1:
tablaNumeros[3][1] = 7;

// Imprimo la posición 3, 1:
System.out.println(tablaNumeros[3][1]);
```

Cómo podemos recorrer un array de varias dimensiones:

```
// Cuando declaro un array bidimensional de 6x8 es como
// crear un array de tamaño 6 que dentro de cada posición
// tiene otro array de tamaño 8
Integer[][] tablero = new Integer[6][8];

// Para recorrerlo, tengo que hacer un doble bucle
// Primero itero sobre el array principal
for (int i = 0; i < tablero.length; i++) {
    // Ahora itero el array que hay en cada posición del anterior
    Integer[] arrayInterior = tablero[i];
    for (int j = 0; j < arrayInterior.length; j++) {
        arrayInterior[j] = 1;
    }
}

// También se puede escribir así
for (int i = 0; i < tablero.length; i++) {
    for (int j = 0; j < tablero[i].length; j++) {
        tablero[i][j] = 1;
    }
}
```