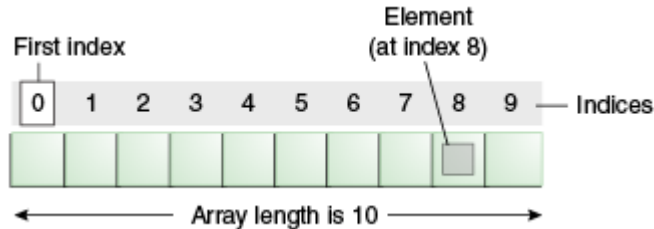


CREACIÓN Y USO DE **ARRAYs**



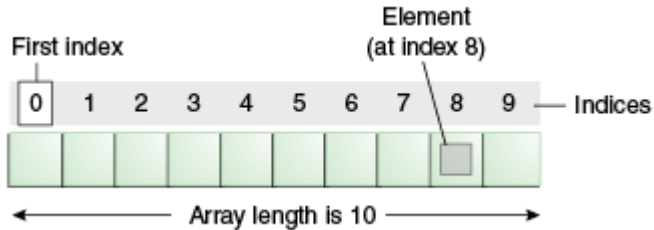
¿QUÉ ES UN **ARRAY**?

- ▶ Es un contenedor de elementos de un mismo tipo, con un tamaño fijo.
- ▶ Su longitud se establece en el momento de crearlo.



¿QUÉ ES UN **ARRAY**?

- ▶ Cada ítem de un array se suele llamar elemento.
- ▶ Se comienza a contar en cero.



CREACIÓN DE UN **ARRAY**

- ▶ Debemos indicar el tipo de dato y el tamaño.
- ▶ Tenemos que usar el operador **new**.

```
int[] unArray = new int[10];
```

Tipo de dato
del array

Los corchetes
indican que es
un array

Nombre
del array

Tamaño
del array

OPERADOR **[]**

- ▶ Nos permite acceder a una posición concreta de un array.
- ▶ Es de lectura/escritura

```
//Asignación de un elemento  
unArray[1] = 100;
```

```
//Lectura de un elemento  
System.out.println(unArray[1]);
```

INICIALIZACIÓN DE UN **ARRAY**

- ▶ Consiste en darle valores iniciales
- ▶ Lo podemos hacer de varias formas.
 - ▶ Elemento a elemento
 - ▶ Atajo mediante la sintaxis con { }
 - ▶ Mediante un bucle (cuando la lógica del programa lo permita).

RECORRIDO DE UN **ARRAY**

- ▶ Todo *array* tiene una propiedad, **.length**, que nos dice su número de elementos.
- ▶ Lo podemos hacer con un bucle **for clásico** o un bucle **for mejorado**.

```
for(int i = 0; i < unArray.length; i++) {  
    System.out.println(unArray[i]);  
}
```

```
for(int i : unArray) {  
    System.out.println(i);  
}
```

ARRAY DE OBJETOS

- ▶ Podemos crear arrays de cualquier **clase**.
- ▶ La sintaxis es idéntica a los tipos primitivos.

```
Persona[] unArray = new Persona[10];
```

- ▶ La inicialización requiere del uso de **new**.

```
unArray[1] = new Persona(...);
```


ARRAYs MULTIDIMENSIONALES

- ▶ Podemos crear arrays de más de una dimensión.
- ▶ Tan solo tenemos que añadir otra pareja de corchetes.

```
int[][] biArray = new int[10][20];
```

- ▶ Para acceder a sus elementos, tenemos que usar también el doble corchete.

```
biArray[3][4] = 78;
```

ARRAYs MULTIDIMENSIONALES

- Para recorrerlos necesitamos bucles anidados.

```
for(int i = 0; i < biArray.length; i++) {  
    for(int j = 0; i < biArray[0].length; j++) {  
        System.out.println(unArray[i][j]);  
    }  
}
```

MANIPULACIÓN RÁPIDA DE ARRAYs

- ▶ Java nos provee de una clase, `java.util.Arrays`, que posee muchos métodos para manipular arrays
- ▶ Ordenación, búsqueda, copia, ...

<https://docs.oracle.com/javase/8/docs/api/java/util/Arrays.html>