



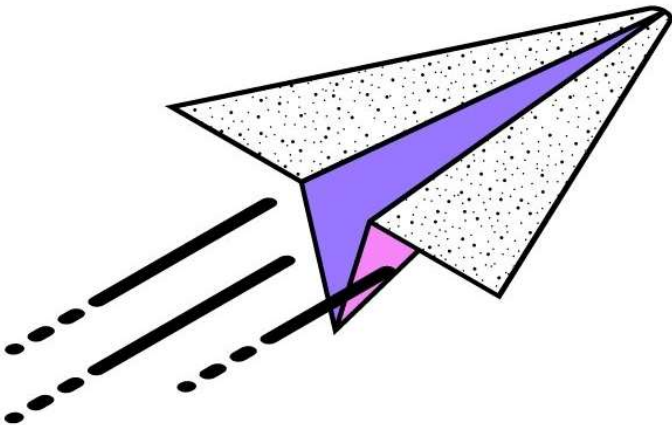
# Java 7 SE Fundamentals





# Seccion 6

Creación y uso de arreglos.



Acceso a un valor en un arreglo o ArrayList.

Declarar, instanciar e inicializar un arreglo unidimensional.

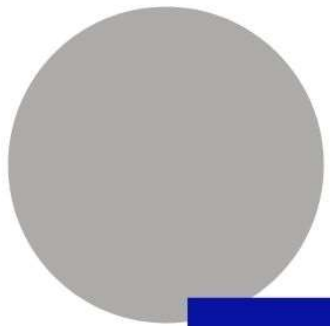
Uso de la sentencia import para trabajar con API's existentes de java.

Declarar, instanciar e inicializar un arreglo bidimensional.

Crear e inicializar un ArrayList.

Utilizar el vector de argumentos. (args Array)

Uso de un ciclo **for** para procesar un arreglo



## Que es un arreglo

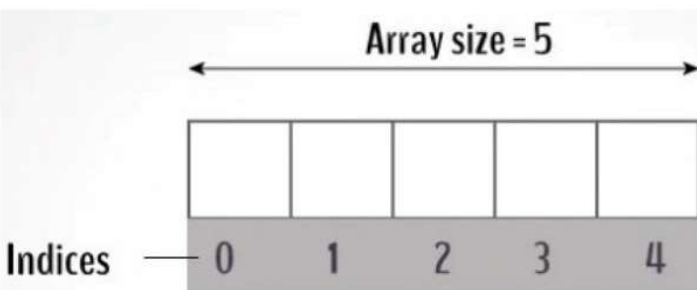
Es una colección de datos de un mismo tipo. Una sola variable de tipo arreglo o array puede hacer referencia a una gran colección de datos.

## Conceptos de arreglos

- Una vez que se crea un arreglo, su tamaño es fijo.
- Una variable de referencia de matriz se utiliza para acceder a los elementos de una matriz mediante un índice.

## Como se declara un arreglo en Java

**tipoElemento** variableReferencia[ ] = **new** tipoElemento [tamaño]





### Creación de un arreglo

```
double[] miArray = new double[10];
```

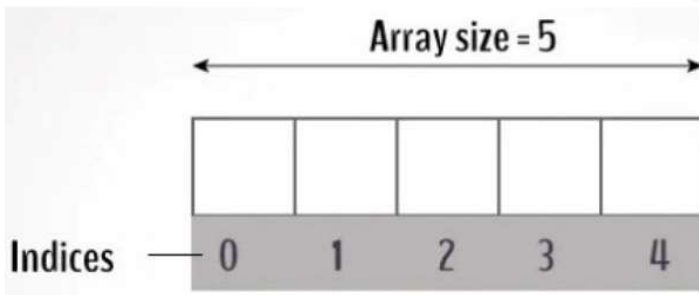
Se crea un arreglo, que guardara datos de tipo double, con una capacidad de 10. El arreglo tiene valores por defecto.

### Creación de un arreglo inicializado

```
double[] miArray= {1.9, 2.9, 3.4, 3.5};
```

Se crea un arreglo, que guarda los valores que están entre las llaves. El arreglo contiene valores específicos desde su creación.

### Asignar valores a un arreglo con valores por defecto.



```
miArreglo[0] = 4.0;  
miArreglo[1] = 34.33;  
miArreglo[2] = 34.0;  
miArreglo[3] = 45.45;  
miArreglo[4] = 99.993;  
miArreglo[5] = 11123;
```



### Creación de un arreglo bidimensional.

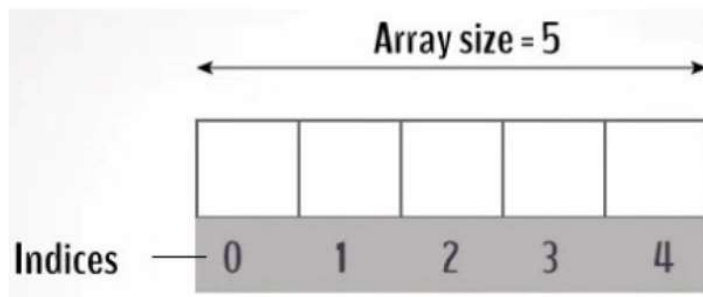
```
int matriz[ ][ ] = new int[3][3];
```

### Creación de un arreglo bidimensional inicializado

```
int[ ][ ] miArreglo= {{1, 2, 3},  
                      {4, 5, 6},  
                      {7, 8, 9},  
                      {10, 11, 12}};
```

### Dar valores a elementos de un arreglo bidimensional.

```
miArreglo[0][0] = 1;  
miArreglo[0][1] = 47;  
miArreglo[0][2] = 88;  
miArreglo[1][0] = 19;  
miArreglo[1][1] = 17;  
miArreglo[1][2] = 189;
```





## Que son los paquetes de java

- Los paquetes son el mecanismo que usa Java para facilitar la modularidad del código.
- Un paquete puede contener una o más definiciones de interfaces y clases, distribuyéndose habitualmente como un archivo.
- Para utilizar los elementos de un paquete es necesario importar este en el módulo de código en curso, usando para ello la sentencia

## Como utilizo los paquetes

- Para utilizar los elementos de un paquete es necesario importar este en el módulo de código en curso, usando para ello la sentencia **import**.





## uso de import

- Importando una sola clase:  
**import java.util.Date;**
- Importando un paquete completo de clases:  
**import java.util.\*;**
- Para utilizar los elementos de un paquete es necesario importar este en el módulo de código en curso, usando para ello la sentencia

## Utilidades con import

- puedes incluir múltiples sentencias **import**:  
**import java.util.Date;**  
**import java.util.Calendar;**
- Por default siempre se importa `java.lang.*`
- No necesitas importar las clases que estén en el mismo paquete.



## ArrayList

Una ArrayList se implementa usando un arreglo. En un arreglo convencional el tamaño de este era fijo una vez definido, y este no podía cambiar, en un ArrayList esto no es así, ya que el arreglo crece de manera dinámica, es decir su tamaño puede variar.

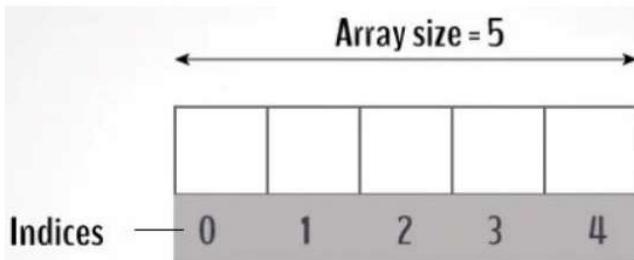
### Creación de un ArrayList

```
import java.util.*;
```

```
ArrayList<String> miArrayList = new ArrayList<>();
```

### Agregar valores a un ArrayList

```
miArrayList.add("Arturo");  
miArrayList.add("Maria");  
miArrayList.add("Pedro");  
miArrayList.add("Fernanda");
```







## Arreglo de argumentos (String[ ] args)

Es un arreglo que guarda objetos de tipo cadena (String), este es pasado justo cuando ejecutamos nuestro programa en la línea de comandos.

## Pasando valores al arreglo (String[ ] args)

Cuando se ejecuta un programa, es posible pasar valores para que estos sean almacenados en el arreglo args.

```
java nombrePrograma [argumento1, argumento2 ... ]
```

## Ejemplo

Suponiendo que tenemos un programa que recibe nombres de personas mediante la línea de comandos, para después imprimir un saludo para estas, la ejecución sería la siguiente.

```
java SaludoPersonas Arturo Sofia Jorge Pedro Camila
```



### Procesado de un arreglo de manera eficaz.

- Existen maneras de realizar tareas que son repetitivas. Para el procesamiento de un arreglo, tenemos al ciclo **for**.
- Con el ciclo **for** es posible recorrer los elementos de un arreglo, para acceder a sus datos o modificarlos de una manera eficaz.

### Ejemplo de uso del ciclo for con un arreglo.

Suponiendo que tenemos definido un arreglo de tamaño 5 y que la referencia se llama `miArreglo`, es posible recorrerlo con la siguiente instrucción:

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