

ENTRADA Y SALIDA BÁSICA

La **SALIDA DE DATOS** en modo texto se puede hacer en Java fácilmente a través del método `println()` de `System.out`. Por ejemplo:

```
System.out.println("El que sabe no habla, el que habla no sabe");
```

Esta sentencia hace que aparezca por pantalla la secuencia de caracteres escrita entre comillas. Esta instrucción es tan flexible que nos permite igualmente escribir enteros, caracteres o números reales.

```
System.out.println(124);  
System.out.println('a');  
System.out.println(1.618033);
```

Muestran por pantalla:

```
124  
a  
1.618033
```

En lugar de escribir la secuencia de caracteres directamente, también nos puede interesar mostrar el valor contenido en una variable, cuyo identificador no debe ir entrecomillado.

```
System.out.println ( area );
```

Si queremos que una instrucción nos muestre varios literales o valores de variables, utilizaremos el símbolo `+` (concatenación).

```
System.out.println ("La suma de los cuadrados de " + numero1 + " y " + numero2 +  
" es: " + suma);
```

Para realizar las **ENTRADAS BÁSICAS DESDE TECLADO**, Java dispone de la clase Scanner, que permite crear objetos capaces de leer información desde una fuente de datos que puede ser un archivo, una cadena de caracteres, el teclado, etc. Los objetos de esta clase, serán los que utilizaremos para pedir los datos que se requieran.

Es imprescindible recordar que es necesaria la clase **java.util** para que todo funcione. Por tanto, al principio del programa incluiremos:

```
import java.util.Scanner;
```

Vamos a suponer que debe solicitarse al usuario la edad de una persona a través del teclado. Lo primero, lógicamente, será declarar la variable en la cual vamos a almacenar el valor leído. El tipo adecuado en este caso es el int.

```
int edad;
```

Una vez declarada, debemos declarar un objeto de tipo Scanner que nos ayudará en la entrada de la información, capturando el valor digitalizado.

```
Scanner teclado;  
teclado = new Scanner(System.in);
```

A continuación, es habitual escribir un literal indicando la entrada:

```
System.out.println("Introduce un número");
```

Para finalizar, se asigna a la variable “edad” el valor devuelto por el método nextInt () del objeto teclado

```
edad=teclado.nextInt( );
```

De la misma forma que es posible capturar un dato de tipo int con el servicio de nextInt(), también existen otros métodos apropiados para otros tipos de datos.

Así, dependiendo del tipo de dato al que corresponda la información que se va a capturar tendremos, para los tipos de datos más importantes:

Para el tipo byte: `nextByte()`
Para el tipo short: `nextShort()`
Para el tipo float: `nextFloat()`
Para el tipo double: `nextDouble()`
Para el tipo long: `nextLong()`
Para el tipo String: `nextLine()`

Ejemplo completo:

```
//importo la clase Scanner para poder utilizarla
import java.util.Scanner;

public class EjemploES{
    public static void main(String[] args){
        double base, altura, area;
        Scanner teclado;
        teclado=new Scanner(System.in);
        //puedo abreviar las dos líneas anteriores en: Scanner teclado = new Scanner(System.in);
        //muestro mensaje
        System.out.print("Introduce base: ");
        //pido un valor por teclado, de tipo double
        base=teclado.nextDouble();
        //Repito los pasos con la altura
        System.out.print("Introduce altura: ");
        altura=teclado.nextDouble();
        area=base*altura/2;
        //muestro los resultados
        System.out.println("El área es: "+area);
    }
}
```

Atención:

Al introducir el valor por teclado debo utilizar la coma para separar los decimales.
En el resultado se muestra el punto como separador decimal.

Resultado:

```
Introduce base: 4,5
Introduce altura: 3
El área es: 6.75
```