SALIDA POR PANTALLA

La vamos a hacer inicialmente en **Java** a través de los métodos **print(), println(), printf()** de la clase **System,** y de una instancia **out** (de tipo PrintStream) asociada a la salida de pantalla por defecto.

```
System.out.println("Este mensaje se mostrará por Pantalla");
```

Usaremos al principio println(), que provoca un salto de línea después de escribir, y que nos permite escribir **cadenas de texto** (entre comillas dobles " "), **caracteres** (entre comillas simples ' '), **números enteros** o **números con decimales**. Directamente, o a través de **variables** que contienen esos tipos de datos.

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

```
124
a
1.618033
```

Si queremos mostrar en un único **println()** varias cadenas de **caracteres/valores/variables**, utilizaremos el símbolo **+ (concatenación).** *(Más adelante veremos que no es conveniente abusar del operador **+** y emplearemos **printf()** para la impresión con formato)*

```
int num1, num2, suma;
num1=4;
num2=7;
suma=num1+num2;
System.out.println ("La suma de " + num1 + " y " + num2  +  " es: " + suma);
```

ENTRADA DE DATOS DESDE TECLADO

Java dispone de la clase **Scanner**, que nos permite crear objetos de tipo Scanner ("streams") para leer desde una fuente de datos (**teclado**, archivo en disco, etc).
Es imprescindible importar la clase **java.util.Scanner** para que podamos crear objetos de tipo Scanner

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

Ejemplo: Solicitar una edad por teclado:

```
int edad;    //declaramos una vble de tipo int para guardar la edad
Scanner sc = new Scanner(System.in);
//declaramos y creamos un objeto de tipo Scanner asociado al teclado del sistema (System.in)
System.out.println("Introduce un número");  //mostramos un mensaje por pantalla
edad=sc.nextInt();
/* llamamos al método nextInt() de la clase System que espera recibir un número entero y lo almacena en la vble edad */
```

Dependiendo del tipo de dato que queremos procesar, usaremos diferentes **métodos** de la clase **System**.

- Para los tipos **numéricos**: nextByte() - nextShort() - nextFloat() - nextDouble() - nextLong()
- Para el tipo **String**: **nextLine**() - (líneas completas hasta \n) **next()** - (palabra hasta un espacio en blanco)

Hay un **pequeño inconveniente** con los métodos para los tipos numéricos **(nextInt(), nextDouble() …**) cuando a continuación ejecutamos **nextLine()** para solicitar una cadena de caracteres.

Se debe a que después de introducir un número por teclado, teclearemos siempre la tecla retorno para confirmar la entrada y eso genera un carácter de retorno de carro **\n** que se queda residualmente en el buffer de teclado.

Como precisamente lo que busca el método **nextLine**() para finalizar su ejecución es un carácter **\n**, la ejecución de **nextLine()** falla, no nos permite teclear nada y devuelve una cadena de caracteres vacía. **Hacer la prueba:**

```
int num;
String frase;
Scanner sc = new Scanner(System.in);
System.out.println("Introduce un número");
num=sc.nextInt();
System.out.println("Introduce una frase");
frase=sc.nextLine();
System.out.println("El número introducido es: " + num );
System.out.println("La frase introducida es: " + frase);
```

Observamos que el programa se salta (aparentemente" la ejecución del método nextLine() y por ello la frase finalmente aparece vacía.
Realmente Java no se saltó ninguna instrucción, lo que sucede es que después del nextInt() se ha quedado un carácter \n en el buffer de teclado y eso le basta al siguiente nextLine() que entiende que hemos tecleado una frase vacía y continua la ejecución.

Solución: de momento lo más sencillo es hacer un llamada aislada e inocua a nextLine() que consuma el carácter \n.

```
int num;
String frase;
Scanner sc = new Scanner(System.in);
System.out.println("Introduce un número");
num=sc.nextInt();
System.out.println("Introduce una frase");
sc.nextLine(); //llamada "extra" para consumir el carácter \n que tras el último nextInt() hemos dejado en el buffer
frase=sc.nextLine();
System.out.println("El número introducido es: " + num );
System.out.println("La frase introducida es: " + frase);
```