



# PRINT Y PRINTLN



La salida de datos por pantalla es una de las funcionalidades fundamentales al momento de desarrollar programas para usuarios, pues como buenos programadores siempre debemos comunicarnos esa persona que usa nuestro programa e informarle de todo lo que está sucediendo. La salida de datos nos permite solicitar información al usuario, pues sería difícil para adivinar qué es lo que esperamos que nos ingrese en algún momento, así que básicamente no habría entrada de información si no hubiera salida de datos por pantalla.

La salida de datos en Java, es bastante sencilla pues basta con usar una línea de código para hacerlo. Para mostrar datos por pantalla en nuestros programas Java. Para mostrar texto por pantalla en Java no es necesario hacer uso de librerías importadas ni similares, pues éstas y están incluidas en la librería `java.lang` que siempre es importada por defecto. Para imprimir por pantalla, se usa la clase `System`, el atributo `out`, y su método `println()` o `print()` así: `System.out.println()` o `System.out.print()` notar que `System` es una clase y siempre debe ir con la "S" mayúscula.



## Diferencia entre print() y println()

La primer pregunta que surge al ver los métodos útiles para salida de datos en Java es en qué se diferencian (print() vs println()), básicamente ambos son útiles para mostrar texto por pantalla y la única diferencia es que println() muestra el texto y luego deja un salto de línea (como si presionara enter) mientras que print() no lo hace, veamos un ejemplo para conocer bien la diferencia:

```
public class Salida
{
    public static void main(String[] args)
    {
        System.out.println("Hola usuario bienvenido");
        System.out.println("A continuación se muestra una
cuenta regresiva");

        for (int i = 10; i >= 1; i--)
        {
            System.out.println(i); //Se muestra cada valor
de i (de 10 al 1)
        }

        System.out.println("Ahora, otra cuenta regresiva");

        for (int i = 10; i >= 1; i--)
        {
```

```
        System.out.print(i);//Se muestra cada valor de  
i (de 1 a 10)  
  
    }  
  
}  
  
}
```

Muy bien, si ejecutamos este código, vamos a ver que el primer ciclo imprime los números del 10 al 1 dejando una línea entre cada número, mientras que el segundo imprime todos los números continuos.

```
Hola usuario bienvenido  
A continuación se muestra una cuenta regresiva  
10  
9  
8  
7  
6  
5  
4  
3  
2  
1  
Ahora, otra cuenta regresiva  
10987654321BUILD SUCCESSFUL (total time: 0 seconds)
```