

System.out.printf() PARA MOSTRAR DATOS POR PANTALLA

Uso:

```
System.out.printf("formato", argumentos a formatear separados por comas);
```

La sintaxis de **formato**, que tiene que ir encerrado entre comillas es la siguiente:

`%[indice_del_argumento$][flags][width][.precisión]conversión`

- Los valores entre `[]` son opcionales, el resto son obligatorios.
- Siempre empiezan con el porcentaje `%`.
- **indice_del_argumento** indica la posición del argumento a formatear, ejemplo: **"1\$"** tratará el primer argumento, **"2\$"** tratará el segundo argumento...
- **flags:**
 - Justificar el argumento a la izquierda.
 - + El argumento tiene un signo (+ o -).
 - 0 Rellenar con ceros a la izquierda hasta llegar al tamaño máximo.
 - , El argumento tiene grupos separados por comas, así: 123,456.
- **width:** Mínimo valor a imprimir.
- **precisión:** Sólo para flotantes. Especifica la precisión, cuantos valores después de la coma.
- **conversión:**
 - b** boolean
 - c** char
 - d** integer, long, short, byte
 - f** float, double
 - s** String, char[]

EJEMPLOS: (`%n` o `\n` salta línea, o `\t` tabulador)

Para redondear y limitar la cantidad de decimales, carácter f double n = 1.2345; System.out.printf("Decimal: %.3f %n", n);	Salida: Decimal: 1,234
Para mostrar el signo + en un número positivo: double n = 1.2345; System.out.printf(" %+.2f %n", n);	Salida: +1,23
Si el número a mostrar es un entero se utiliza el carácter d : int x = 10; System.out.printf("Numero: %d %n", x);	Salida: Numero: 10
Si es una cadena se usa el carácter s : String s1 = "Hola"; String s2 = "Adios"; int suma = 100; System.out.printf(" %s y %s %nLa suma es: %d %n", s1, s2, suma);	Salida: Hola y Adios La suma es: 100
System.out.printf("SUMA : %3\$d %n %2\$s y %1\$s %n", s1, s2, suma);	Salida: SUMA : 100 Adios y Hola

DAR FORMATO CON EL PUNTO DE LOS MILES Y LA COMA DE LOS DECIMALES:

```
System.out.printf("CAMPOS : *%,15.2f*%,12.2f* %n", 1234567.89, 12345.67);
```

Salida: CAMPOS : * 1.234.567,89* 12.345,67*
15 posiciones 12 posiciones

El primero usa 15 posiciones para pintar el número, de esas 2 son decimales: `%,15.2f`

El segunda usa 12 posiciones para pintar el número, de esas 2 son decimales: `%,12.2f`

MOSTRAR ENTEROS CON FORMATO (PUNTO PARA LOS MILES) 15 posiciones: `%,15d`

El signo - justifica a la izquierda:

```
System.out.printf("CAMPOS : *%-,15d*%,15d* %n", 123456789, 123456789);
```

Salida: CAMPOS : *123.456.789 * 123.456.789*

Mostrar datos en columnas:

```
System.out.printf("%5s %-10s %16s\n", "COLUM", "COLUMNA2", "COLUMNA3");
System.out.printf("%5s %10s %16s\n", "====", "=====", "=====");

System.out.printf("%-5d %-10s %-16s\n", 10, "Carlos", "Mauricio");
System.out.printf("%05d %10s %16d\n", 10, "Carlos", 456);
System.out.printf("%-5d %-10s %16s\n", 10, "Carlos", "Mauricio");
System.out.printf("%32s %n", "-----");

int total = 1564320;
System.out.printf("%16s %16d %n", "(0)Suma total: ", total);

DecimalFormat formato = new DecimalFormat("#,###,###,###.##");
String Total;

Total = formato.format(total);
System.out.printf("%16s %16s %n", "(1)Suma total: ", Total);
System.out.printf("%16s %,16d %n", "(2)Suma total: ", total);

double totalfloat = 123450.347;
Total = formato.format(totalfloat);

System.out.printf("%16s %16s %n", "(3)Suma total: ", Total);
System.out.printf("%16s %,16.2f %n", "(4)Suma total: ", totalfloat);
```

Salida:

COLUM	COLUMNA2	COLUMNA3
====	=====	=====
10	Carlos	Mauricio
00010	Carlos	456
10	Carlos	Mauricio

(0)Suma total:	1564320	
(1)Suma total:	1.564.320	
(2)Suma total:	1.564.320	
(3)Suma total:	123.450,35	
(4)Suma total:	123.450,35	

- Cuando se coloca **%-16s** indica que se visualizará cada cadena de texto en 16 posiciones y alineada a la izquierda.
- Con **%16s** la cadena se escribe en 16 posiciones alineada a la derecha.
- Con **%-5d** se indica que se visualizará el número en 5 posiciones alineado a la izquierda.
- Con **%05d** se indica que se visualizará el número en 5 posiciones rellenando 0 a la izquierda.
- Con **%16d** se indica que se visualizará el número en 16 posiciones alineado a la derecha.
- Con **%,16d** se indica que se visualizará el número en 16 posiciones alineado a la derecha colocando el punto para los miles.
- Con **%,16.2f** se indica que se visualizará el número en 16 posiciones alineado a la derecha colocando el punto para los miles y la coma para los decimales.

Para dar formato a cantidades numéricas incluyendo el punto de los miles y la coma de los decimales también podemos usar la clase **DecimalFormat**, es necesario importar el paquete **java.text.DecimalFormat**. Definimos un objeto de ese tipo, con el formato que queramos que tengan las cantidades:

```
DecimalFormat formato = new DecimalFormat("#,###,###,###.##");
```

Y después usamos el método **format()**, indicando ente los paréntesis la cantidad numérica a formatear:

```
String cadena = formato.format(número a dar formato);
```

<https://docs.oracle.com/javase/8/docs/api/java/text/DecimalFormat.html>

Podemos usar *String.format()* para dar formato:

```
float nume =1234.567f;
String cadena = String.format("%,10.3f", nume);
System.out.println("LA CADENA ES: "+ cadena);
```

Visualiza: LA CADENA ES: 1.234,567

```
String nombre="Maria";
double nota = 123456.7;
long entero = 123456789;
int curso =1;
String cadena = String.format("%-20s**%,5.2f**%05d**%-,15d  %n",
                               nombre, nota, curso, entero );
System.out.println("LA CADENA ES: "+cadena);
```

Visualiza:

LA CADENA ES: Maria **123.456,70**00001**123.456.789

*Se puede usar **System.out.format(.....)** en lugar de **System.out.printf(.....)** ambos métodos son equivalentes.*

<https://docs.oracle.com/javase/tutorial/java/data/numberformat.html>