

TEMA 4. ENTRADA/SALIDA EN JAVA

Vamos a ver como utilizar printf para dar formato a los datos que se imprimen por pantalla en Java. Este problema se nos plantea por ejemplo cuando queremos mostrar un número de tipo float o double con un número determinado de decimales y no con los que por defecto muestra Java.

A partir de la versión Java 5 se incorporan los métodos format y printf que permiten aplicar un formato a la salida de datos por pantalla. Ambos realizan la misma función, tienen exactamente el mismo formato.

Veamos primero varios **ejemplos de printf en Java** y después explicaremos en detalle la sintaxis de printf.

Si queremos mostrar el número 12.3698 de tipo double con dos decimales:

```
System.out.printf("%.2f %n", 12.3698);
```

- **El primer %** indica que en esa posición se va a escribir un valor. El valor a escribir se encuentra a continuación de las comillas.
- **.2** indica el número de decimales.
- **La f** indica que el número es de tipo float o double. En la tabla que aparece más adelante podeis ver todos los caracteres de conversión para todos los tipos de datos.
- **%n** indica un salto de línea. Equivale a \n. Con printf podemos usar ambos para hacer un salto de línea.

La salida por pantalla es: 12,37

Comprobamos que *printf realiza un redondeo para mostrar los decimales indicados*. Lo más común será que tengamos el valor en una variable, en ese caso si queremos escribir el valor de n con tres decimales:

```
double n = 1.25036;  
System.out.printf("%.3f %n", n);
```

Salida:
1,250

Para *mostrar el signo + en un número positivo*:

```
double n = 1.25036;
```

```
System.out.printf("%+.3f %n", n);
```

Salida: +1.250

Si el número a mostrar es *un entero se utiliza el caracter d*:

```
int x = 10;
```

```
System.out.printf("%d %n", x);
```

Salida: 10

Para *mostrarlo con signo*:

```
int x = 10;
```

```
System.out.printf("%+d %n", x);
```

Salida: +10

Si queremos *mostrar el número 123.4567 y su cuadrado ambos con dos decimales* debemos escribir:

```
double n = 123.4567;
```

```
System.out.printf("El cuadrado de %.2f es %.2f\n", n, n*n);
```

Salida:

El cuadrado de 123,46 es 15241,56

Explicación de cada una de las partes que aparecen en la instrucción printf:



Especificadores de formato. La sintaxis para los especificadores de formato de printf es:

%[posición_dato\$][indicador_de_formato][ancho][.precision]carácter_de_conversión

Los elementos entre corchetes son opcionales.

- **posición_dato**: indica la posición del dato sobre el que se va aplicar el formato. El primero por la izquierda ocupa la posición 1.
- **indicador_de_formato**: es el conjunto de caracteres que determina el formato de salida.

INDICADORES DE FORMATO			
Indicador	Significado	Indicador	Significado
-	Alineación a la izquierda	+	Mostrar signo + en números positivos
(Los números negativos se muestran entre paréntesis	0	Rellenar con ceros
,	Muestra el separador decimal		

Los indicadores de formato de printf en Java son:

- **ancho**: Indica el tamaño mínimo, medido en número de caracteres, que debe ocupar el dato en pantalla.
- **precisión**: Indica el número de decimales que serán representados. Solo aplicable a datos de tipo float o double.
- **carácter_de_conversión**: Carácter que indica cómo tiene que ser formateado el dato. Los más utilizados se muestran en la tabla

CARACTERES DE CONVERSIÓN			
Carácter	Tipo	Carácter	Tipo
d	Número entero en base decimal	X, x	Número entero en base hexadecimal
f	Número real con punto fijo	s	String
E, e	Número real notación científica	S	String en mayúsculas
g	Número real. Se representará con notación científica si el número es muy grande o muy pequeño	C, c	Carácter Unicode. C: en mayúsculas

Printf permite mostrar valores con un ancho de campo determinado. Por ejemplo, si queremos mostrar el contenido de n en un ancho de campo de 10 caracteres escribimos:

```
double n = 1.25036;
```

```
System.out.printf("%+10.2f %n", n);
```

Salida:

`bbbbbb+1.25`

- Donde cada *b* indica un espacio en blanco.
- El 10 indica el tamaño en caracteres que ocupará el número en pantalla. Se cuentan además de las cifras del número el punto decimal y el signo si lo lleva. En este caso el número ocupa un espacio de 5 caracteres (3 cifras, un punto y el signo) por lo tanto se añaden 5 espacios en blanco al principio para completar el tamaño de 10.

Si queremos que en lugar de espacios en blancos nos muestre el número completando el ancho con ceros escribimos:

```
System.out.printf("%+010.2f %n", n);
```

Salida:

`+000001.25`

Más ejemplos de printf:

Mostrar el número 1.22 en un ancho de campo de 10 caracteres y con dos decimales.

```
double precio = 1.22;  
System.out.printf("%10.2f", precio);
```

Salida:

`bbbbbb1.22` (el carácter b indica un espacio en blanco)

El número ocupa un espacio total de 10 caracteres incluyendo el punto y los dos decimales.

Mostrar la cadena "Total:" con un ancho de 10 caracteres y alineada a la izquierda:

```
System.out.printf("%-10s", "Total:");
```

Salida:

`Total:bbbb`

El carácter s indica que se va a mostrar una cadena de caracteres. El signo - indica alineación a la izquierda.

Mostrar la cadena "Total:" con un ancho de 10 caracteres y alineada a la derecha:

```
System.out.printf("%10s", "Total:");
```

Salida:

*bbbb*Total:

RESUMEN de formatos que pueden ponerse dentro de printf

%d	Entero decimal
%u	Entero decimal con signo
%x	Entero hexadecimal
%c	Carácter
%f	Coma flotante (float)
%lf	Coma flotante (double)

Secuencias de escape que pueden ponerse dentro de print

`\n` Salto de línea `\t` Tabulación

Sin probarlo con Eclipse, ¿Qué mostraría el siguiente fragmente de código ?

```
int una = 1234;  
char otra = 'h';  
System.out.printf( "\n\t Una vale %d; otra vale %c\n", una, otra );
```