





2. Lectura y salida por consola

 Date	Empty
 Status	Not started
 Type	Empty
Units	 2. Estructura de un programa informático

Salida por consola

Lectura por consola

Salida por consola

Para poder sacar un mensaje por consola se puede utilizar el método `println` de la librería `System.out`. Del mismo modo los valores de tipos primitivos y cadenas se pueden mostrar por consola formateados utilizando el método `format` o `printf` de la clase `System.out`. con la siguiente sintaxis

```
System.out.format("Cadena de formato", v1, ... , vn); System.out.printf("Cadena de formato", v1, ... , vn);
```

Donde la cadena de formato está compuesta por textos y elementos. Para cada elemento de la cadena de formato habrá un valor a mostrar con el formato indicado en dicho elemento. Así, v1 será el valor que se mostrará con el formato del primer elemento que aparece en la cadena de formato, y así sucesivamente. El valor puede ser un literal, constante, variable o expresión de tipo el indicado en su correspondiente elemento de formato. Un ejemplo sería:

```
System.out.printf("Me llamo %-10s y tengo %04d años", "Pedro", 22);
```

En la cadena de formato hay dos elementos. El valor "Pedro" se asocia con el elemento %-10s y el valor 22 se asocia con el elemento %04d. El primer elemento indica que se va a mostrar una cadena ocupando 10 espacios y ajustando el valor de la cadena a la izquierda. El segundo elemento indica que se va a mostrar un valor entero ocupando 4 espacios, ajustado por la derecha y rellenando con 0 por la izquierda hasta completar el espacio que ocupa el entero.

Cada uno de los valores que le acompañan al símbolo de % se sustituirán por los argumentos que le son pasados al método. El formato siempre será el siguiente:

```
%[Banderas][Tamaño]Conversor
```

Donde las posibles banderas a utilizar son las siguientes:

- 0 Rellenar con 0 por la izquierda hasta completar el tamaño. Sólo válido para tipos primitivos numéricos.
- Se mostrará el signo + para los positivos. Sólo válido para tipos primitivos numéricos.
- (Los valores negativos se pondrán entre paréntesis y no se mostrará el signo '- '. Sólo válido para tipos primitivos numéricos.
- , Se mostrará el carácter decimal correspondiente a la zona geográfica. Sólo válido para tipos primitivos numéricos reales.
- Se ajustará el valor a la izquierda (por defecto se ajusta a la derecha)

Donde los posibles tamaños son

- n El valor ocupará un total de n espacios (si el valor ocupase más espacios, se mostraría en su totalidad).
- n.m El valor ocupará n espacios y se mostrará con m decimales. Puede ser que n sea menor que m. El valor real se muestra redondeado a "m" decimales. Sólo válido para tipos primitivos numéricos reales.

Donde los posibles conversores son:

- o El valor entero se muestra en octal.
- h El valor entero se muestra en hexadecimal (no permite la bandera 0)
- x El valor entero se muestra en hexadecimal
- d El valor entero se muestra en base 10.
- b Conversor para el tipo primitivo boolean.
- c Conversor para el tipo primitivo char
- s Conversor para cadenas
- n Salto de línea (no admite banderas ni tamaño, y no tiene asociado ningún valor)
- % Muestra el '%' (no admite banderas ni tamaño, y no tiene asociado ningún valor)
- a El valor real se muestra en hexadecimal
- e El valor real se muestra en notación científica
- f El valor real se muestra con notación decimal

Lectura por consola

Antes de empezar y aunque se explicará más adelante en detalle es necesario saber que para poder leer datos por teclado se hará utilizando la clase Scanner que se encuentra en la librería o paquete "java.util".

Se definirá una constante global de tipo Scanner con el valor inicial de fichero de entrada el de la consola: `final Scanner in = new Scanner(System.in);`

Para poder definir una variable de tipo Scanner en una aplicación se tendrá que importar dicha clase en la aplicación de la siguiente manera:

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

Existen métodos específicos para cada uno de los datos los cuales son los siguientes:

```
a = in.next(); //a es de tipo String
a = in.nextByte(); //a es de tipo byte
a = in.nextShort(); //a es de tipo short
a = in.nextInt(); //a es de tipo int
a = in.nextLong(); //a es de tipo long
a = in.nextFloat(); //a es de tipo float introducido con ,
a = in.nextDouble(); //a es de tipo double introducido con ,
a = in.nextBoolean(); //a es de tipo boolean
a = in.next().charAt(0); //a es de tipo char
a = in.nextLine(); //a es de tipo String // la ejecución de in.nextLine() vacía el buffer de lectura
```