

[Área personal](#) / [Mis cursos](#) / [SMC 21028](#) / [Práctica 4: Tutorial de Csound](#) / [Enunciados continuación del tutorial Csound \(NP\)](#).

Información

Los ejercicios de la sesión no presencial del tutorial de Csound están concebidos para servir de complemento a los vistos en la práctica presencial.

Esta práctica no presencial es de especial importancia para comprender aspectos clave de Csound necesarios para el resto de las prácticas.

Debes entregar sólo los ficheros CSD de esta parte de la práctica (NP) a través de la actividad para ello.

## Ejercicio 4.5: Imprimir valores en consola

Csound tiene varios operadores para imprimir valores en la consola (recuerda que en la parte inferior de la ventana de CsoundQt te muestra la sintaxis del operador sobre el que tengas el cursor):

- **print** : Escribe en la consola de salida los valores de tipo i- que se le suministran. Lo hace **una sola vez**, en la inicialización del instrumento.
- **prints** : Escribe cadenas de texto y valores numéricos (o variables), usando una cadena de formato (como en C):
  - **%d** se usa para imprimir enteros; %3d reserva 3 espacios para escribir el número.
  - **%f** se usa para imprimir reales; %5.2f reserva 5 espacios, 2 para los decimales.
  - **%s** se usa para imprimir cadenas; %6s reserva 6 espacios.
  - **EJEMPLO:**
    - **prints** `"%s: %3d+%d=%4.1f\n"`, "Suma", 2, 3, 5
    - imprimiría en la consola: Suma: 2+3= 5.0 (el `\n` final es un salto de línea)
- **printk** : Escribe en la consola el valor de una variable de tipo k-. Lo hace **cada cierto tiempo**, especificado en segundos como primer parámetro.
- **printk2** : Escribe el valor de una variable tipo k- cada vez que cambie su valor.
- **printks** : Escribe valores de variables de tipo k- usando los formatos anteriores.
- **fprintks** : Hace lo mismo, pero imprimiendo en un fichero.

Crea desde cero un nuevo proyecto, de nombre `ej4.5.csd`. Vamos a ver qué poner en cada sección.

<CsOptions> (parámetros de compilación)

Como en este ejercicio no se generan señales, indícalo con el parámetro `-n`

</CsInstruments> (orquesta)

Todo lo que escribamos antes del primer instrumento se denomina **"instrumento cero"**. Ahí se escriben las especificaciones globales para nuestro proyecto. En este caso, debes usar las siguientes:

- **sr = 1000** ; (frecuencia de muestreo) Aunque no vamos a generar sonido, el bucle interno de cálculo de Csound se repetirá 1000 veces en 1 segundo.
- **ksmps = 20** ; las variables de tipo k- se recalcularán cada 20 iteraciones del bucle anterior. Por lo tanto, lo harán 50 veces en 1 segundo.

### Diseño del instrumento:

Usaremos identificadores para los instrumentos.

Crea el instrumento número **CINCO**. Primero escribe el par de palabras que inicia y termina el código:

```
instr CINCO
endin
```

Y ahora el código, que irá entre las palabras anteriores.

Primero probaremos los operadores que imprimen constantes (valores o variables de tipo i-). Para ello creamos una variable cadena (las que empiezan por S) y una constante (las que empiezan por i-). Luego imprimiremos sus valores. Escribe este código.

```
SVamos = "EMPEZAMOS"
iCinco = 5
print iCinco
prints "%sEste no es el ejercicio %d, es el %d\n\n", SVamos, 4, iCinco
```

Para probar los operadores que imprimen variables hay que tener en cuenta que pueden cambiar de valor continuamente. Vamos a crear una variable de tipo k-, iniciarla a 0, incrementarla de 10 en 10 e imprimir su valor cada vez que cambie. Recuerda que todo esto irá dentro de un bucle (que lo hace Csound, no tú). Las órdenes a añadir a continuación de las anteriores son (la primera **-init**- sólo se ejecuta una vez, cuando se inicializa el instrumento):

```
kValor init 0
printk2 kValor
kValor = kValor+10
```

<CsScore> (partitura)

Simplemente, activa el instrumento **CINCO** desde el principio (tiempo 0) durante 1 tiempo (como el tempo es 60 BPM, 1 tiempo dura 1 segundo). La orden será:

```
i "CINCO" 0 1
```

### Funcionamiento:

Compila y prueba el funcionamiento. Como el instrumento funciona durante 1 segundo y las variables de control se recalculan `sr/ksmps = 50` veces por segundo, se imprimirán 50 valores en consola (recuerda que **printk2** imprime cada vez que la variable cambia de valor).

Segunda versión (esta es la que se entrega)

Vamos a hacer que se impriman menos valores; que lo haga sólo cada 2 décimas de segundo (5 veces en un segundo por tanto).

Para ello cambiaremos el operador **printk2** por **printk**, pero a este hay que decirle que imprima en ese periodo de tiempo solamente: **printk** 0.2, **kValor**.

Nada cambiará en la ejecución del programa, sólo cuándo se imprimen valores en la consola.

Compila de nuevo y comprueba que ahora sólo se imprimen 5 valores. Lo que aparece antes de cada valor impreso ahora es qué número de instrumento (sólo hay 1, pues ese) imprime ese valor y en qué tiempo lo hace.

Comprueba que al principio aparece correctamente la frase:

EMPEZAMOS:

Este no es el ejercicio 4, es el 5    (seguida de una línea en blanco)

◀ Material y entrega para la práctica 4 (P)

Ir a...

Entrega de esta parte de la práctica 4 (NP) ▶