

Lenguaje C Mostrando datos por pantalla

Javier Fernández Rivera - www.aurea.es

Muestreo e impresión de datos printf

Cuando empezamos a programar en un lenguaje a todo el mundo le gusta ver los resultados que va obteniendo. Por ello es importante que empecemos por la función "printf". Esta función tiene como única misión, la de imprimir datos por pantalla.

Sintaxis: printf ("texto [cadena de control]" ,[identificador]);

printf : es el nombre de la función, el comando. A continuación situamos los paréntesis y le pasamos a la función unos parámetros o argumentos, para que opere con ellos y nos devuelva un resultado.

"texto": Lo que se encuentra entre comillas, será lo que la función va a imprimir en pantalla.

[cadena de control] : esto es opcional solo si queremos que aparezca el valor de un identificador.

Se Pueden imprimir dos tipos de datos.

O simples cadenas de caracteres invariables (texto sin mas), o datos variables (con cadenas de control)

Será explicado mas adelante

Posteriormente debemos cerrar las comillas y poner una coma.

[identificador] : Es aquí donde especificamos el nombre de la variable o constante a devolver el valor en la ya mencionada cadena de control. Si no se especifica una cadena de control este campo se omite también, junto con la coma que los separa.

Primera aproximación

Para entender esto es mejor verlo mediante ejemplos.

```
#include <stdio.h>
main () {
printf("Primera línea escrita");
printf("Segunda línea escrita");
}
```

Escribamos este pequeño programa en el compilador de C. Y compilémoslo. Al ejecutarlo veremos como imprime en pantalla el texto especificado en la función printf, pero muestra en una misma línea las dos cadenas de caracteres o strings.

Para evitar esto se usan las llamadas secuencias de escape. Que se encargan de alterar y sacar a nuestro gusto los resultados de la función printf.

Secuencias de escape

Sirven para representar caracteres no imprimibles, así como comillas dobles, apostrofos, el signo de interrogación final y la barra inclinada.

Las secuencias de escape nos sirven para hacer un formateado del texto que se va a imprimir, o lo que es lo mismo para que el texto a imprimir salga como nosotros queramos.

Una secuencia de escape empieza siempre por el signo "\ " y le sigue un identificativo, puesto de forma predeterminada por el lenguaje C.

Veremos una serie de secuencias de escape para realizar ciertas operaciones

\a Esto hace sonar el pip del ordenador.

- ␣ Procesa el carácter back space o lo que es lo mismo el carácter correspondiente a la tecla de borrar.
- ␣ Es el tabulador.
- ␣ Procesa una nueva línea.
- ␣ Ejecuta el retorno de carro.
- ␣ Imprime comillas.
- ␣? Imprime signo de interrogación final.
- ␣ Imprime barra inclinada.

Todas las constantes de tipo de cadena de caracteres acaban con una secuencia de escape interna y procesada por el ordenador y oculta para nosotros. Tal secuencia de escape es \0. Esto hace que si una constante de cadena de caracteres almacena esto "OriON ScripT" en realidad lo que almacena es esto "OriON ScripT\0".

Existen unas **secuencias de escape especiales**, estas sirven para representar cualquier carácter de la tabla ascii.

Para su uso se coloca la \ y a continuación el numero que corresponde al carácter ascii, pero el numero ha de estar pasado a octal.

Por ejemplo, queremos representar el signo de marca registrada "®", para ello no tenemos mas que saber su numero en la tabla ascii, este es el 169, a continuación se pasa a octal, para ello se divide entre 8 y obtenemos cogiendo los restos a la inversa el numero "251". Ahora solo tenemos que poner la barra \251 y se representaría mediante esta secuencia de escape el signo de marca registrada.

Una vez visto esto, observemos el resultado del uso de una secuencia de escape con un ejemplo.

```
#include <stdio.h>
main () {
printf("\nPrimera línea escrita");
printf("\nSegunda línea escrita");
}
```

En este otro programa vemos como con el uso de \n las líneas aparecen una debajo de otra. Por resultado de la secuencia de escape.

Limpiando pantalla

Una vez que ya sepamos imprimir texto, veamos ahora como limpiar la misma pantalla.

Para la limpieza de la pantalla en C se usan una serie de funciones dependiendo del SO (sistema operativo) que se use.

La función por excelencia para limpiar la pantalla es: `clrscr()`; Esta función para acordarse mejor saber que viene de `clear screen`.

En otros SO se pueden usar otras funciones con el mismo resultado, como son:

- ❖ Para MSDOS: `system("cls");`
- ❖ Para LINUX: `system("clear");`

```
#include <stdio.h>
main () {
printf("\nPrimera línea escrita");
clrscr();
printf("\nSegunda línea escrita");
}
```

En este ejemplo solo veremos aparecer la segunda línea escrita por acción del `clear screen` que se ejecuta después de la primera línea y por lo tanto la borra.