



APRENDERAPROGRAMAR.COM

PEDIR DATOS POR
CONSOLA (TECLADO) EN
JAVA. ESCAPE BACKSLASH.
PRINT Y SALTO DE LÍNEA
CON \N. (CU00657B)

Sección: Cursos

Categoría: Curso “Aprender programación Java desde cero”

Fecha revisión: 2029

Resumen: Entrega nº57 curso Aprender programación Java desde cero.

Autor: Alex Rodríguez

PEDIR DATOS POR CONSOLA (TECLADO) EN JAVA.

Vamos a crear un programa elemental para pedir datos por consola (entrada de teclado del usuario) y para ello vamos a basarnos en una clase del API de Java: la clase Scanner. Estudia el siguiente código y busca la documentación sobre la clase. Trata de verificar que los métodos que empleamos aparecen en la documentación y consulta su detalle.



Debes acostumbrarte a buscar en el API, leer documentación y utilizar clases y métodos disponibles. Cuando programes en Java tendrás que hacerlo con frecuencia, como vamos a hacer ahora con la clase Scanner. Escribe este código en tu entorno de desarrollo.

```
import java.util.Scanner; //Importación del código de la clase Scanner desde la biblioteca Java
//Código de nuestra clase principal ejemplo aprenderaprogramar.com
public class TestPrograma1 {

    public static void main (String [ ] args) {
        System.out.println ("Empezamos el programa");
        System.out.println ("Por favor introduzca una cadena por teclado:");
        String entradaTeclado = "";
        Scanner entradaEscaner = new Scanner (System.in); //Creación de un objeto Scanner
        entradaTeclado = entradaEscaner.nextLine (); //Invocamos un método sobre un objeto Scanner
        System.out.println ("Entrada recibida por teclado es: \"\" + entradaTeclado +\"\"");
    } //Cierre del main
} //Cierre de la clase
```

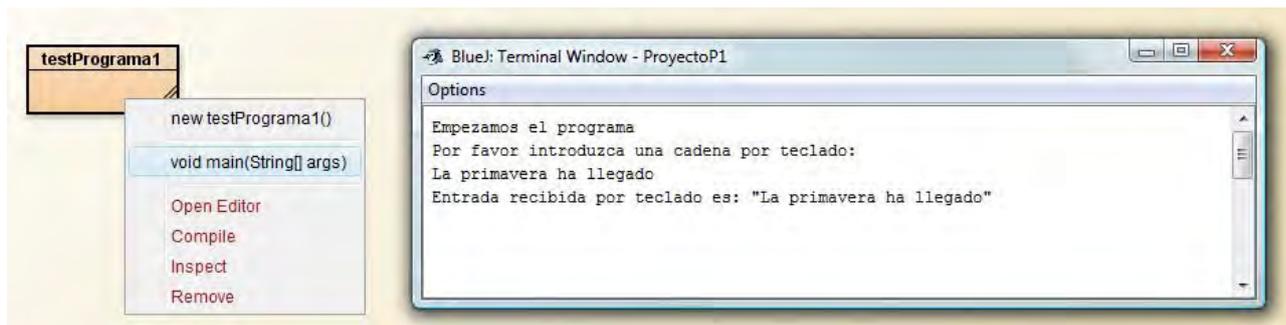
El constructor que hemos utilizado dentro de los varios disponibles en la clase Scanner es el que tiene la signatura Scanner ([InputStream](#) source). Le pasamos como parámetro *System.in*. Ahora no debemos preocuparnos por comprender los detalles, simplemente hemos de entender que estamos creando un objeto de un tipo definido por el API de Java, usando un constructor al que le pasamos un parámetro.

Cuando sobre el objeto Scanner se invoca el método *nextLine()*, se obtiene un String que se corresponde con el texto que haya introducido el usuario mediante el teclado. Este resultado del método lo almacenamos en la variable *entradaTeclado*. Finalmente, mostramos por pantalla cuál ha sido el contenido del texto recibido por teclado. En la sentencia en que imprimimos el mensaje por pantalla hemos usado el carácter \. Este carácter, una **barra invertida o backslash**, no se muestra en pantalla y es interpretado por Java como indicación de que el siguiente carácter forma parte del texto. Esta es la forma que tenemos para incluir unas comillas en un texto que queramos mostrar por pantalla. Sin este escape, cualquier comilla se interpretaría como delimitadora de una cadena. Otra aplicación del backslash es la de forzar un salto de línea. Hasta ahora hemos trabajado invocando el método *println* del objeto *System.out* para imprimir una línea en la ventana de consola. Cabe citar que

otro método disponible para System.out es print, con el resultado de que tras un print no se inserta un cambio de línea y retorno de carro como ocurre con los println, por lo que los textos aparecen “uno a continuación de otro”. Esto es útil, por ejemplo para escribir listas de números separados por comas o guiones sin necesidad de usar una línea por número.

Ahora bien, si usamos print normalmente en el último elemento que queramos mostrar sí queremos insertar un salto de línea y retorno de carro. Esto lo podemos hacer con el carácter de escape ‘\n’. Un ejemplo sería: System.out.print ("Esta es una primera frase. \nY esta es una segunda frase.\n");

Volvamos al código que habíamos escrito. Ejecuta el método main de la clase TestPrograma1.



Nuestra clase TestPrograma1 define un tipo, pero en realidad no nos interesa instanciar esta clase. La utilizaremos como un paquete de código que podemos ejecutar.

Hay que incidir en una cuestión que ya hemos comentado. ¿Qué sentido tiene que una entrada de teclado sea un objeto? Si no lo tienes claro en vez de seguir avanzando vuelve a leer los epígrafes anteriores relacionados con objetos.

Hay distintas maneras de pedir entradas por teclado en Java. Seguiremos viendo cuestiones relacionadas con este asunto en próximos apartados.

EJERCICIO

Crea una clase con un método main que pida una entrada de teclado y usando condicionales, el método length de la clase String y el método substring de la clase String, muestre un mensaje indicando:

- a) Si la cadena introducida tiene menos de 5 caracteres, entre 5 y 15 caracteres o más de 15 caracteres.
- b) Si la cadena introducida comienza por la letra a.

Ejemplo: se pide la entrada y el usuario introduce “vereda”. Por pantalla se debe mostrar: “La cadena introducida tiene entre 5 y 15 caracteres y no comienza por a”.

Puedes comprobar si tus respuestas son correctas consultando en los foros [aprenderaprogramar.com](http://www.aprenderaprogramar.com).

Próxima entrega: CU00658B

Acceso al curso completo en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) -- > Cursos, o en la dirección siguiente:

http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=68&Itemid=188