

CURSO Java Herencia GRATIS APUNTATE!!

El uso de Java 9 JShell puede ser muy socorrido en muchas ocasiones , aunque mucha gente todavía desconoce para qué sirve. JShell es una sencilla consola Java que nos permite ejecutar un bloque de código de una forma rápida y directa . Esta disponible a partir de Java 9 . Vamos a ver unos ejemplos. Para ello es tan sencillo como abrir una consola y escribir JShell con Java 9 instalado en el sistema operativo.

```
| Welcome to JShell -- Version 11.0.1
| For an introduction type: /help intro

jshell> █
```

Java 9 JShell

Esto nos ubicará en la consola y podremos comenzar a escribir comandos Java directamente . El mas sencillo puede ser `System.out.println("hola")`.

```
| Welcome to JShell -- Version 11.0.1
| For an introduction type: /help intro

jshell> System.out.println("hola")
hola

jshell> █
```

Una vez escrito pulsamos intro y el código se ejecutará de forma directa. Esto en principio

nos puede parecer poco útil pero la realidad es que a veces nos puede ayudar a ejecutar pequeños bloques de código y probarlos de una forma muy directa. Imaginemonos que queremos mostrar en una consola el concepto de una compra y el importe de esta . Quizás no nos acordamos de cómo se hacía esta operación con el método printf de Java en System.out. ¿Cómo nos puede ayudar la consola a recordarlo y ejecutar un ejemplo sencillo?. Bueno en principio sería suficiente con declararnos las variables.

```
jshell> String nombre="pedro";  
nombre ==> "pedro"  
  
jshell> double gasto=200.23;  
gasto ==> 200.23  
  
jshell> █
```

Parece que hemos declarado las variables correctamente . Usamos el comando /vars y nos muestra las variables existentes a nivel de jshell en estos momentos.

```
jshell> String nombre="pedro";  
nombre ==> "pedro"  
  
jshell> double gasto=200.23;  
gasto ==> 200.23  
  
jshell> /vars  
| String nombre = "pedro"  
| double gasto = 200.23  
  
jshell> █
```

Es momento de imprimir esos datos por la consola en un texto formateado . ¿Cómo se llamaba el método de System.out? . No hay problema podemos escribir System.out y pulsar

tabulador y Java 9 JShell nos mostrará una ayuda con los métodos existentes.

```
jshell> System.out.print
print(    printf(    println(
jshell> System.out.print
```

Podemos ver cómo nos muestra la lista de métodos que empiezan por print . El que nosotros necesitamos es printf. .Pues nada rellenamos el método y las variables se imprimirán sus valores.

```
jshell> System.out.printf("la persona se llama %s, el gasto es %.2f", nombre,gasto)
la persona se llama pedro, el gasto es 200,23$8 ==> java.io.PrintStream@1f36e637
jshell>
```

JShell también nos puede ayudar a mostrar por ejemplo los valores de las variables de entorno algo muchas veces muy útil.

```
jshell> System.getenv("PATH")
$1 ==> "/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:"
```

Java 9 JShell puede ser una herramienta muy útil en muchas situaciones aportándonos la posibilidad de generar código “sobre la marcha” y ver que es lo que este devuelve.

Otros artículos relacionados

1. [Java Integer Wrapper y certificacion](#)
2. [Java new String y la creación de objetos](#)
3. [Las versiones de Java y su historia](#)
4. [Java 9 JDK](#)

CURSO Introducción Patrones Diseño
GRATIS
APUNTATE!!