

Manual de uso de la Jshell de Java

1) Primero debemos verificar que nuestra versión de jdk sea la versión 9 o superior de lo contrario la jshell no funcionara.

2) Para iniciarla es necesario abrir una terminal y escribir el comando jshell



```

x      x      jshell
+      x      jshell
karinaflo@karinaflo-HP:~/Documents/JavaSemanaUno$ jshell
| Welcome to JShell -- Version 9-internal
| For an introduction type: /help intro
-
Manual de Jshell

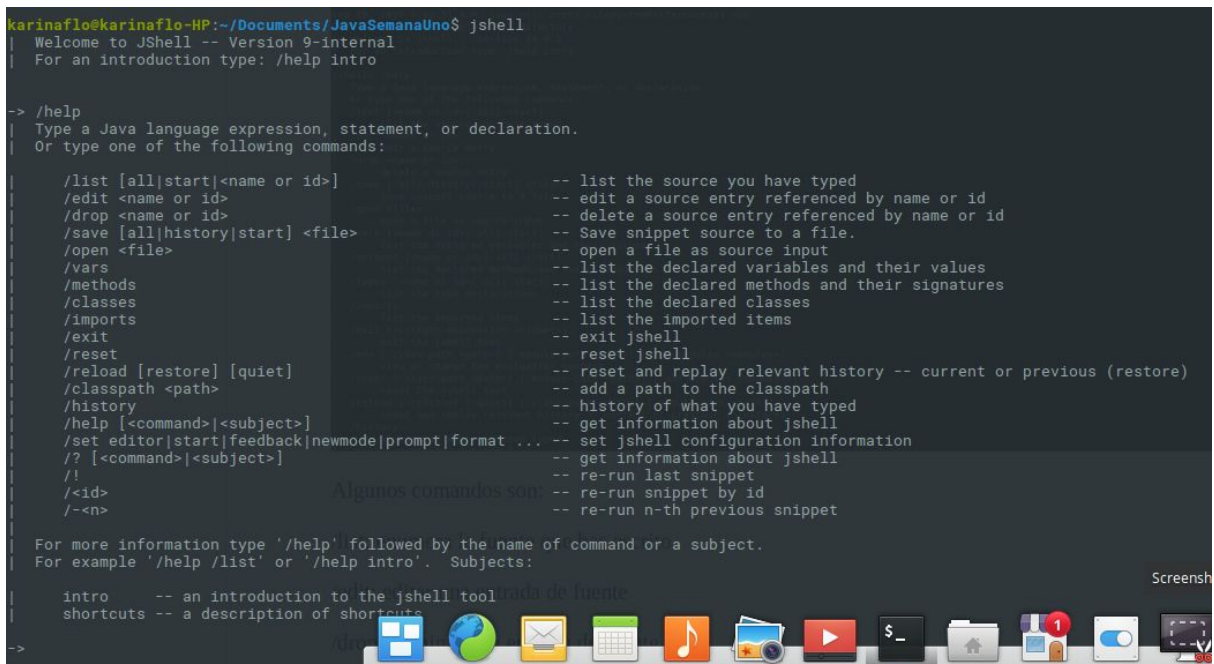
Para acceder al Jshell primero se deberá tener instalado el jdk 9 o superior.
Ya instalado, se puede acceder a través de la línea de comandos con el comando jshell.

Welcome to JShell -- Version 10.0.2

```

Si todo salió bien, ahora veremos un prompt -> esperando una entrada para la jshell.

3) Con el comando /help podemos visualizar las opciones que nos permite ejecutar



```

karinaflo@karinaflo-HP:~/Documents/JavaSemanaUno$ jshell
| Welcome to JShell -- Version 9-internal
| For an introduction type: /help intro
-
/help
Type a Java language expression, statement, or declaration.
Or type one of the following commands:

/!                                -- list the source you have typed
/edit <name or id>                -- edit a source entry referenced by name or id
/drop <name or id>                -- delete a source entry referenced by name or id
/save [all|history|start] <file> -- Save snippet source to a file.
/open <file>                      -- open a file as source input
/vars                            -- list the declared variables and their values
/methods                       -- list the declared methods and their signatures
/classes                         -- list the declared classes
/imports                         -- list the imported items
/exit                            -- exit jshell
/reset                           -- reset jshell
/reload [restore] [quiet]        -- reset and replay relevant history -- current or previous (restore)
/classpath <path>                -- add a path to the classpath
/history                         -- history of what you have typed
/help [<command>|<subject>]      -- get information about jshell
/set editor|start|feedback|newmode|prompt|format ... -- set jshell configuration information
/? [<command>|<subject>]         -- get information about jshell
/!                                -- re-run last snippet
/<id>                             -- re-run snippet by id
/-<n>                             -- re-run n-th previous snippet

Algunos comandos son:

For more information type '/help' followed by the name of command or a subject.
For example '/help /list' or '/help intro'. Subjects:

intro -- an introduction to the jshell
shortcuts -- a description of shortcuts

```

4) Para realizar una impresión en pantalla, se ingresa un SOP igual que en un archivo .java

```
-> System.out.println("Estamos probando la jshell de java");
Estamos probando la jshell de java
->
```

5) Podemos hacer operaciones con números sin la necesidad de declarar variables, ya que la jshell le asignará a la operación una variable por defecto.

```
-> 8+9
| Expression value is: 17
|   assigned to temporary variable $2 of type int
->
```

6) Si realizamos la impresión de esa variable, nos mostrará el resultado de la operación que realizamos.

```
-> 8+9
| Expression value is: 17
|   assigned to temporary variable $2 of type int
-> System.out.println("Prueba de suma: " + $2);
Prueba de suma: 17
->
```

7) También podemos realizar la declaración de variables como comúnmente lo hacemos en un archivo .java

```
-> int num1 = 5;
| Added variable num1 of type int with initial value 5
-> int num2 = 3;
| Added variable num2 of type int with initial value 3
```

8) También podemos declarar funciones.

```
-> int suma(int a, int b){return a+b;}
| Added method suma(int,int)
->
```

9) Si ingresamos / en la jshell, nos mostrará las opciones que podemos visualizar de la Jshell State, que es una clase virtual de la jshell que nos permite trabajar en ella.

```
-> /
/?          /classes  /classpath  /drop       /edit       /exit       /help
/history    /imports    /list       /methods    /open       /reload     /reset     /save
/set        /vars
-> /
```

10) Si escribimos /vars nos mostrará la lista de variables declaradas en el sistema así como su valor.

Si escribimos /methods nos mostrará los métodos declarados por nosotros y uno por defecto que es el printf(String,Object...)void, este es el método que nos permite visualizar impresiones en la consola cada vez que realizamos una acción, sin necesidad de tener que ejecutar un SOP.

```
-> /vars
|   int num1 = 5
|   int num2 = 3
-> /methods
|   printf (String,Object...)void
|   suma (int,int)int
->
```

11) Si declaramos otra variable para que se guarde lo que nos retorne la función que declaramos e imprimimos su valor, todo funcionara perfecto.

```
-> System.out.println("El resultado de la suma es: " + resultado);
El resultado de la suma es: 8
->
```

Entonces visualizamos que las sentencias que realizamos en un documento .java normalmente para la realización de un código son iguales en la jshell, aunque la jshell proporciona algunas otras ventajas como la impresión inmediata de las acciones que realizamos.

12) Para salir de la jshell es suficiente con el comando /exit

```
-> /exit
| Goodbye
```