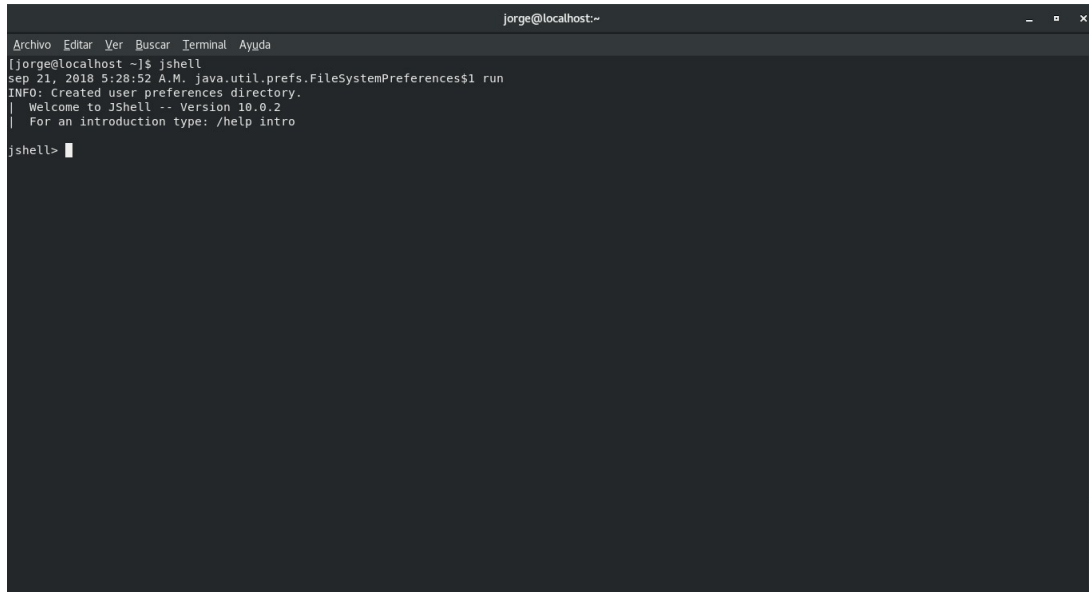


Jshell

Requisitos para ejecutar jshell: Tener jdk 9 en adelante

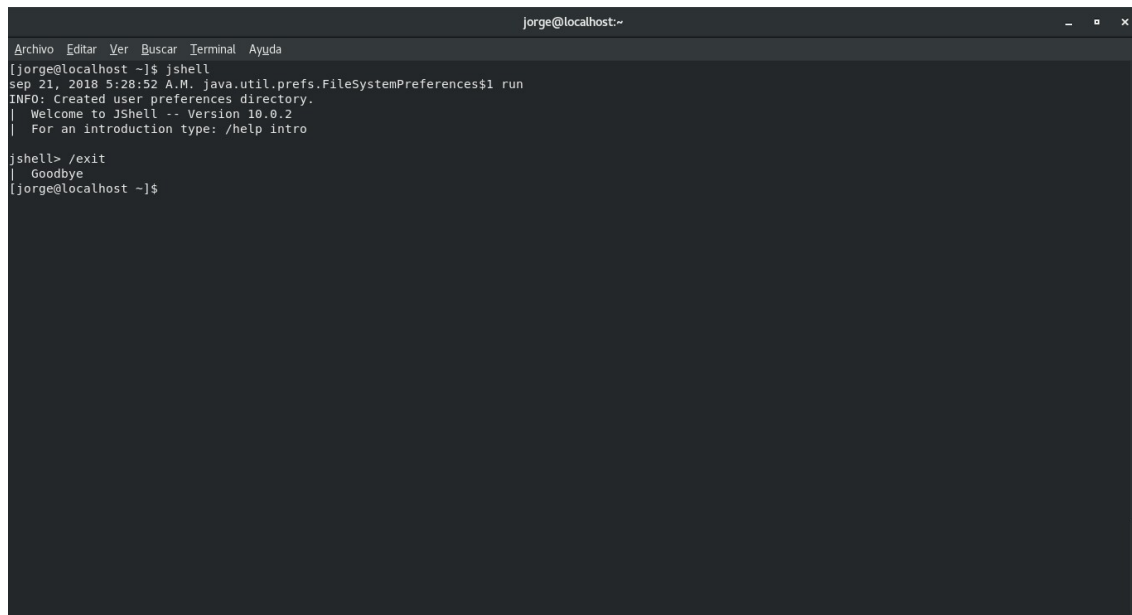
Para iniciar la terminal se usa el comando jshell, si no funciona es porque no se tiene la versión necesaria de jdk.



```
jorge@localhost:~$ jshell
sep 21, 2018 5:28:52 A.M. java.util.prefs.FileSystemPreferences$1 run
INFO: Created user preferences directory.
| Welcome to JShell -- Version 10.0.2
| For an introduction type: /help intro
jshell> 
```

Para utilizar los comandos se utiliza diagonal y comando, ejemplo “/help”

Para salir de la terminal, se utiliza el comando /exit.



```
jorge@localhost:~$ jshell
sep 21, 2018 5:28:52 A.M. java.util.prefs.FileSystemPreferences$1 run
INFO: Created user preferences directory.
| Welcome to JShell -- Version 10.0.2
| For an introduction type: /help intro
jshell> /exit
| Goodbye
jorge@localhost:~$ 
```

* 5 comandos de Jshell

Comandos:

- 1- `"/help [<command>|<subject>]`: muestra los comandos que tiene la jshell
- 2- `"/list [<name or id>|-all|-start]"` muestra las fuentes que has escrito
- 3- `"/history"`: muestra los comandos que has usado previamente
- 4- `"/methods [<name or id>|-all|-start]"`: muestra los métodos declarados y sus firmas
- 5- `"/exit [<integer-expression-snippet>]"`: Sale de la jshell

* Forward references

JShell acepta definiciones de métodos que hacen referencia a métodos, variables o clases que aún no están definidas. Esto se hace para apoyar la programación exploratoria y porque algunas formas de programación lo requieren.

Un ejemplo:

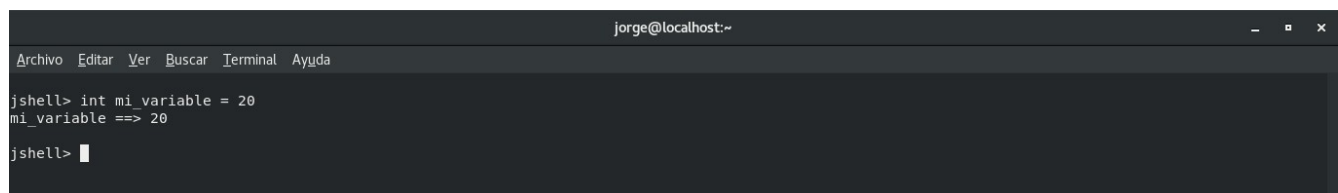
```
jshell> double volume(double radius) {  
    ...> return 4.0 / 3.0 * PI * cube(radius);  
    ...> }
```

// Volumen de método creado (doble), sin embargo, no se puede invocar hasta que se hayan declarado la PI y el cubo del método (doble).

* Para declarar variables

Para declarar una variable se declara el tipo de variable, nombre de la variable y el valor.

Ejemplo:

A screenshot of a JShell terminal window. The title bar shows 'jorge@localhost:~'. The menu bar includes 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The terminal content shows the command 'jshell> int mi_variable = 20' followed by the output 'mi_variable ==> 20'. The prompt 'jshell>' is followed by a cursor.

```
jshell> int mi_variable = 20  
mi_variable ==> 20  
jshell> █
```

* Para declarar métodos

No se necesita preocuparse por ningún modificador, como public, static o final. Simplemente comienza con el tipo de devolución y el nombre del método con posibles parámetros:

```
jorge@localhost:~  
_ □ ×  
Archivo Editar Ver Buscar Terminal Ayuda  
jshell> String diHola(String nombre) { return "Hola, mi nombre es "+nombre  
;}  
| modified method diHola(String)  
jshell> diHola("Jorge")  
$5 ==> "Hola, mi nombre es Jorge"  
jshell> █
```

* Snippets

- if
- if else
- else
- while
- for

- trycatch
- tryfinally