Java Shell Tool (JShell)

It is an interactive Java Shell tool, it allows us to execute Java code from the shell and shows output immediately. JShell is a REPL (Read Evaluate Print Loop) tool and run from the command line.

Advantages of JShell

Jshell has reduced all the efforts that are required to run a Java program and test a business logic.

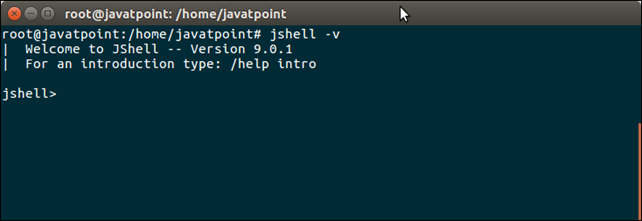
If we don't use Jshell, creating of Java program involves the following steps.

* Open editor and write program
* Save the program
* Compile the program
* Edit if any compile time error
* Run the program
* Edit if any runtime error
* Repeat the process

Jshell does not require above steps. We can evaluate statements, methods and classes, even can write hello program without creating class.

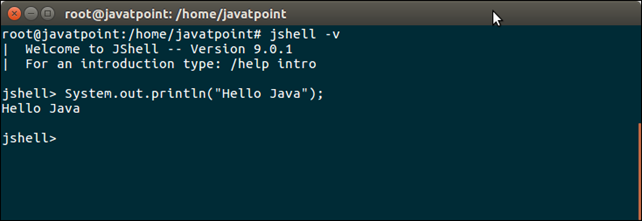
How to Start JShell

To start Jshell, first we must have installed Java 9 then open terminal in Linux or command prompt in windows and type **jshell ?v**. It will start jshell session and displays a welcome message to the console.



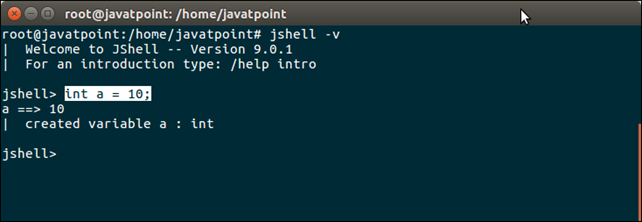
Hello Java Message

To display a simple "Hello Java" message, write print command without creating class and hit enter.

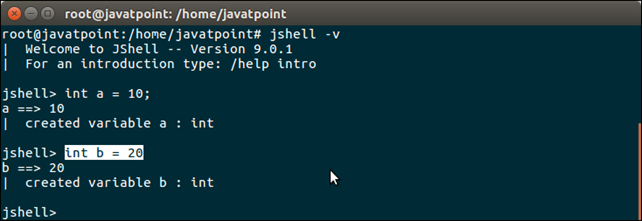


Variables

We can declare variables and use anywhere throughout Jshell session. Let's create an integer variable.

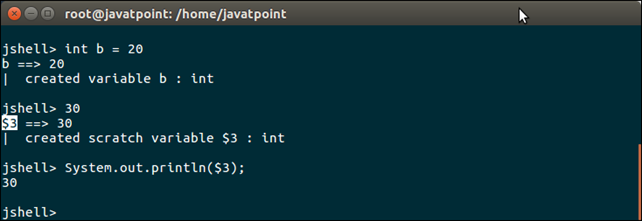


Semicolon (;) is optional, we can leave it and it works fine. See, variable **b** is created without using semicolon.



Scratch Variables

If we don't provide variable name, Java create implicit variable to store the value. These variables start with $ sign. We can use these variable by specifying implicit variable, as we did in the in the following screen-shot.



Expressions

We can test any valid Java expression to get instant output. See, the following example.

**Adding two integers**

1. jshell> 2+3
2. $1 ==> 5
3. |  created scratch variable $1 : **int**
4. jshell>

**Compound expression**

1. jshell> 2+(5\*4+(2+1))
2. $4 ==> 25
3. |  created scratch variable $4 : **int**
4. jshell>

Methods

To test method business logic, create an method and get result immediately. See, the following example.

1. jshell> **void** show(){
2. ...> System.out.println("This is show method");
3. ...> }
4. |  created method show()

**Calling method**

1. jshell> show();
2. This is show method

To create class, write source code for the class and call its method by creating object immediately. See the following example.

Class

1. jshell> **class** Hello{
2. ...> **void** show(){
3. ...> System.out.println("This is show method");
4. ...> }
5. ...> }
6. |  created **class** Hello
8. jshell> **new** Hello().show();
9. This is show method

Package Imports

By default, 10 packages are imported and can also be imported any package by using import statement.

To see, default import packages, we can use following command.

1. jshell> /**import**
2. |    **import** java.io.\*
3. |    **import** java.math.\*
4. |    **import** java.net.\*
5. |    **import** java.nio.file.\*
6. |    **import** java.util.\*
7. |    **import** java.util.concurrent.\*
8. |    **import** java.util.function.\*
9. |    **import** java.util.prefs.\*
10. |    **import** java.util.regex.\*
11. |    **import** java.util.stream.\*

**Importing java.sql package.**

1. jshell> **import** java.sql.\*;

Listing import packages and it will show available accessible packages.

1. jshell> /**import**
2. |    **import** java.io.\*
3. |    **import** java.math.\*
4. |    **import** java.net.\*
5. |    **import** java.nio.file.\*
6. |    **import** java.util.\*
7. |    **import** java.util.concurrent.\*
8. |    **import** java.util.function.\*
9. |    **import** java.util.prefs.\*
10. |    **import** java.util.regex.\*
11. |    **import** java.util.stream.\*
12. |    **import** java.sql.\*

Now number of packages are 11 including new one **java.sql.\*.**

Jshell Commands

Jshell provides various useful commands that we can use to modify environment, manage code and to get code related information. Following are the useful information.

**Package Imports**

1. jshell> /**import**
2. |    **import** java.io.\*
3. |    **import** java.math.\*
4. |    **import** java.net.\*
5. |    **import** java.nio.file.\*
6. |    **import** java.util.\*
7. |    **import** java.util.concurrent.\*
8. |    **import** java.util.function.\*
9. |    **import** java.util.prefs.\*
10. |    **import** java.util.regex.\*
11. |    **import** java.util.stream.\*
12. |    **import** java.sql.\*

Command /**vars** to show variables.

1. jshell> /vars
2. |    **int** $1 = 5
3. |    **int** $2 = 5
4. |    **int** $4 = 25

**To get all written source code, use /list**

1. jshell> /list
3. 1 : 2+3
4. 2 : 10-5
5. 3 : $1
6. 4 : 2+(5\*4+(2+1))
7. 5 : **void** show(){
8. System.out.println("This is show method");
9. }
10. 6 : show();
11. 7 : **class** Hello{
12. **void** show(){
13. System.out.println("This is show method");
14. }
15. }
16. 8 : **new** Hello().show();
17. 9 : **import** java.sql.\*;