Prepared By : Rutu Shah

What is a Groovy Script?

APACHE GROOVY is an object oriented and Java syntax compatible programming language built for the Java platform. This dynamic language has many features which are similar to Python, Ruby, Smalltalk, and Pero. Groovy source code gets compiled into Java Bytecode so it can run on any platform that has JRE is installed. Groovy also performs a lot of tasks behind the scene that makes it more agile and dynamic.

Groovy can be used as a scripting language for the Java platform. It is almost like a super version of Java which offers Java's enterprise capabilities. It also offers many productivity features like DSL support, closures, and dynamic typing. Unlike some other languages, it is designed as a companion, not a replacement for Java.

Why Groovy?

Here, are major reasons why you should use Groovy

- Groovy is an agile and dynamic language
- Seamlessly integration with all existing Java objects and libraries
- Feels easy and natural to Java developers
- More concise and meaningful code compares to Java
- You can use it as much or as little as you like with Java apps

Groovy History

- 2003: Developed by Bob McWhirter & James Strachan
- 2004: Commissioned into JSR 241 but it was abandoned
- 2005: Brought back by Jeremy Rayner & Guillaume Laforge
- 2007: Groovy version 1.0
- 2012: Groovy version 2
- 2014: Groovy version 2.3 (official support given for JDK 8)
- 2015: Groovy became a project at the Apache Software Foundation

Features of Groovy

- List, map, range, regular expression literals
- Multimethod and metaprogramming
- Groovy classes and scripts are usually stored in .groovy files
- Scripts contain Groovy statements without any class declaration.
- Scripts can also contain method definitions outside of class definitions.
- It can be compiled and fully integrated with traditional Java application.
- Language level support for maps, lists, regular expressions
- Supports closures, dynamic typing, metaobject protocol
- Support for static and dynamic typing & operator overloading
- Literal declaration for lists (arrays), maps, ranges, and regular expressions

Groovy Vs. Java

Groovy	**Java**
In Groovy, default access specifier is public. It means a method without any specified access modifier is public and accessible outside of class and package boundaries.	In Java, the default access modifier is a package, i.e., if you don't specify access modifier for fields, methods or class it becomes package-private,
Getters and setters are automatically generated for class members.	Java, you need to define getters and setters method for fields
Groovy allows variable substitution using double quotation marks with strings.	Java does not support variable substitution.
Typing information is optional.	Typing information is mandatory in Java.

Groovy it's not required to end with a semicolon.	In Java, every statement ends with a semicolon.
Groovy is automatically a wrapping class called Script for every program	In Java, you need the main method to make a class executable.

Myths about Groovy

Myth	**Reality**
We can use Groovy just for scripting.	It can be used for scripting. However, you can perform many other tasks apart from it.
Groovy is all about closures. "It's just functional programming language."	Groovy adopts from functional programming languages like Lisp or Closure.
Groovy is an ideal choice if you want do TDD	This statement is true. However, it is certainly not the only reason to use Groovy.
You can use Groovy only if you want to use Grails.	Grails is a powerful web development framework. But Groovy offers more than that.

Cons of using Groovy

- JVM and Groovy script start time is slow which limits OS-level scripting
- Groovy is not entirely accepted in other communities.
- It is not convenient to use Groovy without using IDE
- Groovy can be slower which increased the development time
- Groovy may need lots of memory
- Knowledge of Java is imperative.

Groovy Tools

We will discuss 3 important tools

1. groovysh: Executes code interactively.

2. groovyConsole: GUI for interactive code execution

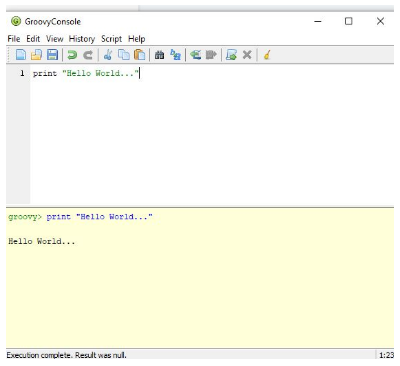
3. groovy: Executes groovy scripts. You can use it like Perl, Python, etc.

Prerequisites before installing groovy.

1. Make sure you have java installed in your system, you can easily check java version in cmd using command
   Java -version

2. Go to http://groovy-lang.org/download.html
3. Select windows installer and install groovy also download entire groovy package which will be useful for configuring groovy in ide.

Printing first groovy program in groovy console.



```
print "Hello World..."
```

```
groovy> print "Hello World..."

Hello World...
```

Execution complete. Result was null. 1:23

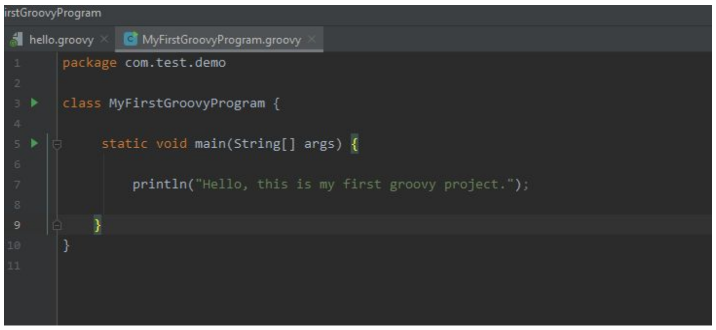
Groovy in IDE (intellije)

To start groovy with intellije idea below is the given link.

https://www.logicbig.com/tutorials/misc/groovy/intellij.html

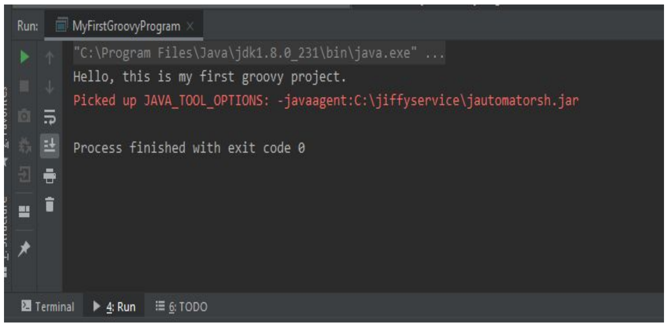
Steps to create groovy class

1. After creating and adding groovy configuration as mentioned in above link in ide,create a new package com.test.demo and inside that create a groovy Class named MyFirstGroovyProgram.



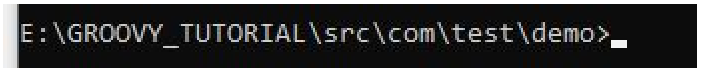
```
package com.test.demo

class MyFirstGroovyProgram {

    static void main(String[] args) {

        println("Hello, this is my first groovy project.");

    }
}
```

2. Click on Run button and see output



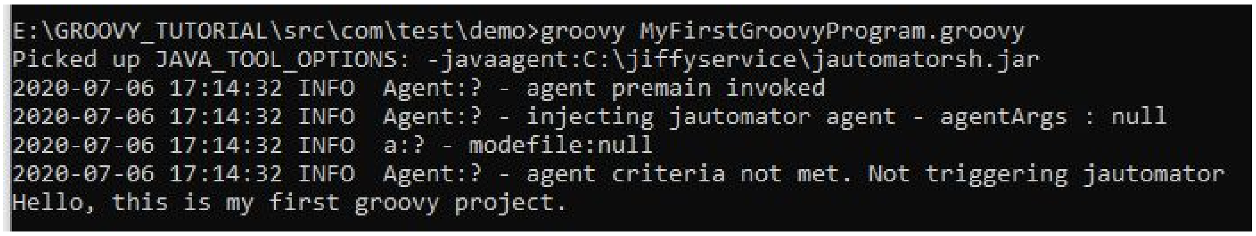
```
"C:\Program Files\Java\jdk1.8.0_231\bin\java.exe" ...
Hello, this is my first groovy project.
Picked up JAVA_TOOL_OPTIONS: -javaagent:C:\jiffyservice\jautomatorsh.jar

Process finished with exit code 0
```

We can even run this groovy program through command line
Copy the path of this particular class and open in cmd



```
E:\GROOVY_TUTORIAL\src\com\test\demo>_
```

3. Then enter the following command
groovy MyFirstGroovyProgram.groovy

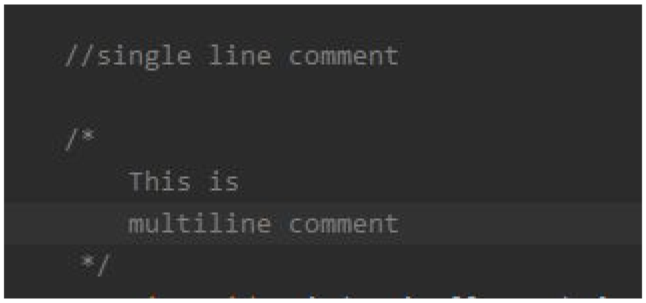


```
E:\GROOVY_TUTORIAL\src\com\test\demo>groovy MyFirstGroovyProgram.groovy
Picked up JAVA_TOOL_OPTIONS: -javaagent:C:\jiffyservice\jautomatorsh.jar
2020-07-06 17:14:32 INFO  Agent:? - agent premain invoked
2020-07-06 17:14:32 INFO  Agent:? - injecting jautomator agent - agentArgs : null
2020-07-06 17:14:32 INFO  a:? - modefile:null
2020-07-06 17:14:32 INFO  Agent:? - agent criteria not met. Not triggering jautomator
Hello, this is my first groovy project.
```

Note : In groovy semicolon and braces are not mandatory, i.e we can even run a groovy program without semicolon.

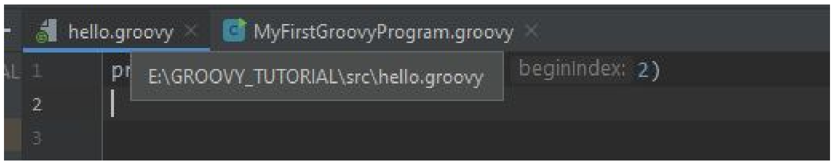
Note : System.out.println is reduced to println

In Groovy comments are similar to java

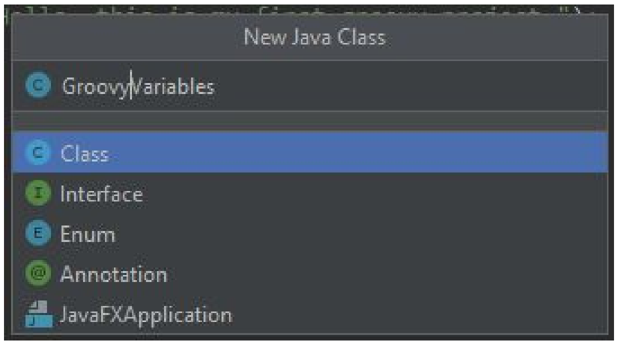


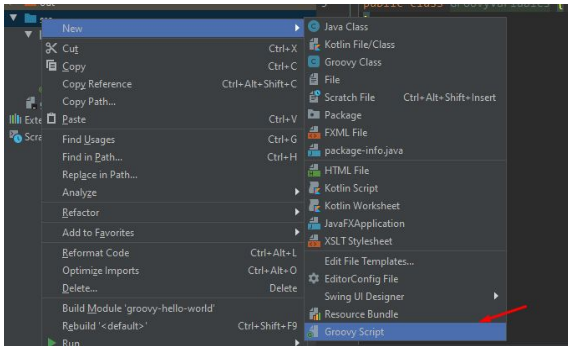
```
//single line comment

/*
    This is
    multiline comment
 */
```

Also, in groovy class and functions are not necessary, you can simply run your groovy script by writing your code using groovy script


hello.groovy
MyFirstGroovyProgram.groovy
E:\GROOVY_TUTORIAL\src\hello.groovy
beginIndex: 2)

In groovy we can not only create groovy class, groovy interface, enum, annotation and JavaFXApplication but also it allows us to create groovy script as well.


New Java Class
GroovyVariables
Class
Interface
Enum
Annotation
JavaFXApplication


New
Cut Ctrl+X
Copy Ctrl+C
Copy Reference Ctrl+Alt+Shift+C
Copy Path...
Paste Ctrl+V
Find Usages Ctrl+G
Find in Path... Ctrl+H
Replace in Path...
Analyze
Refactor
Add to Favorites
Reformat Code Ctrl+Alt+L
Optimize Imports Ctrl+Alt+O
Delete... Delete
Build Module 'groovy-hello-world'
Rebuild '<default>' Ctrl+Shift+F9
Run
Java Class
Kotlin File/Class
Groovy Class
File
Scratch File Ctrl+Alt+Shift+Insert
Package
FXML File
package-info.java
HTML File
Kotlin Script
Kotlin Worksheet
JavaFXApplication
XSLT Stylesheet
Edit File Templates...
EditorConfig File
Swing UI Designer
Resource Bundle
Groovy Script