Jenkins

Jeeva S. Chelladhurai
CEO, Comorin Consulting Services
+91 97319 77222
jeeva@comorin.co

9. Script Supported in Jenkins

Introduction

- Known as Java Scripting Language
- JVM based Dynamic Language
- Java like syntax but semicolon is optional
- Scripting language for Java Platform
- Apache License v2.0

Groovy “Hello World”

println “Hello, World”

```
public class HelloWorld {
        public static void main(String[] args) {
                System.out.println("Hello, World");
        }
}
```

History

- **2003**: Started by James Strachan and Bob McWhirter
- **2004**: Commision into JSR 241 but was almost abandoned
- **2005**: Brought back to life by Guillaume Laforge and Jeremy Rayner
- **2007**: Groovy Version 1.0
- **2012**: Groovy Version 2.0
- **2014**: Groovy Version 2.3(Official support for JDK 8)

Installing Groovy

- Groovy 2.4
  - JDK7+
  - Stable
- Docker
  - docker run --rm -it groovy:2.4.12
- http://groovy-lang.org/download.html

Revisiting Hello World!

```
$ cat helloworld.groovy
class HelloWorld {
        static void main(String[] args) {
                System.out.println("Hello, World");
        }
}
$ groovy helloworld.groovy
Hello World!!
```

Using alias> groovy-run helloworld.groovy

Implicit Import

```
import java.lang.*
import java.util.*
import java.io.*
import java.net.*
import groovy.lang.*
import groovy.util.*
import java.math.BigInteger
import java.math.BigDecimal
```

Comments

- Singleline

```
// my comment
```

- Multiline

```
/* This is my multiline comment
   This is next line */
```

Built-in Data Types

- Integer datatype
  - represents whole number
  - int x = 8;
- Long datatype
  - represents long number
  - long y = 200L;
- Floating point datatype
  - represents 32 bit floating point number
  - float a = 10.86f;
- Double datatype
  - represents 64 bit floating point number
  - double b = 10.6e40;
- BigInteger datatype
  - Immutable arbitrary –precision signed integral numbers
  - BigInteger bi = 40g;
- BigDecimal datatype
  - Immutable arbitrary-precision signed decimal numbers
  - BigDecimal bd = 2.5g;

Variables

- Variable can be composed of letters ,digits and underscore character.
- Groovy is a case sensitive programming language.
- Variable in lower case
  - int x = 8;
- Variable in uppercase
  - Int X = 10;
- Variable with underscore in its name
  - Def _Name = “Kate”;

Operators: Arithmetic

Operator	Description	Example
+	Addition	3 + 5 gives 8
_	Subtraction	5 - 3 gives 2
*	Multiplication	2 * 4 gives 8
/	Division	5 / 2 gives 2 . 5
%	Modulus	5 % 2 gives 1

++	Increment	int x = 4;
		x++;
		x gives 5

--	Decrement	Int x = 4; x -- ; x gives 3

Operators: Relational

Operator	Description	Example
= =	Checks for equality between two objects	3 = = 3 gives true
! =	Checks the difference between two objects	3 != 5 gives true
<	Checks if the left object is less than the right	3<4 gives true
<=	Check if the left object is less than or equal to the right	3<=4 gives true
>	Check if the left object is greater than the right.	4>2 gives true
>=	Check if the left object is greater than or equal to the right	4>=2 gives true

Operator: Bitwise

Description	
Operator	Description
&	“and” operator
|	“or” operator
^	“xor” operator
~	negation operator

Truth Table				
p	q	p & q	p | q	p ^ q
0	0	0	0	0
0	1	0	1	1
1	1	1	1	0
1	0	0	1	1

Operators: Assignment

Operator	Description	Example
+ =	Adds both the operands and assigns the result to the left operand	def B = 8 B += 1 Output is 9
– =	Subtracts both the operands and assigns the result to the right operand	def B = 8 B – = 1 Output is 7
* =	Multiplies both the operands and assigns the result to the left operand	def B = 8 B *= 2 Output is 16
/ =	Divides and assigns the value to the left operand	def B = 8 B /= 2 Output is 4
%=	Modulus of the operands and assigns the value to the left operand	def B = 8 B %= 3 Output is 2

Operator: Range

def range = 6..9

Defines a range of integers stored in a local variable called range with lower bound 6 and upper bound 9

Loops: While statement

```
While(condition) {
    statement 1
    statement 2
    ……
}
```

Loops: For Statement

```
for(variable declaration ; expression ; Increment) {
    statement 1
    statement 2
    ...
}
```

Decision Making

- One or more conditions to be evaluated or tested by the program
- Along with a statement or statements to be executed if the condition is determined to be **true**
- Other statements to be executed if the condition is determined to be **false**

Decision Making Types

Sr.No.	Statements & Description
1	**if Statement**
2	**if/else Statement**
3	**Nested If Statement**
4	**Switch Statement**
5	**Nested Switch Statement**