**Operators in groovy**

In groovy, operators are symbols which are used to tell the compiler to perform specified operations.

Following are the operators in groovy:

* Arithmetic operators
* Unary operators
* Assignment arithmetic operators
* Relational operators
* Logical operators
* Bitwise operators
* Conditional operators

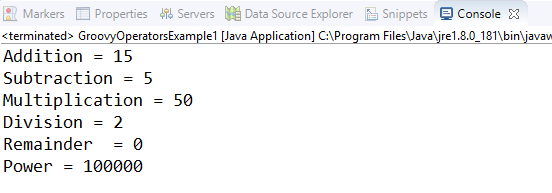
Arithmetic operator

Arithmetic operators are the basic mathematical operators, which are used to perform addition, subtraction, multiplication, division, Remainder and Power.

Example 1:

1. **package** com.app
2. **class** GroovyOperatorsExample1 {
3. **static** **void** main(args) {
4. **int** a = 10
5. **int** b = 5
6. **int** c
7. c = a + b
8. println "Addition = " + c
9. c = a - b
10. println "Subtraction = " + c
11. c = a \* b
12. println "Multiplication = " + c
13. c = a / b
14. println "Division = " + c
15. c = a % b
16. println "Remainder  = " + c
17. c = a \*\* b
18. println "Power = "+c
19. }
20. }

**Output:**

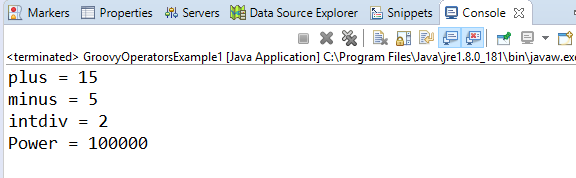


In groovy, we also have some functions which are used to perform Arithmetic operations like plus, minus, intdiv and power. The use of these functions are shown in the example which is given below.

Example 2:

1. **package** com.app
2. **class** GroovyOperatorsExample2 {
3. **static** **void** main(args) {
4. **int** a = 10.3
5. **int** b = 5
6. **int** c
7. c = a.plus(b)
8. println "plus = " + c
9. c = a.minus(b)
10. println "minus = " + c
11. c = a.intdiv(b)
12. println "intdiv = " + c
13. c = a.power(b)
14. println "Power = "+c
15. }
16. }

**Output:**



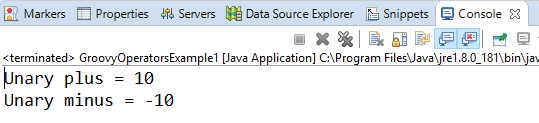
Unary operators

In groovy, Unary operators require only one operator to perform the operation. Unary operators are used to perform the operations such as increment/decrement, negating, and inverting the values of a Boolean.

Example 3:

1. **package** com.app
2. **class** GroovyOperatorsExample3 {
3. **static** **void** main(args) {
4. **int** a = 10
5. **int** c
6. c = +a
7. println "Unary plus = " + c
8. c = -a
9. println "Unary minus = " + c
11. }
12. }

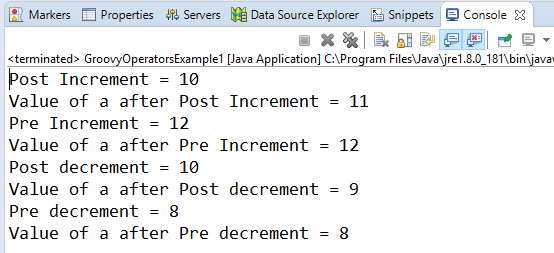
**Output:**



Example 4:

1. **package** com.app
2. **class** GroovyOperatorsExample4 {
3. **static** **void** main(args) {
4. **int** a = 10
5. **int** c
6. c = a++
7. println "Post Increment = " + c
8. println "Value of a after Post Increment = " + a
9. c = ++a
10. println "Pre Increment = " + c
11. println "Value of a after Pre Increment = " + a
12. **int** b = 10
13. c = b--
14. println "Post decrement = " + c
15. println "Value of a after Post decrement = " + b
16. c = --b
17. println "Pre decrement = " + c
18. println "Value of a after Pre decrement = " + b
19. }
20. }

**Output:**



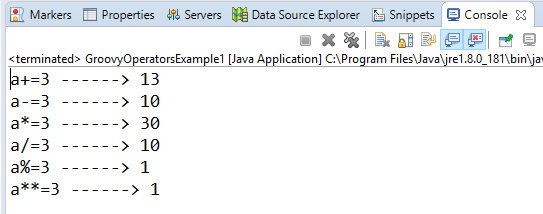
Assignment arithmetic operators

In groovy, assignment arithmetic operators are used to assign a new value to the variable.

Example 5:

1. **package** com.app
2. **class** GroovyOperatorsExample5 {
3. **static** **void** main(args) {
4. **int** a = 10
5. a+=3
6. println "a+=3 ------> " + a
7. a-=3
8. println "a-=3 ------> " + a
9. a\*=3
10. println "a\*=3 ------> " + a
11. a/=3
12. println "a/=3 ------> " + a
13. a%=3
14. println "a%=3 ------> " + a
15. a\*\*=3
16. println "a\*\*=3 ------> " + a
17. }
18. }

**Output:**



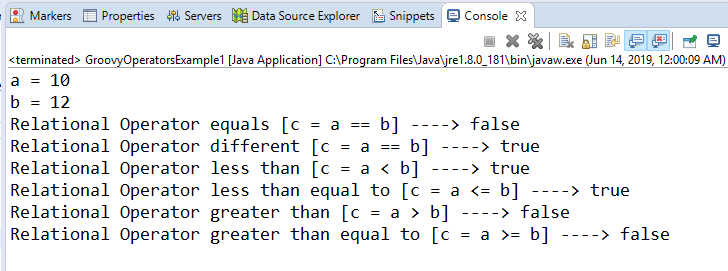
Relational operators

In groovy, relational operators are used to compare two objects to check wether they are same or different or one is greater than, less than or equal to other object.

Example 6:

1. **package** com.app
2. **class** GroovyOperatorsExample6 {
3. **static** **void** main(args) {
4. **int** a = 10
5. **int** b = 12
6. **boolean** c
7. println "a = 10"
8. println "b = 12"
9. c = a == b
10. println "Relational Operator equals [c = a == b] ----> " + c
11. c = a != b
12. println "Relational Operator different [c = a == b] ----> " + c
13. c = a < b
14. println "Relational Operator less than [c = a < b] ----> " + c
15. c = a <= b
16. println "Relational Operator less than equal to [c = a <= b] ----> " + c
17. c = a > b
18. println "Relational Operator greater than [c = a > b] ----> " + c
19. c = a >= b
20. println "Relational Operator greater than equal to [c = a >= b] ----> " + c
22. }
23. }

**Output:**



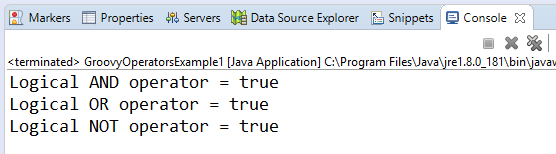
Logical operators

In groovy, there are 3 logical operators for Boolean expression, and these operators are AND(&&), OR(||) and NOT(!)

Example 7:

1. **package** com.app
2. **class** GroovyOperatorsExample7 {
3. **static** **void** main(args) {
4. **boolean** c
5. c = **true** && **true**
6. println "Logical AND operator = " + c
7. c = **true** || **false**
8. println "Logical OR operator = " + c
9. c = !**false**
10. println "Logical NOT operator = " + c
12. }
13. }

**Output:**

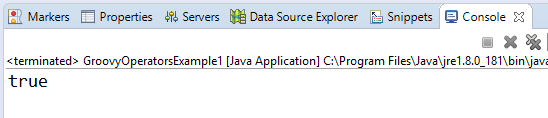


*Note: In groovy, logical "not" is having a higher priority as compared to the logical "and".*

Example 8:

1. **package** com.app
2. **class** GroovyOperatorsExample8 {
3. **static** **void** main(args) {
4. **boolean** c
5. c = (!**false** && **false**)
6. println  c
7. }
8. }

**Output:**

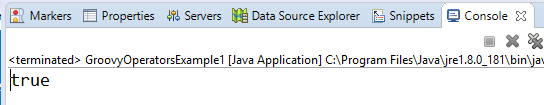


*Note: In groovy, logical "and" is having a higher priority as compared to the logical "or".*

Example 9:

1. **package** com.app
2. **class** GroovyOperatorsExample1 {
3. **static** **void** main(args) {
4. **boolean** c
5. c = **true** || **true** && **false**
6. println  c
7. }
8. }

**Output:**



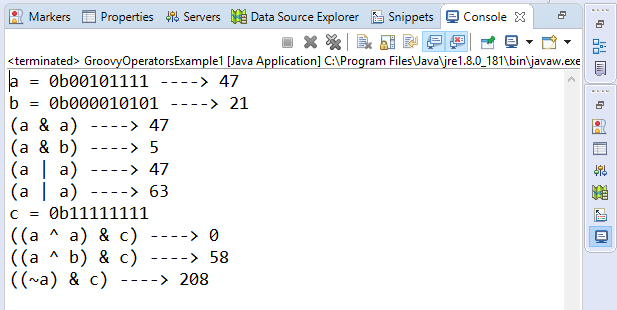
Bitwise operators

In groovy, Bitwise operators are used for operating on binary digits or bits of an integer.

Example 10:

1. **package** com.app
2. **class** GroovyOperatorsExample10 {
4. **static** **void** main(args) {
5. **int** a = 0b00101111
6. println "a = 0b00101111 ----> "+a
7. **int** b = 0b000010101
8. println "b = 0b000010101 ----> "+b
9. println "(a & a) ----> "+(a & a)
10. println "(a & b) ----> "+(a & b)
11. println "(a | a) ----> "+(a | a)
12. println "(a | a) ----> "+(a | b)
14. **int** c = 0b11111111
15. println "c = 0b11111111"
16. println "((a ^ a) & c) ----> "+((a ^ a) & c)
17. println "((a ^ b) & c) ----> "+((a ^ b) & c)
18. println "((~a) & c) ----> "+((~a) & c)
19. }
20. }

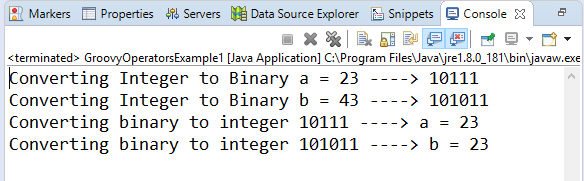
**Output:**



Example 11:

1. **package** com.app
2. **class** GroovyOperatorsExample11 {
3. **static** **void** main(args) {
4. **int** a = 23
5. **int** b = 43
6. println "Converting Integer to Binary a = 23 ----> " + Integer.toBinaryString(a)
7. println "Converting Integer to Binary b = 43 ----> " +Integer.toBinaryString(b)
8. println "Converting binary to integer 10111 ----> a = " + Integer.parseInt("10111", 2)
9. println "Converting binary to integer 101011 ----> b = " + Integer.parseInt("10111",2)
10. }
11. }

**Output:**



Conditional operators

In groovy, there are three types of conditional operators they are as follow:

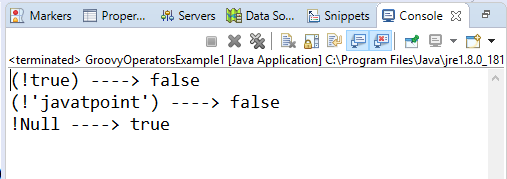
* **Not operator**

In groovy, "not" operator is used invert the result of the Boolean expression.

Example 12:

1. **package** com.app
2. **class** GroovyOperatorsExample12 {
3. **static** **void** main(args) {
4. println "(!true) ----> "+(!**true**)
5. println "(!'javatpoint') ----> "+(!'javatpoint')
6. println "!Null ----> "+(!'')
7. }
8. }

**Output:**



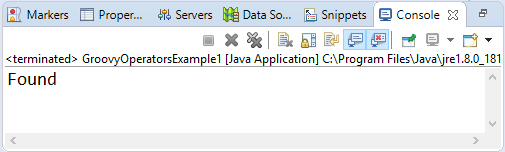
* **Ternary operator**

In groovy, Ternary Operator is the shortcut of if/else

Example 13:

1. **package** com.app
2. **class** GroovyOperatorsExample13 {
3. **static** **void** main(args) {
4. String Answer
5. String s = 'javatpoint'
6. Answer = (s!=**null** && s.length()>0) ? 'Found' : 'Not found'   }
7. }

**Output:**



* **Elvis operator**

In groovy, Elvis operator is a shorthand property of the ternary operator. It only returns when a value is true.

Example 14:

1. **package** com.app
2. **class** GroovyOperatorsExample1 {
3. **static** **void** main(args) {
4. String Answer
5. String s = 'javatpoint'
6. println Answer = s ? 'Found' : 'Not Found'
7. println Answer = s ?: 'Found'
8. }
9. }

**Output:**

