

Groovy

```
groovy> package com.app
groovy> class Demo {
groovy> static void main(args) {
groovy> print("Welcome to Javatpoint tutorial on Groovy... ")
groovy> }
groovy> }
```

Welcome to Javatpoint tutorial on Groovy...

```
groovy> package com.app
groovy> class GroovyOperatorsExample1 {
groovy> static void main(args) {
groovy>     int a = 10
groovy>     int b = 5
groovy>     int c
groovy>     c = a + b
groovy>     println "Addition = " + c
groovy>     c = a - b
groovy>     println "Subtraction = " + c
groovy>     c = a * b
groovy>     println "Multiplication = " + c
groovy>     c = a / b
groovy>     println "Division = " + c
groovy>     c = a % b
groovy>     println "Remainder = " + c
groovy>     c = a ** b
groovy>     println "Power = "+c
groovy> }
groovy> }
```

Addition = 15

Subtraction = 5

Multiplication = 50

Division = 2

Remainder = 0

Power = 100000

```

groovy> package com.app
groovy> class GroovyOperatorsExample2 {
groovy> static void main(args) {
groovy>     int a = 10.3
groovy>     int b = 5
groovy>     int c
groovy>     c = a.plus(b)
groovy>     println "plus = " + c
groovy>     c = a.minus(b)
groovy>     println "minus = " + c
groovy>     c = a.intdiv(b)
groovy>     println "intdiv = " + c
groovy>     c = a.power(b)
groovy>     println "Power = "+c
groovy> }
groovy> }

```

```

plus = 15
minus = 5
intdiv = 2
Power = 100000

```

Unary operators

```

groovy> package com.app
groovy> class GroovyOperatorsExample3 {
groovy> static void main(args) {
groovy>     int a = 10
groovy>     int c
groovy>     c = +a
groovy>     println "Unary plus = " + c
groovy>     c = -a
groovy>     println "Unary minus = " + c
groovy> }
groovy> }

```

```

Unary plus = 10
Unary minus = -10

```

```

groovy> package com.app
groovy> class GroovyOperatorsExample4 {
groovy> static void main(args) {
groovy>     int a = 10
groovy>     int c
groovy>     c = a++
groovy>     println "Post Increment = " + c
groovy>     println "Value of a after Post Increment = " + a
groovy>     c = ++a
groovy>     println "Pre Increment = " + c
groovy>     println "Value of a after Pre Increment = " + a
groovy>     int b = 10
groovy>     c = b--
groovy>     println "Post decrement = " + c
groovy>     println "Value of a after Post decrement = " + b
groovy>     c = --b
groovy>     println "Pre decrement = " + c
groovy>     println "Value of a after Pre decrement = " + b
groovy> }
groovy> }

```

```

Post Increment = 10
Value of a after Post Increment = 11
Pre Increment = 12
Value of a after Pre Increment = 12
Post decrement = 10
Value of a after Post decrement = 9
Pre decrement = 8
Value of a after Pre decrement = 8

```

Assignment Arithmetic operators

```

groovy> package com.app
groovy> class GroovyOperatorsExample5 {
groovy> static void main(args) {
groovy>     int a = 10
groovy>     a+=3
groovy>     println "a+=3 -----> " + a
groovy>     a-=3
groovy>     println "a-=3 -----> " + a
groovy>     a*=3
groovy>     println "a*=3 -----> " + a
groovy>     a/=3
groovy>     println "a/=3 -----> " + a
groovy>     a%=3
groovy>     println "a%=3 -----> " + a
groovy>     a**=3
groovy>     println "a**=3 -----> " + a
groovy> }
groovy> }

```

```

a+=3 -----> 13
a-=3 -----> 10
a*=3 -----> 30
a/=3 -----> 10
a%=3 -----> 1
a**=3 -----> 1

```

Relational operators

```
groovy> package com.app
groovy> class GroovyOperatorsExample6 {
groovy> static void main(args) {
groovy>     int a = 10
groovy>     int b = 12
groovy>     boolean c
groovy>     println "a = 10"
groovy>     println "b = 12"
groovy>     c = a == b
groovy>     println "Relational Operator equals [c = a == b] ----> " + c
groovy>     c = a != b
groovy>     println "Relational Operator different [c = a == b] ----> " + c
groovy>     c = a < b
groovy>     println "Relational Operator less than [c = a < b] ----> " + c
groovy>     c = a <= b
groovy>     println "Relational Operator less than equal to [c = a <= b] ----> " + c
groovy>     c = a > b
groovy>     println "Relational Operator greater than [c = a > b] ----> " + c
groovy>     c = a >= b
groovy>     println "Relational Operator greater than equal to [c = a >= b] ----> " + c
groovy> }
groovy> }
```

a = 10
b = 12
Relational Operator equals [c = a == b] ----> false
Relational Operator different [c = a == b] ----> true
Relational Operator less than [c = a < b] ----> true
Relational Operator less than equal to [c = a <= b] ----> true
Relational Operator greater than [c = a > b] ----> false
Relational Operator greater than equal to [c = a >= b] ----> false

Logical operators

```
groovy> package com.app
groovy> class GroovyOperatorsExample7 {
groovy> static void main(args) {
groovy>     boolean c
groovy>     c = true && true
groovy>     println "Logical AND operator = " + c
groovy>     c = true || false
groovy>     println "Logical OR operator = " + c
groovy>     c = !false
groovy>     println "Logical NOT operator = " + c
groovy> }
groovy> }
```

Logical AND operator = true
Logical OR operator = true
Logical NOT operator = true

```
Logical AND operator = true
Logical OR operator = true
Logical NOT operator = true
groovy> package com.app
groovy> class GroovyOperatorsExample8 {
groovy> static void main(args) {
groovy>     boolean c
groovy>     c = (!false && false)
groovy>     println c
groovy> }
groovy> }

false
```

```
groovy> package com.app
groovy> class GroovyOperatorsExample1 {
groovy> static void main(args) {
groovy>     boolean c
groovy>     c = true || true && false
groovy>     println c
groovy> }
groovy> }

true
```

```
groovy> package com.app
groovy> class GroovyOperatorsExample1 {
groovy> static void main(args) {
groovy>     boolean c
groovy>     c = true || true && false
groovy>     println c
groovy> }
groovy> }

true
```


Bitwise Operators

```
groovy> package com.app
groovy> class GroovyOperatorsExample10 {
groovy>
groovy>     static void main(args) {
groovy>         int a = 0b00101111
groovy>         println "a = 0b00101111 ----> "+a
groovy>         int b = 0b000010101
groovy>         println "b = 0b000010101 ----> "+b
groovy>         println "(a & a) ----> "+(a & a)
groovy>         println "(a & b) ----> "+(a & b)
groovy>         println "(a | a) ----> "+(a | a)
groovy>         println "(a | a) ----> "+(a | b)
groovy>
groovy>         int c = 0b11111111
groovy>         println "c = 0b11111111"
groovy>         println "((a ^ a) & c) ----> "+((a ^ a) & c)
groovy>         println "((a ^ b) & c) ----> "+((a ^ b) & c)
groovy>         println "((~a) & c) ----> "+((~a) & c)
groovy>     }
groovy> }
```

```
a = 0b00101111 ----> 47
b = 0b000010101 ----> 21
(a & a) ----> 47
(a & b) ----> 5
(a | a) ----> 47
(a | a) ----> 63
c = 0b11111111
((a ^ a) & c) ----> 0
((a ^ b) & c) ----> 58
((~a) & c) ----> 208
```

```
groovy> package com.app
groovy> class GroovyOperatorsExample11 {
groovy>     static void main(args) {
groovy>         int a = 23
groovy>         int b = 43
groovy>         println "Converting Integer to Binary a = 23 ----> " + Integer.toBinaryString(a)
groovy>         println "Converting Integer to Binary b = 43 ----> " + Integer.toBinaryString(b)
groovy>         println "Converting binary to integer 10111 ----> a = " + Integer.parseInt("10111", 2)
groovy>         println "Converting binary to integer 101011 ----> b = " + Integer.parseInt("10111",2)
groovy>     }
groovy> }
```

```
Converting Integer to Binary a = 23 ----> 10111
Converting Integer to Binary b = 43 ----> 101011
Converting binary to integer 10111 ----> a = 23
Converting binary to integer 101011 ----> b = 23
```

Conditional operators

```
groovy> package com.app
groovy> class GroovyOperatorsExample12 {
groovy> static void main(args) {
groovy>     println "(!true) ----> "+(!true)
groovy>     println "(!'javatpoint') ----> "+(!'javatpoint')
groovy>     println "!Null ----> "+(!'')
groovy> }
groovy> }
```

(!true) ----> false
(!'javatpoint') ----> false
!Null ----> true

```
groovy> String Answer
groovy> String s = 'javatpoint'
groovy> Answer = (s!=null && s.length()>0) ? 'Found' : 'Not found'
```

Result: Found

```
groovy> String s = 'Bhargava Ram'
groovy> def Answer = s ? 'Found' : 'Not Found' // Correct ternary operation
groovy> println "Answer = ${Answer}"
groovy> Answer = s ?: 'Found' // Elvis operator usage
groovy> println "Answer = ${Answer}"
```

Answer = Found
Answer = Bhargava Ram