

Decision Making in Groovy

If -example programs

Main.groovy439bb38dm

AI

NEW

GROOVY

RUN

```
1 package com.app
2
3 class GroovyDecisionMakingExample1 {
4
5     static void main(args) {
6         int a = 10
7         if (a<50) {
8             println("javatpoint");
9         }
10    }
11
12 }
```

Output:
javatpoint

Main.groovy439bb38dm

AI

NEW

GROOVY

RUN

```
1 package com.app
2
3 class GroovyDecisionMakingExample2 {
4
5     static void main(args) {
6         def x = false
7         if ( !x ) {
8             x = true
9             println x
10        }
11    }
12 }
```

Output:
true

If else program

Main.groovy439bb38dm

AI

NEW

GROOVY

RUN

```
1 package com.app
2
3 class GroovyDecisionMakingExample3 {
4
5     static void main(args) {
6         int a = 20
7         if (a<50) {
8             println("The value is less than 50");
9         } else {
10            println("The value is greater than 50");
11        }
12    }
13 }
```

Output:
The value is less than 50

Nested if program

≡

OneCompiler

PRICING EDITOR CHALLENGES COMPANY & MORE

LOGIN

Main.groovy 439bb38dm

AI NEW GROOVY RUN

```
1 package com.app
2
3 class GroovyDecisionMakingExample5 {
4
5     static void main(args) {
6         int a = 20
7         if (a>50) {
8             println("The value is less than 50");
9         } else
10        if (a>10) {
11            println("The value is greater than 10 and greater than 50");
12        } else {
13            println("The value of a is less than 10");
14        }
15    }
16 }
```

Output:

The value is greater than 10 and greater than 50

Groovy switch program

≡

OneCompiler

PRICING EDITOR CHALLENGES COMPANY & MORE

LOGIN

Main.groovy 439bb38dm

AI NEW GROOVY RUN

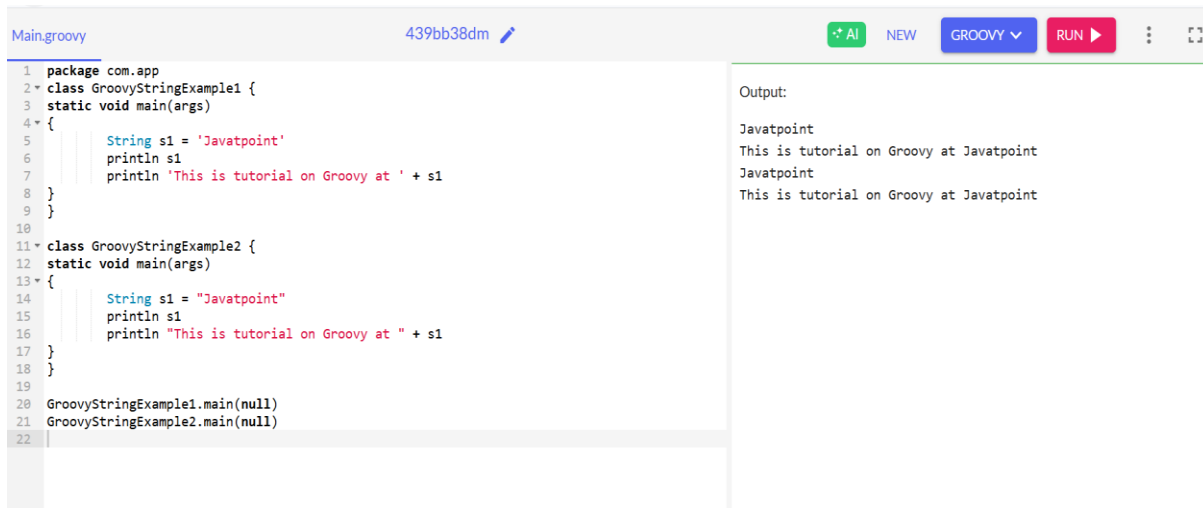
```
1 package com.app
2
3 class GroovyDecisionMakingExample6 {
4
5     static void main(args) {
6         int a = 4
7
8         switch(a) {
9
10            case 1:
11                println("Monday");
12                break;
13            case 2:
14                println("Tuesday");
15                break;
16            case 3:
17                println("Wednesday");
18                break;
19            case 4:
20                println("Thursday");
21                break;
22            case 5:
23                println("Friday");
24                break;
25            case 6:
26                println("Saturday");
27                break;
28            default:
```

Output:

Thursday

String in Groovy

Single and double quoted string



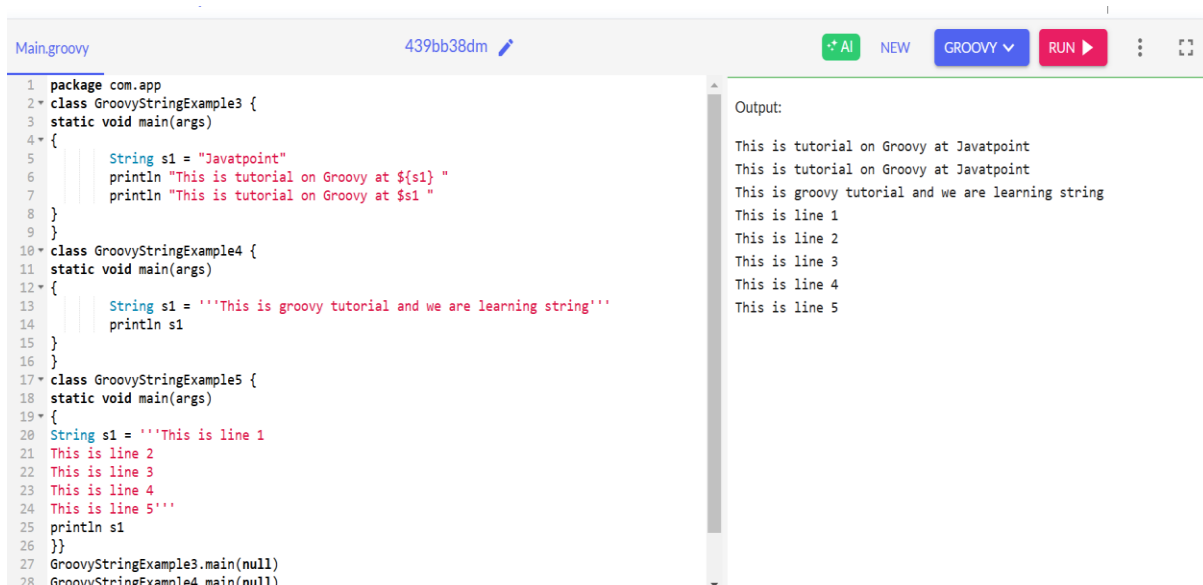
The screenshot shows an IDE window titled 'Main.groovy' with a file icon and the ID '439bb38dm'. The code defines two classes, `GroovyStringExample1` and `GroovyStringExample2`, each with a `main` method. `GroovyStringExample1` uses single quotes for string literals, while `GroovyStringExample2` uses double quotes. The `main` methods call `println` with these strings. The output pane on the right shows the results of running the code, displaying the strings as they appear in the code.

```
1 package com.app
2 class GroovyStringExample1 {
3     static void main(args)
4     {
5         String s1 = 'Javatpoint'
6         println s1
7         println 'This is tutorial on Groovy at ' + s1
8     }
9 }
10
11 class GroovyStringExample2 {
12     static void main(args)
13     {
14         String s1 = "Javatpoint"
15         println s1
16         println "This is tutorial on Groovy at " + s1
17     }
18 }
19
20 GroovyStringExample1.main(null)
21 GroovyStringExample2.main(null)
22
```

Output:

```
Javatpoint
This is tutorial on Groovy at Javatpoint
Javatpoint
This is tutorial on Groovy at Javatpoint
```

Triple-single-quoted string



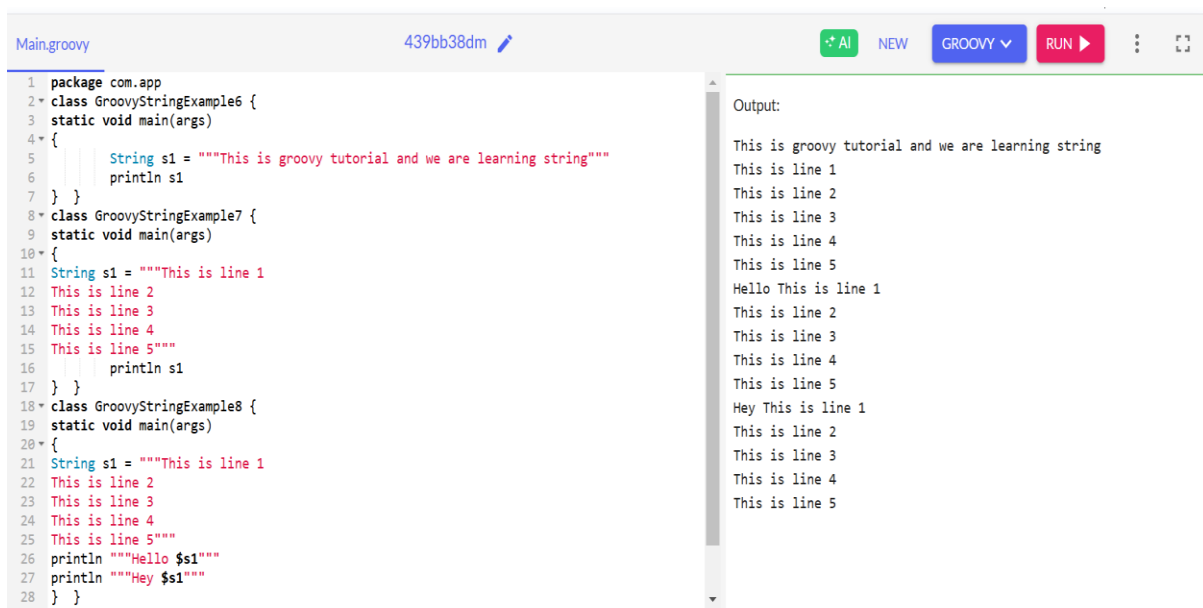
The screenshot shows an IDE window titled 'Main.groovy' with a file icon and the ID '439bb38dm'. The code defines three classes, `GroovyStringExample3`, `GroovyStringExample4`, and `GroovyStringExample5`, each with a `main` method. `GroovyStringExample3` uses single quotes for string literals. `GroovyStringExample4` uses triple single quotes for a multi-line string. `GroovyStringExample5` uses triple single quotes for a multi-line string. The `main` methods call `println` with these strings. The output pane on the right shows the results of running the code, displaying the strings as they appear in the code.

```
1 package com.app
2 class GroovyStringExample3 {
3     static void main(args)
4     {
5         String s1 = "Javatpoint"
6         println "This is tutorial on Groovy at ${s1} "
7         println "This is tutorial on Groovy at $s1 "
8     }
9 }
10
11 class GroovyStringExample4 {
12     static void main(args)
13     {
14         String s1 = '''This is groovy tutorial and we are learning string'''
15         println s1
16     }
17 }
18
19 class GroovyStringExample5 {
20     static void main(args)
21     {
22         String s1 = '''This is line 1
23         This is line 2
24         This is line 3
25         This is line 4
26         This is line 5'''
27         println s1
28     }
29 }
30
31 GroovyStringExample3.main(null)
32 GroovyStringExample4.main(null)
33
```

Output:

```
This is tutorial on Groovy at Javatpoint
This is tutorial on Groovy at Javatpoint
This is groovy tutorial and we are learning string
This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
```

Triple-double-quoted string



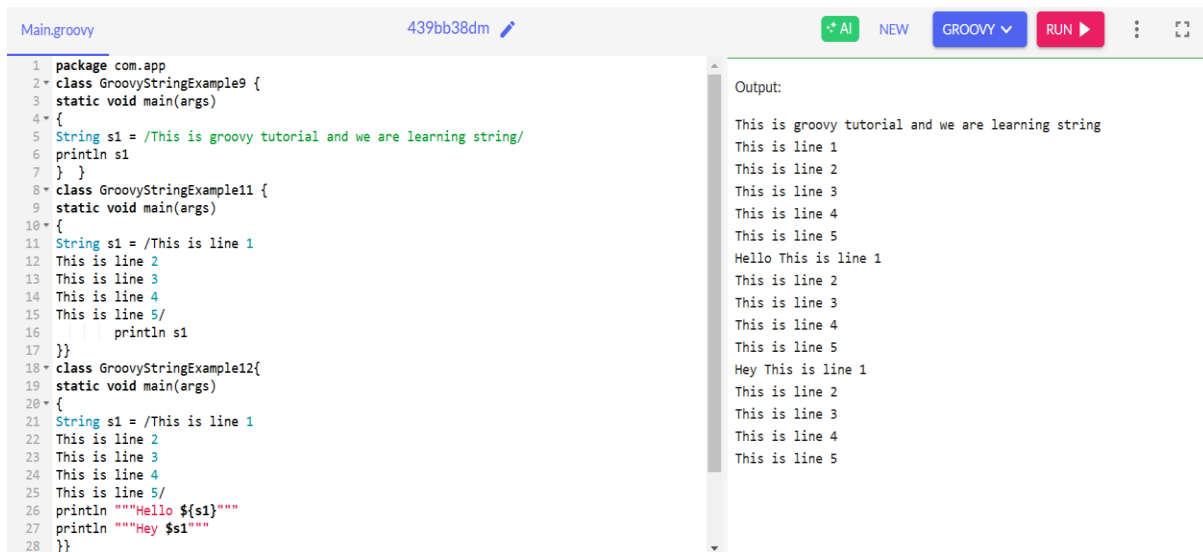
The screenshot shows a code editor with a file named 'Main.groovy' and a user identifier '439bb38dm'. The code defines three classes: GroovyStringExample6, GroovyStringExample7, and GroovyStringExample8. Each class has a static main method that prints a string. GroovyStringExample6 uses a triple-double-quoted string to print a single line. GroovyStringExample7 and GroovyStringExample8 use triple-double-quoted strings to print multiple lines. The output on the right shows the results of running the code, with each line of the triple-double-quoted strings being printed on a new line.

```
1 package com.app
2 class GroovyStringExample6 {
3     static void main(args)
4     {
5         String s1 = """This is groovy tutorial and we are learning string"""
6         println s1
7     }
8 class GroovyStringExample7 {
9     static void main(args)
10    {
11        String s1 = """This is line 1
12        This is line 2
13        This is line 3
14        This is line 4
15        This is line 5"""
16        println s1
17    }
18 class GroovyStringExample8 {
19     static void main(args)
20    {
21        String s1 = """This is line 1
22        This is line 2
23        This is line 3
24        This is line 4
25        This is line 5"""
26        println """Hello $s1"""
27        println """Hey $s1"""
28    }
}
```

Output:

```
This is groovy tutorial and we are learning string
This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
Hello This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
Hey This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
```

Slashy string in Groovy



The screenshot shows a code editor with a file named 'Main.groovy' and a user identifier '439bb38dm'. The code defines three classes: GroovyStringExample9, GroovyStringExample11, and GroovyStringExample12. Each class has a static main method that prints a string. GroovyStringExample9 uses a slashy string to print a single line. GroovyStringExample11 and GroovyStringExample12 use slashy strings to print multiple lines. The output on the right shows the results of running the code, with each line of the slashy strings being printed on a new line.

```
1 package com.app
2 class GroovyStringExample9 {
3     static void main(args)
4     {
5         String s1 = /This is groovy tutorial and we are learning string/
6         println s1
7     }
8 class GroovyStringExample11 {
9     static void main(args)
10    {
11        String s1 = /This is line 1
12        This is line 2
13        This is line 3
14        This is line 4
15        This is line 5/
16        println s1
17    }
18 class GroovyStringExample12 {
19     static void main(args)
20    {
21        String s1 = /This is line 1
22        This is line 2
23        This is line 3
24        This is line 4
25        This is line 5/
26        println """Hello ${s1}"""
27        println """Hey $s1"""
28    }
}}
```

Output:

```
This is groovy tutorial and we are learning string
This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
Hello This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
Hey This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
```

Dollar slashy string

439bb38dm

AI NEW GROOVY RUN

```
1 package com.app
2
3 class GroovyStringExample9 {
4     static void main(args) {
5         String s1 = $/This is a Groovy tutorial, and we are learning strings/$
6         println s1
7     }
8 }
9
10 class GroovyStringExample13 {
11     static void main(args) {
12         String s1 = $/This is line 1
13 This is line 2
14 This is line 3
15 This is line 4
16 This is line 5/$
17         println s1
18     }
19 }
20
21 // Calling both main methods
22 GroovyStringExample9.main(null)
23 GroovyStringExample13.main(null)
24
```

Output:

This is a Groovy tutorial, and we are learning strings
This is line 1
This is line 2
This is line 3
This is line 4
This is line 5