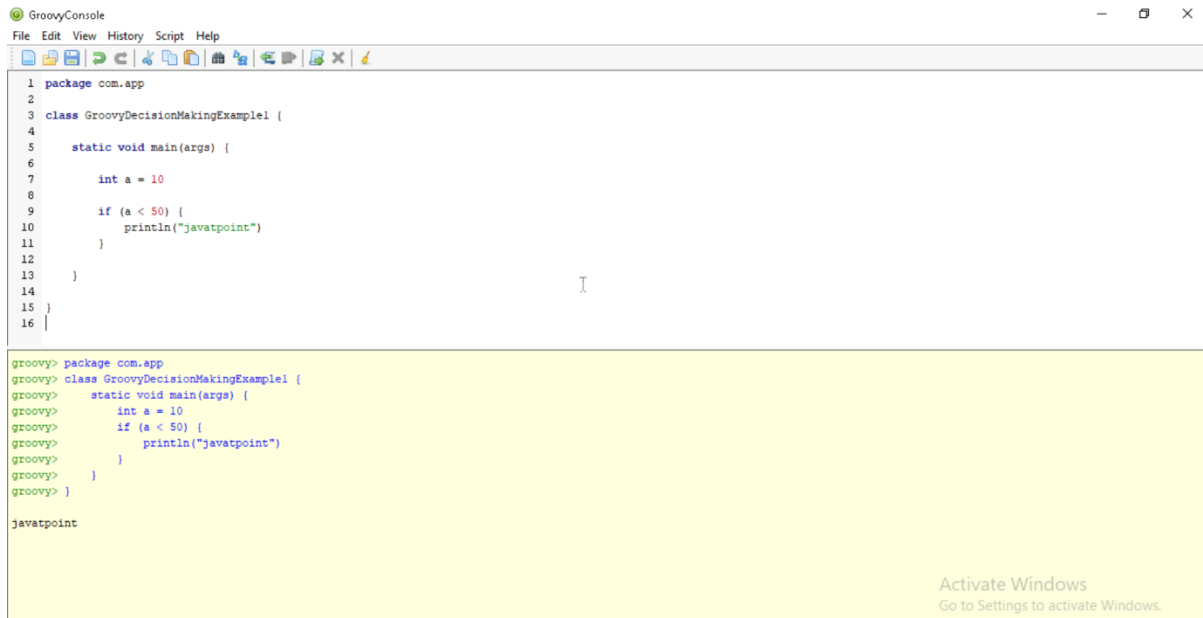


Decision Making in Groovy

1) If statement



The screenshot shows the GroovyConsole application. The top pane contains the following Groovy code:

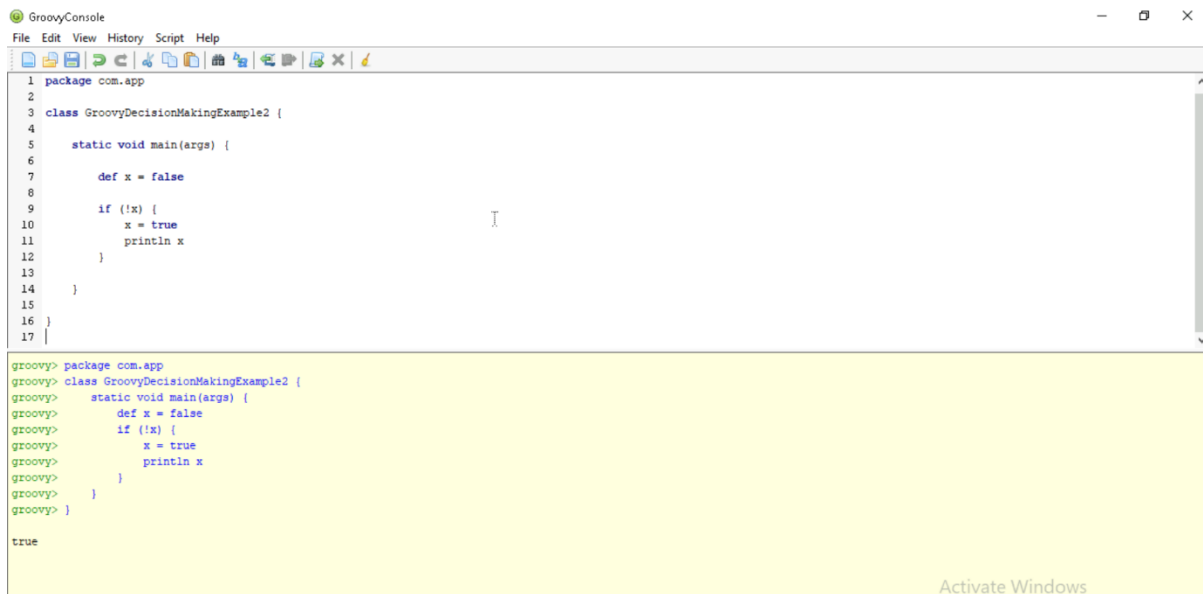
```
1 package com.app
2
3 class GroovyDecisionMakingExample1 {
4
5     static void main(args) {
6
7         int a = 10
8
9         if (a < 50) {
10             println("javatpoint")
11         }
12     }
13 }
14
15
16
```

The bottom pane shows the execution output:

```
groovy> package com.app
groovy> class GroovyDecisionMakingExample1 {
groovy>     static void main(args) {
groovy>         int a = 10
groovy>         if (a < 50) {
groovy>             println("javatpoint")
groovy>         }
groovy>     }
groovy> }
groovy>
javatpoint
```

An "Activate Windows" watermark is visible in the bottom right corner.

2) If statement



The screenshot shows the GroovyConsole application. The top pane contains the following Groovy code:

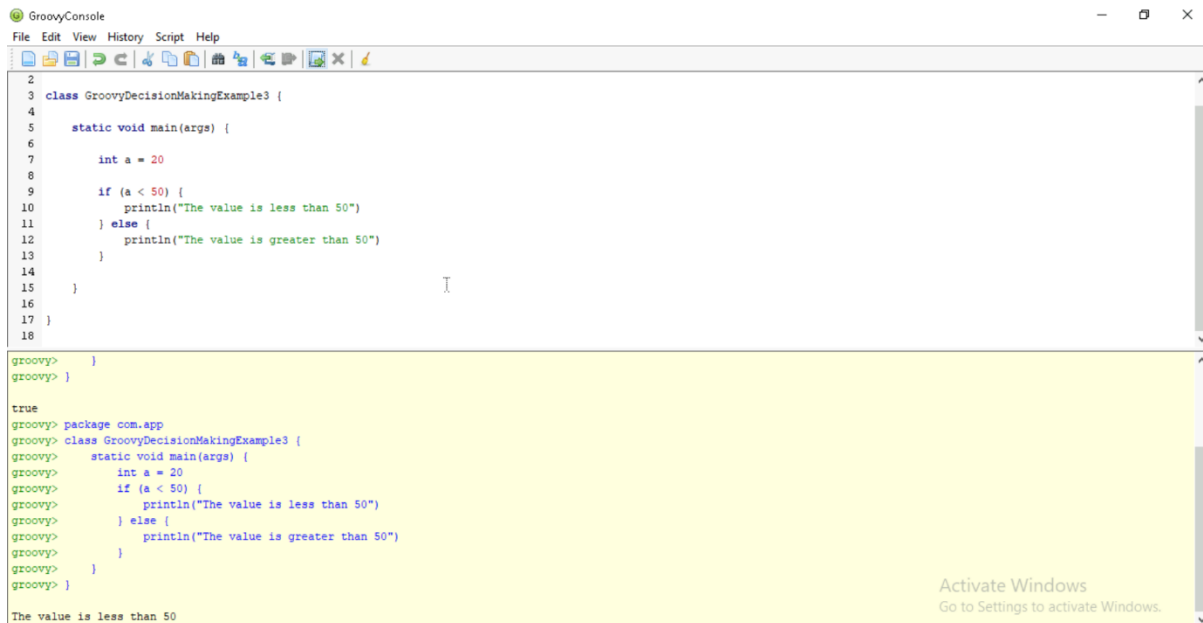
```
1 package com.app
2
3 class GroovyDecisionMakingExample2 {
4
5     static void main(args) {
6
7         def x = false
8
9         if (!x) {
10             x = true
11             println x
12         }
13     }
14 }
15
16
17
```

The bottom pane shows the execution output:

```
groovy> package com.app
groovy> class GroovyDecisionMakingExample2 {
groovy>     static void main(args) {
groovy>         def x = false
groovy>         if (!x) {
groovy>             x = true
groovy>             println x
groovy>         }
groovy>     }
groovy> }
groovy>
true
```

An "Activate Windows" watermark is visible in the bottom right corner.

3) If else statement



The screenshot shows the GroovyConsole application. The top pane contains the following Groovy code:

```
2
3 class GroovyDecisionMakingExample3 {
4
5     static void main(args) {
6
7         int a = 20
8
9         if (a < 50) {
10             println("The value is less than 50")
11         } else {
12             println("The value is greater than 50")
13         }
14     }
15 }
16
17
18
```

The bottom pane shows the execution results:

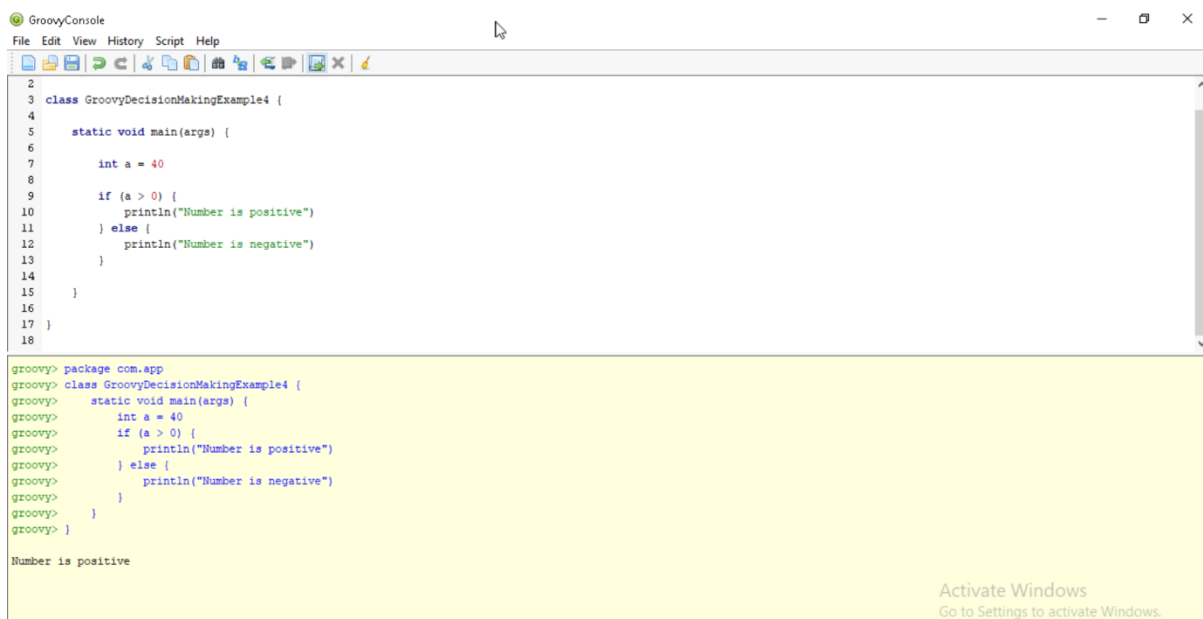
```
groovy>
groovy> ]

true
groovy> package com.app
groovy> class GroovyDecisionMakingExample3 {
groovy>     static void main(args) {
groovy>         int a = 20
groovy>         if (a < 50) {
groovy>             println("The value is less than 50")
groovy>         } else {
groovy>             println("The value is greater than 50")
groovy>         }
groovy>     }
groovy> ]
groovy> ]

The value is less than 50
```

An "Activate Windows" watermark is visible in the bottom right corner of the console window.

4) If else statement



The screenshot shows the GroovyConsole application. The top pane contains the following Groovy code:

```
2
3 class GroovyDecisionMakingExample4 {
4
5     static void main(args) {
6
7         int a = 40
8
9         if (a > 0) {
10             println("Number is positive")
11         } else {
12             println("Number is negative")
13         }
14     }
15 }
16
17
18
```

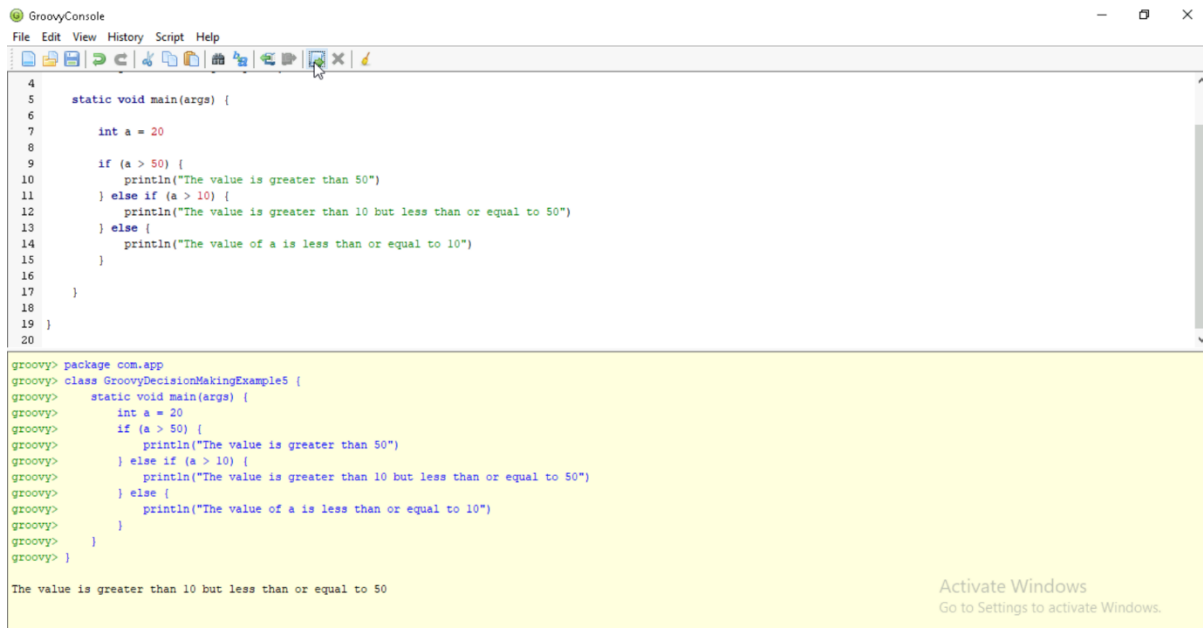
The bottom pane shows the execution results:

```
groovy> package com.app
groovy> class GroovyDecisionMakingExample4 {
groovy>     static void main(args) {
groovy>         int a = 40
groovy>         if (a > 0) {
groovy>             println("Number is positive")
groovy>         } else {
groovy>             println("Number is negative")
groovy>         }
groovy>     }
groovy> ]
groovy> ]

Number is positive
```

An "Activate Windows" watermark is visible in the bottom right corner of the console window.

5) Nested If Statement



The screenshot shows the GroovyConsole application. The top pane contains a Groovy script with a nested if statement. The bottom pane shows the execution output, which is "The value is greater than 10 but less than or equal to 50".

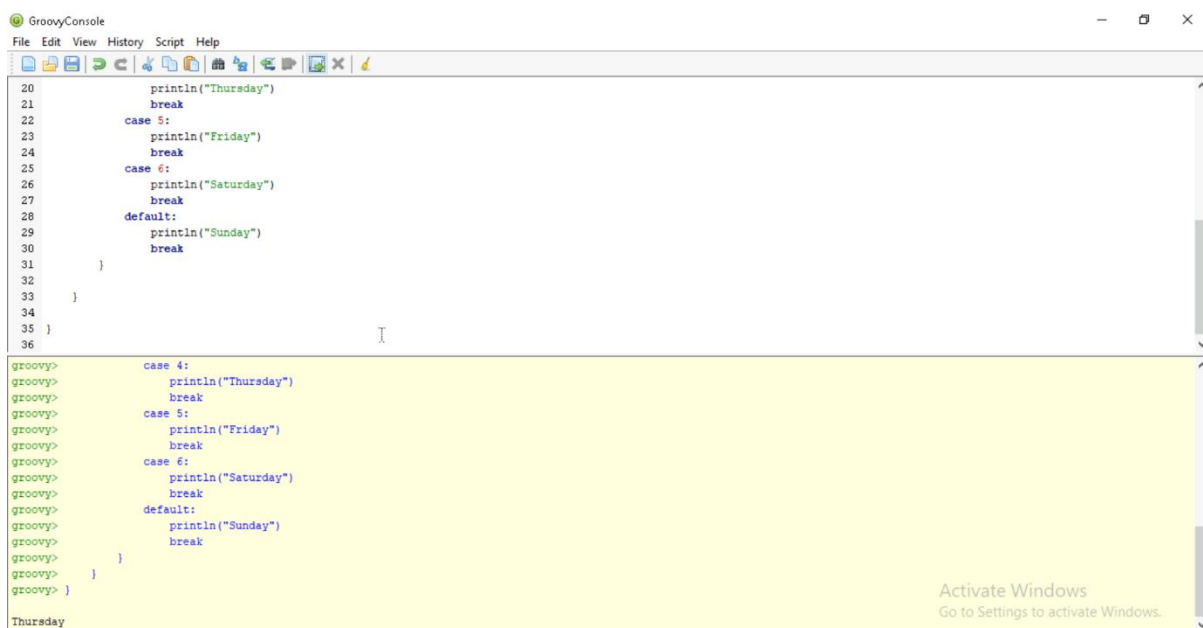
```
4 static void main(args) {
5
6     int a = 20
7
8     if (a > 50) {
9         println("The value is greater than 50")
10    } else if (a > 10) {
11        println("The value is greater than 10 but less than or equal to 50")
12    } else {
13        println("The value of a is less than or equal to 10")
14    }
15 }
16
17 }
18
19 }
20 }
```

```
groovy> package com.app
groovy> class GroovyDecisionMakingExample5 {
groovy>     static void main(args) {
groovy>         int a = 20
groovy>         if (a > 50) {
groovy>             println("The value is greater than 50")
groovy>         } else if (a > 10) {
groovy>             println("The value is greater than 10 but less than or equal to 50")
groovy>         } else {
groovy>             println("The value of a is less than or equal to 10")
groovy>         }
groovy>     }
groovy> }
groovy> }
```

The value is greater than 10 but less than or equal to 50

Activate Windows
Go to Settings to activate Windows.

6) Switch Statement



The screenshot shows the GroovyConsole application. The top pane contains a Groovy script with a switch statement. The bottom pane shows the execution output, which is "Thursday".

```
20 println("Thursday")
21 break
22 case 5:
23     println("Friday")
24     break
25 case 6:
26     println("Saturday")
27     break
28 default:
29     println("Sunday")
30     break
31 }
32
33 }
34
35 }
36 }
```

```
groovy> case 4:
groovy>     println("Thursday")
groovy>     break
groovy> case 5:
groovy>     println("Friday")
groovy>     break
groovy> case 6:
groovy>     println("Saturday")
groovy>     break
groovy> default:
groovy>     println("Sunday")
groovy>     break
groovy> }
groovy> }
```

Thursday

Activate Windows
Go to Settings to activate Windows.

String in Groovy

7) Single-quoted string



The screenshot shows the GroovyConsole application. The top pane contains Groovy code for a class `GroovyStringExample2` with a `main` method. The bottom pane shows the execution output, which matches the code's logic. An "Activate Windows" watermark is visible in the bottom right corner.

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

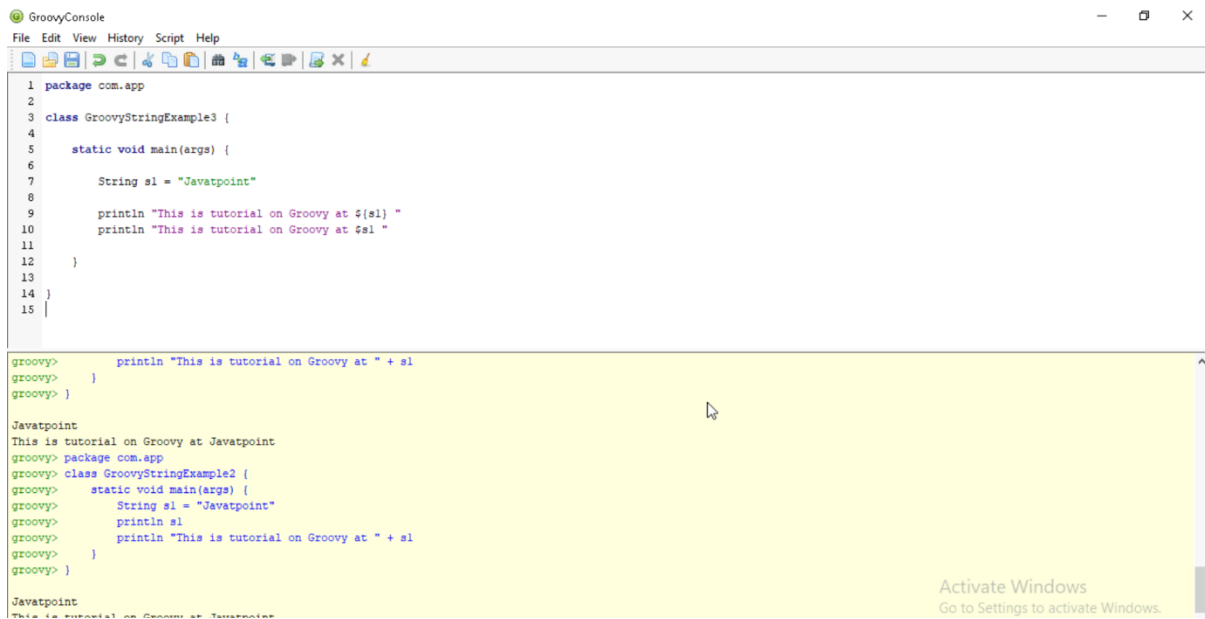
```
groovy>      println "This is tutorial on Groovy at " + s1
groovy>    }
groovy> }

Javatpoint
This is tutorial on Groovy at Javatpoint
groovy> package com.app
groovy> class GroovyStringExample2 {
groovy>     static void main(args) {
groovy>         String s1 = "Javatpoint"
groovy>         println s1
groovy>         println "This is tutorial on Groovy at " + s1
groovy>     }
groovy> }

Javatpoint
This is tutorial on Groovy at Javatpoint
```

Activate Windows
Go to Settings to activate Windows.

8) Double-quoted string



The screenshot shows the GroovyConsole application. The top pane contains Groovy code for a class `GroovyStringExample3` with a `main` method. The bottom pane shows the execution output, which matches the code's logic. An "Activate Windows" watermark is visible in the bottom right corner.

```
1 package com.app
2
3 class GroovyStringExample3 {
4
5     static void main(args) {
6
7         String s1 = "Javatpoint"
8
9         println "This is tutorial on Groovy at ${s1} "
10        println "This is tutorial on Groovy at $s1 "
11    }
12 }
13
14 }
15
```

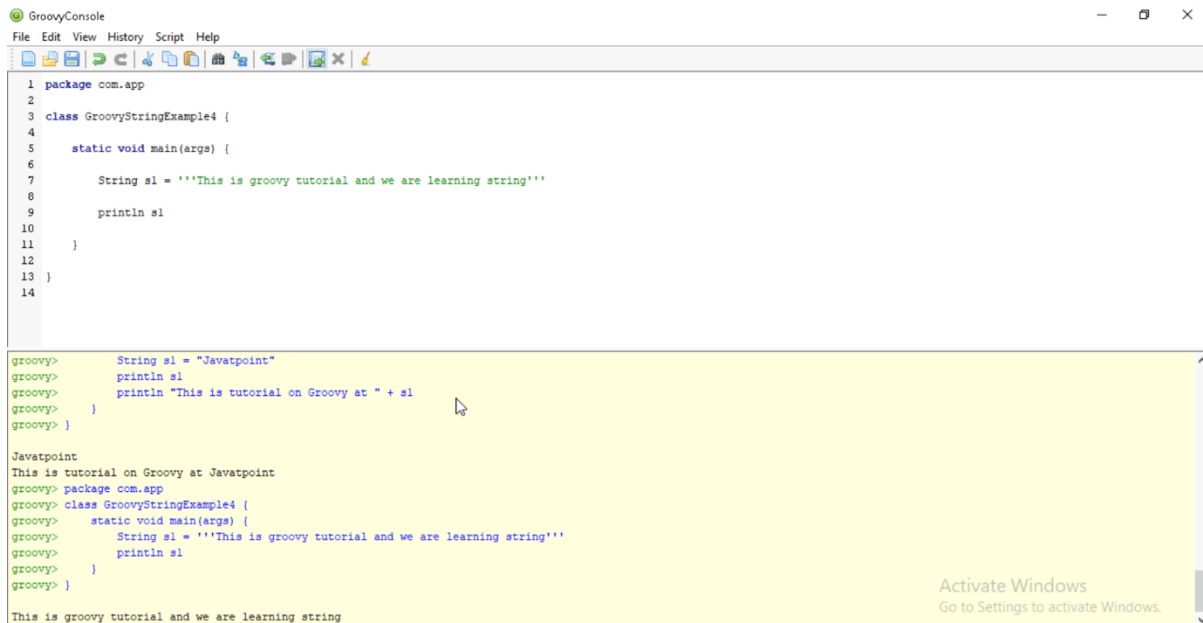
```
groovy>      println "This is tutorial on Groovy at " + s1
groovy>    }
groovy> }

Javatpoint
This is tutorial on Groovy at Javatpoint
groovy> package com.app
groovy> class GroovyStringExample2 {
groovy>     static void main(args) {
groovy>         String s1 = "Javatpoint"
groovy>         println s1
groovy>         println "This is tutorial on Groovy at " + s1
groovy>     }
groovy> }

Javatpoint
This is tutorial on Groovy at Javatpoint
```

Activate Windows
Go to Settings to activate Windows.

9) Double-quoted string



The screenshot shows the GroovyConsole application. The top pane contains a Groovy script with the following code:

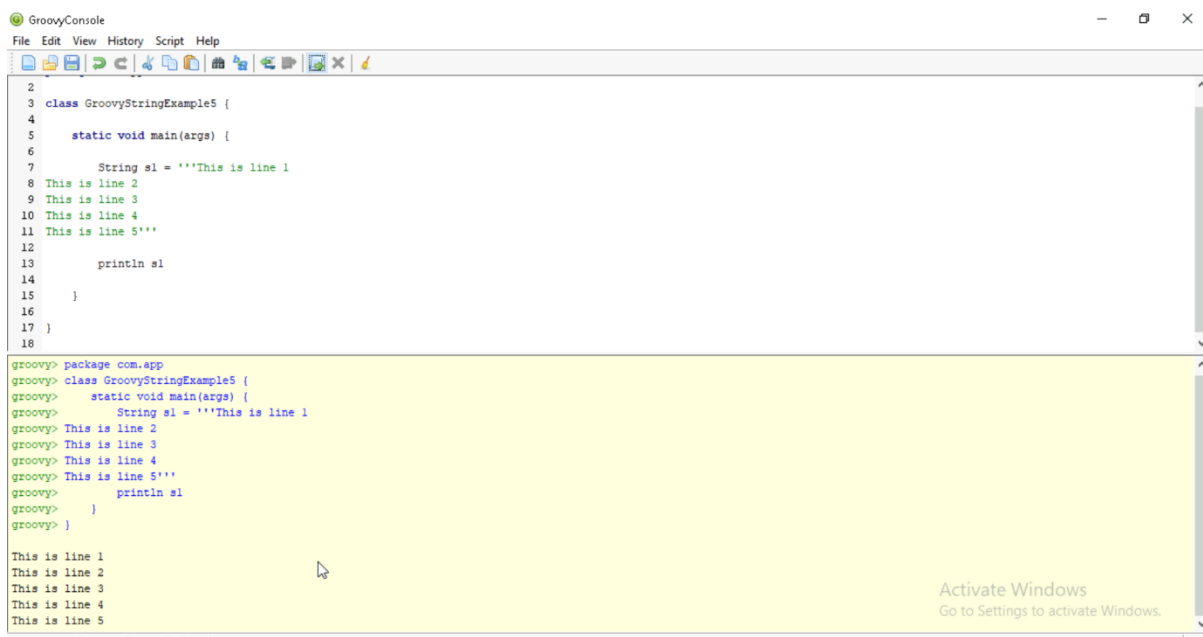
```
1 package com.app
2
3 class GroovyStringExample4 {
4
5     static void main(args) {
6
7         String s1 = "This is groovy tutorial and we are learning string"
8
9         println s1
10
11     }
12 }
13
14
```

The bottom pane shows the execution output:

```
groovy> String s1 = "Javatpoint"
groovy> println s1
groovy> println "This is tutorial on Groovy at " + s1
groovy> ]
Javatpoint
This is tutorial on Groovy at Javatpoint
groovy> package com.app
groovy> class GroovyStringExample4 {
groovy>     static void main(args) {
groovy>         String s1 = "This is groovy tutorial and we are learning string"
groovy>         println s1
groovy>     }
groovy> ]
This is groovy tutorial and we are learning string
```

An "Activate Windows" watermark is visible in the bottom right corner of the console window.

10) Triple-single-quoted string



The screenshot shows the GroovyConsole application. The top pane contains a Groovy script with the following code:

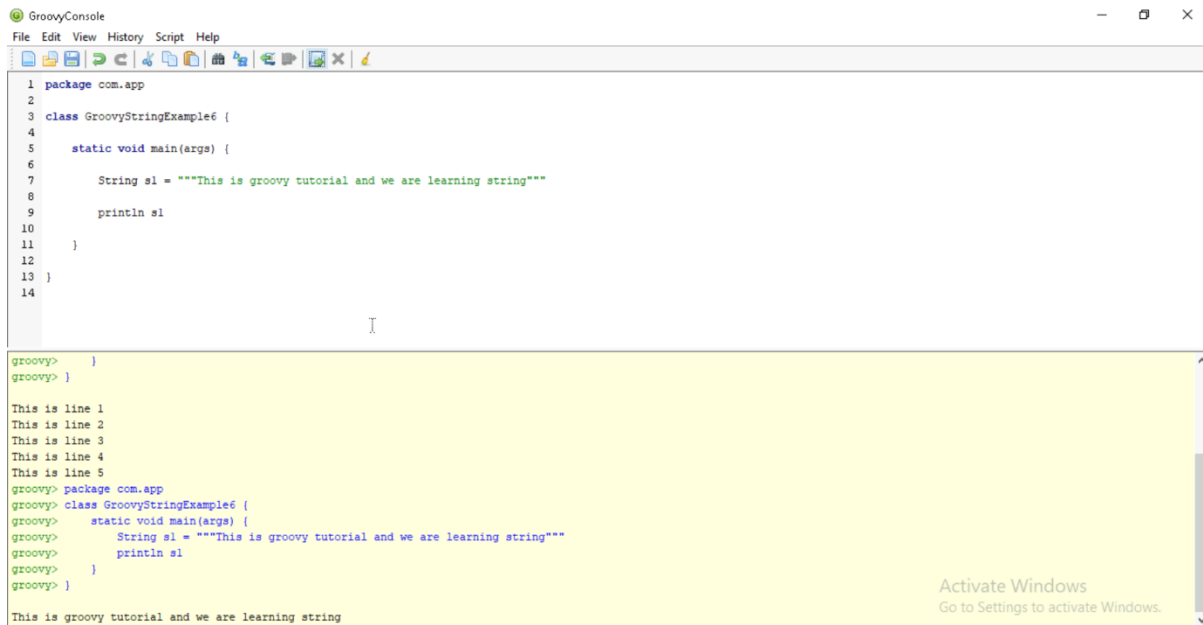
```
2
3 class GroovyStringExample5 {
4
5     static void main(args) {
6
7         String s1 = '''This is line 1
8 This is line 2
9 This is line 3
10 This is line 4
11 This is line 5'''
12
13         println s1
14
15     }
16 }
17
18
```

The bottom pane shows the execution output:

```
groovy> package com.app
groovy> class GroovyStringExample5 {
groovy>     static void main(args) {
groovy>         String s1 = '''This is line 1
groovy> This is line 2
groovy> This is line 3
groovy> This is line 4
groovy> This is line 5'''
groovy>         println s1
groovy>     }
groovy> ]
This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
```

An "Activate Windows" watermark is visible in the bottom right corner of the console window.

11) Triple-single-quoted string



The screenshot shows the GroovyConsole application. The top pane contains a Groovy script with a class `GroovyStringExample6` and a `main` method. Inside the `main` method, a `String` variable `s1` is assigned a value using a triple-single-quoted string: `String s1 = """This is groovy tutorial and we are learning string"""`. The `println s1` statement is also present. The bottom pane shows the output of the script, which is `This is groovy tutorial and we are learning string`. The console also displays the execution of the `package` and `class` statements, and the `println` statement.

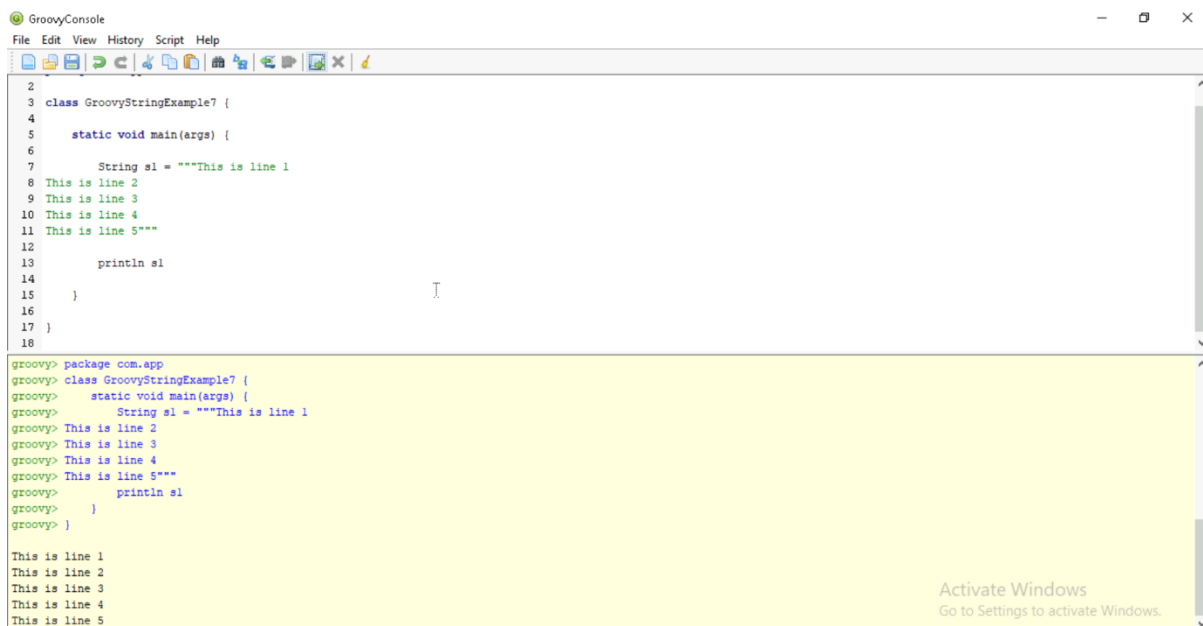
```
1 package com.app
2
3 class GroovyStringExample6 {
4
5     static void main(args) {
6
7         String s1 = """This is groovy tutorial and we are learning string"""
8
9         println s1
10
11     }
12 }
13
14
```

```
groovy> }
groovy> ]

This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
groovy> package com.app
groovy> class GroovyStringExample6 {
groovy>     static void main(args) {
groovy>         String s1 = """This is groovy tutorial and we are learning string"""
groovy>         println s1
groovy>     }
groovy> }
groovy> ]

This is groovy tutorial and we are learning string
```

12) Triple-double-quoted string



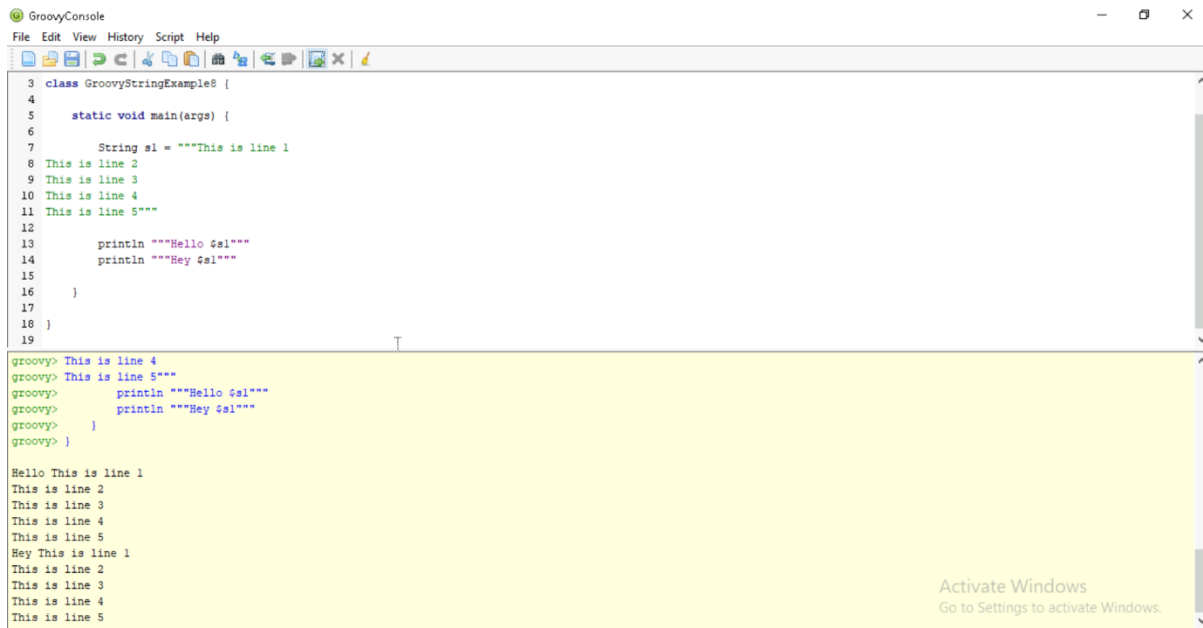
The screenshot shows the GroovyConsole application. The top pane contains a Groovy script with a class `GroovyStringExample7` and a `main` method. Inside the `main` method, a `String` variable `s1` is assigned a value using a triple-double-quoted string: `String s1 = """This is line 1`
`This is line 2`
`This is line 3`
`This is line 4`
`This is line 5"""`. The `println s1` statement is also present. The bottom pane shows the output of the script, which is `This is line 1`
`This is line 2`
`This is line 3`
`This is line 4`
`This is line 5`. The console also displays the execution of the `package` and `class` statements, and the `println` statement.

```
2
3 class GroovyStringExample7 {
4
5     static void main(args) {
6
7         String s1 = """This is line 1
8 This is line 2
9 This is line 3
10 This is line 4
11 This is line 5""
12
13         println s1
14
15     }
16 }
17
18
```

```
groovy> package com.app
groovy> class GroovyStringExample7 {
groovy>     static void main(args) {
groovy>         String s1 = """This is line 1
groovy> This is line 2
groovy> This is line 3
groovy> This is line 4
groovy> This is line 5""
groovy>         println s1
groovy>     }
groovy> }
groovy> ]

This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
```

13) Triple-double-quoted string



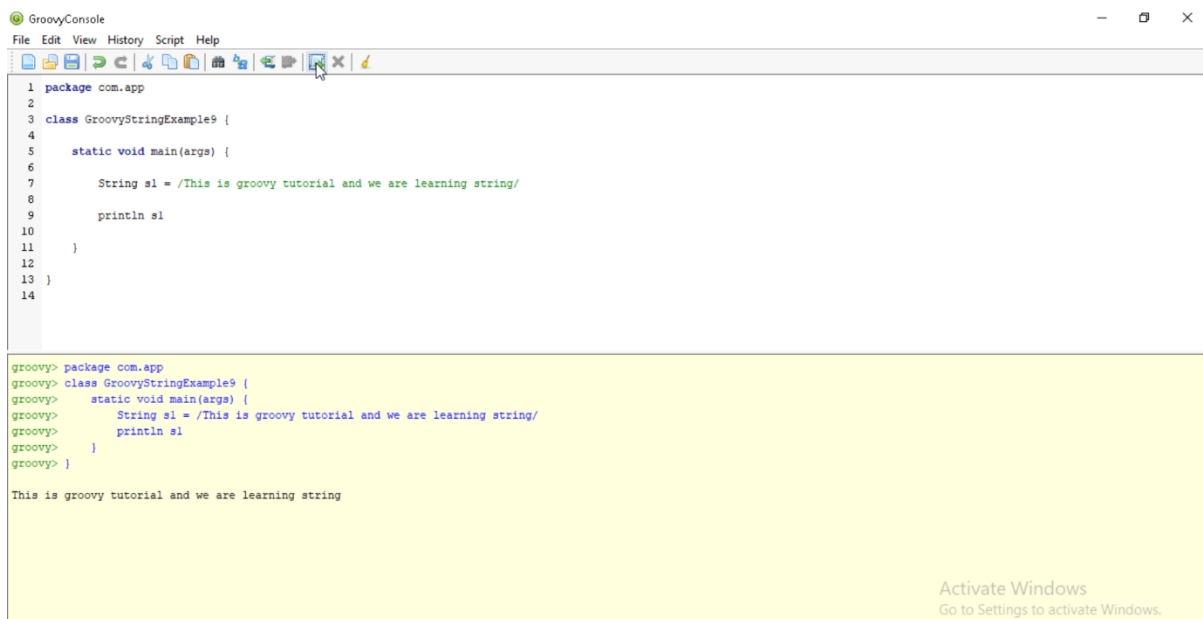
```
3 class GroovyStringExample8 {
4
5     static void main(args) {
6
7         String s1 = """This is line 1
8 This is line 2
9 This is line 3
10 This is line 4
11 This is line 5"""
12
13         println """Hello $s1"""
14         println """Hey $s1"""
15     }
16 }
17
18
19
```

```
groovy> This is line 4
groovy> This is line 5"""
groovy>         println """Hello $s1"""
groovy>         println """Hey $s1"""
groovy>     }
groovy> ]

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
```

Activate Windows
Go to Settings to activate Windows.

14) Slashy string



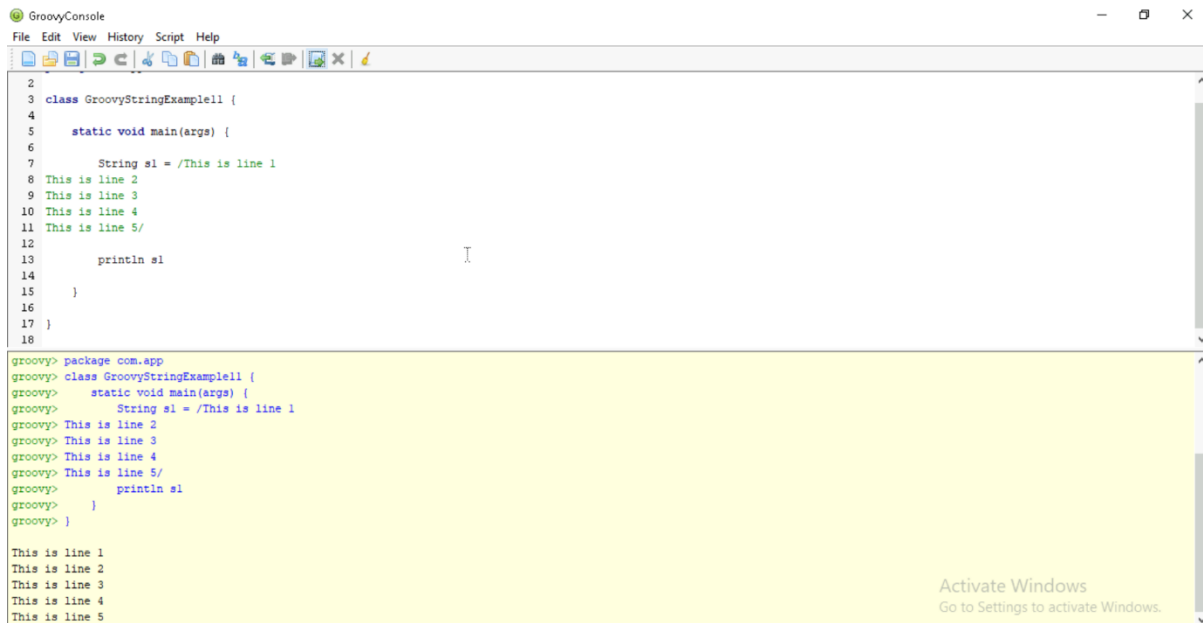
```
1 package com.app
2
3 class GroovyStringExample9 {
4
5     static void main(args) {
6
7         String s1 = /This is groovy tutorial and we are learning string/
8
9         println s1
10    }
11 }
12
13
14
```

```
groovy> package com.app
groovy> class GroovyStringExample9 {
groovy>     static void main(args) {
groovy>         String s1 = /This is groovy tutorial and we are learning string/
groovy>         println s1
groovy>     }
groovy> }
```

```
This is groovy tutorial and we are learning string
```

Activate Windows
Go to Settings to activate Windows.

15) Slashy string



The screenshot shows the GroovyConsole application. The top pane contains a Groovy script with a class `GroovyStringExample11` and a `main` method. The script defines a `String s1` with a slashy string `"/This is line 1` followed by four lines of text. The bottom pane shows the output of the script, which prints each line of the string separately. An "Activate Windows" watermark is visible in the bottom right corner.

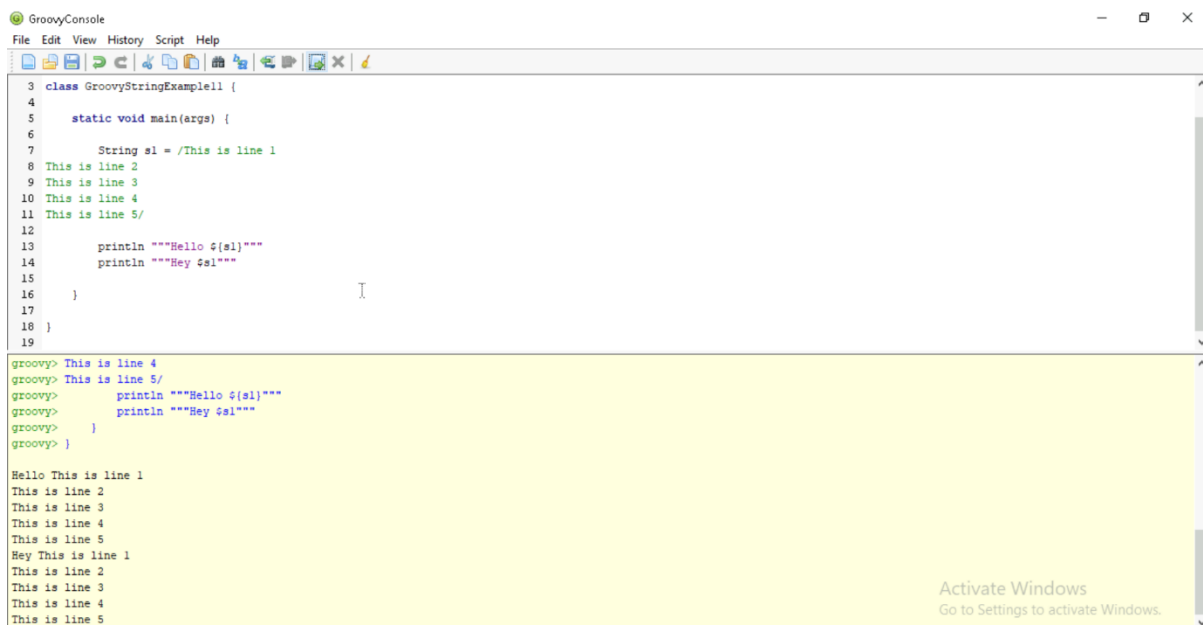
```
2
3 class GroovyStringExample11 {
4
5     static void main(args) {
6
7         String s1 = /This is line 1
8 This is line 2
9 This is line 3
10 This is line 4
11 This is line 5/
12
13         println s1
14
15     }
16
17 }
18
```

```
groovy> package com.app
groovy> class GroovyStringExample11 {
groovy>     static void main(args) {
groovy>         String s1 = /This is line 1
groovy> This is line 2
groovy> This is line 3
groovy> This is line 4
groovy> This is line 5/
groovy>         println s1
groovy>     }
groovy> }

This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
```

Activate Windows
Go to Settings to activate Windows.

16) Slashy string



The screenshot shows the GroovyConsole application. The top pane contains a Groovy script with a class `GroovyStringExample11` and a `main` method. The script defines a `String s1` with a slashy string `"/This is line 1` followed by four lines of text. The bottom pane shows the output of the script, which prints each line of the string separately. An "Activate Windows" watermark is visible in the bottom right corner.

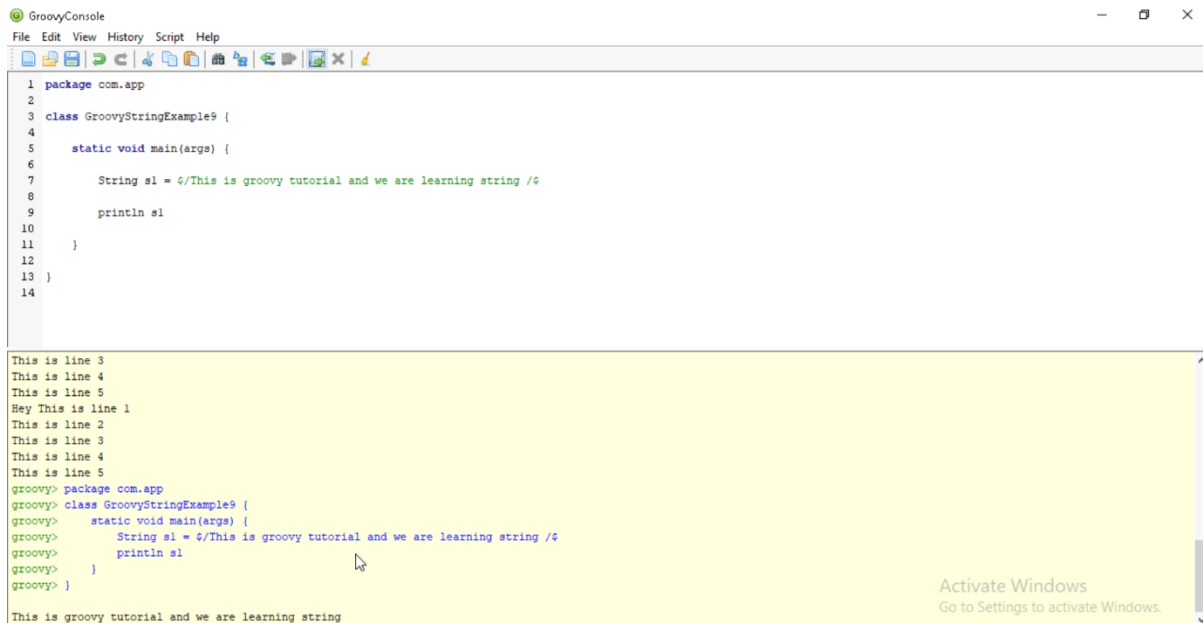
```
3 class GroovyStringExample11 {
4
5     static void main(args) {
6
7         String s1 = /This is line 1
8 This is line 2
9 This is line 3
10 This is line 4
11 This is line 5/
12
13         println ""Hello ${s1}""
14         println ""Hey ${s1}""
15
16     }
17
18 }
19
```

```
groovy> This is line 4
groovy> This is line 5/
groovy>         println ""Hello ${s1}""
groovy>         println ""Hey ${s1}""
groovy>     }
groovy> }

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
```

Activate Windows
Go to Settings to activate Windows.

17) Dollar slashy string



The screenshot shows the GroovyConsole application. The top pane contains a Groovy script with the following code:

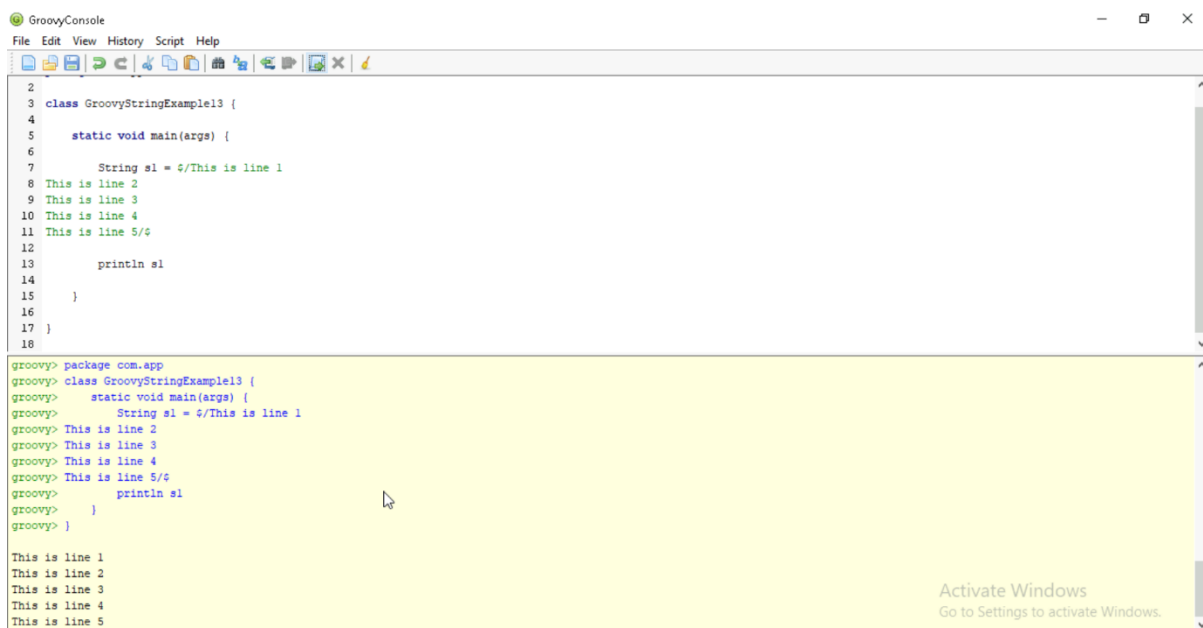
```
1 package com.app
2
3 class GroovyStringExample9 {
4
5     static void main(args) {
6
7         String s1 = "/This is groovy tutorial and we are learning string /"
8
9         println s1
10
11     }
12 }
13
14
```

The bottom pane shows the output of the script:

```
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
groovy> package com.app
groovy> class GroovyStringExample9 {
groovy>     static void main(args) {
groovy>         String s1 = "/This is groovy tutorial and we are learning string /"
groovy>         println s1
groovy>     }
groovy> }
This is groovy tutorial and we are learning string
```

An "Activate Windows" watermark is visible in the bottom right corner of the console window.

18) Dollar slashy string



The screenshot shows the GroovyConsole application. The top pane contains a Groovy script with the following code:

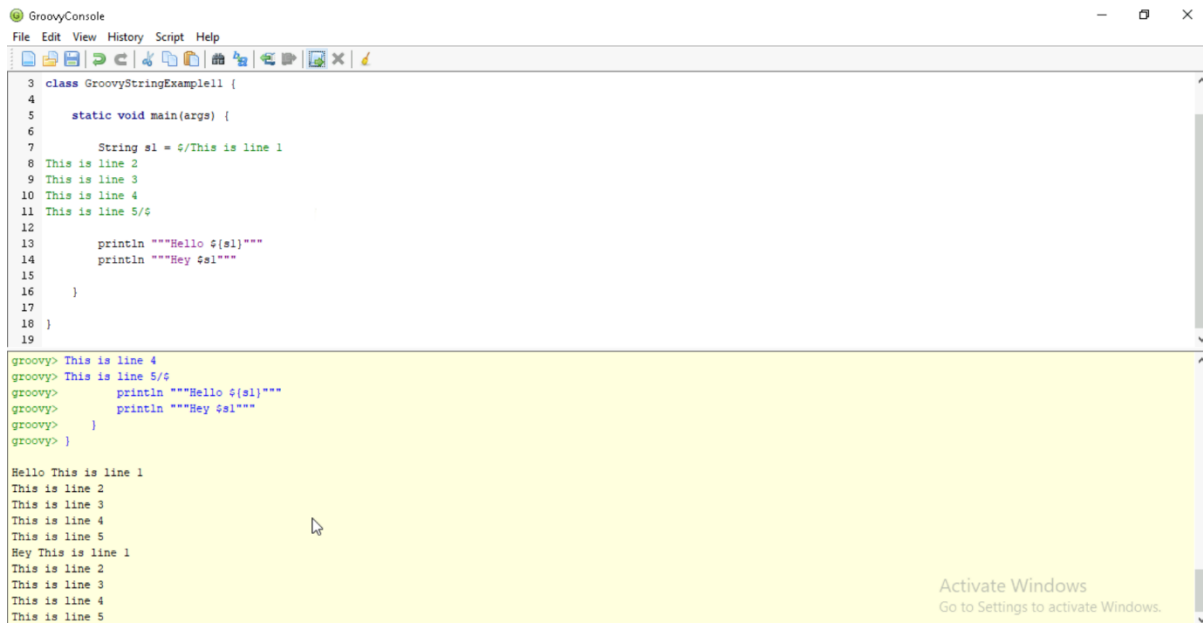
```
2
3 class GroovyStringExample13 {
4
5     static void main(args) {
6
7         String s1 = "/This is line 1
8 This is line 2
9 This is line 3
10 This is line 4
11 This is line 5/"
12
13         println s1
14
15     }
16 }
17
18
```

The bottom pane shows the output of the script:

```
groovy> package com.app
groovy> class GroovyStringExample13 {
groovy>     static void main(args) {
groovy>         String s1 = "/This is line 1
groovy> This is line 2
groovy> This is line 3
groovy> This is line 4
groovy> This is line 5/"
groovy>         println s1
groovy>     }
groovy> }
This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
```

An "Activate Windows" watermark is visible in the bottom right corner of the console window.

19) Dollar slashy string



The screenshot shows the GroovyConsole application window. The title bar reads "GroovyConsole". The menu bar includes "File", "Edit", "View", "History", "Script", and "Help". The toolbar contains icons for file operations and execution. The script editor displays a Groovy class named `GroovyStringExample11` with a `main` method. The script defines a `String` variable `s1` with a dollar slashy string value. The `main` method prints the value of `s1` using `println`. The console output shows the execution results, including the printed string and the output of the `println` statements. An "Activate Windows" watermark is visible in the bottom right corner of the console area.

```
3 class GroovyStringExample11 {
4
5     static void main(args) {
6
7         String s1 = c/This is line 1
8 This is line 2
9 This is line 3
10 This is line 4
11 This is line 5/c
12
13         println ""Hello ${s1}""
14         println ""Hey ${s1}""
15     }
16 }
17
18 }
19
```

```
groovy> This is line 4
groovy> This is line 5/c
groovy>         println ""Hello ${s1}""
groovy>         println ""Hey ${s1}""
groovy>     }
groovy> ]

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
```

Activate Windows
Go to Settings to activate Windows.