Groovy Project

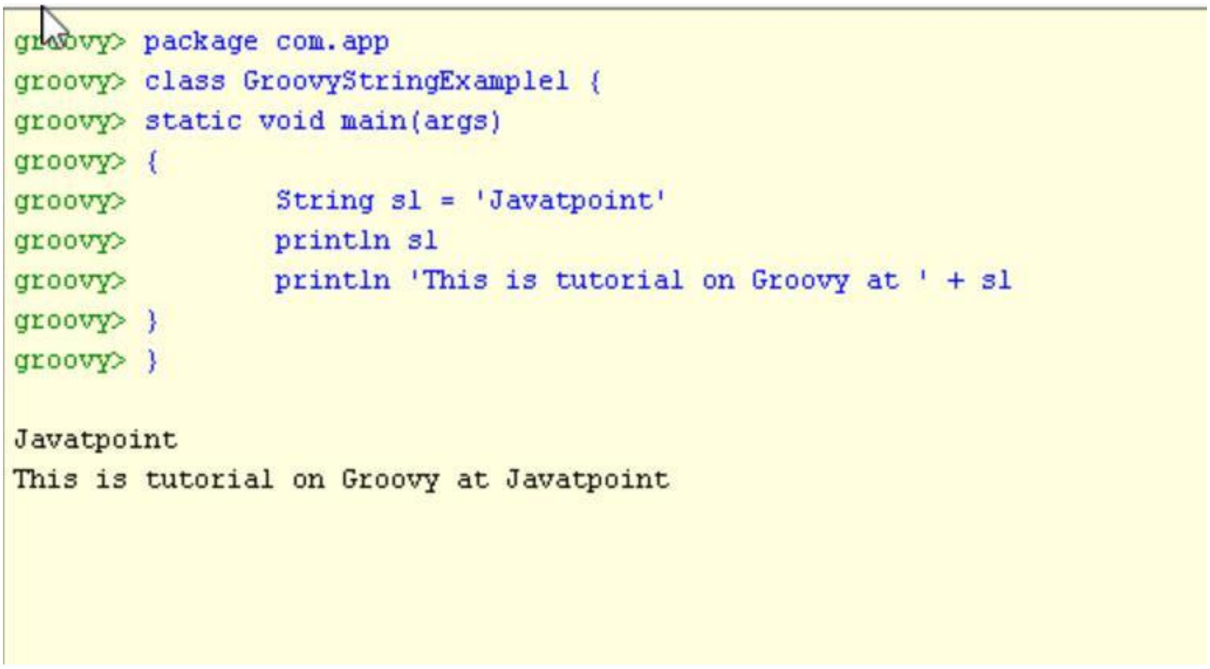
STRINGS

String in Groovy

A string is a sequence of characters. Generally, a string is a data type and implemented as an array of bytes which stores sequence of characters. In Groovy, String class is available in java.lang.String objects as well as in groovy.lang. GString which is called interpolated strings in some programming language.

Single quoted strings

Example 1:

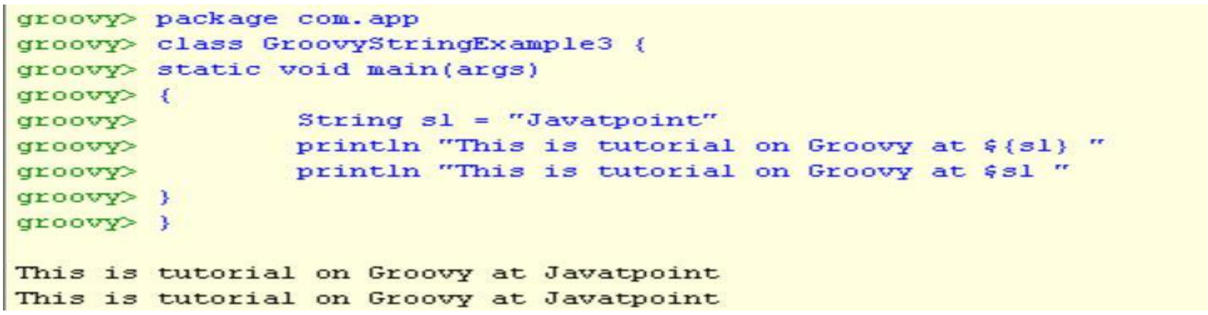


```
groovy> package com.app
groovy> class GroovyStringExample1 {
groovy> static void main(args)
groovy> {
groovy>         String s1 = 'Javatpoint'
groovy>         println s1
groovy>         println 'This is tutorial on Groovy at ' + s1
groovy> }
groovy> }

Javatpoint
This is tutorial on Groovy at Javatpoint
```

Double quoted strings:

In groovy, Double-quoted string is a sequence of characters which is enclosed under double quotes.

Example 2:

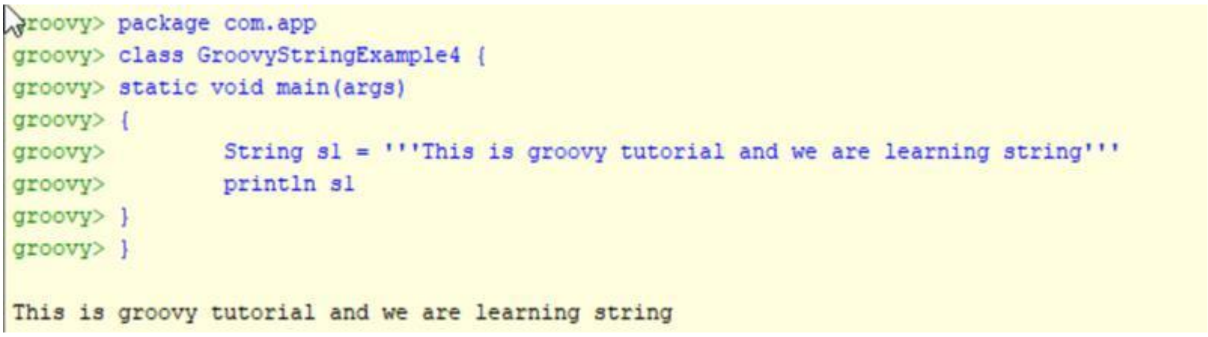


```
groovy> package com.app
groovy> class GroovyStringExample3 {
groovy> static void main(args)
groovy> {
groovy>         String s1 = "Javatpoint"
groovy>         println "This is tutorial on Groovy at ${s1} "
groovy>         println "This is tutorial on Groovy at $s1 "
groovy> }
groovy> }

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

T*riple* quoted strings:

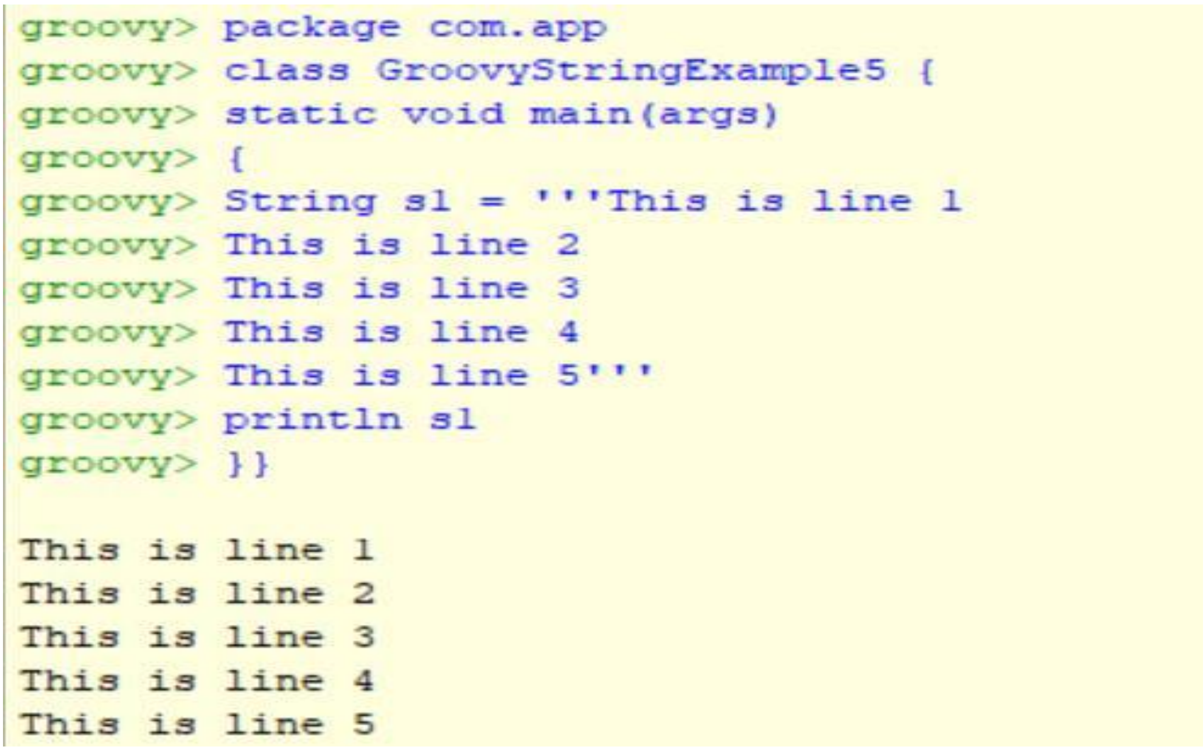
In groovy, Triple-single-quoted string is a sequence of characters which is enclosed under triple single quotes.

Example 4:



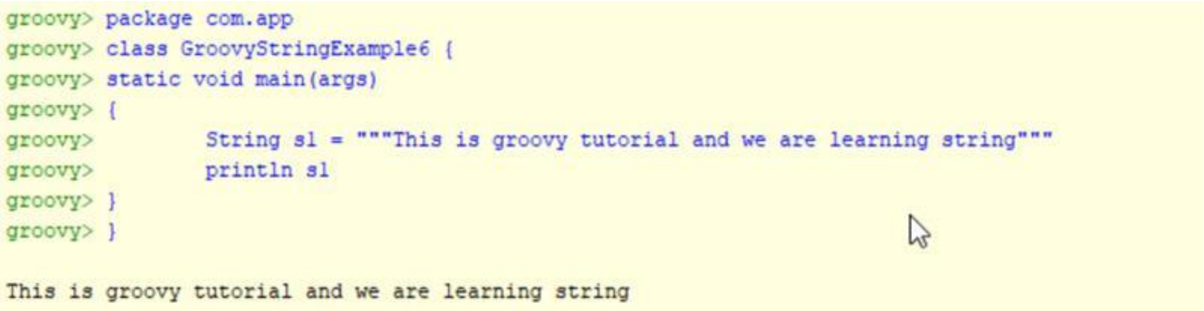
```
groovy> package com.app
groovy> class GroovyStringExample4 {
groovy> static void main(args)
groovy> {
groovy>         String s1 = '''This is groovy tutorial and we are learning string'''
groovy>         println s1
groovy> }
groovy> }

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

Example 5:



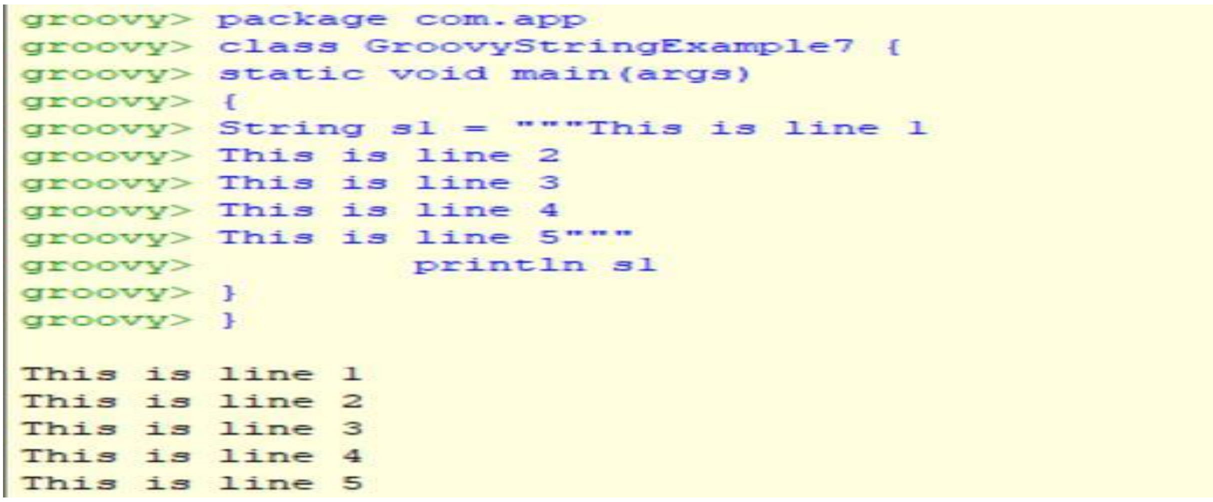
```
groovy> package com.app
groovy> class GroovyStringExample5 {
groovy> static void main(args)
groovy> {
groovy> String s1 = '''This is line 1
groovy> This is line 2
groovy> This is line 3
groovy> This is line 4
groovy> This is line 5'''
groovy> println s1
groovy> }}

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

Example 6:



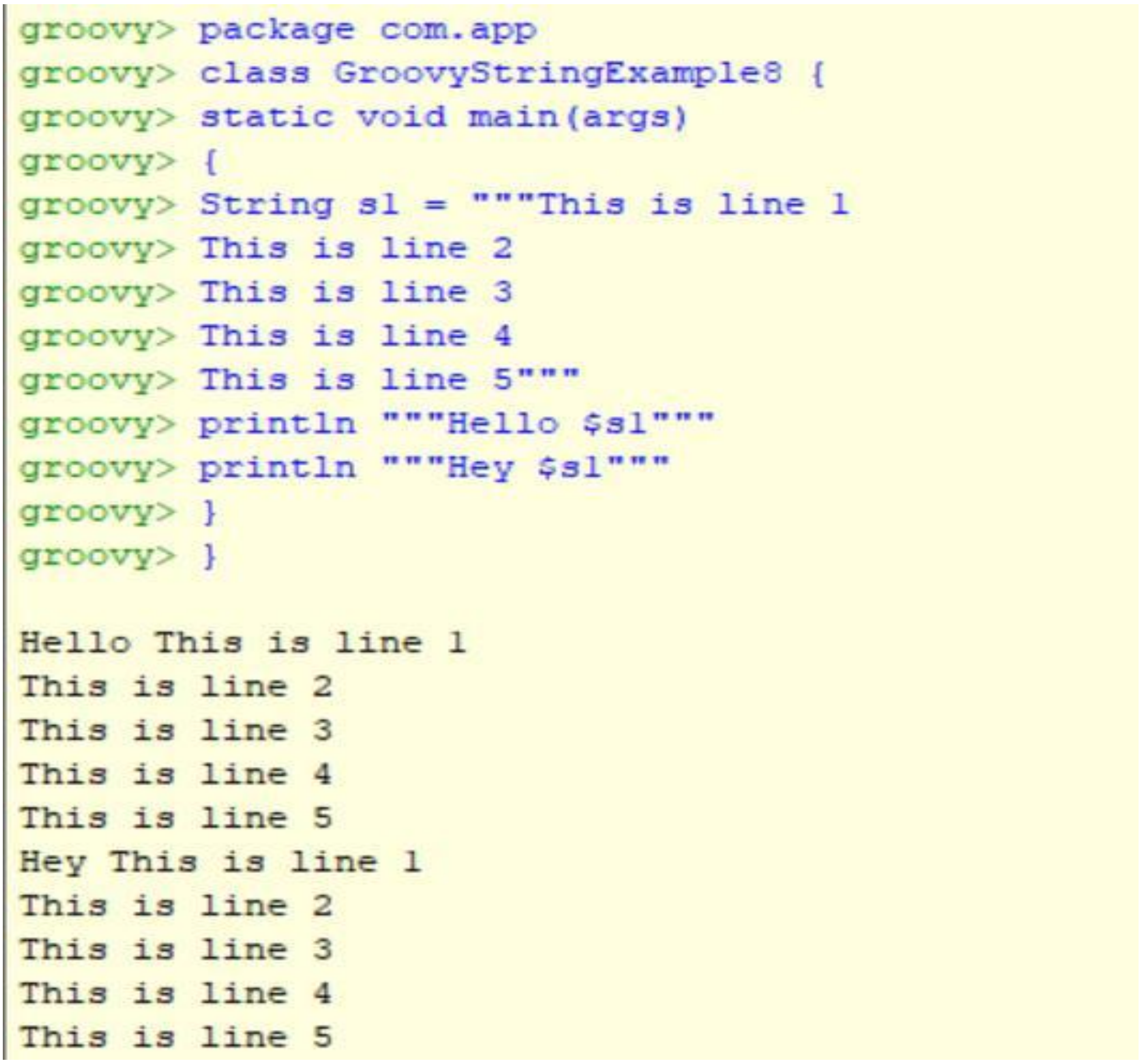
```
groovy> package com.app
groovy> class GroovyStringExample6 {
groovy> static void main(args)
groovy> {
groovy>         String s1 = """This is groovy tutorial and we are learning string"""
groovy>         println s1
groovy> }
groovy> }

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

Example 7:



```
groovy> package com.app
groovy> class GroovyStringExample7 {
groovy> static void main(args)
groovy> {
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
```

Example 8:

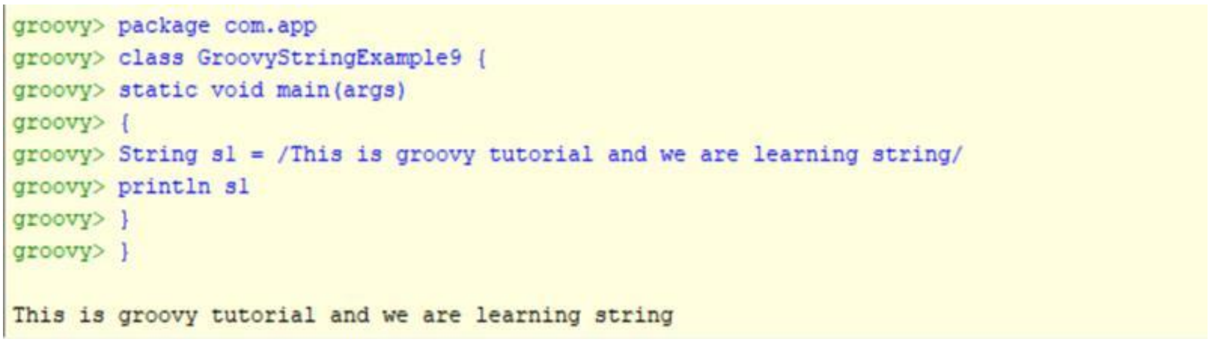


```
groovy> package com.app
groovy> class GroovyStringExample8 {
groovy> static void main(args)
groovy> {
groovy> String s1 = """This is line 1
groovy> This is line 2
groovy> This is line 3
groovy> This is line 4
groovy> This is line 5"""
groovy> println """Hello $s1"""
groovy> println """Hey $s1"""
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
```

Slashy string

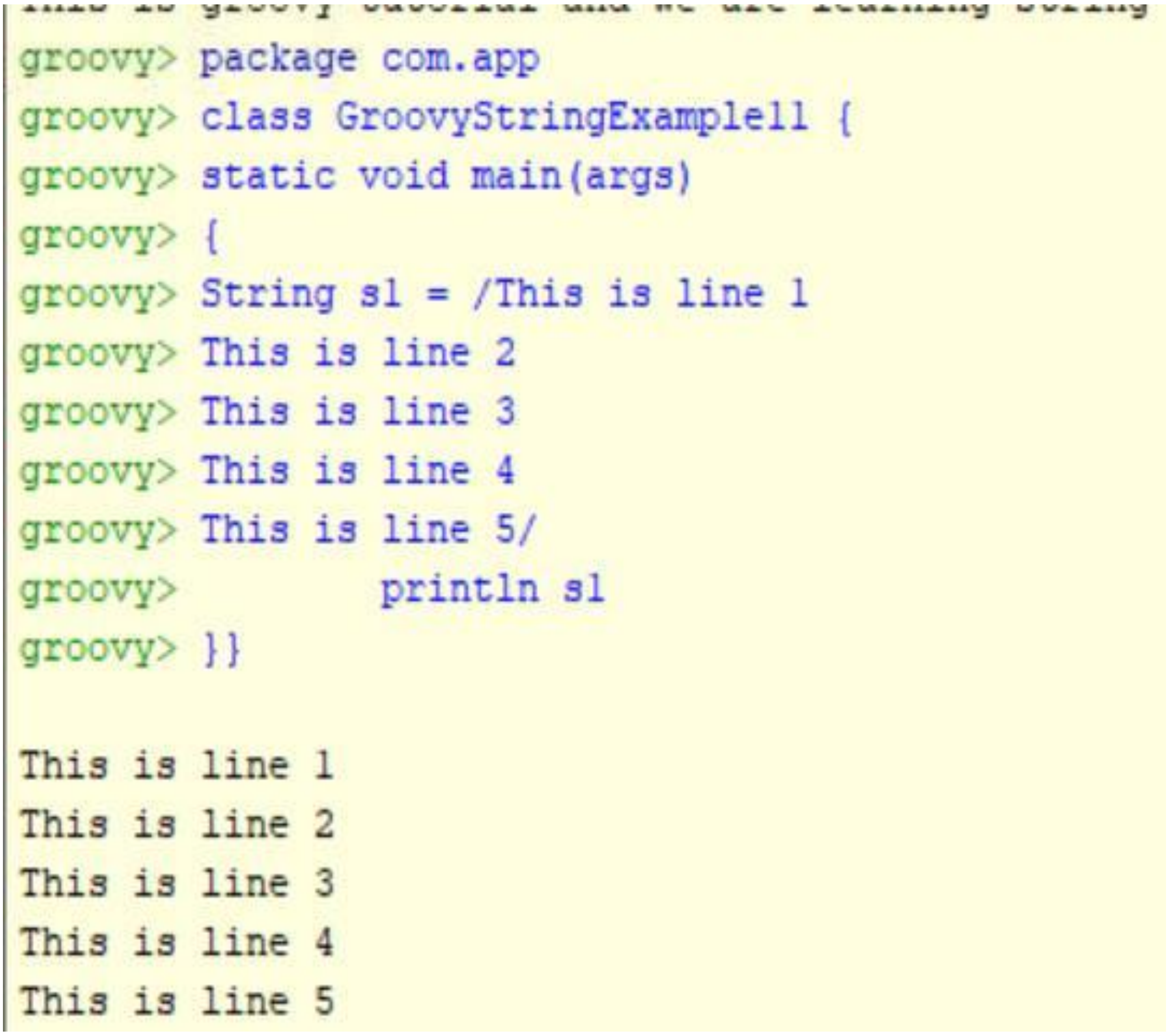
In groovy, slashy string is a sequence of characters which is enclosed in / forward slash. Slashy strings are useful for defining regular expressions and patterns where there is no need to escape backslashes.

Example 9:



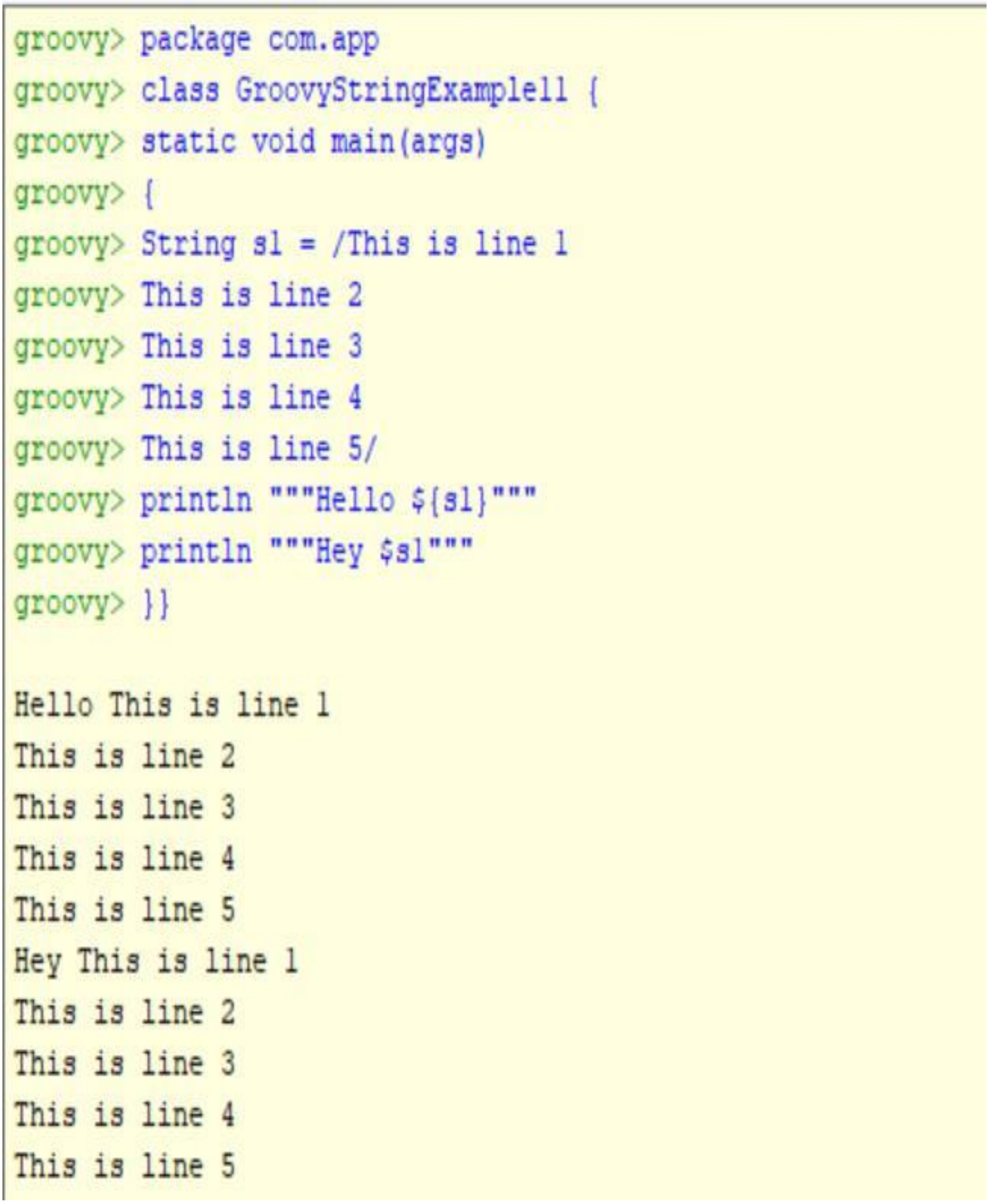
```
groovy> package com.app
groovy> class GroovyStringExample9 {
groovy> static void main(args)
groovy> {
groovy> String s1 = /This is groovy tutorial and we are learning string/
groovy> println s1
groovy> }
groovy> }

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

Example 10:



```
This is groovy tutorial and we are learning string
groovy> package com.app
groovy> class GroovyStringExample11 {
groovy> static void main(args)
groovy> {
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> }}

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

Example 11:



```
groovy> package com.app
groovy> class GroovyStringExample11 {
groovy> static void main(args)
groovy> {
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 """Hello ${s1}"""
groovy> println """Hey $s1"""
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
```

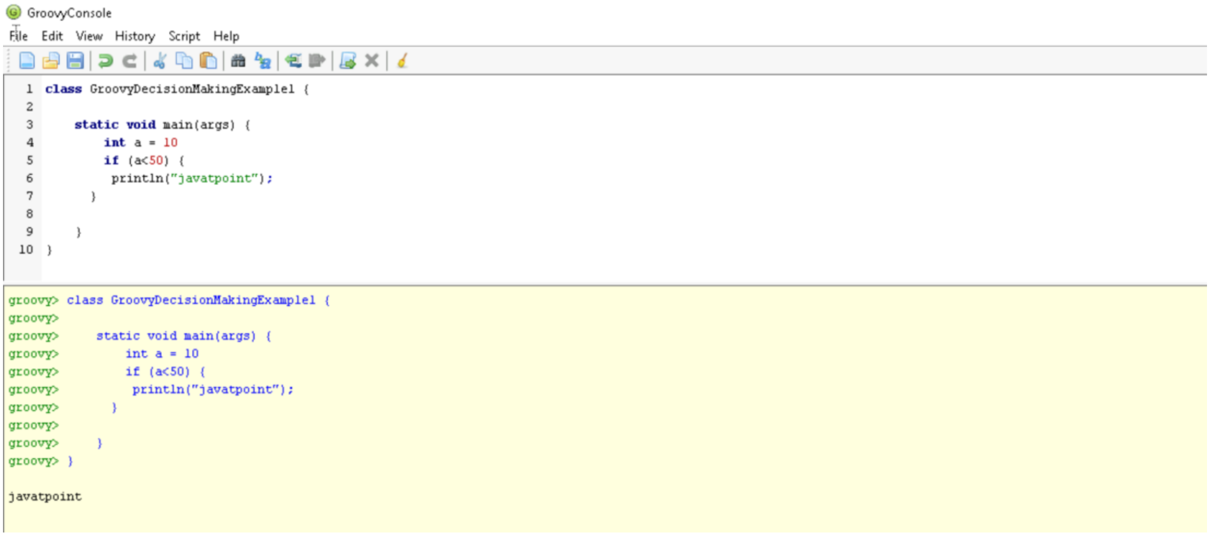
DECISION MAKING:

Decision Making in Groovy

In groovy, decision-making is used to check the condition and execute the statements. if the condition is true then the true block statement is executed and if the condition is false then the false block is executed.

1. IF Statement:

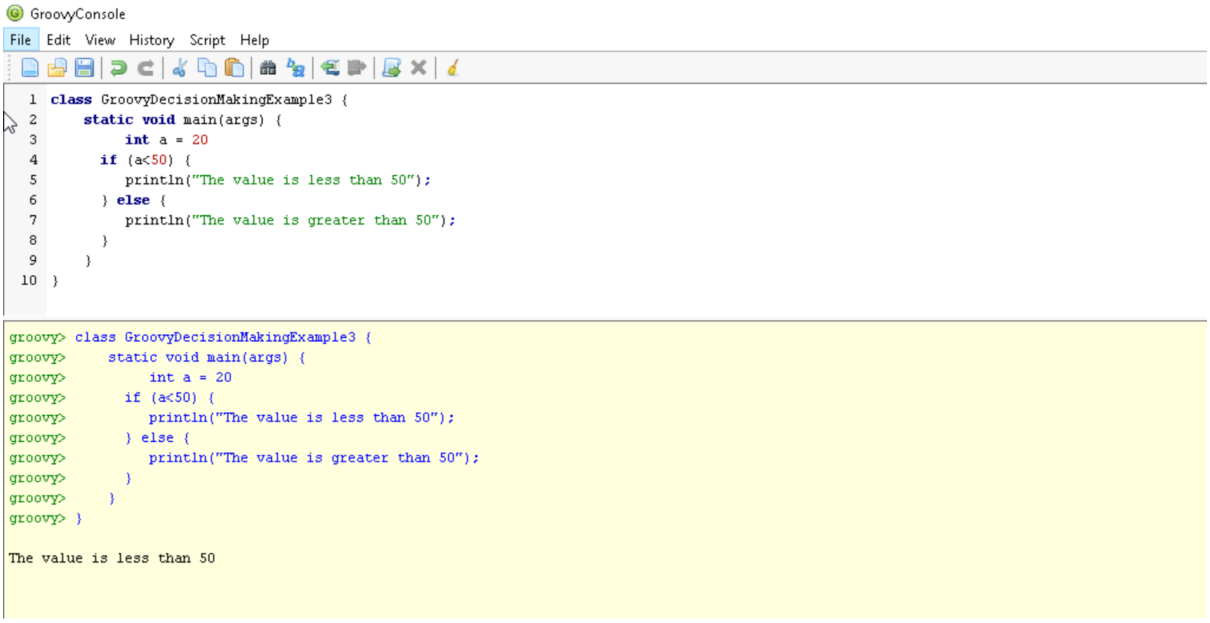
Example 1:



```
class GroovyDecisionMakingExample1 {

    static void main(args) {
        int a = 10
        if (a<50) {
         println("javatpoint");
       }

    }
}
```

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

javatpoint
```

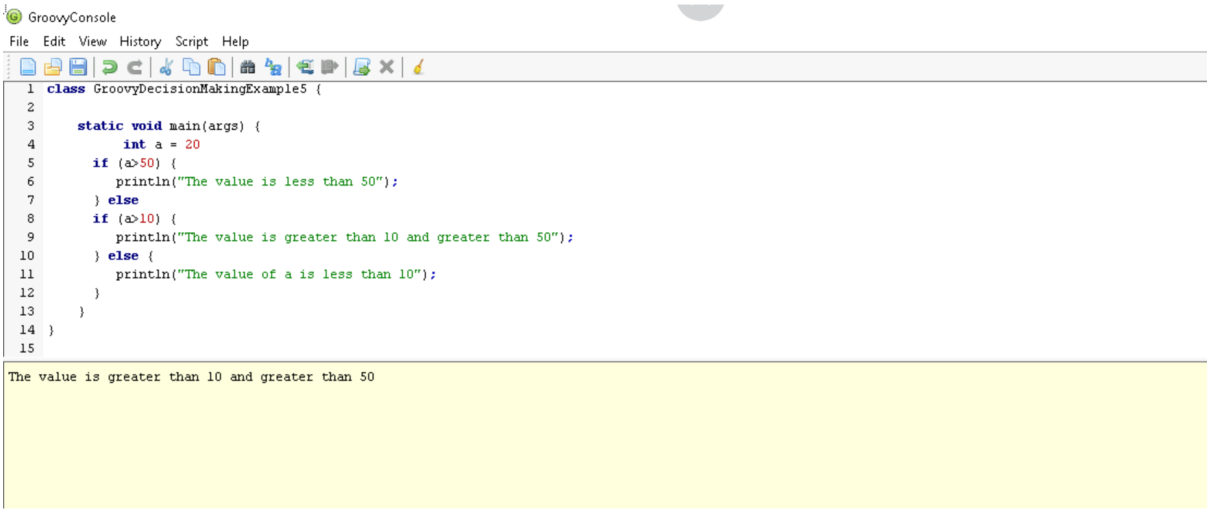
2. IF- ELSE:



```
class GroovyDecisionMakingExample3 {
    static void main(args) {
         int a = 20
      if (a<50) {
         println("The value is less than 50");
      } else {
         println("The value is greater than 50");
      }
    }
}
```

```
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> }

The value is less than 50
```

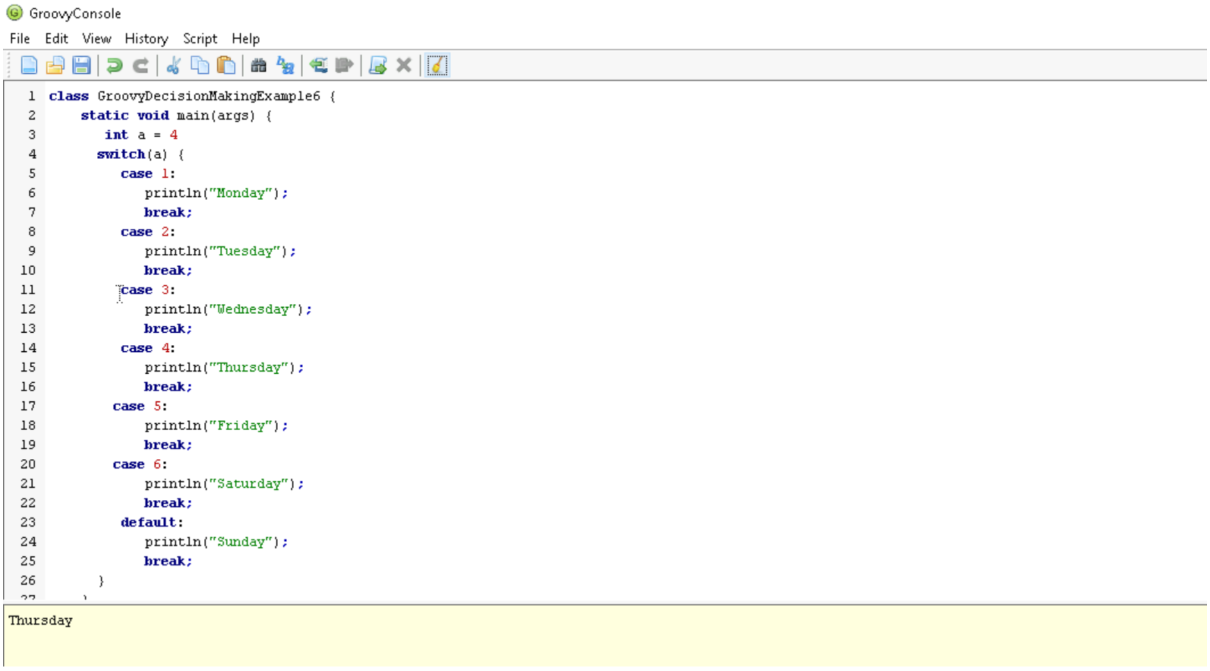
3.NESTED – IF:



```
class GroovyDecisionMakingExample5 {

    static void main(args) {
          int a = 20
      if (a>50) {
         println("The value is less than 50");
      } else
      if (a>10) {
         println("The value is greater than 10 and greater than 50");
      } else {
         println("The value of a is less than 10");
      }
    }
}

```

```
The value is greater than 10 and greater than 50
```

4. SWITCH STATEMENTS:



```
class GroovyDecisionMakingExample6 {
    static void main(args) {
       int a = 4
      switch(a) {
         case 1:
            println("Monday");
            break;
         case 2:
            println("Tuesday");
            break;
         case 3:
            println("Wednesday");
            break;
         case 4:
            println("Thursday");
            break;
        case 5:
            println("Friday");
            break;
        case 6:
            println("Saturday");
            break;
         default:
            println("Sunday");
            break;
      }
```

```
Thursday
```