STRINGS

Single quoted strings

```
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:

```
groovy> package com.app
groovy> class GroovyStringExample2 {
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
```

```
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
```

Triple quoted strings:

```
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
```

```
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
```

```
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
```

```
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
```

```
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
```

```
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
```

```
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
```

```
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:

1. IF Statement:

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

```
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:

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

```
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:

File   Edit   View   History   Script   Help

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

The value is greater than 10 and greater than 50

4. SWITCH STATEMENTS:

```groovy
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
```

WEB SCRAPPING USING GROOVY with Jsoup:

```groovy
@Grab(group='org.jsoup', module='jsoup', version='1.15.3')
import org.jsoup.Jsoup

def topic = "AWS" // Change to any topic
def url = "https://en.wikipedia.org/wiki/${topic}"
def doc = Jsoup.connect(url)
    .userAgent("Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.0.0 Safari/537.36")
    .get()

def summary = doc.select("p")
    .find { it.text().trim() }
    ?.text() ?: "No content found."

println "Wikipedia Summary of ${topic.replace('_', ' ')}:\n$summary"
```

```
Wikipedia Summary of AWS:
Amazon Web Services, Inc. (AWS) is a subsidiary of Amazon that provides on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered,
pay-as-you-go basis. Clients will often use this in combination with autoscaling (a process that allows a client to use more computing in times of high application usage, and then
scale down to reduce costs when there is less traffic). These cloud computing web services provide various services related to networking, compute, storage, middleware, IoT and
other processing capacity, as well as software tools via AWS server farms. This frees clients from managing, scaling, and patching hardware and operating systems. One of the
foundational services is Amazon Elastic Compute Cloud (EC2), which allows users to have at their disposal a virtual cluster of computers, with extremely high availability, which
can be interacted with over the internet via REST APIs, a CLI or the AWS console. AWS's virtual computers emulate most of the attributes of a real computer, including hardware
central processing units (CPUs) and graphics processing units (GPUs) for processing; local/RAM memory; hard-disk (HDD)/SSD storage; a choice of operating systems; networking; and
pre-loaded application software such as web servers, databases, and customer relationship management (CRM).
```

GROOVY LOG FILE ANALYSER:

```groovy
import java.nio.file.*

def logFilePath = "C:/Users/Administrator/Documents/demologfile.txt"
def logFile = new File(logFilePath)
if (!logFile.exists()) {
    println "Error: Log file not found at $logFilePath"
    return
}
def logData = [:].withDefault { 0 }

logFile.eachLine { line ->
    def matcher = line =~ /(\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2})\s+(INFO|WARNING|ERROR)\s+(.+)/
    if (matcher.find()) {
        def timestamp = matcher[0][1]
        def statusCode = matcher[0][2]
        def message = matcher[0][3]
        logData[statusCode]++ // Count occurrences
        println "[${timestamp}] Status: ${statusCode} - ${message}"
    }
}
println "\nLog Analysis Summary:"
if (logData.isEmpty()) {
    println "No matching log entries found."
} else {
    logData.each { key, value -> println "$key : $value occurrences" }
}
```

OUTPUT:

```
[2025-02-17 10:15:32] Status: INFO - Server started successfully.
[2025-02-17 10:16:45] Status: WARNING - High memory usage detected.
[2025-02-17 10:17:12] Status: ERROR - Failed to connect to database.
[2025-02-17 10:20:05] Status: INFO - User 'admin' logged in from IP 192.168.1.100.
[2025-02-17 10:22:30] Status: ERROR - Timeout occurred while processing request.
[2025-02-17 10:25:50] Status: INFO - Scheduled job 'backup' started.
[2025-02-17 10:30:15] Status: WARNING - Disk space running low on /var/log.
[2025-02-17 10:35:40] Status: ERROR - File system corruption detected.
[2025-02-17 10:40:10] Status: INFO - User 'guest' logged out.
[2025-02-17 10:45:25] Status: INFO - Server shutting down for maintenance.

Log Analysis Summary:
INFO : 5 occurrences
WARNING : 2 occurrences
ERROR : 3 occurrences
Result: [INFO:5, WARNING:2, ERROR:3]
```