Faysal Aziz

Groovy Scripting

for Automation and Enterprise Development - 2 days

FA-0236 · 2 days

Software Development · DevOps, Cloud & Infrastructure · Automation, RPA & Power Platform

---

Introduction

Practical Groovy Programming for Real-World Automation, Integration and DevOps

Groovy is widely used across enterprise environments for automation, CI/CD pipelines, build management, system administration, API integration, data processing, and JVM-based application development. Its concise syntax and seamless Java interoperability make it an ideal language for rapidly developing scripts and automation solutions without sacrificing access to the extensive Java ecosystem.

This intensive hands-on course focuses on practical Groovy development from the first session onward. Participants will spend the majority of the course writing code, building scripts, processing data, integrating with external systems, and implementing automation solutions.

The instructor brings over 30 years of industry experience and delivers real-world content based on current enterprise requirements rather than academic theory.

Learning outcomes

Upon completion of this course, participants will be able to:

- Develop Groovy scripts for everyday automation tasks
- Use Groovy collections, closures, and language enhancements effectively
- Create reusable functions and classes
- Process files, XML, and JSON data
- Integrate Groovy scripts with Java libraries

- Consume REST APIs and external services
- Implement robust exception handling and logging
- Develop Jenkins and Gradle automation scripts
- Build maintainable production-ready automation solutions

Prerequisites

- Basic programming experience
- Familiarity with command-line environments
- Basic understanding of scripting concepts
- Java knowledge

Training outline

1. Groovy Development Environment

1. Installing Groovy
2. Installing Java
3. Groovy Console
4. Command-Line Execution
5. IDE Configuration
6. Script Execution Workflows

2. Writing Your First Groovy Scripts

1. Script Structure
2. Variables
3. Data Types
4. User Input
5. Output Formatting
6. Operators
7. Expressions

3. Decision Making and Program Flow

1. if Statements
2. if-else Logic

3. switch Statements
4. Groovy Truth
5. Conditional Expressions

4. Loops and Iteration

1. for Loops
2. while Loops
3. each Iteration
4. Collection Iteration
5. Range-Based Iteration

5. Working with Strings

1. String Operations
2. GStrings
3. String Interpolation
4. Multiline Strings
5. Pattern Matching
6. Regular Expressions

6. Methods and Reusable Code

1. Method Creation
2. Parameters
3. Return Values
4. Optional Parameters
5. Named Parameters
6. Method Overloading

7. Working with Collections

1. Lists
2. Maps
3. Sets
4. Ranges
5. Collection Searching

6. Collection Filtering
7. Collection Transformation
8. Collection Aggregation

8. Groovy Closures

1. Closure Fundamentals
2. Closure Parameters
3. Closure Scope
4. Passing Closures
5. Returning Closures
6. Functional Processing Techniques

9. File Processing

1. Reading Text Files
2. Writing Text Files
3. Directory Traversal
4. File Filtering
5. Log File Processing
6. Batch Processing Workflows

10. Error Handling and Debugging

1. Exception Handling
2. Custom Exceptions
3. Logging Techniques
4. Troubleshooting Scripts
5. Debugging Approaches

11. Object-Oriented Groovy

1. Classes
2. Objects
3. Constructors
4. Properties
5. Methods

6. Inheritance
7. Traits

12. Java Integration

1. Calling Java Classes
2. Using Java Libraries
3. Importing External Dependencies
4. Working with Existing Enterprise APIs

13. JSON Processing

1. Parsing JSON
2. Generating JSON
3. Transforming JSON Data
4. API Payload Processing

14. XML Processing

1. Parsing XML
2. Navigating XML Documents
3. Generating XML
4. XML Transformation

15. REST API Integration

1. HTTP Requests
2. GET Operations
3. POST Operations
4. Authentication Concepts
5. Processing API Responses
6. Error Handling for APIs

16. Automation Use Cases

1. System Administration Scripts
2. Data Processing Scripts
3. Report Generation

4. File Management Automation
5. Scheduled Task Automation
6. Workflow Automation

17. Jenkins Groovy Scripting

1. Jenkins Pipeline Fundamentals
2. Scripted Pipelines
3. Declarative Pipelines
4. Shared Libraries
5. Pipeline Automation Patterns

18. Gradle Scripting

1. Gradle Build Scripts
2. Task Automation
3. Dependency Management
4. Build Customization

19. Production Best Practices

1. Script Organization
2. Reusable Components
3. Coding Standards
4. Configuration Management
5. Security Considerations
6. Maintainability Guidelines

Disclaimer

This outline serves as a proposed training framework and may be adjusted at the instructor's discretion to accommodate participant experience levels, organizational objectives, technology changes, and specific operational requirements. Topic sequencing, depth of coverage, practical exercises, and demonstrations may be modified as necessary to maximize learning outcomes and ensure alignment with current industry practices.

© 2026 Faysal Aziz · Kuala Lumpur