QA ADVANCED 5 DAYS

Groovy Scripting

Build a strong foundation in Groovy programming by mastering core syntax, scripting, and automation concepts. Learn how to integrate Groovy with Java systems, work with databases and web services, and apply testing techniques to support efficient application development and automation.

Why Groovy?

Productivity

With Groovy's concise and expressive syntax, developers can write code faster which can improve productivity and reduce development time.

Automation

Groovy is well-suited for automation and scripting tasks, which can help businesses to streamline their workflows and reduce the burden on employees.

Integration

Groovy can easily integrate with existing Java code and libraries, making it an ideal choice for businesses that have already invested in Java-based systems and applications.

What you'll be able to do

- ✓ Understanding the fundamentals of Groovy syntax, control structures, and data types.
- ✓ Learning how to work with collections, strings, and regular expressions in Groovy.
- ✓ Developing the ability to write scripts and automate tasks using Groovy.
- ✓ Learning how to integrate Groovy with existing Java-based systems and libraries.
- ✓ Understanding how to work with databases and web services using Groovy.
- ✓ Developing the ability to write unit tests and perform automated testing using Groovy.

Who this is for

- Developers
- Test automation engineers
- DevOps engineers
- Technical leads and architects
- Data scientists
- Software engineers
- System administrators
- Quality assurance engineers

Curriculum

01 Groovy Foundations

- Apply groovy foundations using Groovy in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

02 Syntax and Control Flow

- Apply syntax and control flow using Groovy in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

03 Collections

- Apply collections using Groovy in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

04 OOP and Metaprogramming

- Apply oop and metaprogramming using Groovy in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

05 Closures and Builders

- Apply closures and builders using Groovy in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

06 Data, XML, JSON, and SQL

- Apply data, xml, json, and sql using Groovy in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

07 Testing with JUnit/TestNG

- Apply testing with junit/testng using Groovy in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

08 Gradle

- Apply gradle using Groovy in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

09 Debugging

- Apply debugging using Groovy in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

10 Performance

- Apply performance using Groovy in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

11 Best Practices

- Apply best practices using Groovy in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks