

JavaScript and Groovy Programming

Duration : 20hrs(4hrs/day)

Target Audience: Developers

Day 1: JavaScript Fundamentals & Developer Essentials

- Introduction to JavaScript, ECMAScript versions, and real-world use cases
- Variables: var, let, and const, understanding scope
- JavaScript data types and type coercion
- Operators: arithmetic, logical, relational
- Control structures: if-else, switch, for, while, do-while loops
- Function declarations, expressions, arrow functions
- Understanding scope, hoisting, closures (intro)

Lab:

- Write conditional and looping scripts
 - Build a simple calculator or decision-based app using functions
-

Day 2: JavaScript Advanced Concepts

- Arrays and objects: creation, manipulation, iteration
- JSON: parse, stringify, use cases in APIs
- Higher-order functions: map(), filter(), reduce()
- Callback functions and asynchronous behavior
- Promises and async/await
- Error handling with try-catch-finally
- Basic debugging using browser console

Lab:

- Parse JSON data and manipulate it
 - Simulate API call behavior with Promises or async/await
-

Day 3: Introduction to Groovy Scripting

- What is Groovy and where is it used (Jenkins, Gradle, scripting)
- Groovy vs Java syntax
- Script mode vs class mode
- Groovy data types and dynamic typing
- Working with lists, maps, and ranges
- Defining and using closures
- Control flow in Groovy: if, switch, loops

Lab:

- Write scripts using Groovy collections and closures
 - Build small automation tasks using loops and conditions
-

Day 4: Groovy for Automation & DSLs

- Working with strings and files in Groovy
- JSON and XML parsing using Groovy built-in libraries
- Exception handling in Groovy
- Overview of Jenkins pipeline as code
- Declarative vs scripted pipelines in Jenkins
- Writing a Jenkinsfile with Groovy DSL

Lab:

- File operations and content parsing
 - Jenkins pipeline example using scripted and declarative styles
-

Day 5: Integration, Real-world Use Cases & Mini-Project

- Use cases comparison: JavaScript vs Groovy in automation and web
- Integration examples:
 - JavaScript for frontend logic or lightweight scripts
 - Groovy for Jenkins, CI/CD, and build scripting
- Project walkthrough combining both scripts:
 - Use JavaScript to extract and format data
 - Use Groovy to process data or automate deployment

Lab:

- Complete a mini-project integrating both JavaScript and Groovy
- Prepare a reusable script library or template for your environment