

**Jenkins Pipeline Scripting**

**Course Number:** JEN-116WA
**Duration:** 2 days

**Overview**

This Jenkins Pipeline Scripting training course teaches attendees how to automate the continuous integration and continuous delivery (CI/CD) processes using Jenkins. Participants get an in-depth look at the Jenkins pipeline system and learn how to write powerful Groovy scripts to automate builds across multiple machines.

**Prerequisites**

All attendees must know the basics of Jenkins.

**Materials**

All Jenkins training students receive comprehensive courseware.

**Software Needed on Each Student PC**

Attendees will not need to install any software on their computer for this class. The class will be conducted in a remote environment that Accelebrate will provide; students will only need a local computer with a web browser and a stable Internet connection. Any recent version of Microsoft Edge, Mozilla Firefox, or Google Chrome will be fine.

**Objectives**

* Understand CI/CD pipeline concepts
* Use Groovy programming to automate tasks
* Build, test, and deploy activities following Jenkins' best practices
* Use libraries
* Implement best practices for pipeline code

**Outline**

* Introduction
* Jenkins Essentials Refresh
	+ Continuous integration, continuous delivery (CI/CD)
	+ Jenkins as orchestration for build tasks
	+ Jenkins job types
	+ Scheduling and triggering jobs
	+ Security
* Pipeline Concepts
	+ Role for pipelines
	+ Declarative and scripted pipelines
	+ Declarative pipeline end-to-end example
	+ Scripted pipeline end-to-end example
* Pipeline Domain-Specific Language
	+ Agents and nodes
	+ Stages and steps
	+ Workspaces
	+ Shell and tools
	+ Credentials and secret management
	+ User input
	+ Flow control
	+ Stashing and archiving
	+ Notification
* Distributed Builds
	+ Designing for scale
	+ Controlling where jobs run
	+ Configuration management
	+ Considerations for distributed builds
	+ State management
	+ Concurrency concerns
	+ Latency concerns
* Selected Plugins
	+ Git, GitHub, GitLab, Bitbucket
	+ Gerrit
	+ Artifactory
	+ Jira
	+ Sonar
* Groovy for Pipelines
	+ Groovy and Pipeline Groovy
	+ How pipeline Groovy works
	+ @NonCPS annotation
	+ Good practices for pipeline code
* Libraries
	+ Role of libraries
	+ Defining libraries
	+ Library structure
	+ Using libraries
* Extending Pipelines with Groovy
	+ Groovy syntax
	+ Data types
	+ Operators
	+ Flow control
	+ Try/catch and exceptions
	+ Functions
	+ Closures
	+ Classes
* Best Practices
	+ Best practices for writing and using pipelines
	+ Common mistakes and how to avoid them
* Conclusion