

**Git for Developers**

**Course Number:** GIT-100
**Duration:** 2 days

**Overview**

Accelebrate's Git for Developers training course teaches mobile, desktop and web developers to use Git, the leading software version control system. Git is distributed, free, and appropriate for development projects in almost any language.

**Prerequisites**

No prior experience with Git is presumed.  Prior experience with other version control systems is helpful but not required.

**Materials**

All attendees receive comprehensive course materials.

**Software Needed on Each Student PC**

* Git 2.x or later
* Internet access for all attendees and the instructor

**Objectives**

* Understand Git and Git fundamentals
* Review and edit the commit history
* Improve your daily workflow
* Branch, merge, and use remote repositories
* Use Git as a debugging tool
* Learn advanced Git concepts including object types, bundles, and more

**Outline**

* Introduction to Source Code Management
	+ The Core Principles of Change Management
	+ The Power to Undo Changes
	+ Audit Trails and Investigations
	+ Reproducible Software
* Git Introduction and Basics
	+ Introduction to Git
	+ Trees and Commits
	+ Configuring Git
	+ Adding, Renaming, and Removing Files
* Reviewing and Editing the Commit History
	+ Reviewing the Commit History
	+ Revision Shortcuts
	+ Fixing Mistakes
* Improving Your Daily Workflow
	+ Simplifying Common Commands with Aliases
	+ Ignoring Build Artifacts
	+ Saving Changes for Later Use (Stashing)
* Branching
	+ Branching Basics
	+ Listing Differences Between Branches
	+ Visualizing Branches
	+ Deleting Branches
	+ Tagging
* Merging
	+ Merging Basics
	+ Merge Conflicts
	+ Merging Remote Branches
* Remote Repositories
	+ Remote Repositories
	+ Synchronizing Objects with Remotes
	+ Tracking Branches
* Centralizing and Controlling Access
	+ Introduction to GitLab
	+ Git Repositories on GitLab
	+ Daily Workflow
* Reviewing Branching and Merging
	+ Branch Review
	+ Merging Basics
* Rebasing
	+ Rebasing Basics
	+ Rebasing with Local Branches
	+ Rebasing with Remote Branches
	+ Interactive Rebasing
	+ Squashing Commits
	+ Getting Out of Trouble
* Resetting Trees
	+ Introduction to Resetting
	+ Resetting Branch Pointers
	+ Resetting Branches and the Index
	+ Resetting the Working Directory
	+ Making Good Use of the Reset Command
* More on Improving Your Daily Workflow
	+ Interactively Staging Changes
* Including External Repositories
	+ Submodules
	+ Subtrees
	+ Choosing Between Submodules and Subtrees
* Git as a Debugging Tool
	+ Using the Blame Command to See File History
	+ Performing a Binary Search
* Workflow Management
	+ Branch Management
* Advanced Concepts
	+ Git Object Types
	+ Content-Addressable Names
	+ Pack Files
	+ Bundles
* Conclusion