

Git Fundamentals

Introduction

The Git Fundamentals course is focused on helping software developers, DevOps, system administrators, and QA teams wanting to learn how to use Git. Working with Git covers all of the fundamental operations an experienced coder would use on a daily basis. The course begins with an introduction to Git and a comparison of Git to other version control systems. It then transitions into the nuts-and-bolts of working with Git, including everything from setting up a repository to advanced topics like branching and merging. You'll also learn how to incorporate Git with common IDEs.

Duration: 2 days

Course objectives

After this course, you will be able to:

- Understand how Git compares to other systems
- Install and configure Git in your team development environment
- Create new repositories and downloading existing ones
- Describe a typical source control workflow
- Implement a typical source control workflow using Git

Target Audience

- Developers, QA teams and others wanting to learn Git
- Current Git users wanting to deepen their understanding of how to use Git

Prerequisites

- Familiarity with a command line interface or terminal (or interest in learning during class).

Technical Requirements

Students should be prepared to perform the following tasks

- Install Git version 2.25 or above (1.8+ is OK, but they will miss out on some newer commands)
- Have a simple text editor installed and available (Git defaults to "Vim" and many people like to switch away from that). Alternatives: Emacs, Sublime Text, VS Code, Notepad++, Notepad
- Register on vanilla Github.com -- alternatively GitLab.com can be used

Course Modules

Introduction to Git

- History of Git
- Who is using Git
- Core Git concepts
- Adopting Git

Setting up Git

- Installing Git
- Configuring Git
- Accessing Git
- **Lab:** Installing and configuring Git

Working with Git

- Understanding the repository structure
- Adding, committing, and viewing commits
- Renaming, moving and removing files
- Undoing
- Analyzing logs and history
- Ignoring files
- **Lab:** Working with basic Git commands

Remote Repositories

- Working with a remote repository
- Setting up / publishing a repository
- More on the the repository structure
- Working with multiple repositories
- Working with GitHub
- **Lab:** Working with remote repositories

Branching, Merging and Conflict resolution

- How Git manages history
- Understanding tagging
- Creating and managing branches
- Merging branches
- Analyzing diffs and their options
- Resolving and merging conflicts
- Creating remote branches
- Pruning branches
- **Lab:** Working with branches

Getting more out of Git

- Cherry picking
- Stashing
- Hooks
- Visual Tools
- **Lab:** Using advanced features

Integrating Git with Development workflow

- Incorporating Git with IDEs
- Using Git with automated build systems
- Incorporating Git with Jenkins