Git Essentials

1 Day

Classroom, Webinar

The Clearvision Git Essentials training course provides a comprehensive understanding of basic Git concepts and looks at the different ways Git can be used and implemented.

About this course

Is this course right for me?

This course is ideal for anyone new to Git version control, or anyone who is familiar with Git but could do with a stronger understanding.

Objectives and outcomes

With the Clearvision Git Essentials course, you'll become an expert at using Git for day-to-day development.

The instructor will show you how Git really works, not just the commands you'll need but the impact on both your local and remote repository.

You'll learn about branches and strategy and work with the index to manipulate code in your local repository and perform merges while solving conflicts. Finally, you'll dive deep into Git's internal set up so that collaborating with other users won't come as a surprise, preparing you for whatever your colleagues do next.

Prerequisites

-Access to a laptop.

-Internet connection

Modules breakdown

Module 1: Introduction

Module Sections

1. Overview
2. The need for version control
3. A brief history of Git and what it is (i.e. a DVCS)
4. Defining a DVCS
5. Plumbing and porcelain
6. The repository (at high-level)
7. Getting started

Lab Exercises

1. Group discussion, understanding DVCS

Module 2: Git concepts

Module Sections

1. Git object types
   - Blobs
   - Trees
   - Commits
   - Tags
   - Git reference types
   - Object model example
2. Hash values (SHA-1)
3. The Git data model

Lab Exercises

1. Verify Git installations
2. Identify the Git repository
3. Ask Git for help

Module 3: Getting started

Module Sections

1. How to create a new Git repository
2. Setting up your Git environment
3. Basic workflow
4. Adding new files
5. Changing existing files
6. Committing changes to the repository
7. The Git log

Lab Exercises

1. Identify yourself to Git
2. Create a Git repository
3. How to add new files/folders
4. How to commit new files/folders
5. Clone an existing repository
6. Check the status of your Git repository

Module 4: Working with Git

Module Sections

1. Workflow
2. The status command
3. Ignoring file types
   - Negation in ignore files
4. View specific changes
5. Removing files and directories
6. Moving files and directories
7. Stashing changes
8. Undoing or fixing errors
9. Reset a change
10. Revert a change
11. Checkout a change

Lab Exercises

1. Use the Git log command to help you clarify if you are committing the correct version of a file
2. Use the Git status command to help you clarify the current state of your repository

Module 5: Branching and merging

Module Sections

1. Defining a branch
2. Creating a branch
3. Switching between branches
4. Defining a merge
5. Fast forward merge
6. 3-way merge
7. Resolving merge conflicts
8. Merge tools
9. Removing branches
10. Rebasing
    - Autostash
11. Branch management in Git

Lab Exercises

1. Create branches in Git
2. Add files to branches
3. Checkout branches
4. Merge into branches
5. Rebase and squash a feature branch

Module 6: Collaboration basics

Module Sections

1. Cloning repositories
2. Remotes
3. Remote branches and tracking branches
4. Fetching, pulling and pushing changes
5. Bare and development repositories
6. Publishing repositories

Lab Exercises

1. Clone a repository
2. Create or identify a remote
3. Pull changes
4. Push changes
5. Merging between repositories

Module 7: Collaboration strategies

Module Sections

1. Branching strategies
2. Structural strategies

Lab Exercises

1. A flexible exercise, implementing the strategy you intend to use moving forward

Module 8: Tagging

Module Sections

1. Defining a tag
2. Viewing tags
3. Creating tags
4. Signed tags
5. Tagging later
6. Sharing tags

Lab Exercises

1. Create tags in Git
   - Lightweight tags
   - Annotated tags
   - Checkout tags in Git
2. How to view tags
3. How to checkout tags