# Mastering Git

## Introduction

* **What is Version Control?**
* **Introduction to Git**
  + History of Git
  + Why Use Git?
* **Installing Git**
  + Installation on Windows
  + Installation on macOS
  + Installation on Linux
* **Configuring Git for the First Time**
  + Setting Up Username and Email
  + Choosing a Default Text Editor
  + Configuring Aliases
  + Understanding Git Configuration Levels (System, Global, Local)

## Chapter 1: Git Basics

* **Getting Started with Git**
  + Initializing a Repository (git init)
  + Cloning an Existing Repository (git clone)
* **Understanding the Three States**
  + Working Directory
  + Staging Area
  + Repository (Commit History)
* **Basic Workflow**
  + Checking Repository Status (git status)
  + Tracking New Files (git add)
  + Committing Changes (git commit)
* **Viewing the Commit History**
  + Simple Logs (git log)
  + Formatting Log Output
* **Ignoring Files**
  + Creating a .gitignore File
  + Patterns for Ignoring Files and Directories

## Chapter 2: Making Changes

* **Editing Files**
  + Best Practices for Editing
* **Viewing Changes**
  + Difference Between Working Directory and Staging Area (git diff)
  + Difference Between Staging Area and Last Commit (git diff --staged)
* **Undoing Changes**
  + Discarding Changes in Working Directory (git checkout --)
  + Unstaging Files (git reset HEAD)

## Chapter 3: Working with Branches

* **Understanding Branches**
  + What Are Branches?
  + The HEAD Pointer
* **Basic Branching**
  + Listing Branches (git branch)
  + Creating a New Branch (git branch <branch-name>)
  + Switching Branches (git checkout <branch-name>)
  + Creating and Switching Branches (git checkout -b <branch-name>)
* **Merging Branches**
  + Fast-Forward Merge
  + Three-Way Merge
  + Resolving Merge Conflicts
* **Deleting Branches**
  + Deleting a Local Branch (git branch -d <branch-name>)
  + Force Deletion (git branch -D <branch-name>)

## Chapter 4: Remote Repositories

* **Understanding Remotes**
  + What Is a Remote?
  + Viewing Remotes (git remote)
* **Managing Remotes**
  + Adding a Remote (git remote add)
  + Removing and Renaming Remotes
* **Fetching and Pulling**
  + Fetching Changes (git fetch)
  + Pulling Changes (git pull)
* **Pushing Changes**
  + Pushing to a Remote (git push)
  + Tracking Remote Branches

## Chapter 5: Tagging

* **Understanding Tags**
  + Lightweight vs. Annotated Tags
* **Creating Tags**
  + Creating a Lightweight Tag (git tag <tagname>)
  + Creating an Annotated Tag (git tag -a <tagname> -m "message")
* **Managing Tags**
  + Listing Tags
  + Deleting Tags
  + Sharing Tags (git push origin --tags)

## Chapter 6: Stashing and Cleaning

* **Stashing Changes**
  + Saving Unfinished Work (git stash)
  + Listing Stashes (git stash list)
  + Applying Stashes (git stash apply)
  + Dropping Stashes (git stash drop)
* **Cleaning Up Untracked Files**
  + Removing Untracked Files (git clean)

## Chapter 7: Rebasing

* **Introduction to Rebasing**
  + What Is Rebasing?
  + When to Rebase vs. Merge
* **Basic Rebasing**
  + Rebasing Current Branch onto Another (git rebase <base-branch>)
* **Interactive Rebasing**
  + Editing Commits (git rebase -i)
  + Squashing Commits
  + Reordering Commits

## Chapter 8: Resetting, Reverting, and Recovering

* **Resetting Commits**
  + Soft, Mixed, and Hard Resets (git reset)
* **Reverting Changes**
  + Reverting a Commit (git revert)
* **Recovering Lost Commits**
  + Using the Reflog (git reflog)

## Chapter 9: Advanced Git Log

* **Customizing Git Log Output**
  + Pretty Formats
  + Graphical Representation (git log --graph)
* **Searching Through Commit History**
  + Filtering Commits by Author, Date, Message

## Chapter 10: Git Aliases and Configuration

* **Creating Aliases**
  + Shortening Common Commands
* **Advanced Configuration**
  + Customizing Git Behavior
  + Using Include Files

## Chapter 11: Git Hooks

* **Introduction to Hooks**
  + Client-Side vs. Server-Side Hooks
* **Common Hooks**
  + Pre-Commit, Commit-Msg, Post-Commit
* **Writing Custom Hooks**
  + Scripting in Bash, Python, etc.
* **Security Considerations**

## Chapter 12: Submodules and Subtrees

* **Git Submodules**
  + Adding a Submodule (git submodule add)
  + Cloning Repositories with Submodules
  + Updating Submodules
* **Git Subtrees**
  + Adding a Subtree
  + Merging and Pulling in Subtrees

## Chapter 13: Working with Binary Files and Large Files

* **Git LFS (Large File Storage)**
  + Installing Git LFS
  + Tracking Files with LFS
* **Best Practices for Binary Files**

## Chapter 14: Collaborating with Git

* **Pull Requests and Code Reviews**
  + Workflow on Platforms like GitHub, GitLab, Bitbucket
* **Branching Strategies**
  + Feature Branch Workflow
  + Git Flow Workflow
  + Trunk-Based Development
* **Continuous Integration/Continuous Deployment (CI/CD)**
  + Integrating Git with CI/CD Tools
* **Best Practices for Team Collaboration**
  + Commit Message Guidelines
  + Code Style and Linting
  + Handling Merge Conflicts in a Team

## Chapter 15: Git Internals

* **Plumbing and Porcelain Commands**
* **Git Object Model**
  + Blobs, Trees, Commits, Tags
* **Understanding the .git Directory**
  + Exploring Git's Internal Files and Structures
* **Content-Addressable Storage**
* **Object Packing**

## Chapter 16: Security in Git

* **Signing Commits and Tags**
  + GPG Signing
* **Managing Credentials**
  + Credential Helpers
  + Storing Credentials Securely
* **Sensitive Data**
  + Removing Sensitive Data from History
  + Using git filter-repo and BFG Repo-Cleaner

## Chapter 17: Performance Tuning and Large Repositories

* **Optimizing Git Performance**
  + Garbage Collection (git gc)
  + Repacking Objects
* **Shallow Clones**
  + Cloning Repositories Partially
* **Sparse Checkout**
  + Checking Out Specific Paths

## Chapter 18: Git and Other Platforms

* **Working with GitHub**
  + Forking Repositories
  + GitHub Actions
* **Working with GitLab**
  + GitLab CI/CD
* **Working with Bitbucket**
  + Pipelines
* **Integration with Other Tools**
  + IDE Integration (VS Code, IntelliJ, etc.)
  + Issue Trackers

## Chapter 19: New and Advanced Features

* **Partial Clone and Fetch**
* **Commit Graph**
* **Scalar and Performance Improvements**
* **Git Worktrees**
  + Managing Multiple Working Directories

## Chapter 20: Troubleshooting and Best Practices

* **Common Git Problems**
  + Merge Conflicts
  + Detached HEAD State
* **Debugging Techniques**
* **Best Practices**
  + Atomic Commits
  + Clean Commit History
  + Regularly Pulling and Rebasing

## Appendices

### Appendix A: Git Command Reference

* **Common Commands**
* **Advanced Commands**

### Appendix B: Resources

* **Official Documentation**
* **Community Forums**
* **Recommended Books and Tutorials**

### Appendix C: Glossary of Terms

* **Key Git Terminology**

————————

This table of contents is designed to take you on a comprehensive journey through Git, starting from the basics and moving toward advanced topics that will turn you into a Git hero. Each chapter builds upon the previous ones, ensuring a solid understanding of both fundamental and complex aspects of Git, including the latest features up to October 2023.

#software/git