

Day - 1

- **What is Git?**

- Understanding version control
- The history of Git
- About distributed version control
- Who should use Git?

- **Installing Git**

- Installing Git on a Mac
- Installing Git on Windows
- Installing Git on Linux
- Configuring Git
- Exploring Git auto-completion
- Using Git help

- **Backup and Restore**

- Installing Git on a Mac
- Installing Git on Windows
- Installing Git on Linux
- Configuring Git
- Exploring Git auto-completion
- Using Git help

- **Getting Started**

- Initializing a repository
- Understanding where Git files are stored
- Performing your first commit
- Writing commit messages
- Viewing the commit log

- **Git Concepts and Architecture**

- Exploring the three-trees architecture
- The Git workflow
- Using hash values (SHA-1)
- Working with the HEAD pointer

- **Making Changes to Files**

- Adding files
- Editing files
- Viewing changes with diff
- Viewing only staged changes
- Deleting files
- Moving and renaming files
- Using Git with a Real Project
- Introducing the Explore California web site
- Initializing Git
- Editing the support phone number
- Editing the backpack file name and links

- **Undoing Changes**

- Undoing working directory changes
- Unstaging files
- Amending commits
- Retrieving old versions
- Reverting a commit
- Using reset to undo commits
- Demonstrating a soft reset
- Demonstrating a mixed reset
- Demonstrating a hard reset
- Removing untracked files

- **Ignoring Files**

- Using .gitignore files
- Understanding what to ignore
- Ignoring files globally
- Ignoring tracked files
- Tracking empty directories

- **Navigating the Commit Tree**

- Referencing commits
- Exploring tree listings
- Getting more from the commit log
- Viewing commits
- Comparing commits

- **Branching**

- Branching overview
- Viewing and creating branches
- Switching branches
- Creating and switching branches
- Switching branches with uncommitted changes
- Comparing branches
- Renaming branches
- Deleting branches
- Configuring the command prompt to show the branch

- **Merging Branches**

- Merging code
- Using fast-forward merge vs. true merge
- Merging conflicts
- Resolving merge conflicts
- Exploring strategies to reduce merge conflicts
- Fast Forward Merge
- Three way Merge
- Rebasing

- **Remotes**

- Using local and remote repositories
- Adding a remote repository
- Creating a remote branch
- Cloning a remote repository
- Tracking remote branches
- Pushing changes to a remote repository
- Fetching changes from a remote repository
- Merging in fetched changes
- Checking out remote branches
- Pushing to an updated remote branch
- Deleting a remote branch
- Enabling collaboration
- A collaboration workflow

- **Tools and Next Steps**

- Setting up aliases for common commands
- Using SSH keys for remote login
- Exploring integrated development environments
- Exploring graphical user interfaces
- Understanding Git hosting
- Conclusion

- **Tagging**

- Lightweight tags
- Annotated tags
- Signed Tags
- Create tags in Git
- Create tags in Stash
- Checkout tags in Git
- How to view tags
- How to checkout tags

- **Workflow Strategies**

- Branching basics
- Forking basics
- Centralized Workflow
- Feature Branch Workflow
- Gitflow Workflow
- Forking Workflow
- Git Troubleshooting

- **SourceTree: Local Repositories**

- What is SourceTree?
- Installing SourceTree
- Creating a Local Repository Through SourceTree
- Editing a Committed File
- Adding a Tag
- Branching
- .gitignore in SourceTree
- Removing Files

- **SourceTree: Remote Repositories**

- Creating a Repository Through Bitbucket
- Cloning the Repository
- Working with the Remote Repository