1. Initialization and Cloning

git init

Description: Initializes a new Git repository.

Usage: git init

git clone

Description: Clones an existing repository into a new directory.

Usage: git clone <repository\_url>

2. Configuration

git config

Description: Sets configuration values for your Git installation.

Usage: git config [--global] <key> <value>

Example: git config --global user.name "Your Name"

3. Basic Snapshotting

git add

Description: Adds file contents to the staging area.

Usage: git add <file\_or\_directory>

git commit

Description: Records changes to the repository.

Usage: git commit -m "commit message"

git status

Description: Shows the working directory and staging area status.

Usage: git status

git diff

Description: Shows changes between commits, commit and working tree, etc.

Usage: git diff [options] [<commit>] [--] [<path>...]

git reset

Description: Resets current HEAD to the specified state.

Usage: git reset [<mode>] [<commit>]

Example: git reset --hard HEAD~1 (resets to one commit before the current HEAD)

4. Branching and Merging

git branch

Description: Lists, creates, or deletes branches.

Usage: git branch [branch\_name]

git checkout

Description: Switches branches or restores working tree files.

Usage: git checkout <branch\_name>

git switch

Description: Switches branches (newer and more intuitive than checkout).

Usage: git switch <branch\_name>

git merge

Description: Joins two or more development histories together.

Usage: git merge <branch\_name>

git rebase

Description: Reapplies commits on top of another base tip.

Usage: git rebase <base\_branch>

5. Remote Repositories

git remote

Description: Manages set of tracked repositories.

Usage: git remote [command] [remote\_name]

git fetch

Description: Downloads objects and refs from another repository.

Usage: git fetch <remote\_name>

git pull

Description: Fetches from and integrates with another repository or a local branch.

Usage: git pull <remote\_name> <branch\_name>

git push

Description: Updates remote refs along with associated objects.

Usage: git push <remote\_name> <branch\_name>

6. Inspection and Comparison

git log

Description: Shows commit logs.

Usage: git log [options] [<revision range>] [--] [<path>...]

git show

Description: Shows various types of objects.

Usage: git show <object>

git blame

Description: Shows what revision and author last modified each line of a file.

Usage: git blame <file>

git tag

Description: Creates, lists, deletes, or verifies a tag object signed with GPG.

Usage: git tag [options] [<tagname>] [<commit>]

Example: git tag -a v1.0 -m "Version 1.0"

7. Stashing and Cleaning

git stash

Description: Stashes changes in a dirty working directory away.

Usage: git stash [push | pop | list | drop]

git clean

Description: Removes untracked files from the working directory.

Usage: git clean -f [options]

8. Advanced Commands

git cherry-pick

Description: Applies the changes introduced by some existing commits.

Usage: git cherry-pick <commit\_hash>

git bisect

Description: Uses binary search to find the commit that introduced a bug.

Usage: git bisect [start | bad | good | reset | visualize | replay]

git submodule

Description: Initializes, updates, or inspects submodules.

Usage: git submodule [add | update | init | status | summary]

git archive

Description: Creates an archive of files from a named tree.

Usage: git archive [options] <tree-ish>

9. Help and Documentation

git help

Description: Displays help information about Git.

Usage: git help <command>

git --version

Description: Shows the Git version.

Usage: git --version