

# Git Commands

## SETUP AND CONFIGURATION

1. git: git is a distributed version control system for code management.  
Options: -v, -h, -P, -p  
Usage: git add [file names]  
git clone [git repository URL]
2. config: Helps in setting up the repository and global options.  
Options: -replace-all, -get, -add  
Usage: git config --global user.name [username]  
git config --list
3. help: Provides help information about Git.  
Options: -a, -c, -g  
Usage: git help --all  
git status --help

## GETTING AND CREATING PROJECTS

1. init: Initialize an empty git repository or reinitialize an existing one.  
Options: -q, --bare  
Usage: git init
2. clone: Get the remote repository into the directory  
Options: -l, -s  
Usage: git clone [git repository URL]

## BASIC SNAPSHOTTING

1. add: To stage changes.  
Options: -f, -v  
Usage: git add [file name], . [all changes]
2. status: Know the changes between commit, commits, working tree, etc.  
Options: -s, -v, --long, -b  
Usage: git status
3. diff: Display differences between commits, the working tree, or branches.

Options: --color, --[commit1] [commit2], --cached

Usage: git diff, git diff [commit1] [commit2], git diff --color

4. commit: Record changes to the repository.  
Options: -m, -a, -v  
Usage: git commit -m "[commit message]", git commit -am "[commit message]"
5. reset: Reset current HEAD to the specified state.  
Options: --soft, --mixed, --hard  
Usage: git reset --soft [commit], git reset --hard HEAD^

## BRANCHING AND MERGING

1. branch
2. checkout
3. merge
4. log
5. stash
6. worktree

## SHARING AND UPDATING

1. fetch
2. pull
3. push
4. remote

## INSPECTION AND COMPARISON

1. show: shows one or more things [commits, tags, etc]  
Options: --format=[oneline — short — medium — full, --pretty]  
Usage: git show --oneline
2. log: provide commit info  
Options: --source, --full-diff  
Usage: git log

## **PATCHING**

1. apply
2. cherry-pick
3. rebase
4. revert

## **PLUMBING COMMANDS**

1. commit-tree
2. show-ref
3. update-index
4. revert

## **DEBUGGING**

1. grep: Find matching pattern  
Options: -a, -i  
Usage: git grep -i [text]

## **GUIDES**

1. gitignore: Intentionally untrack some files  
Usage: \*.exe [.gitignore]

## **EMAIL**

1. request-pull: Get pending changes summary.  
Options: -p  
Usage: git request-pull [version number] [URL] [branch name]

## **EXTERNAL SYSTEMS**

1. svn: Operate between Subversion repository and git.  
Options: -s, --no-metadata, --parent  
Usage: git svn rebase

## **ADMINISTRATION**

1. clean
2. filter-branch
3. archive
4. bundle

## **SERVER ADMIN**

1. daemon: A git repository server.  
Options: --export-all, --base-path  
Usage: git daemon --export-all --base-path=.
2. update-server-info: To help dumb server update auxiliary info file.  
Options: -f  
Usage: git update-server-info