**Basic Commands**

**1. git init**

- Initializes a new Git repository in the current directory.

- Usage: git init

**2. git clone**

- Creates a copy of an existing repository.

- Usage: git clone <repository\_url>

**3. git status**

- Shows the working directory status and staging area.

- Usage: git status

**4. git add**

- Adds changes in the working directory to the staging area.

- Usage: git add <file> or git add . to add all changes

**5. git commit**

- Records changes to the repository with a descriptive message.

- Usage: git commit -m "Commit message"

**6. git push**

- Uploads local repository content to a remote repository.

- Usage: git push <remote> <branch>

**7. git pull**

- Fetches and integrates changes from a remote repository to the local repository.

- Usage: git pull <remote> <branch>

**8. git fetch**

- Downloads objects and refs from another repository.

- Usage: git fetch <remote>

**9. git merge**

- Merges changes from one branch into another.

- Usage: git merge <branch>

Branching and Tagging

**1. git branch**

- Lists, creates, or deletes branches.

- Usage: git branch to list, git branch <branch\_name> to create, git branch -d <branch\_name> to delete

**2. git checkout**

- Switches branches or restores working tree files.

- Usage: git checkout <branch\_name> or git checkout -b <new\_branch> to create and switch to a new branch

**3. git switch**

- Switches branches (simpler alternative to git checkout for branches).

- Usage: git switch <branch\_name> or git switch -c <new\_branch> to create and switch

**4. git tag**

- Creates, lists, or deletes tags.

- Usage: git tag to list, git tag <tag\_name> to create, git tag -d <tag\_name> to delete

History and Inspection

**1. git log**

- Shows commit logs.

- Usage: git log

**2. git show**

- Shows various types of objects.

- Usage: git show <object>

**3. git diff**

- Shows changes between commits, commit and working tree, etc.

- Usage: git diff

**4. git blame**

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

- Usage: git blame <file>

Undoing Changes

**1. git reset**

- Resets current HEAD to a specified state.

- Usage: git reset <commit>

**2. git revert**

- Reverts a previous commit by creating a new commit.

- Usage: git revert <commit>

**3. git clean**

- Removes untracked files from the working directory.

- Usage: git clean -f

Remote Repositories

**1. git remote**

- Manages set of tracked repositories.

- Usage: git remote to list, git remote add <name> <url> to add, git remote remove <name> to remove

**2. git remote -v**

- Shows URLs that Git has stored for the repository.

- Usage: git remote -v

Stashing

**1. git stash**

- Stashes changes in the working directory.

- Usage: git stash to stash, git stash apply to reapply, git stash pop to reapply and remove from stash

Configuration

**1. git config**

- Gets and sets repository or global options.

- Usage: git config --global user.name "Your Name" to set global username, git config --list to list all configurations

Aliases

**1. git alias**

- Creates a shortcut for a Git command.

- Usage: git config --global alias.<shortcut> <command>

Useful Options

**1. -m**

- Specifies a message (used with git commit ).

- Usage: git commit -m "Your message"

**2. -b**

- Creates a new branch and switches to it (used with git checkout ).

- Usage: git checkout -b <new\_branch>

**3. -d**

- Deletes a branch or a tag.

- Usage: git branch -d <branch\_name> , git tag -d <tag\_name>

**4. -f**

- Forces an action (like git push -f to force push).