

Most used Git Commands

1. Creating and Cloning Repositories

- **Initialize a New Repository**
 - `git init`: Initialize a new Git repository in the current directory.
- **Clone a Repository**
 - `git clone <repository-url>`: Create a local copy of a remote repository.

2. Checking the Status

- **View the Status**
 - `git status`: Show the working directory and staging area status.
- **Show Commit History**
 - `git log`: View the commit history.
- **View Changes**
 - `git diff`: Show changes between working directory and the index (staging area).

3. Staging and Committing Changes

- **Add Files to Staging Area**
 - `git add <file>`: Stage a specific file.
 - `git add .`: Stage all changes in the current directory.
- **Commit Changes**
 - `git commit -m "message"`: Commit staged changes with a message.
 - `git commit -a -m "message"`: Commit all changes (including untracked files) with a message.

4. Branching and Merging

- **Create a Branch**
 - `git branch <branch-name>`: Create a new branch.
- **Switch Branches**

- git checkout <branch-name>: Switch to a different branch.
- git switch <branch-name>: Switch branches (modern alternative to checkout).
- **Create and Switch to a New Branch**
 - git checkout -b <branch-name>: Create and switch to a new branch.
 - git switch -c <branch-name>: Create and switch to a new branch (modern alternative).
- **Merge Branches**
 - git merge <branch-name>: Merge a branch into the current branch.
- **Rebase Branches**
 - git rebase <branch-name>: Reapply commits on top of another base branch.

5. Remote Repositories

- **View Remote Repositories**
 - git remote -v: List remote repositories.
- **Add a Remote Repository**
 - git remote add <name> <url>: Add a new remote repository.
- **Fetch Changes from Remote**
 - git fetch <remote>: Fetch changes from a remote repository.
- **Pull Changes from Remote**
 - git pull <remote> <branch>: Fetch and merge changes from a remote repository.
- **Push Changes to Remote**
 - git push <remote> <branch>: Push changes to a remote repository.

6. Managing Changes

- **Undo Changes in Working Directory**
 - git restore <file>: Discard changes in the working directory.
- **Unstage Files**

- `git reset <file>`: Unstage a file from the staging area.
- **Reset to a Previous Commit**
 - `git reset --hard <commit>`: Reset the working directory and index to a specific commit.
- **Revert a Commit**
 - `git revert <commit>`: Create a new commit that undoes changes from a specified commit.

7. Viewing and Comparing

- **View a Specific Commit**
 - `git show <commit>`: Show details of a specific commit.
- **Compare Branches**
 - `git diff <branch1>..<branch2>`: Compare differences between two branches.

8. Tags

- **Create a Tag**
 - `git tag <tag-name>`: Create a new tag.
- **Push Tags to Remote**
 - `git push <remote> <tag-name>`: Push a tag to a remote repository.
- **List Tags**
 - `git tag`: List all tags.

9. Miscellaneous

- **Stash Changes**
 - `git stash`: Save changes temporarily and revert to the clean working directory.
 - `git stash pop`: Apply stashed changes and remove them from the stash list.
- **Show Configuration**
 - `git config --list`: List all Git configuration settings.
- **Edit Configuration**
 - `git config <key> <value>`: Set or change a Git configuration setting.

