Basic Git Commands

Git is a powerful version control system that helps you manage changes to your codebase. Here is a one-page guide of the most commonly used Git commands you need to manage your projects effectively..

1. Setting Up Git

Configure user information:
git config --global user.name "Your Name"
git config --global user.email "your_email@example.com"

2. Creating and Cloning Repositories

Initialize a new repository: git init
Clone an existing repository: git clone https://github.com/username/repos.git

3. Checking the Status of Your Files

Check the status of your files: git status

4. Staging and Committing Changes

Stage a file for commit: git add <file>
Stage all changed files: git add .
Commit staged changes: git commit -m "Your commit message"

5. Viewing the Commit History

View commit history: git log
View commit history with Graph
git log --graph --oneline

6. Branching and Merging

Create a new branch: git branch <branch-name>

Switch to a branch: git checkout <branch-name>

Create and switch to a new branch: git checkout -b <branch-name>

Merge a branch into the current branch: git merge <branch-name>

7. Pushing and Pulling Changes
Push changes to a remote repository: git push origin <branch-name>

Pull changes from a remote repository: git pull origin <branch-name>

8. Tagging Releases
Create a tag: git tag <tag-name>

Push tag to remote repository: git push origin <tag-name>

9. Undoing Changes
Undo staged changes: git reset <file>

Undo unstaged changes: git checkout -- <file>

Undo last commit: git reset HEAD~ --soft

Undo last commit and discard changes: git reset HEAD~ --hard

10. Remote Repositories
Add a remote repository: git remote add origin <repository-url>

View remote repositories: git remote -v

Remove a remote repository: git remote remove origin

11. Stashing Changes
Stash uncommitted changes: git stash

List stashed changes: git stash list

Apply the most recent stash: git stash apply

Pop the most recent stash: git stash pop

12. Blaming Changes
Blame changes in a file: git blame <file>

13. Diffing Changes
Show differences between working directory and staged changes: git diff

Show differences between staged changes and last commit: git diff --staged

14. Cleaning Up
Remove untracked Files: git clean -fd