TOP 20 GIT COMMANDS

git init

The git init command is used to initialize a new Git repository. When you run this command in a directory, it sets up a new repository, creating the necessary data structures and files that Git uses to store information about the project.
Syntax – git init

git config

The git config command is used to configure Git settings either at a global, local (repository-specific), or system level. Here are some common use cases:
Syntax: - git config --global user.name "Your Name"
git config --global user.email your.email@example.com
check config details : -
git config user.name
git config user.email

git add

The git add command is used to stage changes for the next commit in Git. It tells Git that you want to include updates to a particular file (or files) in the next commit. Here are some common use cases:
Syntax: - git add <file_name>

git commit

The git commit command is used to record changes to the local repository. It takes the changes that you have staged using git add and creates a new commit containing those changes. Each commit in Git represents a snapshot of the project at a specific point in time.
Syntax: - git commit -m "Your commit message"

git log

The git log command is used to display the commit history of a Git repository. It shows a chronological list of commits, including the commit hash, author information, date, and commit message.
Syntax: - git log

git branch

The git branch command is used in Git to manage branches in your repository. Here are some common use cases:
List Local Branches: git branch

git remote

The git remote command is used to manage remote repositories that are connected to your local Git repository. Here are some common use cases:
Syntax :- git remote add <remote_name> <repository_url>

git clone

The git clone command is used to create a copy of a remote Git repository on your local machine. This is typically the first command you run when you want to start working with an existing Git project. Here's how you can use it:
git clone <repository_url>

Create new branch

git branch <branch_name>

git checkout

git checkout – switch to different branch
git checkout <branch_name>

Delete a Branch:

git branch -d <branch_name>

git push

The git push command is used to upload local branch commits to a remote repository. This command is essential when you want to share your local changes with others or update a remote repository with your latest commits.
Syntax:- git push <remote_name> <branch_name>

git pull

The git pull command is used to fetch changes from a remote repository and merge them into the current branch.
Syntax:- git pull <remote_name> <branch_name>

git rebase

git rebase is a powerful Git command used to modify the commit history of a Git repository.
Steps :- git checkout -b feature_branch main
Edit files
git commit -a -m "Adds new feature"
git rebase <base>

git merge

The git merge command is used to integrate changes from one branch into another. When you merge branches, Git combines the changes made on separate branches, creating a new commit that represents the combination of those changes.
Eg:-
Switch to the master branch
git checkout master
Merge changes from a feature branch into master
git merge feature-branch

git reflog :

It is a command in Git that displays the reference logs. It provides a history of changes to the tips of branches and other references in the repository. The reflog is useful for recovering lost commits, branches, or other changes that may have been garbage collected or otherwise lost from view in the normal Git history.
Syntax :- git reflog

git status

The git status command is used to display the status of changes as untracked, modified, or staged in your Git working directory. It provides a summary of the current state of your repository and helps you understand what is happening in terms of changes to your files.
Syntax :- git status

git clean

The git clean command is used to remove untracked files from your working directory. Untracked files are files that are not currently under version control but exist in your working directory
Syntax : - git clean -fd
-f stands for force, and it's required to perform the clean operation.
-d removes untracked directories in addition to untracked files.

git revert

The git revert command is used to create a new commit that undoes the changes made in a previous commit. This is useful when you want to reverse the effects of a particular commit without altering the commit history.
Syntax: - git revert <commit_hash>

git reset

The git reset command is used to reset the current branch to a specific state. It can be a powerful and versatile command, but it should be used with caution, especially when manipulating commit history.
Syntax: - git reset --hard <commit_hash>

For reference
Youtube : -