To initialize git in a repository

* Git init

To add files to the staging area

* git add <filename>

To add all files to the staging area

* git add .

To commit files to the local repo

* git commit -m “msg”

To see the remote changes

* git fetch

to pull the changes to the local

* git pull
* git log – to see the list of commits
* git status – view untracked and staged files
* Set global config email
* git config --global user.email [example@gmail.com](mailto:example@gmail.com)
* Set global config username
* git config --global user.name “name”
* Set local config email
* git config --local user.email [example1@gmail.com](mailto:example1@gmail.com)
* Set local config username
* git config --global user.name “name”
* View **local/global** config list
* git config --local --list
* git config --global --list
* Remove local user name and email
* git config --unset user.name
* git config --unset user.email
* Remove local user name and email
* git config --global --unset user.name
* git config --global --unset user.email
* Add files to staging area
* git add <filename>/.
* Undo all files to staging area
* git restore --staged .
* undo specific file staged file to working directory
* git restore --staged <filename>
* Commit files to local repo
* git commit -m “msg”
* undo commit and take files back to staging area
* git reset --soft HEAD~1
* undo commit and take files back to working area file will be deleted
* git reset --hard HEAD~1

git push -

git branch -M main

git push -u origin main

git push --set -upstream origin main

git remote add origin <https://github.com/suryasaranbs/sample.git>

git branch -M main

git push -u origin main

* Undo commit and record the changes
* git revert <commit\_id>
* Undo commit and do not record the changes
* git reset --hard <commit\_id>
* To list the created branches
* git branch
* Creating new branch
* git branch <branch\_name>
* Switch to branch
* git checkout <branch\_name> / git checkout -b <branchname> (will create and switch to the)
* Renaming branch
* git branch -m feature-branch new-feature-branch
* Deleting branch
* git branch -d <branchname>/ git branch -D <branchname> (force delete)
* Merge branch

Checkout to the branch u want to merge changes

* git merge <branchname>

Merge conflict

Rebase branch

Set remote origin

Git remote origin <url>

git remote set-url origin <new\_url>

git push -u origin <branch>

Push to remote origin

Set upstream