git diff <file\_name> (or leave it blank) 🡪 Shows the differences between the working directory and the staging area (or the last commit if there are no staged changes).

git –staged diff <file\_name> 🡪 This flag tells Git to show the differences between the staging area and the last commit. It shows what is ready to be committed.

git reset <commit\_hash> 🡪

Moves the HEAD to the specified commit and **unstages the changes** but keeps them in the working directory.

git reset <commit\_hash> --hard 🡪 Go back to that specific commit and remove all the changes.

When you first create a new Git repository (using git init) and haven't made any commits yet, there is no branch by default. Git does not automatically create a branch until you make your first commit.

Before the first commit, your repository exists, but it doesn't have a branch yet. Once you make your first commit (using git commit -m "Initial commit"), Git will create a default branch, typically called main (or master, depending on the version of Git).

If you run git branch before the first commit, it will show no branches, because at that point, Git hasn't created a branch yet.



Git reset --mixed HEAD~1 or commit hash 🡪Basically remove the commit and unstage the changes

Only unstages the stahged files(It’s obvious)

A black background with white text

AI-generated content may be incorrect.



Git reset --soft HEAD~1 or commit hash 🡪 remove the commit but let the changes it staging area.

A black background with white text

AI-generated content may be incorrect.



Here it is discarding the changes which are in the staging area/unstaged (every thing) but it can be use to completely remove the commit also