Introduction to GIT commits:

Repository records all the files in the directory.The commit restores the versions of these repositories.Git maintains history of these commits. So every changed commit has an ancestor commit.

Command for commit- { GIT COMMIT }

GIT branches

Branches are lightweight so it easy to store and open them easily. They act as pointers to different commits and it locally divides your work.

Command for branch { GIT BRANCH BUGFIX }

GIT Rebase

Other way of combining work between branches is rebasing.Advantage of rebasing is that it can be used to make a nice linear sequence of commits. The commit log / history of the repository will be a lot cleaner if only rebasing is allowed.

Command for GIT Rebase { -i HEAD ~ 2 }

Head

It is the symbolic name for the currently checked out commit and it always points to the most recent commit which is reflected in the working tree. Removing it just means attaching it to a commit instead of a branch