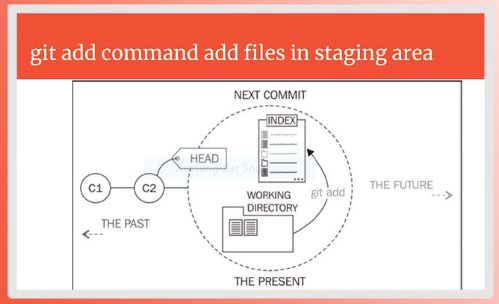
Head :

Head is basically the current branch or u can say the current version of the working directory.

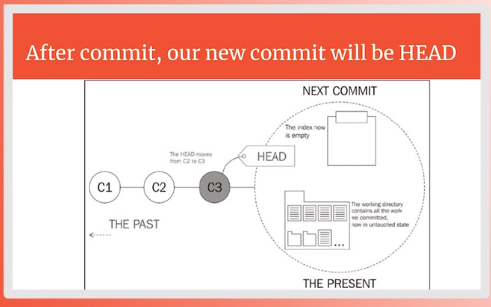
In which branch you are present in it is referred as head .

Past : last commit we did represent our head .

Present : for next commit(working directory) the last commit would be its head



Remember when u commit a file so this file would then become the head as shown :



Note : commits name are not c1 and c2 it is just for your understanding commit name is always a 40 digit hash

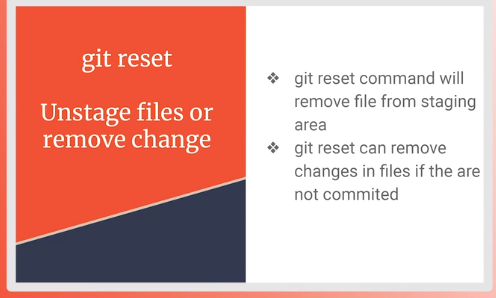
COMMIT HASH : commit hash is 40 character unique identifier of your every commit

For ex :

Two people were working on their local machines and commit their work on their branches when they would merge their work we would be unable to identify because it is possible that both people commit has the same number that’s why we use hash as a commit identifier ….

UNSTAGE FILE OR REMOVE CHANGE :

Git reset : if u had added 3 files through git add command in the staging area but u want to commit only 1 and want the other 2 files to come back from staging are here u use **git reset (file name)** .

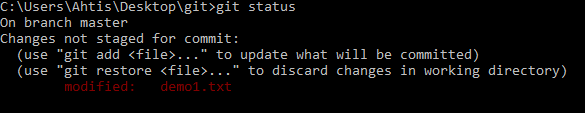


**Git reset –hard :** it allows u to un stage a file as well as to delete the changes u had made in a file also .

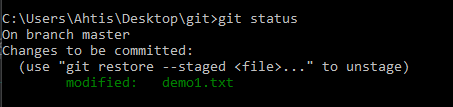
For example : if u had made a change and written “hello world” in a file and staged it using git add command so **git reset – hard** would un stage it as well as delete the **hello world** changes also.

Note : **git reset –hard** remove changes if the file is staged or not staged it would remove the changes .

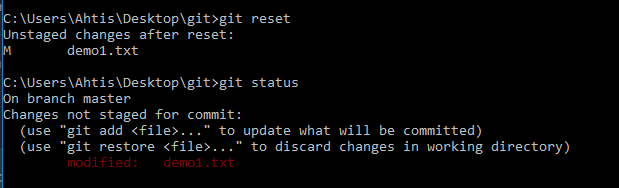
***Modified but unstage :***



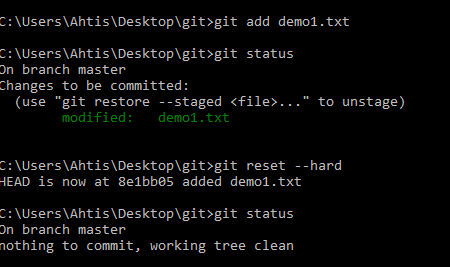
***Staged :***



***Git reset :***

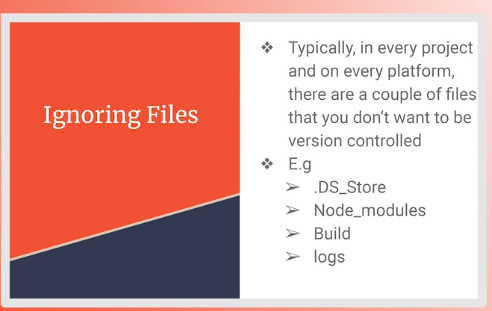
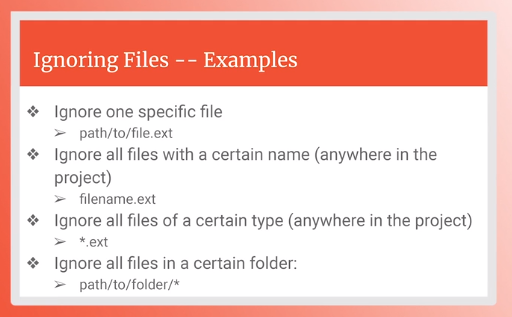
******

***Git reset –hard :***



***IGNORING FILES :***

Git ignore enables u to ignore the files that u don’t want to be the part of a git history and u don’t want git to track these files these files are basically those files in which changes are being made continuously so its hard to commit them each time that’s why we ignore this type of files .



To ignore files we make a folder and add the names of the files as specified in the above picture and these files does not show up when we see git status and are not tracked by git .

We use name.ext command most commonly for ignoring a file . for ex :

First.txt ----🡪 for txt files

\*.xlsx -----🡪 for excel files

/Name of folder/ ---🡪 for ignoring the complete folder like build .