

GIT CLASS 2

To untrack a file : `git rm --cached filename`

`git log --follow --all filename` : used to get all commits for a file

GIT CONFIG : Used to add our name and email for git history

`git config user.name "username"` : used to add our name

`git config user.email "email"` : used to add email

GIT IGNORE:

a ".gitignore" file is used in Git to tell Git which files and directories to ignore when tracking changes in a project.

Let's say you're developing a software project and using Git to manage your code changes. Sometimes, there are files or directories in your project that you don't want Git to keep track of, such as temporary files, compiled code, or sensitive information like passwords. Instead of manually excluding these files every time you commit changes, you can list them in a ".gitignore" file.

GIT RESET: This command is used to delete latest commits

`git reset --hard HEAD~1` : used to delete latest commit along with changes

`git reset --soft HEAD~1` : used to delete only latest commit

`git update-ref -d HEAD`: used to delete all commits at a time (only commits)

GIT BRANCHES:

- A branch represents an independent line of development.
- The git branch command lets you create, list, rename, and delete branches.
- The default branch name in Git is master.
- allows you to work on different features or changes to your code independently, without affecting the main or other branches.
- It's a way to organise and manage your code changes, making it easier to collaborate and maintain your project.

COMMANDS:

<code>git branch</code>	used to see the list of branches
<code>git branch branch-name</code>	to create a branch
<code>git checkout branch-name</code>	to switch one branch to another
<code>git checkout -b branch-name</code>	used to create and switch a branch at a time
<code>git branch -m old-branch new-branch</code>	used to rename a branch
<code>git branch -d branch-name</code>	to delete a branch
<code>git branch -D branch-name</code>	to delete a branch forcefully

The -d option will delete the branch only if it has already been pushed and merged with the remote branch. Use -D instead if you want to force the branch to be deleted, even if it hasn't been pushed or merged yet. The branch is now deleted locally.

Now all the things you have done is on your local system.

GIT MERGE:

Git merge is a command used in the Git version control system to combine changes from one branch.

To merge: `git merge branch_name`

GIT CHERRY-PICK:

Git cherry-pick is a command in Git that allows you to take a specific commit from one branch and apply it to another branch. It's like picking a cherry (commit) from one branch and adding it to another branch, allowing you to selectively copy individual commits without merging the entire branch.

Command: `git cherry-pick commit_id`

GIT REFLOG:

Git reflog is a command that is used to see the commit history.