**SHORT FORMS**

-a (short form of all)

-m (message)

-mv (move)

Git init 🡪 initialize git repository.

touch .gitignore 🡪 create .gitignore.

. Usage: git remote add [variable name] [Remote Server Link]

This command is used to connect your local repository to the remote server.

Git Remote Command - Git Commands - Edureka

. git remote remove origin 🡪 remove remote variable.

To **keep the changes** from the commit you want to undo

$ git reset --soft HEAD^

Git Snapshot -> State of The file.

Git add (command updates the current content of the working tree to the staging area)

Git commit –m “Commit Message”.(Used to commit/saved changes)

Git commit (just writing this command we can use default editor to write commits if we want to add long message.)

Git Congiguration:

Has seetings for name, email, default editor, line ending

Three Levels of configuration:

System 🡪 All users

Global 🡪 All repositories of current users

Local 🡪 Current repository

git config -- global user.name “Name Surname”

git config -– global user.email “xyz@mail.com”

git config -- global code.editor “code --wait"

wait flag 🡪 waits for editor to close the current file which shows .gitignore .

End of Line in Windows

‘\n’ 🡪 line feed

‘\r’ 🡪 carriage return

git config --global core.autocrlf true (for windows)

git config --global core.autocrlf input (for mac/linux)

git ls-files (files in staging area)

**Remove file from working and staging area.**

Method 1:

rm file.txt (removes from local folder)

git add file.txt (updates changes in staging area. Means removing file2 from staging area.)

git commit –m “file removed”

Method 2:

git rm file.txt (removes file from local and staging area.)

git commit- m “file removed”

**Renaming the file.**

git mv file1.txt file2.txt (renames file1.txt to file2.txt)

**Ignoring Files**

\*.log (ignore all files with .log extension)

Logs/ (ignore Logs/ folder)

**Remove File from staging area**

git rm - - cached –r bin/ (removes bin/ folder recursively)

git rm - - cached filename.extension

**To know about changes done in files in staging area (compares files in staging area with previous commit).**

git diff - - staged

**To compare the file in local folder vs staging area.**

Git diff

**For better readability of the code we must use an editor when we use diff command.**

setting vscode as editor for diff command.

git config --global diff.tool vscode

git config --global difftool.vscode.cmd "code --wait --diff $LOCAL $REMOTE"

(Remember) code – Add code to path variable it must contain your default editor location.

**Viewing Commit History**

Git log

Git log -p (with diff)

Git log - - stat( Information of commit in short summary)

Git log - - pretty=oneline (online message of commit)

**To know the changes in commits we have done**

Method 1: git show <unique identifier of commit>

Method 2:

HEAD will reference to last commit.

Git show HEAD~(number of steps we want to go backward from last commit)

e.g git show HEAD~1 (shows second last commit).

**Unstaging files in Git**

Git restore –staged filename.ext (restore file from staging area.)

**Removing Git Repository**

rm –rf .git

**Untrack Files in Git**

git rm –cached “filename.ext”

**Restore the changes in the file w.r.t last commit**

git checkout – filename.ext