**Git and github seris Anisul Islam**

**Step1 : Working disrectory/workspace**

**Where we make a folder and file.**

**Mkdir notes -> creat folder**

**Cd notes -> change folder**

**ls -> show file without hidden files**

**ls -a -> view all file with hiden files**

**git init -> creating / initializing git repository**

**touch day1.txt -> create file**

**git status -> display the state of the working directory and staging area.**

**Step2: Stagging area / index**

**Git add <filename> -> move to stagging area**

**Git add -A -> move to all changed file in stagging area all directory and subdirectry**

**Git add . -> stage all changed file in directory but not subdirectry.**

**Git restore <filename> -> to discard changes in working directory**

**Git diff -> checking the difference**

**rm –cached <filename> -> to unstage the file**

**Step3: Loacal repository**

**For moving staging to local repository we can use following command**

**Git commit -m “message here”**

**Message should be clear and understandable**

**Staging and commiting directly: git commit -am “message here”**

**Git log -> to see the commit history**

**Git log --oneline -> to see the commit in oneline**

**Direct: git add . && git commit -m “message”**

**Uncommit:**

**If all you want to do is undo the act of commiting , leaving everything else intact, use:**

**Git reset --soft HEAD^**

**If all you want to do is undo the act of commiting and also removing from the stagging area :**

**Git reset HEAD^**

**And if you actually want to completely undo it, throwing away all uncommitted changes reseting everything to the previous commit (as the original question asked):**

**Git reset --hard HEAD^**

**Deleting number of commits**

**Git reset -soft HEAD^ ~2**

**Git reset -soft HEAD^ ~3**

**HEAD and UNDO:**

**Git show commit id / git show / git show HEAD**

**OR git show HEAD~ Number**

**Git revert/git checkout/git reset/git clean/ git rm**

**Git checkout <commit ID>**

**.**

**.giigno**

**Touch .gitignore**

**Mark down syntax**

**Connecting Local and remote repository.**

**Check remote connection : git remote**

**Git remote -v : show the remote along with the url.**

**Syntax : git remote add name <Remote\_url>**

**Git Branching and merging**

Git is a new and separate branch of the master repository. In a big project we separate the tasks / features and create branch so editing in the new branch not affect the master branch.

**Git branch <name> -> create branch**

**Git branch -> check branch**

**Git checkout <branch name> -> move to branch**

**Git branch -d <name> -> delete branch**

**Git checkout -b “branch name” -> create branch and switch to this branch.**

**Git push -u origin <current branch name>**

**16. branching and merging locally**

**Create a branch**

**Git merge <branch name>**

**Git pull**

**Git push -u origin main**

**18. GitHub issues**

**19. Fast forward merge or 2-way merge**

**20. 3-way merge**