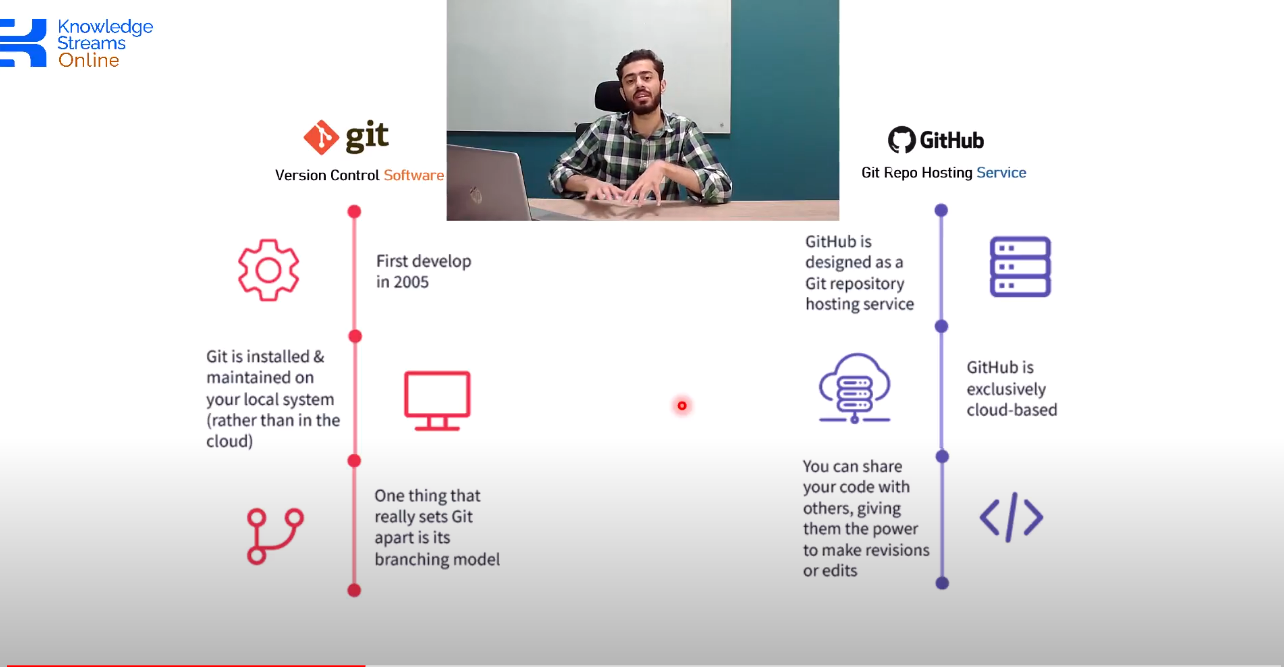
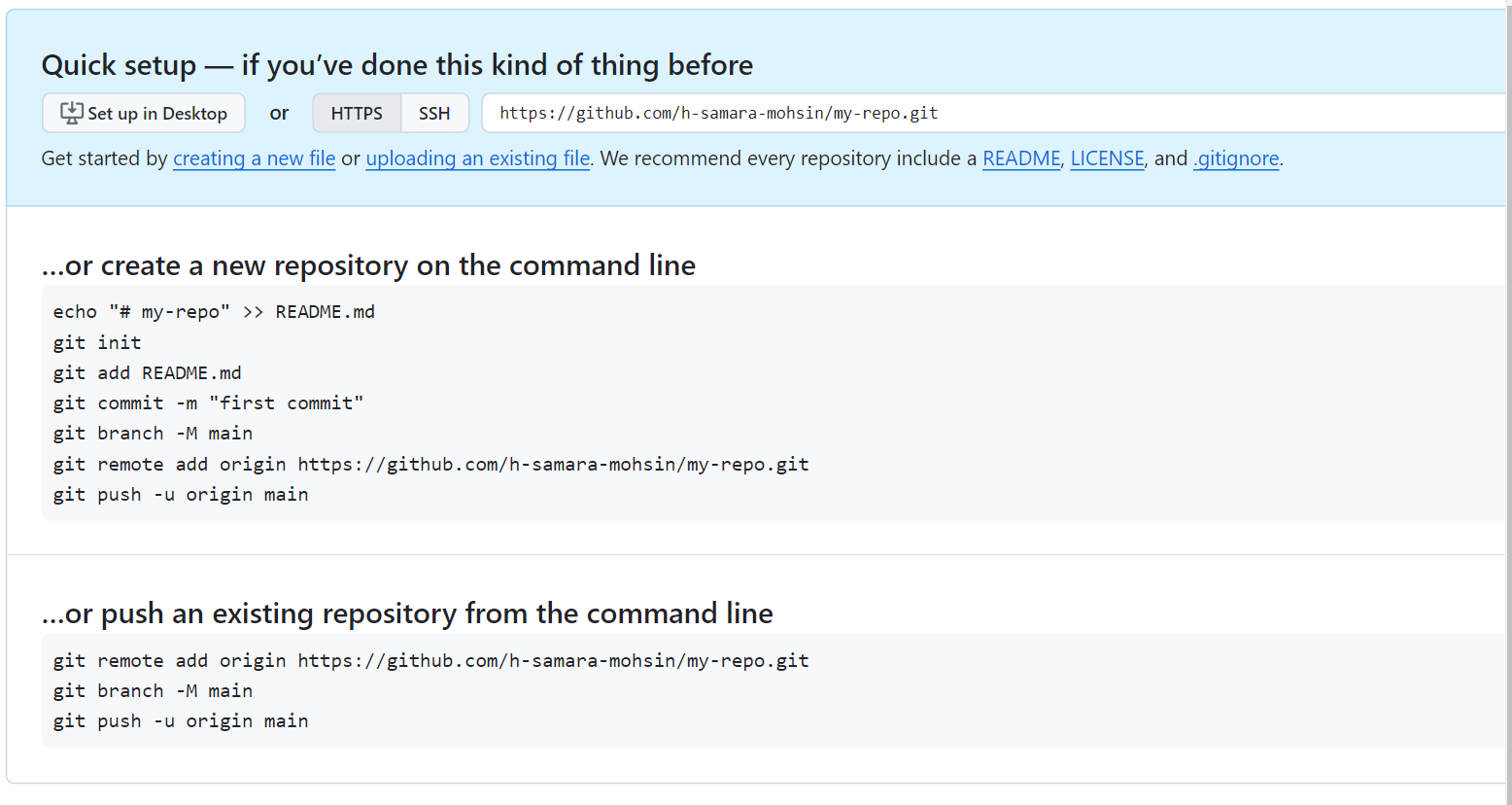
## Git Commands:

## Sequence

1. **git init**
2. **git add .**
3. **git status**
4. **git commit -m “type your msg”** (koi bhi kam kia uska snapshot leke apne pas rakhlia usey commit bolte hain, aur ye sequence me hota ha tabhi hum track kr pate hain k over period of time hamne kia changes kye hain)
5. **git log**: to see all commits written (press “q” to come out of it), every commit has its own unique id
6. **git log - -oneline**: this command will list all commits one in each line(not detailed as prev one , shortly displayed the history in single line) aur currently ma kis commit pe hun wo mujhe aise pata chalega k HEAD jispe ha.
7. .git folder : hidden by default 🡪 contains all commit history if we delete this folder we would not be able to track the changes in future
8. Branch: aik series of commits ka name h branch ; different features pe diff developers kam kr rahe hain
   1. **git branch**: to see current branch jispe ma hun abhi
   2. **git branch nameOfBranch :** to create branch
   3. **git checkout nameOfBranch:** to move to that branch
   4. **git checkout -b nameOfBranch:** combination of above two commands k make this file first and then switch to that
   5. **git merge nameOfBranch :** jis branch me abhi hum hain yani current branch this is destination , and jis branch ka hamne name dia ha uske changes uthke yahan aajayenge
9. **git add main.txt** : main.txt staged area me chali gai h 🡪 jo cheez staging area me ha sirf wohi commit hosakti h ; staging area wo area ha jo local hamare pas changes store krne ki jaga ha
   1. **git add** . : mere saare changes jis bhi file me hue hain usko uthake staging area
10. **untracked: U (**written in front of a file means this is untracked ya tou ye new file h ya hamne git ko mana kia wa tha**) git add .** likhne k baad wo U se update hoke A hogaya means staging area ab commit krsakti hain
11. **Modified (M):** already tracked ha but kuch changes hue hain 🡪 unko staging area me leke jayenge then commit krenge
12. **Github:**
13. **Push : upload the file on github server**
14. **Pull : receive the data from another developer working on same repo from centralized place**





After creation of repo on github.

Two ways:

Create new repo

Or connect already existing : we have already created repo using git init on local machine

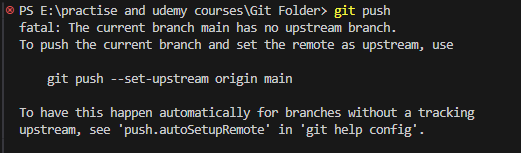
Jb ham git bana rahe hote hain tou by default us brancha ka name kabhi master hota ha kabhi main hota ha tou github ki ye primary convention ha k apni branch ka name main rakho

We have two repos one is local and another is remote **git remote** remote repo k operations k lye use hoti hain **add origin** means aik variable add hoga origin name ka aur jb ma origin ko refer kroonga uska matlab hoga ma is repo ki baat kr raha hun jiske url given h , tou aik tarah se aik variable banadia origin name ka aur usme apni repo ka link rakh dia

Git push means : mera local structure uthake sara ka sara push krdo

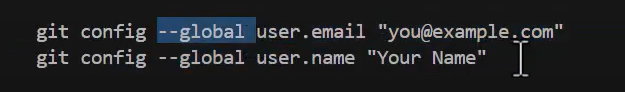
Humne init krlia ha files add krli hain aur commit bhi krdi hain

Ye commands hame master me rehke hi run krni h always because isme hame master branch ka name change krke main krna ha tou make sure to checkout in master first



Pehle hamne repo ka repo se connection banaya tha ab branch ka branch sse banana ha 1 rule follow krna ha k jo name branch ka local me hoga wohi name branch ka remote me hoga. Jb hum pehli dafa push krte hain tou usko remote k upper branch ka upstream nhi milta tou isilye error milta ha aur sath hi uska solution bhi batata ha

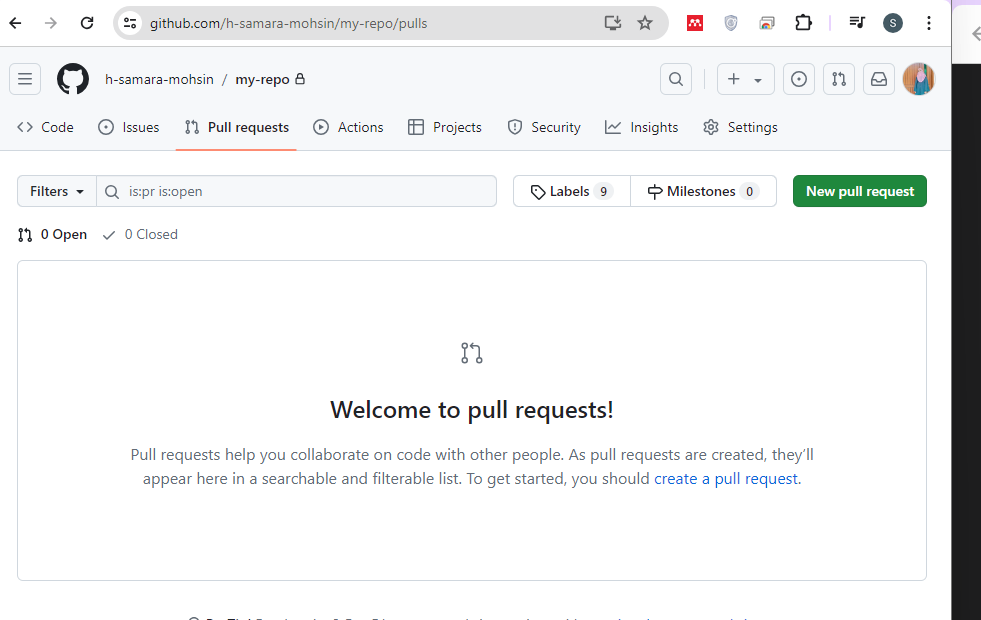
Jaise hi ye upstream wali branch run krenge code push hojayega poora remote pe

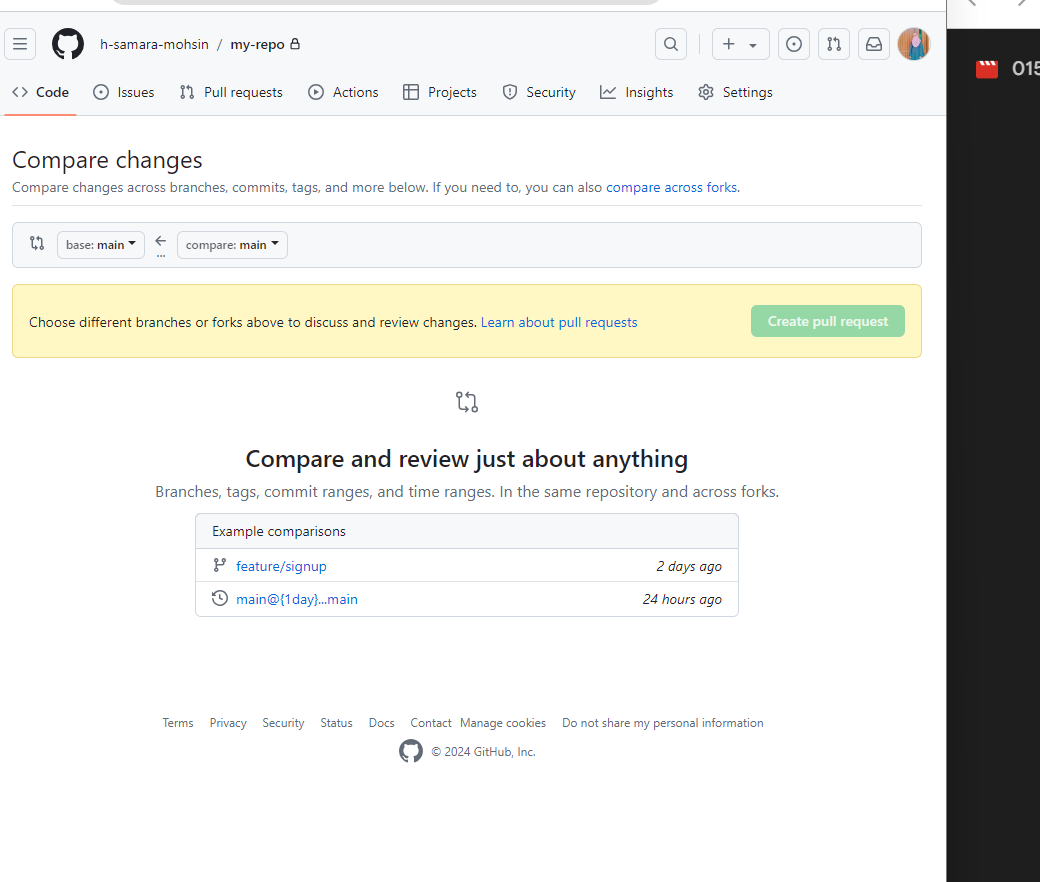
1. 

Agr ham global ki jaga local use krlen tou wo her repo ko initialize krte waqt configure krwayega. Global ka matlab ye h k mere is laptop k ander jahan git install h wahan jaake mera ye name and email store krdo jb bhi koi proj create kroonga tou wohi name aur email retrieve hojayega

1. **Pull and pull request :** they are diff concepts just like java and javascript
   1. **Pull :** aik remote repository se git ka structure uthake meri local machine k upper lata h
   2. **Pull request:** ye git ka feature nhi ha ye github ko use krke hota h aur iska matlab ye h k mera code jo hamari primary branch h usme **p**ull krlo review krke
2. How to generate **PULL Request**

First push all the code before creating pull request





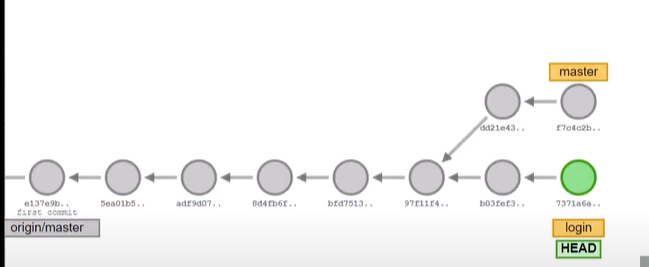
Master me jo code ha wo production level ka ha usme direct merge nhi krsakti apna code baghair PM k approval k, but master ka code tou apni branch me lasakte hain wo tou do able h

So step 1: **git pull** in main branch

Step2 : checkout to your branch and run **git pull(to sync my local work with remote work)**

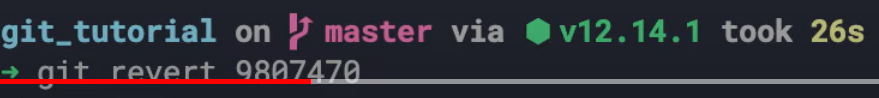
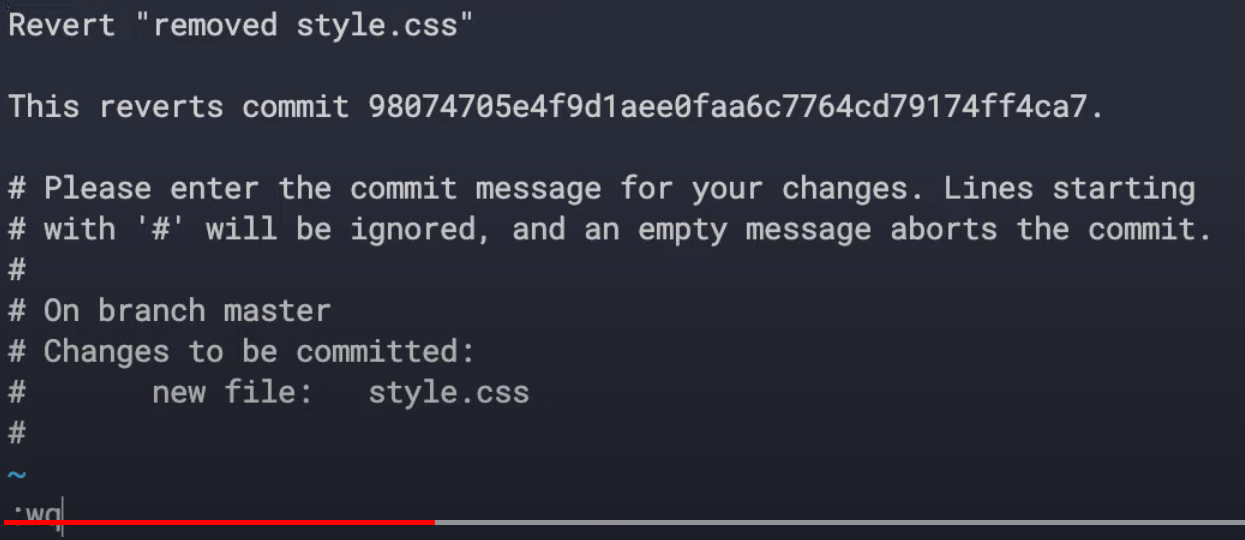
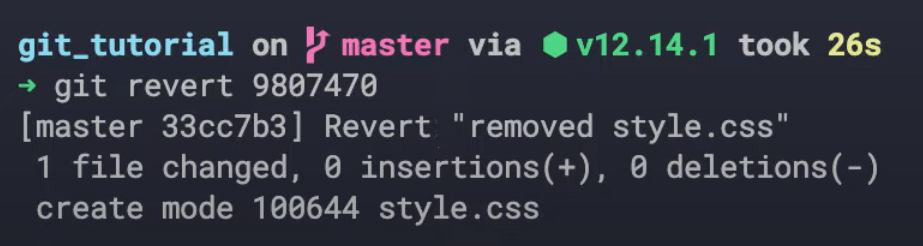
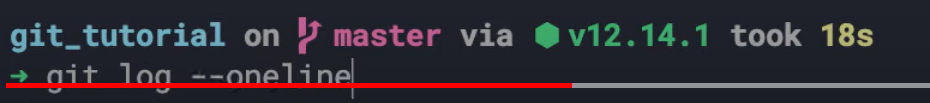
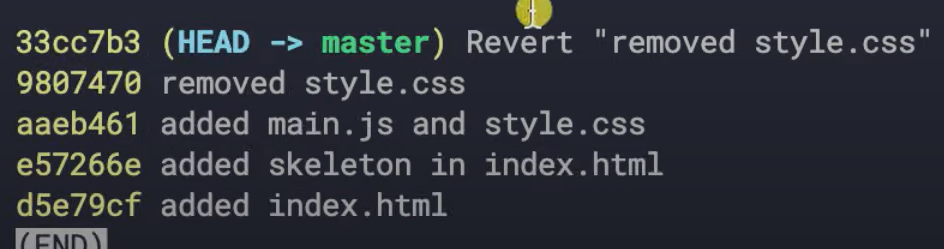
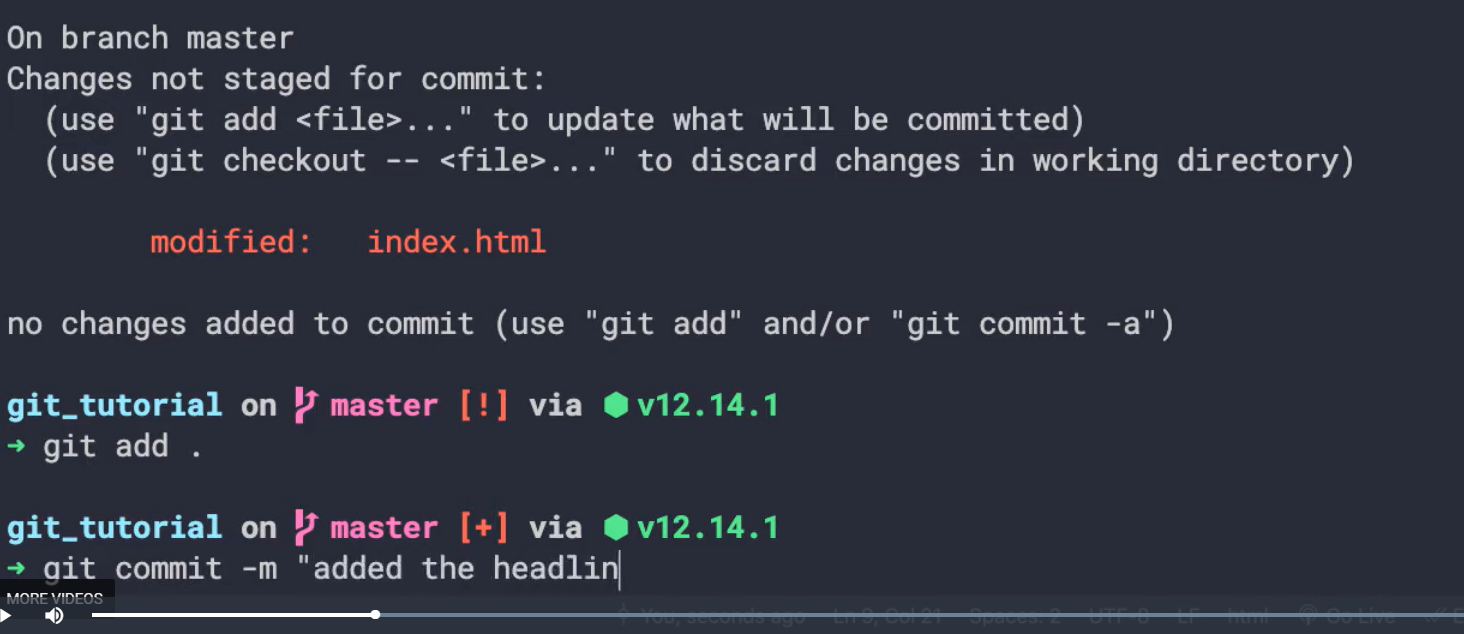
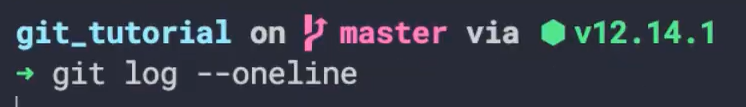
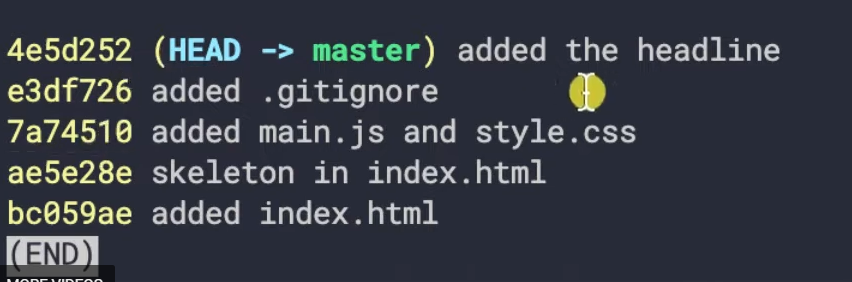
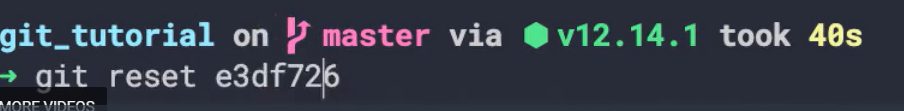
**Step 3: git merge main.txt**

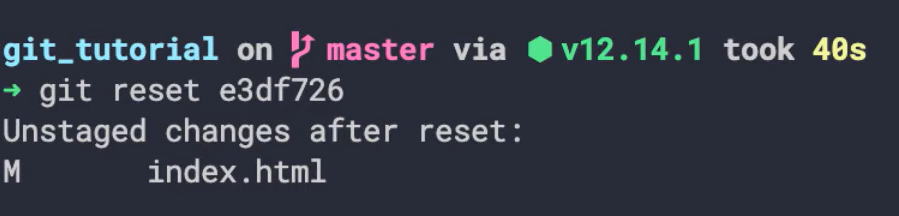
1. **git pull: to pull the latest code**
2. kisi bhi branch ka jo pointer hota ha wo uske latest commet pe hota h , aur head current branch pe hota h
3. **git checkout commetID :** checkout ki command basically aik commet se doosre commet pe travel krti ha , tou head hamara usi commet pe chala jata h
   1. random kisi commit pe move krsakti hain , wahan tk jo code tha usey dekh sakte hain lakin wahan directly commets shuroo nhi krdenen branch bana sakte hain aik new , basically her branch ka aik pointer hota ha jo track krta h
   2. 

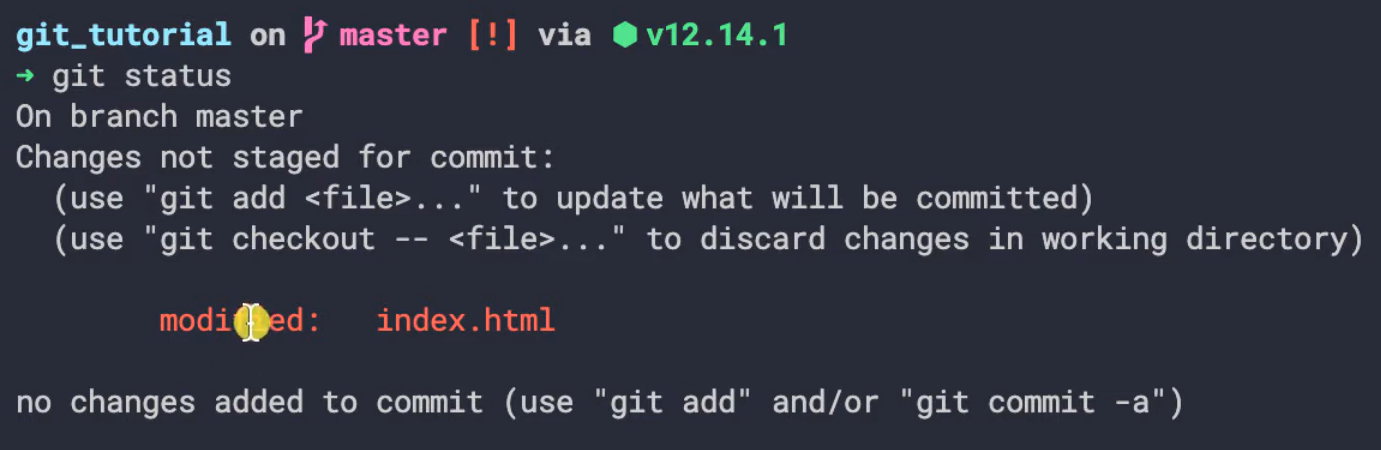


1. To undo the commit:

There are 2 commands:

* 1. Revert
     1. **git revert** **commitID**
     2. **:wq** write and quit
        1. 
        2. 
        3. 
        4. 
        5. 
        6. 
  2. Reset
     1. **git reset** commitID
        1. 
        2. 
        3. 
        4. 



* + - 1. 
    1. **git reset --hard**