**GIT Learning**

Google Links:

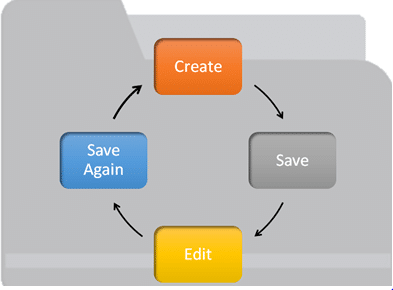
<https://intellipaat.com/blog/tutorial/devops-tutorial/git-tutorial/>

<https://www.tutorialspoint.com/git/index.htm>

**Getting Start With GIT**

**Overview of GIT:**

* Be it a designer, a creator, or a developer, all deal with projects and files every day. Our primary work cycle revolves around creating a file, saving it, editing or making required changes, and saving it again.
* Git is a small yet very efficient version control tool. It helps both programmers and non-programmers keep track of the history of their project files by storing different versions of them.



1. What is GIT & life Cycle:

## What is Git?

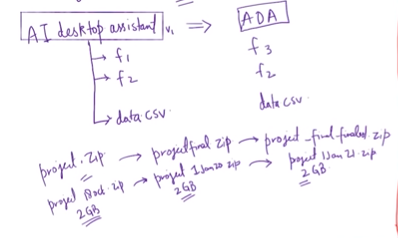
Git helps developers keep track of the history of their code files by storing them in different versions on its own server repository, i.e., GitHub. Git has all the functionality, performance, security, and flexibility that most of the development teams and individual developers need.

## Why Git Version Control

Below are some of the facts that make Git so popular:

* **Works offline**: Git provides users very convenient options such as allowing them to work both online and offline. With other version control systems like SVN or CVS, users need to have access to the Internet to connect to the central repository.
* **Undoes mistakes**: Git allows us to undo our commands in almost every situation. We get to correct the last commit for a minor change, and also we can revert a whole commit for unnecessary changes.
* **Restores the deleted commits**: This feature is very helpful while dealing with large projects when we try out some experimental changes.
* **Provides security**: Git provides protection against secret alteration of any file and helps maintain an authentic content history of the source file.
* **Guarantees performance**: Being a distributed version control system, it has an optimized performance due to its features like committing new changes, branching, merging, comparing past versions of the source file, etc.
* **Offers flexibility**: Git supports different nonlinear development workflows, for both small and large projects.

Eg:

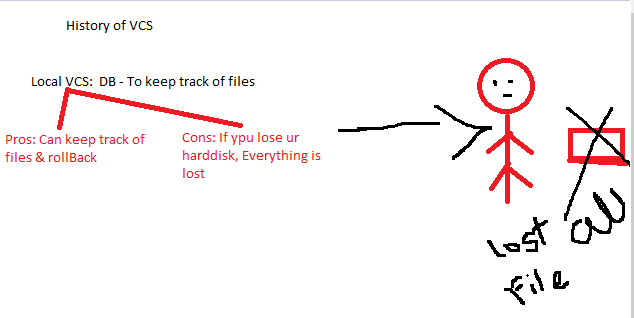


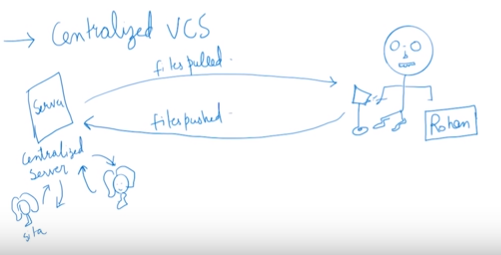
Easily Recover File

Who Introduced an issue and When

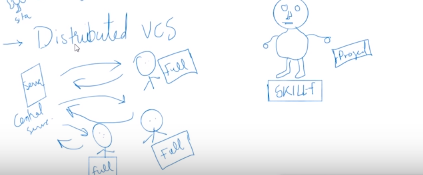
Roll back to previously working state

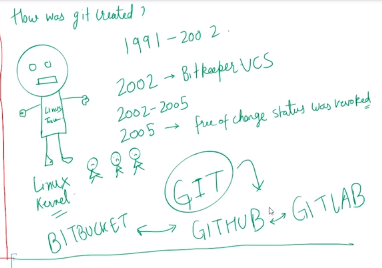
History of Version Control System:

Only one person has every files and collaboration was difficult



Can lose data from server

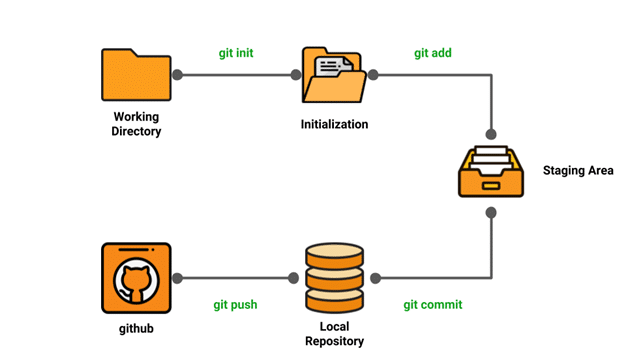




Pull -> get all project/ history at Pulling time->Everyone has his/her own files complete backup -> if lost-> pull from serve-> if server lost-> can setup server again and get all files 😊

## How does Git work/Git Life Cycle

* **Local working directory**: The first stage of a Git project life cycle is the local working directory where our project resides, which may or may not be tracked.



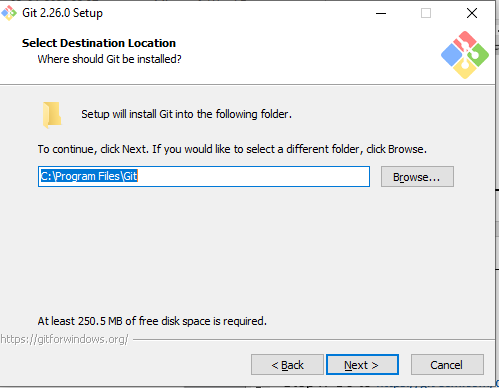
* **Initialization:** To initialize a repository, we give the **command git init**. With this command, we will make Git aware of the project file in our repository.
* **Staging area:** Now that our source code files, data files, and configuration files are being tracked by Git, we will add the files that we want to commit to the staging area by the git add command. This process can also be called indexing. The index consists of files added to the staging area.
* **Commit:** Now, we will commit our files using the **git commit -m ‘our message’ command.**

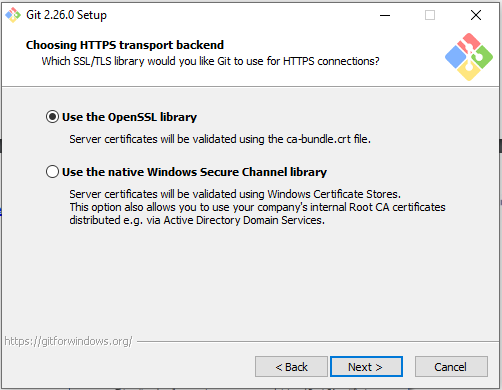
*We have successfully committed our files to the local repository. But how does it help in our projects? The answer is, when we need to collaborative projects, files may have to be shared with our team members.*

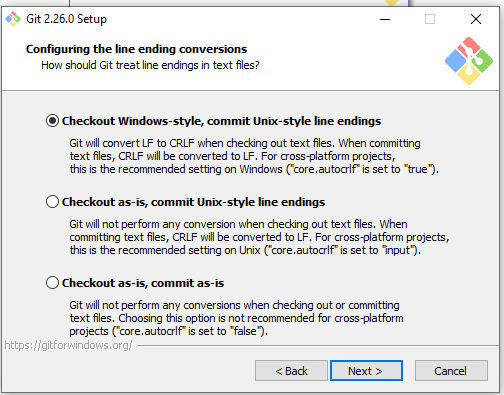
*This is when the next stage of the Git life cycle occurs, i.e., in GitHub, we publish our files from the local repository to the remote repository. And how do we do that? We do that by using the git push command.*

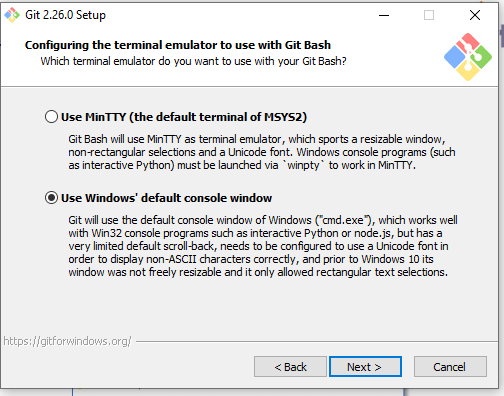
## Installing Git

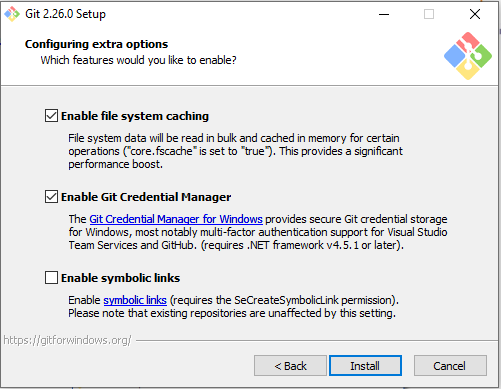
Step1: Go to <https://git-scm.com/download/win>

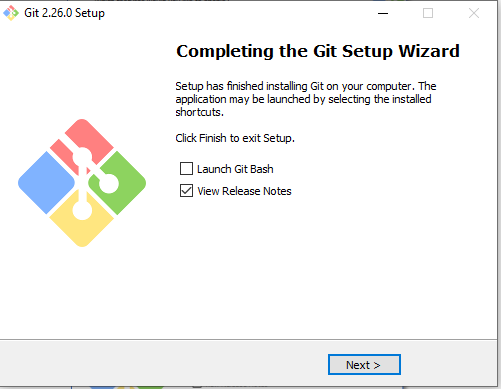








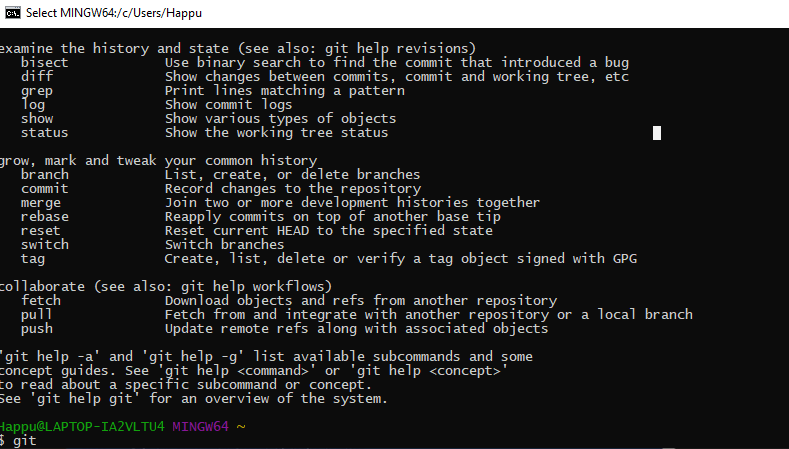


] 

After Insatallation we got Two thigns:

1: Git Command Line Tool (Interface)

2: Git Bash (TEminal Program)



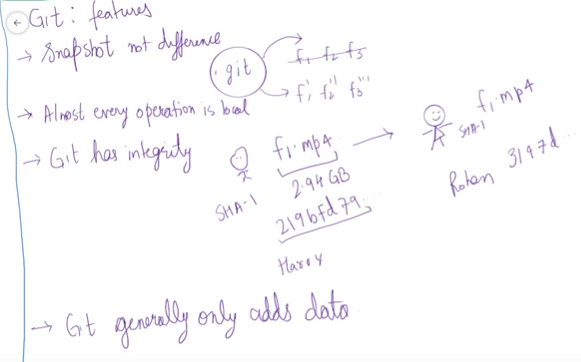
### Environment Setup

* Create a GitHub account
* Configure Git
* Create a local repository

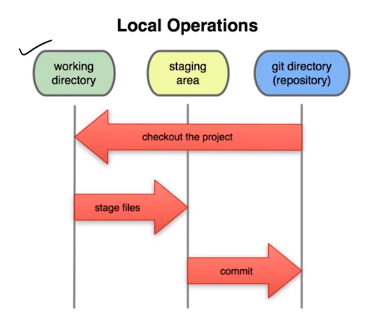
**Creating a GitHub account:**

* Go to https://Github.com
* Create a GitHub account
* Login

GIT Features:



**Git – Three stage architecture/ Workflow:**



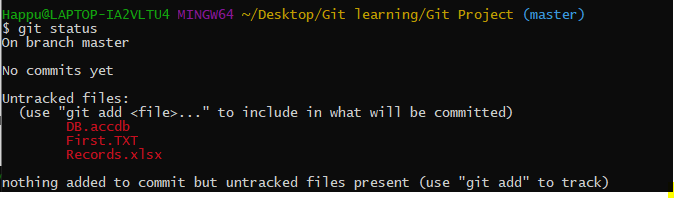
Working Directory:

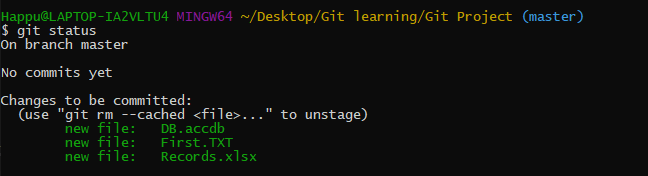
Staging Area:

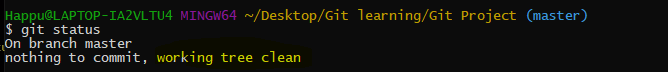
Git directory (repository):

Git Status:

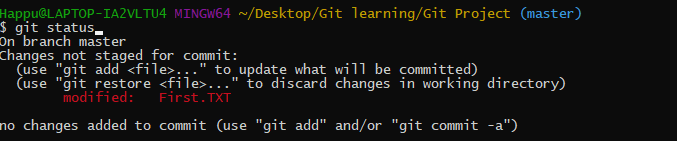




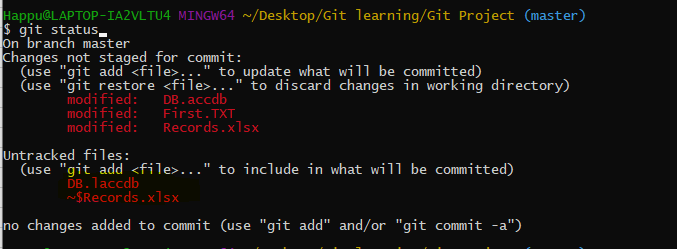




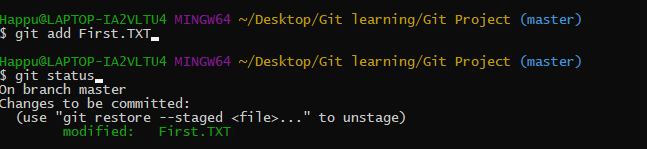
T ocheck firle is modified or not



Edited Two file

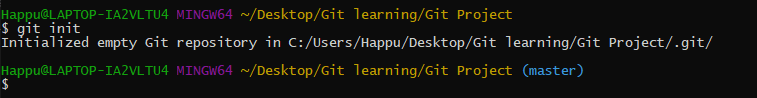


Now I want only one file ot commit:

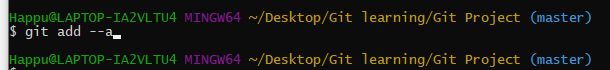


On Branch Master:

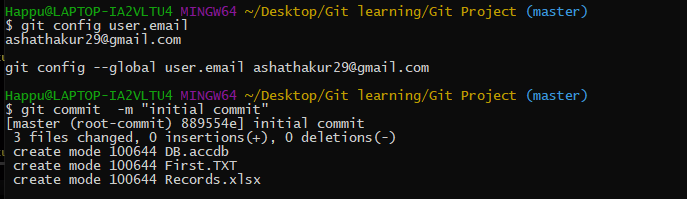
Git init: to create New git reposotry

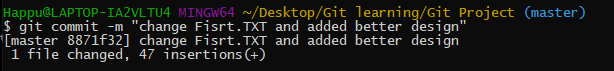


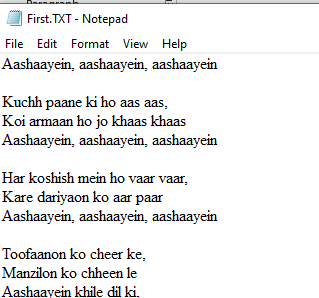
Git add –a:



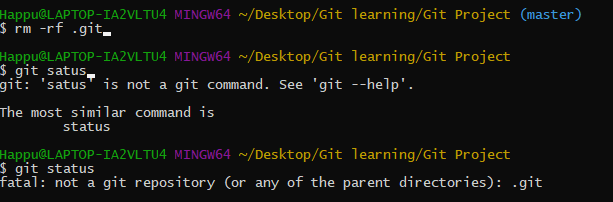
Git commit – m “initial commit”



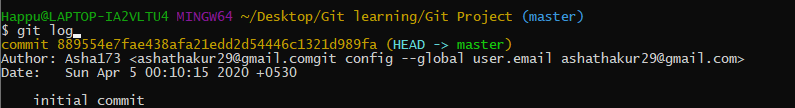




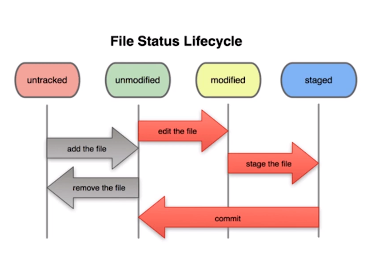
Command rm -rf .git : delete our content as .git will deleted will not able to track

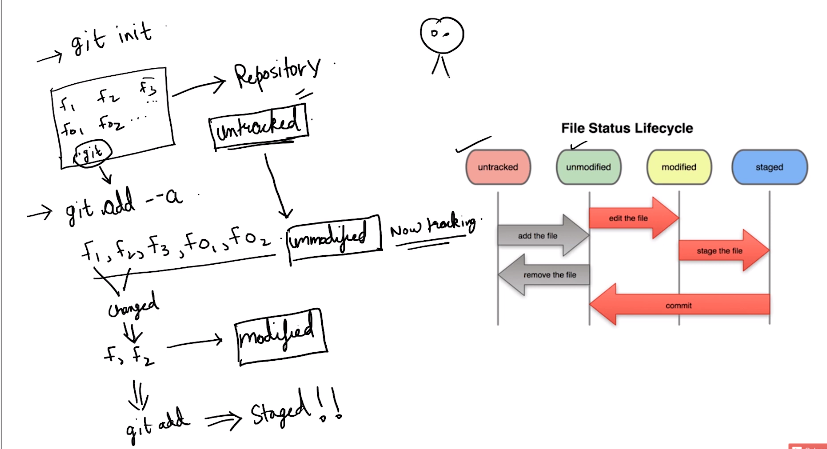


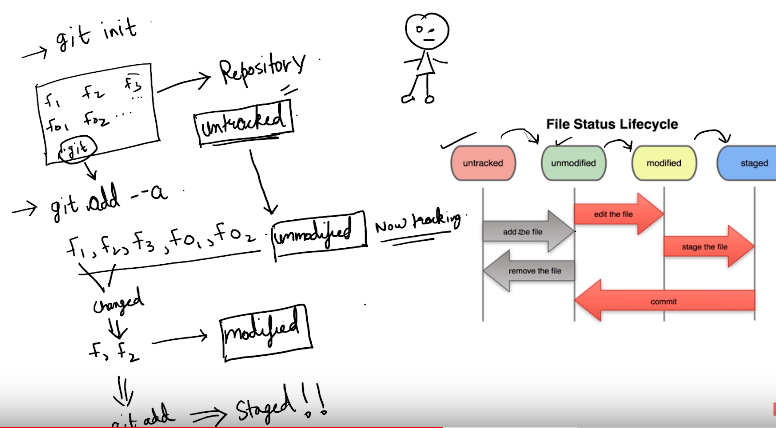
Git log:



**File life cycle:**







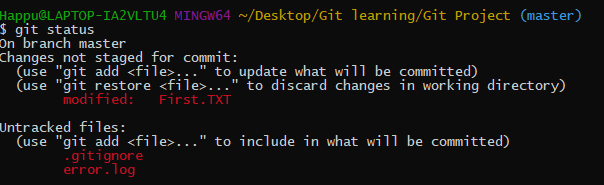
Git Ignore:

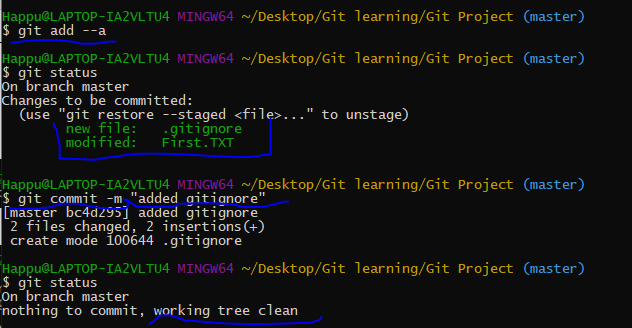
Touch Error.log will create automatic file.



touch .gitignore





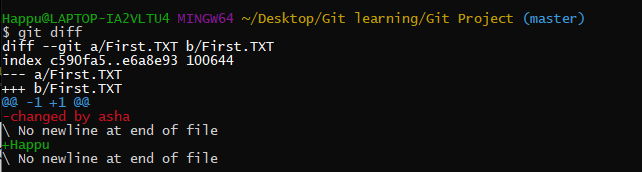


\*.Log

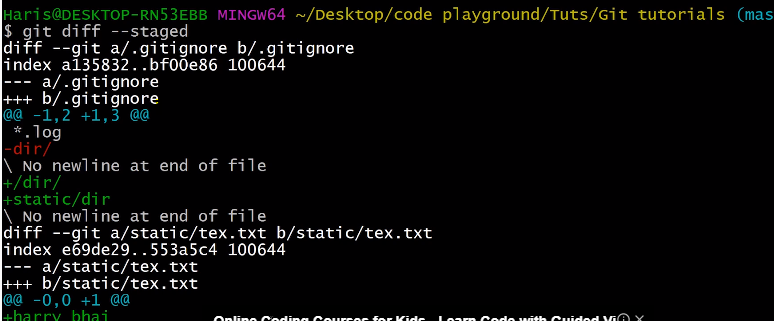
Dir/

/dir/

Git diff: Compare statging area and working directory



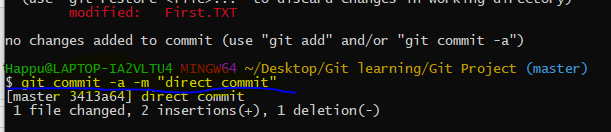
Git diff – stage: compare krta hai Pichele commit ko apki abhi ke staging area se

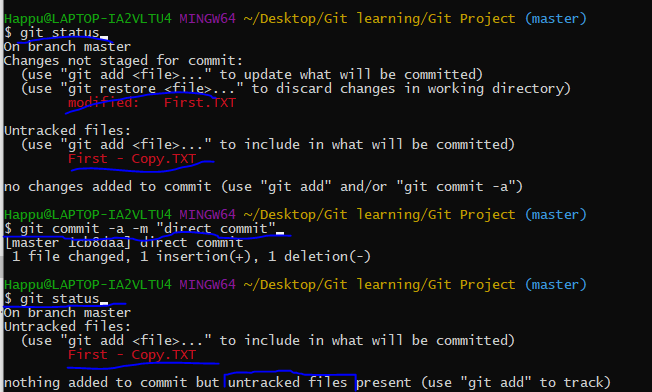


# **Git:** **Skipping The Staging Area**

Git commit -a -m “Direct Commit”

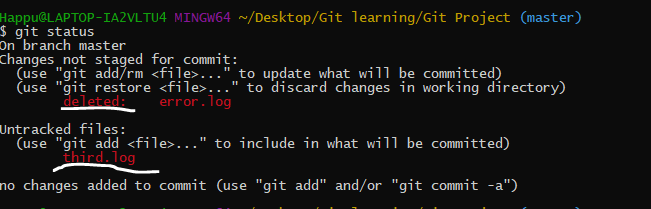
By this we will be able to commite tracked files directly without staging from unmodified to commit..

untracted file will not commit

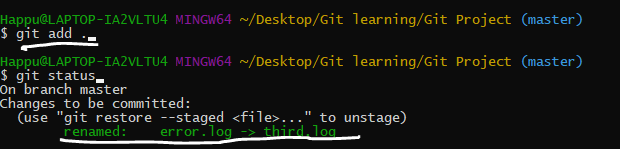


# **Moving and Renaming Files In Git**

Renamed error.log file to third.txt then checked status:



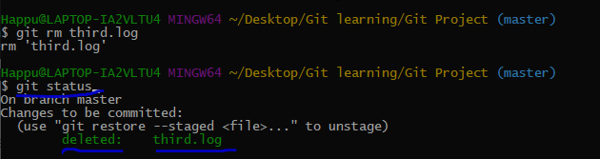
Git will say deleted error.log rather than rename, after doing git ADD. GIT WILL realized that ohhhh it was ranamed



Command to remove files:

Git rm file name:

Git rm third.log



=============================================================

\*.error.log

/dir/

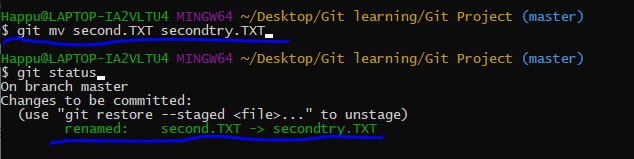
===========================================================

Rename file:

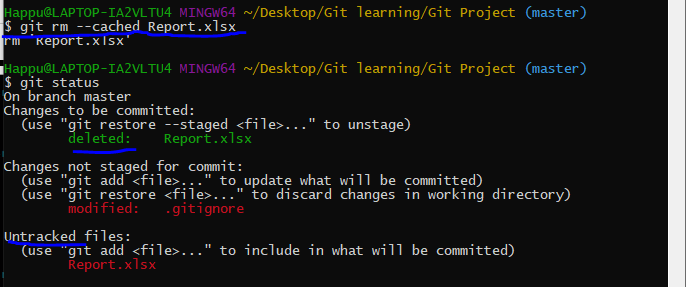
Git mv filename newfilename

Git mv second.TXT secondtry.TXT

This will rename file + staging also done at same time



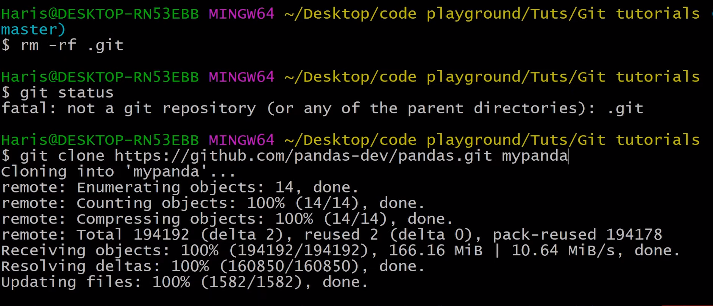
Modified and Untracked Report.txt file



# **Git Log: Viewing & Changing Commits In Git**

Rm -rf .git: to remove Repository

Clone:

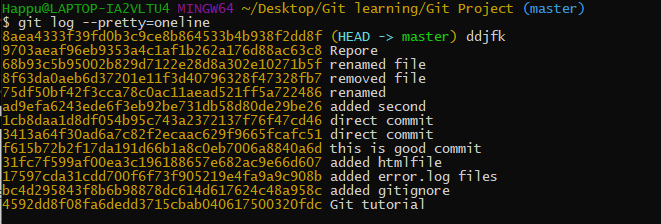


Gitlog – p

Gitlog -m

Gitlog – state

git log --pretty=oneline



git log --pretty=short

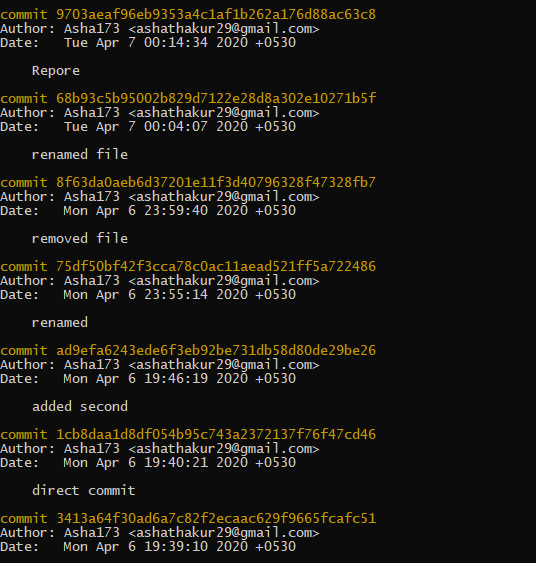
git log --pretty=full

Timer filter:

git log --since=2weeks

git log --since=2days

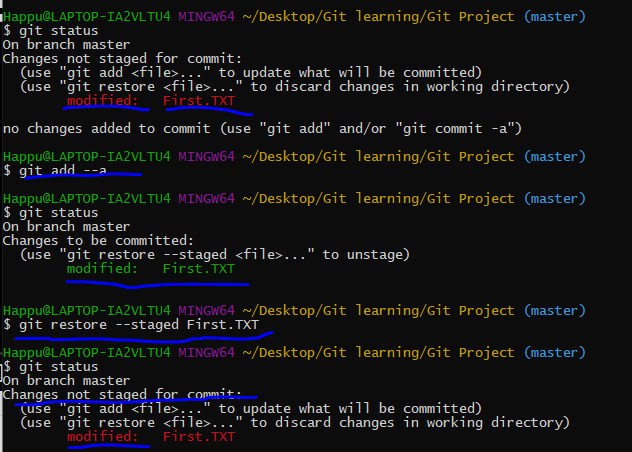
git log --since=2years



# **Unstaging & Unmodifying Files In Git**

Git restore –staged file name

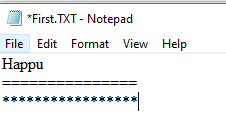
Git restore –staged first.txt

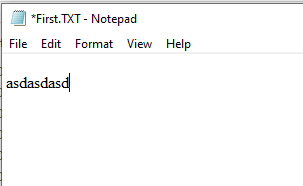


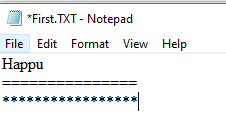
To Restore deleted/changed things we can restore/unmodify file:

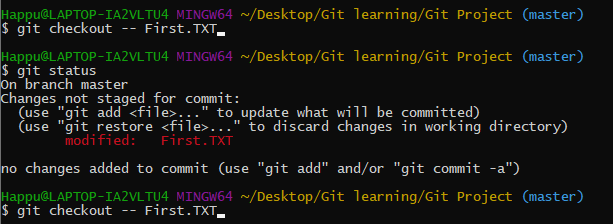
Git checkout – first.TXT



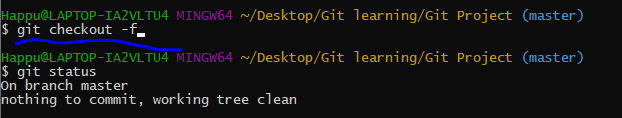








Git checkout -f : apki file last chenges hat jate hai aur hum clearn state me aa jate hai



# **GitHub: Working with Remote Repositories**

Git remote: git is distributed version system means apka ek server hota hai jisme aap apna code +history milti hai ager wo pull krta hai.

Push krta hunt oh sirf incremental push hofa jo ki maine push ki hai

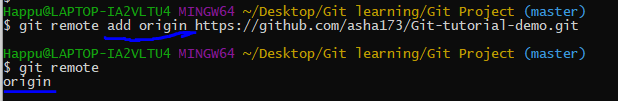
Pull: detail:code ko github se aone computer me lekar ana

Push github me dal dena

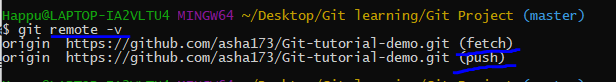
### push an existing repository from the command line

git remote add origin https://github.com/asha173/Git-tutorial-demo.git

git push -u origin master

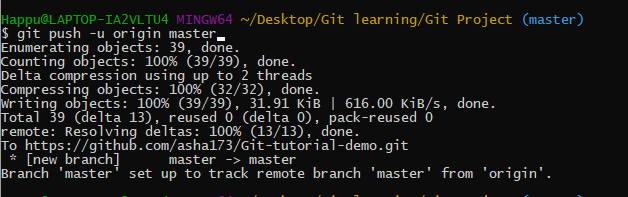


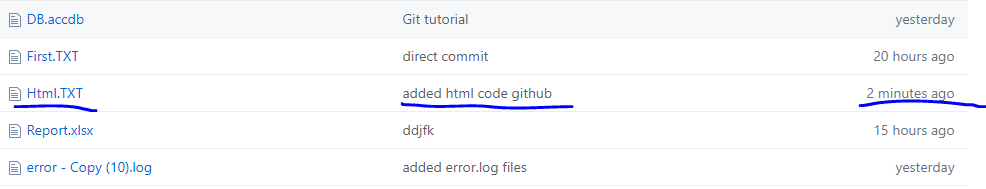
Git remote -v



To push our changes:

Check access by git push -u origin master

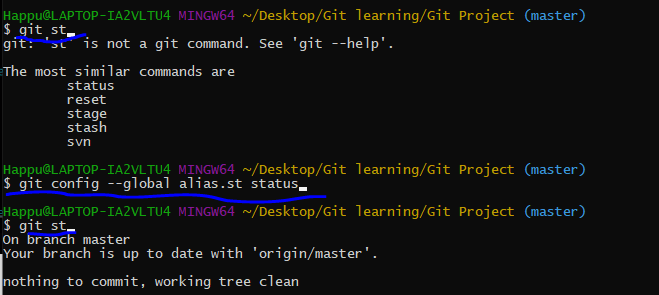




# **Setting Alias In Git**

Alis: means bada command likhne ki bajha koi chota command likhna

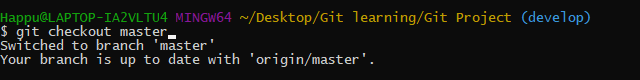
Eg: git config –global alias.st status

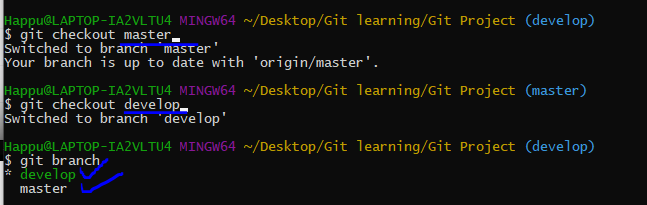


# **Git:** **Creating & Switching Branches In Git**



Can swich to master again





Git commit -a -m “use for direct commit”

# **Branching & Merging a Production Grade Project**

Download vs code to track live project..