>>To check hidden files the command is , **$ ls -a or $ls -lrt**

>> Whenever we initialize the git repository automatically one branch will be created that branch name is **MASTER.**

>>After initializing git first we need to configure username and email why because to check which user taking care of this

>> to check wether username and email configured or not the command is ,

**git config --list**

----> If we hit that command all configuration files will be listed here

**WORKSPACE :**

>> Suppose if we create file by default it will go to WORKSPACE ,

How to check whether files are there in workspace area or not to check we have to hit a command **$git status**

>>Once we hit a git status command it will throw one heading like untracked files it will display the files in red color

**STAGING AREA :**

HOW TO ADD FILES FROM WORKSPACE TO STAGING AREA :

>> By using **$git add filename** command we will add files from workspace to staging area.

HOW TO CHECK WHERE FILES ADDED INTO STAGING AREA OR NOT :

the command is $git status only , by seeing the color we will identify

>>Once we hit a command $git status it will throw one heading like **CHANGES TO BE COMMITED if file name displayed in green color** then files will be in staging area.

**LOCAL REPOSITORY :**

>>If i want to move files from staging area to local repository the command is ,

**$git commit -m "message"**

>>for each and every commit one **commit id** will be created to check the files in commit-id the command is **$git show commit-id**

Now if we hit a command git status ,it will display like **on branch master nothing to commit**

**>>HOW TO CHECK THE FILES WHICH ARE THERE IN LOCAL REPOSITORY :**

**the command is , $git log**

--> If you hit a command $git log , it will display commit-id or shake id and author and date also it will display and label message also

**GITHUB REPOSISTORY :**

>>By using **git push origin master** command we are pushing the code or files from local repository to github repository.

>>We can call GITHUB REPOSITORY AS A CENTRAL OR REMOTE REPOSITORY

------------------------------------------------------------------------------------------------------------

**GITHUB REPOSITORY :**

>>First we have to create GITHUB account after creating the GITHUB account login to GITHUB account create one repository name it as gitrepo and copy the gitrepo url and clone that repository into local machine

create one directory and give a right click select git bash and clone the gitrepo

>>after cloning the gitrepo repository enter into gitrepo directory, inside the **gitrepo** repoditory we will create files and files and we will write the codes.

Once we hit a command git log here it will display two commit-id , one commit-id by default commited the readme file into local repository.

>>In central repository we can create multiple repositories like repo1 , repo2 , repo3, repo4

-------------------------------------------------------------

to delete credentials in windows ,

go to control panel ----> user accounts ----> credential manager

we can remove or delete

------------------------------------------------------------

>>Suppose you have 100 logs if you hit a command $git log command that logs will be displayed in pages

**TO DISPLAY LOGS IN SHORTCUT :**

--> we we hit a command **$git log --oneline** it will display commit id's with only sever number and characterics and label message also it will display

>>if i hit **git log --oneline** command it will display each commit-id in single line

--------------------------------------------------

**TO CHECK LATEST LOGS :**

>>Suppose i have 100 of logs but i want to check recent five logs only the command is

**$git log -n** ,

**$git log -5** if we hit this command it will display latest 5 logs commit-id only.

$git log -2 or git log -10

>>If you want to check latest logs in shortcut the command is , $git log --oneline -n

examples , $git log --oneline -5 , $git log --oneline -2

--------------------------------------------------------------------------------------------------------

**GIT LOG --ONELINE --DECORATE :**

>> The purpose this command is this command will display logs just decorate means we will get the option like branch

>>In each and every branch logs will be there so that logs will be displayed in color wise ,

Suppose if you have created any different branches like **future branch or test branch**

>>If i hit **$git log --online --decorate** command it will display the logs with branch name clearly.

It will display in which branch which commit id's are there ,

-------------------------------------------------------------------------------------------------------------