**GIT commands that are important to learn to use command line**

* **git config –global user.name** **Anil** – For configuring the name
* **git config –global user.email sde.anilyadav@gmail.com**– to set the email
* **git config –global user.name** – For checking the changes have been affected
* **git config** **–global user.email** - For checking the changes have been affected
* **code .** – to open the VS Code
* **git init** – initialize the git repository
* **ls -lart** – To see the Hidden files
* **git status** – To check the status of the files
* **git add index.html** – To add to the staging area
* **git commit** – to send the staged files to commit
* **press Insert then :wq** then write the name of the initial commit (Wim editor)
* **touch about.html** – to create a empty file named with about.html
* **git add -A** – To add all the untracked files to staging area
* **git commit -m “Any Commit message”** – to add the commit and message **shortcut**
* **clear** – to clear the previous files
* **git checkout contact.html** – To match the file with last commit
* **git checkout -f** – All files get matched to the previous files
* **git log** – to check the activity of the files, all the commits are shown and config files
* **git log -p -1**– to see he last 1 commits on the machine and shows the changes we made
* **press q to quit**
* **git diff** – it compares the working tree with staging area(if both are same then no output)
* **git diff –stagged** -compare the last commit with staging area
* **git checkout -f** – matches the file with last commit
* **git commit -a -m “skipped staging area and fixed”** -If you wat to directly commit the file without making it to the staging area( for beginners only)
* **ls** -we can see all the files as it is feature of UNIX.
* **touch delete.html -**to create a empty file
* **git add -A**
* **git commit -a -m “Add waste file”**
* **git rm –cached** – it will remove from staging area only not from hard disk
* **git rm waste.html** -If you want to delete a file
* **git commit -a -m “removing waste”**
* **git log -p -2** – If you want to see 2 commits
* **git status -s** – shows the summarized status(modify the file then use this command )
* git add file.html
* **git status -s** shows the m in green
* **touch .gitignore** -Files that you don’t want tag
* **touch mylog.log** -Example of log file write anything
* **in gitignore write mylogs.log to ignoire all the file with this name**
* **run git status -**it will not show that file added in the gitignore
* **add this in gitignore /mylogs.log -**if you want to add the same file in same folder
* **git rm –cached logs/mylogs.log** -to remove from the stages area
* **\*.log in gitignore – To add file with .log**
* **ignore a folder foldername/**
* **Git commit -a -m “Ignoring the files in the ignore files”**

**Creating a Branch by command line**

* **git status** -Explain about the on branch master
* **git branch -**it shows all the branch in the git
* **git branch update1** -New branch with name update1 is created
* **git branch** – now two branches will be shown
* **git checkout update1** -Switch to the branch update1

**Using the merge feature in the git command line**

*Make some changes in the code you have written in any of the file then use next command*

* **git add -A**
* **git commit -m “Fixed the programs and added more features”**
* **git status**
* **git checkout master** – To move to the master Branch
* **git checkout update1**
* **add the comment in the file any comment**
* **git commit -a -m “added some comments”**
* **git status**
* **git checkout master** -for merging the branch to master then we will switch to master
* **git log**
* **git merge update1** -this merge the update code to master
* **git log -p -2** -We can see the logs of the same commit in update to the master now.
* **git checkout -b nodeintegration** -It will switch and make the branch in this command

**Create a js file and add some printing code in it then commit**

* **git commit -a -m “Added the node file in the code”**
* **git status**
* **when we switch to master the file of node will be lost now**
* **git checkout master**
* **in about.html write anything**
* **git commit -a -m “modified the about page”**
* **git status**
* **git checkout nodeintegration**
* **files again comes**

**Now talk about GitHub service (Microsoft service)**

**On command line**

* **git checkout master**
* **remote repository**
* **paste the repository origin URL in the GitHub**
* **git remote**
* **git remote -v (push and fetch urls)**
* **git push origin master ( as the private repo then it will show there is no repo)**
* **create new folder on desktop on clone**
* **copy the clone url**
* **git clone url foldernanme (clone to the folder name )**