**Git**

* git config --global user.name “Yusufbek” # setting your name
* git config --global user.email [yusufbek02106@gmail.com](mailto:yusufbek02106@gmail.com) # setting you email
* git config --global init.default branch main # setting this branch as main branch
* git config -h # starting help menu
* git help config # opening info for ‘config’ command
* clear # clearing all the commands from the current window
* cd “address” # moving to the address
* git init # initializing empty repository
* git status # getting the current status like modified, staged, cached, removed files info
* git add hi.ppt # adding file as cached into staging
* git rm --cached hi.ppt # removing file from cached files in staging env
* save \*.txt in “.gitignore” file to remove all txt files from tracking
* git add . or all or -A # adding all files in entire directory as cached into staging env
* git commit –m “msg” # committing the changes to the git history with message
* git diff # getting the difference between current state and latest commit
* git restore --staged hi.ppt # removing file from staged env to working files env
* git commit -a -m “msg” # committing changes without adding them to the staged env
* git rm hi.ppt # deleting file from directory via git
* git restore hi.ppt # restoring deleted file
* git mv hi.ppt hello.ppt # renaming from hi.ppt to hello.pt
* git log # viewing all commits to the repository
* git log --oneline # viewing commit history briefly
* git commit –m “new msg” --amend # changing the latest commit
* git log -p # viewing commit story in detail
* git reset ‘commit number’ # going back to the commit with commit number
* git rebase -i --root # opening window to do different operations on the current branch commit history
* git branch FixError # creating a new branch called FixError
* git branch # listing all branches in the repo
* git switch FixError # switching to the FixError branch
* git merge -m “msg” FixError # merging changes from FixError branch into main branch/ if there are conflicts the env changes to master|MERGING and after solving the conflict we can commit the change and the env changes to master.
* git branch -d FixError # deleting a branch
* git switch -c UpdateError # creating and switching to a branch called UpdateError

**Git**

* git remote add origin “repository link in github” # connecting local repo to repo in github
* git branch -M main # setting main branch as default
* git push -u origin main # pushing changes to main branch
* git push --all # pushing all branches to github
* git fetch # downloads changes in the cloud to the local machine
* git merge # merges the downloaded changes
* git pull # git fetch + git merge