**Git Commands (Windows)**

|  |  |  |
| --- | --- | --- |
| Commands |  | Discerption |
| git init |  | (initialize empty git repository/folder) |
| cd |  | (used to move here and there) |
| dir |  | (list folder files in current folder) |
| mkdir |  | (create new folder) |
| copy nul names.txt |  | (create new file of any extension) |
| dir -a |  | (show hidden folder) |
| dir .git |  | (show what is in the git folder) |
| git status |  | (show what is committed or going to be committed or if it is at stage) |
| git add . |  | (move files to staging process) |
| git add names.txt |  | (move file to staging process) |
| git commit -m “comment” |  | (git files and folders will be committed) |
| git restore –staged names.txt |  | (remove from staging process) |
| git log |  | (shows history) |
| del names.txt |  | (delete file or folder) |
| git reset 1d66744e11 |  | (commits above this one, will go in staging area) |
| git stash |  | (move all modifications from staging area to reserved area) (modify project to its previous stage like to here 1d66744e11 from where the mistakes happen) (simply we can go back in project) |
| git stash pop |  | (can able to retrieve the mistake that happened, all files and folders) |
| git stash clear |  | (clear mistakes) |
| git |  | (show commands we can use with git) |

**Remote GitHub**

|  |  |  |
| --- | --- | --- |
| Commands |  | Description |
| git remote -v |  | (show links connected to local project folder) |
| git push origin master |  | (update main branch code base with my code) |
| git clone |  | (get code base from github to local pc to use in git) |
| git remote add upstream |  | (link of the project from where I have forked project) |
| git branch yadnyesh |  | {create new branch} |
| git checkout yadnyesh |  | (move head to yadnyesh branch) |
| git remote add origin https//:repository link |  | (connect local pc git to you GitHub repository to commit files and folders) |
| git push origin master -f |  | (force push) |
| git fetch –all --prune |  | (fetch the main branch of the actual production project to you local directory) |
| git reset –hard upstream/main |  | (make the main branch of our project just like the upstream project) |
| git pull upstream main |  | (above two commands as single) |
| git rebase -i 1d66744e11  # s |  | (merge number of commits to single commit)  Squash (change from pick to squash to merge) |
|  |  |  |