

Github

Commands

1. git status (this commands show tracked and untracked files)
2. git add . / git add file name to make untracked file to tracked.
3. git commit -m "commit message" / to commit the changes in the remote repo
4. git push origin main / to push code to repo
5. git log / to see who made changes
6. git log —oneline / to get commit id
7. git show <commit id >
8. git branch -a / to list all the branches
9. git branch -c <branch-name> / to create new branch
10. git checkout -b <branch-name> / create new branch
11. git checkout <branch-name> / to go into particular branch
12. git switch <branch-name>/to switch into particular branch
13. git push origin <branch-name>
14. git merge <branch-name> / to merge branch content into main branch make sure you are in main branch.
15. git diff / to see changes made in the file . / this command work before staging means before add .
16. git diff —cached // after staging.
17. git diff <commit-id-previous-commit>..<commit-id-latest-commit> / to see diff after the committing .

To Rollback

- 1.if file didn't go to staging area then to rollback
git checkout <filename with extension>

2. if file is staged means git add . has already runs then to go back

git restore —staged <file-name>

3. if file is already committed and to rollback from that.

git revert HEAD.

While we git revert HEAD it gonna create new commit message and stores History of it

```
hemantkumar@hemantkumar-MS-7D90:~/gitrepos/titanwork$ git log --oneline
c9fc568 (HEAD -> main) Revert "playbook"
f974df0 playbook
d0220a2 (origin/sprint1, origin/main, origin/HEAD, sprint1) jupyter changes
33abff0 Update saturn10.py
58d5c75 updated
b9f697f updated saturn moon name
6ba6483 saturn moon
ecce3aa newfiles committed
```

To make sure we donot stores history

git reset —hard <commit-id> / commit id where to rollback

SSH-Login

1. First create ssh login keys

ssh-keygen

ssh-keygen

2. Copy ssh public key and paste in [https ://github.com/settings/keys](https://github.com/settings/keys)

3. Now we can use ssh login keys to clone our repo.

Semantic Versioning

1.To list tags

git tags

2. To see details of particular tag.

git show <tag-name>

3.To create a tag

```
git tag -a "tag name " -m "Release tag-name"
```