github - git

- git init
- git add . or git add filename

- git log or git log --oneline : display commit snapshots

- gitignore : used in git repo to ignore files and folders when using git.

- gitkeep : a convention used by git user to track empty directory in a repo.

* • To create branch
  → git branch branch name

- Switch to different branch
  → git switch branch name or
  git checkout branch name

- create and switch to that branch.
  → git switch -c branch name

- to merge branch
  → go to brach on which you merge (git checkout main)
  → git merge branchname (git merge dorkmode)

- Conflict when chages in same file
  - to abbort → git merge --abort

- rename current branch
  → git branch -m new branch name

- rename any branch
  → git branch -m old branch new branch

- delete branch
  → git branch -d branch name

- ** git diff

- Compare staged area with repo
  → git diff --staged

- Compare b/w branch
  → git diff branch name1 branch name2
  → git diff branch name1 .. branch name2

- Compare specific commit
  → git diff commit hash1 commit hash2

* git stash : A built-in command in Git that temporarily save changes to your working directory so you can switch branch or work on other task without committing your changes.

- git stash or git stash save "file name"
- to get all list → git stash list
- to apply the last stash → git stash apply

- to apply specific stash
  → git stash apply "stash@{1}"

- To remove most recent stash
  → git stash drop

- To remove specific stash
  → git stash drop "stash@{3}"

- To apply most recent stash and delete it
  → git stash pop

- To apply and delete a specific stash.
  → git stash pop "stash@{2}"

* Creating tag.
  - create tage → git tag tagname
    with message → git tag -a tagname -m "message"
  - Show all tags → git tag
  - Tage a specific commit → git tag tagname commithash
  - To show details of a tag → git show tagname
  - push to remote repo → git push origin tagname or
    push all tags git push --tage
  - delete local tag → git tag -d tagname
    from remote repo → git push origin: tagname
  - tag do not get pushed on remote repo when
    we use git push

* Git rebase: used to change the base of a branch it allow to move a brach to a new starting point.

- git rebase main (name of branch to which point)

* git reflog : that keep a record of changes made to a git repo's branch, commits and the pointer

* git log : display the history ~~for~~ of commits for a branch or file.

* Undo changes

- undo staged changes
  → git reset file name or git reset

- undo commit (before or after pushing)
  → git reset --hard commit hash or git reset --hard "HEAD@{2}" then git push --force
  this command absolutely remove the pushed commit from the remote history.

→ git revert commit hash : create a new commit to undo changes. (then push)

Git :- Git is a version control system (VCS) that tracks changes in computer files, especially source code in software development.
tool that manages changes to source code over time

Github :- it is a cloud based platform where we can store, share and work together ~~to~~ with other to write code.

→ git --version
→ git config --global user.name "prince"
→ git config --global user.email "abc@gmail.com"
→ git config --list

→ git status :- to display status of code.
- untracked :- new file that git doesn't track yet
- modified :- changed
- staged :- file ready to commit (file added)
- unmodified :- unchanged

→ fork :- A fork is a new repo that share code and visibility setting with the original "upstream" repo
→ it is rough copy.

* • git init :- initialise the git
• git branch -m main :- change master to main (branch rename)
• git remote add origin url :- add new remote repo
• git push -u origin main :- to set upstream to avoid use of origin main
• git remote -v → verify the remote.