Gayatri Shinde

Git Cheat Sheet

git init : It is used to initialize the GIT
git status : To know the current state of the repository
git add <filename> : It will move a single file to the stagging
git add . : It will move all the files to the stagging
git commit -m "message" : It is used to create a new commit in GIT with a specific commit
message
git log : It will display a list of commits in a GIT repository
git ls-files : It will show all the files that have committed in the GIT repository
git log --oneline : It will show the on-line/short commit ID details
git show <commit ID> : It will show all the details of the particular commit ID
git diff <commit ID1> <commit ID2> : It will show the difference between the two commit ID's
git commit -am "message" : used to create a new commit in Git, automatically including all
tracked files that have been modified or deleted
git config --global user.name <name> : command is used to set the global Git configuration for your
username
git config --global user.email <email> : command is used to set the global Git configuration for your email address
.gitignore : text file used in Git repositories to specify which files and directories should be ignored and not tracked by Git.
git tag --a <tag> <commit-ID> -m "message" : It is used to add a tag to a commit ID
git show <tag ID> : used to display information about a specific Git tag
git diff <tag ID1> <tag ID2> : Used to show the difference between the two tag ID

Steps to create a Repository on GIT-HUB :
1. Open github.com in browser and sign-in/sing-up
2. To create a repository, click on "+" button from the top-right corner.
3. From the dropdown menu select "New Repository"and it will redirect you to "create a new repository"page.
4. Enter Details:
a. Repository Name
b. Description
c. Select if you want to create Public/Private repository
5. Click on create repository

git remote add origin <link of the repository> : command is used to add a remote repository to your local Git repository
Origin - it is a short name that refer to the default remote repository when setting up a new repository or cloning an existing one
git remote -v : To check if we have connected to any remote repository
git push -u origin main : used to push your local branch named "main" to the remote repository named "origin" and set it as the upstream branch

Steps to Generate a TOKEN in GITHUB
1. Go to github.com
2. Click on account icon from top-right corner.
3. Click on settings from the menu
4. Click on developer settings
5. Click on personal access token and from the dropdown menu select Tokens(classic)
6. Click on generate new token(classic)
7. Enter Details:
a. name
b. Expiration
c. Select all the checkbox
8. Click on generate token
Note: after creating the token copy the alpha numeric token before refreshing/going to other page.

git push -u origin main --tags : used to push both the main branch and all tags to the remote
repository named "origin."
git cone <link of the repo> : It will clone the remote repository to another machine
git pull origin main : It will pull the changes from the remote repository to local
git fetch origin main : It will fetch the changes from the remote repository
git stash : used to save changes that you have made in your working directory but do not want to commit yet. It allows you to temporarily store your changes and revert your working directory to the state of the last commit
git stash list : It will list the modification in stash
git stash pop : It will bring back the files from stash state to stagging state
git commit –amend : It will change the message for latest commit ID
git rebase -I HEAD~1 : It will Squash the two commit ID from HEAD
Pick - to keep
squash - To Squash

Difference between CLONE, PULL, FETCH and MERGE
CLONE: it will clone the code from the remote repository to local

PULL: it will bring the changes from the remote repository to local
FETCH: it will check for the difference between remote repository and local.
MERGE: it will merge the changes from remote to
git rm <filename> : It will move the file to the staging area
git mv <old_filename> <new_filename> : It will change the file name and move it to the staging area
git reset --soft <commit ID> : allows you to move the branch pointer to a specific commit while
preserving the changes introduced in the subsequent commits
It will bring back all the items to the staging area
git restore --stage <filename> : It will move the file from staging area to untracked state.
git reset --hard <commit ID> : It will directly move files to the untracked state and remove the
changes will that commit ID
git revert <commit ID> : It is used to create a new commit that undoes the changes made in
a previous commit. It allows you to effectively revert or undo the changes introduced by a specific commit while keeping a record of the reversion in the commit history.
git branch : It will show the branch you on
git branch <branch name> : It will create a new branch
git checkout <branch name> : It will switch to the other branch
git checkout -b <branch name> : It will create and switch to the new branch
git branch -d <branch name> : It will delete the branch
git branch -D <branch name> : It will delete the branch forcefully
git rebase main : It is used to integrate changes from one branch into another by moving or combining commits
git push -d origin <branch name> : To delete remote branches from local and we can run this command from the main branch
git checkout -b <branch name> <commit ID> : It is used to create a new branch from any particular branch
git show : show any object in Git in human-readable format