❖ Git Commands

1. For Version of your Git ->

   `git --version` OR `git -v`

2. For the status of any folder i.e, the folder is git repo or not ->

   `git status`

3. Initialized any folder as a git repo i.e, creating an empty git repository means it creates one hidden folder in that particular folder called `.git` ->

   `git init`

4. Working directory to Staging area

To add files (stage files) ->

```
git add.
```
(All files are added)

5. Staging area to git repository

To commit files ->

```
git commit -m "message"
```

6. Check all the changes user-wise

```
git log
```

7. Configure User name and Email Id

```
git config --global user.name "Your Name"
git config --global user.email you@example.com
```

8. Check out your User name and Email Id ->

```
git config --global user.name
git config --global user.email
```

9. Remove files from the staging area i.e, Unstaged files ->

```
git rm —cached file_name
```

OR

```
git rm —f file_name
```

10. Restore/Discard changes in files in the working directory ->

```
git restore file_name
```

11. Imp step -> Set your SSH Key

12. Create Repository and set Origin ->

Create Repository manually on Github

+

git remote add origin SSH Key

13. Make branch master

git branch -M master

14. Push & Pull -> [Branch is master]

Push - git push origin master

Pull - git pull origin master

OR

-u origin

15. The branch is either main or master

If you want branch main then

replace master with main.

16. Scanning changes to files ->

In the working directory -

```
git diff
```

(For all changes)

OR

```
git diff file_name
```

(For particular file)

In the Staging area -

```
git diff - - staged
```

(For all changes)

OR

```
git diff - - staged file_name
```

(For particular file)

17. Coming to the previous commit
(i.e, Restore changes) ->

Suppose we staged those files,
First of all, unstage those files ->

`git reset file_name` (particular)

OR

`git reset` (all)

Restore changes ->

`git checkout file_name`

(For a particular file)

`# Coming to the previous commit in one click ->`

`git checkout.`

(For all files)

18. # Some Imp flags ->

--staged (For Staging Area)

--global (For Global)

--Version (For version)

-v (For Version)

--cached (For Force Remove)

-f (For Force)

-M (For Branch)

-m (For commit)

19. Several git add commands:

1. git add -A -> [Stages all]

It will stage all the files i.e

new, modified, and deleted files

2. git add. -> (without deleted)
Stages new and modified files only and not deleted files

3. git add -u -> (without new)
Stages modified and deleted files only and not new files

20. Cloning a git repository:
[First, make one folder in your computer in which you want to clone the git repository]

1. Normal way: Download a zip file of the repo folder and then unzip it
-> Download zip file

2. Git way-1: Download the Repo folder in your computer folder

-> git clone SSH Key

3. Git way-2: Without downloading the repo folder make your computer folder a git repository and download the files only into your folder.

-> git clone SSH Key .

21. Branches In Git:

1. Checking current branch ->

git branch

(It will show all the branches corresponding to that repo and current

branch will be highlighted in green colour)

2. By default Branch is -> master

3. Two types of branches are there on git ->
   1.master
   2.main

4. master ->
   1.This is your actual main important branch.
   2.This is the actual official working version of your project.
   3.This is leader branch.
   4.It should be protected.

5.we should not make any changes directly on master branch.

5. main ->

1.It is another default branch on git which will be automatically created for merging.

2.It is a branch where all the changes will occur and it will be merged back into master branch.

6. Changing the branch:

1. main to master ->

git branch -M master

2. master to main ->

```
git branch -M main
```

7. Creating new branch ->

```
git branch [branch name]
```

8. Switching the branch ->

```
git checkout [branch name]
```

9. Merge the branch in master ->

```
git merge [branch name]
```

10. -u meaning ->

```
git push -u origin master
```

(It means that if I want to push anything afterwords I'll use only `git push`

and it will automatically push this on branch master)

11. We can make several branches on git with default branch master.

12. Got ERROR while pushing ->

```
error: failed to push some refs to [remote repo]
```

    1. git pull origin master
    2. git pull --rebase origin master (✔)
       (master OR branch name)

22. Deleting A Local/Remote Branch ->

1. Local -> From your local machine(PC)

```
git branch -d [branch name]
```

OR

```
git branch -D [branch name]
```

(If branch is not fully merge within your actual default branch)

2. Remote -> From your github account

```
git push origin --delete
[branch name]
```