- git commands :- 1) git clone → For cloning the File

2) cd {File name} → To go in that File

3) clear → to clear

list ← 4) ls → To show Readme File

5) git status → To know the status

6) git add <File> → To add File to push /stagging area

7) git commit -m →message → To commit the File , to local database

8) git log → It will show the history of all log.

9) git push → It will push the File on the github.

10) git difftool HEAD → It shows the difference b/w your local changes & previous version of the file.

11) git checkout -- <File> → to discard the changes of the File.

12) git checkout -- . → to discard all the changes of all the file.

13) git revert (commit id) → it is used to undo the file of commited
⇓
with the help of git log

14) git revert -n (commit id) → it is used to do explicitly commit. After typing this command we need to type git commit command.
('message' which can be add)

15) git reset --hard (commitid) → it is used to reset previous commit.

~~To create branch~~

16) git branch → it is used to create the branch.

git branch <newbranch> → it is used to create new branch
eg: git branch thirsty

git checkout <newbranch> → it is used to make new branch active

→ Combining this two → git checkout -b <branch name>

17) To merge the branch

git checkout master → we Ist need to move main branch where we need to merge

git merge <newbranch> → This merges newbranch into the active branch.

18) To delete the branch

git branch -d <branchname> → it is used to delete the branch

Date

18) HEAD → It is a reference to most recent commit in current branch

~~git show head~~ git show HEAD → to show most recent commit

git difftool HEAD-2 HEAD-1 → to show difference in the recent commit

19) .gitignore file

touch .gitignore → to create gitignore file

git status

git add .gitignore

20) To show difference in code we use 'meld' with command -

git difftool origin/master

git difftool -t meld → to lauch the tool
tool

21) Pull Request :- It is used to pull/merge the code in the owner's repository. The request is called pull request.

* Git & Github

→ git --version : To get version of git

→ git config --global user.name "your name here"
: To enter your name

→ git config --global user.email "your email id here"
: To entere your email

→ To change name & email:
git config --global --edit

→ To make new folder : mkdir <dir name>
eg: mkdir examplegit

→ To change directory : cd

→ To intialize git : git init

→ For Status of repository : git status

→ To add your file : git add <File name>

→ To commit : git comit -m "message of commit"
↳(should be meaningful message)

→ To previous history : git log

→ To clear : Ctrl + K or clear

→ To add all file if any file is changed or anything else :-

```
git add .
```

→ To go to previous commit :-
/ To change the branch

```
git checkout <commit hash code / branch name>
```

↳ To return the commit when gone to previous one :

```
git checkout master
```

→ To create new branch:

```
git branch <branch name>
```

→ To ~~ch~~ create both checkout & creating new branch :-

```
git checkout -b <branch-name>
```

→ To merge the file from one branch to other:

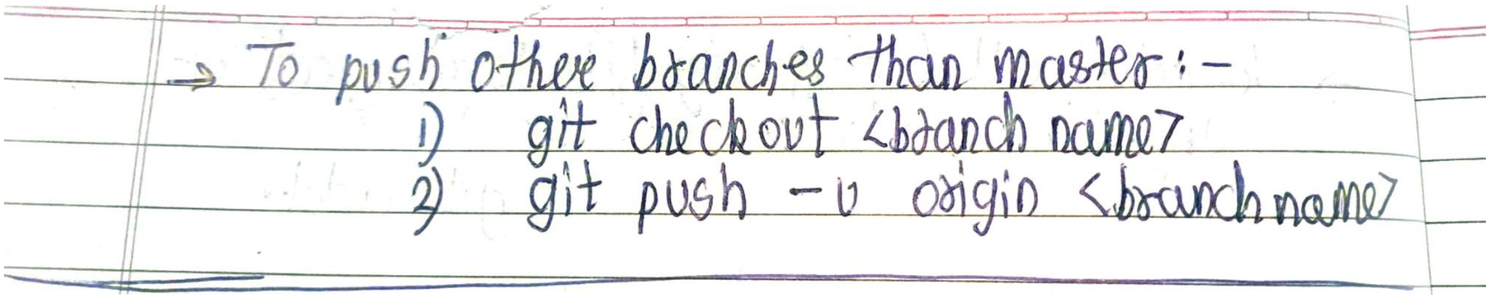
```
git merge <branch name>
```

→ To ~~cr~~ ignore the file & create .gitignore file :-

```
touch .gitignore
```

→ `git remote -v` when pushing the code before

→ `git push` : to push git



→ To push other branches than master :-

1) git checkout <branch name>

2) git push -u origin <branch name>

* Git Tutorial :-

-> Git :- It allows us to maintain history of the project and different versions of project.

-> Github :- It is a platform which allow us to host our projects(ie repository)

=> ls :- list all the files & folders

=> mkdir "Foldername" :- To create a new folder using command line.

eg: mkdir project

=> cd "Folder name"/"File name" :- To go inside any file & folder.

eg: cd project

=> git init :- To intialize the git

=> ls -a :- show all the hidden files

=> touch "File name.extension" :- To create a new file.

=> git status :- to show the status of file, folder, repo

=> git add "File name" :- To add the file

git add . :- To add all the file in a folder

=> git commit -m "type message / comment" :- To comit the file.

⇒ vim "File name":- To go in vim editor to change something in file.

⇒ cat "File name":- It displays whatever is there in the file.

⇒ git restore --staged "File name" :- To discard the changes when file ~~ae~~ is added to stagging area.

⇒ git log :- It is used to see the entire history of file.

⇒ rm -rf "File name" :- To delete the file

⇒ git reset "log id code":- It is basically used if by mistaken some file or folder is deleted or any file is commited and we don't want to commit that than it is used to get back to previous where it was.

⇒ git stash :- It is used to ~~bring~~ save the files and folder in backup whenever required and is not saved in the present file.

git stash pop :- It is used to bring the files & folder back into staging area.

→ To upload or push your personal projects to git

S-1 create a new repository

S-2 copy the repo link

S-3 git remote add origin "repo link"

S-4 git push origin master

→ ~~git branch :- to create a new branch~~

"

git branch "branch name" :- to create a new branch

↳ git checkout "branch name"

⇒ git checkout main :- to upload the files & folder to main branch

⇒ git merge "branch name" :- to merge the code (i.e files & folder) to the main branch

→ To work or contribite to any projects:

S-1 Fork the repository

S-2 git clone "repo url"