Git And GitHub Notes

Navigation

```
ls - list directory
pwd - current directory
cd folderName - change directory
cd.. - previous directory (back)
mkdir - make directory
clear - clears the command directory
ls -a : shows all hidden files
```

Configuration & Setup

```
$ git config --global user.name "John Doe"
$ git config --global user.email
johndoe@example.com
```

Requirement: Install VS Code> Git (with Bash)

https://git-scm.com/

Git Bash Codes

```
git --verion
```

Cloning (From GitHub to Local)

```
git clone copyPasteTheLink
```

Main Codes

```
git status - checks the status
git add . - adds all the files for staging
git add fileName -adds particular file
git commit -m "Commit Message" - commits
git push origin branchName - pushing the code to GitHub
```

Pull Command

This is used when we want files that are updated in the remote repo i.e. GitHub portal to our local repo system.

`git pull origin main`

Git Initialization (local to GitHub)

Navigate to the folder you need to initialize

1. Check existence of git

   `ls -a` if there is no .git file proceed
2. `git init ; initialization of git in that folder`
3. Create a New repo in the GitHub w/o README.md file and run this command on bash
   a. `git remote add origin <--HTTPSlink-->` Which creates a new remote repository named origin, then your local repo will be ready to push to remote
   b. To check which is the current remote `git remote -v`
   c. To check current branch `git branch`
4. To Rename the current branch `git branch -M (new name)`
5. Run all the Main Codes (status , add , commit , push)

Push Operation

`git push -u  origin main`

-u sets upstream, means it will push all the code to origin main after this code , user can use only `git push` after this.

Branches

1. To create `git checkout -b <new branch name>`
2. Navigate to other branch

   `git checkout <branch name>`
3. To delete a branch

   `git branch -d <branch name>`
4. To compare different branches

   `git diff <branch name>`

Show Commits Using graphs

`git log --oneline --decorate --graph --all`

Merge

1. Merge command `git merge <parent branch>`
   The current "child branch" will be merged with the "
   `<parent branch>` "
2. Pull Request on the GitHub .
   After merging we have to use
   `git pull origin main` to replicate it to our local repo.

Resolving Conflicts

An event that takes place when Git is unable to automatically resolve the difference in code between two commits.

Reset staging

1. `git reset <file name>` or `git reset` undo's what had been added using add command

2. To undo a commit by 1 `git reset HEAD~1`
   forgets the previous commits
3. To select a particular commit use the HASH CODE of that commit

   `git reset <HASH CODE>`

   To completely remove all the previous commit

   `git reset --hard` or directly use `git reset --hard <HASHCODE>`

HASH CODE

This is a unique code of every commit

Can be accessed by `git log` which shows history of all the commits with hash codes

Fork

A fork is a new repository that shares code and visibility settings with the original “ Upstream” repository.
A rough copy.