Git
Git is a free and open source version control system
version control system is a tool that helps to track changes in code

GitHub
GitHub is a website where we host repositories online

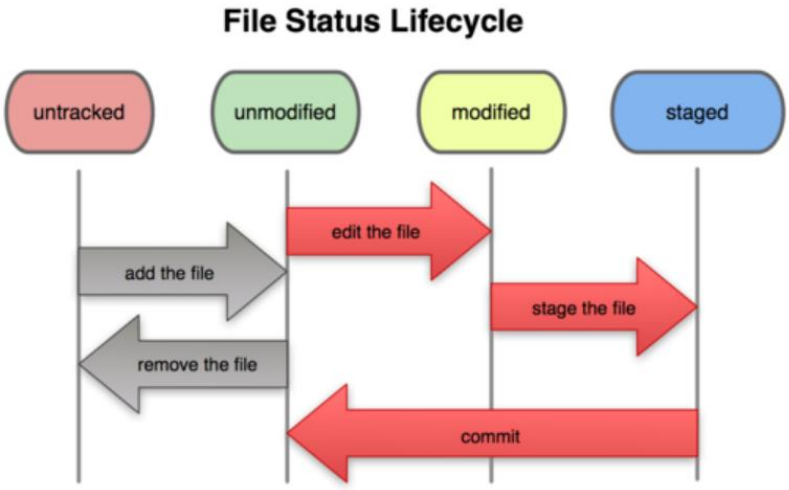
Readme.md -> here the extension md means markdown

Using git
The options to use git are
1. command line (most popular)
2. IDE/code editors (like vs code) using extensions
3. Graphical user Interface (like gitKraken)

Configuring git
```
git config --global user.name "My name"
git config --global user.email "myemail@gmail.com"
git config --list     //to see the configurations
```

Basic commands
clone - cloning a repository on our local machine
```
git clone <-some link->
```

status - displays the status of the code
```
git status
4 statuses available
      untracked
      unmodified
      modified
      staged
```



add - adds new or changed files in your working directory to the git staging area
```
git add <-file name->
git add .        //adds all the files together
```
when a file is added its status changes to staged

commit - it is the record of change
```
git commit -m "some message"      //-m means message
```

push - upload local repo content to remote repo

git push origin main
origin – main location where our repo is
main – branch name

init – used to create a new git repo
git init
cloned repos already have init, only new repositories need init
git init add origin <-link->
link we get after creating a repo on github
this link is set as origin now

git remote -v ->to verify remote(on which github repo we are currently in)
git branch ->to check branch
git branch -M <-new name-> to rename branch
git push origin main
git push -u origin main
-u ->upstream, after giving this command once, can directly push using git push, no need to add origin main again and again

To add and commit together
Git commit -am "msg"

Branch commands
git branch :to check the list of branches and the current branch
git branch -M <-branch name-> : to rename branch
git checkout <-branch name-> : to navigate
git checkout -b <-new branch name-> : to create new branch
git branch -d <-branch name-> : to delete branch
To push a new branch into github, we have to set upstream
git push --set-upstream origin feature
//here feature is the branch name

Merging code
- git diff <branch name-> : to compare commits, branches, files and more
- git merge <-branch name-> : to merge two branches
  Or create a PR
- Pull request lets you tell others about changes you have pushed to a branch in a repository on github
- To get the merged PR on my local
  - Git pull origin main
  - Used to fetch and download content from a remote repo and immediately update the local repo to match that content

Merge conflicts
An event that takes place when git is unable to automatically resolve differences in code between 2 commits

Fixing mistakes
1. **Undo staged(add) changes**
   a. git reset <-file name->
   b. git reset
2. **undo committed changes (for one commit)**
   a. git reset HEAD~1
      HEAD is a pointer that points to last/previous commit
3. **undo commit changes (for many commits)**

a. git reset <-commit hash->
    i. git log shows the commits made with each commit having a hash code . this hash code (commit hash) is used here
    ii. Using this can get back to that branch(commit), but the changes will still say staged(only moved out of commit)
b. git reset –hard <-commit hash->
    i. This will remove the changes from staging area also

Forking
A fork is a new repository that shares code and visibility setting with the original upstream repository
Fork is a rough copy

In simple terms

- Forking is done from another owner’s repository
- Forking a repo creates a copy on my local and github.
- Can create a pull request to merge my changes to the original owner’s branch(repo)
- If the owner is okay with the change, he will merge the pull request.
- Open source contributions are done using fork

Pull requests can not only be made in branches in a single repo , but can be made in 2 different repos