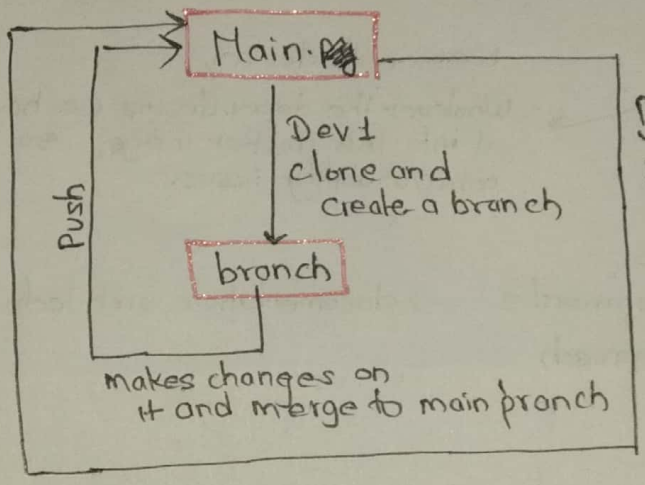
-Y. devkota. 98@gmail.com

Git & GitHub

Git: - distributed version control system that tracks changes in any set of computer fiels, usually used for coordinating work among programmers collaboratively.

Suppose, there is a stable code in main.py. Suppose we have next sprint with 10 stories and each stories is dedicated to each developer.

Then,



Let's do practical

* Create a directory of any name, say GitTest

  `mkdir GitTest`

* Go to that directory and run

  `code .`

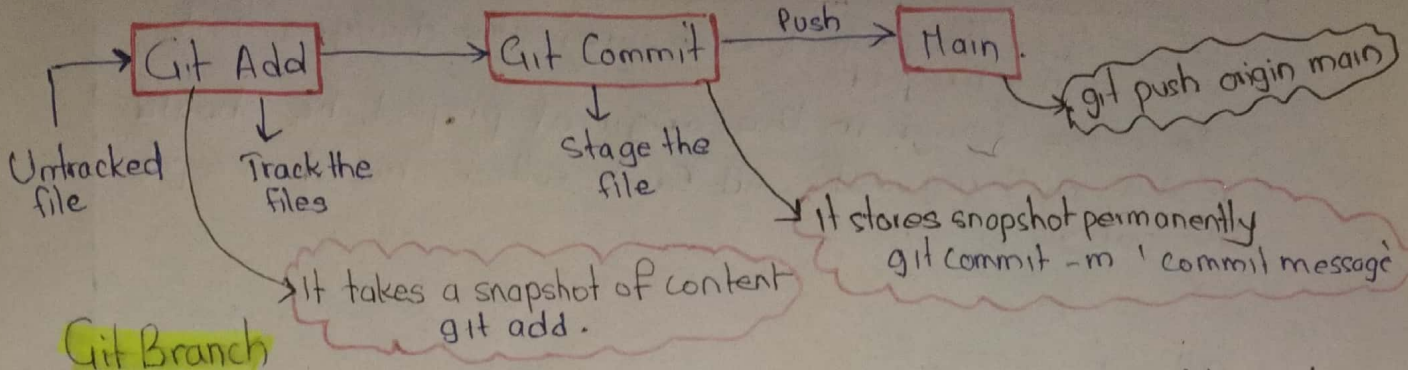
  It will open VS code. Then open terminal & do some configuration.

  `'git config --global user.email 'youremail@gmail''`

  `'git config --global user.name 'your name''`

Then, initialize the git

`'git init'`



Git Branch

When you are working as a developer, you won't push to the main branch. It is the most stable branch. Instead you will create your own branch.

Say, you are creating branch name test1.

`git branch test1` → It will have copy of main branch

To delete the branch:
git branch -d test1

`git checkout test1`
→ This way you can enter to the test1 branch. You can even push it into the test1 but the changes of it won't be visible to the main ~~pg~~ branch

`git merge test1`
It will merge the code of both main & ~~dest~~ branch.

In this way, you will be able to work in a team.

Git logs

Git logs will make you all commit history available.

`git log`

Fork Operation for Collaboration:

Fork operation enables the collaboration. First the first author should add the second author in collaboration. After accepting the invitation, he can fork the project and can make independent copy of the project.

Then, clone the project, and can make changes and push to the project.
To make change in the original project, he should make pull request, and first author should accept the pull request.

Other basic commands

(a) To clone a repo:
```
git clone repo-url
```

(b) To bring the changes from remote repo to local
```
git pull
```

(c) Adding multiple file to the stagging
```
git add file1, file2
```