

Starting with Git

Git is a version control system that helps you to keep track of all your codes and its respective changes

configuring git on Ubuntu

by using command

```
~sudo apt install git
```

Starting with working

firstly by assigning a user name and an email address to start with the working

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

```
git config --global
```

```
user.name "Your Name"
```

you have to assign the user when working from the prompt so that it detects who is working

- in your root directory create a directory for your work then open it
- then use the command

```
git clone https://github.com/user_name/repo_name
```

- open up your repo on which you wanna work
- now you can make changes to your repo such as creating, editing, updating etc

uploading on your github

All the work up till now has been done on your local repo. So, to upload on your git hub repo

- check your git status by the command

```
git status
```

- then add your file with the command

```
git add <file_name>
```

- now commit your file

```
git commit -m "your comment"
```

-m this flag is used for commenting so you may remember why or what you did on this file/ repo

now your file has been committed

- Now you have to push in your master repo for it use the command

```
git push origin master/main
```

use either master or main cause its the same. Depends entirely on your preference

your file has been pushed on your git hub

To delete a file u

if you made a mistake and have to delete your repo or file you can use the command

```
rm -rf <file_name>
```

Branches and merging

for creating branches the command used is

```
git branch <branch_name>
```

now your new and separate branch has been created now start working on it

you can follow the same way for commit add and push

```
mrX@mrX-console:~/Git_Workspace/first_testing$ git log
commit 485767cd3f2f54f3e424a0aa1946203a34847cd9 (HEAD -> main, origin/second_branch, origin/main, origin/HEAD, second_branch)
Author: Muhammad Isbah <muhammadisbah@yahoo.com>
Date: Thu Aug 8 13:28:10 2024 +0500

    second branch

commit fddc5a9947e0546345068f56cac081774548f87e (origin/first_branch, first_branch)
Author: Muhammad Isbah <muhammadisbah@yahoo.com>
Date: Thu Aug 8 13:18:58 2024 +0500

    firstBranch

commit d33bad8a268776586a975aff56fcc6b8d4c1640a
Author: Muhammad Isbah <muhammadisbah@yahoo.com>
Date: Thu Aug 8 11:50:18 2024 +0500

    First Python Code

commit 3e6c67369619a4d7f4b7e4fb8883b534add9e2af
Author: MotivatedChild <165640234+MotivatedChild@users.noreply.github.com>
:...skipping...
commit 485767cd3f2f54f3e424a0aa1946203a34847cd9 (HEAD -> main, origin/second_branch, origin/main, origin/HEAD, second_branch)
Author: Muhammad Isbah <muhammadisbah@yahoo.com>
Date: Thu Aug 8 13:28:10 2024 +0500
```

merging

for merging you use the command

git merge <branch name>

```
commit 485767cd3f2f54f3e424a0aa1946203a34847cd9 (HEAD -> main, origin/second_branch, origin/main, origin/HEAD, second_branch)
Author: Muhammad Isbah <muhammadisbah@yahoo.com>
Date: Thu Aug 8 13:28:10 2024 +0500

    second branch

commit fddc5a9947e0546345068f56cac081774548f87e (origin/first_branch, first_branch)
Author: Muhammad Isbah <muhammadisbah@yahoo.com>
Date: Thu Aug 8 13:18:58 2024 +0500

    firstBranch

commit d33bad8a268776586a975aff56fcc6b8d4c1640a
Author: Muhammad Isbah <muhammadisbah@yahoo.com>
Date: Thu Aug 8 11:50:18 2024 +0500


    First Python Code

commit 3e6c67369619a4d7f4b7e4fb8883b534add9e2af
Author: MotivatedChild <165640234+MotivatedChild@users.noreply.github.com>
Date: Thu Aug 8 11:24:42 2024 +0500

    Initial commit
~
```






Branches

Main


 **first_testing** Public

[Pin](#) [Unwatch](#) 1

[main](#) [3 Branches](#) [0 Tags](#) [Add file](#) [Code](#)

| | | |
|---|-----------------------|-------------|
|  MotivatedChild second branch | 485767c · 2 hours ago | 🕒 4 Commits |
|  README.md | Initial commit | 4 hours ago |
|  first.txt | firstBranch | 2 hours ago |
|  hello.py | First Python Code | 4 hours ago |
|  second.txt | second branch | 2 hours ago |





first branch

 **first_testing** Public


[Pin](#) [Unwatch](#) 1

[first_branch](#) [3 Branches](#) [0 Tags](#) [Add file](#) [Code](#)

This branch is 1 commit behind main . [Contribute](#)

| | | |
|---|-----------------------|-------------|
|  MotivatedChild firstBranch | fddc5a9 · 2 hours ago | 🕒 3 Commits |
|  README.md | Initial commit | 4 hours ago |
|  first.txt | firstBranch | 2 hours ago |
|  hello.py | First Python Code | 4 hours ago |






second branch


first_testing
Public
Pin
Unwatch
1

second_branch
3 Branches
0 Tags

Add file
Code

This branch is up to date with main .
Contribute

| | | | |
|---|-------------------|-----------------------|-----------|
|  MotivatedChild | second branch | 485767c · 2 hours ago | 4 Commits |
|  README.md | Initial commit | 4 hours ago | |
|  first.txt | firstBranch | 2 hours ago | |
|  hello.py | First Python Code | 4 hours ago | |
|  second.txt | second branch | 2 hours ago | |

Rebase

rebase is different from merging . In rebase the commits goes into a linear sequence

To rebase a branch. Follow the steps:

- first checkout in the branch you want t rebase
git checkout <branch name>
- then use the command to rebasethe branch in main
git rebase main

```

mrX@mrX-console:~/Git_Workspace/first_testing$ git checkout first_branch
Switched to branch 'first_branch'
mrX@mrX-console:~/Git_Workspace/first_testing$ git fetch origin
mrX@mrX-console:~/Git_Workspace/first_testing$ git rebase main
Successfully rebased and updated refs/heads/first_branch.
mrX@mrX-console:~/Git_Workspace/first_testing$ git branch -a
* first_branch
  main
  second_branch
  remotes/origin/HEAD -> origin/main
  remotes/origin/first_branch
  remotes/origin/main
  remotes/origin/second_branch
mrX@mrX-console:~/Git_Workspace/first_testing$ 

```

Git pull

git pull retrieves the last changes made in the remote repo and merges them in your local branch you can use this action by using command

```
git pull <remote>
```

```
git pull origin
```

Git fetch

git fetch retrieves the last changes made in the remote repo but does not merges them with or in your local branch it allows you to check and verify before finally merging

it can be used by the command

```
git fetch <remote >
```

Git Revert

- use the command to find the commit in the log

```
git log --oneline
```

- now use the git command with the hash code
git revert 3e6c673
- now the text editor will be open for you to edit the commit message for the new revert commit.

```
git <command> [<revision>...] -- [<title>...]
mrx@mrx-console:~/Git_Workspace/first_testing$ git log --oneline
485767c (HEAD -> main, origin/second_branch, origin/main, origin/HI
fddc5a9 (origin/first_branch) firstBranch
d33bad8 First Python Code
3e6c673 Initial commit
mrx@mrx-console:~/Git_Workspace/first_testing$ git revert ^C
mrx@mrx-console:~/Git_Workspace/first_testing$ ^C
mrx@mrx-console:~/Git_Workspace/first_testing$ ^C
mrx@mrx-console:~/Git_Workspace/first_testing$ git revert 3e6c673
[main da15b31] Revert "Initial commit"
1 file changed, 2 deletions(-)
delete mode 100644 README.md
```

Cherry Pick

git cherry-pick allows you to apply the changes introduced by some commits from one branch onto another branch.

to use cherry pick follow the steps:

- first open the log of the branch from which you want to cherry pick

git log <branch name>--oneline

```
mrx@mrx-console:~/Git_Workspace/first_testing$ git log second_branch --oneline
1984961 (origin/second_branch, second_branch) cherry-pick
485767c (origin/main, origin/HEAD, first_branch) second branch
fddc5a9 (origin/first_branch) firstBranch
d33bad8 First Python Code
3e6c673 Initial commit
```

- now give command for cherry pick with the hash code of the commit
git cherry-pick 1984961

```
mrx@mrx-console:~/Git_Workspace/first_testing$ git cherry-pick 1984961
[main 3685829] cherry-pick
Date: Thu Aug 8 17:28:18 2024 +0500
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 cherry.txt
create mode 100644 test.txt
```

git flow

git flow is the model for working in git

from the main branch the respective teams the different branches are made for the use.

for example

the developing team makes out a separate branch for development and the for features different branches are taken out so that the newly edited addition does not affect the main stable branch