1. **Git config –global user.name “Lupyawgyi”**
2. **Git config –global user.email** [**myominhtun65@gmail.com**](mailto:myominhtun65@gmail.com)

Must be advertise For usinging computer user and user email address with these command

1. Create new folder and change git repo. We can change git repository folder type with below per command

Need to go inside this folder with command line and then use **git init** (we can check show hidden file folder option). The git software store version in this hidden folder.

1. We can check the changes with **git status** command
2. Would like to store the first version with **git add . (or) git add “file name”.**

At this time, we should be use **git status** command again for check the added file name.

1. After that we can use **git commit –m “first commit”**
2. And we create new file or new code in this folder, we need to save this changes again. At this time, we use **git status** commend and check changes folder again.
3. And then we use **git add .** again
4. After that we can use **git commit –m “second commit”** again
5. We check the version log with **git log** command or **git log --pretty=oneline. At this time we can see code version detail. We have two version First commit and second commit**
6. If we would like to go to go any version, we use the **git checkout first commit code number**
7. IF we would like to go the last version, we use the **git checkout master**
8. If we would like to tag the version number custom with **git tag 0.1.0**

**Branch**

1. We would like to test demo environment on main code, we need to divide with branch. This mean clone the main code for we can test on the demo environment.
2. **git branch** command for check how many branch in main code
3. **git branch dev (custom name)** for create a new demo branch
4. **git checkout dev** change **master** branch to **dev** branch.
5. For testing crate or edit some file in this folder and add to git and create new version on **dev** branch with

**git add . , git commit –m “dev commit”.** This is only change on the demo environment.

1. We would like to add the changes to **master branch** from **dev branch changes.** First we need to go to master branch with **git checkout master.**
2. And then **git merge dev** for all changes dev branch to master.

Gid Clone

1. If we would like to clone project folder, firstly, outside the projector from command line. And then

**git clone ./git project2 ,** git🡺 mean current project folder location and second project2 mean the new project clone folder name and location.

1. The clone repo folder can update from the origin repo changes. If some code is changed in the origin repo, we can get the update changes with **git pull origin master.**

(origin mean the origin repo folder location and master mean branch name.) If we would like to know original folder location, we use the **git remote show origin**

1. Note: We can pull only origin project to clone project but we cannot give the changes to origin from the clone repo project.

**Git central Bare  
1)** Create central project folder and go to inside this folder and then write down **git init --bare** command in command line. If we check the central folder, we will see git processing files.

**Note: we will not write down any code in this central folder. The central folder only stores the code version.**

**2)** Now we can clone the central folder and we can use push and pull command.

3) If we would like to clone the central folder, we need to go outside this folder with command line. And then write git clone ./central app1 and git clone ./app2

4) If something updated in app1 folder, the app1 folder make version commit and send to the central folder. The app2 folder can pull this updated data from central folder.

**In app1 folder command box**

git add .

git commit –m “version number”

git push origin master

**In app2 folder command box**

git pull origin master

At this time the app2 get the updated data from central folder.

1. If we change the code on same file and same line, Error will show when we pull from central. At this time, we need to edit the true code on this file and create new commit and push to server again.