* **Git** is a revision control system, a tool to manage your source code history. **GitHub** is a hosting service for **Git** repositories. So they are not the same thing: **Git** is the tool, **GitHub** is the service for projects that use **Git**.
* Git clone link\_project // copy le projet dans un répertoire
* Git status // donne le statut des fichiers existes dans le repository
* Gid add nameFile // ajouter le nameFile dans le staging area
* Git reset head nameFile // retirer le nameFile depuis le staging area
* Git commit -m’ ’message’ // mettre les fichiers dans le local repository
* Git remote -v // donne le nom/lien de remote repository
* Git Push repoName branchName // mettre les fichiers dans remote Repository(server)
* Git pull repoName branchName // importer le remote repo changements dans le local repo
* Ssh-keygen -t rsa -b 4096 -c ’sounhalazoun@gmail.com ’ // crée un public key
* Git init // initialise un repo vide dans le dossier courant, on doit créer le repo dans le remote repo
* Touch file.txt // créer un fichier file
* Git branch nameBranch // créer un branch nameBrach
* Git branch //liste le nom des branchs existes
* Git checkout nameBranch // pointer sur le branch nameBranch
* Git chekout -b nameBranch // créer le branch nameBranch et pointer sur luis en même temps.
* Git branch -d nameBranch // supprimer le branch nameBranch s’il n’y a pas des modifications
* Git branch -D nameBranch // forcer la suppression du branch nameBranch même s’i y a des modifications.
* Git branch -m newBranchName // renommer le branch courant à newBranchName
* Git merge branchName // merger entre le branch courant et le branch branchName
* [jerry@CentOS project]$ git config --global user.name "Jerry Mouse" // setting username
* [jerry@CentOS project]$ git config --global user.email [jerry@tutorialspoint.com](mailto:jerry@tutorialspoint.com) // setting email Id
* [jerry@CentOS ~]$ git config –list // To verify your Git settings of the local repository.
* [root@CentOS ~]# group add dev // # add new group
* [root@CentOS ~]# user add -G devs -d /home/gituser -m -s /bin/bash gituser // # add new user
* [root@CentOS ~]# passwd gituser // # change password
* [tom@CentOS tom\_repo]$ git log // checks the log message by executing the git log command.
* [tom@CentOS tom\_repo]$ git remote add origin gituser@git.server.com:project.git
* [tom@CentOS tom\_repo]$ git push origin master
* [jerry@CentOS jerry\_repo]$ git clone [gituser@git.server.com:project.git](mailto:gituser@git.server.com:project.git)
* [jerry@CentOS project]$ git show cbe1249b140dad24b2c35b15cc7e26a6f02d2277 // to view the commit details. The git show command takes **SHA-1** commit ID as a parameter.
* [jerry@CentOS project]$ git diff // reviews the changes
* [jerry@CentOS project]$ git commit --amend -m 'Changed return type of my\_strlen to size\_t' // commits the new changes with -- amend operation
* [jerry@CentOS project]$ git stash // To push a new stash onto your stack
* [jerry@CentOS project]$ git stash list // view a list of stashed changes
* [jerry@CentOS project]$ git stash pop // to remove the changes from the stack and place them in the current working directory.
* [tom@CentOS project]$ git mv string.c src/ // Tom decides to move the source code into **src** directory
* [jerry@CentOS src]$ git mv string.c string\_operations.c // rename file string.c to string\_operations.c
* [tom@CentOS src]$ git rm string\_operations // Tom decides to remove the string\_operations file from the repository
* [jerry@CentOS src]$ git checkout string\_operations.c // to revert the contents of the string\_operations.c file.
* [tom@CentOS src]$ rm string\_operations.c // Tom deletes the file from the local repository
* [tom@CentOS src]$ git checkout string\_operations.c // to obtain the deleted file from the local repository
* [tom@CentOS src]$ git add string\_operations.c
* [tom@CentOS src]$ git checkout HEAD -- string\_operations.c // Remove Changes from Staging Area
* [jerry@CentOS project]$ git log -2 // the two latest commit ID
* [jerry@CentOS project]$ git reset --soft HEAD~ // reset the HEAD pointer back by one position
* [tom@CentOS project]$ git tag -a 'Release\_1\_0' -m 'Tagged basic string operation code' HEAD // . Tom provides a tag name with -a option and provides a tag message with –m option.
* [jerry@CentOS src]$ git tag -l // Jerry can view all the available tags
* [jerry@CentOS src]$ git show Release\_1\_0 // to view more details about tag.
* [tom@CentOS project]$ git tag -d Release\_1\_0 // to delete tags from the local repository.
* [tom@CentOS project]$ git push origin :Release\_1\_0 // to delete tags from the remote repository.

# Order of commands

## Create Repository From Existing Project

* $ Git init // initialise un repo vide dans le dossier courant, et on doit créer le repo dans le remote repo.
* Git add file.text
* Git commit -m ’Deploy my Project from Pc to GitHub’
* $ git remote add origin <https://github.com/MohammedEnnekhli/FootScoreApp.git>
* Git pull origin master // it’s necessary, if we create README and/or LICENSE file
* Git push -u origin master

## Update Operation

Git is not allowing Jerry to push his changes. Because Git identified that remote repository and Jerry’s local repository are not in sync. Because of this, he can lose the history of the project. To avoid this mess, Git failed this operation. Now, Jerry has to first update the local repository(git pull) and only thereafter, he can push his own changes (git push origin master ).

# Errors

## remote: Permission to MohammedEnnekhli/FootScoreApp.git denied to sounhalazoun.

 I had this problem too but managed to solve it, the error is that ur computer has saved a git username and password so if you shift to another account the error 403 will appear. Below is the solution  
For Windows you can find the keys here:

control panel > user accounts > credential manager > Windows credentials > Generic credentials

Next remove the Github keys.

## error: failed to push some refs to 'https://github.com/MohammedEnnekhli/FootScoreApp.git'

Try this git command

git push origin master --force  
or short of force -f

git push origin master -f