# Git and GitHub

* GitHub:

1. A version control system fort eh management of changes to documents, computer programs, large websites and other collection of information.
2. These changes are called as ‘Versions’. We can revert the changes once commited.
3. By this a backup is available. Central server to local server.
4. Four most popular version control tools: a) **git** b) subversion c) CVS and d) Mercurial.
5. **Git: an open source version control tool.**

* Git:

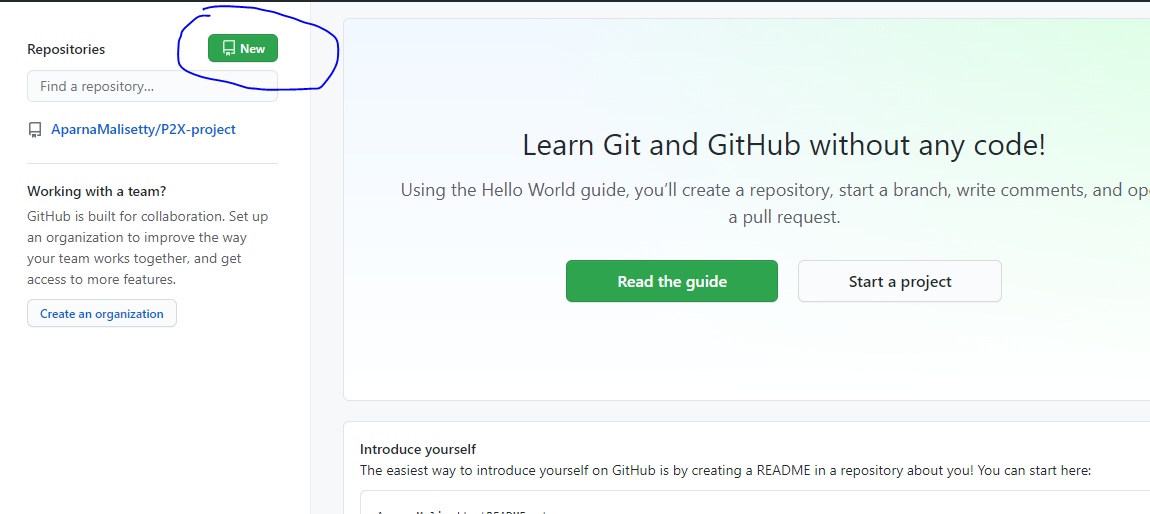
1. Repository: a directory or storage space where your projects can live. It can be local to a folder on your computer or it can be a storage space in the GitHub or another online host. We can keep code files, text files, image files etc.. inside a repository.

Two types of repository: central repository (GitHub) and the locxal repository(Git).

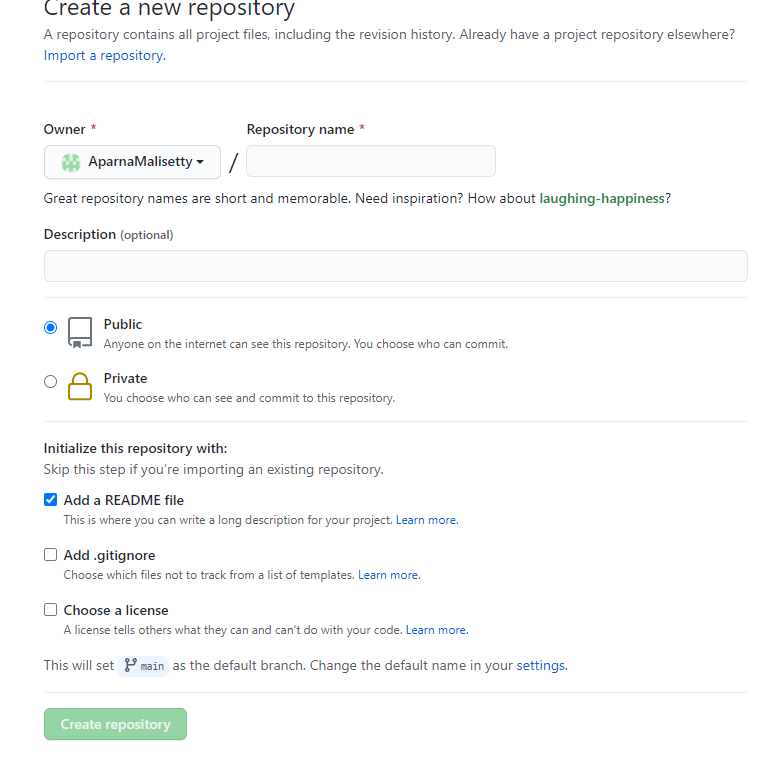
1. Central repository locates on the remote server and consists of .git repository folder and meant for team to share and exchange data.
2. Local repository: on local machine and resides as a .git folder inside your projects root. Only admins of the machine can work on this repository.

* GitHub: set up an account in <https://github.com/>. Create a repository

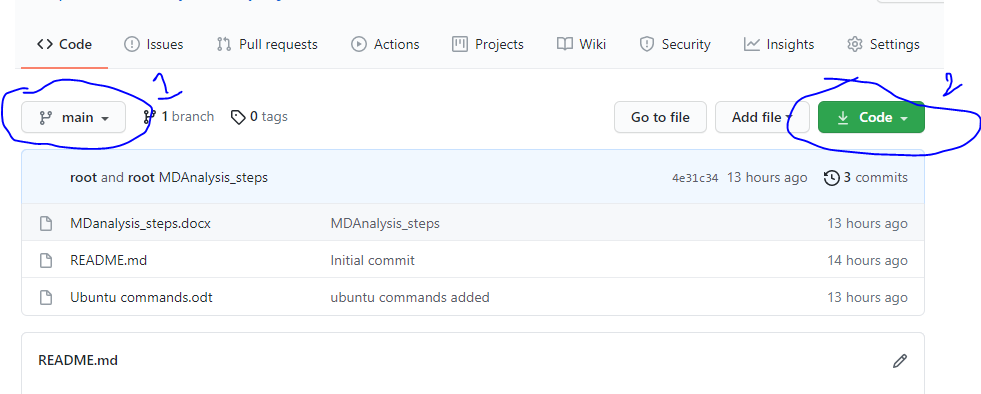
1. Creating repositories:



1. While creating repository: to have a private repository you have to pay or we can secure the folder by using ssh key(later will be mentioned). And check the Add a README file. This is optional but this will be the first file and will generally include the information about the project.



1. Here main(1 in pic) is the default branch and some places it will be master. Code(2 in pic) once you click that there will beth elink for https or ssh in order to be able to connect with our local server.



* Git installation: if you have Ubuntu or linux then git is already installed. We can check the version of it by using the command

git --version

* Git operations and commands:

1. Set the directory where you want your folder to be in the Ubuntu terminal.
2. To in initialise git, you can use either of these two commands

git init #this creates a new local repository and if you use this then you have to link the central repository to this by using these commands. git remote add origin <https://github.com/AparnaMalisetty/P2X-project> #after this command git pull origin main #main is the branch name someplaces it will be master as default branch.

Or

git clone <https://github.com/AparnaMalisetty/P2X-project> #the link available under the code(2 in above pic).

1. Making changes in the local repository:

* If you created a new file in the local folder and it will not be saved in the local repository it will be saved only after commiting it.

1. git status #we can see the untracked files
2. git add edu1.word #edu1.word is the name of the new file. To add multiple files use this command. Git add –A
3. if you use command git status agin we can see that the new files are added and are ready to commit.
4. git commit –m ‘added file and commited’ . #this –m stands for the message. This is important. To commit multiple files the command to be used is : git commit –a –m ‘all files commited’
5. git log #shows the time and commits.
6. Parllel development: a new branch can be made which points to the main branch. This is used if you don’t want to disturb the main branch.

* git branch firstbranch #firstbranch is the name of the new branch.
* git checkout firstbrarnch #this is to be in the firstbranch or else we will be in main branch and now whatever fiels you add will be in the new branch and not on the main branch
* git add fisrtbranchfile.txt #this is the new file added in the firstbranch and tehn add and commit to be able to connect to the firstbranch repository.
* git add firstbranchfile.txt
* git commit firstbranchfile.txt
* ls #lists out all the files in the first branch.
* git checkout main #now we are in the main branch rather than the first branch
* ls #if you list out the files of the main branch. The new file firstbranchfile.txt is not present in this main branch since it belongs to the firstbranch.

1. Merging: to combine the work of different branches together. to add files of the firstbranch to the main branch cab ne done by using merging.

* git merge firstbranch #remember that you have to checkout to main before using this command. Now the fiels of firstbranch area dded to the main branch.
* Ls #now you can see the files of the new branch in the main branch.

Remember the first branch is still a separate branch. If you checkout there and add new files that will not be present in the main.

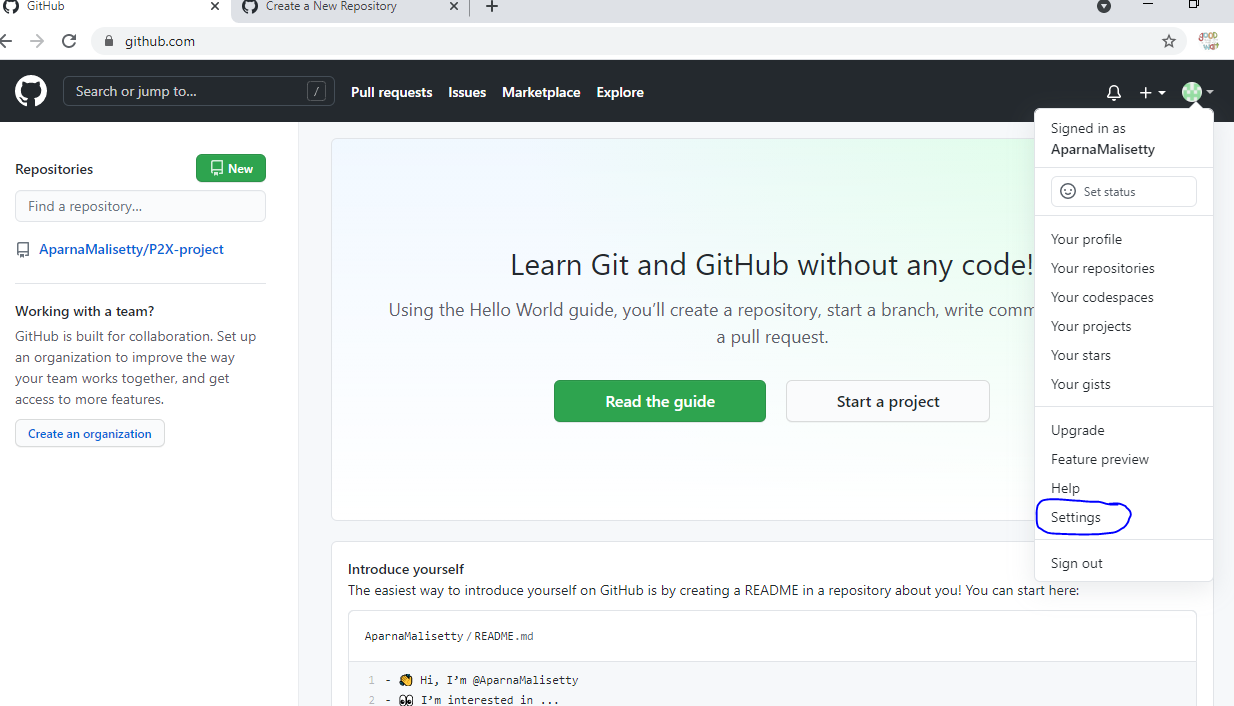
1. Pushing the changes to the central repository(github):

* If we directly push now. Since it is a public repository it will be accessed by everyone and can edit it. To avoid that use ssh key
* Generate a public and private key in the terminal Ubuntu: add the public key to the github

ssh-keygen #to generate the key

cat /c/users/Aparna/ #this is to see what is written in the public key. After cat write the location of the public key which ends with .pub. after executing it copy the result which starts with ssh-rsa.

Paste the code in the github: settings->click on SSH and GPG keys -> new SSH key -> provide a name in title box and paste the copied ssh key and click add ssh key.



* In terminal :

ssh –T [git@github.com](mailto:git@github.com)

* Pushing the changes. You can either push it by creating a new branch firstbranch or to the main branch without creating a separate new branch.

1. Pushing the changes of first branch

* git checkout firstbranch
* git push origin firstbranch #we can see firstbranch in github website. To see the files of it select the firstbranch on the dropdown of the github

1. pushing the changes directly to main branch without separate branch

* git checkout master
* git push origin master
* Revert back to the original version:
* git add edu4.txt #a new file added to the repository
* git commit –m ‘reverting to be done’
* now change the content of the file edu4.text and commit again: git commit –a –m ‘revert2’
* git log #you will see the time and commit. And copy the first 8 digits of the commit code depending on which version you want to revert back.
* git checkout 54hhhh555 edu4.txt #the no. here is the commit code copied nad edu4.txt is the file whose version of it we don’t want to add.
* cat edu4.txt #if you see what is written in that the you will see it will have the content of the previous commit