# Experiment -1 : Version Control using GIT

# Aim : Is to perform version control using git & git-Hub Synchronization commands namely add,clone,push/pull through branch.

# Procedure / Theory :

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

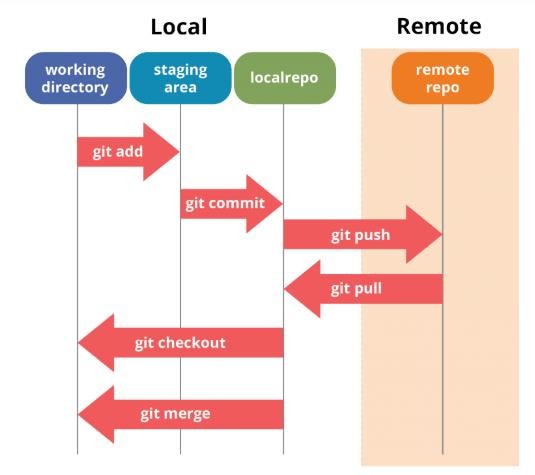
Some of the basic operations in Git are:

1. Initialize
2. Add
3. Commit
4. Pull
5. Push

Some advanced Git operations are:

1. Branching
2. Merging
3. Rebasing

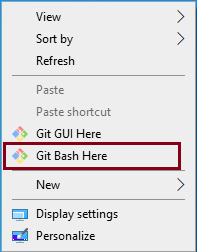
The following diagram depict the all supported operations in GIT



# Installation of GIT

* 1. In windows, download GIT from <https://git-scm.com/> and perform the straightforward installation.
  2. In Ubuntu, install GIT using $sudo apt install git, Confirm the version after installation $git --version

Once installation is done, open the terminal in Ubuntu and perform the following steps or in windows Right click and select Git bash here.



To perform version control, let us create a directory dvcs (Distributed version control system) and change directory to dvcs.

*$ mkdir git-dvcs*

*$ cd git-dvcs/*

Now check the user information using

*$ git config –global*

As there are no users defined, let us define it using following two commands

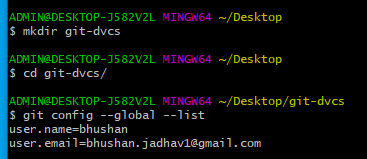
*$ git config --global user.name "dbit"*

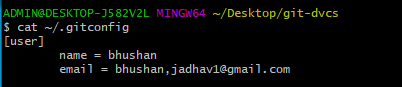
*$ git config --global user.email "dbit@dbit.in*[*"*](mailto:bhushan%2Cjadhav1@gmail.com)

Now, check the list of users

*$ git config --global --list*

user.name=dbit

[user.email=dbit.@dbit.in](mailto:user.email%3Dbhushan.jadhav1@gmail.com)

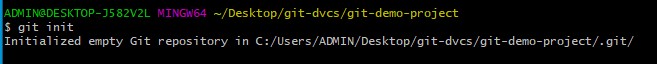


Let us create a repository for version control named ”git-demo-project”

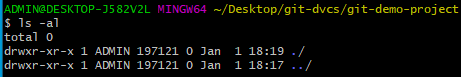
*$ mkdir git-demo-project*

*$ cd git-demo-project/*

Now, initialize the repository using following command

*$ git init*

If you have existing repository, then simply delete .git file and reinitialize it.

*$ rm -rf .git/*

*$ git ini*t

Initialized empty Git repository in C:/Users/ADMIN/Desktop/git-dvcs/git-demo- project/.git/

Now, let us add some files inside our repository “git-demo-project”

To add files in index and staging area, add command is used along with dot (. Dot means current directory)

*$ git add .*

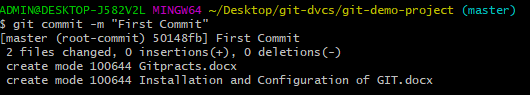
Index and staging area

To check the status of repository, use

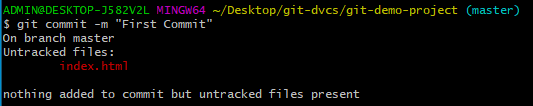
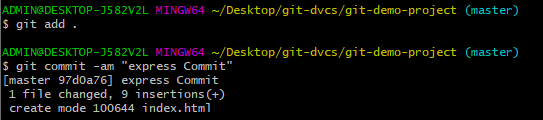
*$ git status*

Which will show you some untrack files, so untracks files can be tracked using commit command.

Now, let us commit the changes

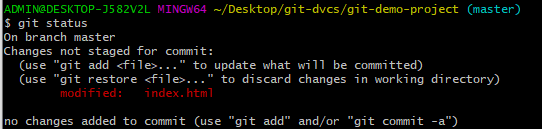
*$ git commit -m "First Commit" (#here -m for message)*

# Add index.html in our directory

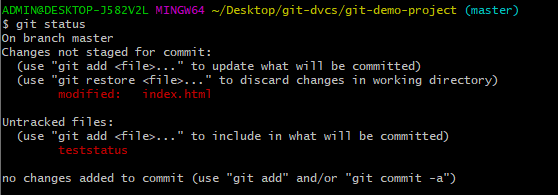
****

*$ git add .*

*$ git commit -am "express Commit" (#Here -a used for express commit)*

*$ nano index.html*

*$ touch teststatus*

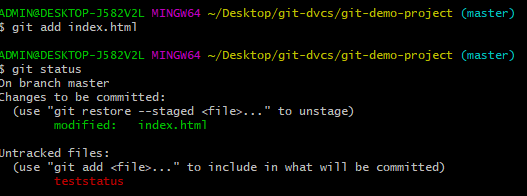




# Changes are Discarded by checkout

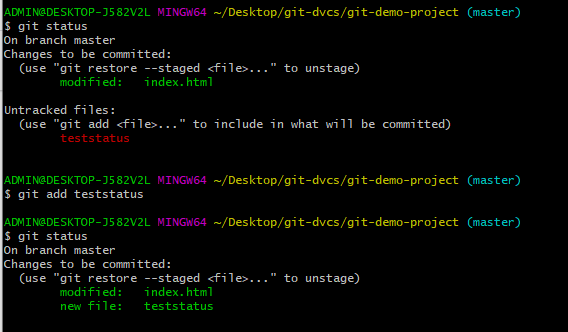
(use "git add <file>..." to update what will be committed)

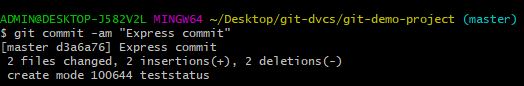
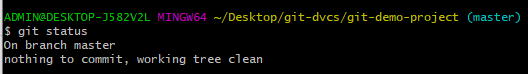
(use "git restore <file>..." to discard changes in working directory)



*$ git add index.html*

*$ git add teststatus*

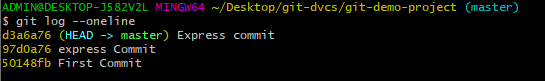


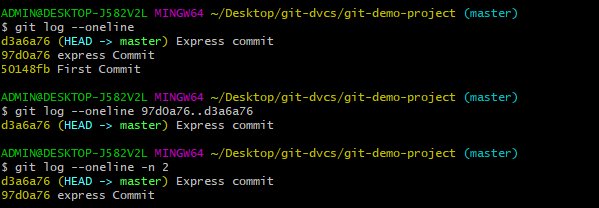
$ git commit -am "Express commit"

# History of Commits

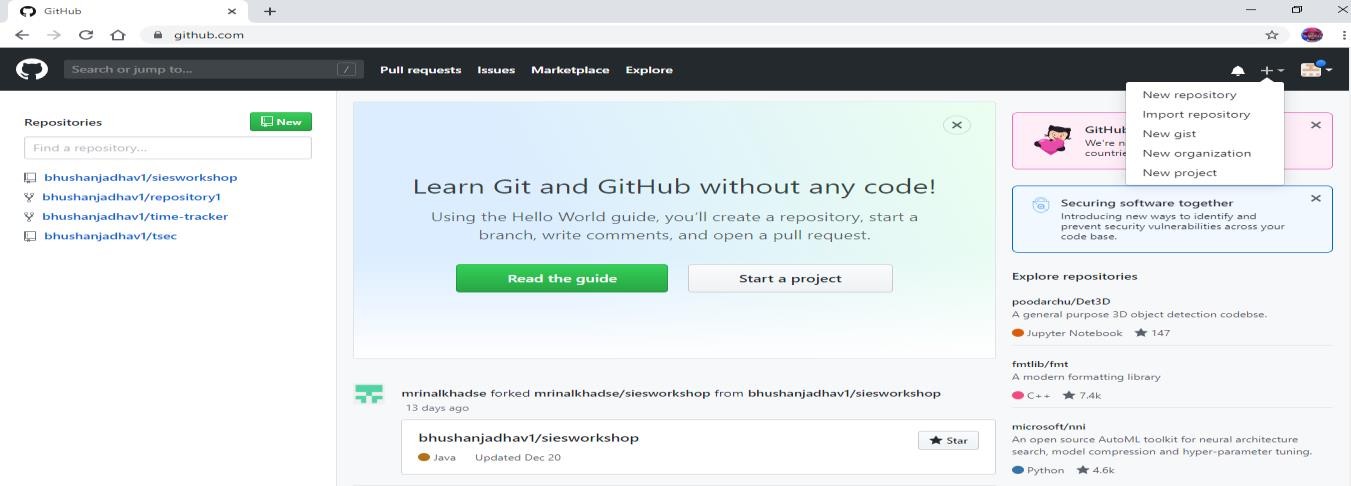
$ git log



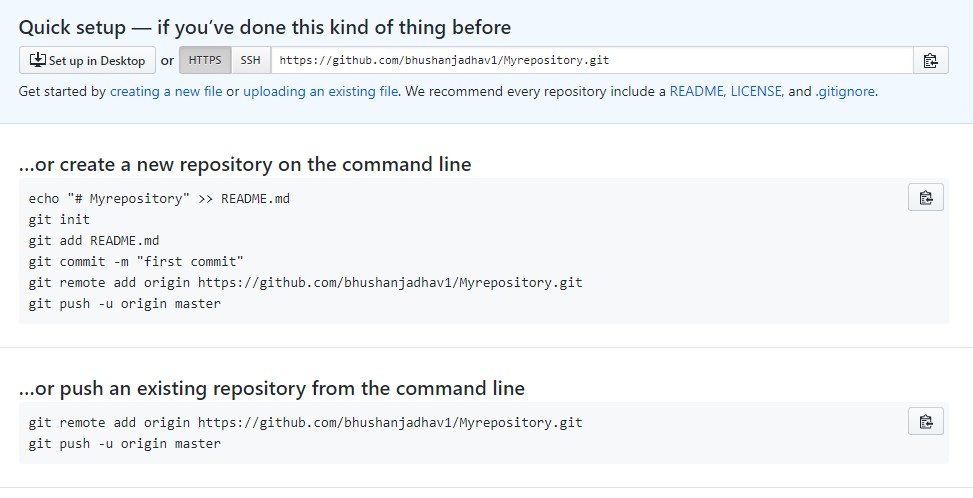




Now Create a Repository on github.com. Open github.com→ create an account→After login Select New repository from the menu.

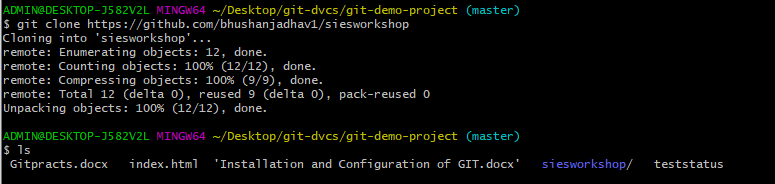


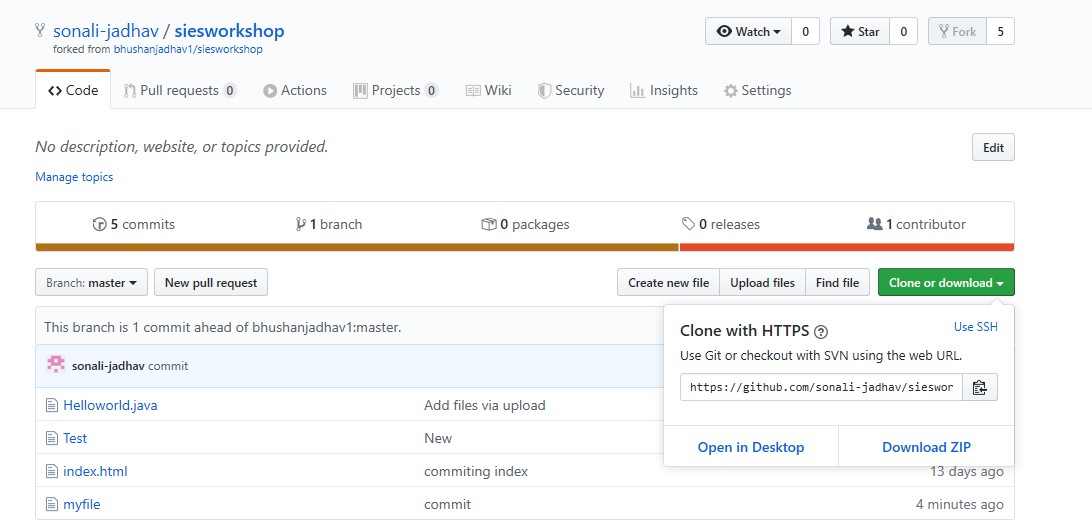
Now Specify a Name to repository and select public option followed by create repository



Now fork the repository (Sharing with other users who wants to contribute).

Login with another account→Copy and Paste URL of repository→then just click on fork to clone to others account.



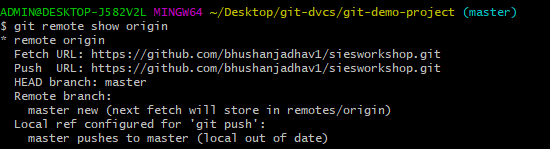


**Pull and Push Processes**

Push → Push changes to Web repository Pull → Pull changes to Local repository

# 1) Push command to remote reference origin master

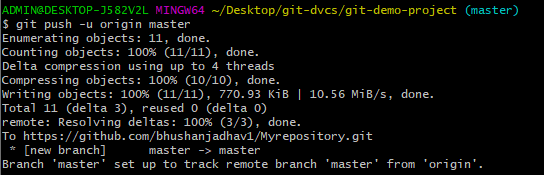
$ git remote add origin <https://github.com/bhushanjadhav1/siesworkshop.git>

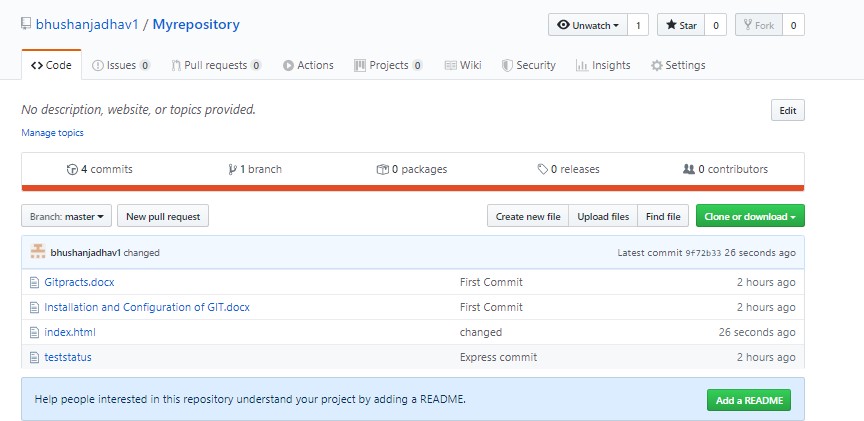
$ git remote show origin

$ git remote add origin https://github.com/bhushanjadhav1/Myrepository.git fatal: remote origin already exists.

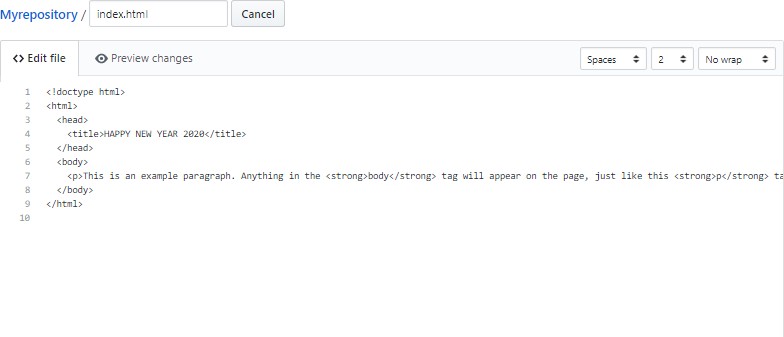
$ git remote rm origin

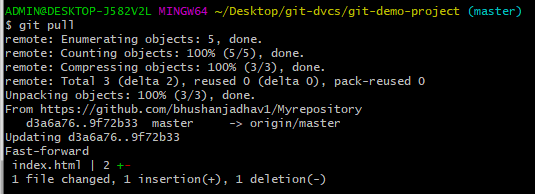
$ git push -u origin master

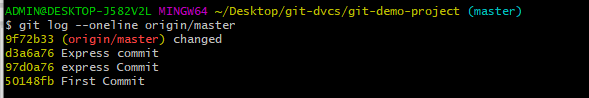
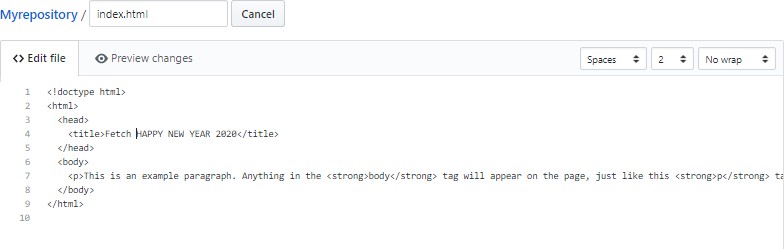




# Pull Changes

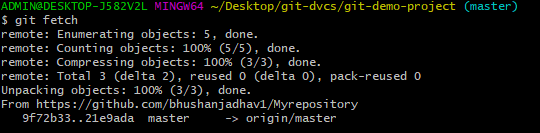
****

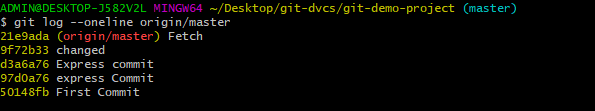
$ git pull

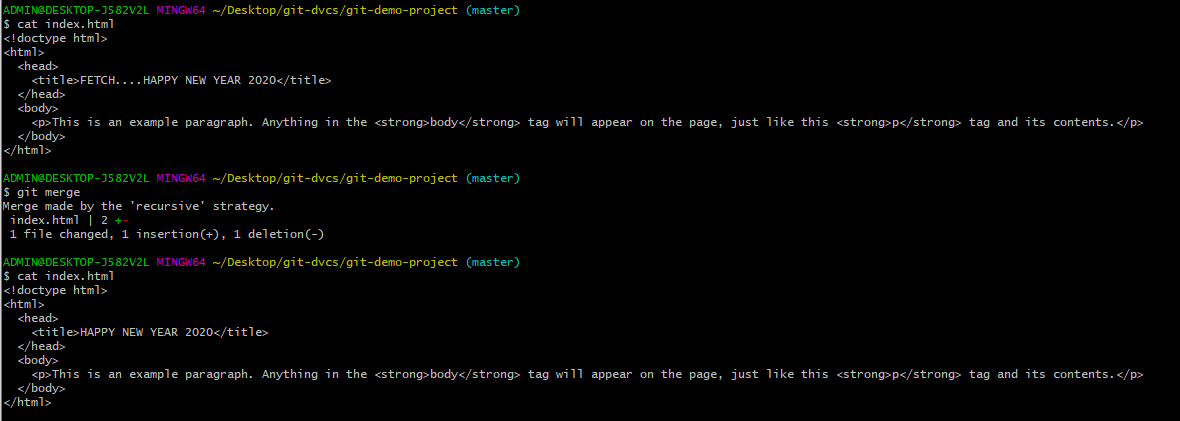
**Fetch**

$ git fetch

Here fetch will not show you like updated changes file as like push. So use merge command to merge the changes.





$ git merge origin/master