**Name:** Sakshi Jitendra Patil

**Roll No:** 05

**PRN:** 181051029

**EXPERIMENT NO. 01**

**Aim:** To perform Version Control using GIT

**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

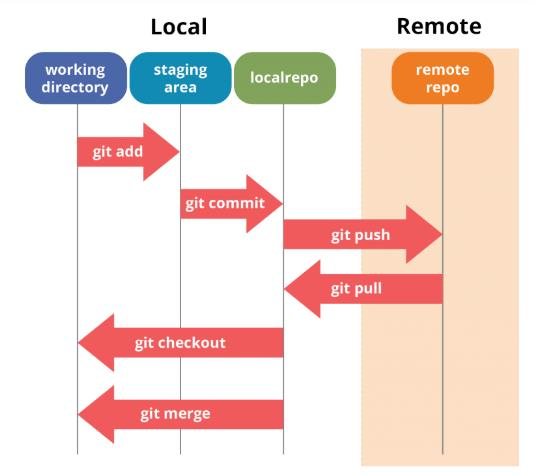
Some advanced Git operations are:

1. Branching

2. Merging

3. Rebasing

The following diagram depict the all supported operations in GIT



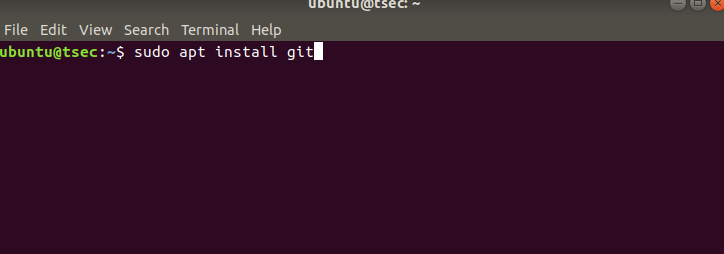
**Installation of GIT**

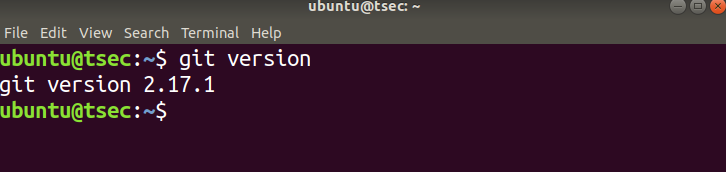
1) In windows, download GIT from https://git-scm.com/ and perform the

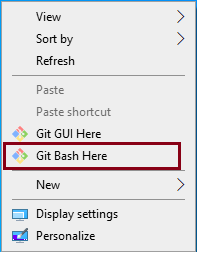
straightforward installation.

2) In Ubuntu, install GIT using $sudo apt install git, Confirm the version after

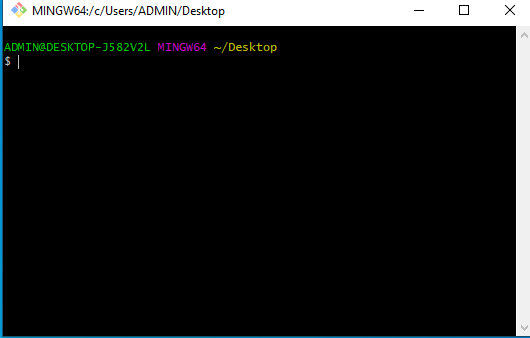
installation using command $git –version

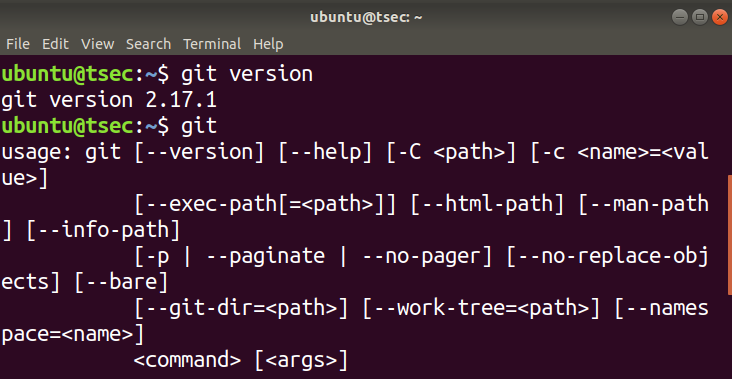


Once installation is done, open the terminal in Ubuntu and perform the following steps or in windows Right click and select Git bash here



The output of GIT Bash in windows and GIT shell in Ubuntu is shown below



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 "bhushan"

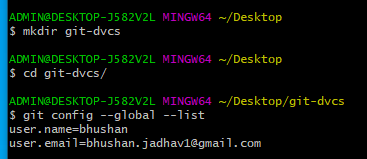
$ git config --global user.email "bhushan,jadhav1@gmail.com"

Now, check the list of users

$ git config --global --list

user.name=bhushan

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



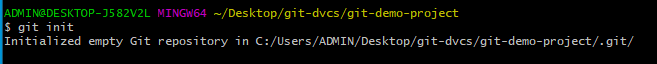
Let us create a repository for version control named ”git-demo-project”

$ mkdir git-demo-project

$ cd git-demo-project/

Now, initialize the repository using following command

$ git init

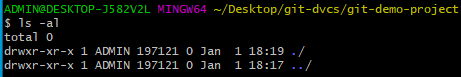


The output of above command shown below which adds .git hidden directory in current repository.

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

$ rm -rf .git/





git init

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 the repository by create or copy some doc,html,image files inside

current directory to see index and staging area. The add command is used along with

dot (. Dot means current directory) for adding files in current repository i.e. making

them in staging mode. They are untracked until we commit them.

$ git add .

Index and staging area

To check the status of repository, use

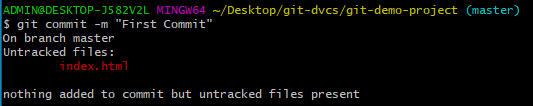
$ git status

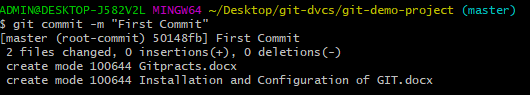
Which will show you some untrack files, so untracks files can be tracked using commit

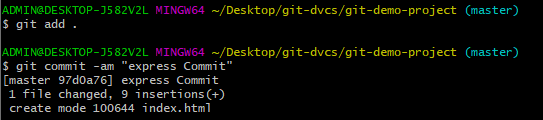
command.

Now, let us commit the changes

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

Add index.html in our directory

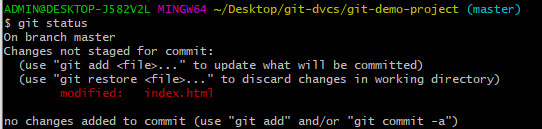




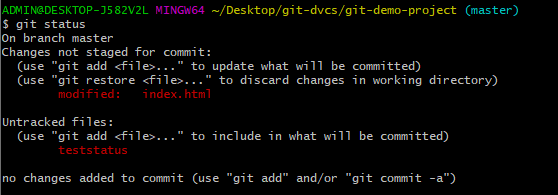
$ git add .

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

$ nano index.html



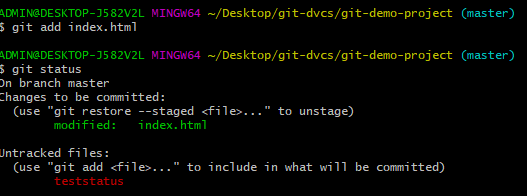
$ touch teststatus



Changes are Discarded by checkout

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

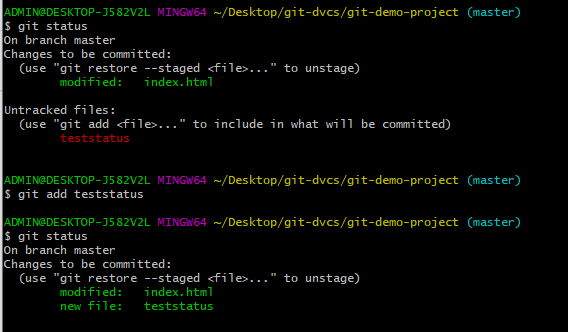
(use "git restore <file>..." to discard changes in working directory)



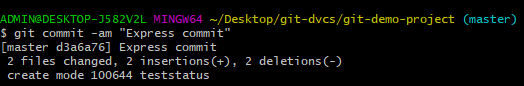
$ git

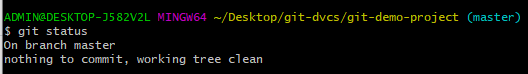
add index.html

$ git add teststatus



$ git commit -am "Express commit"



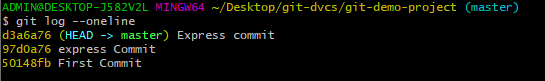


Now let us see history of commits. The log command is used for seeing the commit history.

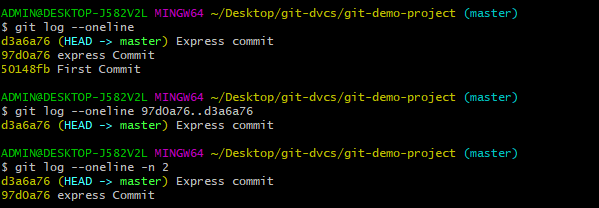
$ git log

To see all the operation in oneline use the –-oneline option in log command

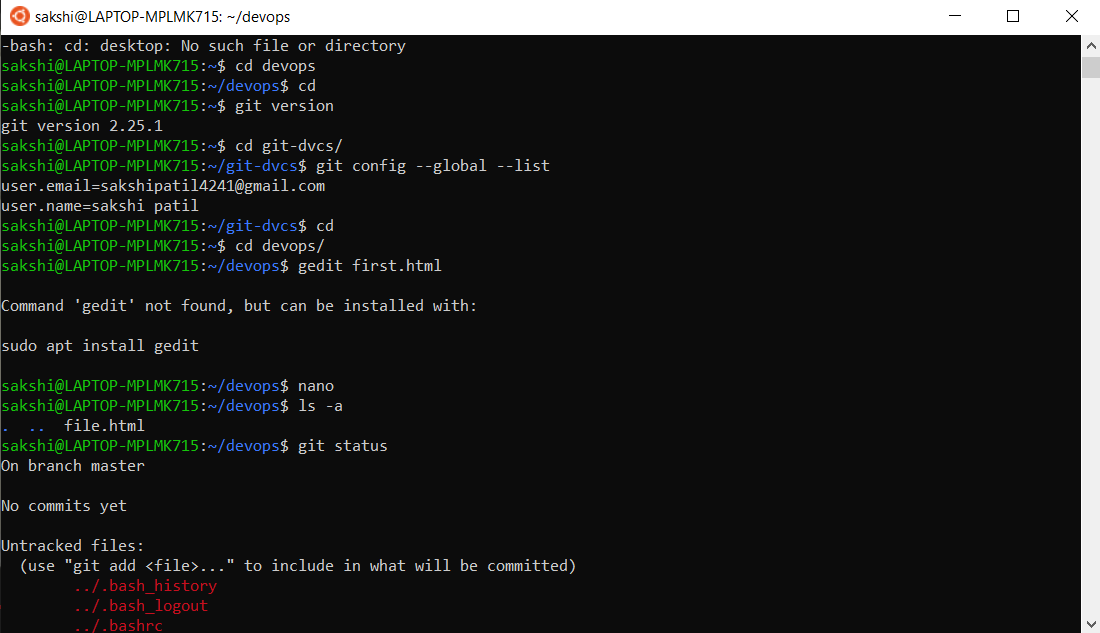
oneline option for particular file in log command

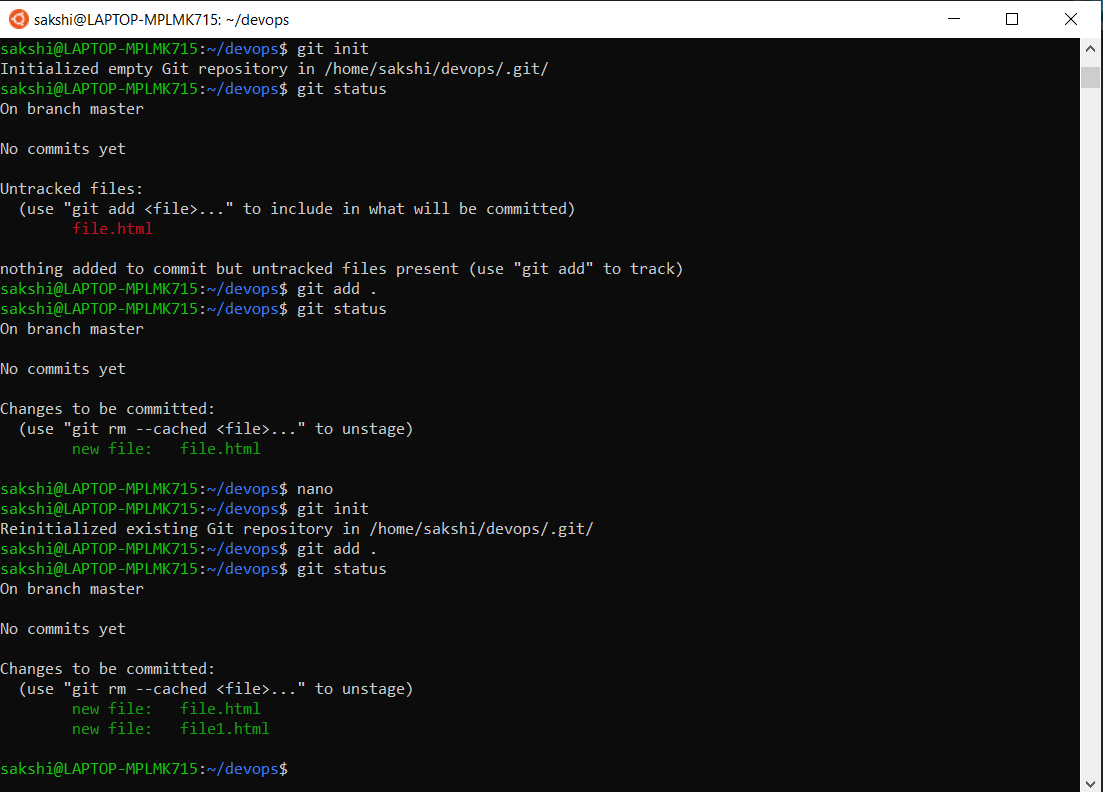


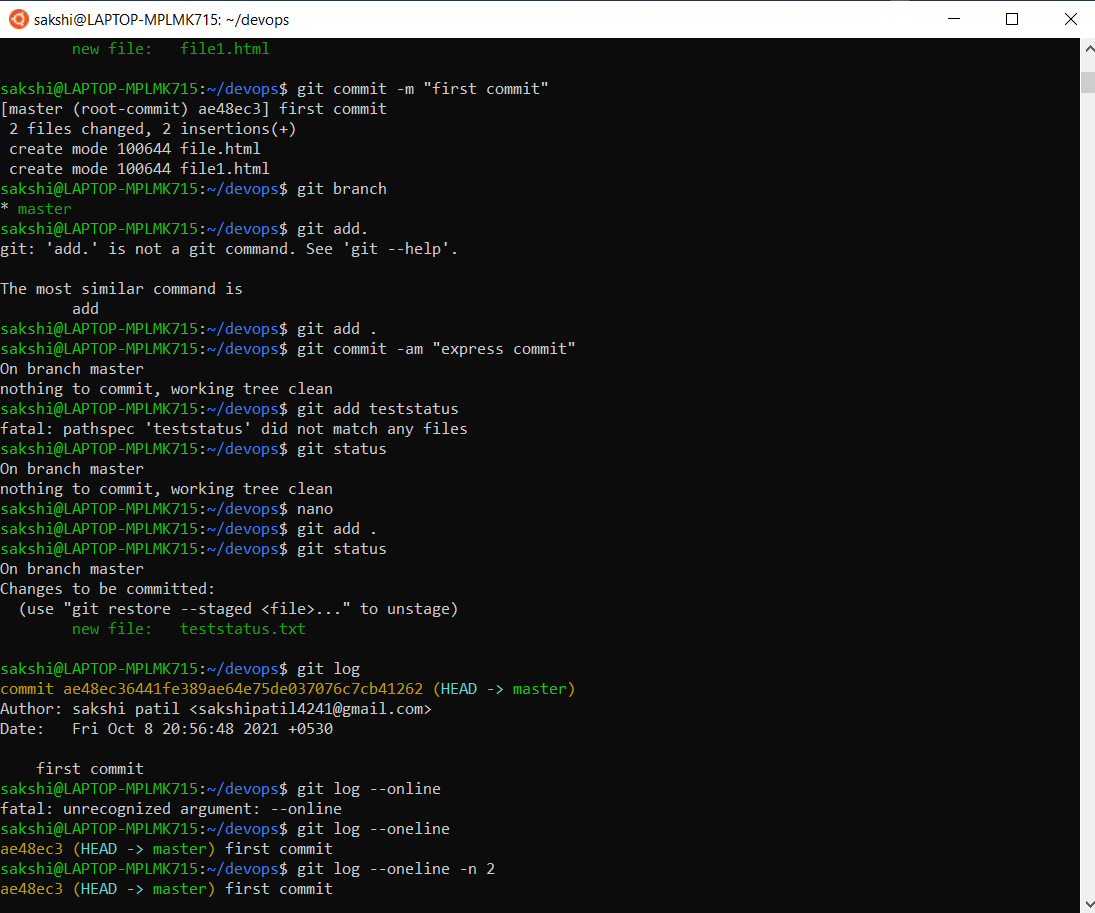




**Output:**

****

****

****