Git Tutorial

Download: https://git-scm.com/download/win

1. git config –global //means this user

```
git config --global user.name "Abhishek Kavrani" // set username
git config --global user.email abhishekkavrani27@gmail.com // set email address

to check everything is set
git config –list

// Help Purpose

git help or git help commit
```

2. Create Repository

```
pwd : present working directory
cd ~ : change directory to home
cd .. : change directory by one step backward
ls : list all the files present in the directory

Technical name of project is the repository
git init //To start a git project

secret command to check everything(Hidden + Secure) present in the folder
ls –la

git status // to check status of git we created in a repository

Note: first we have to add on which file of repository we wish to have eye on it.
      git add “file_X.ext” //check git status now

      now how git says changes to be committed ??
      git commit –m “message” //commit

To add multiple no of files of one type
git add ‘*.ext’

or to add all the files in the directory
git add . // to add all the files
```

Note: git log is a kind of journal that remembers all the changes we've committed so far, in the order we committed them.

```
      git log // Check the log
```

3. Suppose if you are working on a group project and there are three user’s developer, programmer, and assistant manager.

So if you wish to see change by developer then use
git log –-author=”developer”

Boring....
Real use of git status here: it basically compares your commit section to the present working directory.

Three step process:
Working copy > staging area > repository
➔ → → push → → →
← ← ← pull ← ← ←

Here working copy is the copy which is not track by the git it is still has something which is left undone
So to have a track on it we need to use add before commit therefore the adding stage is our staging area
After commit it is in repository where each element has its own history

4. How to edit, delete, move, rename Files

Edit section:
Status of change in first will be shown on git in terms of modified after status use git log you will see the difference.
After commit the previous state of that doc is deleted or preserved don’t know yet I think it depend on the type of work or whether it is shared with others or not as of now master is responsible for the change he accepted the change in the file.
Check this video:
https://www.youtube.com/watch?v=SVirDyRXbA&list=PL6gx4Cwl9DGAKWClAD_iKpNC0bGHxGhcx&index=8

We use git diff for the analysis of repository before staging area.
git diff only shows the difference between the working copy and staging area in the repository.
So to compare files from repository and staging area we use
git diff –-staged

delete section:
→git rm third.txt
Now this change will be shown on git status and in similar way we need to commit it.

Rename section: git use baby steps to learn how file is renamed
After renaming the file git consider it as the previous file is deleted and the new one is created but after adding the new file in staging area git understand what we have done. Now in git status it will show file name is renamed.

Move section: the simple way to perform above work is use
git mv first.txt first_new.txt
or to move to different folder, folder should exists.
git mv first.txt path_from_pwd_folder_name/first_new.txt

Note: Commit and adding together
When you are uploading final copy then you can try this

git commit –am “Commit directly to repository”

Working with a website

1. Suppose a new developer messed up our work so he changed something because of which our website crashed
   Now how to recover our work???
   Here git will replace the present screwed file with working file.

git checkout –- index.html
/// take something from repository and replace it with working copy.

2. Consider a situation when you wish to change two things in a file and by chance you changed one and you push that file from working stage to staging area.

Now you have two choices one is to modify it once again and fill git’s history book so to keep a good track we will reset things as they were at initial stage

git reset HEAD index.html
/// unstaged or staging area to working copy

3. New version to old version

Because new version have so many bugs, suppose three programmers

```
First                           //working
First > second                  //working
First > second > third(fake)    //crash
```

Now we want second one, how delete the third one not possible git’s work is to maintain the timeline of work you never know which code can change the world,

the solution of this problem is to revert the order or initial once again second one

```
              First > second > third(fake) > Second   //working
Commit no:    123…    456…         789…        456…
```

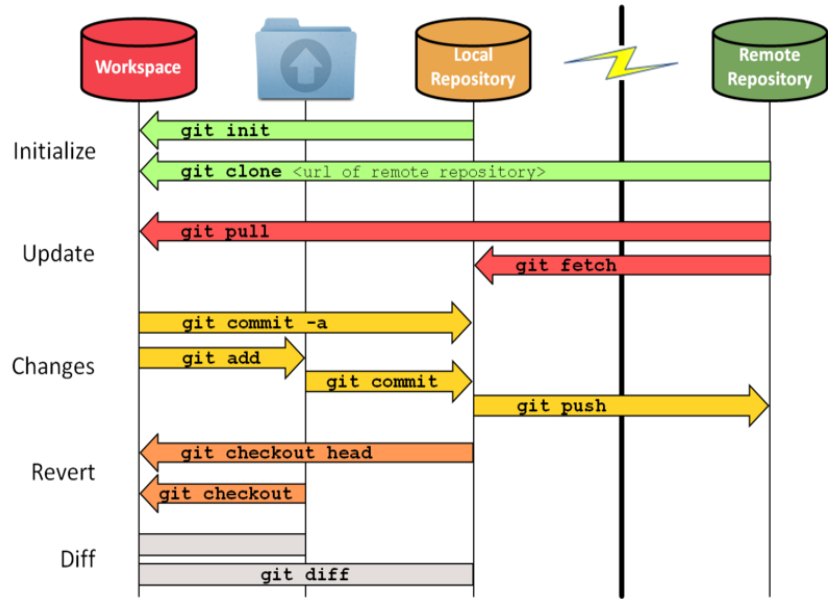
use git log /// list of every update

```
git checkout 456 – index.html
git commit –am “second #working copy is committed”
```

GitHub

It allow you to work on public project, people can suggest you or they can give a working copy which works well with better result.

#GitHub account




```
$ git remote add nick_name_to_url http://url_github_new_repository

$ git remote
nick_name_to_url

$ git push –u nick_name_to_url master
// -u means all the files present in the local repository will be uploaded on remote
       repository
```