Git account:

shamhub

interOP@123

------------------------------------------------------------

Once the Git bash shell prompt is ready,

Run the below command, which tracks about who has changed files inside of Database.

$ git config –global user.name “Sham”

$ git config --global user.email sham.ibmgs@gmail.com

$ git config --global core.editor “vim”

$ git config --list

--global

For writing options: write to global ~/.gitconfig file rather than the repository .git/config

-------------------------------------------------------------------------------------------------------------------------------

If you want to track files in particular **Working** directory, Go to that directory,

Create an empty Git repository for that directory(sub-directories) or reinitialize an existing one, by saying

$ git init

Add file contents to the index(Staging)

$ git add \*.java

This command updates the index using the current content found in the working tree, to prepare the content staged for the next commit.

The "index" holds a snapshot of the content of the working tree, and it is this snapshot that is taken as the contents of the next commit. Thus after making any changes to the working directory, and before running the commit command, you must use the add command to add any new or modified files to the index.

This command can be performed multiple times before a commit. It only adds the content of the specified file(s) at the time the add command is run; if you want subsequent changes included in the next commit, then you must run git add again to add the new content to the index.

----------------------------------------------------------------------------------------------------------------------------

----------------------------------------------------------------------------------------------------------------------------------

If you want to specifically ignore certain files, you need to create a file by name .gitignore in your working directory and take all file types you are working with.

This site <https://github.com/github/gitignore> will have info about all of the common that should be ignored.

gitignore - Specifies intentionally untracked files to ignore

A gitignore file specifies intentionally untracked files that git should ignore. Files already tracked by git are not affected

---------------------------------------------------------------------------------------------------------------------------------

$ git diff

Show changes between commits, commit and working tree, etc

Show changes between the working tree and the index or a tree, changes between the index and a tree, changes between two trees, changes between two blob objects, or changes between two files on disk.

$ git diff

This form is to view the changes you made relative to the index (staging area for the next commit). In other words, the differences are what youcould tell Git to further add to the index but you still haven’t. You can stage these changes by using [git-add(1)](file:///C:\\Program%20Files%20(x86)\\Git\\doc\\git\\html\\git-add.html).

$ git diff --cached

This form is to view the changes you staged for the next commit relative to the named <commit>. Typically you would want comparison with the latest commit, so if you do not give <commit>, it defaults to HEAD. If HEAD does not exist (e.g. unborn branches) and <commit> is not given, it shows all staged changes. --staged is a synonym of --cached.

---------------------------------------------------------------------------------------------------------------------------------

$ git --commit

Record changes to the repository

Stores the current contents of the index in a new commit along with a log message from the user describing the changes.

The content to be added can be specified in several ways:

1. by using git add to incrementally "add" changes to the index before using the commit command (Note: even modified files must be "added");
2. by using git rm to remove files from the working tree and the index, again before using the commit command;
3. by listing files as arguments to the commit command, in which case the commit will ignore changes staged in the index, and instead record the current content of the listed files (which must already be known to Git);
4. by using the -a switch with the commit command to automatically "add" changes from all known files (i.e. all files that are already listed in the index) and to automatically "rm" files in the index that have been removed from the working tree, and then perform the actual commit;
5. by using the --interactive or --patch switches with the commit command to decide one by one which files or hunks should be part of the commit, before finalizing the operation. See the “Interactive Mode” section of [git-add(1)](file:///C:\\Program%20Files%20(x86)\\Git\\doc\\git\\html\\git-add.html) to learn how to operate these modes.

After you run this commit command, You get “**.git/COMMIT\_EDITMSG**” file opened. At this point you add your actual commit message at the top of the document. Any lines starting with '#' will be ignored - Git will put the output of the git status command in there for you as a reminder of what you have modified and staged.

--------------------------------------------------------------------------------------------------------------------------------------

If you want to skip Staging & Commit, then type,

$ git commit –a –m ‘Change commit’

-a

--all

Tell the command to automatically stage files that have been modified and deleted, but new files you have not told Git about are not affected.

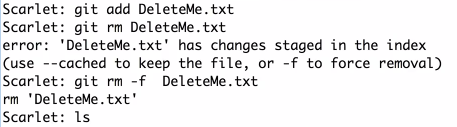
-m <msg>

--message=<msg>

Use the given <msg> as the commit message. If multiple -m options are given, their values are concatenated as separate paragraphs.

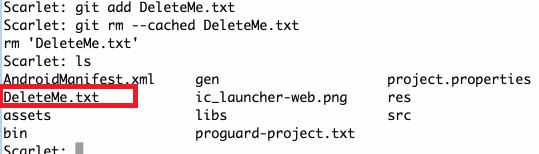
-------------------------------------------------------------------------------------------------------------------------------------

Removing file from working directory



---------------------------------------------------------------------------------------------------------------------------------

Removing file from staging area



--------------------------------------------------------------------------------------------------------------------------------

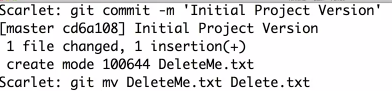
$ git-status

Show the working tree status

Displays paths that have differences between the index file and the current HEAD commit, paths that have differences between the working tree and the index file, and paths in the working tree that are not tracked by Git (and are not ignored by [gitignore(5)](file:///C:\\Program%20Files%20(x86)\\Git\\doc\\git\\html\\gitignore.html)). The first are what you would commit by running git commit; the second and third are what you could commit by running git add before running git commit.

---------------------------------------------------------------------------------------

How to rename the files?



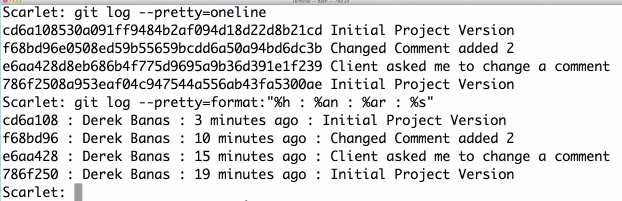
---------------------------------------------------------------------------------------------------------------------------------

-----------------------------------------------------------------------------------------------------------------------------------------

How logs work inside Git?

git --log

Show commit logs



Logs of last 2 commits



Logs with abbreviated Stat



Logs with Commit in last 1 week



Logs of commit since particular date



Logs of changes made by specific author



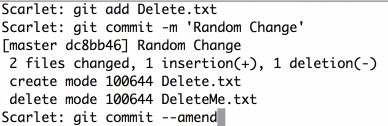
Logs of commits before certain date



------------------------------------------------------------------------------------------------------------------------------

---------------------------------------------------------------------------------------------------------------------

How to undo the commit?



-----------------------------------------------------------------------------------------------------------------------

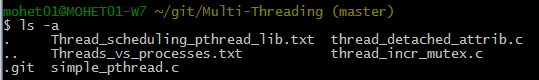
How to undo Staging?



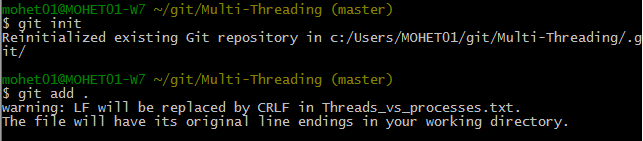
-----------------------------------------------------------------------------------------------------------------------

To push the files into repository.

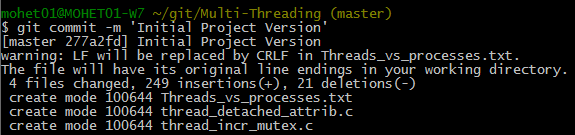
1. On local machine, First go to the directory from where the files have to be pushed to remote repository.



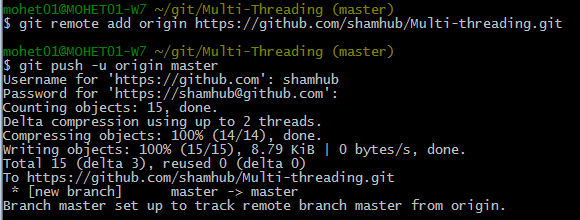
1. Run these two steps.



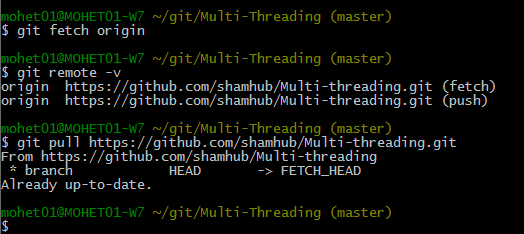
1. Commit the files to local repository.



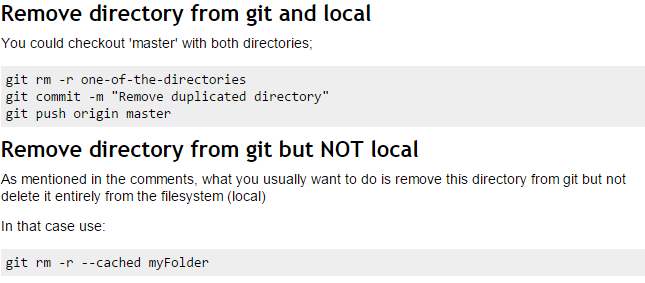
1. Run the below 2 commands to push to remote repository.



1. Pull the files from Repository like this..



1. How to remove a directory in my GitHub repository?



1. What is git remote add ...

As you probably know, git is a distributed version control system. Most operations are done locally. To communicate with the outside world, git uses what are called remotes. These are repositories other than the one on your local disk which you can push your changes into (so that other people can see them) or pull from (so that you can get others changes). The command git remote add origin git@github.com:peter/first\_app.gitcreates a new remote called origin located at git@github.com:peter/first\_app.git. Once you do this, in your push commands, you can push to origin instead of typing out the whole URL.

1. What is git push origin master

This is a command that says "push the commits in the local branch named master to the remote named origin". Once this is executed, all the stuff that you last synchronised with origin will be sent to the remote repository and other people will be able to see them there.

1. What's the difference between git pull and git fetch in Git?

In the simplest terms, git pull does a git fetch followed by a git merge.

You can do a git fetch at any time to update your remote-tracking branches under refs/remotes/<remote>/. This operation never changes any of your own local branches under refs/heads, and is safe to do without changing your working copy. I have even heard of people running git fetch periodically in a cron job in the background (although I wouldn't recommend doing this).

A git pull is what you would do to bring a local branch up-to-date with its remote version, while also updating your other remote-tracking branches.

**git fetch** is the command that says "bring my local copy of the remote repository up to date."

**git pull** says "bring the changes in the remote repository where I keep my own code."