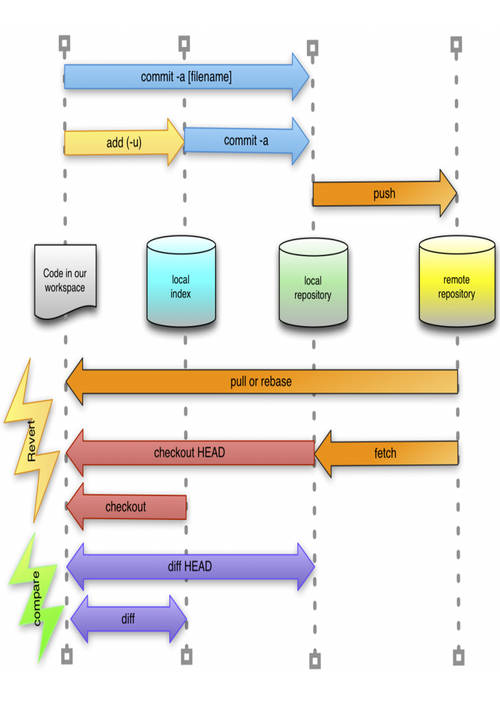
GIT BASH





1. Create Directory for repo by: **mkdir FolderName**
2. Go to inside of created directory by: **cd FolderName**
3. Now initial folder as repository by: **git init**

Now select username as globally by: -

**git config –global user.name “Name”**

Then select email as globally by: -

**git config –global user.email “Mail@Name”**

IF WE WANT TO CREATE LOCAL USER-NAME & MAIL FOR SINGLE REPO THEN: -

For user-name: -

**git config user.name “name”**

For mail-name: -

**git config user.email “email@name”**

For check present username & Mail in git by: -

**git config --list**

Now in list you can see “user.name=Name” & “user.email=Mail”

Now We can make some new file for work by: -

**touch fileNameWithExtention**

We can rescan our repo to see onstage and upstage file by: -

**git status**

We can see UNTRACKED FILE which are new create & new file by: -

**filename**

We can stage our untracked file by: -

**git add fulfillName**

Again rescan our status by: -

**git status**

We can stage all file in a directory by: -

**git add --all**

or **git add .**

We need to commit for take stage file on Local Repo by: -

**git commit**

Now open a new bash with inserting message for commit by: -

Press: I button

Then: type something

Press: Esc button

Complete commit by press: :x

Last press: enter button

Now we can see 2 files changed,

We can commit in a single line by: -

**git commit -m “type message”**

To see all commit file by: -

**git log**

Or we can see committed file shortly: -

**git log --online**

If we change any file which are committed than

we again need to take -> onstage -> then commit.

If we want to go back to backdated file from update then we need to go back to previous version, then we need

first to see all of our update with commit Id by: -

**git log --online**

then we select our previous commit’s ID and use by: -

**git checkout 658251** (id of commit)

For work new commit we need to switch (ID) to (master)

Here we can see all of our created commit by: -

**git checkout master**

We can see any change of our previous commit’s file with change text by: -

**git diff**

We check git diff before make change as new commit.

We can see the difference of two commit by there ID by: -

**git show 03df531 1Fds546** (online ID)

here we see difference from ---a/filename to +++b/filename AND we see result on “-ans which added” then “+ans which added last” to “+ last update result”

We can see difference before stage area by: -

**git diff**

But If we wanna see difference on stage area than we need extra command: -

**git diff --staged**

If we just delete any file from folder this file can’t delete permanently. This file stand on other commit.

So, if we need to Delete, then we need to delete from git by: -

**Git rm filename.extention**

Now we can’t see filename.extention on git local repo Directory.

But our file is inside of other staged, we check by: -

**git status**

Here we see our delete isn’t staged. Git tracked our file.

For remove permanently from stage, we need: -

**git reset HEAD filename.extention**

Now commit our change by: -

**Git commit -m “delete filename permanently”**

**Local Repo to Remote Repo: -**

**N**ow we see how to connect local Repo to remote Repo and take change local to Remote Repo. We complete that by pushing file using CMD for that

1. Login to github from browser
2. Create Repo
3. Copy “push an existing repository” comments
4. Past comment on CMD from HTTPS panel
5. Login git on CMD pupup
6. Enter press with given command which the 2nd line of copy

Below are the commands we are going to use to upload files to GitHub with Git Bash 1st go to repo directory

**ls**

**cd**

**git init**

**git remote add origin [gitLink from 4 HTTPS]**

**git remote -v**

**git add .**

**git commit -m "First commit"**

**git push origin master**

**W**e remove Connection by: -

1. Search on windows “credential” & go “Credential Manager”
2. Go to windows Credentials
3. Click right side of git:https://github.com
4. Click Remove & press Yes

Now we disconnect local & Remote Repo