## Checking out a remote branch in local.

**Q:**

*Currently there are 3 branches in git hub. One is master branch others are develop and release branch.*

*I go on to create another branch from develop called as develop2. Here the parent is develop branch.*

*What will happen now if I do a git pull from develop branch in local???*

*What will happen if I do a git pull from release branch in local??*

**A:**

If we directly do pull from any branch in local. It will just show us that a new branch is added in the remote however it will not create an actual branch in local itself.

We can see all the remote branches created using command git branch –a. However we cannot directly checkout this remote branch.

If we directly try to checkout from the remote using command git checkout branch\_name, then we will fall into detached head state which will basically refer to commit stage of branch and not the actual branch itself.

In order to fix this a local branch needs to be created which must track the remote branch in our case it is the develop2 branch.

Command:

git checkout --track origin/daves\_branch

--track is shorthand for git checkout -b [branch] [remotename]/[branch]

## Pulling source code in local repo.

This command will pull all the latest changes from the remote directory and merges into the local directory you are working.

Git pull command is a combination of fetch and pull.

**Q:**

*What will happen if I do a change in a file in local and then do a git pull??*

**A**

In this case we will get “Already up to date” message only. This is because when we are doing a git pull , git is checking for any changes which are done in remote repository file since the last pull. If there is any change done in remote file then git will ask us to remove the changes from local or commit it.

So we can say like git pull is checking for data drawn from remote since the last pull. It will consider the local change only when that file is changed in remote and we have a change in local as well.

This way we can confirm that the data in our local is in sync with the remote repository and we are just having our local changes in it. If want to remote all those changes then we can do a git stash.

**Q:**

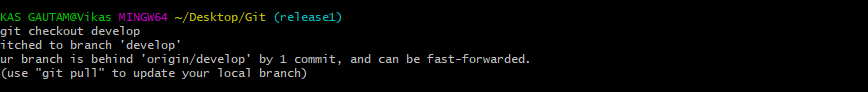
*If we do a git pull from a specific branch in local will it pull all the changes done from all the branches in repository or it will bring only the changes which are done in the branch we are currently in.*

**A:**

“Yes”. If we are doing a git pull lets say from release1 branch then it will pull only changes from this branch. The other branches data will not be pulled.

Lets say I have two branches develop and release. I do some change in remote in both the branches. Later on if I try to pull from the release branch, git pull command will fetch the results and merge with the data in local in release branch only.

Now if I try to switch to develop branch and pull from there then, a message will be displayed to user saying that develop branch is ahead of our local develop branch by XX commits. This basically means that changes have not got pulled from develop branch however a message is shown to user so that he can pull from remote branch and carry on with work.



Do a git pull from develop branch to fix this.

Command:

git pull

**Q:**

*What will happen when a branch which is tracking the origin branch is deleted from the local??*

**A:**

When we are deleting any branch from local tracking remote branch that branch is removed from the local only.

In order to remote that branch from remote we have to issue command git push origin --delete origin/remote\_branch.

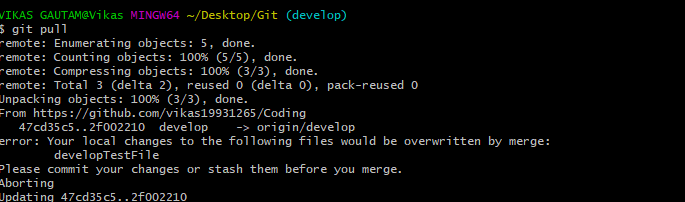
This basically means that local and remote branches are independent of one another. Removing a branch in local does not affect in remote and vice versa.

**Q:**

What will happen if the remote branch is updated and we have a local change and we try to pull??

**A:**

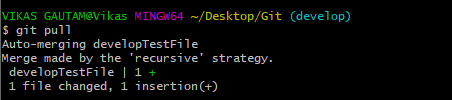
In this case git will ask up to either commit or stash the change.



**Q:**

What will happen if local change is committed but we still do a git pull??

**A:**



In that case git will try to merge the remote update with the committed code. We may get conflict in this scenario. Please note that commit should be done at end only once our task is done. If it’s not done we should stash the change and then do git pull. Once the pull is successful, again we can merge stashed changes with the remote changes and if there is any conflict it can be resolved manually.

## Start new Repository

git init

This command is used to start a new repository.

## Adding url as origin.

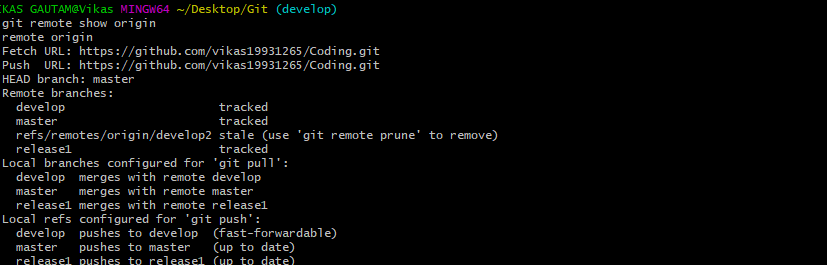
git remote add origin [url]

This command is used to connect the local repository the remote repository. Here we are basically specifying the origin variable as the url of the remote repository and then later use that url.

## Check origin url

git remote show origin

This command will show all the branches in remote repository along with current HEAD that is the current branch or master branch.

****

## Unstage a file

git reset

This command will unstage a file but preserve its contents.

## Undo a commit

git reset [commit name]

This command will undo all the commits after the specified commit

## Deleting a branch in local.

Command**:**

git branch –d “branchName”

## Deleting a branch in remote

Command**:**

git push origin --delete origin/develop2

## Checkout new branch

git branch

This command will show the current branch name in use.

## Adding branch in local.

git checkout –b “BranchName”

This command will create a new branch in the local and also switch to it.

## Creating new branch in local.

git branch branch\_name

This command will create a new branch in the local.

## See list of branches.

git branch

This command will show the current branch name in use.

## Git Stash

The git stash command is used when we want to remove our local changes temporarily from our working area. This is saved at some place which can be later called again.

This command enables us to switch between branches without committing our code or do a git pull do our local repository again without committing our local changes.

git stash

This command will save our local changes temporarily. We can check for any modified files or new files added using git status command. If this command show files modified then it means that there are local changes done. If here are some new files added then that will be shown in untracked files.

git stash save “message”

This command will stash the files temporarily with a stash message.

git stash list

This command will show all the stashed storages. The list will be in order from the most recent to oldest.

git stash apply

This command will apply the stashed commit.

git stash apply <stash id>

Using this command we can choose which stashed item we want to apply from the list.

git stash show

This command will show the stashed list items with all the changes which are done on file. It will show like number of lines updates, insertions made.

git stash show –p

This command will show the stashed list items along with exact data which was changed over it.

git stash pop

This command will apply the stashed change. But please note that unlike stash apply, once the stashed item is applied from the list it will be removed.

git stash drop

This command will drop the stashed item from list.

git stash drop<stash id>

This command will drop the stash with stash id from list.

git stash clear

This command will clear all the stashes from the stack.

## Commit in git

git commit –m “commit message”

This command is used to commit changes we have made into the local repository. It is mandatary to give –m, in order to give the commit message without which we have to do in vi editor but its compulsory.

This command records snapshot of the version permanently.

## Checking status of files in repository

git status :

This command is used to see status of file. If any new file is added it will show here. If any file is modified then that file will also be seen here.

## Cloning git

git clone [url]

This command is used to download a copy of source code from the repository.

## Git logging

git log

This command is used to list the version history of the current branch.

git log --follow <file or folder name>

This command is used to find history of a specific file or a folder.

## Adding file to staging area.

git add –<filename>

git add "Git/GitCommands.docx

This command is used to add a file (single file) to the staging area.

git add\*

This command adds one or more file to the staging area.

git add –A

This command will add all the files in our workspace to the staging area or we can say it will index the files. After this only the files can be committed. –A has to be added that too capital A so that all the files which are to be indexed can be added to the staging area.

## Removing file

git rm-<file or folder name>

This command will remove a file or folder from local and stages it for deletion. Once committed and pushed it will be removed from the remote repository also.