# **Important git commands**

* To **list** all branch in your repo - git branch
* To **checkout the remote branch** say master like a local branch - git checkout master
* To **create** a new branch locally – git branch <new-branch> or git checkout -b ＜new-branch＞
* To **checkout** a new branch – git checkout -b <branch> or git checkout -b ＜new-branch＞ (checkout is used to switch branches)
* To **upstream** - git push --set-upstream origin <branch> or git push -u origin <branch> (to track the branch based on branch so that git pull and git push will work)
* To **duplicate** a branch - git checkout -b <new-branch> <old-branch>
* To **delete** local copy of the branch without merge - git branch -d crazy-experiment or git branch -D crazy-experiment (deletes without warning message
* To **delete** branch from the remote - git push origin --delete <branch> or git push origin :<branch>
* To **get status** of a branch – git status
* To **get log** of a branch – git log
* Steps to **update changes** from master/release to the branch –

git checkout master

git pull

git checkout <branch>

git merge master

git push

Here to add a commit message while merging, press i to insert comment, and then press Esc and enter :wq to save and exit from the editor.

* Another method to **keep the branch updated** and in sync with the master -

git checkout master

git pull

git checkout branch

git pull

git fetch origin master or git fetch upstream master

git rebase upstream/master or git merge origin/master -m 'master sync'

git push origin branch

* To **add** all the branch commits over the merge - git pull –rebase
* To **reset** your local master/branch to the remote branche last commit - git reset --hard origin/master
* To **fetch** contents of a remote branch before checking out – git fetch --all
* To **update** the local representation of the remote branch and prune deleted branches - git fetch -p

**Prior to merge:**

Confirm the receiving branch - git checkout <recieving branch>

Fetch latest remote commits - git fetch

Ensure receiving branch has latest updates - git pull

**To merge** - git merge <branch to be merged>

Fast-forward merge occurs when there is no diverge between the two branches and path is linear.

Another method **to resolve conflicts** using command line is to use, git mergetool

It will bring up the different versions of the file in different Vim splits panels.

+--------------------------------+

| LOCAL | BASE | REMOTE |

+--------------------------------+

| MERGED |

+--------------------------------+  
You can navigate among these views using ctrl+w.

If you want to get changes from REMOTE type :diffg RE

If you want to get changes from BASE type :diffg BA

If you want to get changes from LOCAL type :diffg LO

Type :wqa to save the changes and quit vi

Commit the merge. Use git clean to remove the untracked files (\*.orig) created by the tool.

**Common issues:**

**Your branch is ahead of 'origin/master' by X commits:**

You might get this message if you made changes in your local master and didn't push them to remote. One way to resolve this issue is to push your local changes by using the command, git push origin

If you want to discard your local changes and bring it in synch with the master then use the command, git reset --hard origin/master