* ***Create New Branch***

**git** branch ***<Branch Name>***

* ***Switch To New Branch***

**git checkout <Branch Name>**

**git checkout -b <Branch Name>**

**git checkout -u origin <Branch Name>**

* ***Rename your last git commit-message with new message***

**git commit;**

* ***Stash Changes***

**git stash save 'Commit Message'**

* ***Stash List***

**git stash list**

* ***Apply Stashed Changes***

**git stash apply**

* ***Apply Specific Stash***

**git stash <Stash Id>**

*Example*: Stash Id will look like this **stash@{0}:**

* ***Send local branch to Server***

**git push -u origin <Branch Name>**

* ***Reset branch to as it was earlier when branch was created***

**git reset - - hard**

* ***Un stage specific files which are staged***

**git reset HEAD <filename….>**

* ***How to merge master to branch?***
* **First make sure branch code is up to date & committed**
* **Checkout to master branch**
* **Pull all the latest master code by executing**

***git pull;***

* **Now all the master code will be updated into your local master branch**
* **Then checkout to your local feature where you are expecting the merge should happen using below git command**

***git checkout <feature\_branch\_name>***

* **Now merge the master code to feature branch by executing**

***git merge master;***

* **If there are merge conflicts then the feature branch will go into merging state**
* **Then you have to resolve the merge conflicts locally in Editor**
* **After resolving conflicts, add the conflicted files to the branch by executing**

***git add <filename>;***

* **Later execute**

***git merge - -continue;***

* **Check the status of your feature branch**
* **Then send the merge state to server feature branch by executing below command**

***git commit –m ‘Commit Message Here’***

***git push***