Description

Rebase is a way of merging *master* to your branch when you are working with long running branch.

Steps for Rebase Operation

Step 1 − Go to your project directory and create a new branch with the name *rebase-example* by using the *git checkout* command −

```
root@SandboxHost-637840598252748881:~/MyFirstGitLabRepo/first-gitlab-prjt# git checkout -b rebase-example
Switched to a new branch 'rebase-example'
```

The flag *-b* indicates new branch name.

Step 2 − Now, create a new file and add some content to that file as shown below −

```
root@SandboxHost-637840598252748881:~/MyFirstGitLabRepo/first-gitlab-prjt# echo "Ernesto.Net" >> rebase_file.md
root@SandboxHost-637840598252748881:~/MyFirstGitLabRepo/first-gitlab-prjt# ll
total 28
drwxr-sr-x 3 root users 4096 Mar 29 13:08 ./
drwxr-sr-x 3 root users 4096 Mar 29 12:37 ../
drwxr-sr-x 8 root users 4096 Mar 29 13:08 .git/
-rw-r--r-- 1 root users 6270 Mar 29 12:37 README.md
-rw-r--r-- 1 root users   12 Mar 29 13:08 rebase_file.md
-rw-r--r-- 1 root users    7 Mar 29 12:38 shoppingList.txt
```

The content 'Ernesto.Net' will be added to the *rebase_file.md* file.

Step 3 − Add the new file to working directory and store the changes to the repository along with the message (by using the *git commit* command) as shown below −

```
root@SandboxHost-637840598252748881:~/MyFirstGitLabRepo/first-gitlab-prjt# git add .
root@SandboxHost-637840598252748881:~/MyFirstGitLabRepo/first-gitlab-prjt# git commit -m "Rebase file added"
[rebase-example 63e71d1] Rebase file added
 1 file changed, 1 insertion(+)
 create mode 100644 rebase_file.md
root@SandboxHost-637840598252748881:~/MyFirstGitLabRepo/first-gitlab-prjt#
```

The flag *-m* is used for adding a message on the commit.

Step 4 − Now, switch to the 'master' branch. You can fetch the remote branch(*master* is a branch name) by using the *git checkout* command −

```
C:\first-gitlab-prjt>git checkout master
Switched to branch 'master'
Your branch and 'origin/master' have diverged,
and have 1 and 3 different commits each, respectively.
  (use "git pull" to merge the remote branch into yours)

C:\first-gitlab-prjt>
```

Step 5 − Next, create another new file, add some content to that file and commit it in the *master* branch.

```
C:\first-gitlab-prjt>echo "text in main branch" >> README.md

C:\first-gitlab-prjt>git add .

C:\first-gitlab-prjt>git commit -m "Commit in master"
[master 7dd6a44] Commit in master
 1 file changed, 1 insertion(+)

C:\first-gitlab-prjt>_
```

Step 6 − Switch to the *rebase-branch* to have the commit of *master* branch.

```
C:\first-gitlab-prjt>git checkout rebase-branch
Switched to branch 'rebase-branch'

C:\first-gitlab-prjt>
```

Step 7 − Now, you can combine the commit of *master* branch to *rebase-branch* by using the *git rebase* command −

```
C:\first-gitlab-prjt>git rebase master
First, rewinding head to replay your work on top of it...
Applying: Another commit

C:\first-gitlab-prjt>
```