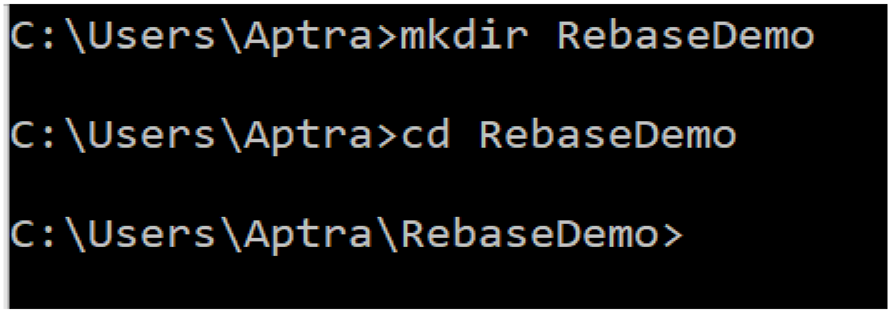
Activity-5

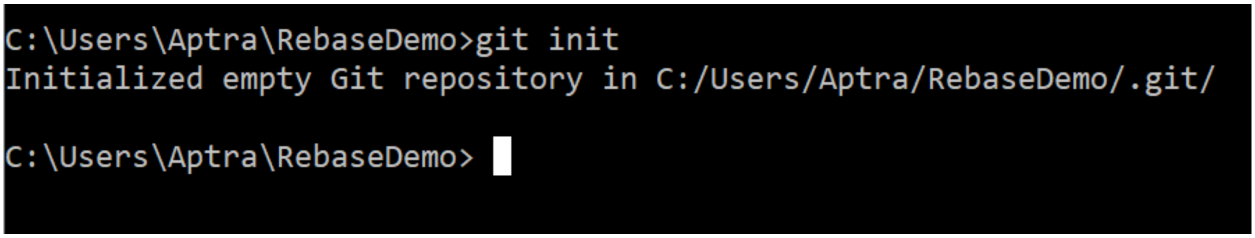
Fetch the latest changes from a remote repository and rebase your local branch onto the updated remote branch.

1. Create directory "RebaseDemo" and go inside the folders



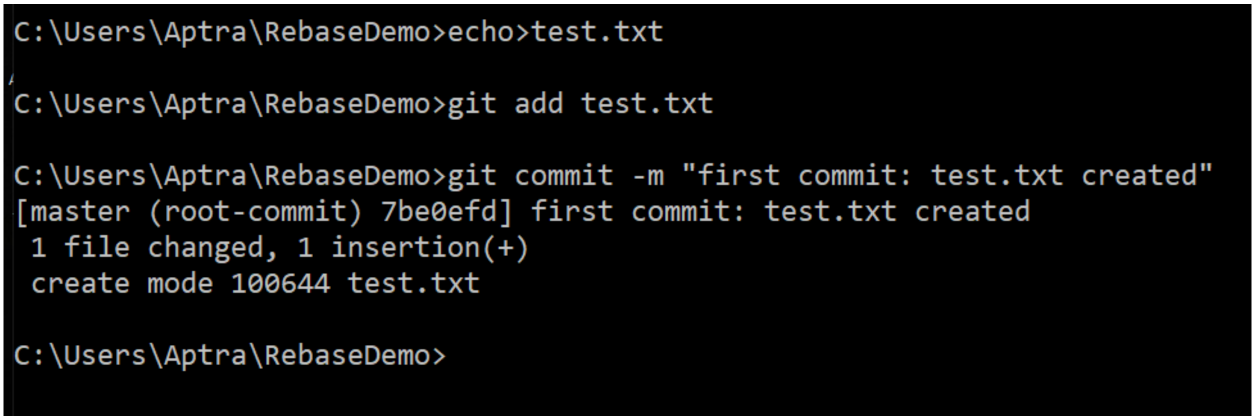
```
C:\Users\Aptra>mkdir RebaseDemo

C:\Users\Aptra>cd RebaseDemo

C:\Users\Aptra\RebaseDemo>
```

2. Initialize the directory to git repository



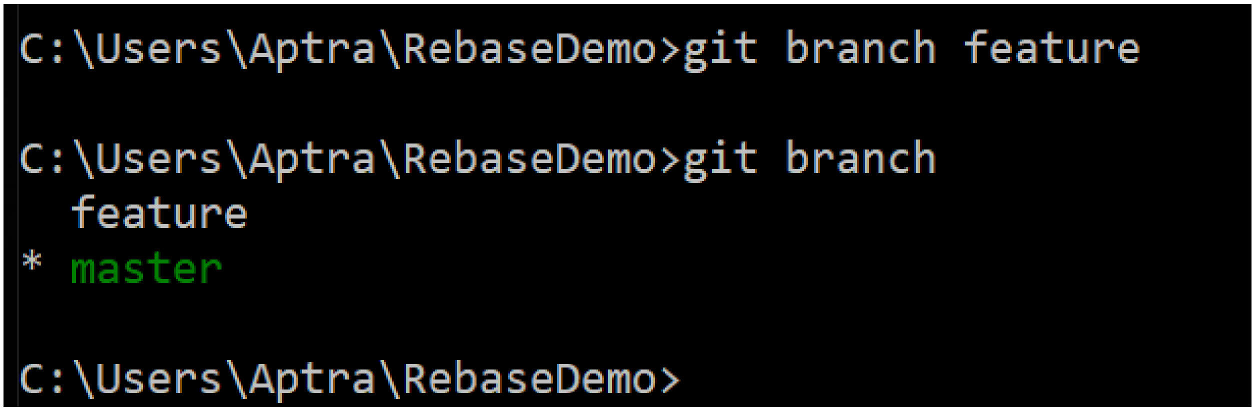
```
C:\Users\Aptra\RebaseDemo>git init
Initialized empty Git repository in C:/Users/Aptra/RebaseDemo/.git/

C:\Users\Aptra\RebaseDemo>
```

3. Create file test.txt and put to staging and then commit it.



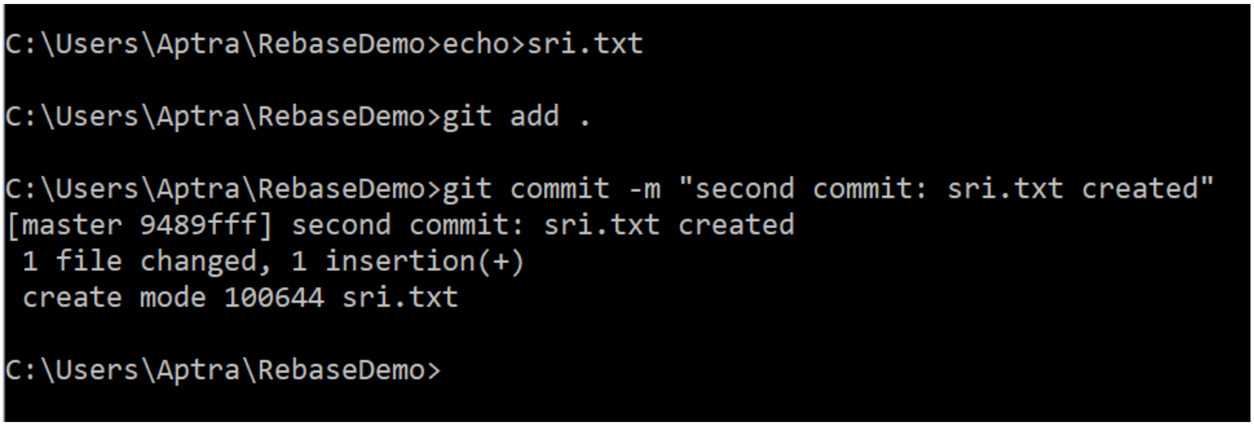
```
C:\Users\Aptra\RebaseDemo>echo>test.txt

C:\Users\Aptra\RebaseDemo>git add test.txt

C:\Users\Aptra\RebaseDemo>git commit -m "first commit: test.txt created"
[master (root-commit) 7be0efd] first commit: test.txt created
 1 file changed, 1 insertion(+)
 create mode 100644 test.txt

C:\Users\Aptra\RebaseDemo>
```

4. Create feature branch and check



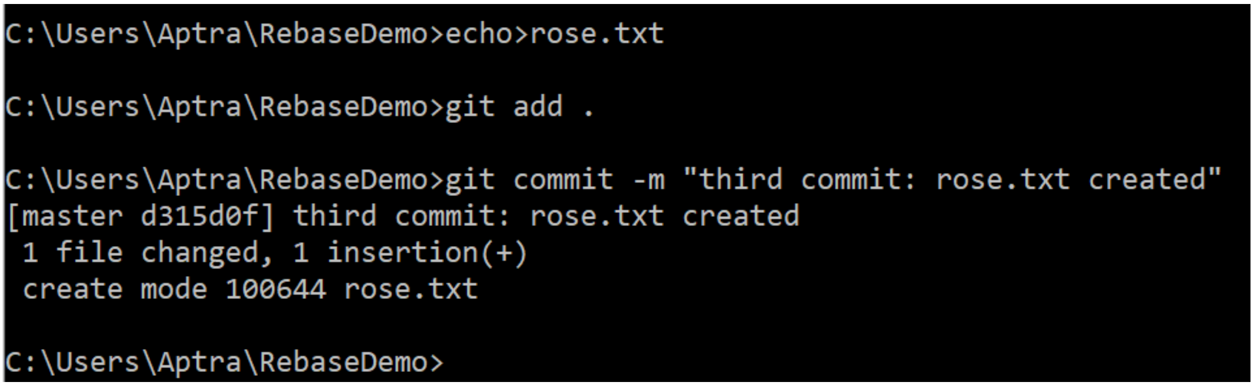
```
C:\Users\Aptra\RebaseDemo>git branch feature

C:\Users\Aptra\RebaseDemo>git branch
  feature
* master

C:\Users\Aptra\RebaseDemo>
```

5. Create one more file "sri.txt" in master branch and add to staging area and then commit it.



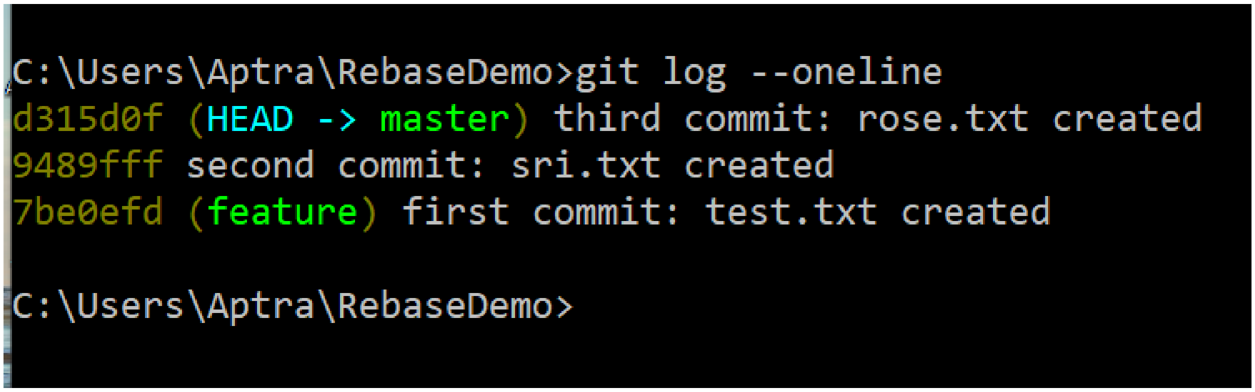
```
C:\Users\Aptra\RebaseDemo>echo>sri.txt

C:\Users\Aptra\RebaseDemo>git add .

C:\Users\Aptra\RebaseDemo>git commit -m "second commit: sri.txt created"
[master 9489fff] second commit: sri.txt created
 1 file changed, 1 insertion(+)
 create mode 100644 sri.txt

C:\Users\Aptra\RebaseDemo>
```

6. Create one more file "rose.txt" in master branch and add to staging area and then commit it.

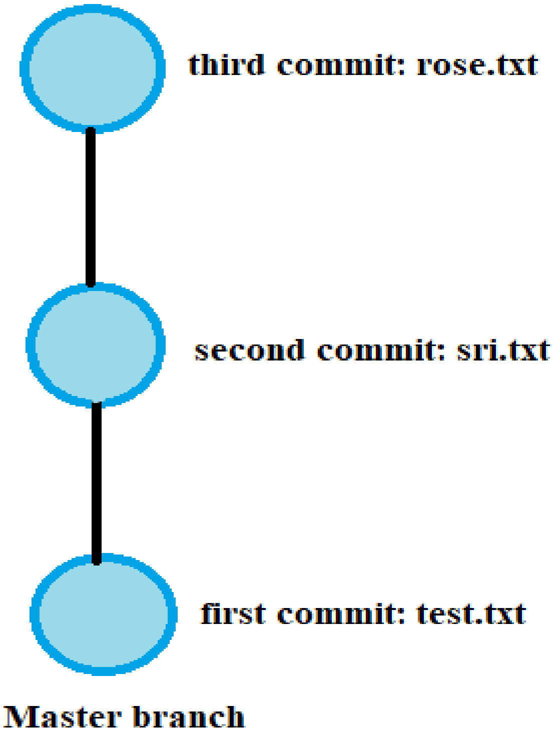


```
C:\Users\Aptra\RebaseDemo>echo>rose.txt

C:\Users\Aptra\RebaseDemo>git add .

C:\Users\Aptra\RebaseDemo>git commit -m "third commit: rose.txt created"
[master d315d0f] third commit: rose.txt created
 1 file changed, 1 insertion(+)
 create mode 100644 rose.txt

C:\Users\Aptra\RebaseDemo>
```

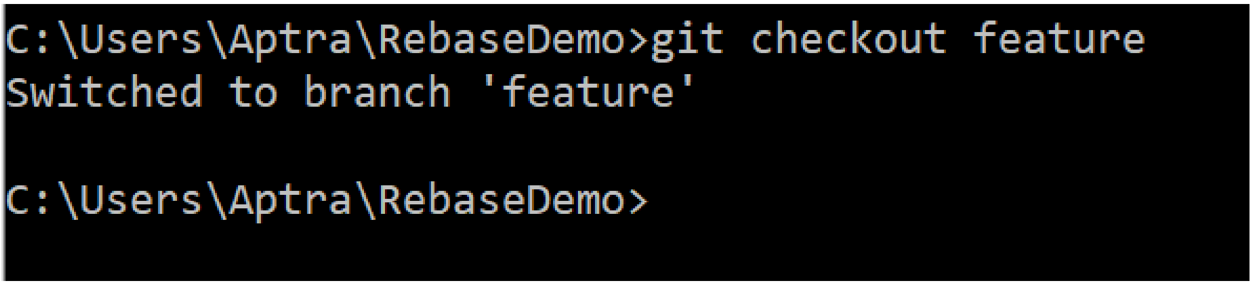
7. Check all the commits git



```
C:\Users\Aptra\RebaseDemo>git log --oneline
d315d0f (HEAD -> master) third commit: rose.txt created
9489fff second commit: sri.txt created
7be0efd (feature) first commit: test.txt created

C:\Users\Aptra\RebaseDemo>
```

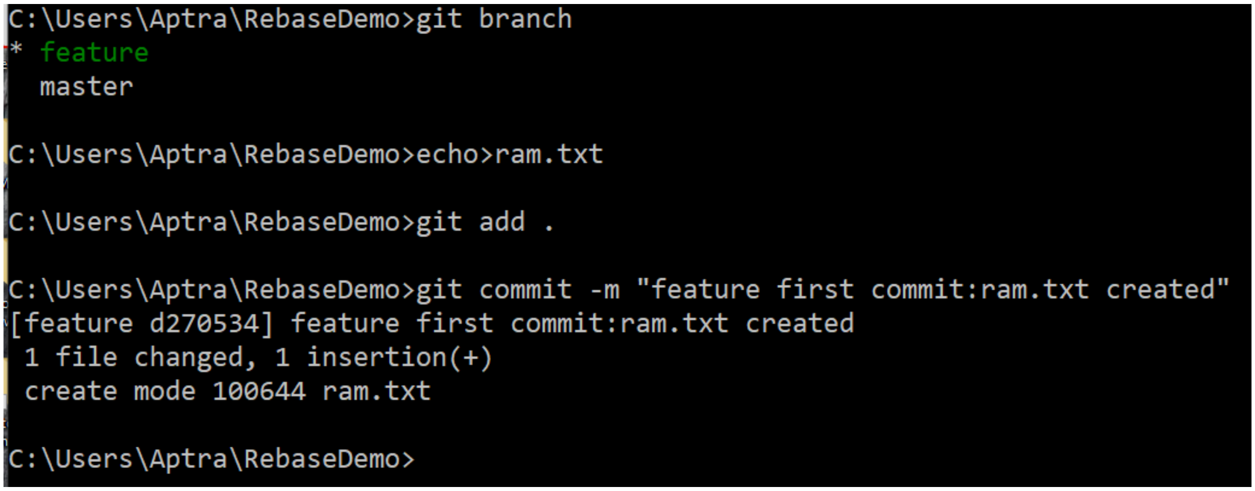


8. Switch to feature branch



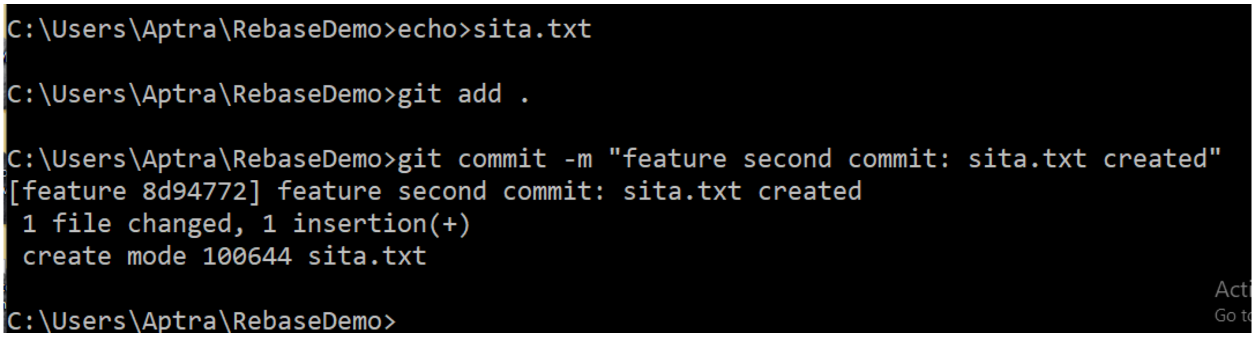
```
C:\Users\Aptra\RebaseDemo>git checkout feature
Switched to branch 'feature'

C:\Users\Aptra\RebaseDemo>
```

9. Create file in feature branch "ram.txt" and add to staging area and then commit it.



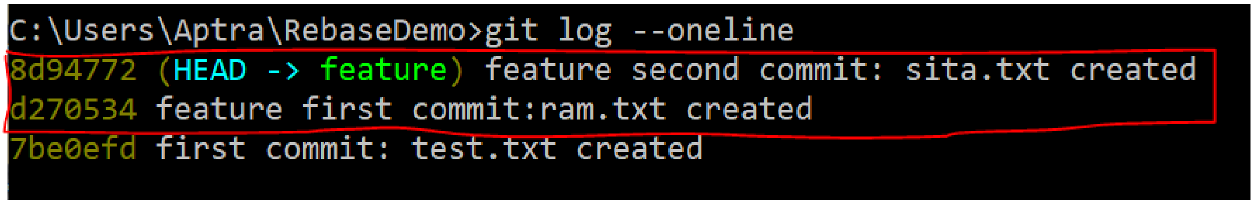
```
C:\Users\Aptra\RebaseDemo>git branch
* feature
  master

C:\Users\Aptra\RebaseDemo>echo>ram.txt

C:\Users\Aptra\RebaseDemo>git add .

C:\Users\Aptra\RebaseDemo>git commit -m "feature first commit:ram.txt created"
[feature d270534] feature first commit:ram.txt created
 1 file changed, 1 insertion(+)
 create mode 100644 ram.txt

C:\Users\Aptra\RebaseDemo>
```

10. Create one more file "sita.txt" in feature branch and add to staging area and then commit it.



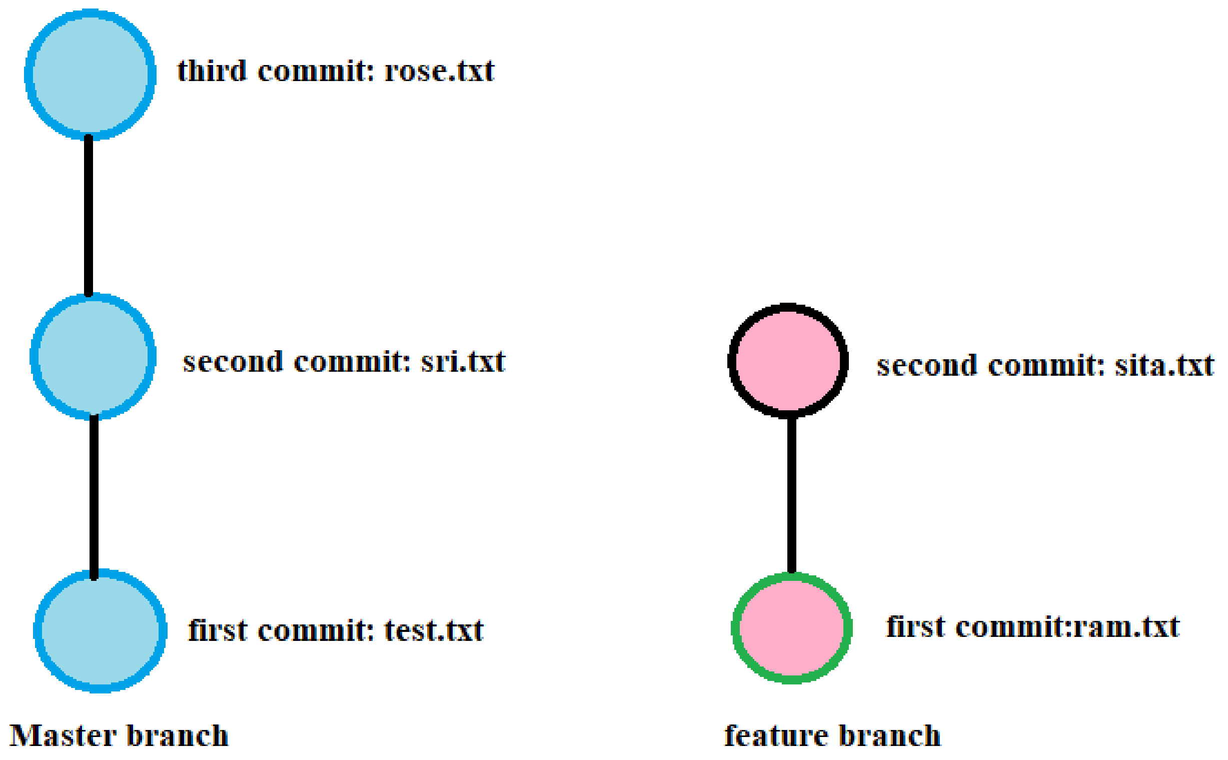
```
C:\Users\Aptra\RebaseDemo>echo>sita.txt

C:\Users\Aptra\RebaseDemo>git add .

C:\Users\Aptra\RebaseDemo>git commit -m "feature second commit: sita.txt created"
[feature 8d94772] feature second commit: sita.txt created
 1 file changed, 1 insertion(+)
 create mode 100644 sita.txt

C:\Users\Aptra\RebaseDemo>
```

11. Check the commits in feature branch



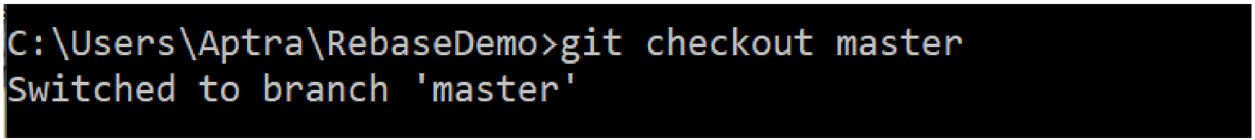
```
C:\Users\Aptra\RebaseDemo>git log --oneline
8d94772 (HEAD -> feature) feature second commit: sita.txt created
d270534 feature first commit:ram.txt created
7be0efd first commit: test.txt created
```

It will show two commits (ram and sita txt file) and you can see on master commit also from above diagram

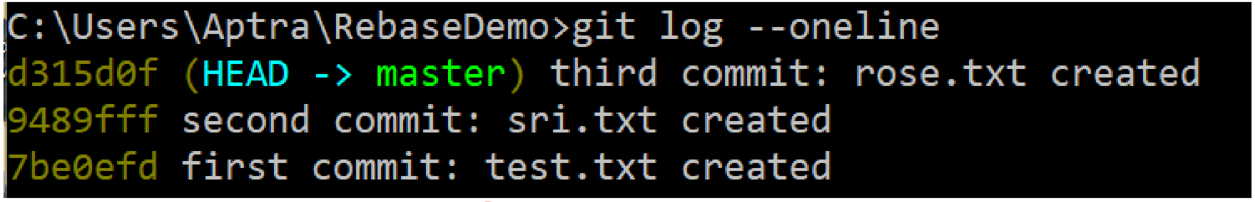


Rebase Process

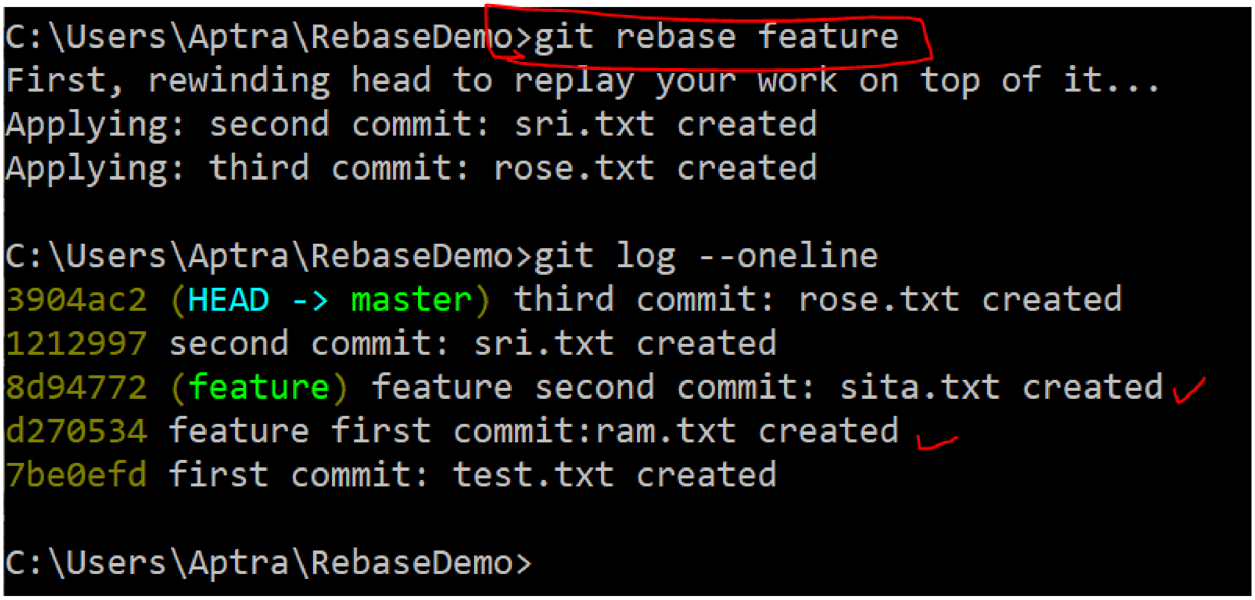
12. Checkout to master branch and use rebase command to move all feature branch commits to master branch.



```
C:\Users\Aptra\RebaseDemo>git checkout master
Switched to branch 'master'
```



```
C:\Users\Aptra\RebaseDemo>git log --oneline
d315d0f (HEAD -> master) third commit: rose.txt created
9489fff second commit: sri.txt created
7be0efd first commit: test.txt created
```

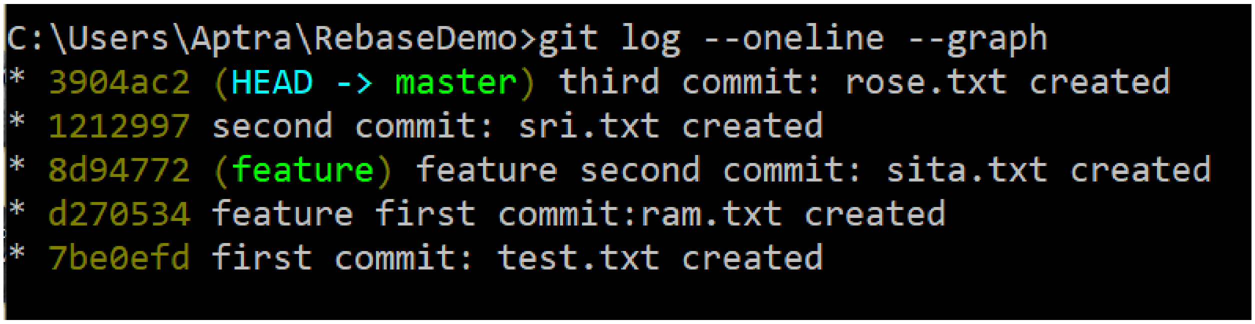


```
C:\Users\Aptra\RebaseDemo>git rebase feature
First, rewinding head to replay your work on top of it...
Applying: second commit: sri.txt created
Applying: third commit: rose.txt created

C:\Users\Aptra\RebaseDemo>git log --oneline
3904ac2 (HEAD -> master) third commit: rose.txt created
1212997 second commit: sri.txt created
8d94772 (feature) feature second commit: sita.txt created
d270534 feature first commit:ram.txt created
7be0efd first commit: test.txt created

C:\Users\Aptra\RebaseDemo>
```

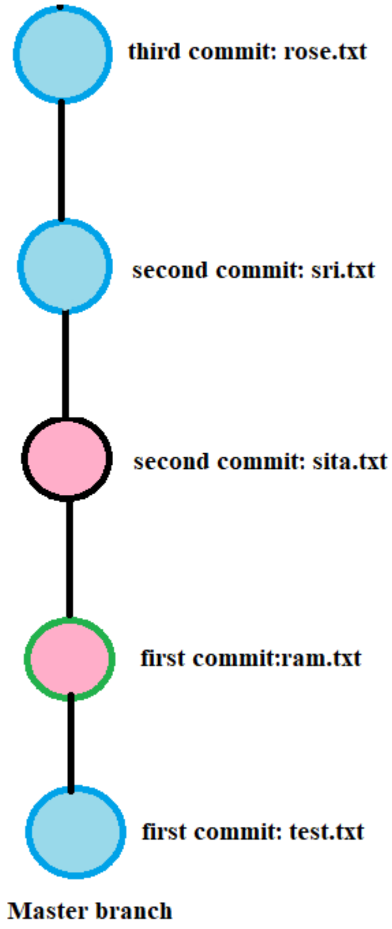
You can see from the above figure that all feature branch commits (ram and sita) now aligned with master commits.

****Note : you can observe except the first hash value remaining hash value are changed**

Note :- you can check the graphical format using git command



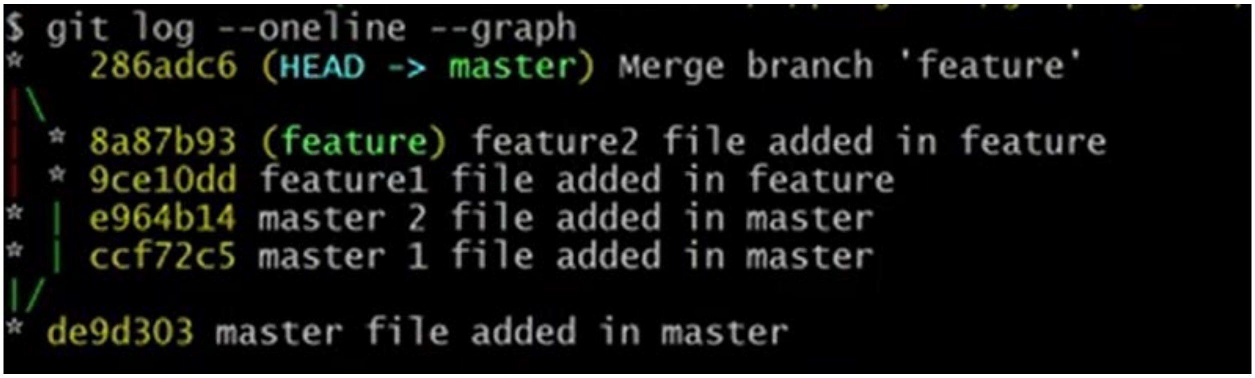
```
C:\Users\Aptra\RebaseDemo>git log --oneline --graph
* 3904ac2 (HEAD -> master) third commit: rose.txt created
* 1212997 second commit: sri.txt created
* 8d94772 (feature) feature second commit: sita.txt created
* d270534 feature first commit:ram.txt created
* 7be0efd first commit: test.txt created
```



Note: rebase Vs merge

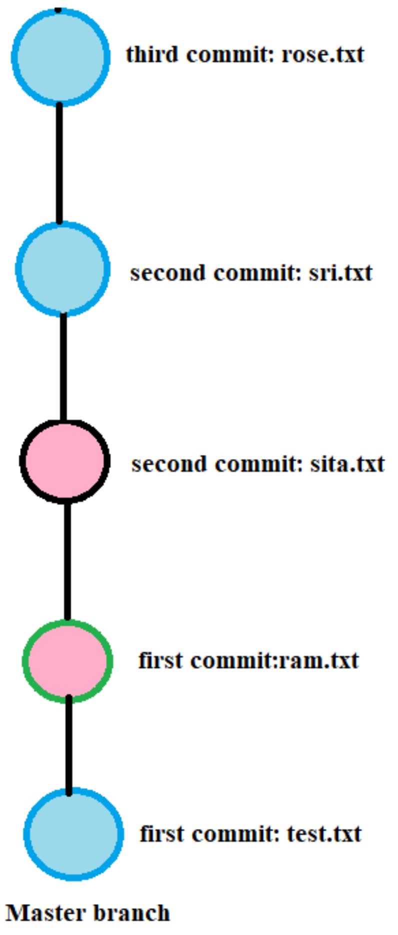
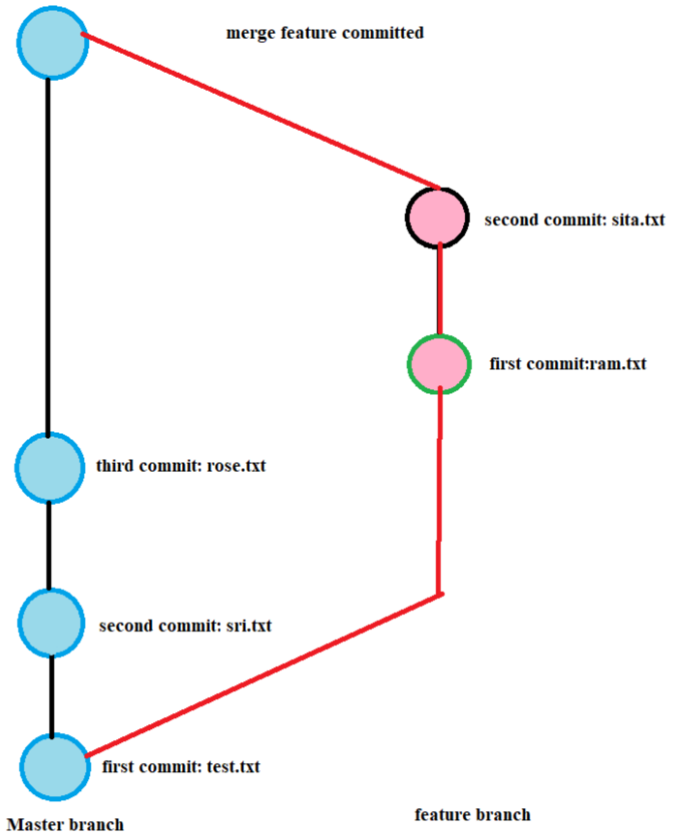
**Rebase will not preserve the history whereas Merge persevere the history.

Do the same steps 1 to 11 and in step 12 instead of using rebase use merge command(i.e >git merge feature) and you can see the output as shown below



```
$ git log --oneline --graph
*   286adc6 (HEAD -> master) Merge branch 'feature'
|\
| * 8a87b93 (feature) feature2 file added in feature
| * 9ce10dd feature1 file added in feature
* | e964b14 master 2 file added in master
* | ccf72c5 master 1 file added in master
|/
* de9d303 master file added in master
```

Prepared by: Srinath K S

Rebase	Merge
 third commit: rose.txt second commit: sri.txt second commit: sita.txt first commit:ram.txt first commit: test.txt Master branch	 merge feature committed second commit: sita.txt first commit:ram.txt third commit: rose.txt second commit: sri.txt first commit: test.txt Master branch feature branch
Hash values changes after rebaseing	Hash value remains same after merging

Prepared by: Srinath K S