

# Git Practical:

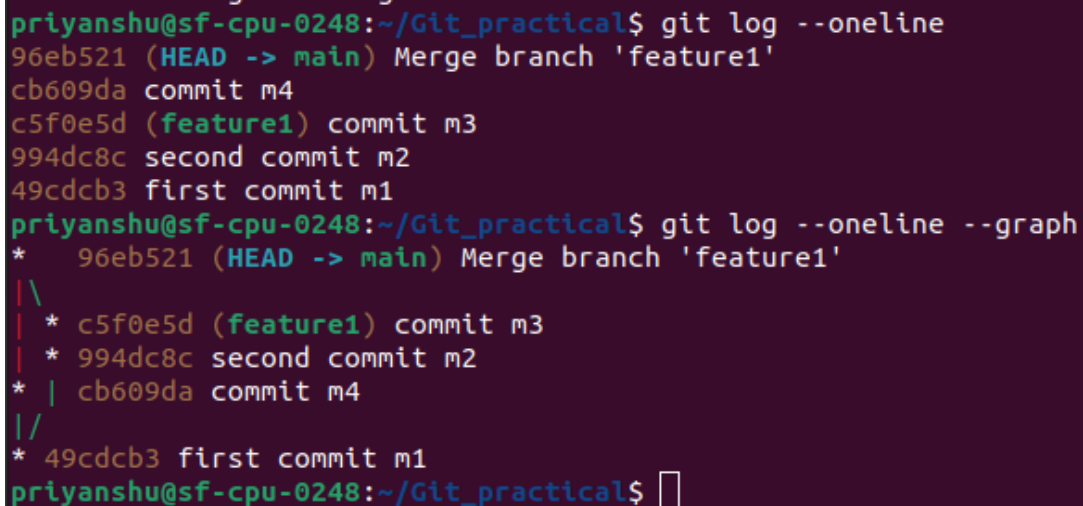
## Practical – 1: Pull and Merge difference

- Make example of pull request and two branch merge events.

Git Merge:

```
git merge [-n] [--stat] [--no-commit] [--squash] [--[no-]edit]
          [--no-verify] [-s <strategy>] [-X <strategy-option>] [-
S[<keyid>]]
          [--[no-]allow-unrelated-histories]
          [--[no-]rerere-autoupdate] [-m <msg>] [-F <file>]
          [--into-name <branch>] [<commit>...]
```

It is used to join two or more development histories



```
priyanshu@sf-cpu-0248:~/Git_practical$ git log --oneline
96eb521 (HEAD -> main) Merge branch 'feature1'
cb609da commit m4
c5f0e5d (feature1) commit m3
994dc8c second commit m2
49cdcb3 first commit m1
priyanshu@sf-cpu-0248:~/Git_practical$ git log --oneline --graph
* 96eb521 (HEAD -> main) Merge branch 'feature1'
| \
| * c5f0e5d (feature1) commit m3
| * 994dc8c second commit m2
* | cb609da commit m4
|/
* 49cdcb3 first commit m1
priyanshu@sf-cpu-0248:~/Git_practical$
```

## Git pull:

```
git pull [<options>] [<repository> [<refspec>...]]
```

It fetch from and integrate with another repository or a local branch

```
priyanshu@sf-cpu-0248:~/Git_practical$ git pull https://github.com/priyanshunotavailable/Git_pract.git
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (7/7), 2.11 KiB | 1.06 MiB/s, done.
From https://github.com/priyanshunotavailable/Git_pract
 * branch            HEAD              -> FETCH_HEAD
Updating 96eb521..ecb1d5c
Fast-forward
 one.txt | 5 +++++
 1 file changed, 4 insertions(+), 1 deletion(-)

priyanshu@sf-cpu-0248:~/Git_practical$ git log --oneline
ecb1d5c (HEAD -> main) Merge pull request #1 from priyanshunotavailable/feature1
4c9c075 Merge branch 'main' into feature1
c2a1944 Update one.txt
96eb521 Merge branch 'feature1'
cb609da commit m4
c5f0e5d (feature1) commit m3
994dc8c second commit m2
49cdcb3 first commit m1
priyanshu@sf-cpu-0248:~/Git_practical$
```

## Pull request:

priyanshunotavailable / Git\_practPublic

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

Pin

Unwatch

Fork

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: main ← compare: feature1

✖ Can't automatically merge. Don't worry, you can still create the pull request.

Updated one.txt

WritePreview

H B I L M U T Y

Some text is added from here

Attach files by dragging & dropping, selecting or pasting them.

Create pull request

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Reviewers

No reviews

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Use [Closing keywords](#) in the description to automatically close issues.

Helpful resources

[GitHub Community Guidelines](#)

priyanshunotavailable / Git\_practPublic

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

Pin

Unwatch

Fork

Updated one.txt #1

Edit<> Code

Open priyanshunotav... wants to merge 1 commit into main from feature1

Conversation0Commits1Checks0Files changed1

priyanshunotav... commented now

Owner

Some text is added from here

Update one.txt

Verifiedc2a1944

Add more commits by pushing to the feature1 branch on priyanshunotavailable/Git\_pract.

This branch has conflicts that must be resolved

Use the web editor or the [command line](#) to resolve conflicts.

Conflicting files

one.txt

Resolve conflicts

Merge pull requestor view command line instructions.

WritePreview

H B I L M U T Y

Leave a comment

Reviewers

No reviews

Still in progress? Convert to draft

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

Updated one.txt #1

Edit<> Code

Open priyanshunotav... wants to merge 2 commits into main from feature1

Conversation0Commits2Checks0Files changed1

priyanshunotav... commented now

Owner

Some text is added from here

priyanshunotavailable added 2 commits 2 minutes ago

Update one.txt

Verifiedc2a1944

Merge branch 'main' into feature1

Verified4c9c875

Add more commits by pushing to the feature1 branch on priyanshunotavailable/Git\_pract.

Continuous integration has not been set up

[GitHub Actions](#) and [several other apps](#) can be used to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull requestor view command line instructions.

WritePreview

H B I L M U T Y

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Reviewers

No reviews

Still in progress? Convert to draft

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

Notifications

Customize

Unsubscribe

You're receiving notifications because you're watching this repository.

1 participant

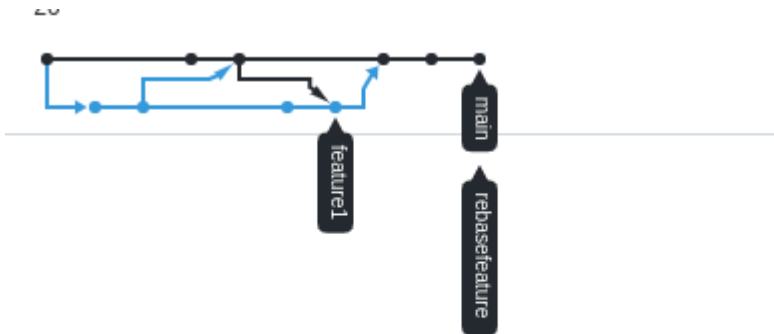
## Practical – 2: Rebase

### - Try to rebase feature branch with master branch

Rebasing is the process of moving or combining a sequence of commits to a new base commit. Rebasing is most useful and easily visualized in the context of a feature branching workflow.

`git rebase <base>`

```
priyanshu@sf-cpu-0248:~/Git_practical$ git rebase rebasefeature
Successfully rebased and updated refs/heads/main.
priyanshu@sf-cpu-0248:~/Git_practical$ git log --oneline
2cac6a4 (HEAD -> main, rebasefeature) commit 2 for rebase
f5c228a commit 1 for rebase
ecb1d5c Merge pull request #1 from priyanshunotavailable/feature1
4c9c075 Merge branch 'main' into feature1
c2a1944 Update one.txt
96eb521 Merge branch 'feature1'
cb609da commit m4
c5f0e5d (feature1) commit m3
994dc8c second commit m2
49cdcb3 first commit m1
```



### Practical – 3: Change commit message

- Commit push on commit in feature branch and then change commit message

```
priyanshu@sf-cpu-0248:~/Git_practical$ git log -1
commit 2cac6a4b8495250a318b5365c22aae8c1d1353ba (HEAD -> main, rebasefeature)
Author: Priyanshu <priyanshu.s@simformsolutions.com>
Date: Mon Feb 20 11:27:13 2023 +0530

    commit 2 for rebase
priyanshu@sf-cpu-0248:~/Git_practical$ git commit --amend -m "new commit msg"
[main 9c77453] new commit msg
Date: Mon Feb 20 11:27:13 2023 +0530
1 file changed, 1 insertion(+)
priyanshu@sf-cpu-0248:~/Git_practical$ git log -1
commit 9c77453a551982836c5618369d0781ebc46d3a5f (HEAD -> main)
Author: Priyanshu <priyanshu.s@simformsolutions.com>
Date: Mon Feb 20 11:27:13 2023 +0530

    new commit msg
priyanshu@sf-cpu-0248:~/Git_practical$
```

## Practical – 4: cherry pick

- Pick some commits from feature branch to master branch

```
priyanshu@sf-cpu-0248:~/Git_practical$ git log --oneline
9c77453 (HEAD -> main) new commit msg
f5c228a commit 1 for rebase
ecb1d5c Merge pull request #1 from priyanshunotavailable/feature1
4c9c075 Merge branch 'main' into feature1
c2a1944 Update one.txt
96eb521 Merge branch 'feature1'
cb609da commit m4
c5f0e5d (feature1) commit m3
994dc8c second commit m2
49cdcb3 first commit m1
priyanshu@sf-cpu-0248:~/Git_practical$ git switch cherrybranch
Switched to branch 'cherrybranch'
priyanshu@sf-cpu-0248:~/Git_practical$ git log --oneline
462d0c5 (HEAD -> cherrybranch) delete commit
6d1bfd2 last commit
8d122b2 cherry commit 2
0796643 cherry commit 1
9c77453 (main) new commit msg
f5c228a commit 1 for rebase
ecb1d5c Merge pull request #1 from priyanshunotavailable/feature1
4c9c075 Merge branch 'main' into feature1
c2a1944 Update one.txt
96eb521 Merge branch 'feature1'
cb609da commit m4
c5f0e5d (feature1) commit m3
994dc8c second commit m2
49cdcb3 first commit m1
```

After performing cherry-pick operation:

```
priyanshu@sf-cpu-0248:~/Git_practical$ git cherry-pick 0796643
[main c25b6f8] cherry commit 1
Date: Mon Feb 20 11:38:10 2023 +0530
1 file changed, 1 insertion(+), 1 deletion(-)
priyanshu@sf-cpu-0248:~/Git_practical$ cat one.txt
template 1

This text is added from Github

this text is added to perform the cherry pick operation
priyanshu@sf-cpu-0248:~/Git_practical$ git log --oneline
c25b6f8 (HEAD -> main) cherry commit 1
9c77453 new commit msg
f5c228a commit 1 for rebase
ecb1d5c Merge pull request #1 from priyanshunotavailable/feature1
4c9c075 Merge branch 'main' into feature1
c2a1944 Update one.txt
96eb521 Merge branch 'feature1'
cb609da commit m4
c5f0e5d (feature1) commit m3
994dc8c second commit m2
49cdcb3 first commit m1
```

## Practical – 5: Drop commit

### - Remove some commit from feature branch

The git reset command is a complex and versatile tool for undoing changes.

It has three primary forms of invocation. These forms correspond to command line arguments --soft, --mixed, --hard.

Before drop commit:

```
priyanshu@sf-cpu-0248:~/Git_practical$ git log --oneline
462d0c5 (HEAD -> cherrybranch) delete commit
6d1bfd2 last commit
8d122b2 cherry commit 2
0796643 cherry commit 1
9c77453 new commit msg
f5c228a commit 1 for rebase
ecb1d5c Merge pull request #1 from priyanshunotavailable/feature1
4c9c075 Merge branch 'main' into feature1
c2a1944 Update one.txt
96eb521 Merge branch 'feature1'
cb609da commit m4
c5f0e5d (feature1) commit m3
994dc8c second commit m2
49cdcb3 first commit m1
```

After drop commit:

```
priyanshu@sf-cpu-0248:~/Git_practical$ git reset --hard HEAD~2
HEAD is now at 8d122b2 cherry commit 2
priyanshu@sf-cpu-0248:~/Git_practical$ git log --oneline
8d122b2 (HEAD -> cherrybranch) cherry commit 2
0796643 cherry commit 1
9c77453 new commit msg
f5c228a commit 1 for rebase
ecb1d5c Merge pull request #1 from priyanshunotavailable/feature1
4c9c075 Merge branch 'main' into feature1
c2a1944 Update one.txt
96eb521 Merge branch 'feature1'
cb609da commit m4
c5f0e5d (feature1) commit m3
994dc8c second commit m2
49cdcb3 first commit m1
priyanshu@sf-cpu-0248:~/Git_practical$
```