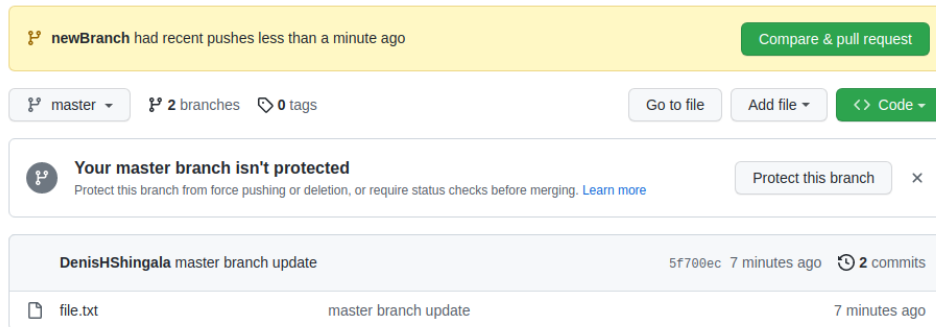


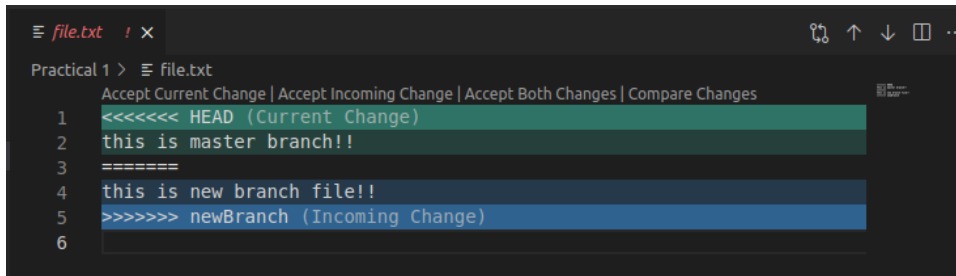
# GitHub Practical's

## 1. Pull and Merge difference

- First of all, we have created two branch 1) **master** & 2) **newBranch**
- Then do commit and push event on both branch.
- Due **newBranch's** push event you will get **pull request** on **master** branch that please pull it and then merge it.



- After pull, you can merge it using ***"git merge newBranch"***
- you will get merge conflict that's happen due to same lines has different content in both branches.



- Therefore, we have to solve it manually. And then commit and push it on remote sever.

```
denis@sf-cpu-387:~/Denis Shingala/Github Practice/Practical 1$ git merge newBranch
Auto-merging file.txt
CONFLICT (content): Merge conflict in file.txt
Automatic merge failed; fix conflicts and then commit the result.
denis@sf-cpu-387:~/Denis Shingala/Github Practice/Practical 1$ git add .
denis@sf-cpu-387:~/Denis Shingala/Github Practice/Practical 1$ git commit -m 'Final commit(after merged!)'
[master 8b3ed0b] Final commit(after merged!)
```

## 2. Rebase

- First of all, we have created two branches. 1) **master** and 2) **Feature**
- I have done **3 commits** on **master branch** and **2 commits** on **Feature branch** using

- **git add .**

- **git commit -m "message"**

```
1 file changed, 2 insertions(+), 1 deletion(-)
• denis@sf-cpu-387:~/Denis Shingala/Github Practice/Practical 2$ git log
commit ca969120df792e26b0eead23991d964e254e7ea5 (HEAD -> master)
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 14:41:54 2023 +0530
commit ca969120df792e26b0eead23991d964e254e7ea5 (HEAD -> master)
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 14:41:54 2023 +0530

master3

commit 8ac81c34a8d714e8983717620b7469f9ed79833d
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 14:41:46 2023 +0530

master2

commit 1474411d0868c3d51a33916677b075cb17526674
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 14:41:25 2023 +0530

master1

commit 9c060658a3632965d78a939426f70b1a729b9be0
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 14:40:32 2023 +0530

initial commit

denis@sf-cpu-387:~/Denis Shingala/Github Practice/Practical 2$ git log
commit df658e978574a615ae85df1076eb86c92c8fc494 (HEAD -> feature)
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 14:49:09 2023 +0530

feature2

commit 5e04b3fd1559aac9f952705f405c28120cd1b810
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 14:48:58 2023 +0530

feature1

commit 9c060658a3632965d78a939426f70b1a729b9be0
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 14:40:32 2023 +0530

initial commit
```

- After that you have to goto the **master** branch using “**git checkout master**”.
- Then use “**git rebase feature**” command to rebase **master** on **feature** branch. Here, **all commits** of **feature branch** will add into the **master branch** and we can see that in below screenshot.

```

denis@sf-cpu-387:~/Denis Shingala/Github Practice/Practical 2$ git log
commit 6db6c0c37f0e80005f52f5b324a564065c0f07cc (HEAD -> master)
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 14:41:54 2023 +0530

    master3

commit c5b5571997482fc9649b70b5d9da7061d12f46f9
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 14:41:46 2023 +0530

    master2

commit 6100dc53e88e06b21ea82fdccffe367c3c1667f7
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 14:41:25 2023 +0530

    master1

commit df658e978574a615ae85df1076eb86c92c8fc494 (feature)
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 14:49:09 2023 +0530

    feature2

commit 5e04b3fd1559aac9f952705f405c28120cd1b810
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 14:48:58 2023 +0530

    feature1

commit 9c060658a3632965d78a939426f70b1a729b9be0
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 14:40:32 2023 +0530

    initial commit

```

**Warning:** Here, we have applied rebase on **master** branch but in real time we **should never** apply it on **master** branch.

### 3. Change commit message

- First of all, I have created two branches. 1) master 2) feature.
- Do some commits on both branch.

```
denis@sf-cpu-387:~/Denis Shingala/Github Practice/Practical 3$ git commit -a -m "New Feature branch commit"
[master 3df8f0b] New Feature branch commit
1 file changed, 2 insertions(+), 1 deletion(-)
denis@sf-cpu-387:~/Denis Shingala/Github Practice/Practical 3$ git lof
git: 'lof' is not a git command. See 'git --help'.

The most similar command is
log
denis@sf-cpu-387:~/Denis Shingala/Github Practice/Practical 3$ git log
commit 3df8f0b58f263b15faec47f17defb7193a139418 (HEAD -> master)
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 15:42:08 2023 +0530

    New Feature branch commit

commit fac43fd64b23b76a18db9613548ec29c8761fa12 (feature)
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 15:41:24 2023 +0530

    Third commit

commit 4653ed106c87f2432be0fd63720e6eb5e5eb91e1
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 15:40:59 2023 +0530

    Second commit

commit 1755edb1abbca69e24258917568b5478555ba2da
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 15:40:41 2023 +0530

    First commit

commit 005f0088d65e394417ald416bcac5e7beab3b4d9
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 15:35:17 2023 +0530

    initial commit
```

- Then if we do some mistake during enter commit message that time, we used “**git commit –amend**” command to do change commit message.
- But this command will change only head commit I mean current branch’s top commit.
- So, that problem if we want to change another commit’s message then what will we do so for that we use **rebase**.

```

• denis@sf-cpu-387:~/Denis Shingala/Github Practice/Practical 3$ git commit --amend
[feature f7d4fa5] New Feature branch commit using --amend
Date: Thu Feb 16 15:51:54 2023 +0530
1 file changed, 2 insertions(+), 1 deletion(-)
• denis@sf-cpu-387:~/Denis Shingala/Github Practice/Practical 3$ git log
commit f7d4fa577a3193d1739b16ba3846096ae92d39ed (HEAD -> feature)
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 15:51:54 2023 +0530

New Feature branch commit using --amend

commit 3f6e1075df5f0481092bb2ded16915f44a131a6d
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 15:41:24 2023 +0530

Third commit

commit 4653ed106c87f2432be0fd63720e6eb5e5eb91e1
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 15:40:59 2023 +0530

Second commit

commit 1755edb1abbca69e24258917568b5478555ba2da
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 15:40:41 2023 +0530

First commit

commit 005f0088d65e394417a1d416bcac5e7beab3b4d9
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 15:35:17 2023 +0530

initial commit

```

- We can change any commit's message of current branch using **rebase** command. We are use ***"git rebase -i HEAD~{numbers of commit}"***.
- Using it you will see below screen. Here I have taken only first commit so it's shown like this. In below screen you see that I have written 'r' (reword) for change the message of commit. It's **tag** for **change message**.

```

GNU nano 6.2 /home/denis/Denis Shingala/Github Practice/Practical 3/.git/rebase-merge/git-rebase-todo *
# f7d4fa5 New Feature branch commit using rebase command
# Rebase 3f6e107..f7d4fa5 onto 3f6e107 (1 command)
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup [-C | -c] <commit> = like "squash" but keep only the previous
# commit's log message, unless -C is used, in which case
# keep only this commit's message; -C is same as -c but
# opens the editor
# x, exec <command> = run command (the rest of the line) using shell
# b, break = stop here (continue rebase later with 'git rebase --continue')
# d, drop <commit> = remove commit
# l, label <label> = label current HEAD with a name
# t, reset <label> = reset HEAD to a label
# m, merge [-C <commit> | -c <commit>] <label> [# <oneline>]
# . create a merge commit using the original merge commit's
# message (or the oneline, if no original merge commit was
# specified); use -c <commit> to reword the commit message
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be aborted.
#

```

- Then after you will see below screen to change the commit message of particular commit. So, Using rebase we can change message of any commit.

```

GNU nano 6.2 /home/denis/Denis Shingala/Github Practice/Practical 3/.git/COMMIT_EDITMSG *
New Feature branch commit using rebase command]
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date: Thu Feb 16 15:51:54 2023 +0530
#
# interactive rebase in progress; onto 3f6e107
# Last command done (1 command done):
# reword f7d4fa5 New Feature branch commit using rebase command
# No commands remaining.
# You are currently editing a commit while rebasing branch 'feature' on '3f6e107'.
#
# Changes to be committed:
#   modified:   file.txt
#
# Untracked files:
#   1.png
#   2.png
#   3.png
#

```

- Then after you will get changed message.

```

Successfully rebased and updated refs/heads/feature.
• denis@sf-cpu-387:~/Denis Shingala/Github Practice/Practical 3$ git rebase -i HEAD~
[detached HEAD 3ad2fd1] New Feature branch commit using rebase command
Date: Thu Feb 16 15:51:54 2023 +0530
1 file changed, 2 insertions(+), 1 deletion(-)
Successfully rebased and updated refs/heads/feature.
• denis@sf-cpu-387:~/Denis Shingala/Github Practice/Practical 3$ git log
commit 3ad2fd1150c51428bb3bec7489cd3b435fea04ef (HEAD -> feature)
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 15:51:54 2023 +0530

New Feature branch commit using rebase command

commit 3f6e1075df5f0481092bb2ded16915f44a131a6d
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 15:41:24 2023 +0530

Third commit

commit 4653ed106c87f2432be0fd63720e6eb5e5eb91e1
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 15:40:59 2023 +0530

Second commit

commit 1755edb1abbca69e24258917568b5478555ba2da
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 15:40:41 2023 +0530

First commit

commit 005f0088d65e394417a1d416bcac5e7beab3b4d9
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 15:35:17 2023 +0530

initial commit

```

#### 4. Cherry pick

- First of all, I have created two branches with some commits 1) master & 2) feature.

```
denis@sf-cpu-387:~/Denis Shingala/Github Practice/Practical 4$ git log
commit 15b900b18bea4c8ba4ca373b33da30d1f2b7e751 (HEAD -> master)
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 16:08:47 2023 +0530

    master3

commit 5672af904901bb6322d6f51cacfb02fb7a8f7db4
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 16:08:36 2023 +0530

    master2

commit 79b6cf204dfc983157c4319e1bca13e3159bc2c4
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 16:08:14 2023 +0530

    master1

commit 35c578232ee4ad2b31f12996c069c40c267d5a2f
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 16:05:38 2023 +0530

    Initial commit
```

```
denis@sf-cpu-387:~/Denis Shingala/Github Practice/Practical 4$ git log
commit 4622512127e6ea82d3d86125385fadcf2a16e53b (HEAD -> feature)
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 16:10:57 2023 +0530

    master3

commit 617be0e77b10e9aa9215b3953ddb12c625138250
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 16:10:38 2023 +0530

    feature3

commit 423cfbcda0b14b2ba4f7f33e7f8893a4c51ba27f
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 16:09:41 2023 +0530

    feature2

commit d7cac159a1d3626b533aa1201721c120550392e4
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 16:09:31 2023 +0530

    feature1

commit 35c578232ee4ad2b31f12996c069c40c267d5a2f
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 16:05:38 2023 +0530

    Initial commit
```

- Now, if we want to take some commit from **feature branch to master branch** then we can it using ***"git cherry-pick {commit hash}"*** command.
- Below we have use **cherry-pick** and take commit from **feature branch to master branch**.

- But remind that sometimes it's generated the conflict so you have to solve it manually and commit it.

```

denis@sf-cpu-387:~/Denis Shingala/Github Practice/Practical 4$ git log
commit 1e149e7f446be067a83e2b2ee6e7dc87e00c93cb (HEAD -> master)
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 16:10:57 2023 +0530

    master4

commit 15b900b18bea4c8ba4ca373b33da30d1f2b7e751
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 16:08:47 2023 +0530

    master3

commit 5672af904901bb6322d6f51cacfb02fb7a8f7db4
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 16:08:36 2023 +0530

    master2

commit 79b6cf204dfc983157c4319e1bca13e3159bc2c4
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 16:08:14 2023 +0530

    master1

commit 35c578232ee4ad2b31f12996c069c40c267d5a2f
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 16:05:38 2023 +0530

    Initial commit

```



## 5. Drop commit

- First of all, we create two branch and do some commits on it. 1)master 2) feature

```
• denis@sf-cpu-387:~/Denis Shingala/Github Practice/Practical 5$ git log
commit 805fa9f25f155428a7b8acd1cf6a000ba9b71cca (HEAD -> feature)
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 16:23:55 2023 +0530

    Third commit

commit 264a178c7b94b52f13aeeda41929bc37d9be4b56
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 16:23:36 2023 +0530

    Second commit

commit b313d2f64f743538938f135fc3efa29c4d641f55
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 16:22:36 2023 +0530

    First commit

commit 59d273245bcae1a3d83a56818d16b957e2642c58 (master)
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 16:21:23 2023 +0530

    Initial commit
```

- If we want to drop some commit then that time, we use **reset** command like ***"git reset --hard HEAD~{number of commits}"***.
- Here there are many flags like --hard, --soft, --mixed. But --hard is drop commit permanently.

```
• denis@sf-cpu-387:~/Denis Shingala/Github Practice/Practical 5$ git reset --hard HEAD~
HEAD is now at 264a178 Second commit
• denis@sf-cpu-387:~/Denis Shingala/Github Practice/Practical 5$ git log
commit 264a178c7b94b52f13aeeda41929bc37d9be4b56 (HEAD -> feature)
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 16:23:36 2023 +0530

    Second commit

commit b313d2f64f743538938f135fc3efa29c4d641f55
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 16:22:36 2023 +0530

    First commit

commit 59d273245bcae1a3d83a56818d16b957e2642c58 (master)
Author: DenisHShingala <denis.s@simformssolutions.com>
Date: Thu Feb 16 16:21:23 2023 +0530

    Initial commit
```