# Git Commands

GIT is a Version Control System also known as VCS

GIT Software is developed by the Linus Torvalds,who developed the Linux.

Basically VCS is the software designed to record changes made to the file over time.

GIT not only records the source code it also tracks the images , research papers or any type of files.

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref (master)

$ mkdir sampleproject/

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref (master)

$ cd sampleproject/

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ touch index.html

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ ls

index.html

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ ls -a

./ ../ index.html

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git init

Initialized empty Git repository in C:/Users/vlab/Desktop/Gitref/sample

project/.git/

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ ls -a

./ ../ .git/ index.html

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git config user.email

varalaxmiganta066@gmail.com

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git config --local user.email varalaxmiganta77@gmail.com

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git config user.email

varalaxmiganta77@gmail.com

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git config user.name vara

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git config user.name

vara

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git config --unset user.name

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git config user.name

varalaxmi ganta

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git config user.email

varalaxmiganta77@gmail.com

**To remove either name or email by using --unset**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git config --unset user.email

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git config user.email

varalaxmiganta066@gmail.com

**To remove the complete section**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git config --local --remove-section user

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git config --local user.email varalaxmiganta77@gmail.com

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git config --local user.name vara

### Help :

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git help -a

Or

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git help --all

Or

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git help

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git help init

One new web page will be open

### Status:

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git status

On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ ls

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ ls -a

./ ../ .git/

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ touch index.html

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ ls

index.html

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git status

On branch master

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

index.html

nothing added to commit but untracked files present (use "git add" to tr

ack)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git add index.html

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git status

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: index.html

**To Remove a file from Staging area by using --cached**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git rm --cached index.html

rm 'index.html'

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git status

On branch master

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

index.html

nothing added to commit but untracked files present (use "git add" to tr

ack)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git add index.html

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git status

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: index.html

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git commit -m "Initial commit"

[master (root-commit) d4b73c8] Initial commit

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 index.html

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git status

On branch master

nothing to commit, working tree clean

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ vi index.html

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory

)

modified: index.html

no changes added to commit (use "git add" and/or "git commit -a")

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git add index.html

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: index.html

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git commit -m " name add "

[master 5e712cd] name add

1 file changed, 2 insertions(+)

**To know the history**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git log

commit 5e712cdb8192c85b9d49d46286bc8f9a4865f31c (HEAD -> master)

Author: vara <varalaxmiganta77@gmail.com>

Date: Wed Mar 6 15:27:35 2024 +0530

name add

commit d4b73c8da7418956ce99baf9febc96abb2d1d45e

Author: vara <varalaxmiganta77@gmail.com>

Date: Wed Mar 6 15:25:33 2024 +0530

Initial commit

### Git DIFF :

Diff command is used in git to track the difference between the changes made on a file.

Diff command takes two inputs and reflects the differences between them.

Git diff : Difference between the working and staging areas

Git diff --staged : Difference between the Staging area and Repository area

Git diff head : Difference between the working area and Repository area

**To check the changes in the working area and the staged area**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ vi contact.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git status

On branch master

Untracked files:

(use "git add <file>..." to include in what will be committed)

contact.txt

nothing added to commit but untracked files present (use "git add" to

track)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git add contact.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: contact.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ vi contact.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: contact.txt

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory

)

modified: contact.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git diff

diff --git a/contact.txt b/contact.txt

index 4edc036..fe09b22 100644

--- a/contact.txt

+++ b/contact.txt

@@ -1 +1,2 @@

contact text

+changes made again

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git diff --staged

diff --git a/contact.txt b/contact.txt

new file mode 100644

index 0000000..4edc036

--- /dev/null

+++ b/contact.txt

@@ -0,0 +1 @@

+contact text

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git diff head

diff --git a/contact.txt b/contact.txt

new file mode 100644

index 0000000..fe09b22

--- /dev/null

+++ b/contact.txt

@@ -0,0 +1,2 @@

+contact text

+changes made again

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git commit -m "contact file added"

[master 48d405b] contact file added

1 file changed, 1 insertion(+)

create mode 100644 contact.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory

)

modified: contact.txt

no changes added to commit (use "git add" and/or "git commit -a")

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git diff

diff --git a/contact.txt b/contact.txt

index 4edc036..fe09b22 100644

--- a/contact.txt

+++ b/contact.txt

@@ -1 +1,2 @@

contact text

+changes made again

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git diff --staged

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git diff head

diff --git a/contact.txt b/contact.txt

index 4edc036..fe09b22 100644

--- a/contact.txt

+++ b/contact.txt

@@ -1 +1,2 @@

contact text

+changes made again

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory

)

modified: contact.txt

no changes added to commit (use "git add" and/or "git commit -a")

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git add contact.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git diff

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git diff --staged

diff --git a/contact.txt b/contact.txt

index 4edc036..fe09b22 100644

--- a/contact.txt

+++ b/contact.txt

@@ -1 +1,2 @@

contact text

+changes made again

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/sampleproject (master)

$ git diff head

diff --git a/contact.txt b/contact.txt

index 4edc036..fe09b22 100644

--- a/contact.txt

+++ b/contact.txt

@@ -1 +1,2 @@

contact text

+changes made again

### How git stores the data ?

Git stores the data in the form of keys and values.

Values is nothing but the contents of the file.

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ git init

Initialized empty Git repository in C:/Users/vlab/Desktop/Gitref/chann

elproj/.git/

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ vi channel.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ git status

On branch master

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

channel.txt

nothing added to commit but untracked files present (use "git add" to

track)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ git log --oneline --all --graph

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ git add channel.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ git commit -m "Initial commit"

[master (root-commit) 2104add] Initial commit

1 file changed, 1 insertion(+)

create mode 100644 channel.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ git log

commit 2104add0ab1048363997de04637283eab092a0ce (HEAD -> master)

Author: varalaxmi ganta <varalaxmiganta066@gmail.com>

Date: Thu Mar 7 09:21:41 2024 +0530

Initial commit

commit 2104add0ab1048363997de04637283eab092a0ce

Commit is the key and SH1 value is 2104add0ab1048363997de04637283eab092a0ce (data)

21 represented key and 04add0ab1048363997de04637283eab092a0ce is value

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ git cat-file 2104add0ab1048363997de04637283eab092a0ce -p

tree 7a8a37e1f0ab40e8547dfa6a84f0655fe27c2064

author varalaxmi ganta <varalaxmiganta066@gmail.com> 1709783501 +0530

committer varalaxmi ganta <varalaxmiganta066@gmail.com> 1709783501 +05

30

Initial commit

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

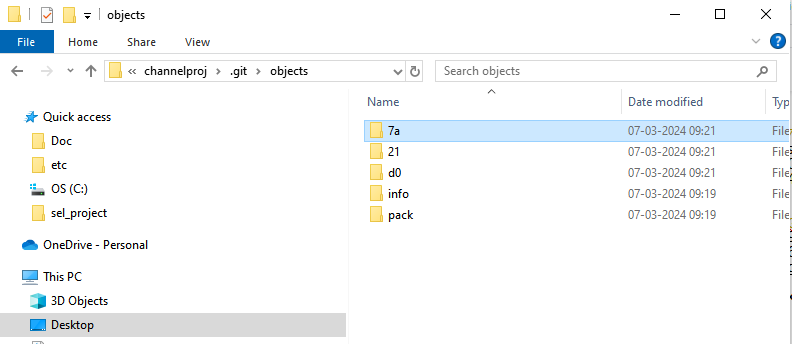
$ git cat-file 7a8a37e1f0ab40e8547dfa6a84f0655fe27c2064 -p

100644 blob d0c9f8634faba69dc02498f904e7fa30b2065726 channel.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ git cat-file d0c9f8634faba69dc02498f904e7fa30b2065726 -p

hello veda world

21 file is linking the 7a tree file and 7a file is linking to the blob d0

d0 has the content of that file

### Rename and Restore files in GIT Repository using mv and the restore command :

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ ls

channel.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ git status

On branch master

nothing to commit, working tree clean

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ mv channel.txt channel.html

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ ls

channel.html

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add/rm <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory

)

deleted: channel.txt

Untracked files:

(use "git add <file>..." to include in what will be committed)

channel.html

no changes added to commit (use "git add" and/or "git commit -a")

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ git add channel.html

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: channel.html

Changes not staged for commit:

(use "git add/rm <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory

)

deleted: channel.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ git add channel.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

renamed: channel.txt -> channel.html

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ git commit -m "rename channel file"

[master b4aa2dd] rename channel file

1 file changed, 0 insertions(+), 0 deletions(-)

rename channel.txt => channel.html (100%)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ git status

On branch master

nothing to commit, working tree clean

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ ls

channel.html

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ git mv channel.html channel.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

renamed: channel.html -> channel.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ git commit -m "channel rename"

[master 2d189df] channel rename

1 file changed, 0 insertions(+), 0 deletions(-)

rename channel.html => channel.txt (100%)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ vi channel.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory

)

modified: channel.txt

no changes added to commit (use "git add" and/or "git commit -a")

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ git add channel.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: channel.txt

**If we want to remove channel.txt file from staging area**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ git restore --staged channel.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory

)

modified: channel.txt

no changes added to commit (use "git add" and/or "git commit -a")

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ git restore channel.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ git status

On branch master

nothing to commit, working tree clean

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/channelproj (master)

$ vi channel.txt

**In that file modified content will be gone**

### GIT Branching :

Local Repository

HEAD : master

Head is current pointer when we commit then a new hash will formed and head will be moved

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git init

Initialized empty Git repository in C:/Users/vlab/Desktop/Gitref/ecomm

erceproj/.git/

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ vi index.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git status

On branch master

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

index.txt

nothing added to commit but untracked files present (use "git add" to

track)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git commit -m "Initial commit"

[master (root-commit) c527478] Initial commit

1 file changed, 1 insertion(+)

create mode 100644 index.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git log

commit c527478df7ddf80fa6411b1d4826432b825b46b1 (HEAD -> master)

Author: varalaxmi ganta <varalaxmiganta066@gmail.com>

Date: Thu Mar 7 09:59:47 2024 +0530

Initial commit

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git log --oneline

c527478 (HEAD -> master) Initial commit

**Checkout -b option used for creating a new branch and move that branch**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git checkout -b login\_feature

Switched to a new branch 'login\_feature'

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (login\_feature)

$ git log

commit c527478df7ddf80fa6411b1d4826432b825b46b1 (HEAD -> login\_feature

, master)

Author: varalaxmi ganta <varalaxmiganta066@gmail.com>

Date: Thu Mar 7 09:59:47 2024 +0530

Initial commit

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (login\_feature)

$ mkdir login

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (login\_feature)

$ cd login

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj/login (login\_feature)

$ vi login.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj/login (login\_feature)

$ git status

On branch login\_feature

Untracked files:

(use "git add <file>..." to include in what will be committed)

./

nothing added to commit but untracked files present (use "git add" to

track)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj/login (login\_feature)

$ cd ..

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (login\_feature)

$ git status

On branch login\_feature

Untracked files:

(use "git add <file>..." to include in what will be committed)

login/

nothing added to commit but untracked files present (use "git add" to

track)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (login\_feature)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (login\_feature)

$ git status

On branch login\_feature

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: login/login.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (login\_feature)

$ git commit -m "Implement login feature ui"

[login\_feature 64f9dcf] Implement login feature ui

1 file changed, 2 insertions(+)

create mode 100644 login/login.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (login\_feature)

$ git log

commit 64f9dcf4ee0a2741360c6dc89e2ad45decdbabfb (HEAD -> login\_feature

)

Author: varalaxmi ganta <varalaxmiganta066@gmail.com>

Date: Thu Mar 7 10:06:07 2024 +0530

Implement login feature ui

commit c527478df7ddf80fa6411b1d4826432b825b46b1 (master)

Author: varalaxmi ganta <varalaxmiganta066@gmail.com>

Date: Thu Mar 7 09:59:47 2024 +0530

Initial commit

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (login\_feature)

$ git log --oneline

64f9dcf (HEAD -> login\_feature) Implement login feature ui

c527478 (master) Initial commit

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (login\_feature)

$ ls

index.txt login/

**By using checkout command change to the other branch**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (login\_feature)

$ git checkout master

Switched to branch 'master'

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ ls

index.txt

**Here we have only one commit in the master branch other commits are not visible**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git log

commit c527478df7ddf80fa6411b1d4826432b825b46b1 (HEAD -> master)

Author: varalaxmi ganta <varalaxmiganta066@gmail.com>

Date: Thu Mar 7 09:59:47 2024 +0530

Initial commit

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git log --oneline

c527478 (HEAD -> master) Initial commit

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ ls

index.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ vi index.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory

)

modified: index.txt

no changes added to commit (use "git add" and/or "git commit -a")

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git commit -m "minor bug fixes"

[master 4daae3c] minor bug fixes

1 file changed, 2 insertions(+), 1 deletion(-)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git log --oneline

4daae3c (HEAD -> master) minor bug fixes

c527478 Initial commit

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ ls

index.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git checkout login\_feature

Switched to branch 'login\_feature'

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (login\_feature)

$ git log --oneline

64f9dcf (HEAD -> login\_feature) Implement login feature ui

c527478 Initial commit

**Whatever the commits you are making in the master it will be there in the master branch only whatever the commits you are making in the branch login it will be there in the login branch only**

**List of all the branch**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (login\_feature)

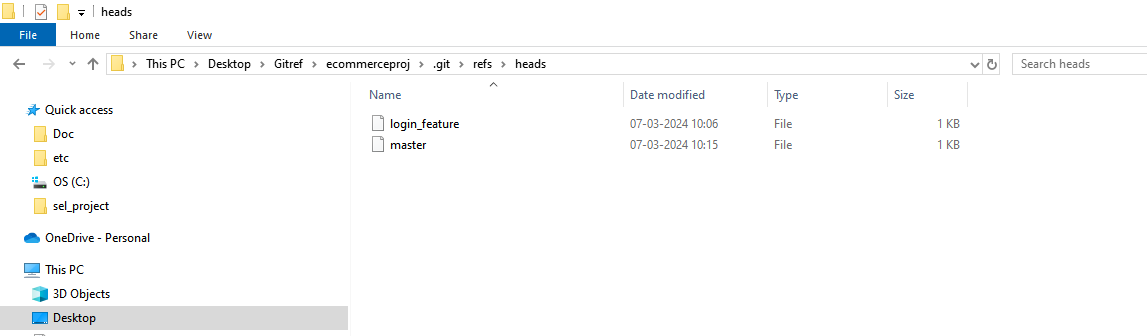
$ git branch

\* login\_feature -> current branch will show the green color

master

#### Rename and Delete branch using -m and -d commands :

Branches are nothing but references(refs folder)



**Create one new branch**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (login\_feature)

$ git branch dummybranch

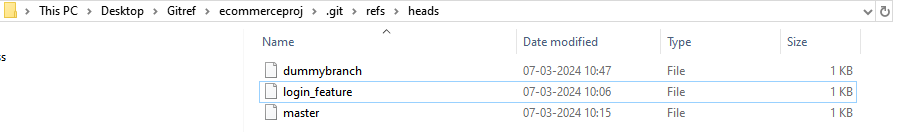
vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (login\_feature)

$ git branch

dummybranch

\* login\_feature

master



**What are contents present inside the branch ?**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (login\_feature)

$ cat .git/refs/heads/master

4daae3c6324f0ca554b635ad5d288c5d4e735fbc

**Here some hash will be there**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (login\_feature)

$ git log

commit 64f9dcf4ee0a2741360c6dc89e2ad45decdbabfb (HEAD -> login\_feature

, dummybranch)

Author: varalaxmi ganta <varalaxmiganta066@gmail.com>

Date: Thu Mar 7 10:06:07 2024 +0530

Implement login feature ui

commit c527478df7ddf80fa6411b1d4826432b825b46b1

Author: varalaxmi ganta <varalaxmiganta066@gmail.com>

Date: Thu Mar 7 09:59:47 2024 +0530

Initial commit

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (login\_feature)

$ cat .git/refs/heads/login\_feature

64f9dcf4ee0a2741360c6dc89e2ad45decdbabfb

**See the above hash whatever the commit has it is the head in the login features so the same data is saved here**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (login\_feature)

$ cat .git/refs/heads/dummybranch

64f9dcf4ee0a2741360c6dc89e2ad45decdbabfb

**To change the master branch then check the commit once**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (login\_feature)

$ git checkout master

Switched to branch 'master'

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git log

commit 4daae3c6324f0ca554b635ad5d288c5d4e735fbc (HEAD -> master)

Author: varalaxmi ganta <varalaxmiganta066@gmail.com>

Date: Thu Mar 7 10:15:47 2024 +0530

minor bug fixes

commit c527478df7ddf80fa6411b1d4826432b825b46b1

Author: varalaxmi ganta <varalaxmiganta066@gmail.com>

Date: Thu Mar 7 09:59:47 2024 +0530

Initial commit

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ cat .git/refs/heads/master

4daae3c6324f0ca554b635ad5d288c5d4e735fbc

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git branch

dummybranch

login\_feature

\* master

**Here master is shows green color that mean current branch but how come this git able to know that I am present in master branch ?**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ cat .git/HEAD

ref: refs/heads/master

**If I shift the branch**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git checkout dummybranch

Switched to branch 'dummybranch'

**Know if I check the head file it’s referencing to the dummy branch**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (dummybranch)

$ cat .git/HEAD

ref: refs/heads/dummybranch

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (dummybranch)

$ git branch

\* dummybranch

login\_feature

master

**To rename this dummy branch :**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (dummybranch)

$ git branch -m new\_branch

Automatically the dummy branch renamed to the new\_branch

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (new\_branch)

$ git branch

login\_feature

master

\* new\_branch

**To delete a branch :**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (new\_branch)

$ git branch -d new\_branch

error: cannot delete branch 'new\_branch' used by worktree at 'C:/Users/v

lab/Desktop/Gitref/ecommerceproj'

What it is trying to tell it is telling that cannot delete the branch because we are already existing in that branch So

Change to other branch

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (new\_branch)

$ git checkout master

Switched to branch 'master'

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git branch -d new\_branch

error: the branch 'new\_branch' is not fully merged

hint: If you are sure you want to delete it, run 'git branch -D new\_bran

ch'

hint: Disable this message with "git config advice.forceDeleteBranch fal

se"

**But it is not fully matched**

**For example**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git branch delete\_branch

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git branch -d delete\_branch

Deleted branch delete\_branch (was 4daae3c).

**It’s automatically deleted so know create the same branch and moved to the delete branch**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git branch delete\_branch

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git checkout delete\_branch

Switched to branch 'delete\_branch'

Modify one file and then commit it

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (delete\_branch)

$ vi index.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (delete\_branch)

$ git status

On branch delete\_branch

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: index.txt

no changes added to commit (use "git add" and/or "git commit -a")

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (delete\_branch)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (delete\_branch)

$ git commit -m "test commit"

[delete\_branch 2474f1b] test commit

1 file changed, 2 insertions(+), 1 deletion(-)

**Know we are commited into the delete\_branch then move into the master branch**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (delete\_branch)

$ git checkout master

Switched to branch 'master'

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git branch -d delete\_branch

error: the branch 'delete\_branch' is not fully merged

hint: If you are sure you want to delete it, run 'git branch -D delete\_b

ranch'

hint: Disable this message with "git config advice.forceDeleteBranch fal

se"

**It shows same not fully matched that means some changes are done then it shows**

If you are sure you want to delete it, run 'git branch **-D** delete\_b

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git branch -D delete\_branch

Deleted branch delete\_branch (was 2474f1b).

### Git Merge :

Merging the two branches

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git branch

login\_feature

\* master

new\_branch

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git status

On branch master

nothing to commit, working tree clean

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git checkout -b new\_feature

Switched to a new branch 'new\_feature'

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (new\_feature)

$ ls

index.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (new\_feature)

$ vi new.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (new\_feature)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (new\_feature)

$ git commit -m "new file added"

[new\_feature 772a43a] new file added

1 file changed, 1 insertion(+)

create mode 100644 new.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (new\_feature)

$ git checkout master

Switched to branch 'master'

Here master and new\_feature will be merged

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git merge new\_feature

Updating 4daae3c..772a43a

Fast-forward

new.txt | 1 +

1 file changed, 1 insertion(+)

create mode 100644 new.txt

**It shows to fast forward because there is no commits in the master then it called as fast forward**

**Create another branch**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ ls

index.txt new.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ vi index.txt

**-am is used to specified both add and message**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git commit -am "indexfile will be changed"

[master 7a8b73c] indexfile will be changed

1 file changed, 1 insertion(+)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git status

On branch master

nothing to commit, working tree clean

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git branch

login\_feature

\* master

new\_branch

new\_feature

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ ls

dummy.txt index.txt new.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git checkout -b dummy\_branch

Switched to a new branch 'dummy\_branch'

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (dummy\_branch)

$ vi dummy1.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (dummy\_branch)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (dummy\_branch)

$ git commit -m "modified one dummy file"

[dummy\_branch 2ca95e6] modified one dummy file

1 file changed, 1 insertion(+)

create mode 100644 dummy1.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (dummy\_branch)

$ git checkout master

Switched to branch 'master'

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ vi index.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git commit -m "modifed index.txt file"

[master 32bb04d] modifed index.txt file

1 file changed, 2 insertions(+), 1 deletion(-)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git merge dummy\_branch

Merge made by the 'ort' strategy.

dummy1.txt | 1 +

1 file changed, 1 insertion(+)

create mode 100644 dummy1.txt

**If we commit changes in the dummy branch and master branch , merge it then it’s shows the ort strategy or recursive strategy**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/ecommerceproj (master)

$ git log

commit aec2d9d72d457c776748c59acb4c1fe12be1d6fa (HEAD -> master)

Merge: 32bb04d 2ca95e6

Author: varalaxmi ganta <varalaxmiganta066@gmail.com>

Date: Thu Mar 7 11:40:48 2024 +0530

Merge branch 'dummy\_branch' -> T**his is the new commit after two branches are merged**

commit 32bb04d943db5ed912cbd37d01bd5c78c5957b6d

Author: varalaxmi ganta <varalaxmiganta066@gmail.com>

Date: Thu Mar 7 11:40:30 2024 +0530

modifed index.txt file

commit 2ca95e6476f245361230c75d784041229085c638 (dummy\_branch)

Author: varalaxmi ganta <varalaxmiganta066@gmail.com>

Date: Thu Mar 7 11:39:19 2024 +0530

modified one dummy file

commit 0c30fc7b82275be211ae3f85083c1fbf32464ad5

Author: varalaxmi ganta <varalaxmiganta066@gmail.com>

Date: Thu Mar 7 11:33:33 2024 +0530

**If change the same file in the master then conflicts can be occurred**

#### Merge Conflicts :

In the master branch if one file has been changed and also in the branch which you are going to merge into the master the same files will be changed with that branch then the git will not do the automatic merging then conflicts may raised.

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref (master)

$ mkdir mergeConflicts

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref (master)

$ cd mergeConflicts

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/mergeConflicts (master)

$ git init

Initialized empty Git repository in C:/Users/vlab/Desktop/Gitref/mergeC

onflicts/.git/

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/mergeConflicts (master)

$ vi index.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/mergeConflicts (master)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/mergeConflicts (master)

$ git commit -m "index file add"

[master (root-commit) 3703683] index file add

1 file changed, 1 insertion(+)

create mode 100644 index.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/mergeConflicts (master)

$ git status

On branch master

nothing to commit, working tree clean

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/mergeConflicts (master)

$ git log

commit 3703683349bca8d9f76dad5e4cc85ae997b1c6d6 (HEAD -> master)

Author: varalaxmi ganta <varalaxmiganta066@gmail.com>

Date: Thu Mar 7 11:54:18 2024 +0530

index file add

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/mergeConflicts (master)

$ git checkout -b feature/some\_feature

Switched to a new branch 'feature/some\_feature'

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/mergeConflicts (feature/soature)

$ vi index.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/mergeConflicts (feature/soature)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/mergeConflicts (feature/soature)

$ git commit -m "changes in index file feature"

[feature/some\_feature 6a09706] changes in index file feature

1 file changed, 1 insertion(+)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/mergeConflicts (feature/soature)

$ git checkout master

Switched to branch 'master'

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/mergeConflicts (master)

$ ls

index.txt

There is no content will be modified in the master branch

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/mergeConflicts (master)

$ cat index.txt

created one new file

Modify the same file in master branch

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/mergeConflicts (master)

$ vi index.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/mergeConflicts (master)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/mergeConflicts (master)

$ git commit -m "changes done in the master"

[master bb1e2f5] changes done in the master

1 file changed, 1 insertion(+)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/mergeConflicts (master)

$ git log

commit bb1e2f553122f756534c152927fae9e9456bb88e (HEAD -> master)

Author: varalaxmi ganta <varalaxmiganta066@gmail.com>

Date: Thu Mar 7 12:00:05 2024 +0530

changes done in the master

commit 3703683349bca8d9f76dad5e4cc85ae997b1c6d6

Author: varalaxmi ganta <varalaxmiganta066@gmail.com>

Date: Thu Mar 7 11:54:18 2024 +0530

index file add

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/mergeConflicts (master)

$ git merge feature/some\_feature

Auto-merging index.txt

CONFLICT (content): Merge conflict in index.txt

Automatic merge failed; fix conflicts and then commit the result.

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/mergeConflicts (master|MERGING)

$ git status

On branch master

You have unmerged paths.

(fix conflicts and run "git commit")

(use "git merge --abort" to abort the merge)

Unmerged paths:

(use "git add <file>..." to mark resolution)

both modified: index.txt

no changes added to commit (use "git add" and/or "git commit -a")

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/mergeConflicts (master|MERGING) -> Here master merging is going on

$ vi index.txt

created one new file

<<<<<<< HEAD

changes done in the master branch

=======

changes done in the feature branch

>>>>>>> feature/some\_feature

~

**Git wiill be automatically added the code in both master commit and branch commit**

**HEAD is currently pointing to the master branch**

**Delete some line whatever we don’t want**

**--abort is used to the stops the ongoing merge process and attempts to restore you per-merge state.**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/mergeConflicts (master|MERGING)

$ git merge --abort

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/mergeConflicts (master)

$ git merge feature/some\_feature

Auto-merging index.txt

CONFLICT (content): Merge conflict in index.txt

Automatic merge failed; fix conflicts and then commit the result.

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/mergeConflicts (master|MERGING)

$ vi index.txt

In this file contents have only master commit

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/mergeConflicts (master|MERGING)

$ cat index.txt

created one new file

changes done in the master branch

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/mergeConflicts (master|MERGING)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/mergeConflicts (master|MERGING)

$ git commit

[master 6cb5f1c] Merge branch 'feature/some\_feature'

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/mergeConflicts (master)

$ git log

commit 6cb5f1ce3196f4130d2ae9975640eadcc1b779ee (HEAD -> master)

Merge: bb1e2f5 6a09706

Author: varalaxmi ganta <varalaxmiganta066@gmail.com>

Date: Thu Mar 7 12:13:14 2024 +0530

Merge branch 'feature/some\_feature'

commit bb1e2f553122f756534c152927fae9e9456bb88e

Author: varalaxmi ganta <varalaxmiganta066@gmail.com>

Date: Thu Mar 7 12:00:05 2024 +0530

changes done in the master

commit 6a097068d60af8752b6d45827c8806ad8539aaf2 (feature/some\_feature)

Author: varalaxmi ganta <varalaxmiganta066@gmail.com>

Date: Thu Mar 7 11:56:19 2024 +0530

changes in index file feature

commit 3703683349bca8d9f76dad5e4cc85ae997b1c6d6

Author: varalaxmi ganta <varalaxmiganta066@gmail.com>

Date: Thu Mar 7 11:54:18 2024 +0530

:q

### Git Rebase :

* Rebase is often used as an alternative to merging.
* Rebasing a branch updates one branch with another by applying the commits of one branch on top of the commits of another branch.
* Git rebase is used to clean up our local commit history.
* Merge preserve history
* Rebase doesn’t preserve history

**Do not use Rebase when :**

The branch is public when it is shared to all the developers.

Most of the teams prefer merge over rebase.

**Common places where we use rebases :**

Cleaning up your commits before sharing your branch.

Pulling changes from another branch without merge.

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref (master)

$ mkdir rebseproj

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref (master)

$ cd rebseproj

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebseproj (master)

$ git init

Initialized empty Git repository in C:/Users/vlab/Desktop/Gitref/rebse

proj/.git/

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebseproj (master)

$ vi master.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebseproj (master)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebseproj (master)

$ git commit -m "master file added in master"

[master (root-commit) cf71ba5] master file added in master

1 file changed, 1 insertion(+)

create mode 100644 master.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebseproj (master)

$ git log --oneline

cf71ba5 (HEAD -> master) master file added in master

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebseproj (master)

$ git branch feature

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebseproj (master)

$ git branch

feature

\* master

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebseproj (master)

$ vi master1.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebseproj (master)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebseproj (master)

$ git commit -m "master1 file added in master"

[master 1d32fd1] master1 file added in master

1 file changed, 1 insertion(+)

create mode 100644 master1.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebseproj (master)

$ vi master2.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebseproj (master)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebseproj (master)

$ git commit -m "master 2 file added in master"

[master d852996] master 2 file added in master

1 file changed, 1 insertion(+)

create mode 100644 master2.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebseproj (master)

$ git log --oneline

d852996 (HEAD -> master) master 2 file added in master

1d32fd1 master1 file added in master

cf71ba5 (feature) master file added in master

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebseproj (master)

$ git checkout feature

Switched to branch 'feature'

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebseproj (feature)

$ vi feature1.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebseproj (feature)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebseproj (feature)

$ git commit -m "feature 1 added in feature"

[feature 939a275] feature 1 added in feature

1 file changed, 1 insertion(+)

create mode 100644 feature1.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebseproj (feature)

$ vi feature2.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebseproj (feature)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebseproj (feature)

$ git commit -m "feature 2 file addedin feature"

[feature 4174079] feature 2 file addedin feature

1 file changed, 1 insertion(+)

create mode 100644 feature2.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebseproj (feature)

$ git log --oneline

4174079 (HEAD -> feature) feature 2 file addedin feature

939a275 feature 1 added in feature

cf71ba5 master file added in master

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebseproj (feature)

$ git checkout master

Switched to branch 'master'

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebseproj (master)

$ git log --oneline

d852996 (HEAD -> master) master 2 file added in master

1d32fd1 master1 file added in master

cf71ba5 master file added in master

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebseproj (master)

$ git rebase feature

Successfully rebased and updated refs/heads/master.

Here all commits are shows in master branch only

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebseproj (master)

$ git log --oneline

a2f714d (HEAD -> master) master 2 file added in master

0ea6393 master1 file added in master

4174079 (feature) feature 2 file addedin feature

939a275 feature 1 added in feature

cf71ba5 master file added in master

#### Interactive Rebasing :

Rebase is used to combine the multiple commits to single commit

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref (master)

$ mkdir git\_irebasing

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref (master)

$ cd git\_irebasing/

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git\_irebasing (master)

$ git init

Initialized empty Git repository in C:/Users/vlab/Desktop/Gitref/git\_irebasing/.gi

t/

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git\_irebasing (master)

$ vi home.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git\_irebasing (master)

$ git add .

i

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git\_irebasing (master)

$ git commit -m "home page added"

[master (root-commit) 1da5c81] home page added

1 file changed, 1 insertion(+)

create mode 100644 home.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git\_irebasing (master)

$ git checkout -b new\_feature

Switched to a new branch 'new\_feature'

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git\_irebasing (new\_feature)

$ vi feature.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git\_irebasing (new\_feature)

$ git add .

g

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git\_irebasing (new\_feature)

$ git commit -m "still working on the gfeature"

[new\_feature ed0c7d9] still working on the gfeature

1 file changed, 1 insertion(+)

create mode 100644 feature.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git\_irebasing (new\_feature)

$ vi feature.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git\_irebasing (new\_feature)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git\_irebasing (new\_feature)

$ git commit -m "fix logic issues"

[new\_feature 242faca] fix logic issues

1 file changed, 1 insertion(+)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git\_irebasing (new\_feature)

$ vi feature.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git\_irebasing (new\_feature)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git\_irebasing (new\_feature)

$ git commit -m "completed new feature"

[new\_feature f2e5cc5] completed new feature

1 file changed, 1 insertion(+)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git\_irebasing (new\_feature)

$ git log --oneline

f2e5cc5 (HEAD -> new\_feature) completed new feature

242faca fix logic issues

ed0c7d9 still working on the gfeature

1da5c81 (master) home page added

These three commit are converted into single commit or implemented a new feature

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git\_ireabsing (new\_feature)

$ git rebase -i master

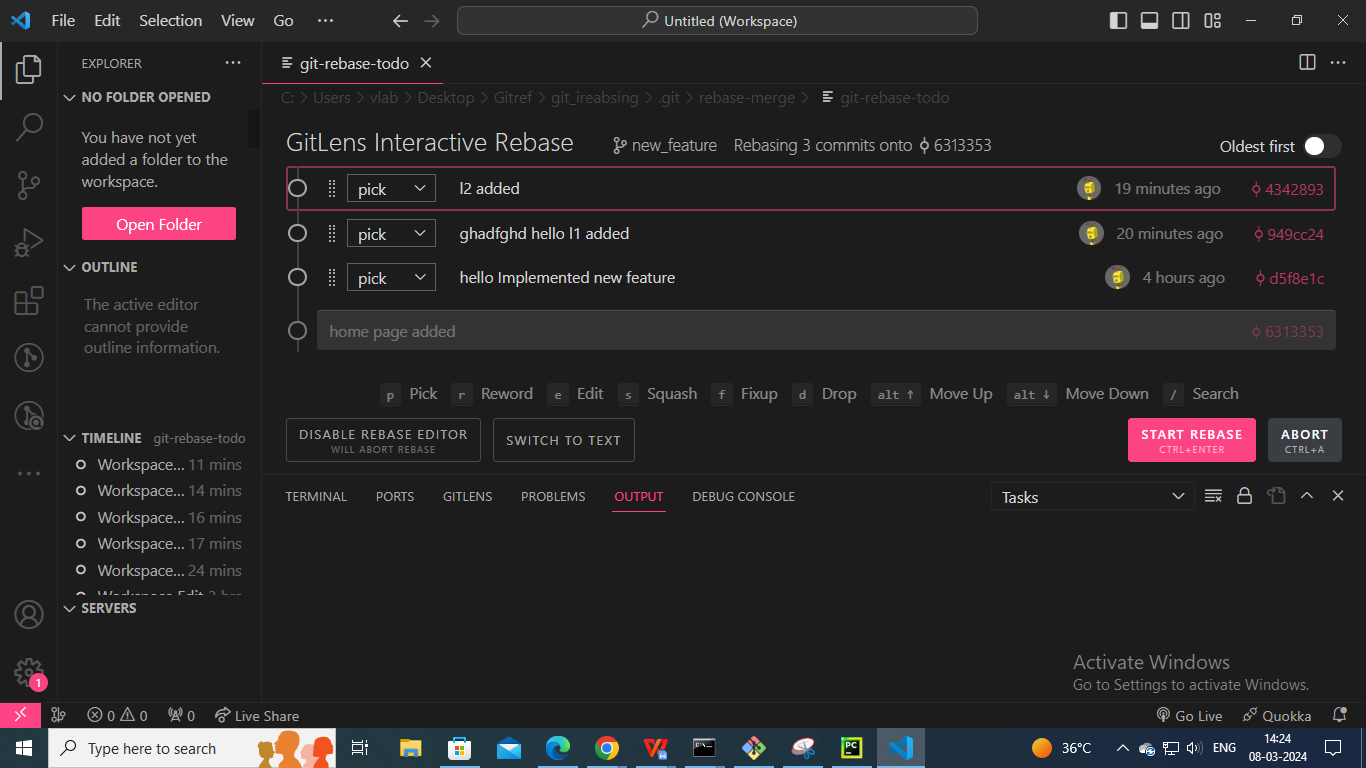
[detached HEAD 8b2d7b5] Implemented new feature

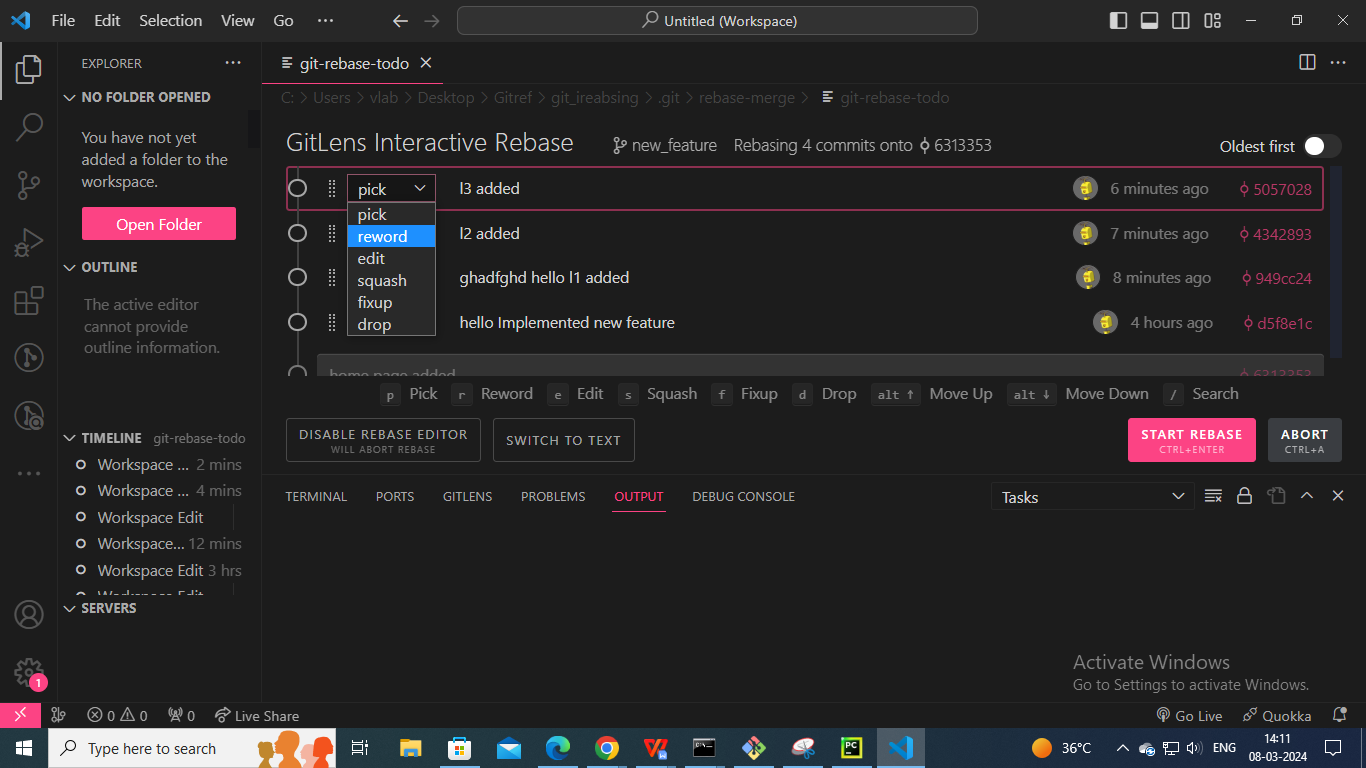
Date: Fri Mar 8 09:59:14 2024 +0530

1 file changed, 4 insertions(+)

create mode 100644 feature.txt

Successfully rebased and updated refs/heads/new\_feature.





**Note :**

* If we see all commits in the present branch , then give the first commit ID git rebase -i 1da5c81
* all the commits we have to delete, but the root commit isn't deleted(To select the drop down box).

In above figure shows pick drop-down box

* **Reward** : Edit particular commit and combine multiple commits to one commit in the master branch.
* **Drop** : Used to remove the particular commit and also Deleted that file in all stages (working area , staging area and repository).
* **Fixup** : Used to detach the commit in a particular files but these files are not Deleted available in the current branch

* **Edit :** same used as reward but when we close that file then it shows **you can amend the commit with some commands**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git\_ireabsing (new\_feature)

$ git rebase -i master

Stopped at d5f8e1c... hello Implemented new feature

You can amend the commit now, with

git commit --amend

Once you are satisfied with your changes, run

git rebase --continue

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git\_ireabsing (new\_feature|REBASE 1/2)

$ git commit --amend

[detached HEAD b33020a] hiii Implemented new feature

Date: Fri Mar 8 09:59:14 2024 +0530

1 file changed, 4 insertions(+)

create mode 100644 feature.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git\_ireabsing (new\_feature|REBASE 1/2)

$ ls

feature.txt home.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git\_ireabsing (new\_feature|REBASE 1/2)

$ git rebase --continue

Successfully rebased and updated refs/heads/new\_feature.

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git\_ireabsing (new\_feature)

$ ls

feature.txt home.txt l1.txt l2.txt

* **Squash** : Two combined multiple commits into single commits .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git\_ireabsing (new\_feature)

$ ls

feature.txt home.txt l1.txt l2.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git\_ireabsing (new\_feature)

$ git rebase -i master

[detached HEAD f1f50a2] l1 and l2 commited

Date: Fri Mar 8 14:03:48 2024 +0530

2 files changed, 4 insertions(+)

create mode 100644 l1.txt

create mode 100644 l2.txt

Successfully rebased and updated refs/heads/new\_feature.

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git\_ireabsing (new\_feature)

$ git log --oneline

f1f50a2 (HEAD -> new\_feature) l1 and l2 commited

d5f8e1c hello Implemented new feature

6313353 (master) home page added

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git\_ireabsing (new\_feature)

$ git log --oneline

8b2d7b5 (HEAD -> new\_feature) Implemented new feature

6313353 (master) home page added

#### Modify the latest commit :

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref (master)

$ mkdir change

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref (master)

$ cd change

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/change (master)

$ git init

Initialized empty Git repository in C:/Users/vlab/Desktop/Gitref/chang

e/.git/

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/change (master)

$ vi index.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/change (master)

$ git status

On branch master

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

index.txt

nothing added to commit but untracked files present (use "git add" to

track)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/change (master)

$ git log

fatal: your current branch 'master' does not have any commits yet

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/change (master)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/change (master)

$ git commit -m "Initial commit"

[master (root-commit) a91dc3e] Initial commit

1 file changed, 1 insertion(+)

create mode 100644 index.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/change (master)

$ cat index.txt

hii veena

**If I want change the content in the index.txt file, we have to get another commit to modify so that we can use the initial commit.**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/change (master)

$ git log

commit a91dc3edc629e0cbb13a9aab259e8b613585503e (HEAD -> master)

Author: varalaxmi ganta <varalaxmiganta066@gmail.com>

Date: Fri Mar 8 10:47:32 2024 +0530

Initial commit

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/change (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory

)

modified: index.txt

no changes added to commit (use "git add" and/or "git commit -a")

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/change (master)

$ cat index.txt

hello veena

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/change (master)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/change (master)

$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: index.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/change (master)

$ git show a91dc

commit a91dc3edc629e0cbb13a9aab259e8b613585503e (HEAD -> master)

Author: varalaxmi ganta <varalaxmiganta066@gmail.com>

Date: Fri Mar 8 10:47:32 2024 +0530

Initial commit

diff --git a/index.txt b/index.txt

new file mode 100644

index 0000000..4b4e84b

--- /dev/null

+++ b/index.txt

@@ -0,0 +1 @@

+hii veena

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/change (master)

$ git diff --cached

diff --git a/index.txt b/index.txt

index 4b4e84b..c469a19 100644

--- a/index.txt

+++ b/index.txt

@@ -1 +1 @@

-hii veena

+hello veena

If it is local only then use amend it went into the public branch it may confusion

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/change (master)

$ git commit --amend

[master 229f8a2] Initial commited

Date: Fri Mar 8 10:47:32 2024 +0530

1 file changed, 1 insertion(+)

create mode 100644 index.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/change (master)

$ git status

On branch master

nothing to commit, working tree clean

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/change (master)

$ git log --oneline

229f8a2 (HEAD -> master) Initial commited

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/change (master)

$ git show 229f8a

commit 229f8a2322ce641462c2e78cc90a1786b0d7d7a8

Author: varalaxmi ganta <varalaxmiganta066@gmail.com>

Date: Fri Mar 8 10:47:32 2024 +0530

Initial commited

diff --git a/index.txt b/index.txt

new file mode 100644

index 0000000..c469a19

--- /dev/null

+++ b/index.txt

@@ -0,0 +1 @@

+hello veena

#### Conflict :

#### If one file has three commits (a, b, and c), if you want to delete the b commit, then conflict may occur. to rectify that by using rebase --continue after adding and committing that file.

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ vi f1

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git init

Initialized empty Git repository in C:/Users/vlab/Desktop/Gitref/rebase1/.git/

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git commit -m "create one new file"

[master (root-commit) 8a6551d] create one new file

1 file changed, 1 insertion(+)

create mode 100644 f1

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git log --oneline

8a6551d (HEAD -> master) create one new file

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git switch -c branch1

Switched to a new branch 'branch1'

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ ls

f1

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ vi f2

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git commit -m "created one f2 file"

[branch1 94a59a0] created one f2 file

1 file changed, 1 insertion(+)

create mode 100644 f2

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ vi f2

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git commit -m "modify 1 f2"

[branch1 85847fc] modify 1 f2

1 file changed, 2 insertions(+), 1 deletion(-)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ vi f2

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git commit -m "modify 2 f2"

[branch1 444fe35] modify 2 f2

1 file changed, 2 insertions(+), 1 deletion(-)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git log --oneline

444fe35 (HEAD -> branch1) modify 2 f2

85847fc modify 1 f2

94a59a0 created one f2 file

8a6551d (master) create one new file

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git rebase -i master

Auto-merging f2

CONFLICT (content): Merge conflict in f2

error: could not apply 444fe35... modify 2 f2

hint: Resolve all conflicts manually, mark them as resolved with

hint: "git add/rm <conflicted\_files>", then run "git rebase --continue".

hint: You can instead skip this commit: run "git rebase --skip".

hint: To abort and get back to the state before "git rebase", run "git rebase --

abort".

Could not apply 444fe35... modify 2 f2

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1|REBASE 3/3)

$ git rebase --continue

f2: needs merge

You must edit all merge conflicts and then

mark them as resolved using git add

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1|REBASE 3/3)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1|REBASE 3/3)

$ git commit -m "conflict fixed"

[detached HEAD b15bfd5] conflict fixed

1 file changed, 6 insertions(+)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1|REBASE 3/3)

$ git log --oneline

b15bfd5 (HEAD) conflict fixed

94a59a0 created one f2 file

8a6551d (master) create one new file

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1|REBASE 3/3)

$ git rebase --abort

**--abort is used to terminate the re basing.**

#### Case1:

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ vi f1

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git init

Initialized empty Git repository in C:/Users/vlab/Desktop/Gitref/rebase1/.git/

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git commit -am "Created f1"

On branch master

Initial commit

Untracked files:

(use "git add <file>..." to include in what will be committed)

f1

nothing added to commit but untracked files present (use "git add" to track)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git commit -am "Created f1"

[master (root-commit) 3ae5c43] Created f1

1 file changed, 1 insertion(+)

create mode 100644 f1

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ vi f1

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git commit -m "modify1 f1"

[master 817f837] modify1 f1

1 file changed, 2 insertions(+), 1 deletion(-)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ vi f1

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git commit -m "modify2 f1"

[master 5860ec2] modify2 f1

1 file changed, 1 insertion(+)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git log --oneline

5860ec2 (HEAD -> master) modify2 f1

817f837 modify1 f1

3ae5c43 Created f1

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git rebase -i master

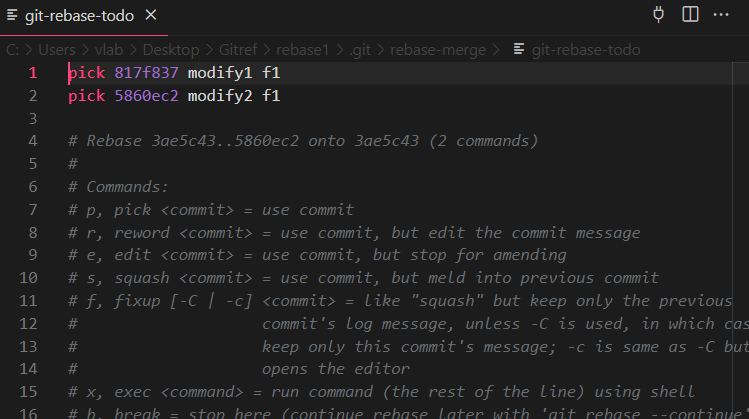
Successfully rebased and updated refs/heads/master.



vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git rebase -i 3ae5c43

Successfully rebased and updated refs/heads/master.

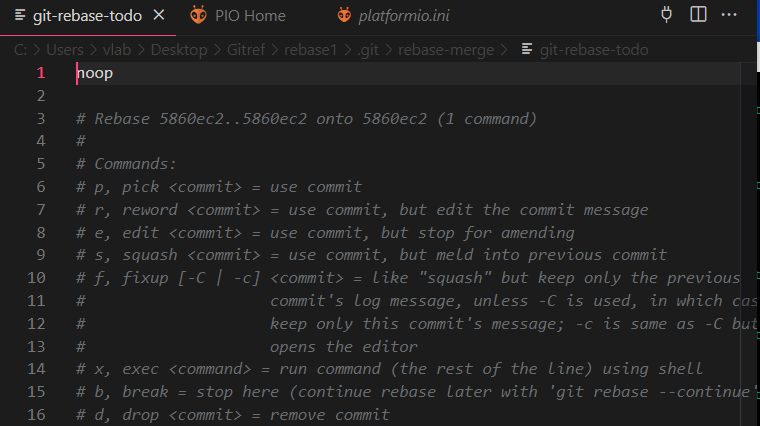


The above figure shows two commits because 3ae5c43 is the first commit. After that commit, there are two new commits and used any one  required pick-outs.

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git rebase -i 5860ec2

Successfully rebased and updated refs/heads/master.



The above figure shows noop (no operation) because there are no commits after that commit, and the head will point to that commit only.

#### Case 2 :

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git log --oneline --graph

\* 5860ec2 (HEAD -> master) modify2 f1

\* 817f837 modify1 f1

\* 3ae5c43 Created f1

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git checkout -b branch1

Switched to a new branch 'branch1'

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git log --oneline --graph

\* 5860ec2 (HEAD -> branch1, master) modify2 f1

\* 817f837 modify1 f1

\* 3ae5c43 Created f1

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git rebase -i master

Successfully rebased and updated refs/heads/branch1.



The above figure shows noop because there are no changes in the currebt branch.

#### Case 3:

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ vi f2

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git commit -m "created f2"

[branch1 5faab4f] created f2

1 file changed, 1 insertion(+)

create mode 100644 f2

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ vi f2

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git commit -m "modify1 f2"

[branch1 7860294] modify1 f2

1 file changed, 1 insertion(+)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ vi f2

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git commit -m "modify2 f2"

[branch1 ac2c589] modify2 f2

1 file changed, 1 insertion(+)

Here master will pointed to the modify f1 and branch1 has modify f2

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git log --oneline

ac2c589 (HEAD -> branch1) modify2 f2

7860294 modify1 f2

5faab4f created f2

5860ec2 (master) modify2 f1

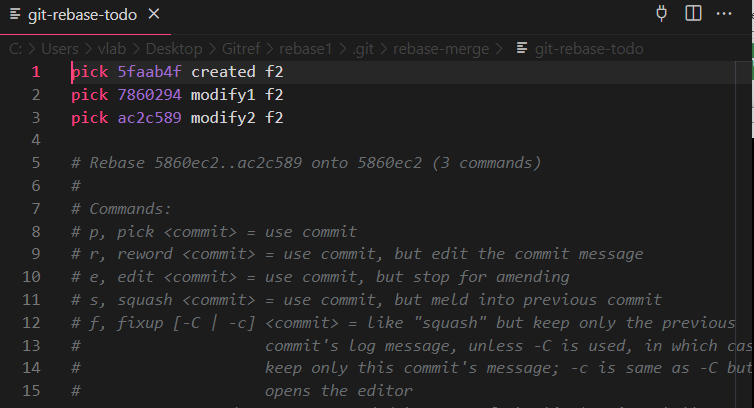
817f837 modify1 f1

3ae5c43 Created f1

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git rebase -i master

Successfully rebased and updated refs/heads/branch1.



Here master has three commits and switches to branch 1, which has three commits. These three commits were created after master brach commits.

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git checkout master

Switched to branch 'master'

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git log --oneline

5860ec2 (HEAD -> master) modify2 f1

817f837 modify1 f1

3ae5c43 Created f1

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git rebase -i branch1

Successfully rebased and updated refs/heads/master.

In the above command shows the no commit because there is no new commit and also merge .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git log --oneline

ac2c589 (HEAD -> branch1, master) modify2 f2

7860294 modify1 f2

5faab4f created f2

5860ec2 modify2 f1

817f837 modify1 f1

3ae5c43 Created f1

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ vi f2

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git commit -m "again modify f2"

[branch1 5139907] again modify f2

1 file changed, 2 insertions(+), 1 deletion(-)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git checkout master

Switched to branch 'master'

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git rebase -i branch1

Successfully rebased and updated refs/heads/master.

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git log --oneline

5139907 (HEAD -> master, branch1) again modify f2

ac2c589 modify2 f2

7860294 modify1 f2

5faab4f created f2

5860ec2 modify2 f1

817f837 modify1 f1

3ae5c43 Created f1

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git log --oneline

5139907 (HEAD -> branch1, master) again modify f2

ac2c589 modify2 f2

7860294 modify1 f2

5faab4f created f2

5860ec2 modify2 f1

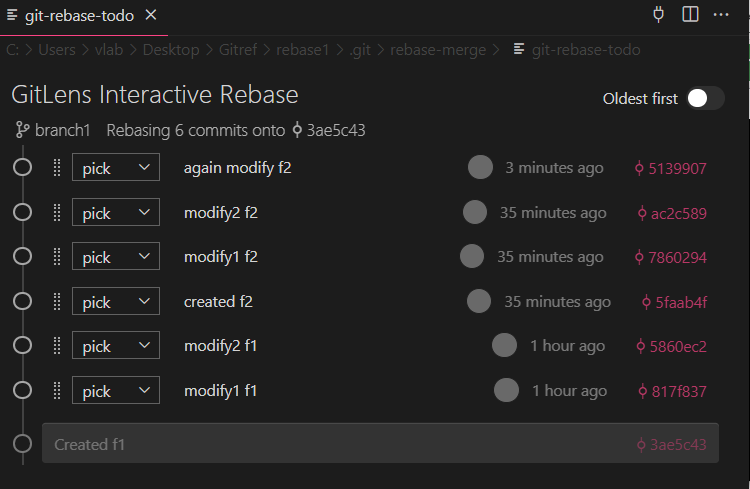
817f837 modify1 f1

3ae5c43 Created f1

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git rebase -i 3ae5c43

Successfully rebased and updated refs/heads/branch1.



#### Case4:

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ ls

f1

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git log --oneline

9f2b907 (HEAD -> master) add one file

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ vi f1

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git commit -m "modify f1 file"

[master d3490fd] modify f1 file

1 file changed, 2 insertions(+), 1 deletion(-)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ vi f1

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git commit -m "modify f1 file"

[master d34c443] modify f1 file

1 file changed, 1 insertion(+), 1 deletion(-)

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git log --oneline

d34c443 (HEAD -> master) modify f1 file

d3490fd modify f1 file

9f2b907 add one file

**If use this command all the commits are removed including root commit also**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git update-ref -d HEAD

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git log --oneline

fatal: your current branch 'master' does not have any commits yet

**master branch has no commit, but if we want to change the master branch to another branch, it shows errors. so we can commit the changes or stash them, then only move to another branch.**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git switch branch1

error: Your local changes to the following files would be overwritten by

checkout:

f1

Please commit your changes or stash them before you switch branches.

Aborting

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git status

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: f1

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git commit -m "delete"

[master (root-commit) e30514e] delete

1 file changed, 2 insertions(+)

create mode 100644 f1

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git switch branch1

Switched to branch 'branch1'

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git log --oneline

5139907 (HEAD -> branch1) again modify f2

ac2c589 modify2 f2

7860294 modify1 f2

5faab4f created f2

5860ec2 modify2 f1

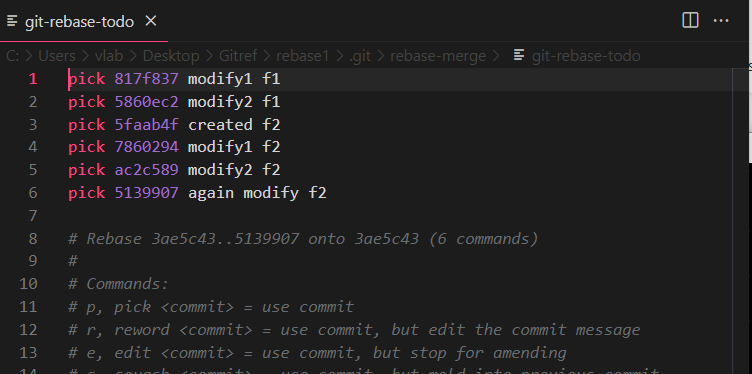
817f837 modify1 f1

3ae5c43 Created f1

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git rebase -i 817f837^

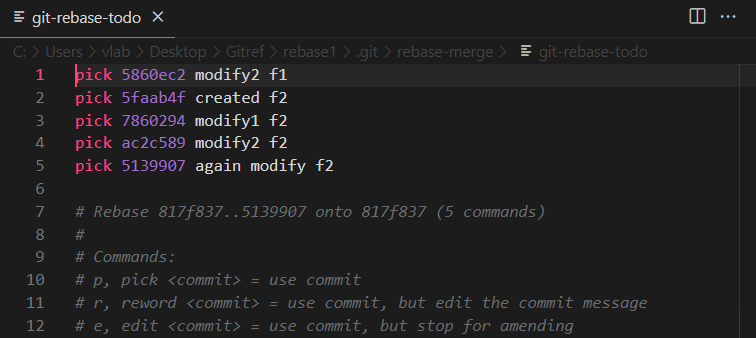
Successfully rebased and updated refs/heads/branch1.



vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git rebase -i 817f837

Successfully rebased and updated refs/heads/branch1.



#### Case 4: **By using update-ref delete all commit history**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git log --oneline

5139907 (HEAD -> branch1) again modify f2

ac2c589 modify2 f2

7860294 modify1 f2

5faab4f created f2

5860ec2 modify2 f1

817f837 modify1 f1

3ae5c43 Created f1

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git update-ref -d HEAD

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (branch1)

$ git log --oneline

fatal: your current branch 'branch1' does not have any commits yet

#### Case 5:

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git log --oneline

7f3bfe5 (HEAD -> master) delete

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

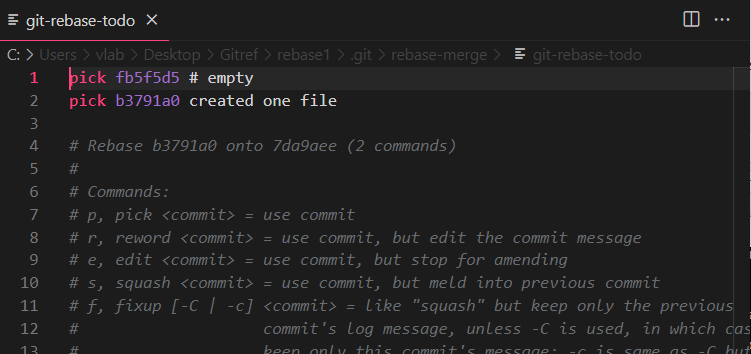
$ git rebase -i --root

Successfully rebased and updated refs/heads/master.

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git log --oneline

fb5f5d5 (HEAD -> master)



vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ vi f1

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git commit -m "created one file"

[master b3791a0] created one file

1 file changed, 1 insertion(+)

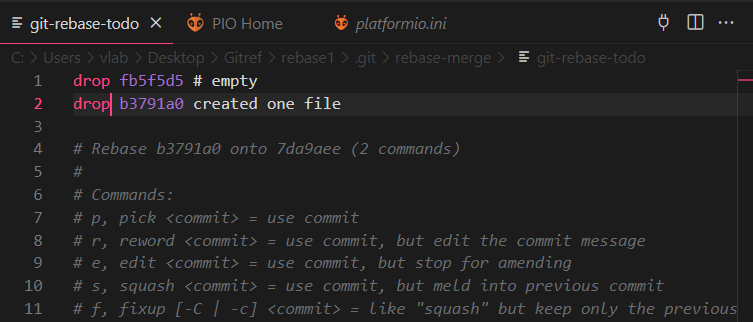
create mode 100644 f1

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git log --oneline

b3791a0 (HEAD -> master) created one file

fb5f5d5



vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git rebase -i --root

Successfully rebased and updated refs/heads/master.

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/rebase1 (master)

$ git log --oneline

7da9aee (HEAD -> master)

### GIT Cherry Pick

Cherry Pick is used if more commits are available in one branch and you want to apply particular commit from one branch into another branch.

Cherry pick is mainly used if you don’t want to merge the whole branch and you want some of the commits.

Cherry pick just like rebase it is and advanced concept and also a powerful command.

Cherry pick is a useful tool, but always it is not a good option.

It can cause duplicate commits.

Mainly cherry pick is used for the bug fixes where you want to place that bugfix commit in all the version branches.

It is also used when we accidentally made a commit in wrong branch.

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref (master)

$ mkdir git-cherrypick

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref (master)

$ cd git-cherrypick

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (master)

$ git init

Initialized empty Git repository in C:/Users/vlab/Desktop/Gitref/git-cherrypic

k/.git/

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (master)

$ ls

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (master)

$ vi index.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (master)

$ git add .

g

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (master)

$ git commit -m "Initial commit"

[master (root-commit) 1c7cd88] Initial commit

1 file changed, 1 insertion(+)

create mode 100644 index.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (master)

$ git log --oneline

1c7cd88 (HEAD -> master) Initial commit

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (master)

$ git branch 1.0

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (master)

$ git branch 2.0

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (master)

$ git branch 3.0

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (master)

$ git branch

1.0

2.0

3.0

\* master

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (master)

$ git checkout 3.0

Switched to branch '3.0'

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (3.0)

$ vi feature.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (3.0)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (3.0)

$ git commit -m "working on the 3.0 feature"

[3.0 e660c81] working on the 3.0 feature

1 file changed, 1 insertion(+)

create mode 100644 feature.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (3.0)

$ git log --oneline

e660c81 (HEAD -> 3.0) working on the 3.0 feature

1c7cd88 (master, 2.0, 1.0) Initial commit

**Here we find one bug in this 3.0 version and also having the same bug in the 1.0 and 2.0 versions.**

**For example :**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (3.0)

$ vi bugfix.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (3.0)

$ git add .

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (3.0)

$ git commit -m "bug fix done"

[3.0 7af3a73] bug fix done

1 file changed, 1 insertion(+)

create mode 100644 bugfix.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (3.0)

$ git log --oneline

7af3a73 (HEAD -> 3.0) bug fix done

e660c81 working on the 3.0 feature

1c7cd88 (master, 2.0, 1.0) Initial commit

**Here we fixed the bug and add it to the 1.0 and 2.0 but not merge the 1.0 , 2.0 and 3.0 because already still working in this feature**

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (3.0)

$ git checkout 2.0

Switched to branch '2.0'

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (2.0)

$ git log --oneline

1c7cd88 (HEAD -> 2.0, master, 1.0) Initial commit

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (2.0)

$ ls

index.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (2.0)

$ git cherry-pick 7af3a73

[2.0 10551e4] bug fix done

Date: Fri Mar 8 15:53:50 2024 +0530

1 file changed, 1 insertion(+)

create mode 100644 bugfix.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (2.0)

$ ls

bugfix.txt index.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (2.0)

$ git log --oneline

10551e4 (HEAD -> 2.0) bug fix done

1c7cd88 (master, 1.0) Initial commit

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (2.0)

$ git checkout master

Switched to branch 'master'

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (master)

$ git checkout 1.0

Switched to branch '1.0'

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (1.0)

$ ls

index.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (1.0)

$ git log --oneline

1c7cd88 (HEAD -> 1.0, master) Initial commit

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (1.0)

$ git cherry-pick 7af3a73

[1.0 d0ec93d] bug fix done

Date: Fri Mar 8 15:53:50 2024 +0530

1 file changed, 1 insertion(+)

create mode 100644 bugfix.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (1.0)

$ ls

bugfix.txt index.txt

vlab@VEDA2F118 MINGW32 ~/Desktop/Gitref/git-cherrypick (1.0)

$ git log --oneline

d0ec93d (HEAD -> 1.0) bug fix done

1c7cd88 (master) Initial commit

### GIT Head :

When we are working we only checkout one branch at a time.This is also called as Head branch.

Git makes note of this branch and store it in .git/Head as the reference for the path of the branch.

Head is nothing but the reference to the branch.

Head not only reference a branch it also reference the commit sha1.

If the head points to a specific commit then it is called as detached head.

https://docs.platformio.org/en/latest/tutorials/ststm32/stm32cube\_debugging\_unit\_testing.html