

Assignment:-

1) **git init** : Create empty Git repo in specified directory. Run with no arguments to initialize the current directory as a git repository.

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24
$ git init
Initialized empty Git repository in C:/Users/lenovo/Desktop/New folder (2)/June24/.git/

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ |
```

2) **git config --global user.name <name>**: Define author name to be used for all commits in current repo.

git config --global user.email <email>: Define author email-id to be used for all commits in current repo.

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24
$ git config --global user.name "preeti"

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24
$ git config --global user.email "preetipatil094@gmail.com"

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24
$
```

3) git add [file/directory] : Stage all changes in for the next commit. Replace with a to change a specific file.

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24
$ git config --global user.name "preeti"

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24
$ git config --global user.email "preetipatil094@gmail.com"

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24
$ git init
Initialized empty Git repository in C:/Users/lenovo/Desktop/New folder (2)/June24/.git/

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ touch file1 file2

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ ls
file1 file2

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ git add file1

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ git add file2

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   file1
        new file:   file2
```

4) git commit -m "<message>" : Commit the staged snapshot, but instead of launching a text editor, use as the commit message.

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24
$ git config --global user.name "preeti"

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24
$ git config --global user.email "preetipatil094@gmail.com"

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24
$ git init
Initialized empty Git repository in C:/Users/lenovo/Desktop/New folder (2)/June24/.git/

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ touch file1 file2

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ ls
file1 file2

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ git add file1

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ git add file2

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   file1
        new file:   file2

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ git commit -m "first commit"
[master (root-commit) 26503dd] first commit
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file1
 create mode 100644 file2

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
```

5) git status : List which files are staged, unstaged, and untracked.

```
MINGW64/c/Users/lenovo/Desktop/New folder (2)/June24
lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24
$ git config --global user.name "preeti"
lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24
$ git config --global user.email "preetipatil094@gmail.com"
lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24
$ git init
Initialized empty Git repository in C:/Users/lenovo/Desktop/New folder (2)/June24/.git/
lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ touch file1 file2
lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ ls
file1 file2
lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ git add file1
lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ git add file2
lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ git status
On branch master

no commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   file1
        new file:   file2

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$
```

6) git log : Display the entire commit history using the default format. For customization see additional options.

```
MINGW64/c/Users/lenovo/Desktop/New folder (2)/June24
lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ git log
commit 26503ddd77c28ea1e9d641eeb7151e02b34eae16 (HEAD -> master)
Author: preeti <preetipatil094@gmail.com>
Date:   Fri Aug 2 01:25:59 2024 +0530

    first commit

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ |
```

7) git branch : List all of the branches in your repo. Add a argument to create a new branch with the name .

```
MINGW64/c/Users/lenovo/Desktop/New folder (2)/June24

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ git log
commit 26503ddd77c28ea1e9d641eeb7151e02b34eae16 (HEAD -> master)
Author: preeti <preetipatil094@gmail.com>
Date:   Fri Aug 2 01:25:59 2024 +0530

    first commit

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ git branch feature

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ |
```

```
MINGW64/c/Users/lenovo/Desktop/New folder (2)/June24

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ git branch
  feature
* master

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ |
```

8) git checkout <branch> : switch to another branch.

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ git checkout feature
Switched to branch 'feature'

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (feature)
$ |
```

9) git checkout -b <branch> : Create and check out a new branch named . Drop the -b flag to checkout an existing branch.

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ git checkout -b feature4
Switched to a new branch 'feature4'

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (feature4)
$ |
```

10) git merge <branch> : Merge into the current branch.

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (feature)
$ git merge feature1
Already up to date.

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (feature)
$ |
```

11) git remote add [remote name] [URL] : creates a new connection to a remote repository and give it a name to serve as a shortcut to the URL.

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ git remote add origin git@github.com:preetpatil/myproject.git

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$
```

12) git clone <url> : Clone repo located at onto local machine. Original repo can be located on the local filesystem or on a remote machine via HTTP or SSH.

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ git clone git@github.com:preetzpatil/myproject.git
Cloning into 'myproject'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 9 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (9/9), done.
Resolving deltas: 100% (1/1), done.

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$
```

13) git push origin : push a branch to a remote repository with all its commits and objects.

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git push origin
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 302 bytes | 302.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:preetzpatil/myproject.git
   6ee4496..488309f  main -> main

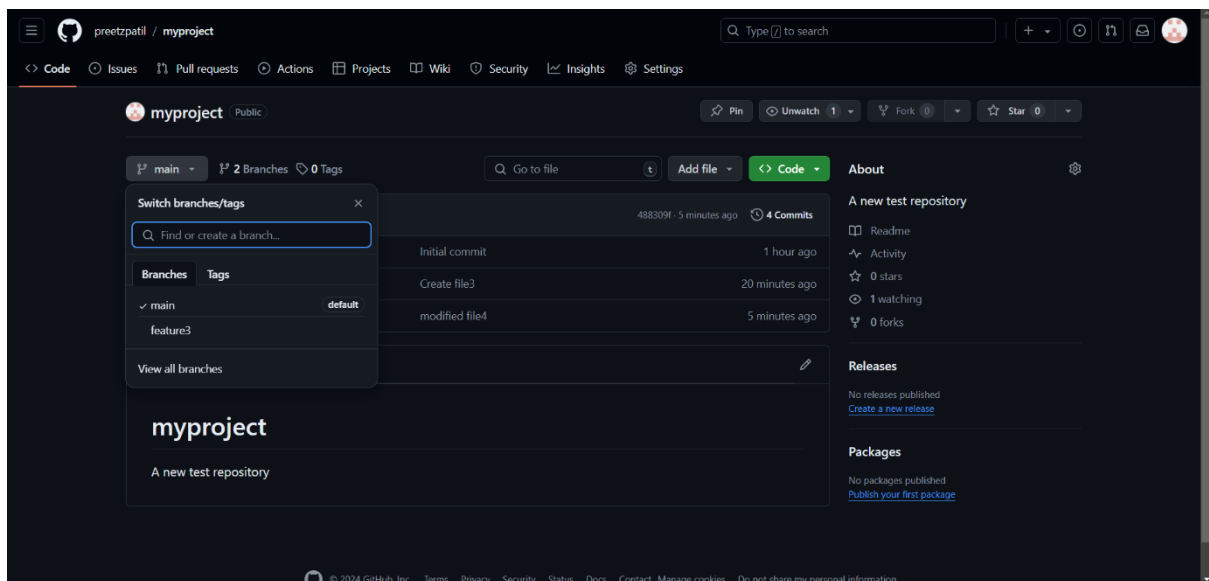
lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$
```

14) git push <remote> <branch> : Push the branch to , along with necessary commits and objects. Creates named branch in the remote repo if it doesn't exist.

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git push origin feature3
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature3' on GitHub by visiting:
remote:   https://github.com/preetzpatil/myproject/pull/new/feature3
remote:
remote: To github.com:preetzpatil/myproject.git
* [new branch]      feature3 -> feature3

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ |
```



15) git pull <remote> : Fetch the specified remote's copy of current branch and immediately merge it into the local copy.

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ git pull origin
remote: Enumerating objects: 14, done.
remote: Counting objects: 100% (14/14), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 14 (delta 4), reused 4 (delta 2), pack-reused 0
Unpacking objects: 100% (14/14), 3.78 KiB | 71.00 KiB/s, done.
From github.com:preetzpatil/myproject
 * [new branch]      feature3 -> origin/feature3
 * [new branch]      main -> origin/main
You asked to pull from the remote 'origin', but did not specify
a branch. Because this is not the default configured remote
for your current branch, you must specify a branch on the command line.

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$
```

16) git fetch <remote> <branch> : Fetches a specific , from the repo. Leave off to fetch all remote refs.

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git fetch origin
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 1.05 KiB | 63.00 KiB/s, done.
From github.com:preetzpatil/myproject
 c528c19..3fe60a4  main -> origin/main

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$
```

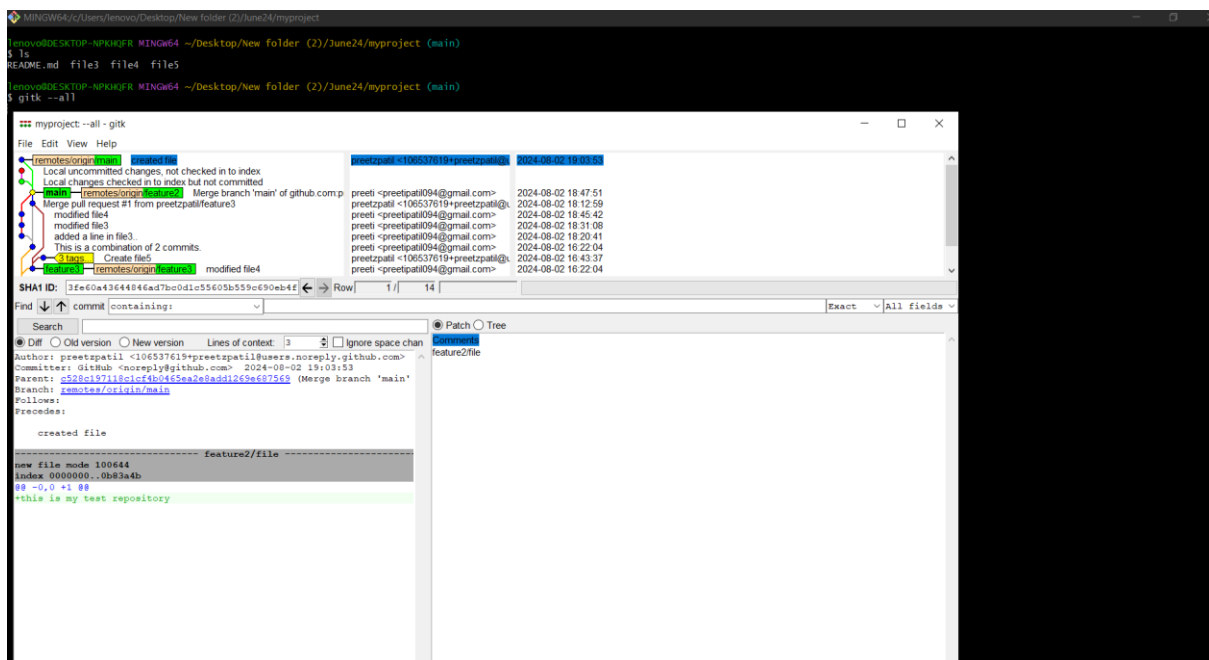
```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git remote -v
origin  git@github.com:preetzpatil/myproject.git (fetch)
origin  git@github.com:preetzpatil/myproject.git (push)

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git fetch origin
From github.com:preetzpatil/myproject
* [new branch]      feature2    -> origin/feature2

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$
```

17) gitk --all : Git will launch a graphical window showing the commit history of your repository, including all branches, tags, and remote tracking branches. This can be a helpful visual tool for understanding the structure and history of your repository.



18) git rebase [base] : Rebase the current branch onto , can be a commit ID, a branch name, a tag, or a relative reference to HEAD.

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git log --oneline
3e5cab8 (HEAD -> main, tag: v2.0, tag: v1.0.1, tag: v1.0, origin/main, origin/HEAD) Create file5
488309f (origin/feature3, feature3) modified file4
6ee4496 hello world program
6d64a6a Create file3
9a7ea81 Initial commit

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git rebase -i HEAD~2
```

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24/myproject
pick 488309f modified file4
squash! 3e5cab8 Create file5

# Rebase 6ee4496..3e5cab8 onto 6ee4496 (2 commands)
#
# 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
# u, update-ref <ref> = track a placeholder for the <ref> to be updated
#                       to this position in the new commits. The <ref> is
#                       updated at the end of the rebase
#
# 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.
#
~
```

```

MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24/myproject
This is a combination of 2 commits.
# This is the 1st commit message:

modified file4

# This is the commit message #2:

Create file5

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Fri Aug 2 16:22:04 2024 +0530
#
# interactive rebase in progress; onto 6ee4496
# Last commands done (2 commands done):
#   pick 488309f modified file4
#   squash 3e5cab8 Create file5
# No commands remaining.
# You are currently rebasing branch 'main' on '6ee4496'.
#
# Changes to be committed:
#   modified:   file4
#   new file:   file5
#
~
~
~
~
~

```

```

MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git log --oneline
3e5cab8 (HEAD -> main, tag: v2.0, tag: v1.0.1, tag: v1.0, origin/main, origin/HEAD) Create file5
488309f (origin/feature3, feature3) modified file4
6ee4496 hello world program
6d64a6a Create file3
9a7ea81 Initial commit

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git rebase -i HEAD~2
[detached HEAD 67fe227] This is a combination of 2 commits.
  Date: Fri Aug 2 16:22:04 2024 +0530
  2 files changed, 3 insertions(+)
  create mode 100644 file5
Successfully rebased and updated refs/heads/main.

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$

```

```

MINGW64:/c:/Users/lenovo/Desktop/New folder (2)/June24/myproject
lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git log --one-line
3e5cab8 (HEAD -> main, tag: v2.0, tag: v1.0.1, tag: v1.0, origin/main, origin/HEAD) Create file5
488309f (origin/feature3, feature3) modified file4
6ee4496 hello world program
6d64a6a Create file3
9a7ea81 Initial commit

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git rebase -i HEAD~2
[detached HEAD 67fe227] This is a combination of 2 commits.
Date: Fri Aug 2 16:22:04 2024 +0530
2 files changed, 3 insertions(+)
create mode 100644 file5
Successfully rebased and updated refs/heads/main.

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git log
commit 67fe22724e6eb409d5b1e5a042e5015b413999f4 (HEAD -> main)
Author: preeti <preetipatil094@gmail.com>
Date: Fri Aug 2 16:22:04 2024 +0530

    This is a combination of 2 commits.

    modified file4

    Create file5

commit 6ee44963eaa4989f79fb458a37d8288ab1674652
Author: preetipatil <106537619@preetipatil@users.noreply.github.com>
Date: Fri Aug 2 16:18:10 2024 +0530

    hello world program

commit 6d64a6a8044ca542fe95d67c413aa80a94dd1695
Author: preetipatil <106537619@preetipatil@users.noreply.github.com>
Date: Fri Aug 2 14:36:07 2024 +0530

    Create file3

commit 9a7ea817c3a88072e626cc86932f3acb37e72fb5
Author: preetipatil <106537619@preetipatil@users.noreply.github.com>
Date: Fri Aug 2 14:36:07 2024 +0530

    Initial commit

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$

```

19) git stash -m “message” : Put current changes in your working directory into stash for later use.

```

MINGW64:/c:/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ ls
README.md  file3  file4  file5

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ vi file3

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git stash -m "added a line in file3"
Saved working directory and index state On main: added a line in file3

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$

```

git stash list : to check the all stash entries.

```
MINGW64:/c:/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ ls
README.md file3 file4 file5

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ vi file3

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git stash -m "added a line in file3"
Saved working directory and index state On main: added a line in file3

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git stash list
stash@{0}: On main: added a line in file3

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$
```

git stash apply : to apply stash to current working directory.

```
MINGW64:/c:/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git stash list
stash@{0}: On main: added a line in file3

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git add .

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git commit -m "added a line in file3"
[main 0fb48a7] added a line in file3
1 file changed, 1 insertion(+), 1 deletion(-)

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git stash apply
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ |
```

git stash pop : Apply stored stash content into working directory, and clear stash.

```
MINGW64/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git stash list

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ ls
README.md file3 file4 file5

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ vi file3

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git stash -m "modified file3"
Saved working directory and index state On main: modified file3

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git stash list
stash@{0}: On main: modified file3

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git stash pop
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

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:   file3

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (c01724e1004bd920847d1edd17f3da7914c2fc58)

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$
```

```
MINGW64/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git stash list

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ ls
README.md file3 file4 file5

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ vi file3

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git stash -m "modified file3"
Saved working directory and index state On main: modified file3

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git stash list
stash@{0}: On main: modified file3

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git stash pop
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

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:   file3

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (c01724e1004bd920847d1edd17f3da7914c2fc58)

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git status list
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$
```

git stash drop : Delete a specific stash from all your previous stashes.

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git stash list
stash@{0}: On main: added a line in file3

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git stash drop stash@{0}
Dropped stash@{0} (2168bdbcc61f683c6a7f31edbe61b1a4658a2430)

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git stash list

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$
```

20) git reset [file] : unstage the specified file without overwriting changes.

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ ls
README.md file3 file4 file5

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git status
On branch main
Your branch is behind 'origin/main' by 5 commits, and can be fast-forwarded.
(use "git pull" to update your local 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:   file5

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

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git reset file5
Unstaged changes after reset:
M       file5

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git status
On branch main
Your branch is behind 'origin/main' by 5 commits, and can be fast-forwarded.
(use "git pull" to update your local 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:   file5

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

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$
```


git reset --hard [target_commit_hash] : Switches the current branch to the target reference, leaving a difference as an uncommitted change. When --hard is used, all changes are discarded. It's easy to lose uncommitted changes with --hard.

```
MINGW64/c/Users/lenovo/Desktop/New folder (2)/June24/myproject
lenovo@DESKTOP-NPKKHJF MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git log --oneline
97bfc60 (HEAD -> main) Revert "added new line to file4"
c528c19 (origin/feature2) Merge branch 'main' of github.com:preetpatil/myproject
24e38a5 modified file4
83b0f65 modified file3
1c639f4 added a line in file3..
29e6ffc Merge pull request #1 from preetpatil/feature3
67fe227 This is a combination of 2 commits.
488309f (origin/feature3, feature3) modified file4
6ee4496 hello world program
6d64a6a Create file3
9a7ea81 Initial commit

lenovo@DESKTOP-NPKKHJF MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git reset --hard 24e38a5
HEAD is now at 24e38a5 modified file4

lenovo@DESKTOP-NPKKHJF MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git log
commit 24e38a521a0bac0992c83aa12a1169ed9308c128 (HEAD -> main)
Author: preeti <preetipatil094@gmail.com>
Date: Fri Aug 2 18:45:42 2024 +0530

    modified file4

commit 83b0f65f70e175acb67cfa73d774aa5af020a13f
Author: preeti <preetipatil094@gmail.com>
Date: Fri Aug 2 18:31:08 2024 +0530

    modified file3

commit 1c639f4b1eaa77d3b42b946aaf163fb337eb2f5
Author: preeti <preetipatil094@gmail.com>
Date: Fri Aug 2 18:20:41 2024 +0530

    added a line in file3..

commit 67fe22724e6eb409d5b1e5a042e5015b413999f4
Author: preeti <preetipatil094@gmail.com>
Date: Fri Aug 2 16:22:04 2024 +0530

    This is a combination of 2 commits.

    modified file4

    Create file5

commit 6ee44963eaa4989f79fb458a37d8288ab1674652
Author: preetpatil <106537619-preetzpatil@users.noreply.github.com>
Date: Fri Aug 2 16:18:10 2024 +0530

    hello world program

commit 6d64a6a8044ca542fe95d67c413aa80a94dd1695
Author: preetpatil <106537619-preetzpatil@users.noreply.github.com>
Date: Fri Aug 2 16:07:39 2024 +0530
```

git reset --soft [target_commit_hash] : performs a soft reset to a specified commit. Here's a breakdown of what it does:

Effects:

1. **Moves the branch pointer:** Updates the current branch to point to the specified [target_commit_hash].
2. **Preserves changes:** Keeps all changes made since the target commit in:
 - **Staging area (index):** Changes are still staged and ready to be committed.
 - **Working directory:** Changes are still present in your local files.
3. **No commit history change:** The commit history remains intact; no commits are deleted or rewritten.

```
MINGW64/C:/Users/lenovo/Desktop/New folder (2)/June24/myproject
$ git log --oneline
24e38a5 (HEAD -> main) modified file4
83b0f65 modified file3
1c639f4 added a line in file3..
57e227 This is a combination of 2 commits.
8ee4496 hello world program
6d64a6a Create file3
9a7ea81 Initial commit

lenovo@DESKTOP-NPKKHJF MINGW64 ~/Desktop/new folder (2)/June24/myproject (main)
$ git reset --soft 83b0f65

lenovo@DESKTOP-NPKKHJF MINGW64 ~/Desktop/new folder (2)/June24/myproject (main)
$ git status
On branch main
Your branch is behind 'origin/main' by 5 commits, and can be fast-forwarded.
  (use "git pull" to update your local branch)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   file4

lenovo@DESKTOP-NPKKHJF MINGW64 ~/Desktop/new folder (2)/June24/myproject (main)
$ git restore --staged file4

lenovo@DESKTOP-NPKKHJF MINGW64 ~/Desktop/new folder (2)/June24/myproject (main)
$ git status
On branch main
Your branch is behind 'origin/main' by 5 commits, and can be fast-forwarded.
  (use "git pull" to update your local 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:   file4

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

lenovo@DESKTOP-NPKKHJF MINGW64 ~/Desktop/new folder (2)/June24/myproject (main)
$ git add file4

lenovo@DESKTOP-NPKKHJF MINGW64 ~/Desktop/new folder (2)/June24/myproject (main)
$ git commit -m "modified file4"
[main 7a7df62] modified file4
1 file changed, 1 insertion(+), 1 deletion(-)

lenovo@DESKTOP-NPKKHJF MINGW64 ~/Desktop/new folder (2)/June24/myproject (main)
$ git status
On branch main
Your branch and 'origin/main' have diverged,
and have 1 and 5 different commits each, respectively.
  (use "git pull" if you want to integrate the remote branch with yours)

nothing to commit, working tree clean

lenovo@DESKTOP-NPKKHJF MINGW64 ~/Desktop/new folder (2)/June24/myproject (main)
$
```

git reset --mixed [target_commit_hash] : performs a mixed reset to a specified commit.

Here's a breakdown of what it does:

Effects:

1. **Moves the branch pointer:** Updates the current branch to point to the specified [target_commit_hash].
2. **Resets staging area (index):** Removes all changes since the target commit from the staging area (index).
3. **Preserves changes in working directory:** Keeps all changes made since the target commit in your local files (working directory).

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git log --oneline
7a7df62 (HEAD -> main) modified file4
83b0f65 modified file3
1c639f4 added a line in file3..
67fe227 This is a combination of 2 commits.
6ee4496 hello world program
6d64a6a Create file3
9a7ea81 Initial commit

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git reset --mixed 83b0f65
Unstaged changes after reset:
M   file4

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git status
On branch main
Your branch is behind 'origin/main' by 5 commits, and can be fast-forwarded.
  (use "git pull" to update your local 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:   file4

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

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$
```

21) git squash : Squashing multiple commits into one simplifies a project's Git tree and a makes long development history more organized.

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git merge --squash feature3
Automatic merge went well; stopped before committing as requested
Squash commit -- not updating HEAD

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ |
```

22) git revert <commit_hash> : Create new commit that undoes all of the changes made in , then apply it to the current branch.

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git revert 24e38a521a0bac0992c83aa12a1169ed9308c128
[main 97bfc60] Revert "added new line to file4"
1 file changed, 1 insertion(+), 1 deletion(-)

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$
```



```
MINGW64/c/Users/lenovo/Desktop/New folder (2)/June24/myproject
lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git revert 24e38a521a0bac0992c83aa12a1169ed9308c128
main 97bfc60] Revert "added new line to file4"
1 file changed, 1 insertion(+), 1 deletion(-)

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git log
commit 97bfc601bea3b51b079a685e4c0a7aa98434e9ad (HEAD -> main)
author: preeti <preetipatil1094@gmail.com>
date: Fri Aug 2 21:52:26 2024 +0530

    Revert "added new line to file4"

    This reverts commit 24e38a521a0bac0992c83aa12a1169ed9308c128.

commit c528c197118c1cf4b0465ea2e8add1269e687569 (origin/feature2)
merge: 24e38a5 29e6ffc
author: preeti <preetipatil1094@gmail.com>
date: Fri Aug 2 18:47:51 2024 +0530

    Merge branch 'main' of github.com:preetipatil/myproject

commit 24e38a521a0bac0992c83aa12a1169ed9308c128
author: preeti <preetipatil1094@gmail.com>
date: Fri Aug 2 18:45:42 2024 +0530

    modified file4

commit 83b0f65f70e175acb67cfa73d774aa5af020a13f
author: preeti <preetipatil1094@gmail.com>
date: Fri Aug 2 18:51:08 2024 +0530

    modified file3

commit 1c639f44b1eaa77d3b42b946aaf163fb337eb2f5
author: preeti <preetipatil1094@gmail.com>
date: Fri Aug 2 18:20:41 2024 +0530

    added a line in file3..

commit 29e6ffc9035112fe8969cbe1e1727b2932a2c03a
merge: 67fe227 488309f
author: preetipatil <106537619@preetipatil0users.noreply.github.com>
date: Fri Aug 2 18:12:59 2024 +0530

    Merge pull request #1 from preetipatil/feature3

    modified file4

commit 67fe22724e6eb409d5b1e3a042e5015b413999f4
author: preeti <preetipatil1094@gmail.com>
date: Fri Aug 2 16:22:04 2024 +0530

    This is a combination of 2 commits.

    modified file4
```

23) git tag [name] : Create a tag reference named name for current commit. Add commit sha to tag a specific commit instead of current one.

```
MINGW64/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git tag v1.0

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git tag v1.0.1

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git tag
v1.0
v1.0.1

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ |
```

git tag -a "tagname" -m "message" : Create a tag object named name with the tag message.

```
MINGW64/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git tag v2.0 -m "tag for releasing version v2.0"

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git tag
v1.0
v1.0.1
v2.0

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$
```

git show tagname : displays the detail info. of tag.

```
MINGW64/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git tag v2.0 -m "tag for releasing version v2.0"

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git tag
v1.0
v1.0.1
v2.0

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git show v2.0
tag v2.0
Tagger: preeti <preetipatil094@gmail.com>
Date: Fri Aug 2 17:55:23 2024 +0530

tag for releasing version v2.0

commit 3e5cab80877b8c21556ef25238ad961b725b5a7e (HEAD -> main, tag: v2.0, tag: v1.0.1, tag: v1.0, origin/main, origin/HEAD)
Author: preetpatil <106537619+preetpatil@users.noreply.github.com>
Date: Fri Aug 2 16:43:37 2024 +0530

    Create file5

diff --git a/file5 b/file5
new file mode 100644
index 0000000..8b13789
--- /dev/null
+++ b/file5
@@ -0,0 +1 @@
+
lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$
```

git push --tags : to push tags to remote repo.

```
MINGW64:/c:/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 169 bytes | 84.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:preetzpatil/myproject.git
 * [new tag]          v1.0 -> v1.0
 * [new tag]          v1.0.1 -> v1.0.1
 * [new tag]          v2.0 -> v2.0

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$
```

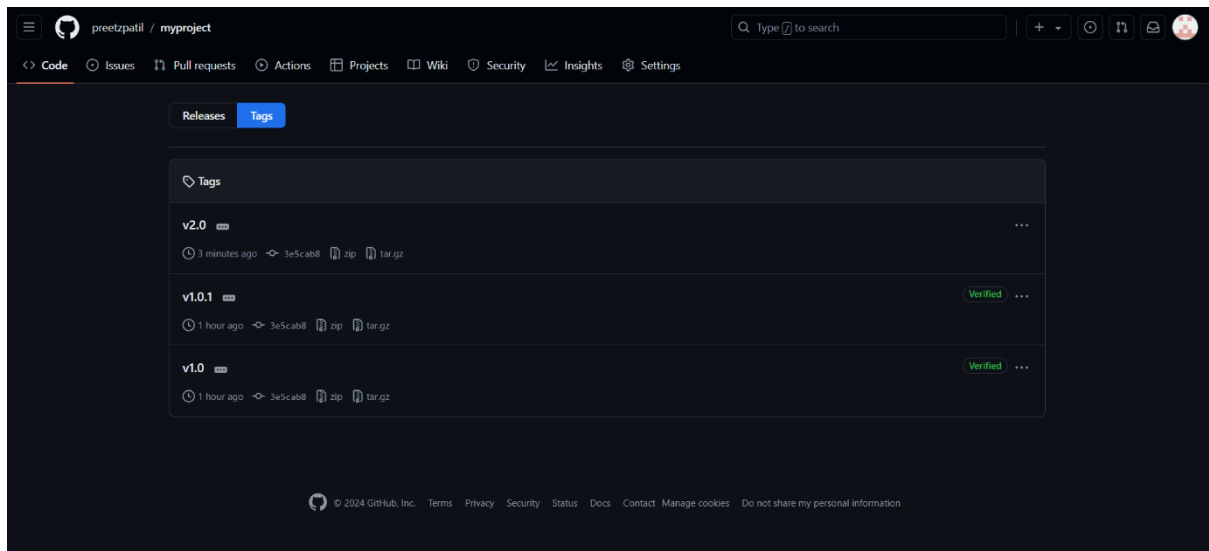
git ls-remote --tags origin : to list the remote tags.

```
MINGW64:/c:/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 169 bytes | 84.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:preetzpatil/myproject.git
 * [new tag]          v1.0 -> v1.0
 * [new tag]          v1.0.1 -> v1.0.1
 * [new tag]          v2.0 -> v2.0

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git ls-remote --tags origin
3e5cab80877b8c21556ef25238ad961b725b5a7e      refs/tags/v1.0
3e5cab80877b8c21556ef25238ad961b725b5a7e      refs/tags/v1.0.1
33e6dcaa8c76260c1b9af098bfbf0c63ec7c48aa      refs/tags/v2.0
3e5cab80877b8c21556ef25238ad961b725b5a7e      refs/tags/v2.0^{}

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ |
```

git tag -d "tagname" : Remove a tag from local repository.

```
MINGW64~/Desktop/New folder (2)/June24/myproject
lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git tag
v1.0
v1.0.1
v2.0

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git tag -d v2.0
Deleted tag 'v2.0' (was 33e6dca)

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git tag
v1.0
v1.0.1

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ |
```

25) git commit --amend : Replace the last commit with the staged changes and last commit combined. Use with nothing staged to edit the last commit's message.

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24/myproject
added a line in file3..|

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Fri Aug 2 18:20:41 2024 +0530
#
# On branch main
# Your branch is ahead of 'origin/main' by 1 commit.
#   (use "git push" to publish your local commits)
#
# Changes to be committed:
#   modified:   file3
#
# Changes not staged for commit:
#   modified:   file3
#
~
~
~
~
~
~
~
~
~
```

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git log --oneline
0fb48a7 (HEAD -> main) added a line in file3
67fe227 (origin/main, origin/HEAD) This is a combination of 2 commits.
6ee4496 hello world program
6d64a6a Create file3
9a7ea81 Initial commit

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git commit --amend
[main 1c639f4] added a line in file3..
Date: Fri Aug 2 18:20:41 2024 +0530
1 file changed, 1 insertion(+), 1 deletion(-)

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$
```

26) git cherry-pick [commit_hash] : it is used to apply a specific commit or set of commits from one branch to another branch.

```
MINGW64/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git checkout feature3
Switched to branch 'feature3'

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (feature3)
$ git log
commit 488309f4cf63f27297507e7ca1f94f320176cbeb (HEAD -> feature3, origin/feature3)
Author: preeti <preetipatil094@gmail.com>
Date: Fri Aug 2 16:22:04 2024 +0530

    modified file4

commit 6ee44963eaa4989f79fb458a37d8288ab1674652
Author: preetzpatil <106537619+preetzpatil@users.noreply.github.com>
Date: Fri Aug 2 16:18:10 2024 +0530

    hello world program

commit 6d64a6a8044ca542fe95d67c413aa80a94dd1695
Author: preetzpatil <106537619+preetzpatil@users.noreply.github.com>
Date: Fri Aug 2 16:07:39 2024 +0530

    Create file3

commit 9a7ea817c3a88072e626cc86932f3acb37e72fb5
Author: preetzpatil <106537619+preetzpatil@users.noreply.github.com>
Date: Fri Aug 2 14:56:07 2024 +0530

    Initial commit

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (feature3)
$ |
```

```
MINGW64/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (feature3)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git cherry-pick 9a7ea817c3a88072e626cc86932f3acb37e72fb5
The previous cherry-pick is now empty, possibly due to conflict resolution.
If you wish to commit it anyway, use:

  git commit --allow-empty

otherwise, please use 'git cherry-pick --skip'

In branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

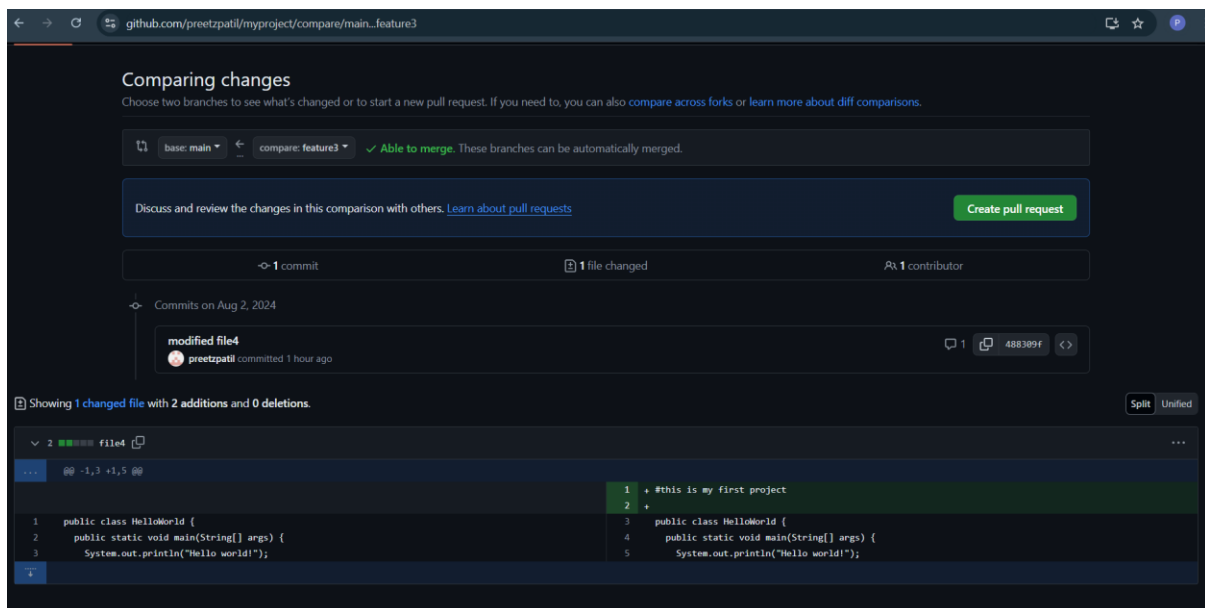
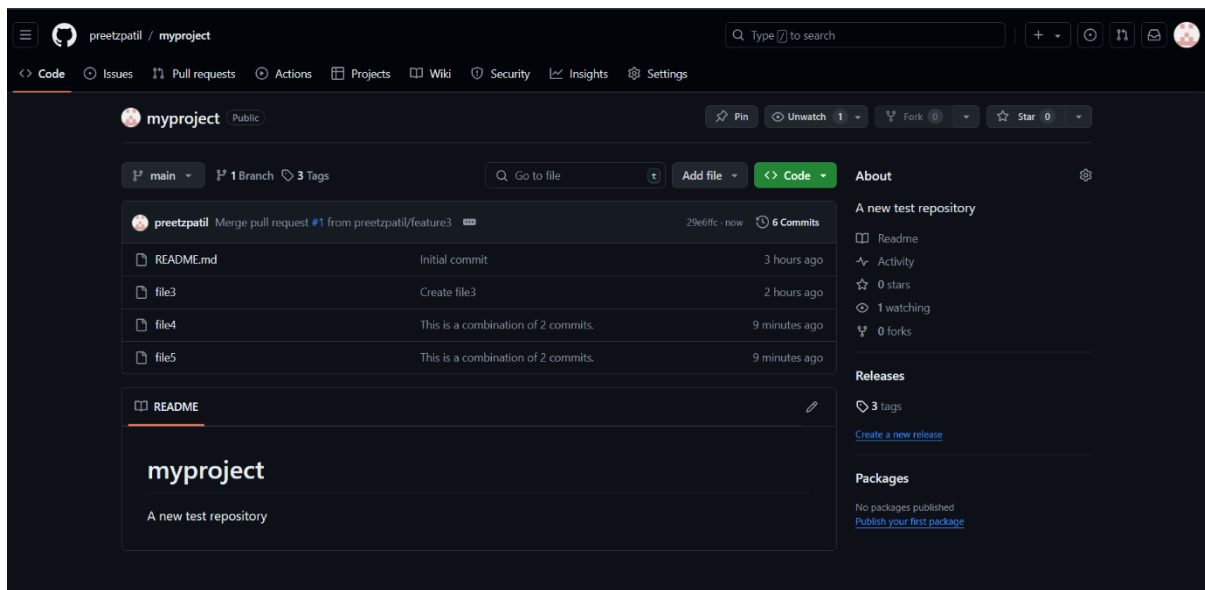
You are currently cherry-picking commit 9a7ea81.
  (all conflicts fixed: run "git cherry-pick --continue")
  (use "git cherry-pick --skip" to skip this patch)
  (use "git cherry-pick --abort" to cancel the cherry-pick operation)

nothing to commit, working tree clean

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main|CHERRY-PICKING)
$ git log --oneline
b3b0f65 (HEAD -> main) modified file3
fc639f4 added a line in file3.
b7fe227 (origin/main, origin/HEAD) This is a combination of 2 commits.
6ee4496 hello world program
6d64a6a Create file3
9a7ea81 Initial commit

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main|CHERRY-PICKING)
$
```

27) pull request : it is a way to propose changes to a repository's codebase. It's a request to the repository maintainers to review and merge your changes into their codebase.



github.com/preetzpatil/myproject/compare/main...feature3

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](#). [Learn more about diff comparisons here](#).

base: maincompare: feature3

✓ Able to merge. These branches can be automatically merged.

Add a title

modified file4

Write

Preview

Add your description here...

Markdown is supported

Paste, drop, or click to add files

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](#)

Create pull request

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

github.com/preetzpatil/myproject/pull/1

modified file4 #1

Openpreetzpatil wants to merge 1 commit into main from feature3

Conversation0

Commits1

Checks0

Files changed1

+2-0

preetzpatil commented now

Owner

No description provided.

modified file4

488309f

Require approval from specific reviewers before merging

[Rulesets](#) ensure specific people approve pull requests before they're merged.

Add rule

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 request

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

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

Add a comment

28) .gitignore : it is a file in a Git repository that tells Git which files or directories to ignore and not track.

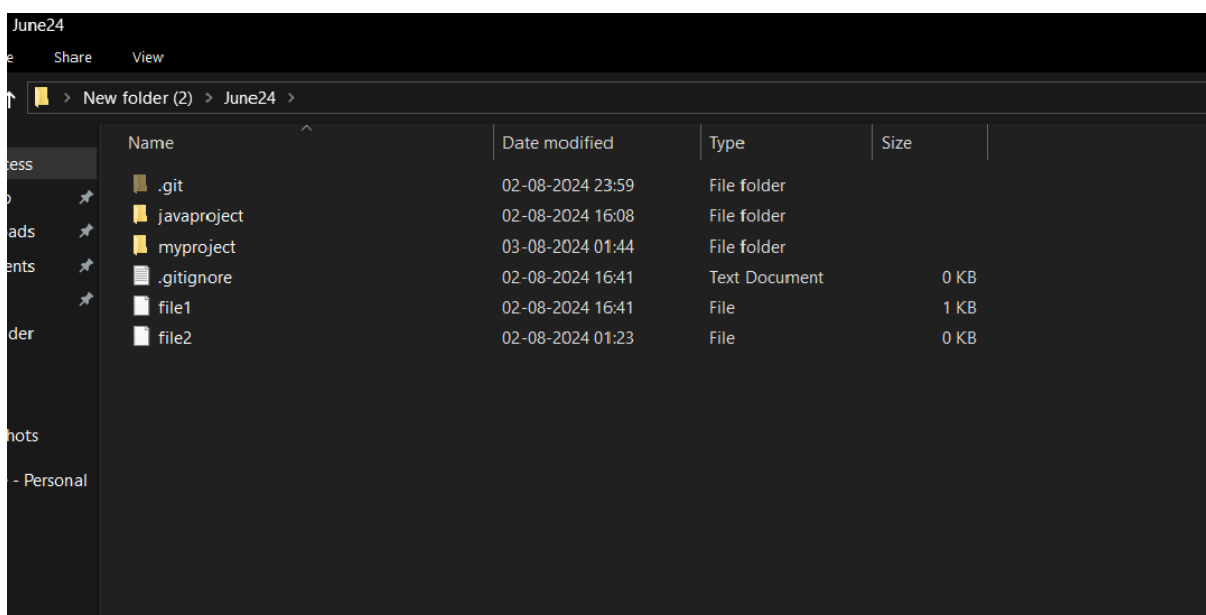
- Ignore files that are not essential to the project, such as:
 - Temporary files
 - Build artifacts
 - Log files
 - Personal settings
- Keep sensitive information, like passwords or API keys, out of the repository.
- Reduce repository size and improve performance.

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ touch .gitignore

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ ls
file1  file2  javaproject/  myproject/

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24 (master)
$ |
```



The screenshot shows a Windows File Explorer window with the address bar set to 'New folder (2) > June24'. The left sidebar shows the 'This PC' view with 'June24' selected. The main pane displays a table of files and folders.

Name	Date modified	Type	Size
.git	02-08-2024 23:59	File folder	
javaproject	02-08-2024 16:08	File folder	
myproject	03-08-2024 01:44	File folder	
.gitignore	02-08-2024 16:41	Text Document	0 KB
file1	02-08-2024 16:41	File	1 KB
file2	02-08-2024 01:23	File	0 KB

29) git diff : see the difference between the working directory and the index (which changes have not been committed yet).

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ ls
README.md  file3  file4  file5

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ vi file4

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git diff
diff --git a/file4 b/file4
index 310f706..cbe2857 100644
--- a/file4
+++ b/file4
@@ -1,4 +1,4 @@
-#this is my first project
+#this is my first project..

public class HelloWorld {
    public static void main(String[] args) {

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ |
```

git diff HEAD : see the difference between the last commit and the working directory.

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git log --oneline
83b0f65 (HEAD -> main) modified file3
1c639f4 added a line in file3..
67fe227 (origin/main, origin/HEAD) This is a combination of 2 commits.
6ee4496 hello world program
6d64a6a Create file3
9a7ea81 Initial commit

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git diff HEAD
diff --git a/file4 b/file4
index 310f706..cbe2857 100644
--- a/file4
+++ b/file4
@@ -1,4 +1,4 @@
-#this is my first project
+#this is my first project..

public class HelloWorld {
    public static void main(String[] args) {

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ |
```


git diff --cached : see the difference between the last commit and the index.

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git log --oneline
83b0f65 (HEAD -> main) modified file3
1c639f4 added a line in file3..
67fe227 (origin/main, origin/HEAD) This is a combination of 2 commits.
6ee4496 hello world program
6d64a6a Create file3
9a7ea81 Initial commit

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git diff --cached

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git diff 83b0f65 1c639f4
diff --git a/file3 b/file3
index 8c779d6..ce01362 100644
--- a/file3
+++ b/file3
@@ -1,1 @@
-hello world..
+hello

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$
```

git diff <branch1> <branch2> : shows changes between two branches.

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git diff main feature3
diff --git a/file3 b/file3
index 8c779d6..8b13789 100644
--- a/file3
+++ b/file3
@@ -1,1 @@
-hello world..
+
diff --git a/file5 b/file5
deleted file mode 100644
index 8b13789..0000000
--- a/file5
+++ /dev/null
@@ -1,0,0 @@
-

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$
```

git diff --stat : shows summary of changes (added,deleted,modified files).

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git diff --stat
file4 | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$
```

git diff -p : shows patch output (detailed changes).

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git diff -p
diff --git a/file4 b/file4
index 310f706..cbe2857 100644
--- a/file4
+++ b/file4
@@ -1,4 +1,4 @@
-#this is my first project
+#this is my first project..

public class HelloWorld {
    public static void main(String[] args) {

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$
```

30) git bisect : it is used to find the commit that introduced a bug or an issue in our codebase.it uses binary search algorithm to identify the problematic commit.

```
MINGW64/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New Folder (2)/June24/myproject (main)
$ git log --oneline
764336c (HEAD -> main) modified file5
83b0f65 modified file3
1c639f4 added a line in file3..
67fe227 This is a combination of 2 commits.
6ee4496 hello world program
6d64a6a Create file3
9a7ea81 Initial commit

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New Folder (2)/June24/myproject (main)
$ git bisect start
status: waiting for both good and bad commits

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New Folder (2)/June24/myproject (main|BISECTING)
$ git bisect bad HEAD
status: waiting for good commit(s), bad commit known

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New Folder (2)/June24/myproject (main|BISECTING)
$ git bisect good 6d64a6a
bisecting: 2 revisions left to test after this (roughly 1 step)
[67fe22724e6eb409d5b1e5a042e5015b413999f4] This is a combination of 2 commits.

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New Folder (2)/June24/myproject ((67fe227...)|BISECTING)
$ git bisect good
bisecting: 0 revisions left to test after this (roughly 1 step)
[83b0f65f70e175acb67cfa73d774aa5af020a13f] modified file3

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New Folder (2)/June24/myproject ((83b0f65...)|BISECTING)
$ git bisect good
764336c11cd575f4f5d7b3b8074349e14972277f is the first bad commit
commit 764336c11cd575f4f5d7b3b8074349e14972277f (main)
Author: preeti <preetipatil1094@gmail.com>
Date: Sat Aug 3 03:27:35 2024 +0530

    modified file5

file5 | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New Folder (2)/June24/myproject ((83b0f65...)|BISECTING)
$ git bisect reset
Previous HEAD position was 83b0f65 modified file3
Switched to branch 'main'
Your branch and 'origin/main' have diverged,
and have 1 and 5 different commits each, respectively.
(use 'git pull' if you want to integrate the remote branch with yours)

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New Folder (2)/June24/myproject (main)
$
```

31) git relog : Show a log of changes to the local repository's HEAD. Add --relative-date flag to show date info or --all to show all refs.

```
MINGW64/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New Folder (2)/June24/myproject (main)
$ git relog
764336c (HEAD -> main) HEAD@{0}: checkout: moving from 83b0f65f70e175acb67cfa73d774aa5af020a13f to main
83b0f65 HEAD@{1}: checkout: moving from 67fe22724e6eb409d5b1e5a042e5015b413999f4 to 83b0f65f70e175acb67cfa73d774aa5af020a13f
67fe227 HEAD@{2}: checkout: moving from main to 67fe22724e6eb409d5b1e5a042e5015b413999f4
764336c (HEAD -> main) HEAD@{3}: checkout: moving from 83b0f65f70e175acb67cfa73d774aa5af020a13f to main
83b0f65 HEAD@{4}: checkout: moving from 67fe22724e6eb409d5b1e5a042e5015b413999f4 to 83b0f65f70e175acb67cfa73d774aa5af020a13f
67fe227 HEAD@{5}: checkout: moving from main to 67fe22724e6eb409d5b1e5a042e5015b413999f4
764336c (HEAD -> main) HEAD@{6}: commit: modified file5
83b0f65 HEAD@{7}: checkout: moving from main to main
83b0f65 HEAD@{8}: checkout: moving from main to main
83b0f65 HEAD@{9}: reset: moving to 83b0f65
7a7df62 HEAD@{10}: commit: modified file4
83b0f65 HEAD@{11}: reset: moving to 83b0f65
24e38a5 HEAD@{12}: reset: moving to 24e38a5
97bfc60 HEAD@{13}: reset: moving to 97bfc601bea3b51b079a685e4c0a7aa98434e9ad
97bfc60 HEAD@{14}: revert: Revert "added new line to file4"
c528c19 (origin/feature2) HEAD@{15}: rebase (abort): returning to refs/heads/main
24e38a5 HEAD@{16}: rebase (start): checkout HEAD-6
c528c19 (origin/feature2) HEAD@{17}: checkout: moving from feature2 to main
1d068e1 (feature2) HEAD@{18}: rebase (finish): returning to refs/heads/feature2
1d068e1 (feature2) HEAD@{19}: rebase (squash): This is a combination of 6 commits.
e83a6ee HEAD@{20}: rebase (continue): This is a combination of 6 commits.
9956a91 HEAD@{21}: rebase (squash): # This is a combination of 4 commits.
9c46898 HEAD@{22}: rebase (squash): # This is a combination of 3 commits.
a5c4e05 HEAD@{23}: rebase (squash): # This is a combination of 2 commits.
67fe227 HEAD@{24}: rebase (start): checkout HEAD-6
9012f16 HEAD@{25}: commit: file
c528c19 (origin/feature2) HEAD@{26}: checkout: moving from main to feature2
c528c19 (origin/feature2) HEAD@{27}: checkout: moving from main to main
24e38a5 HEAD@{28}: pull origin: Merge made by the 'ort' strategy.
24e38a5 HEAD@{29}: commit: modified file4
83b0f65 HEAD@{30}: reset: moving to 83b0f65f70e175acb67cfa73d774aa5af020a13f
83b0f65 HEAD@{31}: checkout: moving from feature3 to main
188309f (origin/feature3, feature3) HEAD@{32}: checkout: moving from main to feature3
83b0f65 HEAD@{33}: checkout: moving from feature3 to main
188309f (origin/feature3, feature3) HEAD@{34}: checkout: moving from main to feature3
83b0f65 HEAD@{35}: commit: modified file3
1c639f4 HEAD@{36}: commit (amend): added a line in file3..
9fb48a7 HEAD@{37}: reset: moving to HEAD
9fb48a7 HEAD@{38}: reset: moving to HEAD
9fb48a7 HEAD@{39}: commit: added a line in file3
67fe227 HEAD@{40}: reset: moving to HEAD
67fe227 HEAD@{41}: checkout: moving from feature3 to main
188309f (origin/feature3, feature3) HEAD@{42}: checkout: moving from main to feature3
67fe227 HEAD@{43}: rebase (finish): returning to refs/heads/main
67fe227 HEAD@{44}: rebase (squash): This is a combination of 2 commits.
188309f (origin/feature3, feature3) HEAD@{45}: rebase (start): checkout HEAD-2
9e3cab8 (tag: v1.0.1, tag: v1.0) HEAD@{46}: pull origin: Fast-forward
188309f (origin/feature3, feature3) HEAD@{47}: commit: modified file4
6ee4496 HEAD@{48}: clone: from github.com:preetipatil/myproject.git

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New Folder (2)/June24/myproject (main)
$
```

32) git blame <file> : shows the last modification for each line in a file, along with the commit information. It's useful for tracking who made changes to a specific line of code and when.

```
MINGW64/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ ls
README.md file3 file4 file5

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git blame file3
83b0f65f (preeti 2024-08-02 18:31:08 +0530 1) hello world..

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git blame -e file3
83b0f65f (<preetipatil094@gmail.com> 2024-08-02 18:31:08 +0530 1) hello world..

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git blame -e -C file3
83b0f65f (<preetipatil094@gmail.com> 2024-08-02 18:31:08 +0530 1) hello world..

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$
```

33) git clean -f : Deletes untracked files and directories.

```
MINGW64/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ ls
README.md file3 file4 file5

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ touch file6

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ ls
README.md file3 file4 file5 file6

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git status
On branch main
Your branch and 'origin/main' have diverged,
and have 1 and 5 different commits each, respectively.
(use "git pull" if you want to integrate the remote branch with yours)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  file6

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

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git clean -f
Removing file6

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git status
On branch main
Your branch and 'origin/main' have diverged,
and have 1 and 5 different commits each, respectively.
(use "git pull" if you want to integrate the remote branch with yours)

nothing to commit, working tree clean

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ ls
README.md file3 file4 file5

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$
```

33) git log --graph --oneline --all : This command will display a concise graph of your commit history, including all branches.

Here's what each option does:

--graph: Displays a text-based graph of the commit history.

--oneline: Condenses each commit to a single line, showing the commit hash and message.

--all: Includes all branches and commits in the graph, not just the current branch.

This command will give you a compact and informative visualization of your repository's commit history.

```
MINGW64/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git log --graph --oneline --all
* 764336c (HEAD -> main) modified file5
* 1d068e1 (feature2) This is a combination of 6 commits.
  | * 3fe60a4 (origin/main, origin/HEAD) created file
  | * c528c19 (origin/feature2) Merge branch 'main' of github.com:preetzpatil/myproject
  |
  | * 29e6ffc Merge pull request #1 from preetzpatil/feature3
  |
  | * 24e38a5 modified file4
  |
  | * 83b0f65 modified file3
  | * 1c639f4 added a line in file3..
  |
  | * 67fe227 This is a combination of 2 commits.
  |
  | * 3e5cab8 (tag: v1.0.1, tag: v1.0) Create file5
  |
  | * 488309f (origin/feature3, feature3) modified file4
  |
  | * 6ee4496 hello world program
  | * 6d64a6a Create file3
  | * 9a7ea81 Initial commit
$
```

34) git fsck : Check the integrity of the repository.

```
MINGW64/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git fsck
Checking object directories: 100% (256/256), done.
Checking objects: 100% (9/9), done.
dangling commit 1d780d25a42555e0ddc89d1d63775320fee2458b
dangling commit 2168bdbcc61f683c6a7f31edbe61b1a4658a2430
dangling tag 33e6dcaa8c76260c1b9af098bfbf0c63ec7c48aa
dangling commit c01724e1004bd920847d1edd17f3da7914c2fc58
dangling tree e0fa54b048b4ed3e5a2cfdd65faa2d2af25363bb

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ |
```

35) git gc : Garbage collect and optimize the repository.

```
MINGW64:/c/Users/lenovo/Desktop/New folder (2)/June24/myproject

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$ git gc
Enumerating objects: 45, done.
Counting objects: 100% (45/45), done.
Delta compression using up to 4 threads
Compressing objects: 100% (36/36), done.
Writing objects: 100% (45/45), done.
Total 45 (delta 22), reused 7 (delta 1), pack-reused 0 (from 0)
Enumerating cruft objects: 11, done.
Traversing cruft objects: 18, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 4 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (11/11), done.
Total 11 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)

lenovo@DESKTOP-NPKHQFR MINGW64 ~/Desktop/New folder (2)/June24/myproject (main)
$
```