

//Git repo initialization

```
vlab@VEDA2F107 MINGW64 ~/Desktop  
$ cd MyGit/
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit  
$ git init  
Initialized empty Git repository in C:/Users/vlab/Desktop/MyGit/.git/
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)  
$ ls  
hello.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)  
$ git status  
On branch master
```

No commits yet

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

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

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)  
$ git add hello.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)  
$ git status  
On branch master
```

No commits yet

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

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)  
$ touch script.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)  
$ touch web.html
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)  
$ git status  
On branch master
```

No commits yet

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

Untracked files:

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

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git add .
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git status
On branch master
```

No commits yet

Changes to be committed:
(use "git rm --cached <file>..." to unstage)
new file: hello.txt
new file: script.txt
new file: web.html

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git status
On branch master
```

No commits yet

Changes to be committed:
(use "git rm --cached <file>..." to unstage)
new file: hello.txt
new file: script.txt
new file: web.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: web.html

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git add .
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git status
On branch master
```

No commits yet

Changes to be committed:
(use "git rm --cached <file>..." to unstage)
new file: hello.txt
new file: script.txt

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git rm script.txt
error: the following file has changes staged in the index:
```

```
script.txt
(use --cached to keep the file, or -f to force removal)
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git rm -f script.txt
rm 'script.txt'
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git status
On branch master

No commits yet
```

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

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ touch script.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git add ;
git: 'add.' is not a git command. See 'git --help'.
```

```
The most similar command is
    add
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git add .
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git status
On branch master
```

```
No commits yet
```

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

//to keep committed file back to the staging area

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git rm --cached script.txt
rm 'script.txt'
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git status
On branch master
```

```
No commits yet
```

```
Changes to be committed:
```

(use "git rm --cached <file>..." to unstage)
new file: hello.txt

Untracked files:

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

//Git username gmail initialization

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git config --global user.email "lochu5vilehya@gmail.com"
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git config --global user.name "Lochani"
```

//to set cr-lf

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (1.0)
$ git config --global core.autocrlf true
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git commit --help
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git commit -m "initial commit"
[master (root-commit) c33d4dc] initial commit
3 files changed, 1 insertion(+)
create mode 100644 hello.txt
create mode 100644 script.txt
create mode 100644 web.html
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git log
commit c33d4dc1e853391825b5afc6b4b9f25278100555 (HEAD -> master)
Author: Lochani <lochu5vilehya@gmail.com>
Date: Mon Mar 4 17:17:33 2024 +0530

    initial commit
```

//detached head

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git log
commit c45367190f3b5d37a4c3747f57833a5ae4514985 (HEAD -> master)
Author: Lochani <lochu5vilehya@gmail.com>
Date: Mon Mar 4 17:33:55 2024 +0530

    modified text in script
```

```
commit bb8ca8e68f8ad2b2f1c6d25893438f304831fac3
Author: Lochani <lochu5vilehya@gmail.com>
Date: Mon Mar 4 17:23:33 2024 +0530

    untracked files
```

```
commit 093da3c369fce507e351acb6dfe52aab7afcd7ba
Author: Lochani <lochu5vilehya@gmail.com>
Date: Mon Mar 4 17:21:17 2024 +0530

    new file added
```

```
commit c33d4dc1e853391825b5afc6b4b9f25278100555
Author: Lochani <lochu5vilehya@gmail.com>
Date: Mon Mar 4 17:17:33 2024 +0530

    initial commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git checkout bb8ca8e68f8ad2b2f1c6d25893438f304831fac3
Note: switching to 'bb8ca8e68f8ad2b2f1c6d25893438f304831fac3'.
```

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using `-c` with the switch command. Example:

```
git switch -c <new-branch-name>
```

Or undo this operation with:

```
git switch -
```

Turn off this advice by setting config variable `advice.detachedHead` to `false`

HEAD is now at bb8ca8e untracked files

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit ((bb8ca8e...))
$ git checkout master
Previous HEAD position was bb8ca8e untracked files
Switched to branch 'master'
```

//GIT revert

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit ((c453671...))
$ git checkout master
Previous HEAD position was c453671 modified text in script
Switched to branch 'master'
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git revert 92b0f51f57fcb615182c7b38d08bd1e84f2b4491
[master fc86003] Reapply "modified text in script"
1 file changed, 1 insertion(+)
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git log
```

```
commit fc860031729715825f3a3a08f7de9ce7062d2137 (HEAD -> master)
Author: Lochani <lochu5vilehya@gmail.com>
Date: Tue Mar 5 08:55:55 2024 +0530
```

Reapply "modified text in script"

This reverts commit 92b0f51f57fcb615182c7b38d08bd1e84f2b4491.

```
commit 92b0f51f57fcb615182c7b38d08bd1e84f2b4491
Author: Lochani <lochu5vilehya@gmail.com>
Date: Tue Mar 5 08:50:11 2024 +0530
```

Revert "modified text in script"

This reverts commit c45367190f3b5d37a4c3747f57833a5ae4514985.

```
commit c45367190f3b5d37a4c3747f57833a5ae4514985
Author: Lochani <lochu5vilehya@gmail.com>
Date: Mon Mar 4 17:33:55 2024 +0530
```

modified text in script

```
commit bb8ca8e68f8ad2b2f1c6d25893438f304831fac3
Author: Lochani <lochu5vilehya@gmail.com>
Date: Mon Mar 4 17:23:33 2024 +0530
```

untracked files

```
commit 093da3c369fce507e351acb6dfe52aab7afcd7ba
Author: Lochani <lochu5vilehya@gmail.com>
Date: Mon Mar 4 17:21:17 2024 +0530
```

new file added

```
commit c33d4dc1e853391825b5afc6b4b9f25278100555
Author: Lochani <lochu5vilehya@gmail.com>
Date: Mon Mar 4 17:17:33 2024 +0530
```

initial commit

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git log --oneline
2164071 (HEAD -> master) added two more lines in bye.txt
871a6f9 text added in bye.txt
fc86003 Reapply "modified text in script"
```

```
92b0f51 Revert "modified text in script"
c453671 modified text in script
bb8ca8e untracked files
093da3c new file added
c33d4dc initial commit
```

//Git reset

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git reset --hard fc86003
HEAD is now at fc86003 Reapply "modified text in script"

vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git log --oneline
fc86003 (HEAD -> master) Reapply "modified text in script"
92b0f51 Revert "modified text in script"
c453671 modified text in script
bb8ca8e untracked files
093da3c new file added
c33d4dc initial commit
```

//Gitignore

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ cat .gitignore
#auto generated file
script.txt

#pictures
Capture.png

#auto gen dirs.
f1/*

vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ mkdir f1

vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ cd f1

vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit/f1 (master)
$ cat >> f1.txt
hello
```



```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit/f1 (master)
$ cd ..
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (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:   .gitignore
        modified:   AutoGen/a.txt
```

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

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

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git add .
warning: in the working copy of 'a.txt', LF will be replaced by CRLF
the next time Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git commit -m "new folder f1"
[master 03a8e0a] new folder f1
 3 files changed, 4 insertions(+), 1 deletion(-)
 create mode 100644 a.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git status
On branch master
nothing to commit, working tree clean
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ cd f1
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit/f1 (master)
$ cat >> f1.txt
gfixgy
ljhx
wkjh
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit/f1 (master)
$ cd ..
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/MyGit (master)
$ git status
On branch master
nothing to commit, working tree clean
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git checkout -b dev
Switched to a new branch 'dev'
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (dev)
$ git checkout master
Switched to branch 'master'
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git branch bugs
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git branch -a
  bugs
  dev
* master
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git checkout bugs
Switched to branch 'bugs'
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (bugs)
$ git branch -a
* bugs
  dev
  master
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (bugs)
$ git checkout master
Switched to branch 'master'
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git branch -d bugs
Deleted branch bugs (was eb8a4a5).
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git branch -a
  dev
* master
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git checkout dev
Switched to branch 'dev'
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (dev)
$ git status
On branch dev
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: lib/file2.txt

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

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (dev)
$ git add .
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (dev)
$ git commit -m "lib/file2 edited"
[dev 1a14447] lib/file2 edited
1 file changed, 1 insertion(+)
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (dev)
$ git add .
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (dev)
$ git status
On branch dev
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   src/file.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (dev)
$ git commit -m "src/file edited"
[dev 23c2812] src/file edited
1 file changed, 1 insertion(+)
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (dev)
$ git log
commit 23c2812d56566f9401e211927f6a6a3245552ff0 (HEAD -> dev)
Author: Lochani <lochu5vilehya@gmail.com>
Date: Tue Mar 5 10:50:34 2024 +0530
```

src/file edited

```
commit 1a144478891293a254b06ca6c9bd304c3c1e6e8a
Author: Lochani <lochu5vilehya@gmail.com>
Date: Tue Mar 5 10:49:34 2024 +0530
```

lib/file2 edited

```
commit eb8a4a59f2789932b58b0969683605c9d0c8f3c1 (master)
Author: Lochani <lochu5vilehya@gmail.com>
Date: Tue Mar 5 10:43:44 2024 +0530
```

initial commit

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (bugs)
$ cd login

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching/login (bugs)
$ vi hello.txt

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching/login (bugs)
$ ls
hello.txt

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching/login (bugs)
$ git add .
warning: in the working copy of 'login/hello.txt', LF will be replaced
by CRLF the next time Git touches it

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching/login (bugs)
$ git commit -m "hello.txt in login"
[bugs b41f54b] hello.txt in login
1 file changed, 1 insertion(+)
create mode 100644 login/hello.txt

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching/login (bugs)
$ cd ..

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (bugs)
$ ls
lib/  log.txt  login/  src/

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (bugs)
$ git checkout master
Switched to branch 'master'

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ ls
lib/  src/

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git checkout bugs
Switched to branch 'bugs'
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (bugs)
$ git branch dummy
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (bugs)
$ git branch
* bugs
  dev
  dummy
  master
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (bugs)
$ cat .git/refs/heads/master
1ea96362bee48a4cb0186a96016cbc04123e5258
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (bugs)
$ git log
commit b41f54bbb7e50f46235ac6256755e8e69e8f7c9e (HEAD -> bugs, dummy)
Author: Lochani <lochu5vilehya@gmail.com>
Date: Tue Mar 5 11:11:27 2024 +0530
```

hello.txt in login

```
commit 0f1b3a224b5ebca0b133d54631d00d0041fad379
Author: Lochani <lochu5vilehya@gmail.com>
Date: Tue Mar 5 11:04:59 2024 +0530
```

login folder added for logging

```
commit bf567b881138289c5ab34262ea73b1962765bf6e (dev)
Author: Lochani <lochu5vilehya@gmail.com>
Date: Tue Mar 5 10:54:58 2024 +0530
```

release version 1.0

```
commit 23c2812d56566f9401e211927f6a6a3245552ff0
Author: Lochani <lochu5vilehya@gmail.com>
Date: Tue Mar 5 10:50:34 2024 +0530
```

src/file edited

```
commit 1a144478891293a254b06ca6c9bd304c3c1e6e8a
Author: Lochani <lochu5vilehya@gmail.com>
Date: Tue Mar 5 10:49:34 2024 +0530
```

lib/file2 edited

```
commit eb8a4a59f2789932b58b0969683605c9d0c8f3c1
Author: Lochani <lochu5vilehya@gmail.com>
Date: Tue Mar 5 10:43:44 2024 +0530
```

initial commit

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (bugs)
$ cat .git/refs/heads/bugs
b41f54bbb7e50f46235ac6256755e8e69e8f7c9e
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (bugs)
$ cat .git/refs/heads/dummy
b41f54bbb7e50f46235ac6256755e8e69e8f7c9e
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (bugs)
$ cat .git/HEAD
ref: refs/heads/bugs
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (bugs)
$ git checkout dummy
Switched to branch 'dummy'
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (dummy)
$ cat .git/HEAD
ref: refs/heads/dummy
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (dummy)
$ git branch
  bugs
  dev
* dummy
  master
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (dummy)
$ git branch -m new_branch           //to create branch
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (new_branch)
$ git branch
  bugs
  dev
  master
* new_branch
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git branch
  bugs
  dev
* master
  new_branch
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git checkout new_branch
Switched to branch 'new_branch'
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (new_branch)
$ ls
lib/  log.txt  login/  src/
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (new_branch)
$ cat >> log.txt
hello mam
how are you
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (new_branch)
```

```
$ git status
On branch new_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:   log.txt

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

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (new_branch)
$ git add .
warning: in the working copy of 'log.txt', LF will be replaced by CRLF
the next time Git touches it
g
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (new_branch)
$ git commit -m "log.txt edited"
[new_branch 4481071] log.txt edited
1 file changed, 2 insertions(+)

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (new_branch)
$ git checkout master
Switched to branch 'master'
```

//Git delete branch

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (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_branch'
hint: Disable this message with "git config advice.forceDeleteBranch
false"

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git branch -D new_branch
Deleted branch new_branch (was 4481071).
```

//Git merging

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git branch
  bugs
  dev
* master

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git checkout -b new_feature //to create branch
Switched to a new branch 'new_feature'

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (new_feature)
$ ls
index.txt  lib/  src/

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (new_feature)
$ vi new.txt

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (new_feature)
$ git add .
warning: in the working copy of 'new.txt', LF will be replaced by CRLF
the next time Git touches it

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (new_feature)
$ git commit -m "new file added"
[new_feature 6ba3361] new file added
1 file changed, 1 insertion(+)
create mode 100644 new.txt

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (new_feature)
$ git checkout master
Switched to branch 'master'

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git merge new_feature //to merge
Updating 1ea9636..6ba3361
Fast-forward
 new.txt | 1 +
1 file changed, 1 insertion(+)
create mode 100644 new.txt

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git branch dummy_branch

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ ls
index.txt  lib/  new.txt  src/

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ vi index.txt

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git add .

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
```



```

$ git commit -m "changes done to index file"
[master 4931eba] changes done to index file
1 file changed, 1 insertion(+)

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git checkout dummy_branch
Switched to branch 'dummy_branch'

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (dummy_branch)
$ vi dummy.txt

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (dummy_branch)
$ git add .
warning: in the working copy of 'dummy.txt', LF will be replaced by
CRLF the next time Git touches it

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (dummy_branch)
$ git commit -m "dummy file"
[dummy_branch fdc643b] dummy file
1 file changed, 1 insertion(+)
create mode 100644 dummy.txt

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (dummy_branch)
$ git checkout master //to switch branch
Switched to branch 'master'

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git merge dummy_branch
Merge made by the 'ort' strategy.
dummy.txt | 1 +
1 file changed, 1 insertion(+)
create mode 100644 dummy.txt

```

//Git merge conflict!!!

```

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/merging (master)
$ cat >> hello.txt
helloo
hiii

```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/merging (master)
$ git add .
warning: in the working copy of 'hello.txt', LF will be replaced by
CRLF the next time Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/merging (master)
$ git commit -m "added text hello"
[master 35faf3d] added text hello
1 file changed, 2 insertions(+)
create mode 100644 hello.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/merging (master)
$ git checkout -b hello_branch
Switched to a new branch 'hello_branch'
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/merging (hello_branch)
$ ls
hello.txt  index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/merging (hello_branch)
$ vi hello.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/merging (hello_branch)
$ git add .
warning: in the working copy of 'hello.txt', LF will be replaced by
CRLF the next time Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/merging (hello_branch)
$ git commit -m "hello branch first commit"
[hello_branch 5451a9b] hello branch first commit
1 file changed, 1 insertion(+)
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/merging (hello_branch)
$ git checkout master
Switched to branch 'master'
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/merging (master)
$ vi hello.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/merging (master)
$ git add .
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/merging (master)
$ git commit -m "added new text"
[master 8ab7b39] added new text
1 file changed, 3 insertions(+)
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/merging (master)
$ git merge hello_branch
Auto-merging hello.txt
CONFLICT (content): Merge conflict in hello.txt
Automatic merge failed; fix conflicts and then commit the result.
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/merging (master|MERGING)
$ git merge --abort
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/merging (master)
$ git merge hello_branch
```

Auto-merging hello.txt
CONFLICT (content): Merge conflict in hello.txt
Automatic merge failed; fix conflicts and then commit the result.

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/merging (master|MERGING)
$ vi hello.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/merging (master|MERGING)
$ git add .
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/merging (master|MERGING)
$ git commit -m "merged successfully"
[master 0d501c6] merged successfully
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/merging (master)
$ git log
commit 0d501c6c452780fb89db2516461f1e1d477d3023 (HEAD -> master)
Merge: 8ab7b39 5451a9b
Author: Lochani <lochu5vilehya@gmail.com>
Date: Tue Mar 5 14:24:02 2024 +0530
```

merged successfully

```
commit 8ab7b3934d6ed0d3132858ddb4a832a2f46e2160
Author: Lochani <lochu5vilehya@gmail.com>
Date: Tue Mar 5 14:20:53 2024 +0530
```

added new text

```
commit 5451a9ba928bbac20c12a553a4380c6f4db1251f (hello_branch)
Author: Lochani <lochu5vilehya@gmail.com>
Date: Tue Mar 5 14:19:58 2024 +0530
```

hello branch first commit

```
commit 35faf3d82a3225b4da0b96e57e2298e1783c623d
Author: Lochani <lochu5vilehya@gmail.com>
Date: Tue Mar 5 14:18:55 2024 +0530
```

added text hello

```
commit 465fc7232e3a03addfd973353c70ae49e8c5c9c6
Merge: 0e72694 be8d6d0
Author: Lochani <lochu5vilehya@gmail.com>
Date: Tue Mar 5 14:15:16 2024 +0530
```

Merge branch 'feature'

```
commit 0e72694835aaabd5eb4e3f1c4ad6b4eab176455b
Author: Lochani <lochu5vilehya@gmail.com>
Date: Tue Mar 5 14:08:19 2024 +0530
```

second change

```
commit c396d83d88a253226f4bcf48d5c37164f50b38ac
Author: Lochani <lochu5vilehya@gmail.com>
Date: Tue Mar 5 14:06:27 2024 +0530
```

master changes

```
commit be8d6d0cb5c4f1c11e22b156428d670d59ed5ad2 (feature)
Author: Lochani <lochu5vilehya@gmail.com>
Date: Tue Mar 5 14:05:36 2024 +0530
```

changes in feature branch

```
commit bfa79d8448ec9d89e330ebd537da2427d5f0c1c1
Author: Lochani <lochu5vilehya@gmail.com>
Date: Tue Mar 5 14:04:23 2024 +0530
```

initial commit

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/merging (master)
$ git log --oneline -graph //log graph
* 0d501c6 (HEAD -> master) merged successfully
| \
| * 5451a9b (hello_branch) hello branch first commit
* | 8ab7b39 added new text
|/
* 35faf3d added text hello
* 465fc72 Merge branch 'feature'
| \
| * be8d6d0 (feature) changes in feature branch
* | 0e72694 second change
* | c396d83 master changes
|/
```

//Git rebase

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT
$ mkdir rebase

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT
$ cd rebase

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase
```

```
$ git init
Initialized empty Git repository in
C:/Users/vlab/Desktop/GIT/rebase/.git/
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase (master)
$ vi master.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase (master)
$ git add .
warning: in the working copy of 'master.txt', LF will be replaced by
CRLF the next time Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase (master)
$ git commit -m "master commit"
[master (root-commit) 84eb478] master commit
1 file changed, 1 insertion(+)
create mode 100644 master.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase (master)
$ git branch feature
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase (master)
$ git branch
feature
* master
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase (master)
$ vi master1.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase (master)
$ git add .
warning: in the working copy of 'master1.txt', LF will be replaced by
CRLF the next time Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase (master)
$ git commit -m "master1 commit"
[master b8f9ef7] master1 commit
1 file changed, 1 insertion(+)
create mode 100644 master1.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase (master)
$ vi master2.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase (master)
$ git add .
warning: in the working copy of 'master2.txt', LF will be replaced by
CRLF the next time Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase (master)
$ git commit -m "master2 commit"
[master b15768d] master2 commit
1 file changed, 1 insertion(+)
create mode 100644 master2.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase (master)
$ git log
commit b15768d28a43e435c3508944b042a6a9e130edcf (HEAD -> master)
Author: Lochani <lochu5vilehya@gmail.com>
```

Date: Tue Mar 5 15:00:01 2024 +0530

master2 commit

commit b8f9ef769e3de0493b84c164cffca2c4029d4745

Author: Lochani <lochu5vilehya@gmail.com>

Date: Tue Mar 5 14:59:33 2024 +0530

master1 commit

commit 84eb4787737115c6e96e8580fd7717b88b10ae (feature)

Author: Lochani <lochu5vilehya@gmail.com>

Date: Tue Mar 5 14:58:37 2024 +0530

master commit

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase (master)

\$ git checkout feature

Switched to branch 'feature'

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase (feature)

\$ vi feature1.txt

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase (feature)

\$ git add .

warning: in the working copy of 'feature1.txt', LF will be replaced by CRLF the next time Git touches it

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase (feature)

\$ git commit -m "feature1 added"

[feature d15d371] feature1 added

1 file changed, 1 insertion(+)

create mode 100644 feature1.txt

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase (feature)

\$ vi feature2.txt

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase (feature)

\$ git add .

warning: in the working copy of 'feature2.txt', LF will be replaced by CRLF the next time Git touches it

g

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase (feature)

\$ git commit -m "feature2 added"

[feature ddf3e75] feature2 added

1 file changed, 1 insertion(+)

create mode 100644 feature2.txt

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase (feature)

\$ git clog

git: 'clog' is not a git command. See 'git --help'.

The most similar commands are

clone

log

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase (feature)

\$ git log

```
commit ddf3e752517e1f63c37f742fa5ffb6295ff23a40 (HEAD -> feature)
Author: Lochani <lochu5vilehya@gmail.com>
Date: Tue Mar 5 15:02:07 2024 +0530
```

feature2 added

```
commit d15d371a81838add35c077edc466575684e79477
Author: Lochani <lochu5vilehya@gmail.com>
Date: Tue Mar 5 15:01:34 2024 +0530
```

feature1 added

```
commit 84eb4787737115c6e96e8580fd7717b88b10aead
Author: Lochani <lochu5vilehya@gmail.com>
Date: Tue Mar 5 14:58:37 2024 +0530
```

master commit

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase (feature)
$ git log --oneline
ddf3e75 (HEAD -> feature) feature2 added
d15d371 feature1 added
84eb478 master commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase (feature)
$ git checkout master
Switched to branch 'master'
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase (master)
$ git log --oneline
b15768d (HEAD -> master) master2 commit
b8f9ef7 master1 commit
84eb478 master commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase (master)
$ git rebase feature
Successfully rebased and updated refs/heads/master.
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase (master)
$ git log --oneline
891090b (HEAD -> master) master2 commit
2afcd7 master1 commit
ddf3e75 (feature) feature2 added
d15d371 feature1 added
84eb478 master commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/rebase (master)
$ git log --oneline --graph
* 891090b (HEAD -> master) master2 commit
* 2afcd7 master1 commit
* ddf3e75 (feature) feature2 added
* d15d371 feature1 added
* 84eb478 master commit
```

//Git interacting rebase

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase
```

```
$ git init
```

```
Initialized empty Git repository in  
C:/Users/vlab/Desktop/GIT/git_interacting_rebase/.git/
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (master)
```

```
$ git status
```

```
On branch master
```

```
No commits yet
```

```
Untracked files:
```

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

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

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (master)
```

```
$ git add .
```

```
warning: in the working copy of 'home.txt', LF will be replaced by  
CRLF the next time Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (master)
```

```
$ git commit -m "master first commit"
```

```
[master (root-commit) 0850c31] master first commit  
1 file changed, 1 insertion(+)  
create mode 100644 home.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (master)
```

```
$ git checkout -b new
```

```
Switched to a new branch 'new'
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
```

```
$ vi feature.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
```

```
$ git add .
```

```
warning: in the working copy of 'feature.txt', LF will be replaced by  
CRLF the next time Git touches it
```



```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
```

```
$ git commit -m "new first commit"
```

```
[new 02920c7] new first commit  
1 file changed, 1 insertion(+)  
create mode 100644 feature.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
```

```
$ vi feature.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
```

```
$ git add .
```

```
warning: in the working copy of 'feature.txt', LF will be replaced by  
CRLF the next time Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
```

```
$ git commit -m "new second commit"
```

```
[new f25a817] new second commit  
1 file changed, 1 insertion(+)
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
```

```
$ vi feature.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
```

```
$ git add .
```

```
warning: in the working copy of 'feature.txt', LF will be replaced by  
CRLF the next time Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
```

```
$ git commit -m "new third commit"
```

```
[new 1868f6d] new third commit  
1 file changed, 1 insertion(+)
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
```

```
$ vi feature.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
```

```
$ git add .
```

```
warning: in the working copy of 'feature.txt', LF will be replaced by  
CRLF the next time Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
```

```
$ git commit -m "new fourth commit"
```

```
[new 165eb5f] new fourth commit  
1 file changed, 1 insertion(+)
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
```

```
$ git log --oneline
```

```
165eb5f (HEAD -> new) new fourth commit  
1868f6d new third commit  
f25a817 new second commit  
02920c7 new first commit  
0850c31 (master) master first commit
```

```

pick 7f7192d started working on the new feature
s| 74bce80 WIP: still working on the gfeature
s 4679e3d WIP: fix logic issues
s b33b38e Completed new feature

# Rebase 807814d..b33b38e onto 807814d (4 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 <commit> - like "squash", but discard this commit's log message
# 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
# L, reset <label> - reset HEAD to a label
# m, merge [ C <commit> | C <commit> ] <label> [# <oneline>]
# . create a merge commit using the original merge commit's
# . message (or the oneline, if no original merge commit was
# . specified). Use C <commit> to reword the commit message.
#
# These lines can be re-ordered; they are executed from top to bottom.
#

```

```

# This is a combination of 4 commits.
# This is the 1st commit message:

# This is the commit message #2:

# This is the commit message #3:

Completed new feature

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Sun Apr 18 00:05:52 2021 +0530
#
# interactive rebase in progress; onto 807814d
# Last commands done (4 commands done):
#   squash 4679e3d WIP: fix logic issues
#   squash b33b38e Completed new feature
# No commands remaining.
# You are currently rebasing branch 'new_feature' on '807814d'.
#
# Changes to be committed:
#   new file:   feature.txt

```

```

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
$ git rebase -i master
[detached HEAD 2bbe718] implemented new feature
Date: Tue Mar 5 15:31:19 2024 +0530
1 file changed, 4 insertions(+)
create mode 100644 feature.txt
Successfully rebased and updated refs/heads/new.

```

```

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
$ git log --oneline
2bbe718 (HEAD -> new) implemented new feature
0850c31 (master) master first commit

```

//Git diff and git show

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
$ vi home.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
$ git status
On branch new
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:   home.txt
```

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

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
$ git add .
warning: in the working copy of 'home.txt', LF will be replaced by
CRLF the next time Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
$ git commit -m "wrong text in home.txt"
[new d11983a] wrong text in home.txt
1 file changed, 1 insertion(+)
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
$ git log --oneline
d11983a (HEAD -> new) wrong text in home.txt
2bbe718 implemented new feature
0850c31 (master) master first commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
$ git add .
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
$ vi home.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
$ git add .
warning: in the working copy of 'home.txt', LF will be replaced by
CRLF the next time Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
$ git status
On branch new
```

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

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
$ git show d11983a
commit d11983a95dfbff478ad4e8594e1e67fd6d7f7e6a (HEAD -> new)
Author: Lochani <lochu5vilehya@gmail.com>
Date:   Tue Mar 5 15:48:03 2024 +0530
```

wrong text in home.txt

```
diff --git a/home.txt b/home.txt
index 8a07d72..15cd727 100644
--- a/home.txt
+++ b/home.txt
@@ -1,2 @@
 hello home
+welcm
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
$ git diff --cached
diff --git a/home.txt b/home.txt
index 15cd727..5d8392a 100644
--- a/home.txt
+++ b/home.txt
@@ -1,2 +1,2 @@
 hello home
-welcm
+welcome
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
$ git status
On branch new
nothing to commit, working tree clean
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
$ git log
commit 38a4d29435c82020c83aa9b35752a1e737980414 (HEAD -> new)
Author: Lochani <lochu5vilehya@gmail.com>
Date:   Tue Mar 5 15:48:03 2024 +0530
```

wrong text in home.txt

```
commit 2bbe718e6a22833b30c985b0a736e33dd41f7426
Author: Lochani <lochu5vilehya@gmail.com>
Date:   Tue Mar 5 15:31:19 2024 +0530
```

implemented new feature

```
commit 0850c31c82c8229a5a50ccda89760b78407636af (master)
Author: Lochani <lochu5vilehya@gmail.com>
Date:   Tue Mar 5 15:30:22 2024 +0530
```

master first commit

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
$ git log --oneline
38a4d29 (HEAD -> new) wrong text in home.txt
2bbe718 implemented new feature
0850c31 (master) master first commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_interacting_rebase (new)
$ git show 38a4d29
commit 38a4d29435c82020c83aa9b35752a1e737980414 (HEAD -> new)
Author: Lochani <lochu5vilehya@gmail.com>
Date: Tue Mar 5 15:48:03 2024 +0530
```

wrong text in home.txt

```
diff --git a/home.txt b/home.txt
index 8a07d72..5d8392a 100644
--- a/home.txt
+++ b/home.txt
@@ -1,2 @@
 hello home
+welcome
```

//Git cherry-pick

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (master)
$ vi index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (master)
$ git add .
warning: in the working copy of 'index.txt', LF will be replaced by
CRLF the next time Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (master)
$ git commit -m "initial commit"
[master (root-commit) b7441c3] initial commit
1 file changed, 1 insertion(+)
create mode 100644 index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (master)
$ git branch 1.0
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (master)
$ git branch 2.0
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (master)
$ git branch 3.0
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (master)
$ git branch
```

```
1.0
2.0
3.0
* master
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (master)
```

```
$ git checkout 3.0
Switched to branch '3.0'
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (3.0)
```

```
$ vi feature.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (3.0)
```

```
$ git add .
warning: in the working copy of 'feature.txt', LF will be replaced by
CRLF the next time Git touches it
```

```
g
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (3.0)
```

```
$ git commit -m "feature 3.0"
```

```
[3.0 0d653df] feature 3.0
1 file changed, 1 insertion(+)
create mode 100644 feature.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (3.0)
```

```
$ git log --oneline
```

```
0d653df (HEAD -> 3.0) feature 3.0
b7441c3 (master, 2.0, 1.0) initial commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (3.0)
```

```
$ vi bug.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (3.0)
```

```
$ git add .
```

```
warning: in the working copy of 'bug.txt', LF will be replaced by CRLF
the next time Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (3.0)
```

```
$ git commit -m "fixed bug 3.0"
```

```
[3.0 b45906f] fixed bug 3.0
1 file changed, 1 insertion(+)
create mode 100644 bug.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (3.0)
```

```
$ git log --oneline
```

```
b45906f (HEAD -> 3.0) fixed bug 3.0
0d653df feature 3.0
b7441c3 (master, 2.0, 1.0) initial commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (3.0)
```

```
$ git checkout 2.0
```

```
Switched to branch '2.0'
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (2.0)
```

```
$ git log --oneline
```

```
b7441c3 (HEAD -> 2.0, master, 1.0) initial commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (2.0)
```

```
$ ls
```

index.txt

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (2.0)

```
$ git cherry pick b45906f
fatal: unknown commit pick
```

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (2.0)

```
$ git cherry pick b45906f
fatal: unknown commit pick
```

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (2.0)

```
$ git cherry-pick b45906f
[2.0 aed562f] fixed bug 3.0
Date: Tue Mar 5 16:25:01 2024 +0530
1 file changed, 1 insertion(+)
create mode 100644 bug.txt
```

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (2.0)

```
$ git log
commit aed562f12c27ec9a846d6caab9a83dc430e95b12 (HEAD -> 2.0)
Author: Lochani <lochu5vilehya@gmail.com>
Date: Tue Mar 5 16:25:01 2024 +0530
```

fixed bug 3.0

commit b7441c3e88be640e7c3d9ae6b40d8f90d306cf4a (master, 1.0)

```
Author: Lochani <lochu5vilehya@gmail.com>
Date: Tue Mar 5 16:22:54 2024 +0530
```

initial commit

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (2.0)

```
$ ls
bug.txt index.txt
```

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (2.0)

```
$ git checkout 1.0
Switched to branch '1.0'
```

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (1.0)

```
$ ls
index.txt
```

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (1.0)

```
$ git cherry-pick b45906f
[1.0 ff55624] fixed bug 3.0
Date: Tue Mar 5 16:25:01 2024 +0530
1 file changed, 1 insertion(+)
create mode 100644 bug.txt
```

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (1.0)

```
$ ls
bug.txt index.txt
```

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (1.0)

```
$ git log --oneline
ff55624 (HEAD -> 1.0) fixed bug 3.0
b7441c3 (master) initial commit
```


//Git diff and git diff head

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (1.0)
$ git diff head
diff --git a/bug.txt b/bug.txt
index e7aa317..6bb4631 100644
--- a/bug.txt
+++ b/bug.txt
@@ -1,3 @@
    fixed the bug    //committed
+hgghghghhj        //staged
+helloooo           //working dir
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (1.0)
$ git diff
diff --git a/bug.txt b/bug.txt
index 3f39ebc..6bb4631 100644
--- a/bug.txt
+++ b/bug.txt
@@ -1,2 +1,3 @@
    fixed the bug
    hgghghghhj
+helloooo
```

//Git Reset

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (1.0)
$ vi index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (1.0)
$ git add .
warning: in the working copy of 'index.txt', LF will be replaced by
CRLF the next time Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (1.0)
$ git config --global core.autocrlf true
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (1.0)
$ git add .
```



```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (1.0)
$ git status
On branch 1.0
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (1.0)
$ git log --oneline
6897f47 (HEAD -> 1.0) new bug commit
ff55624 fixed bug 3.0
b7441c3 (master) initial commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (1.0)
$ git reset head --mixed
Unstaged changes after reset:
M      index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (1.0)
$ git status
On branch 1.0
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@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (1.0)
$ vi index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (1.0)
$ git status
On branch 1.0
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@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (1.0)
$ git add .
warning: in the working copy of 'index.txt', LF will be replaced by
CRLF the next time Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (1.0)
$ git status
On branch 1.0
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (1.0)
```

```

$ vi index.txt

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (1.0)
$ git status
On branch 1.0
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.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:   index.txt

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (1.0)
$ git reset head --hard
HEAD is now at 6897f47 new bug commit

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/cherrypick (1.0)
$ git status
On branch 1.0
nothing to commit, working tree clean

```

//Git stash

```

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
$ vi index.txt

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
$ git add .

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
$ git commit -m "initial commit"
[master (root-commit) 2db6840] initial commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 index.txt

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
$ vi index.txt

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
$ git checkout -b feature
Switched to a new branch 'feature'

```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
```

```
$ git checkout master
Switched to branch 'master'
M      index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (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@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
```

```
$ git checkout feature
Switched to branch 'feature'
M      index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
```

```
$ git checkout master
Switched to branch 'master'
M      index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
```

```
$ vi index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
```

```
$ git checkout feature
Switched to branch 'feature'
M      index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
```

```
$ git status
On branch feature
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@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
```

```
$ git add .
warning: in the working copy of 'index.txt', LF will be replaced by
CRLF the next time Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
```

```
$ git commit -m "text in index"
[feature dc1b67c] text in index
1 file changed, 2 insertions(+)
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
```

```
$ git checkout master
Switched to branch 'master'
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
$ vi index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
$ git checkout feature
error: Your local changes to the following files would be overwritten
by checkout:
    index.txt
Please commit your changes or stash them before you switch branches.
Aborting
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
$ git stash
warning: in the working copy of 'index.txt', LF will be replaced by
CRLF the next time Git touches it
Saved working directory and index state WIP on master: 2db6840 initial
commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
$ git stash list
stash@{0}: WIP on master: 2db6840 initial commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
$ git stash pop
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")
Dropped refs/stash@{0} (c2abc5e9ebff79d99e55109bd987bd509e93a552)
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
$ git stash list
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
$ git stash save 'new one'
> ^C
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
$ git stash save "new one"
Saved working directory and index state On master: new one
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
$ git stash list
stash@{0}: On master: new one
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
$ git stash apply
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@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
```

```
$ git stash list
stash@{0}: On master: new one
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
```

```
$ git checkout master
Already on 'master'
M       index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
```

```
$ git checkout feature
error: Your local changes to the following files would be overwritten
by checkout:
      index.txt
Please commit your changes or stash them before you switch branches.
Aborting
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
```

```
$ git stash list
stash@{0}: On master: new one
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (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@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
```

```
$ git reset head --hard
HEAD is now at 2db6840 initial commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
```

```
$ git checkout feature
Switched to branch 'feature'
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
```

```
$ git status apply
On branch feature
nothing to commit, working tree clean
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
```

```
$ git stash apply
Auto-merging index.txt
CONFLICT (content): Merge conflict in index.txt
On branch feature
Unmerged paths:
  (use "git restore --staged <file>..." to unstage)
  (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@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
$ vi index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
$ git status
On branch feature
Unmerged paths:
  (use "git restore --staged <file>..." to unstage)
  (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@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
$ git add .
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
$ git commit -m "final commit"
[feature 562455b] final commit
1 file changed, 1 insertion(+), 2 deletions(-)
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
$ git status
On branch feature
nothing to commit, working tree clean
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
$ git stash list
stash@{0}: On master: new one
```

//Git stash commands

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
$ git stash list
stash@{0}: On master: new one
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
$ git checkout master
Switched to branch 'master'
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
$ vi index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
$ git add .
warning: in the working copy of 'index.txt', LF will be replaced by
CRLF the next time Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
$ git commit -m "heelo"
```

> ^C

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
```

```
$ git commit -m "heelo"
```

```
[master c14dd34] heelo
```

```
1 file changed, 1 insertion(+)
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (master)
```

```
$ git checkout feature
```

```
Switched to branch 'feature'
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
```

```
$ vi index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
```

```
$ git status
```

```
On branch feature
```

```
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@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
```

```
$ git stash save "feature file"
```

```
Saved working directory and index state On feature: feature file
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
```

```
$ git stash list
```

```
stash@{0}: On feature: feature file
```

```
stash@{1}: On master: new one
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
```

```
$ git stash apply
```

```
On branch feature
```

```
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@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
```

```
$ git reset head --hard
```

```
HEAD is now at 562455b final commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
```

```
$ git status
```

```
On branch feature
```

```
nothing to commit, working tree clean
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
```

```
$ git stash list
```

```
stash@{0}: On feature: feature file
```

```
stash@{1}: On master: new one
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
```

```
$ git stash show
index.txt | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
```

```
$ git stash show -p
diff --git a/index.txt b/index.txt
index 8b13789..098a11d 100644
--- a/index.txt
+++ b/index.txt
@@ -1,1 @@
```

```
+featureeee
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
```

```
$ git stash show stash@{1} -p
diff --git a/index.txt b/index.txt
index e69de29..ce01362 100644
--- a/index.txt
+++ b/index.txt
@@ -0,0 +1 @@
+hello
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
```

```
$ git stash drop stash@{1}
Dropped stash@{1} (7a0f171d217995e7040b7dc7d62804b369e03b42)
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
```

```
$ git stash list
stash@{0}: On feature: feature file
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
```

```
$ git stash clear
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
```

```
$ git stash list
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
```

```
$ vi index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
```

```
$ git stash save "new feature stash"
Saved working directory and index state On feature: new feature stash
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
```

```
$ git stash list
stash@{0}: On feature: new feature stash
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (feature)
```

```
$ git stash branch new_feature stash@{0}
Switched to a new branch 'new_feature'
On branch new_feature
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")
Dropped stash@{0} (61d4cc393637a1a34008a0363a1d0501afebc248)

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/stash (new_feature)
$ git status
On branch new_feature
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")
```

//Git checkout command other usage

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ ls
dummy.txt  hello.txt  index.txt  lib/  new.txt  src/

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git log --oneline
e2796cc (HEAD -> master) Merge branch 'feature'
a17b5c6 changes done in hello.txt (master)
b2b7c09 (feature) edited hello.txt
bb6fab0 hello.txt added
e48c9a3 Merge branch 'dummy_branch'
fdc643b (dummy_branch) dummy file
4931eba changes done to index file
6ba3361 (new_feature) new file added
1ea9636 index.txt is committed
bf567b8 (dev) release version 1.0
23c2812 src/file edited
1a14447 lib/file2 edited
```

eb8a4a5 initial commit

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
```

```
$ git checkout Head~4
```

Note: switching to 'Head~4'.

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may

do so (now or later) by using -c with the switch command. Example:

```
git switch -c <new-branch-name>
```

Or undo this operation with:

```
git switch -
```

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 4931eba changes done to index file

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching ((4931eba...))
```

```
$ git checkout master
```

Previous HEAD position was 4931eba changes done to index file

Switched to branch 'master'

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
```

```
$ vi dummy.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
```

```
$ vi hello.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (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: dummy.txt

modified: hello.txt

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

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
```

```
$ git checkout head hello.txt
```

Updated 1 path from fc2bee8

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (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: dummy.txt

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

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
```

```
$ git checkout -- dummy.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
```

```
$ git status
```

On branch master

nothing to commit, working tree clean

//Git switch and restore commands

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
```

```
$ git branch
```

bugs

dev

dummy_branch

feature

* master

new_feature

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
```

```
$ git switch feature
```

Switched to branch 'feature'

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (feature)
```

```
$ git switch -c hello
```

Switched to a new branch 'hello'

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (hello)
```

```
$ git switch master
```

Switched to branch 'master'

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
```

```
$ git log --oneline
```

e2796cc (HEAD -> master) Merge branch 'feature'

a17b5c6 changes done in hello.txt (master)

b2b7c09 (hello, feature) edited hello.txt

bb6fab0 hello.txt added

e48c9a3 Merge branch 'dummy_branch'

fdc643b (dummy_branch) dummy file

4931eba changes done to index file

```
6ba3361 (new_feature) new file added
1ea9636 index.txt is committed
bf567b8 (dev) release version 1.0
23c2812 src/file edited
1a14447 lib/file2 edited
eb8a4a5 initial commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git switch a17b5c6
fatal: a branch is expected, got commit 'a17b5c6'
hint: If you want to detach HEAD at the commit, try again with the --
detach option.
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ ls
dummy.txt  hello.txt  index.txt  lib/  new.txt  src/
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ vi index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git add .
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ vi index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.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:   index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git restore --staged index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (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@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git restore index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
```

```
$ git status
On branch master
nothing to commit, working tree clean
```

//Git cloning

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/GitRemoteConnec (master)
$ cd ..
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT
$ mkdir clone
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT
$ cd clone
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/clone
$ git clone https://github.com/JDStraughan/html5-snake.git
Cloning into 'html5-snake'...
remote: Enumerating objects: 46, done.
remote: Total 46 (delta 0), reused 0 (delta 0), pack-reused 46
Receiving objects: 100% (46/46), 9.39 KiB | 505.00 KiB/s, done.
Resolving deltas: 100% (15/15), done.
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/clone
$ ls
html5-snake/
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/clone
$ cd html5-snake/
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/clone/html5-snake (master)
$ ls
README.md  game.js  index.html  page.css
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/clone/html5-snake (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/clone/html5-snake (master)
```

```
$ git log --oneline
4e30495 (HEAD -> master, origin/master, origin/HEAD) Merge pull
request #1 from themightychris/master
7d11cf1 var declarations, minor syntax errors, some style
inconsistency, and no global object prototype modification
ab188d1 Updating for firefox - thanks simevidas for fix!
e3fe18a README for GitHub
d63b8e6 Adding some instructions.
1668a1a Snakes should be green Messages should be drawn last for
layering
004e697 Add control key text
0d773b9 Speed up more often
c1f6202 Basic layout updates
bcc6926 Moving css to external and updating title
45ef7bd Snake size is proportionate to canvas size Canvas must be
640x480 or 320x240
14e4c21 Show the score
e00d93a Replaced this with named scope for readability Updated food
spawn generation to proper ratio against snake size (provided snake is
8px in 320x240 and 16px in 640x480)
d939ae2 Added requestAnimationFrame() Added fps clocking Refactored
drawing of boxes to game.drawBox() Reset snake also resets direction
(defaults left) Increase fps every 8 food consumptions to increase
difficulty Simplified food.set()
b4b4271 Initial commit of working poc game

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/clone/html5-snake (master)
$ code . //to open file in vs code
```

```
//Git SSH key set up
```

```
vlab@VEDA2F107 MINGW64 ~  
$ ssh-keygen  
Generating public/private ed25519 key pair.  
Enter file in which to save the key (/c/Users/vlab/.ssh/id_ed25519):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /c/Users/vlab/.ssh/id_ed25519  
Your public key has been saved in /c/Users/vlab/.ssh/id_ed25519.pub  
The key fingerprint is:  
SHA256:Tr2v08aH1ldPHwFb2ts84Yqw85huE+bMT1DqK33X/iO vlab@VEDA2F107  
The key's randomart image is:  
+--[ED25519 256]--+  
|                    |  
|. .                |  
| *                 |
```

```

|          .o  + |
|          So.   ..=|
|          o.+..  ==|
|          B.=+..oo*|
|          . X*+=OE.*|
|          +=B*oo.o=|
+-----[SHA256]-----+

```

```

vlab@VEDA2F107 MINGW64 ~
$ cd .ssh

```

```

vlab@VEDA2F107 MINGW64 ~/.ssh
$ ls
id_ed25519  id_ed25519.pub  known_hosts

```

```

vlab@VEDA2F107 MINGW64 ~/.ssh
$ cat id_ed25519
id_ed25519      id_ed25519.pub

```

```

vlab@VEDA2F107 MINGW64 ~/.ssh
$ cat id_ed25519
id_ed25519      id_ed25519.pub

```

```

vlab@VEDA2F107 MINGW64 ~/.ssh
$ cat id_ed25519.pub
ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIJ6JtI2YQKiUPN3lujgXAgiCdm35IzOjAqKRd8t1Z6xb
vlab@VEDA2F107

```

```

vlab@VEDA2F107 MINGW64 ~/.ssh
$ ssh -T git@github.com
Hi lochu-55! You've successfully authenticated, but GitHub does not
provide shell access.

```

```

vlab@VEDA2F107 MINGW64 ~/.ssh

```

//Git remote add,remove,rename

```

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT
$ cd MyGit

```

```

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/MyGit (master)
$ git branch
* master

```

```

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/MyGit (master)
$ git remote

```

```

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/MyGit (master)
$ git remote -v

```

```

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/MyGit (master)
$ git remote add origin https://github.com/lochu-55/My_GIT_Repo.git

```

```

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/MyGit (master)
$ git remote -v
origin https://github.com/lochu-55/My_GIT_Repo.git (fetch)
origin https://github.com/lochu-55/My_GIT_Repo.git (push)

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/MyGit (master)
$ git remote rename origin githuburl

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/MyGit (master)
$ git remote
githuburl

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/MyGit (master)
$ git remote -v
githuburl https://github.com/lochu-55/My_GIT_Repo.git (fetch)
githuburl https://github.com/lochu-55/My_GIT_Repo.git (push)

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/MyGit (master)
$ git remote remove githuburl

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/MyGit (master)
$ git remote -v

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/MyGit (master)
$ git remote add origin https://github.com/lochu-55/My_GIT_Repo.git

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/MyGit (master)
$ git remote -v
origin https://github.com/lochu-55/My_GIT_Repo.git (fetch)
origin https://github.com/lochu-55/My_GIT_Repo.git (push)

```

//Git push

```

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (branch1)
$ git remote add origin1 git@github.com:lochu-55/god.git

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (branch1)
$ git push origin1 branch1
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 218 bytes | 218.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:lochu-55/god.git
 * [new branch]      branch1 -> branch1

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (branch1)
$ git switch -
Switched to branch 'master'

```



```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (master)
```

```
$ git push origin1 branch1  
Everything up-to-date
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (master)
```

```
$ git push origin1 master  
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
remote:  
remote: Create a pull request for 'master' on GitHub by visiting:  
remote:      https://github.com/lochu-55/god/pull/new/master  
remote:  
To github.com:lochu-55/god.git  
* [new branch]      master -> master
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (master)
```

```
$ ls  
file1
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (master)
```

```
$ git push origin master  
remote: Repository not found.  
fatal: repository 'https://github.com/lochu-55/god.git/' not found
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (master)
```

```
$ git remote  
origin  
origin1
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (master)
```

```
$ git remote remove origin1
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (master)
```

```
$ git remote remove origin
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (master)
```

```
$ git remote add origin git@github.com:lochu-55/god.git
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (master)
```

```
$ ls  
file1
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (master)
```

```
$ git push origin master  
Everything up-to-date
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (master)
```

```
$ cd ..
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT
```

```
$ cd git1
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (master)
```

```
$ cat >> file2  
hello  
file2
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (master)
```

```
$ git push origin master
```

Everything up-to-date

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file2
```

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

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (master)
$ git add .
warning: in the working copy of 'file2', LF will be replaced by CRLF
the me Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (master)
$ git commit -m "file2 added"
[master 8e58fb6] file2 added
1 file changed, 2 insertions(+)
create mode 100644 file2
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (master)
$ ls
file1  file2
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (master)
$ git push origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 275 bytes | 275.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:lochu-55/god.git
    27db2cc..8e58fb6  master -> master
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (master)
$ git pull origin master
From github.com:lochu-55/god
 * branch      master      -> FETCH_HEAD
Already up to date.
```

//git push upstream (-u option)

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (feature)
$ git branch
```

```
branch1
* feature
master
new_branch
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (feature)
$ git push origin feature:new_branch
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 280 bytes | 280.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:lochu-55/god.git
    07f4b20..d64b3b4  feature -> new_branch
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (feature)
$ vi feature.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (feature)
$ git add .
warning: in the working copy of 'feature.txt', LF will be replaced by
CRLF the next time Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (feature)
$ git commit -m "feature added"
[feature 99fb4b9] feature added
1 file changed, 1 insertion(+)
create mode 100644 feature.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (feature)
$ git push
fatal: The current branch feature has no upstream branch.
To push the current branch and set the remote as upstream, use
```

```
git push --set-upstream origin feature
```

To have this happen automatically for branches without a tracking upstream, see 'push.autoSetupRemote' in 'git help config'.

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (feature)
$ git push -u origin1 feature
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 280 bytes | 280.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'feature' on GitHub by visiting:
remote:   https://github.com/lochu-55/god/pull/new/feature
remote:
To https://github.com/lochu-55/god.git
 * [new branch]      feature -> feature
branch 'feature' set up to track 'origin1/feature'.
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (feature)
$ vi feature.txt

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (feature)
$ git add .
warning: in the working copy of 'feature.txt', LF will be replaced by
CRLF the next time Git touches it

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (feature)
$ git commit -m "second feature commit"
[feature e376f75] second feature commit
1 file changed, 1 insertion(+)

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git1 (feature)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 280 bytes | 280.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/lochu-55/god.git
99fb4b9..e376f75 feature -> feature
```

//git remote branch tracking

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT
$ cd git_remote_tracking/

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_remote_tracking
$ git clone https://github.com/lochu-55/god.git
Cloning into 'god'...
remote: Enumerating objects: 24, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 24 (delta 6), reused 18 (delta 2), pack-reused 0
Receiving objects: 100% (24/24), done.
Resolving deltas: 100% (6/6), done.

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_remote_tracking
$ ls
god/
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_remote_tracking
$ cd god

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_remote_tracking/god (branch1)
$ git checkout master
Switched to a new branch 'master'
branch 'master' set up to track 'origin/master'.

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_remote_tracking/god (master)
$ git branch
  branch1
* master

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_remote_tracking/god (master)
$ ls
bye.txt  file1  file2

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_remote_tracking/god (master)
$ git branch -r
  origin/HEAD -> origin/branch1
  origin/branch1
  origin/feature
  origin/master
  origin/new_branch

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_remote_tracking/god (master)
$ ls
bye.txt  file1  file2

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_remote_tracking/god (master)
$ git switch feature
Switched to a new branch 'feature'
branch 'feature' set up to track 'origin/feature'.

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_remote_tracking/god (feature)
$ ls
branch.txt  bye.txt  feature.txt  file1  file2  hello.txt  new_branch

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_remote_tracking/god (feature)
$ git branch -r
  origin/HEAD -> origin/branch1
  origin/branch1
  origin/feature
  origin/master
  origin/new_branch

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_remote_tracking/god (feature)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_remote_tracking/god (master)
$ vi master.txt

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_remote_tracking/god (master)
$ git add .
```

warning: in the working copy of 'master.txt', LF will be replaced by CRLF the next time Git touches it

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_remote_tracking/god (master)
$ git commit -m "master added"
[master d9b336b] master added
1 file changed, 1 insertion(+)
create mode 100644 master.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_remote_tracking/god (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
    (use "git push" to publish your local commits)
```

nothing to commit, working tree clean

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_remote_tracking/god (master)
$ git remote
origin
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_remote_tracking/god (master)
$ git push origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 278 bytes | 278.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/lochu-55/god.git
    30408c3..d9b336b  master -> master
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_remote_tracking/god (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.
```

nothing to commit, working tree clean

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_remote_tracking/god (master)
$ git log --oneline
d9b336b (HEAD -> master, origin/master) master added
30408c3 Create bye.txt
8e58fb6 file2 added
27db2cc (origin/branch1, origin/HEAD, branch1) first commit
```

//restore command usage

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-fetch (master)
$ vi kk.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-fetch (master)
$ ls
fetch.txt  index.txt  kk.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-fetch (master)
$ git add .
warning: in the working copy of 'kk.txt', LF will be replaced by CRLF
the next time Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-fetch (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   kk.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-fetch (master)
$ git commit -m "kk"
[master 3b67c82] kk
1 file changed, 1 insertion(+)
create mode 100644 kk.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-fetch (master)
$ vi kk.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-fetch (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:   kk.txt
```

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

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-fetch (master)
$ git restore kk.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-fetch (master)
$ vi kk.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-fetch (master)
$ git status
On branch master
nothing to commit, working tree clean
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-fetch (master)
$ vi kk.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-fetch (master)
```

```
$ git add kk.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-fetch (master)
```

```
$ cat kk.txt
```

```
hiiii
```

```
bye
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-fetch (master)
```

```
$ git restore --staged kk.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-fetch (master)
```

```
$ cat kk.txt
```

```
hiiii
```

```
bye
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-fetch (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:   kk.txt
```

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

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-fetch (master)
```

```
$ git restore kk.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-fetch (master)
```

```
$ cat kk.txt
```

```
hiiii
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-fetch (master)
```

```
$ git ls-files
```

```
fetch.txt
```

```
index.txt
```

```
kk.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-fetch (master)
```

```
$ git restore --staged kk.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-fetch (master)
```

```
$ git ls-files
```

```
fetch.txt
```

```
index.txt
```

```
kk.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-fetch (master)
```

```
$ cat kk.txt
```

```
hiiii
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-fetch (master)
```

```
$ git status
```

```
On branch master
```

```
nothing to commit, working tree clean
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-fetch (master)
```



```
$ git status
On branch master
nothing to commit, working tree clean
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-fetch (master)
$ git restore --staged kk.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-fetch (master)
$ git status
On branch master
nothing to commit, working tree clean
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-fetch (master)
$ git ls-files
fetch.txt
index.txt
kk.txt
```

//Git fetch and pull

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull (master)
$ git fetch origin master
From github.com:lochu-55/My_GIT_Repo
* branch          master       -> FETCH_HEAD
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull (master)
$ ls
AutoGen/ a.txt  bye.txt  hello.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull (master)
$ git status
On branch master
nothing to commit, working tree clean
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull (master)
$ git merge FETCH_HEAD
Updating 62447d7..36756ac
Fast-forward
 bye.txt | 1 +
 1 file changed, 1 insertion(+)
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull (master)
$ LS
AutoGen a.txt  bye.txt  hello.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull (master)
$ cat bye.txt
hello
everyone
how are you
bye
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull (master)
$ git fetch origin master
From github.com:lochu-55/My_GIT_Repo
* branch          master       -> FETCH_HEAD
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull (master)
$ cat bye.txt
hello
everyone
how are you
bye
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull (master)
$ git branch -r
origin/master
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull (master)
$ git log origin/master
commit 01e3f078b14142f28ced1e13f231baeb08e503a4 (origin/master)
Author: lochu-55 <93853130+lochu-55@users.noreply.github.com>
Date:   wed Mar 13 10:57:48 2024 +0530
```

Update byeeeeee

```
commit 36756acf7f6acd476874e7406aa6cf13c4d17aff (HEAD -> master)
Author: lochu-55 <93853130+lochu-55@users.noreply.github.com>
Date:   Wed Mar 13 10:51:29 2024 +0530
```

Update bye.txt

```
commit 62447d75ab8b2c2c5569692b68f583a148ccbfd4
Author: lochu-55 <93853130+lochu-55@users.noreply.github.com>
Date:   Wed Mar 13 10:45:36 2024 +0530
```

Update a.txt

```
commit 03a8e0a30f67708dd4817619cb493b51739ab15d
Author: Lochani <lochu5vilehya@gmail.com>
Date:   Tue Mar 5 10:00:49 2024 +0530
```

new folder f1

```
commit afe6c767cc0236ba5618e34422edae0b2cc4d084
Author: Lochani <lochu5vilehya@gmail.com>
Date:   Tue Mar 5 09:58:46 2024 +0530
```

modified folder autogen

```
commit f167cb884765b6a3218071533002d5094b797181
Author: Lochani <lochu5vilehya@gmail.com>
Date:   Tue Mar 5 09:49:57 2024 +0530
```

modified the autogen in gitignore

```
commit 4588fe90995d536fb7cd16e5ec5dc1c995b4e83e
Author: Lochani <lochu5vilehya@gmail.com>
Date:   Tue Mar 5 09:47:54 2024 +0530
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull (master)
$ ^C
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull (master)
$ git show 01e3f078b14142f28ced1e13f231baeb08e503a4
commit 01e3f078b14142f28ced1e13f231baeb08e503a4 (origin/master)
Author: lochu-55 <93853130+lochu-55@users.noreply.github.com>
Date:   Wed Mar 13 10:57:48 2024 +0530
```

Update byeeeeee

```
diff --git a/bye.txt b/bye.txt
index 8abec41..dbf34eb 100644
--- a/bye.txt
+++ b/bye.txt
@@ -2,3 +2,4 @@ hello
     everyone
     how are you
     bye
+lo!!!
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull (master)
```

```
$ git status
```

```
On branch master
```

```
nothing to commit, working tree clean
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull (master)
```

```
$ git pull origin
```

```
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.
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull (master)
```

```
$ git pull origin master
```

```
From github.com:lochu-55/My_GIT_Repo
```

```
* branch                master      -> FETCH_HEAD
```

```
Updating 36756ac..01e3f07
```

```
Fast-forward
```

```
bye.txt | 1 +
```

```
1 file changed, 1 insertion(+)
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull (master)
```

```
$ git log --oneline
```

```
01e3f07 (HEAD -> master, origin/master) Update byeeeeee
```

```
36756ac Update bye.txt
```

```
62447d7 Update a.txt
```

```
03a8e0a new folder f1
```

```
afe6c76 modified folder autogen
```

```
f167cb8 modified the autogen in gitignore
```

```
4588fe9 modify text in a
```

```
1a83c98 added autogen dir to gitignore
```

```
a3d713a added dir
```

```
e7839b6 fixed file tracking
```

```
84a1091 added gitignore
```

```
0e37699 text edited in script
```

```
90b91c6 added text to the bye file
```

```
fc86003 Reapply "modified text in script"
```

```
92b0f51 Revert "modified text in script"
```

```
c453671 modified text in script
```

```
bb8ca8e untracked files
```

```
093da3c new file added
```

//Git pull conflicts(if we modify the same file in both remote and local repo)

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull (master)
$ vi hello.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull (master)
$ git commit -am "modified in local repo"
warning: in the working copy of 'hello.txt', LF will be replaced by
CRLF the next time Git touches it
[master 58e5a21] modified in local repo
1 file changed, 2 insertions(+), 1 deletion(-)
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull (master)
$ git pull origin master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 936 bytes | 7.00 KiB/s, done.
From github.com:lochu-55/My_GIT_Repo
 * branch                master      -> FETCH_HEAD
    01e3f07..6000d0c      master      -> origin/master
Auto-merging hello.txt
CONFLICT (content): Merge conflict in hello.txt
Automatic merge failed; fix conflicts and then commit the result.
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull (master|MERGING)
$ vi hello.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull (master|MERGING)
$ git add .
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull (master|MERGING)
$ git commit -m "resolved pull conflict"
[master 220cb8b] resolved pull conflict
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull (master)
$ git status
On branch master
nothing to commit, working tree clean
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull (master)
$ git log --oneline
220cb8b (HEAD -> master) resolved pull conflict
6000d0c (origin/master) modify in remote repo
58e5a21 modified in local repo
01e3f07 Update byeeeeee
36756ac Update bye.txt
62447d7 Update a.txt
03a8e0a new folder f1
afe6c76 modified folder autogen
f167cb8 modified the autogen in gitignore
4588fe9 modify text in a
1a83c98 added autogen dir to gitignore
```

```
a3d713a added dir
e7839b6 fixed file tracking
84a1091 added gitignore
0e37699 text edited in script
90b91c6 added text to the bye file
fc86003 Reapply "modified text in script"
92b0f51 Revert "modified text in script"
c453671 modified text in script
bb8ca8e untracked files
093da3c new file added
c33d4dc initial commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull (master)
$ git push origin master
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 588 bytes | 588.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:lochu-55/My_GIT_Repo.git
6000d0c..220cb8b master -> master
```

//Gitpages creation

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page
$ git clone git@github.com:lochu-55/gitpage.github.io.git
Cloning into 'gitpage.github.io'...
warning: You appear to have cloned an empty repository.
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page
$ cd gitpage.github.io
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page/gitpage.github.io (main)
$ vi index.html
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page/gitpage.github.io (main)
$ git add .
warning: in the working copy of 'index.html', LF will be replaced by
CRLF the next time Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page/gitpage.github.io (main)
$ git commit -m "index.html added"
[main (root-commit) e7fba2c] index.html added
1 file changed, 1 insertion(+)
create mode 100644 index.html
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page/gitpage.github.io (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
```

```
Writing objects: 100% (3/3), 229 bytes | 32.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:lochu-55/gitpage.github.io.git
* [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page/gitpage.github.io (main)
$ vi index.html
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page/gitpage.github.io (main)
$ git add .
warning: in the working copy of 'index.html', LF will be replaced by
CRLF the next time Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page/gitpage.github.io (main)
$ git commit -m "index file modified"
[main 9264c11] index file modified
1 file changed, 1 insertion(+)
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page/gitpage.github.io (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 269 bytes | 67.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:lochu-55/gitpage.github.io.git
e7fba2c..9264c11  main -> main
```

The screenshot shows the GitHub Pages configuration interface. On the left is a sidebar with navigation links: General, Access, Collaborators, Moderation options, Code and automation, Branches, Tags, Rules, Actions, Webhooks, Environments, Codespaces, and Pages (which is highlighted). The main content area is titled 'GitHub Pages' and contains the following information:

- A description: 'GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.'
- A status box indicating the site is live at <https://lochu-55.github.io/gitpage.github.io/>, last deployed by 'lochu-55'.
- A 'Build and deployment' section with a 'Source' dropdown set to 'Deploy from a branch'.
- A 'Branch' section stating the site is built from the 'main' branch, with a link to learn more about configuring the publishing source.
- Buttons for selecting the branch ('main') and the directory ('/ (root)'), along with a 'Save' button.
- A link to 'Learn how to add a Jekyll theme to your site.'

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page
$ git clone git@github.com:lochu-55/GitwebPage.git
Cloning into 'GitwebPage'...
warning: You appear to have cloned an empty repository.
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page
$ cd GitwebPage/
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page/GitWebPage (main)
$ vi index.html
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page/GitWebPage (main)
$ git add .
warning: in the working copy of 'index.html', LF will be replaced by
CRLF the next time Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page/GitWebPage (main)
$ git commit -m "index added"
[main (root-commit) 19c77cc] index added
1 file changed, 44 insertions(+)
create mode 100644 index.html
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page/GitWebPage (main)
$ git checkout -b feature
Switched to a new branch 'feature'
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page/GitWebPage (feature)
$ vi feature.html
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page/GitWebPage (feature)
$ vi styles.css
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page/GitWebPage (feature)
$ git status
On branch feature
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    feature.html
    styles.css
```

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

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page/GitWebPage (feature)
$ git add .
warning: in the working copy of 'feature.html', LF will be replaced by
CRLF the next time Git touches it
warning: in the working copy of 'styles.css', LF will be replaced by
CRLF the next time Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page/GitWebPage (feature)
$ git commit -m "added feature styles"
[feature 20fedac] added feature styles
2 files changed, 93 insertions(+)
create mode 100644 feature.html
create mode 100644 styles.css
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page/GitWebPage (feature)
$ git push -u origin feature
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 1.36 KiB | 1.36 MiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
```



```
To github.com:lochu-55/GitWebPage.git
* [new branch]      feature -> feature
branch 'feature' set up to track 'origin/feature'.
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page/GitWebPage (feature)
$ git checkout main
Switched to branch 'main'
Your branch is based on 'origin/main', but the upstream is gone.
(use "git branch --unset-upstream" to fixup)
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page/GitWebPage (main)
$ git push -u origin main
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'main' on GitHub by visiting:
remote:      https://github.com/lochu-55/GitWebPage/pull/new/main
remote:
To github.com:lochu-55/GitWebPage.git
* [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page/GitWebPage (main)
$ git log
commit 19c77cceb5a0d4dc26b931dc07bf00fef7c79f74 (HEAD -> main,
origin/main)
Author: lochani <lochusvilehya@gmail.com>
Date:   Wed Mar 13 14:22:46 2024 +0530
```

index added

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page/GitWebPage (main)
$ git checkout feature
Switched to branch 'feature'
Your branch is up to date with 'origin/feature'.
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page/GitWebPage (feature)
$ git log
commit 20fedacbdccd8a45e059fe28c17d922178340427 (HEAD -> feature,
origin/feature)
Author: lochani <lochusvilehya@gmail.com>
Date:   Wed Mar 13 14:25:16 2024 +0530
```

added feature styles

```
commit 19c77cceb5a0d4dc26b931dc07bf00fef7c79f74 (origin/main, main)
Author: lochani <lochusvilehya@gmail.com>
Date:   Wed Mar 13 14:22:46 2024 +0530
```

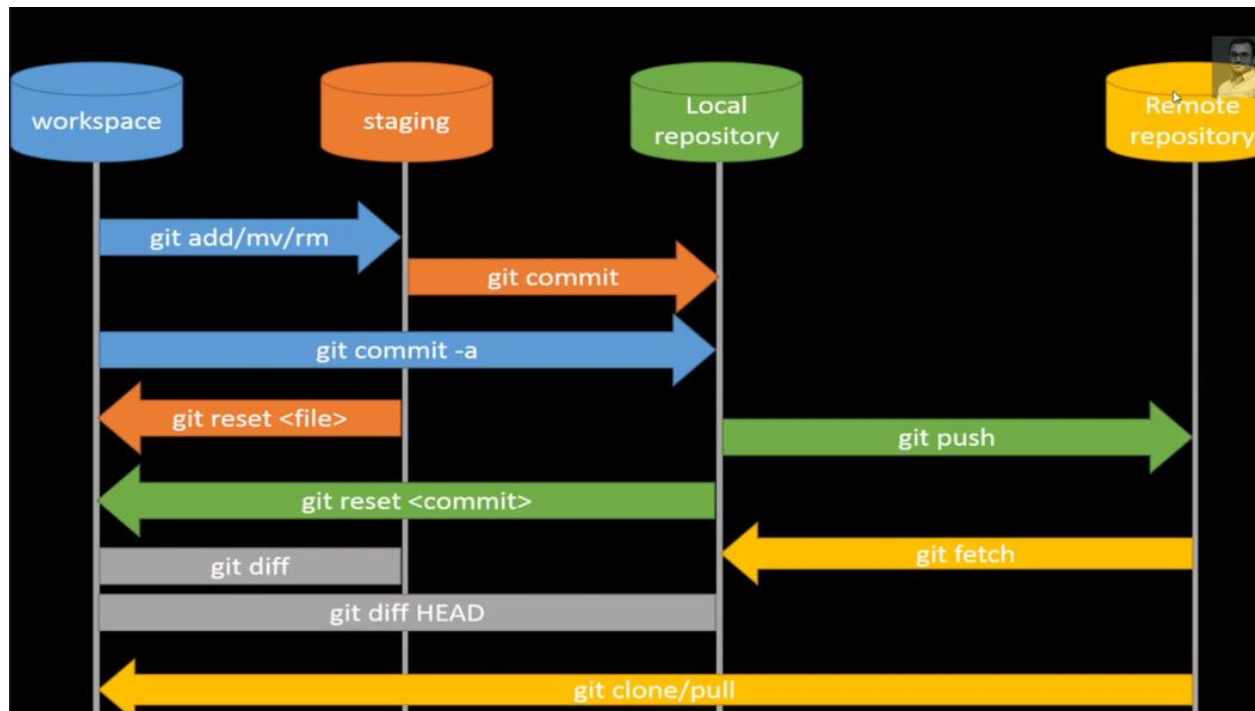
index added

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page/GitWebPage (feature)
$ git log --oneline
20fedac (HEAD -> feature, origin/feature) added feature styles
19c77cc (origin/main, main) index added
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page/GitWebPage (feature)
$ git branch -r
origin/feature
```

origin/main

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-page/GitWebPage (feature)
$ git branch
* feature
  Main
```



```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict
$ git clone git@github.com:lochu-55/GitWebPage.git
Cloning into 'GitWebPage'...
remote: Enumerating objects: 41, done.
remote: Counting objects: 100% (41/41), done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 41 (delta 11), reused 30 (delta 7), pack-reused 0
Receiving objects: 100% (41/41), 7.84 KiB | 730.00 KiB/s, done.
Resolving deltas: 100% (11/11), done.
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict
$ cd GitWebPage/
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitWebPage
(feature)
$ ls
feature.html  hello.txt  index.html  main.txt  styles.css
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitwebPage
(feature)
$ vi main.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitwebPage
(feature)
$ git add .
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitwebPage
(feature)
$ git commit -m "added hii in main git"
[feature eef824d] added hii in main git
1 file changed, 1 insertion(+)
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitwebPage
(feature)
$ git branch
* feature
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitwebPage
(feature)
$ git push -u origin feature
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 270 bytes | 67.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote: Bypassed rule violations for refs/heads/feature:
remote:
remote: - Changes must be made through a pull request.
remote:
To github.com:lochu-55/GitwebPage.git
   78e65b9..eef824d  feature -> feature
branch 'feature' set up to track 'origin/feature'.
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitwebPage
(feature)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 929 bytes | 4.00 KiB/s, done.
From github.com:lochu-55/GitwebPage
 * [new branch]      lochu-55-patch-1 -> origin/lochu-55-patch-1
Already up to date.
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitwebPage
(feature)
$ git branch
* feature
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitwebPage
(feature)
$ git checkout main
Switched to a new branch 'main'
```

branch 'main' set up to track 'origin/main'.

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitwebPage
(main)
$ git branch
  feature
* main
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitwebPage
(main)
$ git pull
Already up to date.
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitwebPage
(main)
$ git push
Everything up-to-date
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitwebPage
(main)
$ ls
feature.html  hello.txt  hii.txt  index.html  main.txt  styles.css
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitwebPage
(main)
$ vi hii.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitwebPage
(main)
$ git add .
```

g

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitwebPage
(main)
$ git commit -m "update hii in main git"
[main 7b057f3] update hii in main git
1 file changed, 1 insertion(+)
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitwebPage
(main)
```

```
$ git push
To github.com:lochu-55/GitwebPage.git
! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'github.com:lochu-55/GitwebPage.git'
hint: Updates were rejected because the remote contains work that you
do not
hint: have locally. This is usually caused by another repository
pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for
details.
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitwebPage
(main)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
```

```
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 922 bytes | 5.00 KiB/s, done.
From github.com:lochu-55/GitWebPage
   b619faf..3fb1c52  main       -> origin/main
Auto-merging hii.txt
CONFLICT (content): Merge conflict in hii.txt
Automatic merge failed; fix conflicts and then commit the result.
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitWebPage
(main|MERGING)
$ vi hii.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitWebPage
(main|MERGING)
$ git add .
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitWebPage
(main|MERGING)
$ git commit -m "resolved conflict in hii"
[main 3ccalle] resolved conflict in hii
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitWebPage
(main)
$ git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 548 bytes | 548.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:lochu-55/GitWebPage.git
   3fb1c52..3ccalle  main -> main
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitWebPage
(main)
$ ls
feature.html  hello.txt  hii.txt  index.html  main.txt  styles.css
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitWebPage
(main)
$ git branch
feature
* main
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitWebPage
(main)
$ git branch -r
origin/HEAD -> origin/feature
origin/bugfix
origin/feature
origin/lochu-55-patch-1
origin/main
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitWebPage
(main)
$ git branch -D origin/lochu-55-patch-1
```

error: branch 'origin/lochu-55-patch-1' not found.
Did you forget --remote?

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitwebPage
(main)
```

```
$ git branch -r
  origin/HEAD -> origin/feature
  origin/bugfix
  origin/feature
  origin/lochu-55-patch-1
  origin/main
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitwebPage
(main)
```

```
$ git fetch --prune
From github.com:lochu-55/GitwebPage
- [deleted]          (none)        -> origin/lochu-55-patch-1
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (2/2), 947 bytes | 9.00 KiB/s, done.
    eef824d..e0cade1  feature      -> origin/feature
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitwebPage
(main)
```

```
$ git branch -r
  origin/HEAD -> origin/feature
  origin/bugfix
  origin/feature
  origin/main
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitwebPage
(main)
```

```
$ git switch bugfix
Switched to a new branch 'bugfix'
branch 'bugfix' set up to track 'origin/bugfix'.
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitwebPage
(bugfix)
```

```
$ ls
feature.html  hello.txt  hii.txt  index.html  main.txt  styles.css
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitwebPage
(bugfix)
```

```
$ vi main.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitwebPage
(bugfix)
```

```
$ git status
On branch bugfix
Your branch is up to date with 'origin/bugfix'.
```

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: main.txt

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

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitwebPage
(bugfix)
$ git add .
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitwebPage
(bugfix)
$ git commit -m "edited main in bugfix"
[bugfix decd7e9] edited main in bugfix
1 file changed, 1 insertion(+)
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-pull-req-conflict/GitwebPage
(bugfix)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 272 bytes | 272.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:lochu-55/GitwebPage.git
   b50eafd..decd7e9  bugfix -> bugfix
```

lochu-55 / GitWebPage

bugfix had recent pushes on Mar 14 [Compare & pull request](#)

feature 3 Branches 0 Tags [Add file](#) [Code](#)

lochu-55	Merge pull request #6 from lochu-55/main	e0cade1 · 1 hour ago	21 Commits
feature.html	edited feature file	19 hours ago	
hello.txt	resolved conflicts in pull request	18 hours ago	
hii.txt	resolved conflict in hii	1 hour ago	
index.html	modified index	19 hours ago	
main.txt	added hii in main git	1 hour ago	

About
No description, website, or topics provided.

Activity
0 stars
1 watching
0 forks

Releases
No releases published
[Create a new release](#)

Packages
No packages published
[Publish your first package](#)

lochu-55 / GitWebPage

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#) or [learn more about diff comparisons](#).

base: feature ← compare: bugfix ✖ Can't automatically merge. Don't worry, you can still create the pull request.

Add a title

Add a description
Write Preview **H B I** **≡ <> ↻** **≡ ≡ ≡** **🔗 @ ↻ ↶**

Add your description here...

Reviewers
No reviews

Assignees
No one—[assign yourself](#)

Labels
None yet

Projects
None yet

//GIT tags

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT
$ git clone https://github.com/InstaPy/InstaPy.git
Cloning into 'InstaPy'...
remote: Enumerating objects: 12944, done.
remote: Counting objects: 100% (2/2), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 12944 (delta 0), reused 2 (delta 0), pack-reused 12942
Receiving objects: 100% (12944/12944), 81.70 MiB | 4.38 MiB/s, done.
Resolving deltas: 100% (8717/8717), done.

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT
$ cd InstaPy/

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/InstaPy (master)
$ git branch -r
origin/HEAD -> origin/master
origin/dependabot/npm_and_yarn/docusaurus/express-4.18.2
origin/dependabot/npm_and_yarn/docusaurus/http-cache-semantics-4.1.1
origin/dependabot/npm_and_yarn/docusaurus/json5-1.0.2
origin/dependabot/npm_and_yarn/docusaurus/ua-parser-js-0.7.33
origin/dev
origin/gh-pages
origin/login_xpath_update
origin/master
origin/new_graphql_approach_integrate-it-on-relationship_later

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/InstaPy (master)
$ git tag
0.1.1
0.1.2
0.1.3
0.2.0
0.2.1
0.2.2
0.2.3
0.3.0
0.3.1
0.3.2
0.3.3
0.3.4
0.4.0
0.4.1
0.4.2
0.4.3
0.5.0
0.5.1
0.5.2
0.5.3
0.5.4
0.5.5
0.5.6
0.5.7
0.5.8
0.6.0
```

```
0.6.1
0.6.10
0.6.11
0.6.12
0.6.13
0.6.14
0.6.15
0.6.16
0.6.2
0.6.3
0.6.4
0.6.5
0.6.6
0.6.7
0.6.8
0.6.9
legacy
v0.1.0
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/InstaPy (master)
```

```
$ git tag -l *6*
```

```
0.5.6
0.6.0
0.6.1
0.6.10
0.6.11
0.6.12
0.6.13
0.6.14
0.6.15
0.6.16
0.6.2
0.6.3
0.6.4
0.6.5
0.6.6
0.6.7
0.6.8
0.6.9
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/InstaPy (master)
```

```
$ git diff 0.6.2 0.6.3
```

```
diff --git a/.travis.yml b/.travis.yml
```

```
index 850a2b7..918f8dd 100644
```

```
--- a/.travis.yml
```

```
+++ b/.travis.yml
```

```
@@ -1,5 +1,4 @@
```

```
dist: xenial
```

```
-sudo: false
```

```
language: python
```

```
python: "3.7"
```

```
cache: pip
```

```
@@ -9,7 +8,7 @@ branches:
```

```
- legacy
```

```
- experimental
```

```
-install: pip install tox tox-venv
```

```
+install: pip install tox
```

//GIT light-weighted tags

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT
$ mkdir git-tag
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT
$ cd git-tag
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag
$ git init
Initialized empty Git repository in C:/Users/vlab/Desktop/GIT/git-tag/.git/
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ vi index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git add .
warning: in the working copy of 'index.txt', LF will be replaced by
CRLF the next time Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git commit -m "initial commit"
[master (root-commit) aec5ff0] initial commit
1 file changed, 1 insertion(+)
create mode 100644 index.txt
```

g

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git log
commit aec5ff0ce22808d1d5415730a42f5c8d29b10bb8 (HEAD -> master)
Author: lochani <lochu5vilehya@gmail.com>
Date: Thu Mar 14 14:07:05 2024 +0530
```

initial commit

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git tag
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git tag v1.0.0
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git tag
v1.0.0
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ vi index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git add .
warning: in the working copy of 'index.txt', LF will be replaced by
CRLF the next time Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git commit -m "second commit"
```

```
[master 65ae680] second commit
1 file changed, 1 insertion(+)
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ fit log
bash: fit: command not found
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git log
commit 65ae680011b2f65768a76118ec230f1badb4bc9d (HEAD -> master)
Author: lochani <lochu5vilehya@gmail.com>
Date: Thu Mar 14 14:08:24 2024 +0530
```

second commit

```
commit aec5ff0ce22808d1d5415730a42f5c8d29b10bb8 (tag: v1.0.0)
Author: lochani <lochu5vilehya@gmail.com>
Date: Thu Mar 14 14:07:05 2024 +0530
```

initial commit

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git checkout v1.0.0
Note: switching to 'v1.0.0'.
```

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using -c with the switch command. Example:

```
git switch -c <new-branch-name>
```

Or undo this operation with:

```
git switch -
```

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at aec5ff0 initial commit

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag ((v1.0.0))
$ cat index.txt
this is first commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag ((v1.0.0))
$ git switch -
Previous HEAD position was aec5ff0 initial commit
Switched to branch 'master'
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ cat index.txt
this is first commit
```

this is 2nd commit

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git log --oneline
65ae680 (HEAD -> master) second commit
aec5ff0 (tag: v1.0.0) initial commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git tag v1.0.1
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git log --oneline
65ae680 (HEAD -> master, tag: v1.0.1) second commit
aec5ff0 (tag: v1.0.0) initial commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git tag
v1.0.0
v1.0.1
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git diff v1.0.0 v1.0.1
diff --git a/index.txt b/index.txt
index 2ed441a..a0c73a5 100644
--- a/index.txt
+++ b/index.txt
@@ -1,2 @@
  this is first commit
+this is 2nd commit
```

//GIT Annotated tags and push in github

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ vi new.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git add .
warning: in the working copy of 'new.txt', LF will be replaced by CRLF
the next time Git touches it
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git commit -m "new feature"
[master ccb6107] new feature
1 file changed, 1 insertion(+)
create mode 100644 new.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
```

```
$ git tag -a v1.1.0
hint: waiting for your editor to close the file... unix2dos:
converting file C:/Users/vlab/Desktop/GIT/git-tag/.git/TAG_EDITMSG to
DOS format...
dos2unix: converting file C:/Users/vlab/Desktop/GIT/git-
tag/.git/TAG_EDITMSG to Unix format...
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
```

```
$ git show v1.1.0
tag v1.1.0
Tagger: lochani <lochu5vilehya@gmail.com>
Date: Thu Mar 14 14:52:24 2024 +0530
```

this is annotated tag :)

```
commit ccb6107f094306ad7961cec76c77dcd809467429 (HEAD -> master, tag:
v1.1.0)
Author: lochani <lochu5vilehya@gmail.com>
Date: Thu Mar 14 14:52:12 2024 +0530
```

new feature

```
diff --git a/new.txt b/new.txt
new file mode 100644
index 0000000..3e75765
--- /dev/null
+++ b/new.txt
@@ -0,0 +1 @@
+new
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
```

```
$ git show v1.0.0
commit aec5ff0ce22808d1d5415730a42f5c8d29b10bb8 (tag: v1.0.0)
Author: lochani <lochu5vilehya@gmail.com>
Date: Thu Mar 14 14:07:05 2024 +0530
```

initial commit

```
diff --git a/index.txt b/index.txt
new file mode 100644
index 0000000..2ed441a
--- /dev/null
+++ b/index.txt
@@ -0,0 +1 @@
+this is first commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
```

```
$ git log --oneline
ccb6107 (HEAD -> master, tag: v1.1.0) new feature
65ae680 (tag: v1.0.1) second commit
aec5ff0 (tag: v1.0.0) initial commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
```

```
$ git remote add origin https://github.com/lochu-55/Git_tag.git
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
```

```
$ git push -u origin master
Enumerating objects: 9, done.
```

```
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (9/9), 713 bytes | 356.00 KiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/lochu-55/Git_tag/pull/new/master
remote:
To https://github.com/lochu-55/Git_tag.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git tag
v1.0.0
v1.0.1
v1.1.0
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git push origin v1.1.0
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 170 bytes | 170.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/lochu-55/Git_tag.git
 * [new tag]         v1.1.0 -> v1.1.0
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git push origin --tags
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/lochu-55/Git_tag.git
 * [new tag]         v1.0.0 -> v1.0.0
 * [new tag]         v1.0.1 -> v1.0.1
```


Unit 1 (8) 60 STM32 x Gettin Trash (8) De W STM32 How welco Insta Search (8) ST Veda + -

github.com/lochu-55/Git_tag/releases

YouTube Maps Gmail My CV - made usin... Aptitude Questions... Antenna Patterns an... All Bookmarks

lochu-55 / Git_tag

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Releases Tags

Draft a new release Find a release

in 13 seconds

lochu-55

v1.1.0

ccb6107

Compare

LaTeSt ReLeAsE Latest

This is latest release

- point1
- point2

Assets 2

Source code (zip) 40 minutes ago

Source code (tar.gz) 40 minutes ago

Type here to search 35°C Partly sunny 15:32 14-03-2024

Unit 1 (8) 60 STM32 x Gettin Trash (8) De W STM32 How welco Insta Search (8) ST Veda + -

github.com/lochu-55/Git_tag/tags

YouTube Maps Gmail My CV - made usin... Aptitude Questions... Antenna Patterns an... All Bookmarks

lochu-55 / Git_tag

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Releases Tags

Tags

v1.1.0 ...

40 minutes ago ccb6107 zip tar.gz Notes

v1.0.1 ...

1 hour ago 65ae680 zip tar.gz

v1.0.0 ...

1 hour ago aec5ff0 zip tar.gz

Type here to search 35°C Partly sunny 15:32 14-03-2024

//Git reflog

```
$ cd git_branching/
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git reflog
e2796cc (HEAD -> master) HEAD@{0}: checkout: moving from hello to
master
b2b7c09 (hello, feature) HEAD@{1}: checkout: moving from feature to
hello
b2b7c09 (hello, feature) HEAD@{2}: checkout: moving from master to
feature
e2796cc (HEAD -> master) HEAD@{3}: checkout: moving from
4931eba2eaf4d1c3d92fe2a8c5b1d2ec5e30fe76 to master
4931eba HEAD@{4}: checkout: moving from master to Head~4
e2796cc (HEAD -> master) HEAD@{5}: merge feature: Merge made by the
'ort' strategy.
a17b5c6 HEAD@{6}: commit: changes done in hello.txt (master)
bb6fab0 HEAD@{7}: checkout: moving from feature to master
b2b7c09 (hello, feature) HEAD@{8}: checkout: moving from master to
feature
bb6fab0 HEAD@{9}: checkout: moving from feature to master
b2b7c09 (hello, feature) HEAD@{10}: commit: edited hello.txt
bb6fab0 HEAD@{11}: checkout: moving from master to feature
bb6fab0 HEAD@{12}: commit: hello.txt added
e48c9a3 HEAD@{13}: checkout: moving from new_feature to master
6ba3361 (new_feature) HEAD@{14}: checkout: moving from dummy_branch to
new_feature
fdc643b (dummy_branch) HEAD@{15}: checkout: moving from master to
dummy_branch
e48c9a3 HEAD@{16}: merge dummy_branch: Merge made by the 'ort'
strategy.
4931eba HEAD@{17}: checkout: moving from dummy_branch to master
fdc643b (dummy_branch) HEAD@{18}: commit: dummy file
6ba3361 (new_feature) HEAD@{19}: checkout: moving from master to
dummy_branch
4931eba HEAD@{20}: commit: changes done to index file
6ba3361 (new_feature) HEAD@{21}: merge new_feature: Fast-forward
1ea9636 HEAD@{22}: checkout: moving from new_feature to master
6ba3361 (new_feature) HEAD@{23}: commit: new file added
1ea9636 HEAD@{24}: checkout: moving from master to new_feature
1ea9636 HEAD@{25}: checkout: moving from new_branch to master
4481071 HEAD@{26}: commit: log.txt edited
b41f54b (bugs) HEAD@{27}: checkout: moving from master to new_branch
1ea9636 HEAD@{28}: checkout: moving from new_branch to master
b41f54b (bugs) HEAD@{29}: Branch: renamed refs/heads/dummy to
refs/heads/new_branch
b41f54b (bugs) HEAD@{31}: checkout: moving from bugs to dummy
b41f54b (bugs) HEAD@{32}: checkout: moving from master to bugs
1ea9636 HEAD@{33}: commit: index.txt is committed
bf567b8 (dev) HEAD@{34}: checkout: moving from bugs to master
b41f54b (bugs) HEAD@{35}: checkout: moving from master to bugs
bf567b8 (dev) HEAD@{36}: checkout: moving from bugs to master
b41f54b (bugs) HEAD@{37}: checkout: moving from master to bugs
bf567b8 (dev) HEAD@{38}: checkout: moving from bugs to master
b41f54b (bugs) HEAD@{39}: commit: hello.txt in login
0f1b3a2 HEAD@{40}: checkout: moving from master to bugs
```

```
bf567b8 (dev) HEAD@{41}: checkout: moving from bugs to master
0f1b3a2 HEAD@{42}: checkout: moving from master to bugs
bf567b8 (dev) HEAD@{43}: checkout: moving from bugs to master
0f1b3a2 HEAD@{44}: commit: login folder added for logging
bf567b8 (dev) HEAD@{45}: checkout: moving from dev to bugs
bf567b8 (dev) HEAD@{46}: checkout: moving from master to dev
bf567b8 (dev) HEAD@{47}: merge dev: Fast-forward
eb8a4a5 HEAD@{48}: checkout: moving from dev to master
bf567b8 (dev) HEAD@{49}: commit: release version 1.0
23c2812 HEAD@{50}: commit: src/file edited
1a14447 HEAD@{51}: commit: lib/file2 edited
eb8a4a5 HEAD@{52}: checkout: moving from master to dev
eb8a4a5 HEAD@{53}: checkout: moving from bugs to master
eb8a4a5 HEAD@{54}: checkout: moving from master to bugs
eb8a4a5 HEAD@{55}: checkout: moving from dev to master
eb8a4a5 HEAD@{56}: checkout: moving from master to dev
eb8a4a5 HEAD@{57}: commit (initial): initial commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
```

```
$ git branch
bugs
dev
dummy_branch
feature
hello
* master
new_feature
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
```

```
$ git reflog show bugs
b41f54b (bugs) bugs@{0}: commit: hello.txt in login
0f1b3a2 bugs@{1}: commit: login folder added for logging
bf567b8 (dev) bugs@{2}: branch: Created from HEAD
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
```

```
$ git reflog show hello
b2b7c09 (hello, feature) hello@{0}: branch: Created from HEAD
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
```

```
$ git reflog show feature
b2b7c09 (hello, feature) feature@{0}: commit: edited hello.txt
bb6fab0 feature@{1}: branch: Created from HEAD
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
```

```
$ git reflog show dev
bf567b8 (dev) dev@{0}: commit: release version 1.0
23c2812 dev@{1}: commit: src/file edited
1a14447 dev@{2}: commit: lib/file2 edited
eb8a4a5 dev@{3}: branch: Created from HEAD
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
```

```
$ git reflog show dev@{2}
1a14447 dev@{2}: commit: lib/file2 edited
eb8a4a5 dev@{3}: branch: Created from HEAD
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
```

```
$ git log --oneline
e2796cc (HEAD -> master) Merge branch 'feature'
```

```
a17b5c6 changes done in hello.txt (master)
b2b7c09 (hello, feature) edited hello.txt
bb6fab0 hello.txt added
e48c9a3 Merge branch 'dummy_branch'
fdc643b (dummy_branch) dummy file
4931eba changes done to index file
6ba3361 (new_feature) new file added
1ea9636 index.txt is committed
bf567b8 (dev) release version 1.0
23c2812 src/file edited
1a14447 lib/file2 edited
eb8a4a5 initial commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git reflog show master
e2796cc (HEAD -> master) master@{0}: merge feature: Merge made by the
'ort' strategy.
a17b5c6 master@{1}: commit: changes done in hello.txt (master)
bb6fab0 master@{2}: commit: hello.txt added
e48c9a3 master@{3}: merge dummy_branch: Merge made by the 'ort'
strategy.
4931eba master@{4}: commit: changes done to index file
6ba3361 (new_feature) master@{5}: merge new_feature: Fast-forward
1ea9636 master@{6}: commit: index.txt is committed
bf567b8 (dev) master@{7}: merge dev: Fast-forward
eb8a4a5 master@{8}: commit (initial): initial commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git reflog show master@{1.week.ago}
e2796cc (HEAD -> master) master@{Tue Mar 5 12:16:55 2024 +0530}: merge
feature: Merge made by the 'ort' strategy.
a17b5c6 master@{Tue Mar 5 12:15:16 2024 +0530}: commit: changes done
in hello.txt (master)
bb6fab0 master@{Tue Mar 5 12:12:06 2024 +0530}: commit: hello.txt
added
e48c9a3 master@{Tue Mar 5 11:42:52 2024 +0530}: merge dummy_branch:
Merge made by the 'ort' strategy.
4931eba master@{Tue Mar 5 11:41:40 2024 +0530}: commit: changes done
to index file
6ba3361 (new_feature) master@{Tue Mar 5 11:40:23 2024 +0530}: merge
new_feature: Fast-forward
1ea9636 master@{Tue Mar 5 11:20:08 2024 +0530}: commit: index.txt is
committed
bf567b8 (dev) master@{Tue Mar 5 10:55:24 2024 +0530}: merge dev: Fast-
forward
eb8a4a5 master@{Tue Mar 5 10:43:44 2024 +0530}: commit (initial):
initial commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git diff master master@{1.week.ago}
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git reflog show master@{3.week.ago}
warning: log for 'master' only goes back to Tue, 5 Mar 2024 10:43:44
+0530
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git reflog show master@{2.week.ago}
```

warning: log for 'master' only goes back to Tue, 5 Mar 2024 10:43:44 +0530

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git reflog show master@{yesterday}
e2796cc (HEAD -> master) master@{Tue Mar 5 12:16:55 2024 +0530}: merge
feature: Merge made by the 'ort' strategy.
a17b5c6 master@{Tue Mar 5 12:15:16 2024 +0530}: commit: changes done
in hello.txt (master)
bb6fab0 master@{Tue Mar 5 12:12:06 2024 +0530}: commit: hello.txt
added
e48c9a3 master@{Tue Mar 5 11:42:52 2024 +0530}: merge dummy_branch:
Merge made by the 'ort' strategy.
4931eba master@{Tue Mar 5 11:41:40 2024 +0530}: commit: changes done
to index file
6ba3361 (new_feature) master@{Tue Mar 5 11:40:23 2024 +0530}: merge
new_feature: Fast-forward
1ea9636 master@{Tue Mar 5 11:20:08 2024 +0530}: commit: index.txt is
committed
bf567b8 (dev) master@{Tue Mar 5 10:55:24 2024 +0530}: merge dev: Fast-
forward
eb8a4a5 master@{Tue Mar 5 10:43:44 2024 +0530}: commit (initial):
initial commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git reflog show master@{6.day.ago}
e2796cc (HEAD -> master) master@{Tue Mar 5 12:16:55 2024 +0530}: merge
feature: Merge made by the 'ort' strategy.
a17b5c6 master@{Tue Mar 5 12:15:16 2024 +0530}: commit: changes done
in hello.txt (master)
bb6fab0 master@{Tue Mar 5 12:12:06 2024 +0530}: commit: hello.txt
added
e48c9a3 master@{Tue Mar 5 11:42:52 2024 +0530}: merge dummy_branch:
Merge made by the 'ort' strategy.
4931eba master@{Tue Mar 5 11:41:40 2024 +0530}: commit: changes done
to index file
6ba3361 (new_feature) master@{Tue Mar 5 11:40:23 2024 +0530}: merge
new_feature: Fast-forward
1ea9636 master@{Tue Mar 5 11:20:08 2024 +0530}: commit: index.txt is
committed
bf567b8 (dev) master@{Tue Mar 5 10:55:24 2024 +0530}: merge dev: Fast-
forward
eb8a4a5 master@{Tue Mar 5 10:43:44 2024 +0530}: commit (initial):
initial commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git_branching (master)
$ git diff master master@{2}
diff --git a/hello.txt b/hello.txt
index 410ca14..ce01362 100644
--- a/hello.txt
+++ b/hello.txt
@@ -1,2 +1 @@
 hello
 -bye
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (login)
$ vi index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (login)
$ git add .
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (login)
$ git commit -m "hello in index"
[login e1e01d0] hello in index
1 file changed, 2 insertions(+)
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (login)
$ cat index.txt
this is first commit
this is 2nd commit
this is 3rd commit
hello
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (login)
$ git log --oneline
e1e01d0 (HEAD -> login) hello in index
5a04b45 (org/master, master) added bye to new
1d7c8ce added line to new
ccb6107 (tag: v1.1.0, origin/master) new feature
65ae680 (tag: v1.0.1) second commit
aec5ff0 (tag: v1.0.0) initial commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (login)
$ git reset 5a04b45
Unstaged changes after reset:
M      index.txt
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (login)
$ git log --oneline
5a04b45 (HEAD -> login, org/master, master) added bye to new
1d7c8ce added line to new
ccb6107 (tag: v1.1.0, origin/master) new feature
65ae680 (tag: v1.0.1) second commit
aec5ff0 (tag: v1.0.0) initial commit
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (login)
$ cat index.txt
this is first commit
this is 2nd commit
this is 3rd commit
hello
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (login)
$ git reset 5a04b45 --hard
HEAD is now at 5a04b45 added bye to new
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (login)
$ cat index.txt
this is first commit
this is 2nd commit
```



```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (login)
$ git reflog show login
5a04b45 (HEAD -> login, org/master, master) login@{0}: reset: moving
to 5a04b45
e1e01d0 login@{1}: commit: hello in index
5a04b45 (HEAD -> login, org/master, master) login@{2}: reset: moving
to 5a04b45
9d5bd4f login@{3}: commit: third line added to index
5a04b45 (HEAD -> login, org/master, master) login@{4}: branch: Created
from HEAD
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (login)
$ git reset login@{1} --hard
HEAD is now at e1e01d0 hello in index
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (login)
$ git log --oneline
e1e01d0 (HEAD -> login) hello in index
5a04b45 (org/master, master) added bye to new
1d7c8ce added line to new
ccb6107 (tag: v1.1.0, origin/master) new feature
65ae680 (tag: v1.0.1) second commit
aec5ff0 (tag: v1.0.0) initial commit
```

//Git alias

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git config --list --show-origin
file:C:/Users/vlab/AppData/Local/Programs/Git/etc/gitconfig
diff.astextplain.textconv=astextplain
file:C:/Users/vlab/AppData/Local/Programs/Git/etc/gitconfig
filter.lfs.clean=git-lfs clean -- %f
file:C:/Users/vlab/AppData/Local/Programs/Git/etc/gitconfig
filter.lfs.smudge=git-lfs smudge -- %f
file:C:/Users/vlab/AppData/Local/Programs/Git/etc/gitconfig
filter.lfs.process=git-lfs filter-process
file:C:/Users/vlab/AppData/Local/Programs/Git/etc/gitconfig
filter.lfs.required=true
file:C:/Users/vlab/AppData/Local/Programs/Git/etc/gitconfig
http.sslbackend=openssl
file:C:/Users/vlab/AppData/Local/Programs/Git/etc/gitconfig
http.sslcainfo=C:/Users/vlab/AppData/Local/Programs/Git/mingw64/etc/ss
l/certs/ca-bundle.crt
file:C:/Users/vlab/AppData/Local/Programs/Git/etc/gitconfig
core.autocrlf=true
file:C:/Users/vlab/AppData/Local/Programs/Git/etc/gitconfig
core.fscache=true
file:C:/Users/vlab/AppData/Local/Programs/Git/etc/gitconfig
core.symlinks=false
file:C:/Users/vlab/AppData/Local/Programs/Git/etc/gitconfig
pull.rebase=false
```

```
file:C:/Users/vlab/AppData/Local/Programs/Git/etc/gitconfig
credential.helper=manager
file:C:/Users/vlab/AppData/Local/Programs/Git/etc/gitconfig
credential.https://dev.azure.com.usehttppath=true
file:C:/Users/vlab/AppData/Local/Programs/Git/etc/gitconfig
init.defaultbranch=master
file:C:/Users/vlab/.gitconfig      core.editor=notepad
file:C:/Users/vlab/.gitconfig      core.autocrlf=true
file:C:/Users/vlab/.gitconfig      user.name=lochani
file:C:/Users/vlab/.gitconfig      user.email=lochu5vilehya@gmail.com
file:C:/Users/vlab/.gitconfig      credential.usehttppath=true
file:./git/config                  core.repositoryformatversion=0
file:./git/config                  core.filemode=false
file:./git/config                  core.bare=false
file:./git/config                  core.logallrefupdates=true
file:./git/config                  core.symlinks=false
file:./git/config                  core.ignorecase=true
file:./git/config                  remote.origin.url=https://github.com/lochu-
55/Git_tag.git
file:./git/config
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
file:./git/config                  branch.master.remote=origin
file:./git/config                  branch.master.merge=refs/heads/master
file:./git/config                  remote.org.url=https://github.com/lochu-
55/Git_tag.git
file:./git/config
remote.org.fetch=+refs/heads/*:refs/remotes/org/*
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git config --list --show-origin --global
file:C:/Users/vlab/.gitconfig      core.editor=notepad
file:C:/Users/vlab/.gitconfig      core.autocrlf=true
file:C:/Users/vlab/.gitconfig      user.name=lochani
file:C:/Users/vlab/.gitconfig      user.email=lochu5vilehya@gmail.com
file:C:/Users/vlab/.gitconfig      credential.usehttppath=true
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)
```

nothing to commit, working tree clean

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git s
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)
```

nothing to commit, working tree clean

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git config --global alias.l "log"
```

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git l
```



```
commit 5a04b45d7f1900c00a50e3ab5676f60192f151da (HEAD -> master,  
org/master)  
Author: lochani <lochu5vilehya@gmail.com>  
Date: Thu Mar 14 15:50:41 2024 +0530
```

added bye to new

```
commit 1d7c8ce4fd5035c9048928276ddb2d434973289f  
Author: lochani <lochu5vilehya@gmail.com>  
Date: Thu Mar 14 15:48:11 2024 +0530
```

added line to new

```
commit ccb6107f094306ad7961cec76c77dcd809467429 (tag: v1.1.0,  
origin/master)  
Author: lochani <lochu5vilehya@gmail.com>  
Date: Thu Mar 14 14:52:12 2024 +0530
```

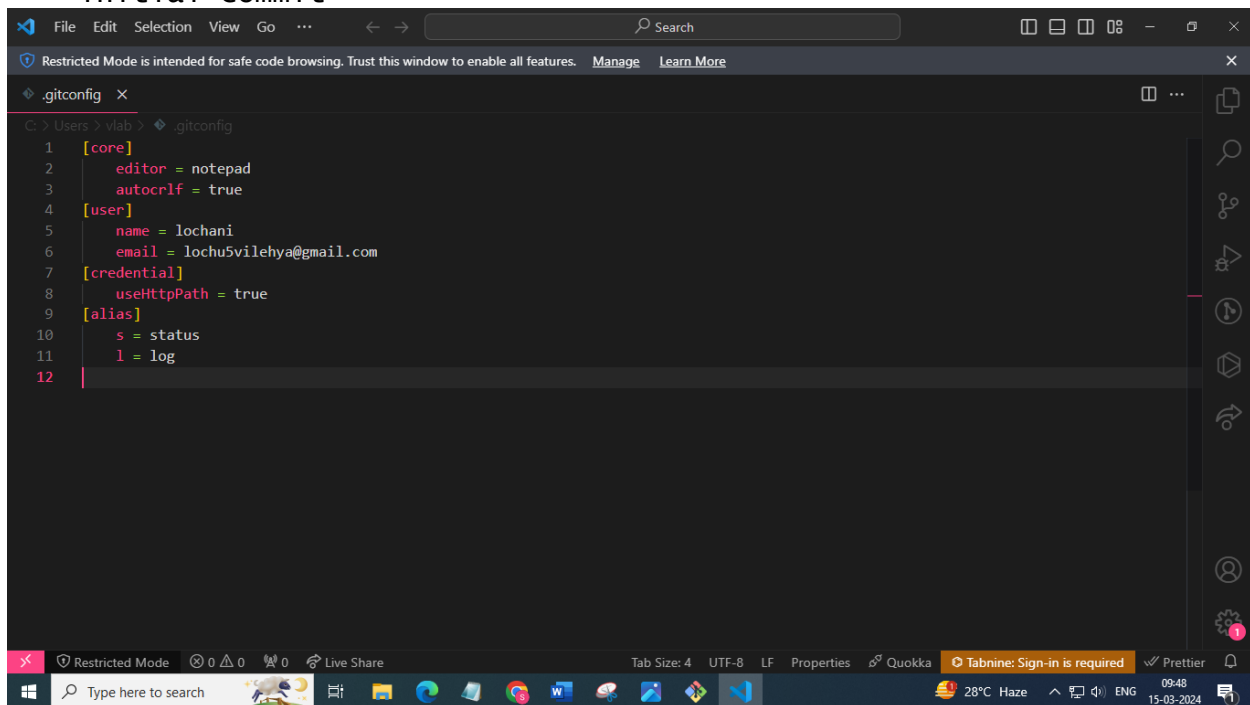
new feature

```
commit 65ae680011b2f65768a76118ec230f1badb4bc9d (tag: v1.0.1)  
Author: lochani <lochu5vilehya@gmail.com>  
Date: Thu Mar 14 14:08:24 2024 +0530
```

second commit

```
commit aec5ff0ce22808d1d5415730a42f5c8d29b10bb8 (tag: v1.0.0)  
Author: lochani <lochu5vilehya@gmail.com>  
Date: Thu Mar 14 14:07:05 2024 +0530
```

initial commit

A screenshot of a code editor window displaying a .gitconfig file. The editor has a dark theme and a sidebar on the right with various icons. The .gitconfig file content is as follows:

```
1 [core]
2     editor = notepad
3     autocrlf = true
4 [user]
5     name = lochani
6     email = lochu5vilehya@gmail.com
7 [credential]
8     useHttpPath = true
9 [alias]
10    s = status
11    l = log
12
```

The editor's status bar at the bottom shows "Restricted Mode", "0 0 0", "Live Share", "Tab Size: 4", "UTF-8", "LF", "Properties", "Quokka", "Tabnine: Sign-in is required", and "Prettier". The Windows taskbar is visible at the very bottom with the search bar and several application icons.

The `git archive` command is used to create a tar or zip archive of a specified Git tree. This can be useful for creating snapshots of a project at a specific commit or branch without including the entire Git history.

Here's the basic syntax for `git archive`:

Where:

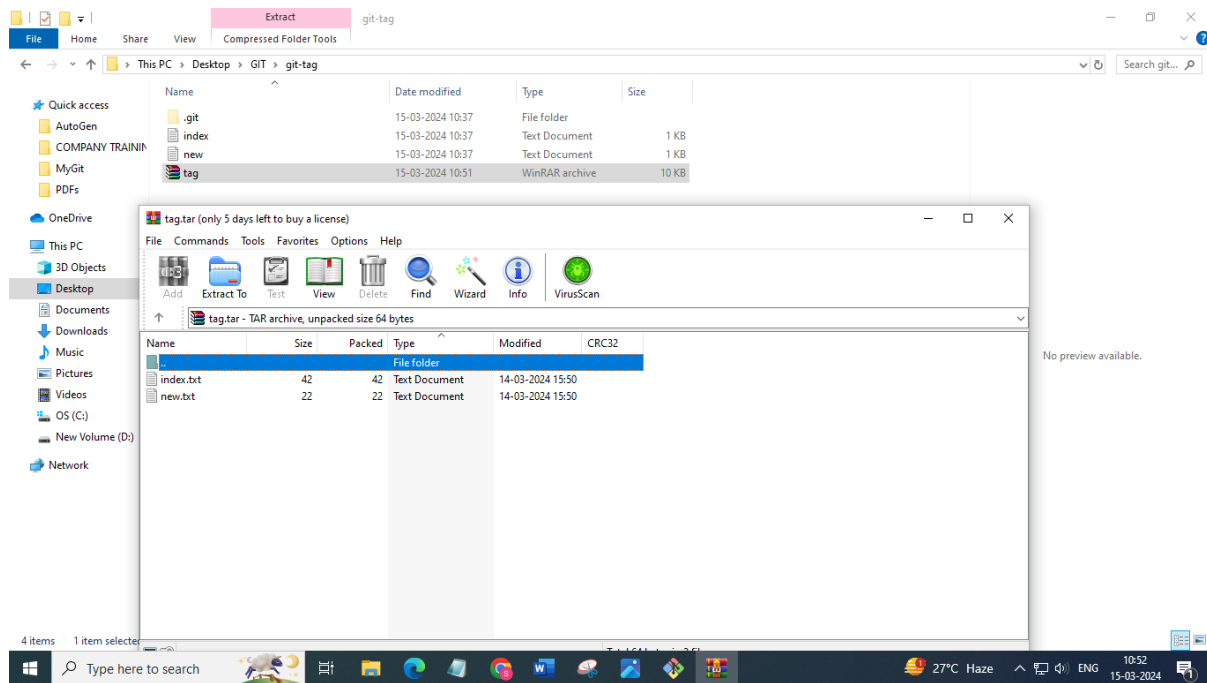
- `<format>` specifies the format of the archive (e.g., tar, zip).
- `<file>` specifies the name of the output archive file.
- `<branch/commit>` specifies the branch or commit to create the archive from.

For example, to create a tar archive of the `main` branch and save it as `project.tar`, you would run:

Similarly, you can create a zip archive by specifying `--format=zip`.

```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git archive --format=tar --output=tag.tar master

vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ ls
index.txt  new.txt  tag.tar
```



```
vlab@VEDA2F107 MINGW64 ~/Desktop/GIT/git-tag (master)
$ git ls-remote -h --
From https://github.com/lochu-55/Git_tag.git
de77fae13e85881c7bd127a3556166262b8a3ff8      refs/heads/main
5a04b45d7f1900c00a50e3ab5676f60192f151da      refs/heads/master
```