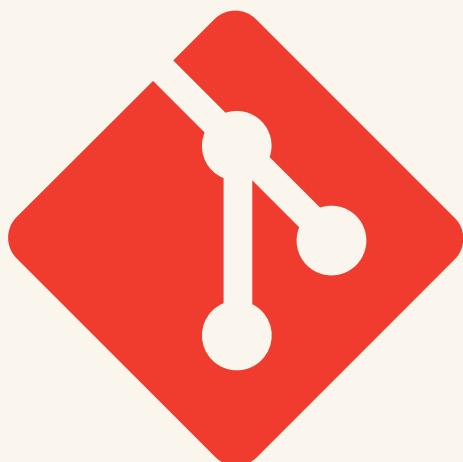


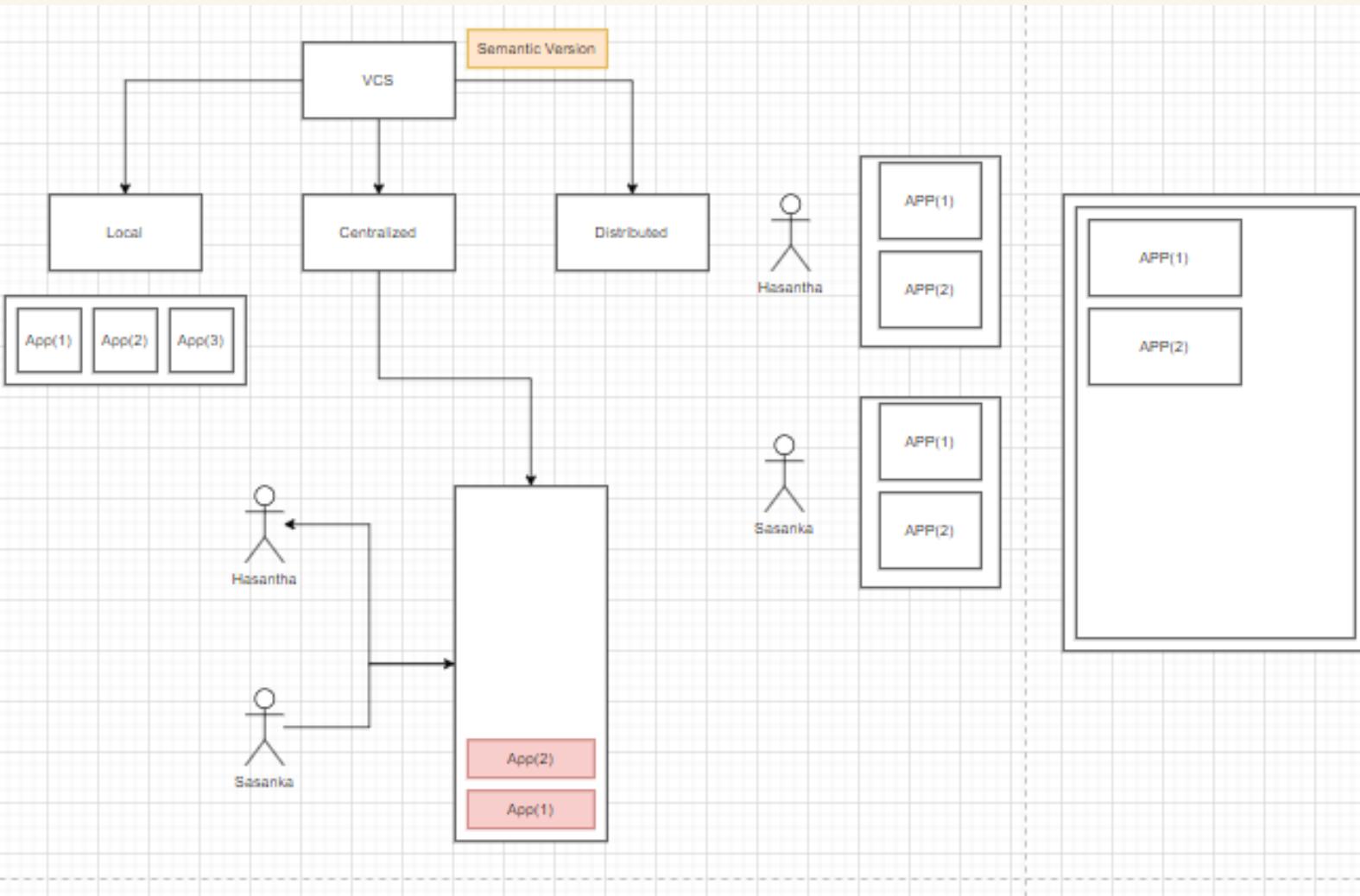
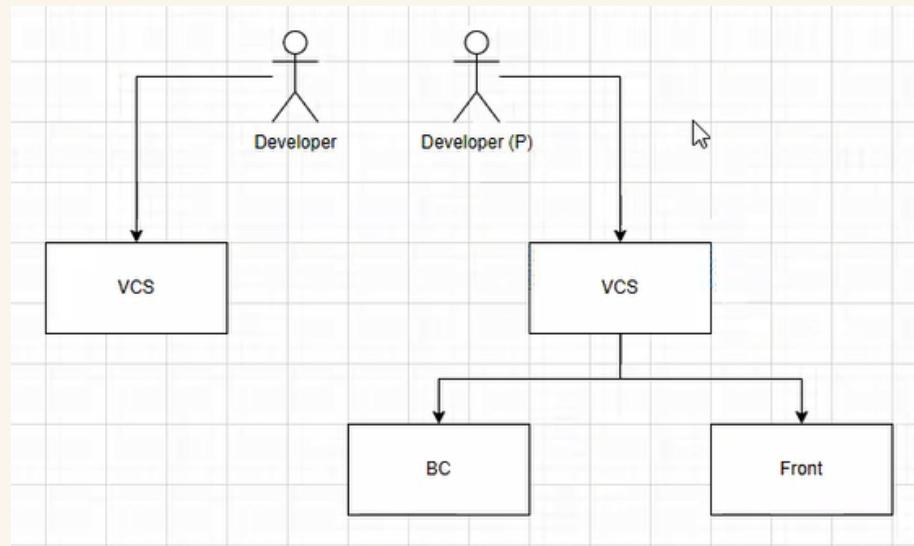
Git A - Z

Version Controller System



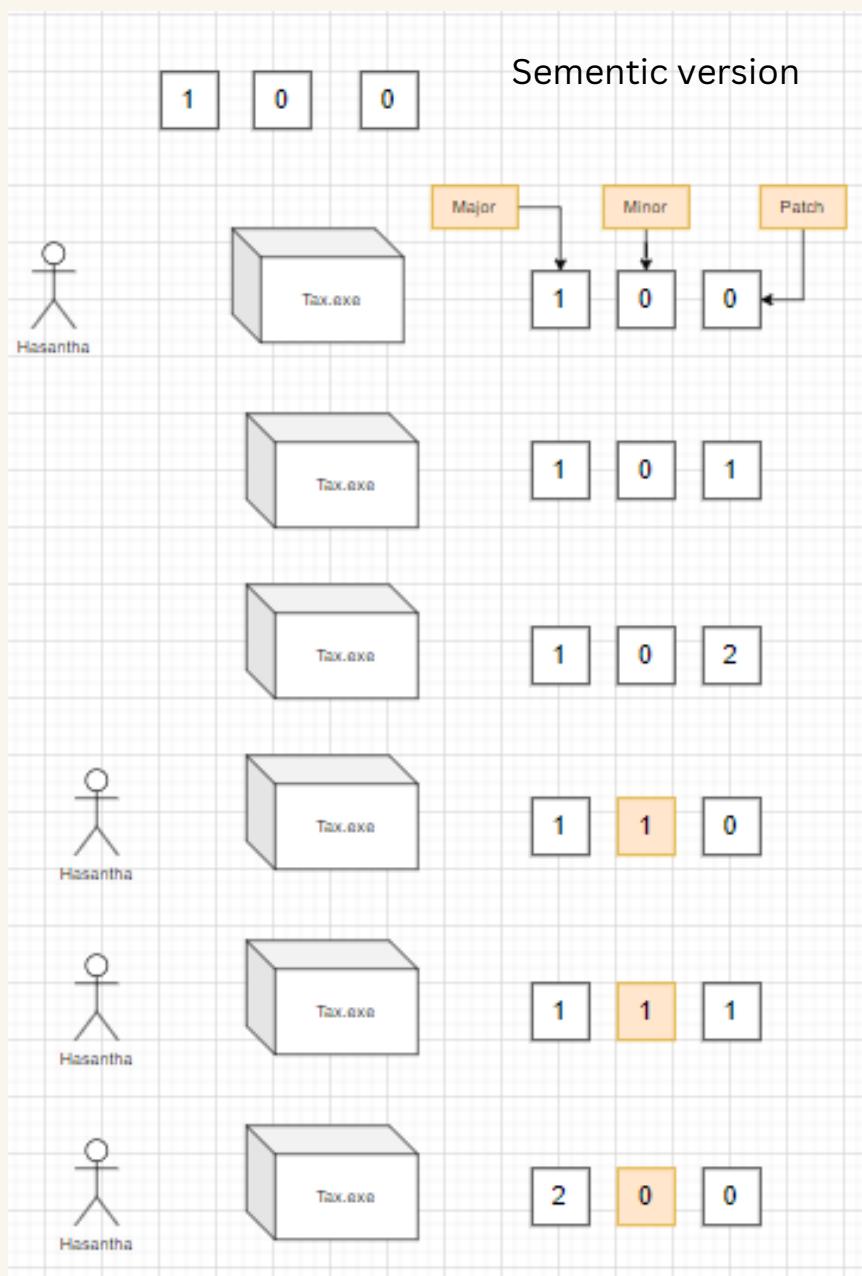
git

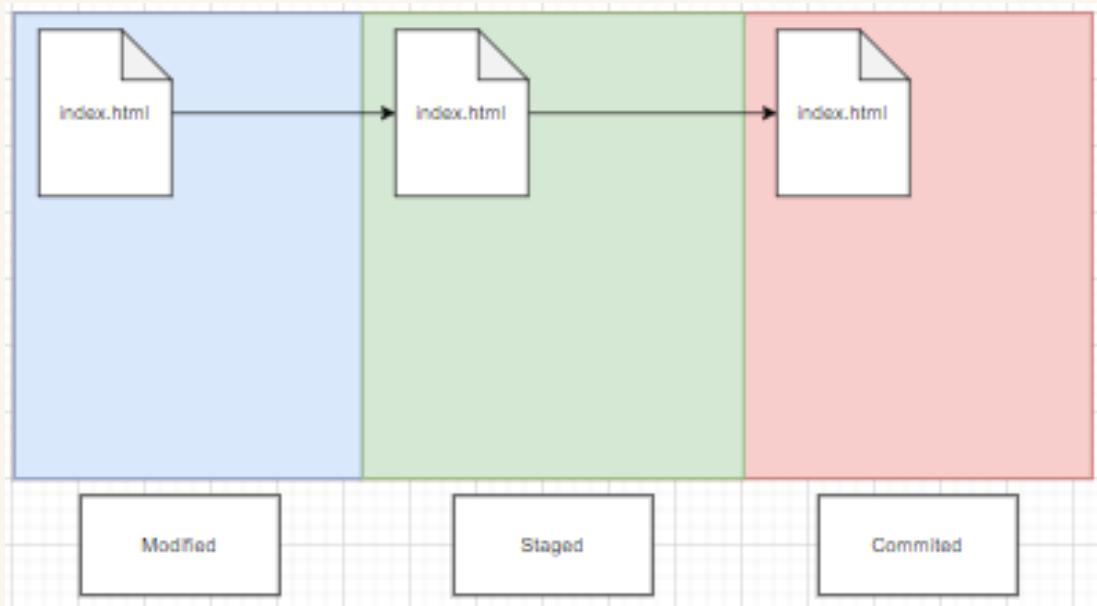
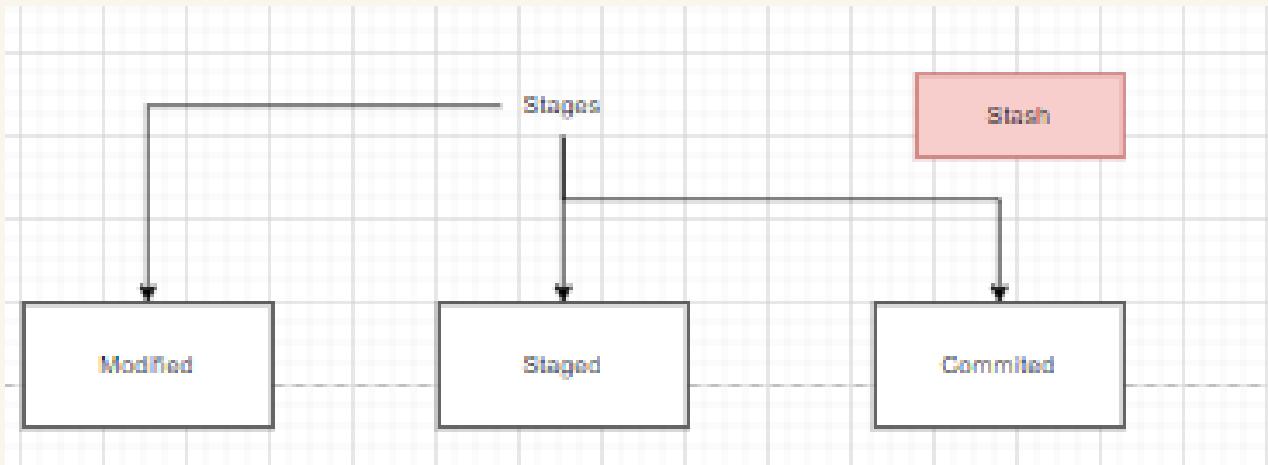
Prasad Karunananayaka



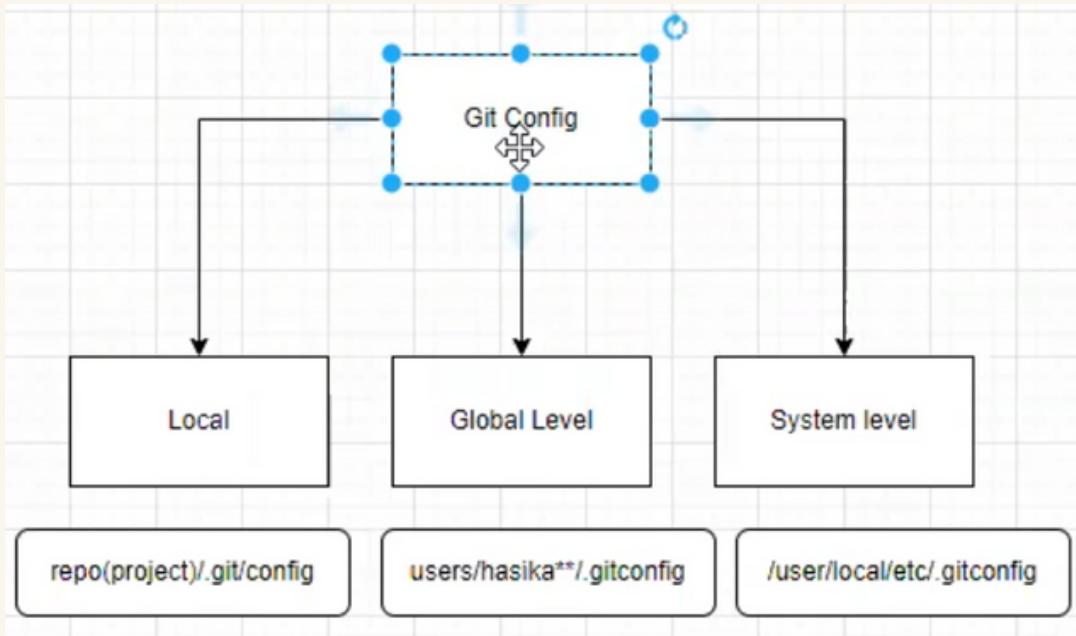
In a CVCS, there is a **central** repository that holds the latest version of the codebase. Developers check out code from this central repository, make changes, and then check their changes back in. The central repository is the authoritative source for the codebase, and all changes are made directly to it.

In contrast, a **DVCS** allows each developer to have their own copy of the codebase, complete with its full history. Developers make changes to their local copy of the code, and then push those changes to a central repository or pull changes from other developers' repositories. The central repository in a DVCS is just one of many copies of the codebase.





Git configuration allows you to customize your Git environment by setting various parameters such as your name, email address, default editor, and more.



```
$ git config --list --show-origin
```

The git config --list --show-origin command shows the Git configuration values along with the file they were read from.

\$ git init

↓ file become modified
stage

The git init command is used to create a new Git repository. When you run git init in a directory, it creates a new subdirectory named .git that contains all the necessary files for a Git repository.

```
$ ls
```

The ls command is used to list the files and directories in the current working directory. When you run ls in a terminal or command prompt, it will show you a list of all the files and directories in the current directory.

```
$ git config user.email
```

The git config user.email command is used to display the email address associated with the user who made the commits in the Git repository.

```
$ git config --local user.email  
prasadkarunanayaka2016@gmail.com
```

The git config --local user.email command is used to set the email address associated with the user who made the commits in the Git repository at the local repository level.

```
$ git config user.email  
prasadkarunanayaka2016@gmail.com
```

```
$ git config --local user.name prasad  
Prasa@DESKTOP-C27UDNE MINGW64 /e/developer  
$ git config user.name  
prasad
```

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

The git config --local --remove-section user command is used to remove the [user] section from the local Git repository configuration file.

\$ git status

The git status command is used to show the current state of the Git repository. When you run this command in the terminal or command prompt, it will show you information about the branch you are currently on, any changes that have been made to files in the repository, and any changes that have been staged or are waiting to be committed.

\$git help init

The git help init command is used to display the manual page for the git init command, which is used to create a new Git repository.

\$ git add index.html

↓ file become
stage phase

The git add command is used to add changes made to a file or files to the

Git staging area, which is the first step in committing changes to a Git repository. In your example, you are using the git add command to add the changes made to the index.html file to the staging area.

```
$ git add index.html

Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/New f
r)
$ git status
On branch master

No commits yet

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

\$git commit m-"index.html"

↓ Start
commit
Stage

before this stage please
add name and email

```
r)
$ git commit -m"add index.html"
[master (root-commit) 8a6faa3] add index.html
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 index.html
```

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/New f
r)
$ git status
On branch master
nothing to commit, working tree clean
```

```
$ git add .
```

The git add . command is used to add all changes made in the current directory and its subdirectories to the Git staging area. The . (dot) represents the current directory, so running git add . will add **all files** and changes made in the current directory and its subdirectories to the staging area.

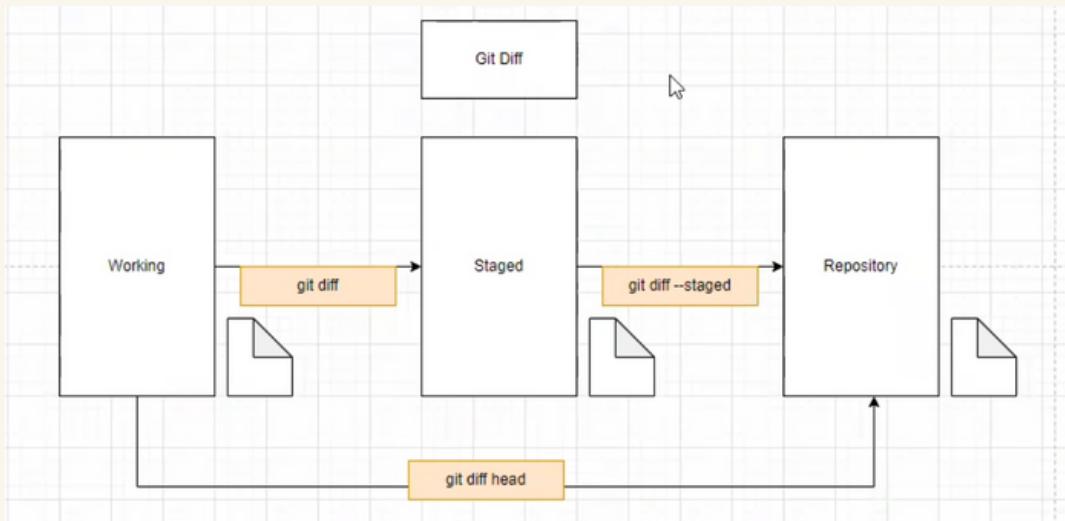
```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/New fo1
$ git commit -m"add all files"
[master 36f2afe] add all files
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 main.js
 create mode 100644 style.css

Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/New fo1
$ git status
On branch master
nothing to commit, working tree clean
```

```
Hasika@DESKTOP-D4N5O1
$ git diff

Hasika@DESKTOP-D4N5O1
$ git diff --staged

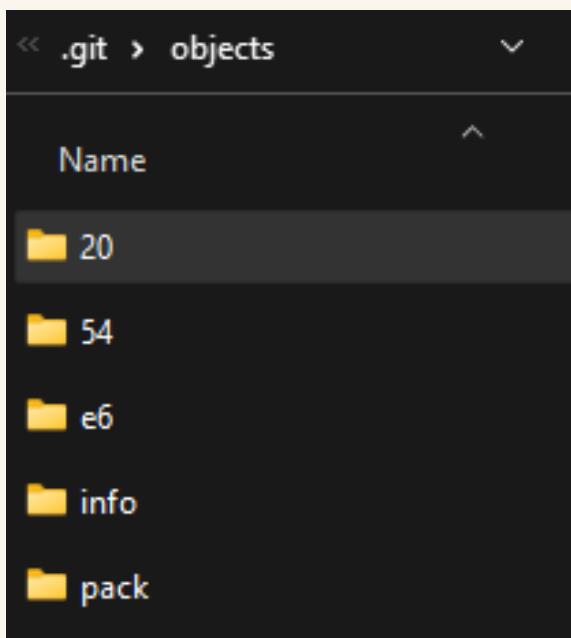
Hasika@DESKTOP-D4N5O1
$ git diff head
```



The `git diff` command is used to show the differences between two versions of a file, or between two commits in a Git repository. When you run the `git diff` command, Git will show you the differences between the files in the working directory (i.e., the current state of the files on your local machine) and the files in the Git repository.

```
$ git log  
commit 208299eedf2bb4bd91e8f8a2ffa8308fcc401811 (HEAD -> master)  
Author: prasad <prasadkarunanayaka2016@gmail.com>  
Date:   Sat May 13 23:41:42 2023 +0530  
  
    first commit  
  
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 38 Git (master)  
$
```

every commit has a unique hash code



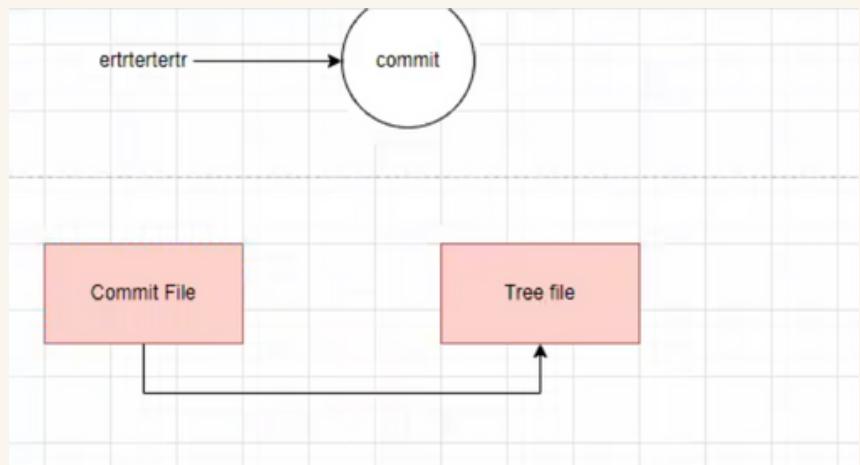
Commit file
Start number
(hash²⁰ code)

```
$ git cat-file 208299eedf2bb4bd91e8f8a2ffa8308fcc401811 -p
```

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 38 Git (master)  
$ git cat-file 208299eedf2bb4bd91e8f8a2ffa8308fcc401811 -p  
tree 54754f5610638d8485c1b601177b26c5e90e7697  
author prasad <prasadkarunanayaka2016@gmail.com> 1684001502 +0530  
committer prasad <prasadkarunanayaka2016@gmail.com> 1684001502 +0530
```

first commit

↑
tree file



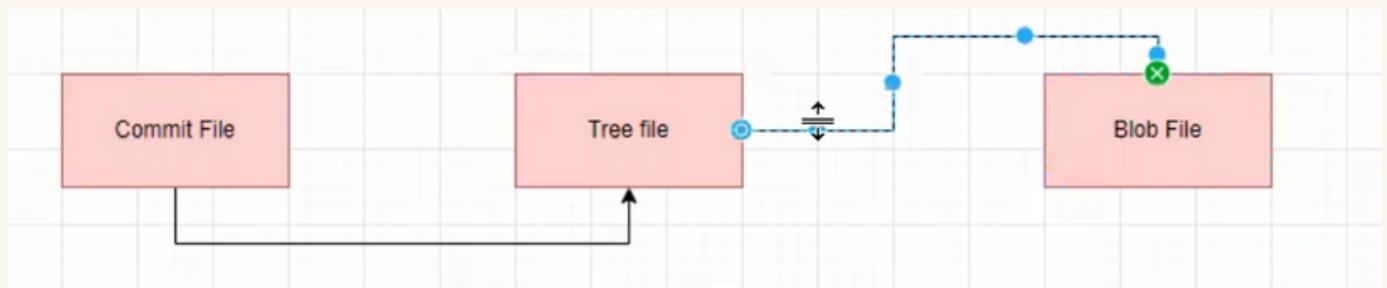
The git cat-file command is a Git command used to view the contents of a Git object. This command can be used to display the contents of a specific Git object, such as a commit, tag, or blob, and can be helpful in troubleshooting Git repositories or examining the history of a project.

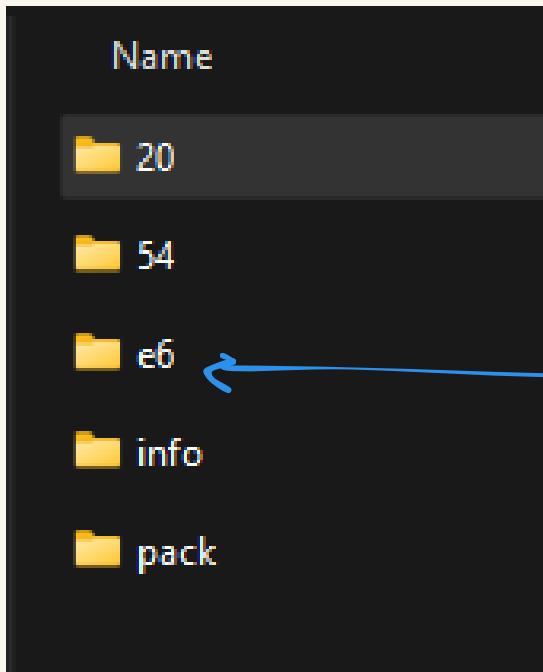
```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 38 Git (master)
$ git cat-file 54754f5610638d8485c1b601177b26c5e90e7697 -p
100644 blob e69de29bb2d1d6434b8b29ae775ad8c2e48c5391 Example.txt
```

tree file (start 54)

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 38 Git (master)
$ git cat-file 54754f5610638d8485c1b601177b26c5e90e7697 -p
100644 blob e69de29bb2d1d6434b8b29ae775ad8c2e48c5391 Example.txt
```

You will find blob file





blob file

Look output here

```
$ git cat-file 17dc9f6092eeccbc2d732244a1eee04f410d37cf -p  
sample code
```

Renamed file

```
$ mv Example.txt newFile.txt
```

we can change
obj type .html

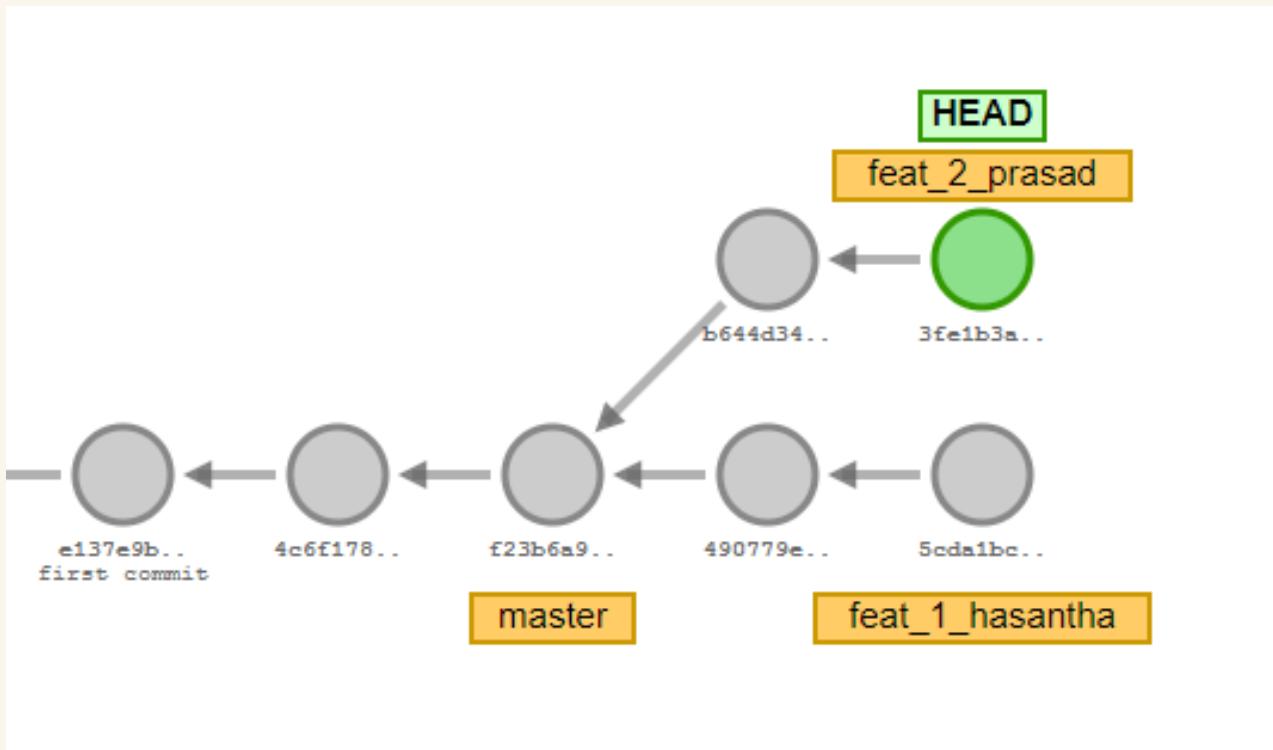
The command "mv Example.txt newFile.txt" would rename the file "Example.txt" to "newFile.txt". The contents of the file would remain the same, but it would now be accessible using the new name.

```
$ mv Example.txt newFile.txt  
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 38 Git (master)  
$ git status  
On branch master  
Changes not staged for commit:  
  (use "git add/rm <file>..." to update what will be committed)  
    (use "git restore <file>..." to discard changes in working directory)  
      deleted:   Example.txt  
  
Untracked files:  
  (use "git add <file>..." to include in what will be committed)  
    newFile.txt  
  
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git add .  
  
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 3  
$ git status  
On branch master  
Changes to be committed:  
  (use "git restore --staged <file>..." to unstage)  
    renamed:   Example.txt -> newFile.txt
```

```
$ git commit -m"renamed File"  
[master d24cbcc] renamed File  
 1 file changed, 0 insertions(+), 0 deletions(-)  
 rename Example.txt => newFile.txt (100%)  
  
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 3  
$ git status  
On branch master  
nothing to commit, working tree clean
```

Branch



checkout →
mean change head

```
$ git commit  
$ git commit  
$ git checkout -b  
feat_1_hasantha  
$ git commit  
$ git commit  
$ git checkout master  
I don't understand that.  
$ git checkout master  
$ git checkout -b  
feat_2_prasad  
$ git commit  
$ commit  
I don't understand that.  
$ git commit
```

```
$ git commit -m "initialize commit"
[master (root-commit) b1e384b] initialize commit
 1 file changed, 1 insertion(+)
 create mode 100644 Example.txt

Hasika@DESKTOP-D4N5010 MINGW64 /d/bootcamp/full-stack
$ git branch
* master

Hasika@DESKTOP-D4N5010 MINGW64 /d/bootcamp/full-stack
$ git log
commit b1e384bf3560a3626cd03ede441f5dbfaf15601a (HEAD
Author: ozen-hub <info.ozenhub@gmail.com>
Date:   Mon Apr 10 20:19:54 2023 +0530

    initialize commit

Hasika@DESKTOP-D4N5010 MINGW64 /d/bootcamp/full-stack
$ git log --oneline
b1e384b (HEAD -> master) initialize commit
```

```
$ 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:   Example.txt

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

Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 38 Git (master)
$ git add .

Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 38 Git (master)
$ git commit -m "updated from the master branch"
[master dbdb19a] updated from the master branch
 1 file changed, 5 insertions(+), 1 deletion(-)

Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 38 Git (master)
$ git status
On branch master
nothing to commit, working tree clean
```

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/
$ git branch
  feat-1-hasantha
* master

Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/
$ git checkout feat-1-hasantha
Switched to branch 'feat-1-hasantha'

Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/
$ git branch
* feat-1-hasantha
  master
```

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 38
$ git checkout feat-1-hasantha
Switched to branch 'feat-1-hasantha'
M       Example.txt

Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 38
$ git add .

Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 38
$ git commit -m "hasantha updated the file"
[feat-1-hasantha b009dc4] hasantha updated the file
 1 file changed, 3 insertions(+)

Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 38
$ git checkout master
Switched to branch 'master'
```

you modified "feat-1-hasantha" but you checkout master after open the source file you see modify things in master branch.

How to remove branch

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git
$ git checkout -b feat-2-amila
Switched to a new branch 'feat-2-amila'
```

Write something in source file "modified feat-2-amila"

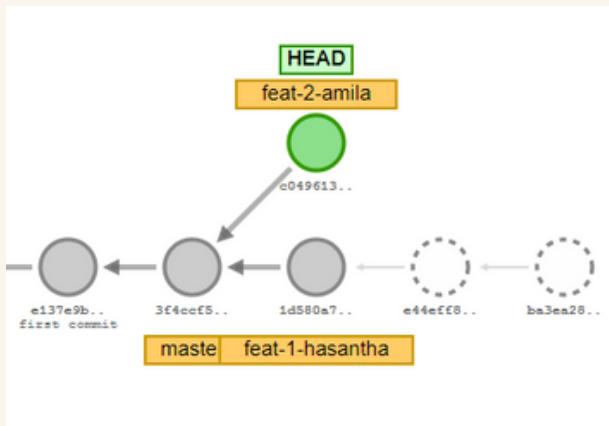
```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/D
$ git add .
```

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 38 Git (feat-2-amila)
$ git branch -d feat-2-amila
error: Cannot delete branch 'feat-2-amila' checked out at 'E:/developerstack/Git/Day
38 Git'
```

You cannot delete file in the file (branch)
So you have to move master branch
(Head)

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/D
$ git checkout master
Switched to branch 'master'
M           Example.txt
```

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/D
$ git branch -d feat-2-amila
Deleted branch feat-2-amila (was dbdb19a).
```



```
$ git commit
$ git checkout -b feat-1-
hasantha
$ git commit
$ git checkout master
$ git checkout feat-1-
hasantha
$ git checkout -b feat-2-
amila
$ comiit
I don't understand that.
$ gitcommit
I don't understand that.
$ git commit
$ commit
I don't understand that.
$ git omiit
I don't understand that.
$ git commit
$ git checkout master
$ git branch -d feat-2-amila
$ git checkout -b feat-2-
amila
$ git commit
```

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git,
$ git add .

Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git,
$ git commit -m "init"
1[feat-1-ranil b92fda3] init
 1 file changed, 1 insertion(+)

Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git,
$ git checkout master
Switched to branch 'master'

Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git,
$ git branch -d feat-1-ranil
error: The branch 'feat-1-ranil' is not fully merged.
If you are sure you want to delete it, run 'git bran
```



```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git,
$ git branch -D feat-1-ranil
Deleted branch feat-1-ranil (was b92fda3).
```

*if you want to delete so not fully merged
use -D*

The git branch -d feat-1-ranil command is used to delete the branch named feat-1-ranil. However, in this case, the branch deletion was not successful and returned an error message stating that the branch is not fully merged.

This means that there are some changes in the branch feat-1-ranil that have not been merged into the branch you are currently on. Git is designed to prevent accidental deletion of unmerged changes, so it throws an error message to warn you that deleting the branch would result in losing unmerged changes.

To delete the branch despite the unmerged changes, Git provides the -D flag as a more forceful option. When you ran git branch -D feat-1-ranil, Git deleted the branch feat-1-ranil without checking whether it was fully merged into the current branch.

The message "Deleted branch feat-1-ranil (was b92fda3)" indicates that the branch feat-1-ranil was successfully deleted and that it was pointing to the commit with the hash b92fda3.

Git supports different types of merge operations, depending on the branching and merging strategies used. Here are some of the most common types of merge in Git:

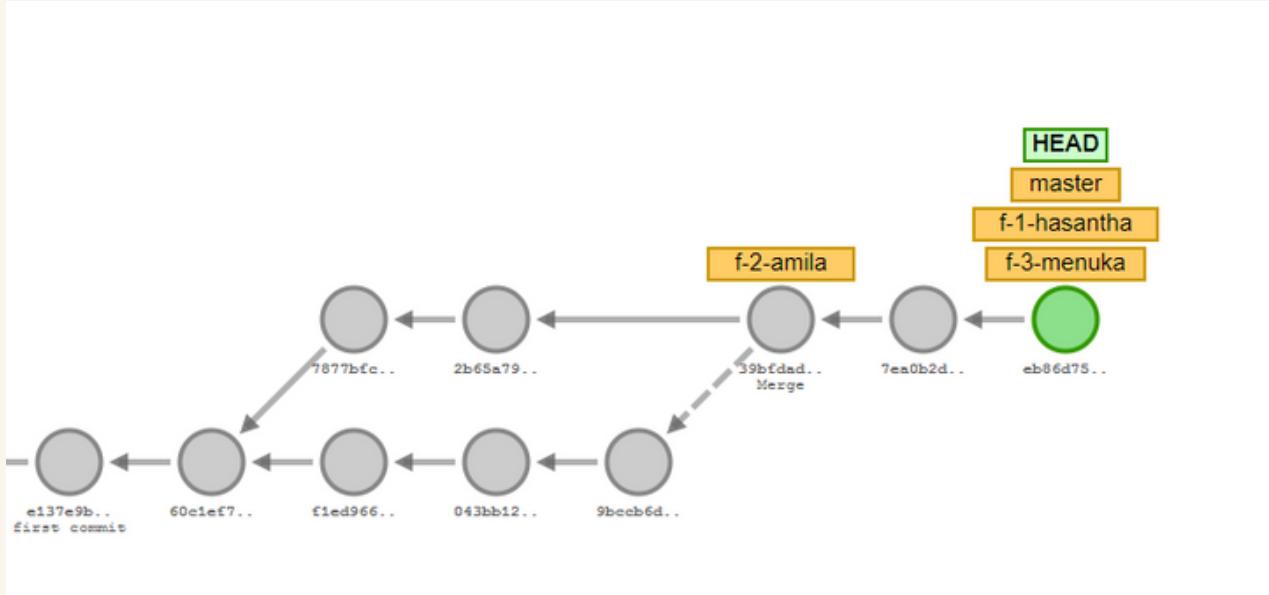
Fast-forward merge: This is the simplest and most common type of merge. It occurs when the branch you want to merge into has progressed since the creation of the branch you want to merge. In this case, Git simply moves the branch pointer forward to match the pointer of the branch being merged.

This results in a linear history without any additional commit or merge nodes.

Recursive merge: This type of merge occurs when the branch being merged contains commits that are not in the current branch, and the two branches have diverged. In this case, Git creates a new merge commit that combines the changes from both branches. The merge commit has two or more parent commits, and it creates a new merge node in the commit history.

Squash merge: This type of merge combines the changes from a feature branch into a single commit that is then applied to the main branch. It is useful when you want to keep the commit history clean and avoid merge commits cluttering the commit log. In a squash merge, Git combines all changes from the feature branch into a single commit that is applied on top of the main branch.

Octopus merge: This type of merge is used to combine three or more branches simultaneously. In an octopus merge, Git combines the changes from all branches into a single commit that is applied to the main branch. This is useful when you have multiple feature branches that depend on each other and need to be merged together.



```
$ git commit
$ gitcomiit
I don't understand that.

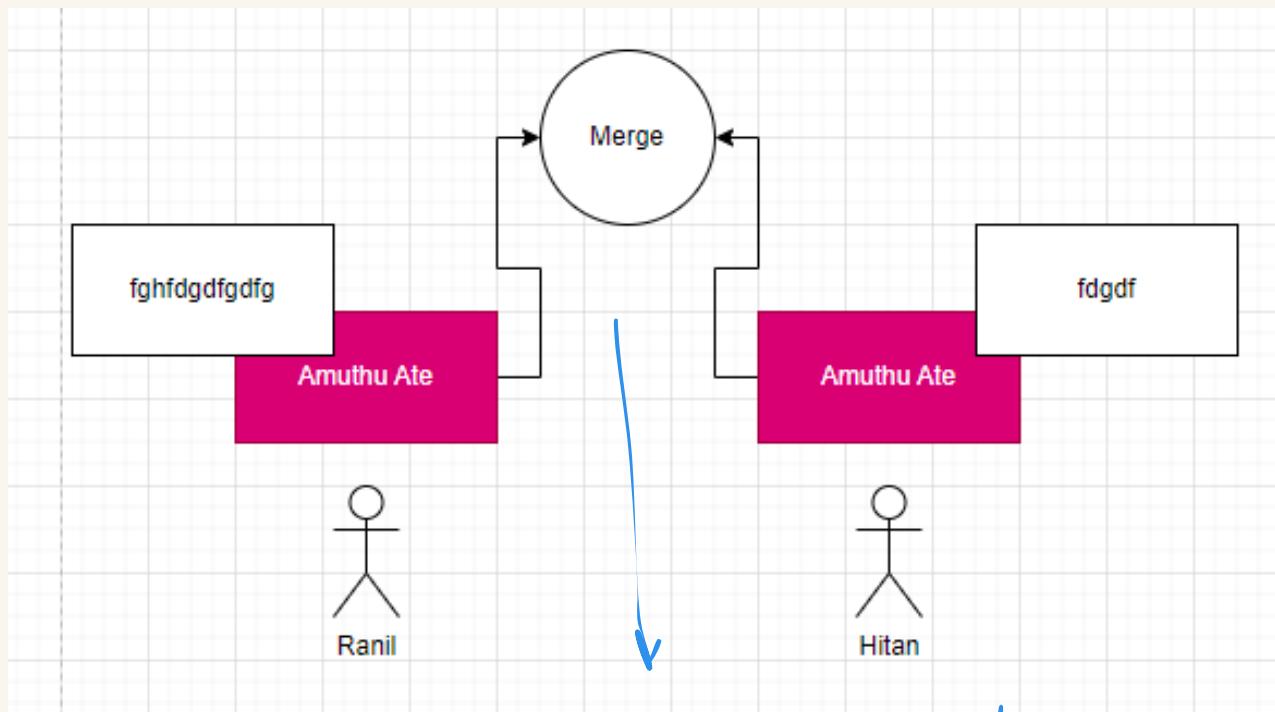
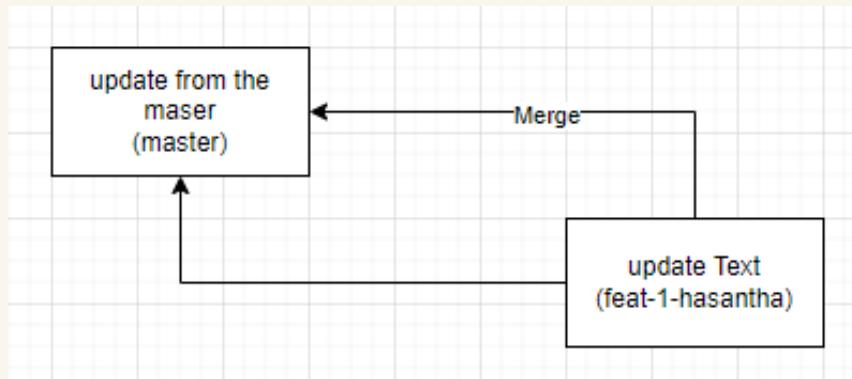
$ git checkout -g f-1-
hasantha
Cannot find commit: undefined

$ git checkout -b f-1-
hasantha
$ commit
I don't understand that.

$ git commit
$ git commit
$ git commit
$ git checkout master
$ git checkout -b f-2-amila
$ git commit
$ git commit
$ git merge f-1-hasantha
$ git checkout -b f-3-menuka
$ git commit
$ git commit
$ git checkout f-1-hasantha
$ git merge f-3-menuka
You have performed a fast-forward
merge.

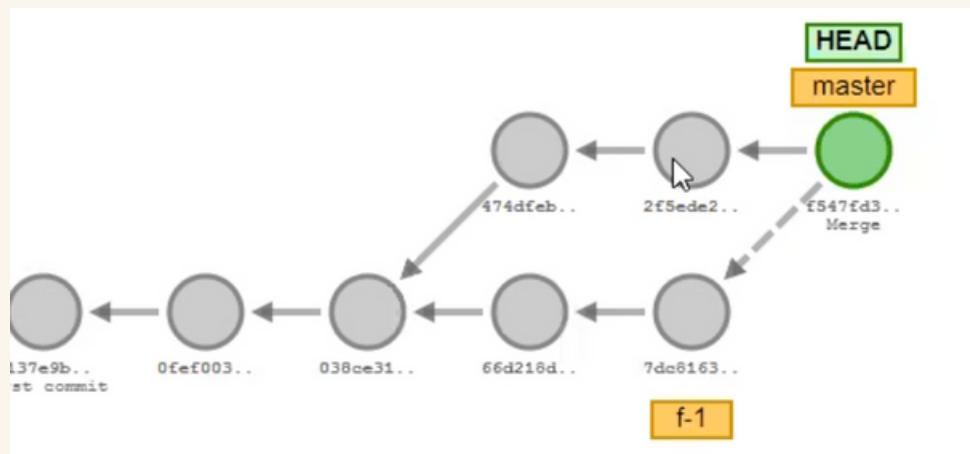
$ git checkout master
$ git merge f-3-menuka
You have performed a fast-forward
merge.
```

Conflict

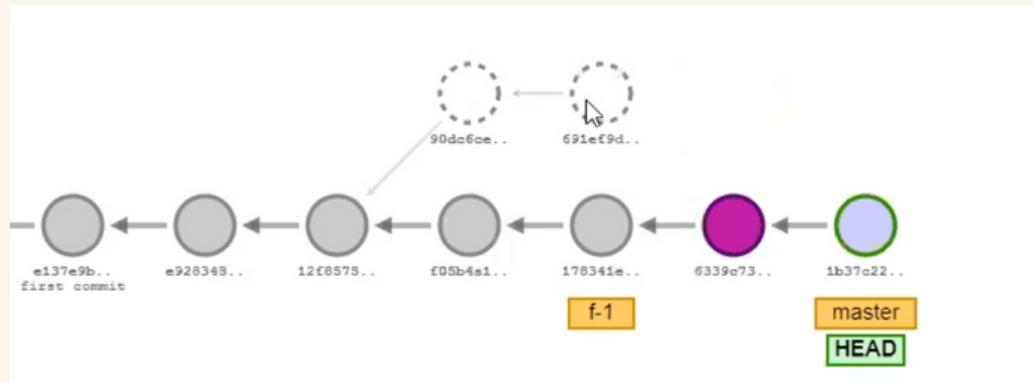


system cannot understand which is
got

Merge



Rebase



both same but Rebase not
Protect their previous code

How to update commit new commit

```
$ git add .  
Hasika@DESKTOP-D4N5010 MINGW64 /d/bootcamp/full-stack/3-batch/stage-1/day-39/  
app (master)  
$ git diff --chached  
error: invalid option: --chached  
  
Hasika@DESKTOP-D4N5010 MINGW64 /d/bootcamp/full-stack/3-batch/stage-1/day-39/  
app (master)  
$ git diff --cached  
diff --git a/index.html b/index.html  
index 5b24f92..5e4beee 100644  
--- a/index.html  
+++ b/index.html  
@@ -1 +1 @@  
-this is my completeted project! [10+10=15]  
\ No newline at end of file  
+this is my completeted project! [10+10=15] => new Updated Logic => [10+10=20]  
\ No newline at end of file
```

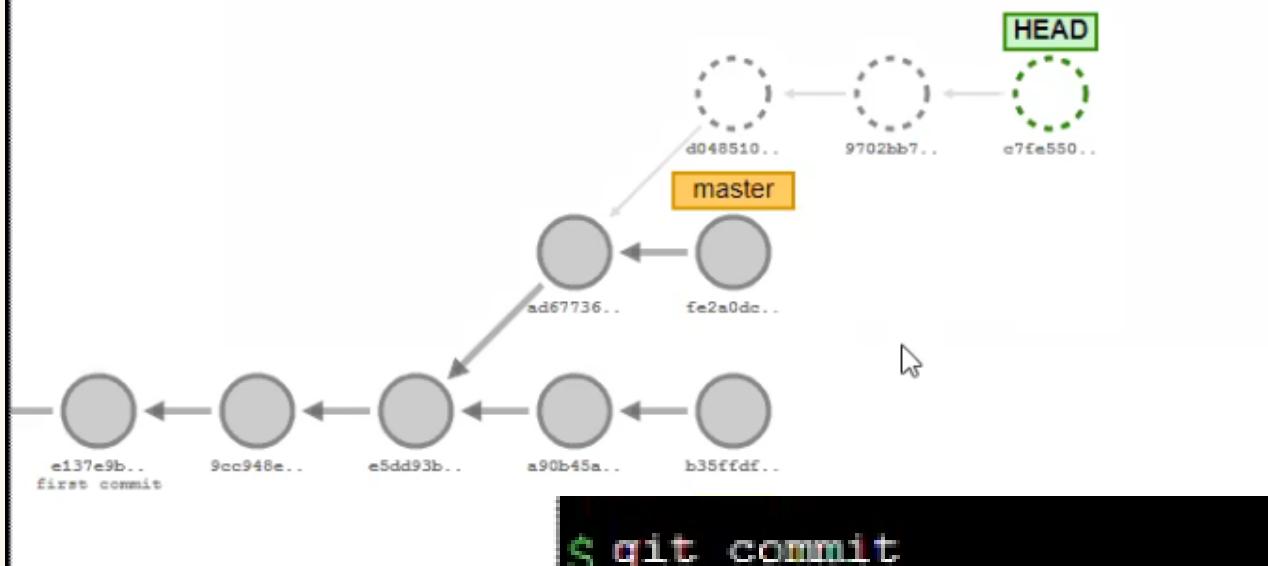
```
Hasika@DESKTOP-D4N5010 MINGW64 /d/bootcamp/full-stack/3-batch/st  
app (master)  
$ git commit --amend  
[master 5fd9c33] finalized the code  
Date: Thu Apr 13 20:57:18 2023 +0530  
1 file changed, 1 insertion(+)  
create mode 100644 index.html
```

```
$ git show 5fd9c33  
commit 5fd9c337fb9796f7edbffcad53128749daecffae (HEAD -> master)  
Author: ozen-hub <info.ozenhub@gmail.com>  
Date:   Thu Apr 13 20:57:18 2023 +0530  
  
        finalized the code  
  
diff --git a/index.html b/index.html  
new file mode 100644  
index 0000000..5e4beee  
--- /dev/null  
+++ b/index.html  
@@ -0,0 +1 @@           ||  
+this is my completeted project! [10+10=15] => new Updated Logic =>  
\ No newline at end of file
```

```
$ git cherry-pick ec9e6aa
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
error: could not apply ec9e6aa... new feature commit 1
last
hint: After resolving the conflicts, mark them with
hint: "git add/rm <pathspec>", then run
hint: "git cherry-pick --continue".
hint: You can instead skip this commit with "git cherr
y-pick --skip".
hint: To abort and get back to the state before "git c
herry-pick",
hint: run "git cherry-pick --abort".
```

Local Repository

HEAD: (detached head)



```
$ git commit
$ git commit
$ git checkout -b f-1
$ git commit
$ git commit
$ git commit
$ git checkout master
$ git commit
$ git commit
$ git checkout ad67736
$ git commit
$ git commit
$ git commit
```

Git Cloaning

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 39 Git/clone/step 1  
t-1-hasantha|REBASE 3/3)  
$ git clone https://github.com/gd4Ark/star-battle.git  
Cloning into 'star-battle'...  
remote: Enumerating objects: 307, done.  
remote: Counting objects: 100% (9/9), done.  
remote: Compressing objects: 100% (3/3), done.  
remote: Total 307 (delta 6), reused 6 (delta 6), pack-reused 298  
Receiving objects: 100% (307/307), 2.01 MiB | 4.02 MiB/s, done.  
Resolving deltas: 100% (88/88), done.
```

```
SKTOP-C27UDNE MINGW64 /e/developerstack/Git/Da  
ntha|REBASE 3/3)  
r-battle/  
/star-battle/: No such file or directory
```

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 39 Git/clo  
-battle (master)  
$ ls -a  
./ .git/ README-CN.md css/ index.html sound/  
../ LICENSE README.md img/ js/
```

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 39 Git/clone/step 1/st
-battle (master)
$ git log --oneline
566af57 (HEAD -> master, origin/master, origin/HEAD) docs: 去除失效图片
ceff758 docs: readme.md
273ee92 Merge pull request #2 from gd4Ark/fix/file-name
e19fc55 (origin/fix/file-name) fix: 文件命名
5b6a0ba feat(add license):
1fcf225 (origin/dev) update README.md
253b611 update README.md
92f27e1 add README-CN.md
cedb0f1 增加 footer
075fb9d update README.md
f69acf0 修正排行榜排序逻辑
8e9a53b 修正排行榜排序逻辑
1a3188a 修复重新开始后 P按键不监听的 BUG, 为数字增加格式化, 更新部分 README.md
a0ddcf7 更新界面, 包括中文说明等
46e663f update README.md
409ee55 更新 Canvas z-index
93027b2 update
472e33a delete game-controller
abbeb17 压缩图标资源
b331a03 删除 videos 文件 并且 修复游戏结束后仍然监听按键事件的 BUG
0cf331e update README.md
fb3d0a8 update README.md
06cd907 update util.js and rank scene
...skipping...
566af57 (HEAD -> master, origin/master, origin/HEAD) docs: 去除失效图片
ceff758 docs: readme.md
273ee92 Merge pull request #2 from gd4Ark/fix/file-name
e19fc55 (origin/fix/file-name) fix: 文件命名
5b6a0ba feat(add license):
1fcf225 (origin/dev) update README.md
253b611 update README.md
92f27e1 add README-CN.md
cedb0f1 增加 footer
075fb9d update README.md
f69acf0 修正排行榜排序逻辑
8e9a53b 修正排行榜排序逻辑
1a3188a 修复重新开始后 P按键不监听的 BUG, 为数字增加格式化, 更新部分 README.md
```

SSH cloning

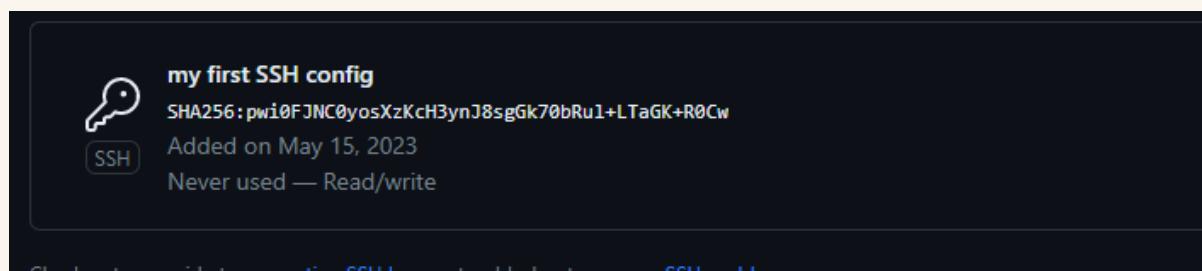
Generating a new SSH key

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 39 Git/clone/step-1-hasantha|REBASE 3/3)
$ ssh-keygen -t ed25519 -C "prasadkarunanayaka2016@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/Prasa/.ssh/id_ed25519): |
```

Adding your SSH key to the ssh-agent

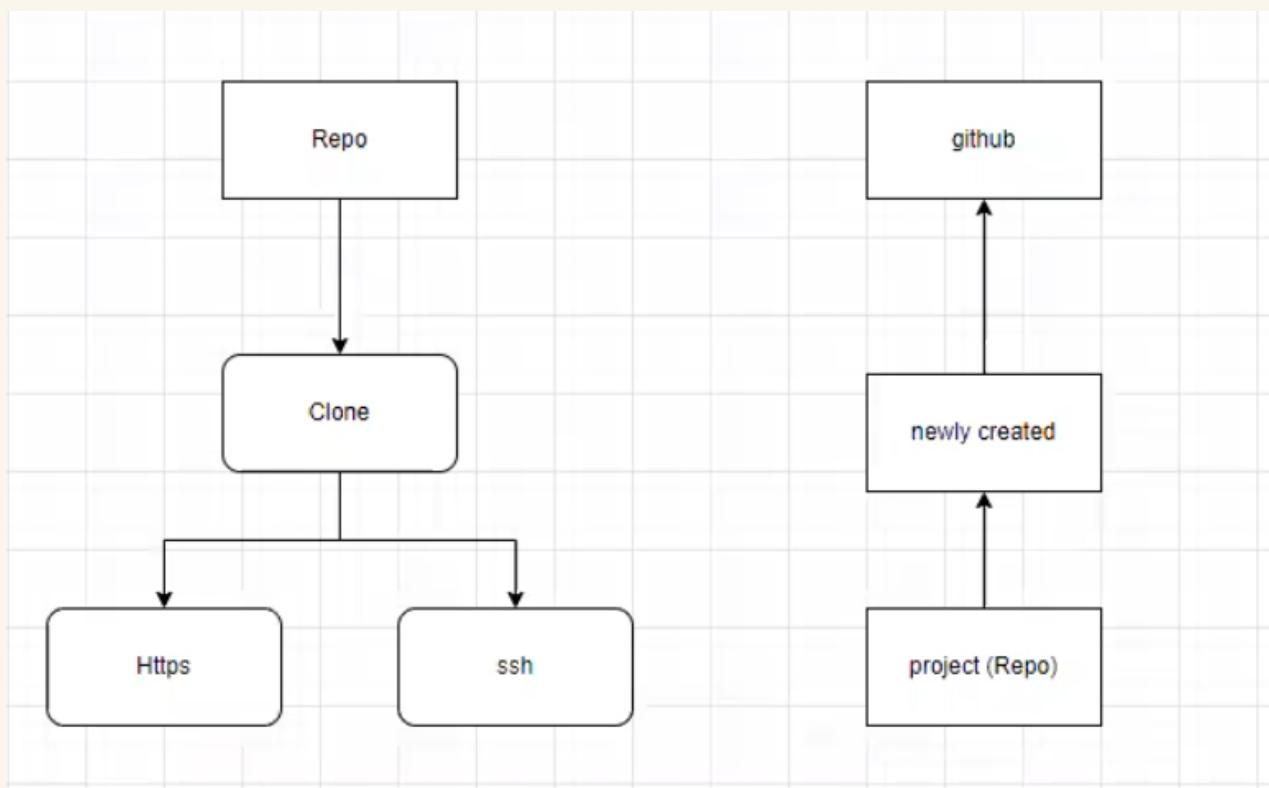
```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 39 Git/clone/step-1-hasantha|REBASE 3/3)
$ eval "$(ssh-agent -s)"
Agent pid 532
```

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 39 Git/clone/step 2 (feature-1-hasantha|REBASE 3/3)
$ ssh-add ~/.ssh/id_ed25519
Identity added: /c/Users/Prasa/.ssh/id_ed25519 (prasadkarunanayaka2016@gmail.com)
```



```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 39 Git/clone/step-1-hasantha|REBASE 3/3)
$ ssh -t git@github.com
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvH.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
PTY allocation request failed on channel 0
```

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 39 Git/clo  
t-1-hasantha|REBASE 3/3)  
$ git clone git@github.com:gd4Ark/star-battle.git  
Cloning into 'star-battle'...  
remote: Enumerating objects: 307, done.  
remote: Counting objects: 100% (9/9), done.  
remote: Compressing objects: 100% (3/3), done.  
remote: Total 307 (delta 6), reused 6 (delta 6), pack-reused 298  
Receiving objects: 100% (307/307), 2.01 MiB | 1.42 MiB/s, done.  
Resolving deltas: 100% (88/88), done.
```



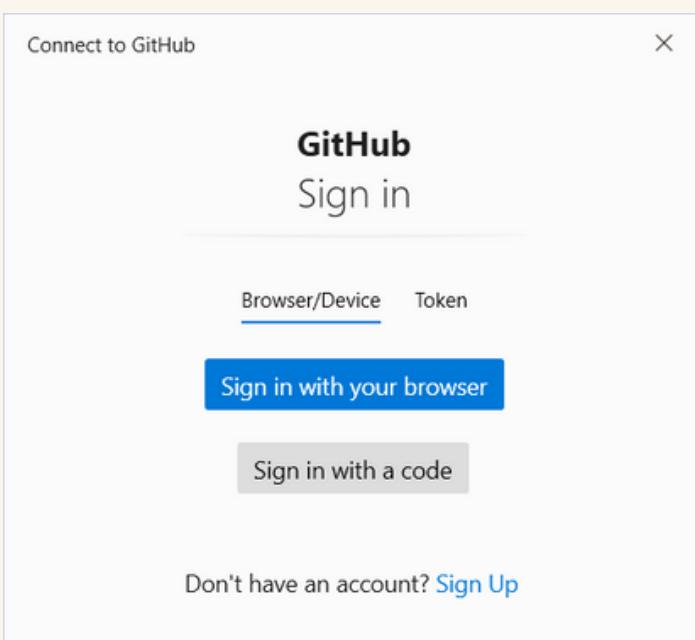
```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack
$ git status
On branch master
nothing to commit, working tree clean

Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack
$ git remote add origin https://github.com/pr
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack
$ git remote
origin

Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack
$ git remote -v
origin  https://github.com/prasadkaru/my-first-repo.git (fetch)
origin  https://github.com/prasadkaru/my-first-repo.git (push)

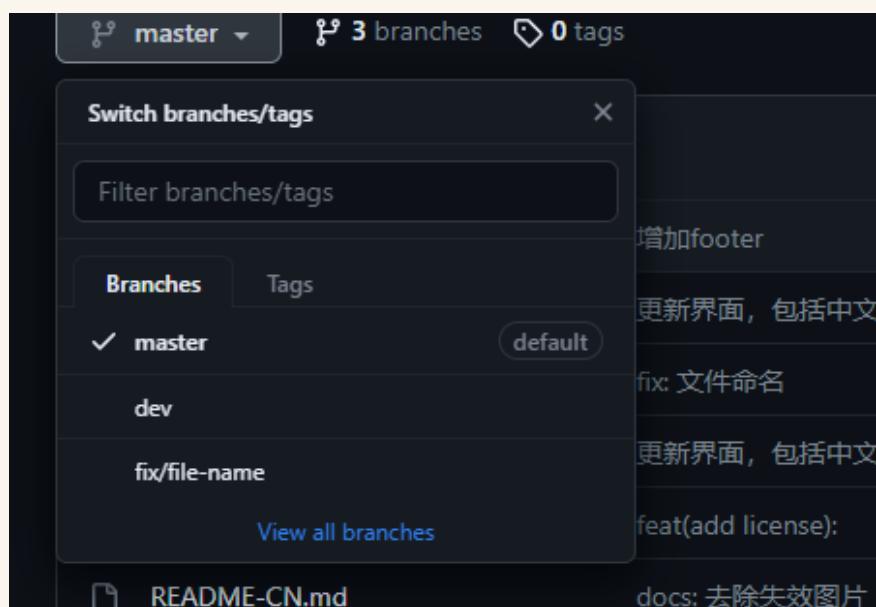
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack
$ git branch -M main
```

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 39 Git/my-first-repo
$ git push -u origin main
```



After modify as you want

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 39 Git  
at-1-test)  
$ git push -u feat-1-test  
fatal: 'feat-1-test' does not appear to be a git repository  
fatal: Could not read from remote repository.  
  
Please make sure you have the correct access rights  
and the repository exists.  
  
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 39 Git  
at-1-test)  
$ git push -u origin feat-1-test  
Enumerating objects: 4, done.  
Counting objects: 100% (4/4), done.  
Delta compression using up to 8 threads  
Compressing objects: 100% (2/2), done.  
Writing objects: 100% (3/3), 276 bytes | 276.00 KiB/s, done.  
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0  
remote: Resolving deltas: 100% (1/1), completed with 1 local o  
remote:  
remote: Create a pull request for 'feat-1-test' on GitHub by v  
remote:     https://github.com/prasadkaru/my-first-repo/pull/  
remote:
```



Code Issues Pull requests Actions Projects Wiki Security Insights Settings

General GitHub Pages

Access GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Beta Actions

Webhooks

Environments

Codespaces

Pages

Build and deployment

Source Deploy from a branch

Branch GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. Learn more

main / (root) Save

Visibility GITHUB ENTERPRISE

With a GitHub Enterprise account, you can restrict access to your GitHub Pages site by publishing it privately. published site can only be accessed by people with read access to the repository the site is published from. Yo

This screenshot shows the GitHub Pages settings page for a repository. The left sidebar contains links for General, Access, Collaborators, Moderation options, Code and automation (with sub-links for Branches, Tags, Rules, Actions, Webhooks, Environments, Codespaces, and Pages), and Build and deployment. The Build and deployment section is expanded, showing Source (Deploy from a branch), Branch (disabled), and Visibility (GITHUB ENTERPRISE). A note at the bottom explains that with a GitHub Enterprise account, access to the site can be restricted.

Upstream

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 40 Git/ups
$ git init
Initialized empty Git repository in E:/developerstack/Git/Day 40 Git/ups/
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 40 Git/ups
$ cd upstream/
bash: cd: upstream/: No such file or directory
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 40 Git/ups
$ git init
Reinitialized existing Git repository in E:/developerstack/Git/Day 40 Git/ups/.git/
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 40 Git/ups
$ git add .
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 40 Git/ups
$ git status
On branch master

No commits yet

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

Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 40 Git/ups
$ git config --local user.email prasadkarunanayaka2016@gmail.com

Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 40 Git/ups
$ git config --local user.name prasad

Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 40 Git/ups
$ git commit -m "init from the local repo"
[master (root-commit) 7e2659d] init from the local repo
 1 file changed, 1 insertion(+)
 create mode 100644 index.html

Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 40 Git/ups
$ git remote add origin https://github.com/prasadkaru/upstream.git

Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 40 Git/ups
$ git branch -m main

Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 40 Git/ups
$ git branch
* main

Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 40 Git/ups
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 238 bytes | 238.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/prasadkaru/upstream.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

add Code
login to
main
branch.

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 40 Git/upstream (feat-1-login)
$ git push origin feat-1-login
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 292 bytes | 292.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feat-1-login' on GitHub by visiting:
remote:     https://github.com/prasadkaru/upstream/pull/new/feat-1-login
remote:
To https://github.com/prasadkaru/upstream.git
 * [new branch]      feat-1-login -> feat-1-login
```

add code to feat-1-login branch.

If we want to "git push" command
and use default one

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 40 Git/upstream (feat-1-login)
$ git push
fatal: The current branch feat-1-login has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin feat-1-login

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

```

Prasa@DESKTOP-C27UDNE MINGW64 /e/developer
$ git add .

Prasa@DESKTOP-C27UDNE MINGW64 /e/developer
$ git commit -m "finalized log"
[feat-1-login 7712a66] finalized log
 1 file changed, 3 insertions(+), 1 deletion(-)

Prasa@DESKTOP-C27UDNE MINGW64 /e/developer
$ git log --oneline
7712a66 (HEAD -> feat-1-login) finalized log
62b692f (origin/main, main) another message
8de6217 init
7e2659d init from the local repo

Prasa@DESKTOP-C27UDNE MINGW64 /e/developer
$ git push origin feat-1-login
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 292 bytes | 292 bytes
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feat-1-login'!
remote:   https://github.com/prasadkaru/upstream
remote:
To https://github.com/prasadkaru/upstream.
 * [new branch]      feat-1-login -> feat-1-login

```

← traditional

way
↓

But we want
to "push" only
↓

```

initialize
work 2
work 31
#####
login development
#####
local repo
#####
last line

```

```

Prasa@DESKTOP-C27UDNE MINGW64 /e/developer
$ git push -u origin feat-1-login
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 303 bytes | 303 bytes
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/prasadkaru/upstream
    7712a66..fac61f1  feat-1-login -> feat-1-login
branch 'feat-1-login' set up to track 'origin/feat-1-login'

Prasa@DESKTOP-C27UDNE MINGW64 /e/developer
$ git add .

Prasa@DESKTOP-C27UDNE MINGW64 /e/developer
$ git commit -m "uostream com"
[feat-1-login dca22fd] uostream com
 1 file changed, 3 insertions(+), 1 deletion(-)

Prasa@DESKTOP-C27UDNE MINGW64 /e/developer
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 287 bytes | 287 bytes
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), com-
To https://github.com/prasadkaru/upstream
    fac61f1..dca22fd  feat-1-login -> feat-1-login

```

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developop  
$ notepad index.html
```

open the notepad

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/D  
$ notepad index.html  
  
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/D  
$ git init  
Initialized empty Git repository in E:/developerstack/  
/  
  
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/D  
$ git add .  
  
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/D  
$ git commit -m "first"  
Author identity unknown  
  
*** Please tell me who you are.  
  
Run  
  
git config --global user.email "you@example.com"  
git config --global user.name "Your Name"  
  
to set your account's default identity.  
Omit --global to set the identity only in this repository  
  
fatal: unable to auto-detect email address (got 'Pras  
  
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/D  
$ git config --local user.email prasadkarunanya2016  
  
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/D  
$ git config --local name.email prasad  
  
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/D  
$ git commit -m "first"  
[master (root-commit) cdfc587] first  
1 file changed, 1 insertion(+)  
create mode 100644 index.html  
  
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/D  
$ git remote
```

Create local
Operation

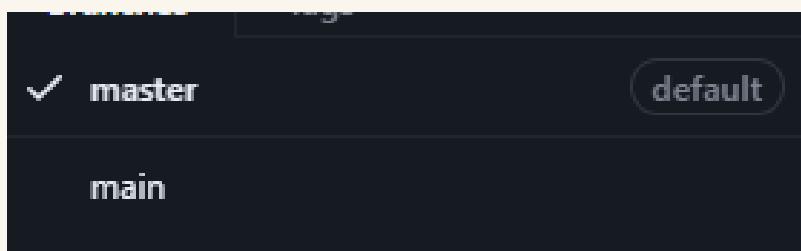
add Repository

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 40 Git/Main vs master  
$ git remote add origin https://github.com/prasadkaru/main-vs-master.git
```

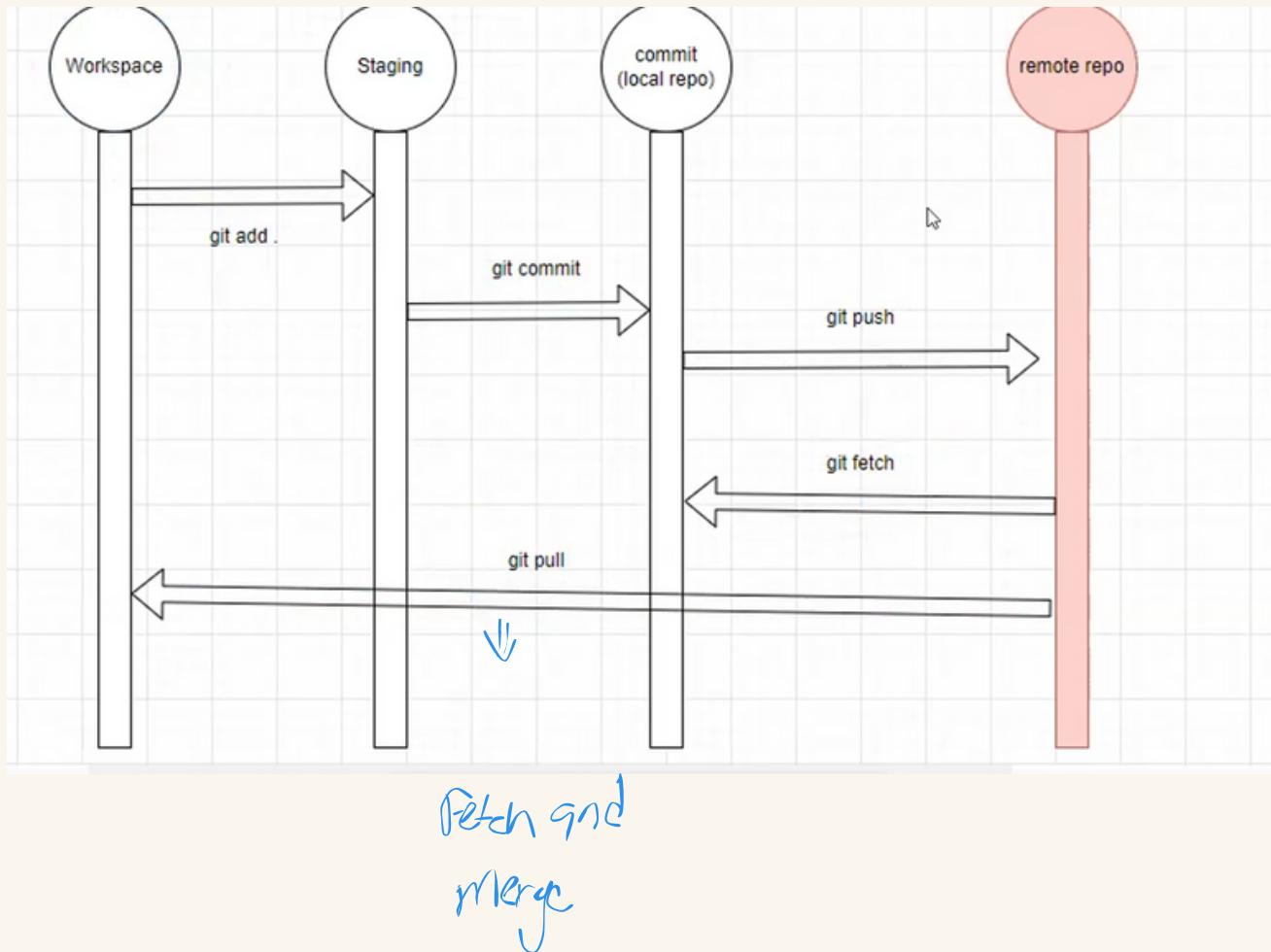
```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git/Day 40 Git/Main
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 226 bytes | 226.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/prasadkaru/main-vs-master.git
 * [new branch]      master -> master
```

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/
$ git checkout main
Switched to branch 'main'
```

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developerstack/Git
$ git push -u origin main
Total 0 (delta 0), reused 0 (delta 0), pack-reused
remote:
remote: Create a pull request for 'main' on GitHub
remote:      https://github.com/prasadkaru/main-vs-
remote:
To https://github.com/prasadkaru/main-vs-master.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```



Change
default



```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developersta
$ git fetch origin
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (6/6), done
remote: Total 9 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (9/9), 1.98 KiB | 2.0
From https://github.com/prasadkaru/fetch-full
  a074f6f..793109a  main      -> origin/main
```

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developersta
$ git branch
* main
```

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/developersta
$ git status
On branch main
Your branch is behind 'origin/main' by 3 commits
  (use "git pull" to update your local branch)

nothing to commit, working tree clean
```

```
Prasa@DESKTOP-C27UDNE MINGW64 /e/  
$ git pull origin  
Updating a074f6f..793109a  
Fast-forward  
 first-page.html | 1 +  
 1 file changed, 1 insertion(+)  
 create mode 100644 first-page.html
```

Read me File

```
# heading 1
## heading 2
### heading 3
---
##### heading 6

# Languages
+ java
+ c
  + c++
+ ruby
+ swift
---
## Order List
1. namal
2. Ranil
3. Sajith

# code snipt
```
code data
```
```

heading 1

heading 2

heading 3

heading 6

Languages

- java
- c
 - c++
- ruby
- swift

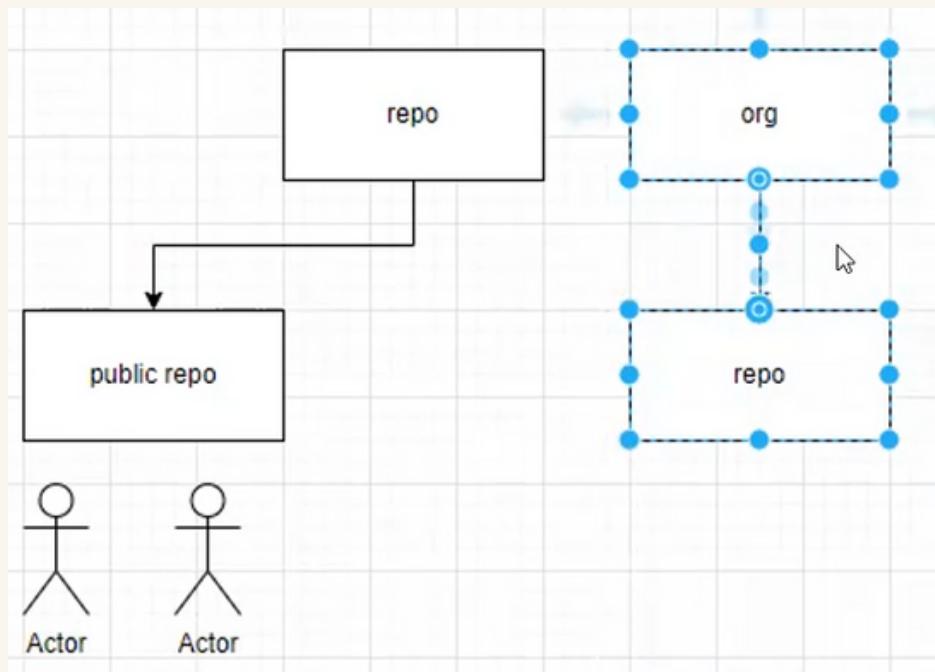
Order List

1. namal
2. Ranil
3. Sajith

code snipt

code data

Collabration



fork

Create organization

Cherry pick

```
Prasa@DESKTOP-C27UDNE MINGW64 /$ git commit -m "init"  
[master (root-commit) c26d085] init  
 1 file changed, 1 insertion(+)  
 create mode 100644 index
```

```
Prasa@DESKTOP-C27UDNE MINGW64 /$ git log --oneline  
c26d085 (HEAD -> master) init
```

```
Prasa@DESKTOP-C27UDNE MINGW64 /$ git branch v2
```

```
Prasa@DESKTOP-C27UDNE MINGW64 /$ git branch v1
```

```
Prasa@DESKTOP-C27UDNE MINGW64 /$ git branch v3
```

```
Prasa@DESKTOP-C27UDNE MINGW64 /$ git branch  
* master  
  v1  
  v2  
  v3
```

```
Prasa@DESKTOP-C27UDNE MINGW64 /$ git checkout v3  
Switched to branch 'v3'
```

```
Prasa@DESKTOP-C27UDNE MINGW64 /$ git branch  
  master  
  v1  
  v2  
* v3
```

Changes move another
branches

```
Prasa@DESKTOP-C27UDNE MINGW64 /$ git add .
```

```
Prasa@DESKTOP-C27UDNE MINGW64 /$ git commit -m "bug fix"  
[v3 77e6794] bug fix  
 1 file changed, 0 insertions(+)  
 create mode 100644 fix-bug.txt
```

```
Prasa@DESKTOP-C27UDNE MINGW64 /$ git log --oneline  
fatal: unrecognized argument: --
```

```
Prasa@DESKTOP-C27UDNE MINGW64 /$ git log --oneline  
77e6794 (HEAD -> v3) bug fix  
c26d085 (v2, v1, master) init
```

revert

Reset

