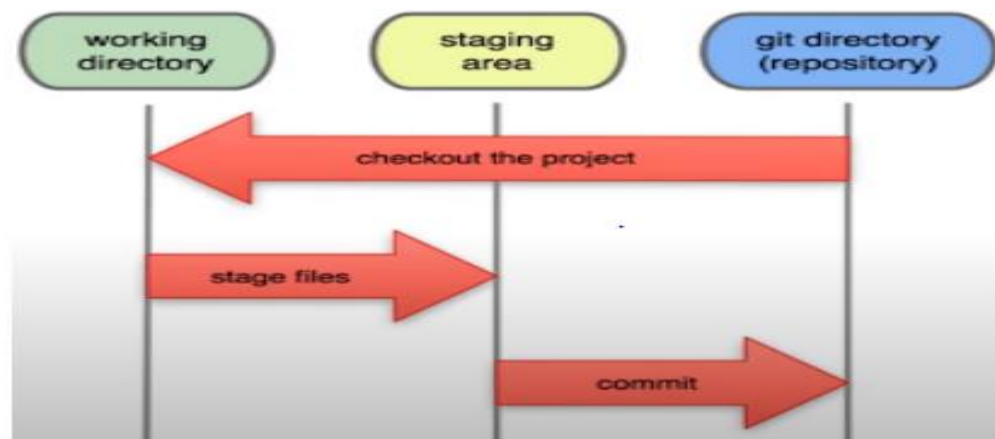


# GIT Assignment

## Git - architecture



### 1. Working Directory

- The path of the project folder currently working on and on which version control is needed.

### 2. Staging Area

- It has the files which the programmer wants to commit to the new version in remote repository.

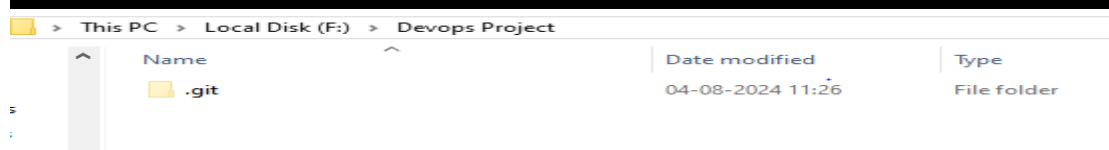
### 3. Git Repository

- Here the staged files are committed to the git repository(repo), which stores the new version of the files having the changes done.

## GIT Initializing

- `git init` : command is used to initialize the respective directory as git repository
- how to check a directory is git repository or not ??  
if `.git` folder is present in the current directory, then it is a git repository.

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project
$ git init
Initialized empty Git repository in F:/Devops Project/.git/
```



## Configuring

- git config --global user.name "Rajendra"
- git config --global user.email "[prp01928@gmail.com](mailto:prp01928@gmail.com)"

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git config --global user.name "Rajendra"

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git config --global user.email "prp01928@gmail.com"
```

## Add file and commit, status

- touch <file name> :to create a file
- git add : it add the file to the staging area for temporary storage.
- git add . :it will add all the files which are modified to staging area
- git status :it will show whether files are in workspace or staging area on in git repo.
- git commit -m "message" :this command commits the staged changes to the local repository.
- git log : gives history of file or repo.

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ touch file1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git add file1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git status
On branch master

No commits yet

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

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git commit -m "adding file1"
[master (root-commit) c37415f] adding file1
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git log
commit c37415f0f4a5899c5518cacec66d6b1d0b5ff263 (HEAD -> master)
Author: Rajendra <prp01928@gmail.com>
Date:   Sun Aug 4 11:30:08 2024 +0530

    adding file1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ touch {f1,f2,f3,f4}

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git add .

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git status
On branch master

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   f1
    new file:   f2
    new file:   f3
    new file:   f4
```

## Log's checking types

- git log --oneline
- git log <file name>
- git log --author="username"

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git log --oneline
895dbbb (HEAD -> master) updating file1
c37415f adding file1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git log file1
commit 895dbbb5fbae636dd95fe2b27f263bfc69b2679a (HEAD -> master)
Author: Rajendra <prp01928@gmail.com>
Date: Sun Aug 4 11:40:36 2024 +0530

    updating file1

commit c37415f0f4a5899c5518cacec66d6b1d0b5ff263
Author: Rajendra <prp01928@gmail.com>
Date: Sun Aug 4 11:30:08 2024 +0530

    adding file1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git log --author="Rajendra"
commit 895dbbb5fbae636dd95fe2b27f263bfc69b2679a (HEAD -> master)
Author: Rajendra <prp01928@gmail.com>
Date: Sun Aug 4 11:40:36 2024 +0530

    updating file1

commit c37415f0f4a5899c5518cacec66d6b1d0b5ff263
Author: Rajendra <prp01928@gmail.com>
Date: Sun Aug 4 11:30:08 2024 +0530

    adding file1
```

## Restore

- git restore --staged <file name> --to unstage the files
- Git restore --staged . --to unstage all the files

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   f1
    new file:   f2
    new file:   f3
    new file:   f4

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git restore --staged f1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   f2
    new file:   f3
    new file:   f4

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    f1
```

```

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   f1
        new file:   f2
        new file:   f3
        new file:   f4

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git restore --staged .

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        f1
        f2
        f3
        f4

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

```

## Branching

- git branch <branch name> --> to create a branch
- git branch --> to check number of branches. or list all branches
  - \* --> indicates the current branch or indicates we are on that branch
- git checkout <branch name> --> to switch to another branch
- git branch -d branch name -> to delete the branch
- git checkout -b branch name -> to create new branch

```

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git branch
* master

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git branch lab1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git branch
lab1
* master

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git checkout lab1
Switched to branch 'lab1'

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (lab1)
$ git branch
* lab1
  master

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (lab1)
$ git checkout master
Switched to branch 'master'

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git branch
lab1
* master

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git branch -d lab1
Deleted branch lab1 (was 895dbbb).

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git branch
* master

```



## Tagging

- `git tag tagname` --> to create a tag name for recent commit id
- `git checkout tagname` --> switch to tag
- `git tag` --> to list all tags
- `git tag -d tagname` -> to delete the tag
- `git tag tagname <commit-id>` --> for specific commit id

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git log --oneline
782ec02 (HEAD -> master) t1 created
895dbbb updating file1
c37415f adding file1
```

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git tag p1
```

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git log --oneline
782ec02 (HEAD -> master, tag: p1) t1 created
895dbbb updating file1
c37415f adding file1
```

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git tag p2 c37415f
```

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git log --oneline
782ec02 (HEAD -> master, tag: p1) t1 created
895dbbb updating file1
c37415f (tag: p2) adding file1
```

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git tag
p1
p2
```

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git tag -d p1
Deleted tag 'p1' (was 782ec02)
```

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git log --oneline
782ec02 (HEAD -> master) t1 created
895dbbb updating file1
c37415f (tag: p2) adding file1
```

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git tag
p2
```

## Merging

- `git merge -->` to merge specified branch to checked out branch  
merging is only 1 way -- from source to destination.

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git branch
* master

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git branch lab1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git checkout lab1
Switched to branch 'lab1'

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (lab1)
$ touch file1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (lab1)
$ vi file1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (lab1)
$ git add file1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (lab1)
$ git commit -m "adding file1 from lab1"
[lab1 afe3e03] adding file1 from lab1
1 file changed, 1 insertion(+)

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (lab1)
$ cat file1
adding line1
adding llin2 for merging

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (lab1)
$ git checkout master
Switched to branch 'master'

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ cat file1
adding line1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git merge lab1
Updating 782ec02..afe3e03
Fast-forward
 file1 | 1 +
1 file changed, 1 insertion(+)

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git log --oneline
afe3e03 (HEAD -> master, lab1) adding file1 from lab1
782ec02 t1 created
895dbbb updating file1
c37415f (tag: p2) adding file1
```

## Rebase

Rebase:-similar to merge, rewrite the commit histroy

➤ git rebase <branch name>

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (lab1)
$ git log --oneline
afe3e03 (HEAD -> lab1, master) adding file1 from lab1
782ec02 t1 created
895dbbb updating file1
c37415f (tag: p2) adding file1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (lab1)
$ vi t1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (lab1)
$ git add t1
warning: in the working copy of 't1', LF will be replaced by

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (lab1)
$ git status
On branch lab1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   t1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (lab1)
$ git commit -m "updated t1 in lab1"
[lab1 6cfd78d] updated t1 in lab1
 1 file changed, 1 insertion(+)

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (lab1)
$ git log --oneline
6cfd78d (HEAD -> lab1) updated t1 in lab1
afe3e03 (master) adding file1 from lab1
782ec02 t1 created
895dbbb updating file1
c37415f (tag: p2) adding file1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (lab1)
$ vi t1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (lab1)
$ git add t1
warning: in the working copy of 't1', LF will be replaced by

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (lab1)
$ git commit -m "updated2 t1 in lab1"
[lab1 18fcdce] updated2 t1 in lab1
 1 file changed, 1 insertion(+)

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (lab1)
$ git log --oneline
18fcdce (HEAD -> lab1) updated2 t1 in lab1
6cfd78d updated t1 in lab1
afe3e03 (master) adding file1 from lab1
782ec02 t1 created
895dbbb updating file1
c37415f (tag: p2) adding file1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (lab1)
$ git checkout master
Switched to branch 'master'

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ cat t1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
```



```

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ cat t1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git rebase lab1
Successfully rebased and updated refs/heads/master.

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git log --oneline
18fcdce (HEAD -> master, lab1) updated2 t1 in lab1
6cfd78d updated t1 in lab1
afe3e03 adding file1 from lab1
782ec02 t1 created
895dbbb updating file1
c37415f (tag: p2) adding file1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ cat t1
adding line to rebase
adding line for 2nd commit

```

### cherry\_pick

- git cherry\_pick commit\_id --> to merge specific commit on branch(currently checkout branch)

```

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git checkout lab1
Switched to branch 'lab1'

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (lab1)
$ git log --oneline
b4d1605 (HEAD -> lab1) adding pick
389b44e adding cherry
18fcdce (master) updated2 t1 in lab1
6cfd78d updated t1 in lab1
afe3e03 adding file1 from lab1
782ec02 t1 created
895dbbb updating file1
c37415f (tag: p2) adding file1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (lab1)
$ git checkout master
Switched to branch 'master'

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git log --oneline
18fcdce (HEAD -> master) updated2 t1 in lab1
6cfd78d updated t1 in lab1
afe3e03 adding file1 from lab1
782ec02 t1 created
895dbbb updating file1
c37415f (tag: p2) adding file1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ ls
file1  t1

```



```

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git cherry-pick 389b44e
[master bad67f3] adding cherry
Date: Sun Aug 4 14:06:14 2024 +0530
1 file changed, 1 insertion(+)
create mode 100644 cherry

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ ls
cherry  file1  t1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Devops Project (master)
$ git log --oneline
bad67f3 (HEAD -> master) adding cherry
18fcdce updated2 t1 in lab1
6cfd78d updated t1 in lab1
afe3e03 adding file1 from lab1
782ec02 t1 created
895dbbb updating file1
c37415f (tag: p2) adding file1

```

### git stash

Git temporarily saves your data safely without committing. We can switch the branches easily without committing current branch working files .

- git stash --will add the working files to temporary location without committing
- git stash list -- to check which all files are stored in temporary location with id no
- git stash pop --to retrieve the file from storage to workspace

```

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (pro1)
$ ls
file1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (pro1)
$ vi file1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (pro1)
$ git add file1
warning: in the working copy of 'file1', LF will be repla

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (pro1)
$ git commit -m "updating file1 from master"
[pro1 6a055af] updating file1 from master
1 file changed, 2 insertions(+)

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (pro1)
$ git checkout pro1
Already on 'pro1'

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (pro1)
$ git checkout master
Switched to branch 'master'

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (master)
$ vi file1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (master)
$ git add file1
warning: in the working copy of 'file1', LF will be repla

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (master)
$ git commit -m "updating file1"
[master 372ad88] updating file1
1 file changed, 1 insertion(+)

```

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (master)
$ git checkout pro1
Switched to branch 'pro1'

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (pro1)
$ vi file1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (pro1)
$ git checkout master
error: Your local changes to the following files would be overwritten by checkout:
      file1
Please commit your changes or stash them before you switch branch
Aborting

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (pro1)
$ git stash
Saved working directory and index state WIP on pro1: 6a055af updating file1 from master

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (pro1)
$ git checkout master
Switched to branch 'master'

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (master)
$ git stash list
stash@{0}: WIP on pro1: 6a055af updating file1 from master

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (master)
$ git checkout pro1
Switched to branch 'pro1'

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (pro1)
$ git stash list
stash@{0}: WIP on pro1: 6a055af updating file1 from master

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (pro1)
$ git stash pop
On branch pro1
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:   file1

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (74f82cf43ffffafb4e4e82fa266386fe10431bc88)

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (pro1)
$ git stash list

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (pro1)
$ |
```

## Git Squash

Squash : it is a technique to condense large number of commits to make into small number of meaningful commits so that we can make git history clear.

➤ `git rebase -i HEAD~n`

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (pro1)
$ git log
commit 12ed6d129f145c33c20b71e9c443ff4bb1d2b090 (HEAD -> pro1)
Author: Rajendra <prp01928@gmail.com>
Date: Sun Aug 4 15:35:46 2024 +0530

    updating file1 pro1

commit 6a055af32fb8448de18becc4691715da038bede0
Author: Rajendra <prp01928@gmail.com>
Date: Sun Aug 4 15:22:42 2024 +0530

    updating file1 from master

commit 73386a3bde903ab17bfc7f60781118b1aaf6a7f6
Author: Rajendra <prp01928@gmail.com>
Date: Sun Aug 4 15:18:13 2024 +0530

    adding file1 in master

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (pro1)
$ git rebase -i HEAD~3
fatal: invalid upstream 'HEAD~3'

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (pro1)
$ git rebase -i HEAD~2
[detached HEAD 1f78925] updating file1 from master
Date: Sun Aug 4 15:22:42 2024 +0530
1 file changed, 2 insertions(+)
Successfully rebased and updated refs/heads/pro1.

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (pro1)
$ git log
commit 1f7892538f5eaf621837f2d93800a89159247436 (HEAD -> pro1)
Author: Rajendra <prp01928@gmail.com>
Date: Sun Aug 4 15:22:42 2024 +0530

    updating file1 from master

    updating file1 pro1

commit 73386a3bde903ab17bfc7f60781118b1aaf6a7f6
Author: Rajendra <prp01928@gmail.com>
Date: Sun Aug 4 15:18:13 2024 +0530

    adding file1 in master

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (pro1)
pick 6a055af updating file1 from master
squash| 12ed6d1 updating file1 pro1

# Rebase 73386a3..12ed6d1 onto 73386a3 (2 commands)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup [-C | -c] <commit> = like "squash" but keep only the p
#
```

## Git revert

- git revert - we can go back to the workspace after committing, but commit history will be stored
- git revert HEAD~1

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (master)
$ git log
commit 372ad88d30fd7fe25299d74879d38df92b398e39 (HEAD -> master)
Author: Rajendra <prp01928@gmail.com>
Date: Sun Aug 4 15:24:14 2024 +0530

    updating file1

commit 73386a3bde903ab17bfc7f60781118b1aaf6a7f6
Author: Rajendra <prp01928@gmail.com>
Date: Sun Aug 4 15:18:13 2024 +0530

    adding file1 in master

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (master)
$ git log --oneline
372ad88 (HEAD -> master) updating file1
73386a3 adding file1 in master

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (master)
$ git revert 372ad88
[master 5f50362] Revert "updating file1"
 1 file changed, 1 deletion(-)

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (master)
$ git log
commit 5f503625ca239a35154406761dfc740dca3cc5ca (HEAD -> master)
Author: Rajendra <prp01928@gmail.com>
Date: Sun Aug 4 16:01:38 2024 +0530

    Revert "updating file1"

    This reverts commit 372ad88d30fd7fe25299d74879d38df92b398e39.

commit 372ad88d30fd7fe25299d74879d38df92b398e39
Author: Rajendra <prp01928@gmail.com>
Date: Sun Aug 4 15:24:14 2024 +0530

    updating file1

commit 73386a3bde903ab17bfc7f60781118b1aaf6a7f6
Author: Rajendra <prp01928@gmail.com>
Date: Sun Aug 4 15:18:13 2024 +0530

    adding file1 in master
```

```
Revert "updating file1"

This reverts commit 5f503625ca239a35154406761dfc740dca3cc5ca.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# Changes to be committed:
#   modified:   file1
#
```

wq!

## Git reset

- git reset - we can go back to the workspace after committing, but commit history is removed
- git reset --hard HEAD~1
- git reset --soft HEAD~1
- git reset --mixed HEAD~1

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (master)
$ git log --oneline
80e87ce (HEAD -> master) adding t3
57277ac adding t2
f32bbd2 adding t1
5dd37bc Revert "updating file1"
5f50362 Revert "updating file1"
372ad88 updating file1
73386a3 adding file1 in master
```

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (master)
$ git reset --hard HEAD~1
HEAD is now at 57277ac adding t2
```

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (master)
$ git log --oneline
57277ac (HEAD -> master) adding t2
f32bbd2 adding t1
5dd37bc Revert "updating file1"
5f50362 Revert "updating file1"
372ad88 updating file1
73386a3 adding file1 in master
```

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (master)
$ git log --oneline
57277ac (HEAD -> master) adding t2
f32bbd2 adding t1
5dd37bc Revert "updating file1"
5f50362 Revert "updating file1"
372ad88 updating file1
73386a3 adding file1 in master
```

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (master)
$ git reset --soft HEAD~1
```

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (master)
$ git log --oneline
f32bbd2 (HEAD -> master) adding t1
5dd37bc Revert "updating file1"
5f50362 Revert "updating file1"
372ad88 updating file1
73386a3 adding file1 in master
```

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   t2
```

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (master)
$ git reset --mixed HEAD~1
```

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (master)
$ git log --oneline
5dd37bc (HEAD -> master) Revert "updating file1"
5f50362 Revert "updating file1"
372ad88 updating file1
73386a3 adding file1 in master
```

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Project (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        t1
        t2
```

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

## Git Clone

- Git clone github url

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central (master)
$ git init
Reinitialized existing Git repository in F:/Git Central/.git/

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central (master)
$ ls

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central (master)
$ git branch

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central (master)
$ git clone git@github.com:raja1928/Git-assig.git
Cloning into 'Git-assig'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central (master)
$ ls
Git-assig/

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central (master)
$ cd Git-assig

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central/Git-assig (main)
$ ls
File1  file2

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central/Git-assig (main)
$ git branch
* main

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central/Git-assig (main)
```

# Git push

- git push -pushing the updated code to central from local

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central/Git-assig (main)
$ git branch
* main

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central/Git-assig (main)
$ vi file1



Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central/Git-assig (main)
$ vi file2


Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central/Git-assig (main)
$ git add .




Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central/Git-assig (main)
$ git commit -m "updating file1,2 in local"
[main 788f774] updating file1,2 in local
2 files changed, 2 insertions(+)

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central/Git-assig (main)
$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 364 bytes | 182.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:raja1928/Git-assig.git
   eec8ed7..788f774  main -> main
```

Git-assig / File1

 **raja1928** updating file1,2 in local 788f774 · 16 minutes ago 



Code Blame 2 lines (2 loc) · 43 Bytes  Code 55% faster with GitHub Copilot


Raw   




1 Adding file1 to clone

2 Adding line in local

Git-assig / file2

 **raja1928** updating file1,2 in local 788f774 · 17 minutes ago 



Code Blame 2 lines (2 loc) · 45 Bytes  Code 55% faster with GitHub Copilot


Raw   



1 adding file2 for clone


2 adding line in local


## Commits




History for Git-assig / File1 on main  All users  All time

 Commits on Aug 4, 2024

updating file1,2 in local 788f774   <> ...

 raja1928 committed 10 minutes ago

Create File1 


 raja1928 committed 30 minutes ago Verified 0073051   <> ...



# Git pull

➤ `git pull` :- pulling the updated code from the central to local

[Git-assig](#) / [File1](#)

 **raja1928** Update File1 9bafae · now Hist

Code

Blame


3 lines (3 loc) · 66 Bytes

Code 55% faster with GitHub Copilot

Raw

```
1 Adding file1 to clone
2 Adding line in local
3 Adding line in central
```

[Git-assig](#) / [file2](#)

 **raja1928** Update file2 ad35f87 · 2 minutes ago Hist

Code

Blame

3 lines (3 loc) · 67 Bytes

Code 55% faster with GitHub Copilot

Raw

```
1 adding file2 for clone
2 adding line in local
3 adding line in central
```

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central/Git-assig (main)
$ cat file1
Adding file1 to clone
Adding line in local

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central/Git-assig (main)
$ cat file2
adding file2 for clone
adding line in local

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central/Git-assig (main)
$ git pull
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), 1.86 KiB | 7.00 KiB/s, done.
From github.com:raja1928/Git-assig
   788f774..ad35f87  main      -> origin/main
Updating 788f774..ad35f87
Fast-forward
 File1 | 1 +
 file2 | 3 ++-
 2 files changed, 3 insertions(+), 1 deletion(-)

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central/Git-assig (main)
$ cat file1
Adding file1 to clone
Adding line in local
Adding line in central

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central/Git-assig (main)
$ cat file2
adding file2 for clone
adding line in local
adding line in central
```

## Git Fetch

- `git fetch <repo> <branch>`
- `git fetch origin master`

```
Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central/Git-assig (main)
$ ls
File1  file2

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central/Git-assig (main)
$ git fetch
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 9 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (9/9), 2.81 KiB | 7.00 KiB/s, done.
From github.com:raja1928/Git-assig
   ad35f87..7cee8a3  main       -> origin/main

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central/Git-assig (main)
$ ls -la
total 6
drwxr-xr-x 1 Vishnu 197121  0 Aug  4 17:12 ./
drwxr-xr-x 1 Vishnu 197121  0 Aug  4 16:36 ../
drwxr-xr-x 1 Vishnu 197121  0 Aug  4 17:12 .git/
-rw-r--r-- 1 Vishnu 197121 69 Aug  4 17:12 File1
-rw-r--r-- 1 Vishnu 197121 70 Aug  4 17:12 file2

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central/Git-assig (main)
$ cd /.git
bash: cd: /.git: No such file or directory

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central/Git-assig (main)
$ cd .git

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central/Git-assig/.git (GIT_DIR!)
$ ls -la
total 20
drwxr-xr-x 1 Vishnu 197121  0 Aug  4 17:12 ./
drwxr-xr-x 1 Vishnu 197121  0 Aug  4 17:12 ../
-rw-r--r-- 1 Vishnu 197121 26 Aug  4 16:42 COMMIT_EDITMSG
-rw-r--r-- 1 Vishnu 197121 89 Aug  4 21:35 FETCH_HEAD
-rw-r--r-- 1 Vishnu 197121 21 Aug  4 16:36 HEAD
-rw-r--r-- 1 Vishnu 197121 41 Aug  4 17:12 ORIG_HEAD
-rw-r--r-- 1 Vishnu 197121 296 Aug  4 16:36 config
-rw-r--r-- 1 Vishnu 197121 73 Aug  4 16:36 description
drwxr-xr-x 1 Vishnu 197121  0 Aug  4 16:36 hooks/
-rw-r--r-- 1 Vishnu 197121 209 Aug  4 17:12 index
drwxr-xr-x 1 Vishnu 197121  0 Aug  4 16:36 info/
drwxr-xr-x 1 Vishnu 197121  0 Aug  4 16:36 logs/
drwxr-xr-x 1 Vishnu 197121  0 Aug  4 21:35 objects/
-rw-r--r-- 1 Vishnu 197121 112 Aug  4 16:36 packed-refs
drwxr-xr-x 1 Vishnu 197121  0 Aug  4 16:36 refs/

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central/Git-assig (main)
$ git branch -r
  origin/HEAD -> origin/main
  origin/main
```

```

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central/Git-assig (main)
$ git merge origin/main
Updating ad35f87..7cee8a3
Fast-forward
 File1 | 2 ++
 file3 | 1 +
 file2 | 2 ++
 3 files changed, 5 insertions(+)
 create mode 100644 file3

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central/Git-assig (main)
$ ls
File1 'file 3' file2

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central/Git-assig (main)
$ cat file1
Adding file1 to clone
Adding line in local
Adding line in central
adding file for fetch
adding lines

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Git Central/Git-assig (main)
$ cat file2
adding file2 for clone
adding line in local
adding line in central
adding to check fetch
line 5

```

## Git Pull Request

```

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Pull request
$ git init
Initialized empty Git repository in F:/Pull request/.git/

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Pull request (master)
$ git clone git@github.com:raja1928/Git-assig.git
Cloning into 'Git-assig'...
remote: Enumerating objects: 31, done.
remote: Counting objects: 100% (31/31), done.
remote: Compressing objects: 100% (22/22), done.
remote: Total 31 (delta 4), reused 10 (delta 1), pack-reused 0
Receiving objects: 100% (31/31), 7.13 KiB | 1.78 MiB/s, done.
Resolving deltas: 100% (4/4), done.

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Pull request (master)
$ ls
Git-assig/

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Pull request (master)
$ cd Git-assig

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Pull request/Git-assig (main)
$ ls
File1 'file 3' file2 file4 file5

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Pull request/Git-assig (main)
$ git checkout -b branch1
Switched to a new branch 'branch1'

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Pull request/Git-assig (branch1)
$ touch file1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Pull request/Git-assig (branch1)
$ vi file1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Pull request/Git-assig (branch1)
$ vi file10

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Pull request/Git-assig (branch1)
$ git add file10
warning: in the working copy of 'file10', LF will be replaced by CRLF

```

```

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Pull request/Git-assig (branch1)
$ git commit -m "adding file10"
[branch1 c9b537e] adding file10
1 file changed, 1 insertion(+)
create mode 100644 file10

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Pull request/Git-assig (branch1)
$ git push origin branch1
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 226 bytes | 226.00 KiB/s, done.
Total 2 (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 'branch1' on GitHub by visiting:
remote:   https://github.com/raja1928/Git-assig/pull/new/branch1
remote:
To github.com:raja1928/Git-assig.git
* [new branch]      branch1 -> branch1

Vishnu@DESKTOP-IABD7PN MINGW64 /f/Pull request/Git-assig (branch1)

```

raja1928 / Git-assig

Type [/](#) to search

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

Git-assig

Public

Pin

Unwatch 1

branch1

had recent pushes 1 second ago

Compare & pull request

main

1 Branch

0 Tags

Go to file

Add file

<> Code

raja1928

adding file5

c4f510f · 26 minutes ago

10 Commits

File1	Update File1	44 minutes ago
file 3	Create file 3	40 minutes ago
file2	Update file2	45 minutes ago
file4	adding file4	28 minutes ago

## adding file10 #1

Edit <> Code

[Open](#) raja1928 wants to merge 1 commit into [main](#) from [branch1](#)

Conversation 0

Commits 1

Checks 0

Files changed 1

+1 -0

raja1928 commented now

Owner

...

adding file10

adding file10

c9b537e

Require approval from specific reviewers before merging

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

Add rule X

Continuous integration has not been set up

[GitHub Actions](#) and [several other apps](#) can be used to automatically catch bugs and enforce style.

✓

This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request

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

Reviewers

No reviews

Still in progress? [Convert to draft](#)

Assignees

No one—[assign yourself](#)

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

## adding file10 #1

[Open](#) raja1928 wants to merge 1 commit into [main](#) from [branch1](#)

Conversation 0

Commits 1

Checks 0

Files changed 1

raja1928 commented 1 minute ago

Owner

...

adding file10

adding file10

c9b537e

Merge pull request #1 from raja1928/branch1

adding file10

This commit will be authored by 53326027+raja1928@users.noreply.github.com

Confirm merge

Cancel

Add a comment

Write

Preview

H B I

≡ <> 🔗


≡ ≡ ≡

🔗 @ ↩


# adding file10 #1


 Open raja1928 wants to merge 1 commit into `main` from `branch1` 


 Conversation **0**  Commits **1**  Checks **0**  Files changed **1**

 raja1928 commented 2 minutes ago Owner ...


adding file10

 adding file10 c9b537e












 raja1928 merged commit `8ca9fc6` into `main` now Revert

 Pull request successfully merged and closed Delete branch

You're all set—the `branch1` branch can be safely deleted.

 Add a comment


Write Preview


H B I           


Add your comment here...


Label issues and pull requests for new contributors


Now, GitHub will help potential first-time contributors [discover issues](#)!


Filters 


 is:pr is:closed


 Clear current search query, filters, and sorts

☐  0 Open


☒  1 Closed

Author 

Label 

☐  adding file10

#1 by raja1928 was merged 1 minute ago

 ProTip! Adding `no:label` will show everything with

————— **END** —————

