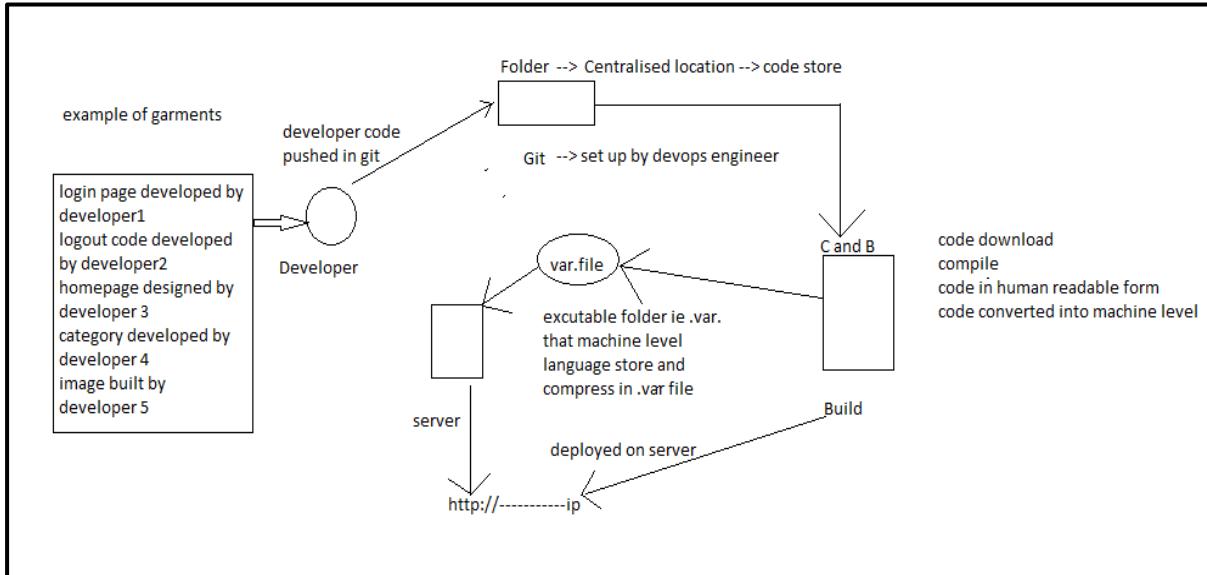


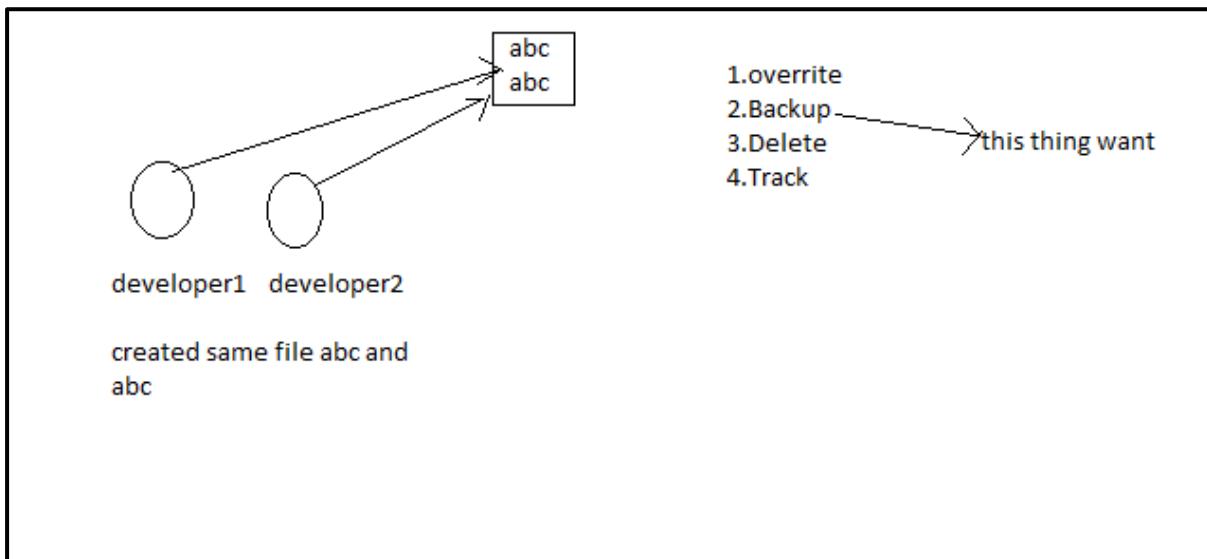
Date-5/10/2023

1.Git

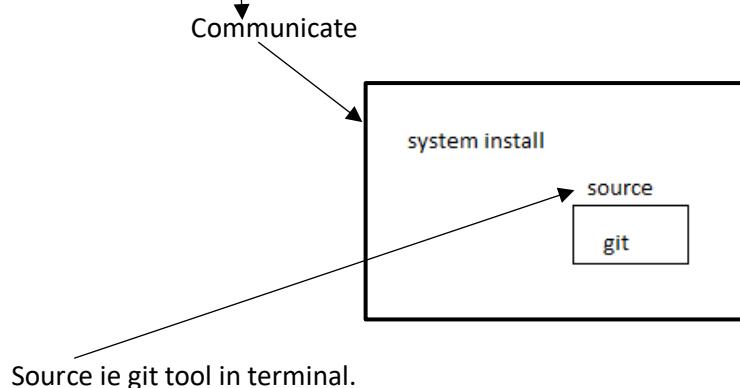
AJILE → Provide planning to devops.



Date-6/10/23



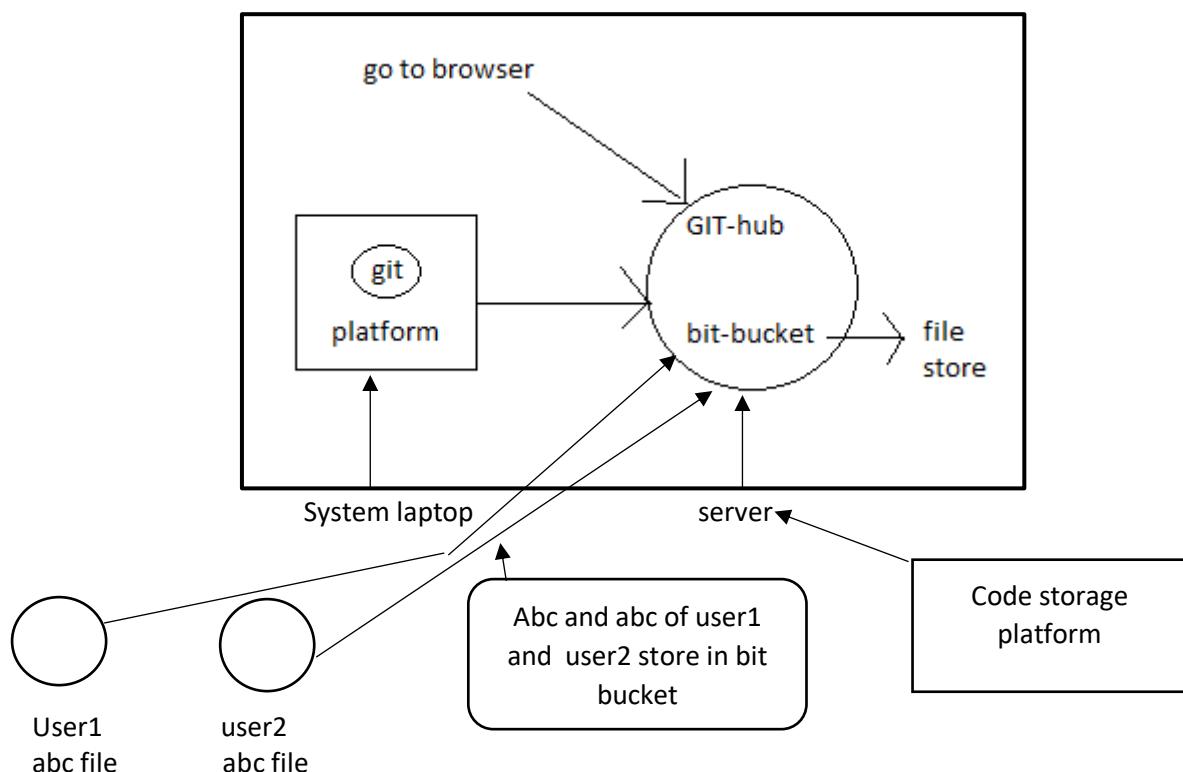
- Store on centralized server
- Git hub is code storage platform.
- It is source code management tools.
- Code pushed on [Git-hub and bit-bucket.]



Three type 3

- 1.SCM → source code management tool
- 2.VCS → version control system
- 3.RCS → Revision control system

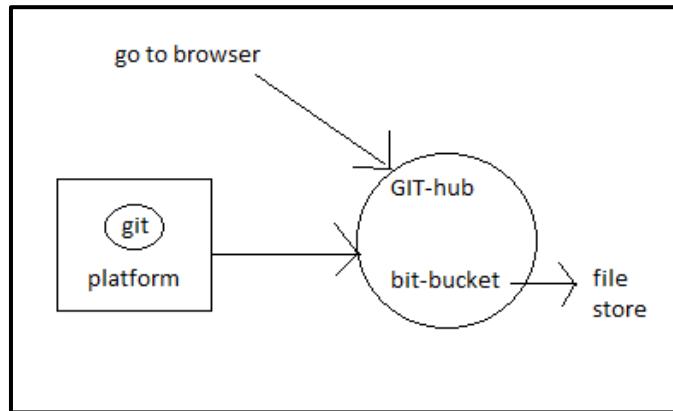
1st scenario



- 2nd store also same name abc file
- Overrite problem possible in git hub solve.

2nd scenario

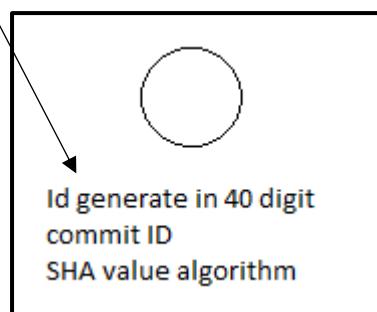
Version control system



Each File → move → commit

All developers of Each file having different ID generate after pushed on git hub or bit-bucket.

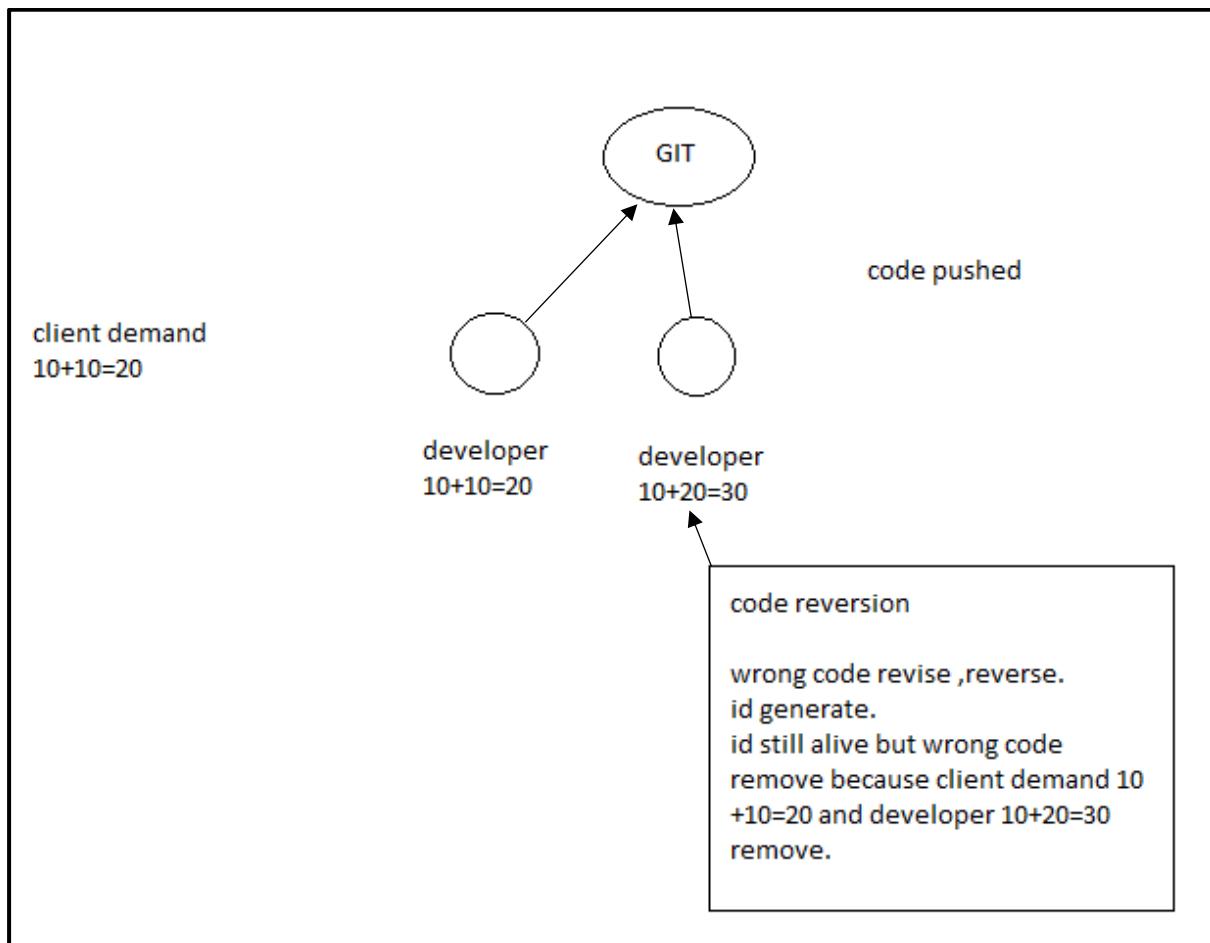
Generated id called vcs



One file → n no. of id

In vcs , access ID

RCS



"ID" - id remove called note

Deletion.backup

Tracking facility provide.

History → detail of time stamp

Know about file delete at what time

Which version control you used?

We work on git.

Git has temporary deletion with same no. of code and no. of line.

Recovery possible but deletion id must important.

Installation of git

if already install then remove it.

```
$ sudo apt-get remove git
```

```
$ cd /home
```

```
$ /home/s ls -a
```

```
kunal@ubuntu-machines:~$ sudo apt-get update
Hit:1 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:3 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:5 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
kunal@ubuntu-machines:~$ sudo apt-get install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.34.1-1ubuntu1.10).
0 upgraded, 0 newly installed, 0 to remove and 287 not upgraded.
kunal@ubuntu-machines:~$ git --version
git version 2.34.1
```

Date-8/10/23

```
kunal@ubuntu-machines:~$ git --version
git version 2.34.1
kunal@ubuntu-machines:~$ which git
/usr/bin/git
kunal@ubuntu-machines:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 22.04.2 LTS
Release:        22.04
Codename:       jammy
```

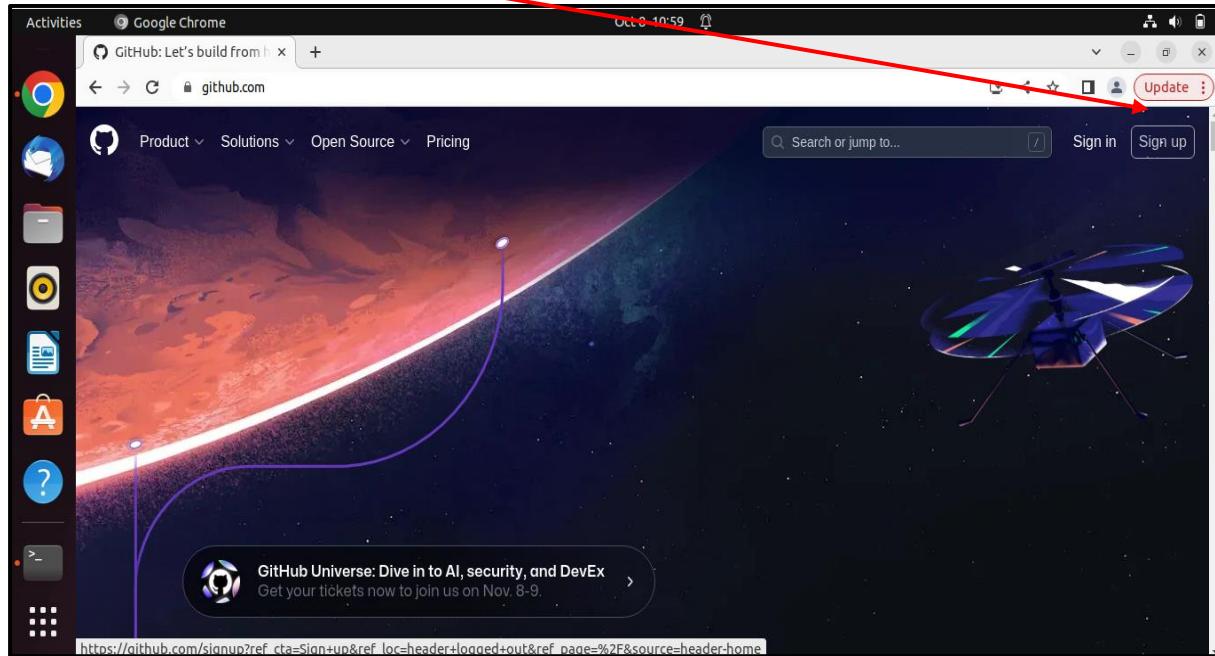
```
kunal@ubuntu-machines:~$ git config --global user.name "kunal"
kunal@ubuntu-machines:~$ git config --global user.email "kunalshiwarkar786@gmail.com"
kunal@ubuntu-machines:~$ cd /home/
kunal@ubuntu-machines:/home$ ls -a
. .. kunal
kunal@ubuntu-machines:/home$ pwd
/home
kunal@ubuntu-machines:/home$ cd
kunal@ubuntu-machines:~$ ls -a
. .config .local snap .vboxclient-draganddrop-tty2-service.pid Videos
.. Desktop Music .ssh .vboxclient-hostversion-tty2-control.pid .vim
.bash_history Documents Pictures Templates .vboxclient-seamless-tty2-control.pid .viminfo
.bash_logout Downloads .pki .vboxclient-clipboard-tty2-control.pid .vboxclient-seamless-tty2-service.pid .wget-hsts
.bashrc .gitconfig .profile .vboxclient-clipboard-tty2-service.pid .vboxclient-vmsvga-session-tty2-control.pid
.cache .gnupg Public .vboxclient-draganddrop-tty2-control.pid .vboxclient-vmsvga-session-tty2-service.pid
kunal@ubuntu-machines:~$ cat .gitconfig
[user]
  name = kunal
  email = kunalshiwarkar786@gmail.com
```

Account set up

Go to google terminal of ubuntu

1.github.com → click

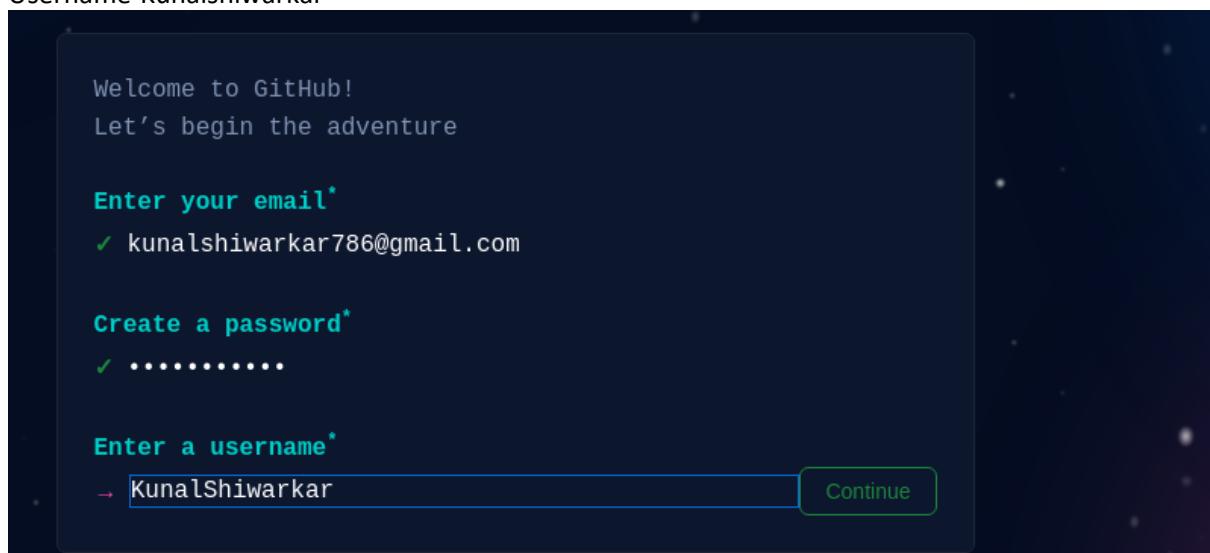
Click on sign up

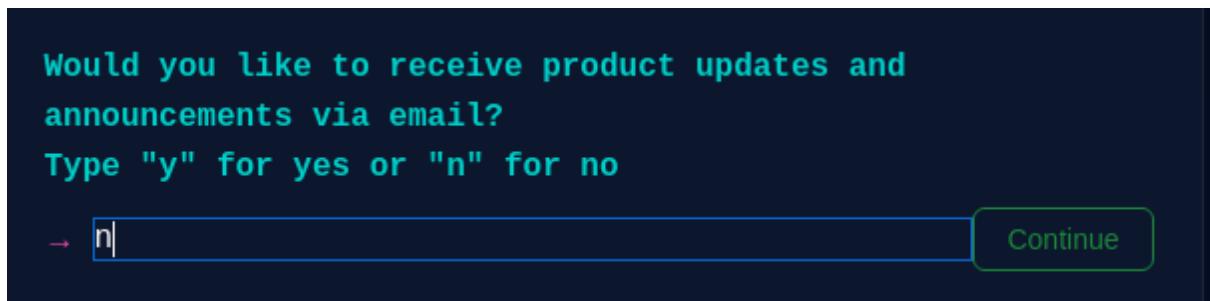


Password-Kunnu@24!!

Then click on continue

Username-Kunalshiwarkar





Verify

Account create

Otp come from gmail

Enter → just me

Student

N.A.

Collaboration

Security

Client App

Team

Click → continue

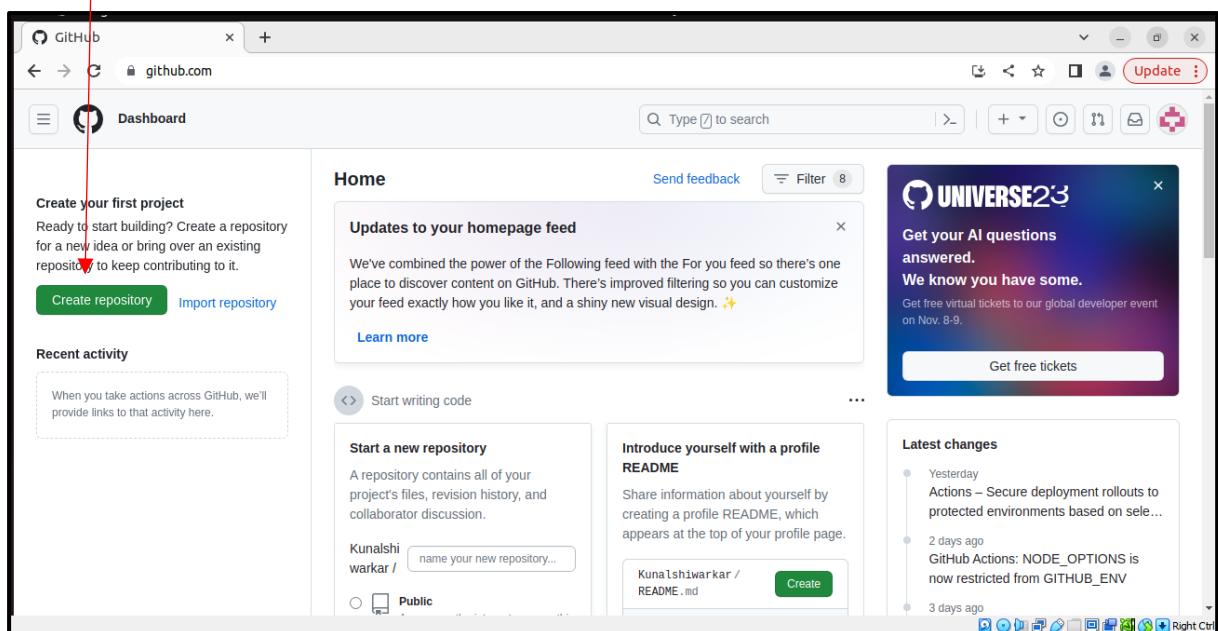
Free

Continue for free

Save

Repository create

Create repository click



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (*).

Owner *



Repository name *

/ test

test is available.

Great repository names are short and memorable. Need inspiration? How about [ideal-rotary-phone](#) ?

Description (optional)

Public

Anyone on the internet can see this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file

This is where you can write a long description for your project. [Learn more about READMEs](#).

Add .gitignore

.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

Choose a license

License: None ▾

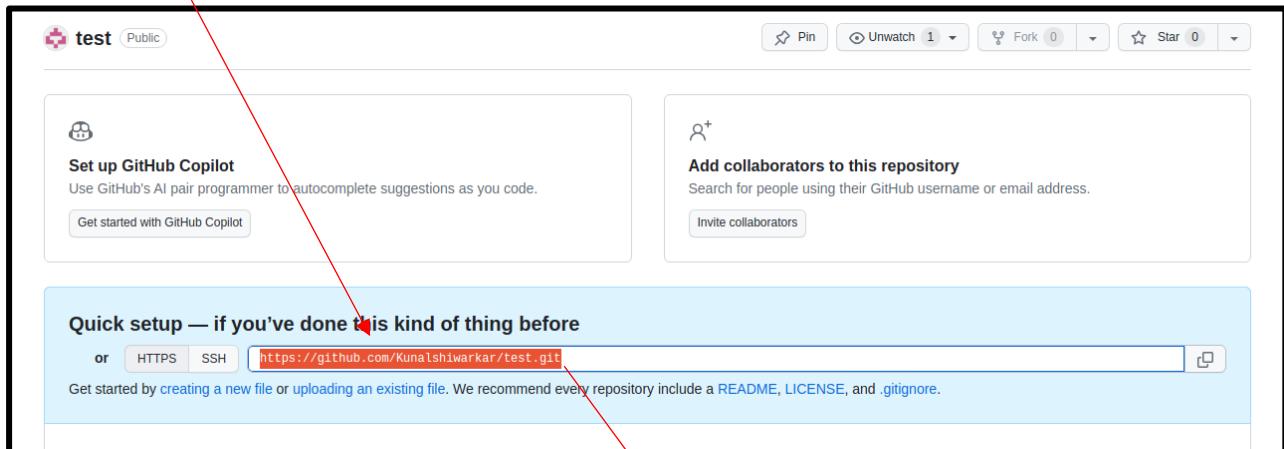
A license tells others what they can and can't do with your code. [Learn more about licenses](#).

You are creating a public repository in your personal account.

Create repository

Then click on create repository

Then copy url



Paste url

```
kunal@ubuntu-machines:~/Documents$ cd
kunal@ubuntu-machines:~$ cd Documents/
kunal@ubuntu-machines:~/Documents$ git clone https://github.com/Kunalshiwarkar/test.git
Cloning into 'test'...
warning: You appear to have cloned an empty repository.
kunal@ubuntu-machines:~/Documents$ mkdir sample
kunal@ubuntu-machines:~/Documents$ ls -a
. .. sample sample-repo test 'token file'
kunal@ubuntu-machines:~/Documents$ cd
```

```
kunal@ubuntu-machines:~$ cd Documents/test
kunal@ubuntu-machines:~/Documents/test$ ls -a
. .. .git
kunal@ubuntu-machines:~/Documents/test$ vim abc.java
kunal@ubuntu-machines:~/Documents/test$ git status
On branch main

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
kunal@ubuntu-machines:~/Documents/test$ git add
```

Source area in red colour → abc.java

```
kunal@ubuntu-machines:~/Documents/test$ git add .
kunal@ubuntu-machines:~/Documents/test$ git status
On branch main

No commits yet

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

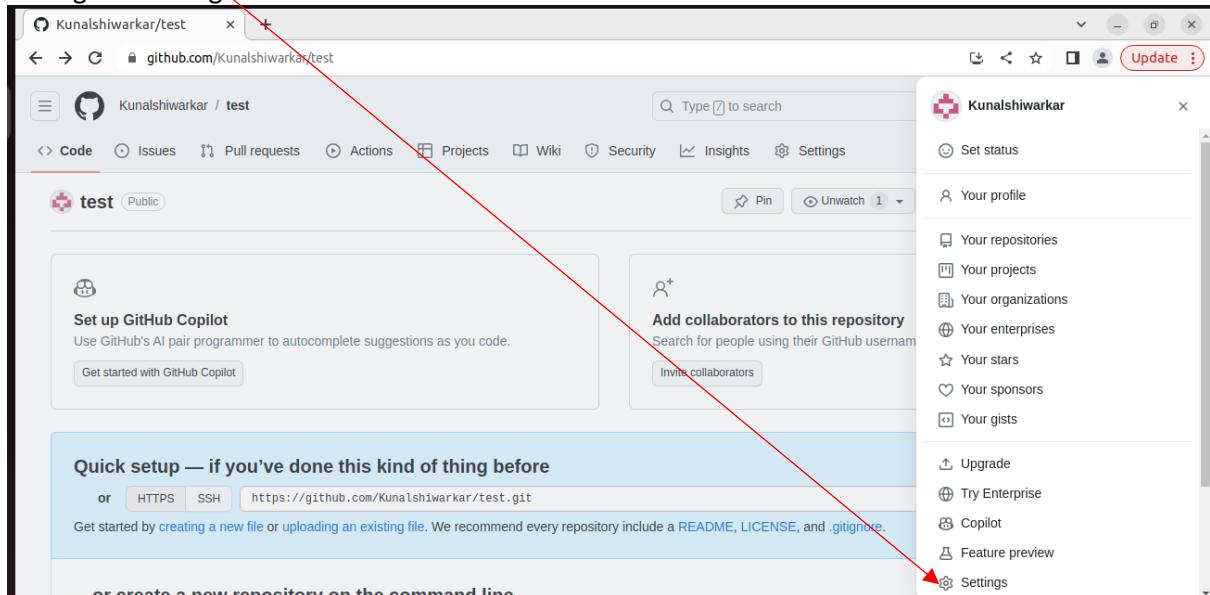
Stage area in green colour →abc.java

```
kunal@ubuntu-machines:~/Documents/test$ git config --global user.name "kunal"
kunal@ubuntu-machines:~/Documents/test$ git config --global user.email "kunalshiwarkar786@gmail.com"
```

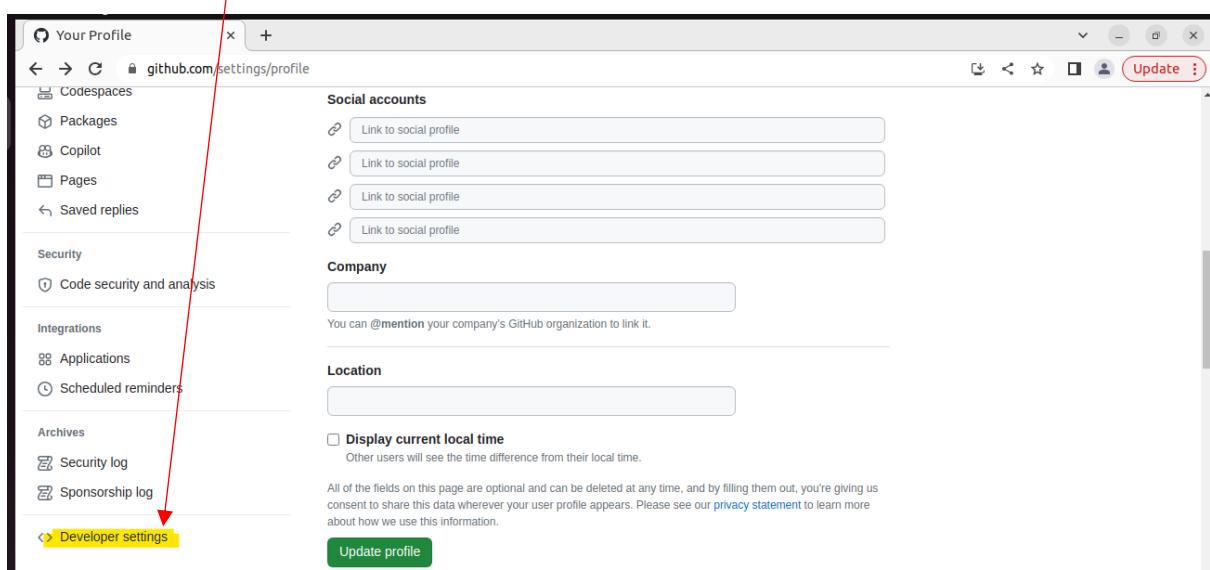
```
kunal@ubuntu-machines:~/Documents/test$ ls -a
.  ..  abc.java  .git
kunal@ubuntu-machines:~/Documents/test$ cd
kunal@ubuntu-machines:~$ ls -a
.  ..  .config  .local  snap
.  ..  Desktop  Music   .ssh
.bash_history  Documents  Pictures  Templates
.bash_logout  Downloads  .pki   .vboxclient-clipboard-tty2-control.pid
.bashrc      .gitconfig  .profile  .vboxclient-clipboard-tty2-service.pid
.cache       .gnupg    Public   .vboxclient-draganddrop-tty2-control.pid
.kunal        .vim     .viminfo .wget-hsts
kunal@ubuntu-machines:~$ cat .gitconfig
[user]
  name = kunal
  email = kunalshiwarkar786@gmail.com
[user]
  name = kunal
  email = kunalshiwarkar786@gmail.com
kunal@ubuntu-machines:~$ cd -
/home/kunal/Documents/test
kunal@ubuntu-machines:~/Documents/test$ git commit -m "file created"
[main (root-commit) c6bb391] file created
 1 file changed, 1 insertion(+)
 create mode 100644 abc.java
kunal@ubuntu-machines:~/Documents/test$ git log
commit c6bb3915c9483300cc980d46b18f73c866719fdf (HEAD -> main)
Author: kunal <kunalshiwarkar786@gmail.com>
Date:  Sun Oct 8 13:11:50 2023 +0530

  file created
```

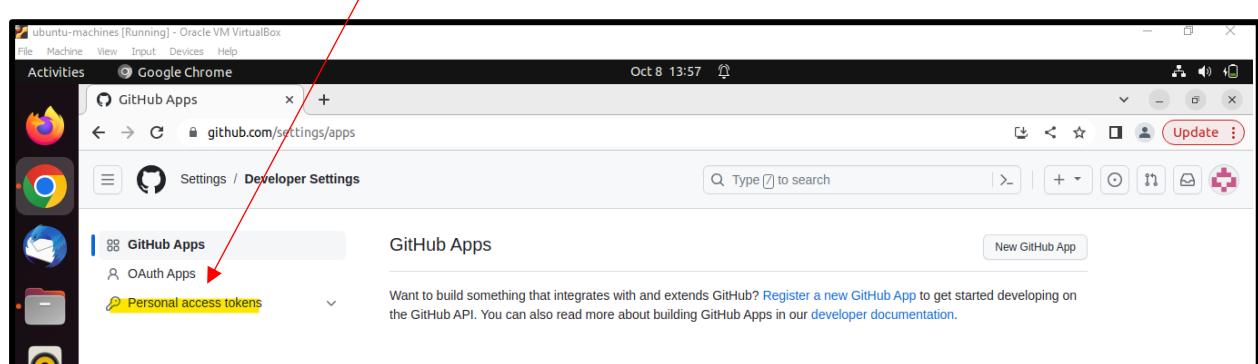
Then go to setting



Then click Developer Settings



Then click on personal access tokens



Then click on token classic

A screenshot of the GitHub Developer Settings interface. On the left, there's a sidebar with options: GitHub Apps, OAuth Apps, Personal access tokens (with a 'Beta' badge), Fine-grained tokens, and Tokens (classic). A red arrow points from the text above to the 'Tokens (classic)' link. The main content area is titled 'Personal access tokens (classic)' and contains a button labeled 'Generate new token'. Below the button, there's a note about API tokens for scripts or testing and a link to generate a personal access token.

Then click on generate new token classic

A screenshot of the GitHub Developer Settings interface, similar to the previous one but with a different focus. A red arrow points from the text above to the 'Generate new token' button. A dropdown menu is open, showing two options: 'Generate new token (Beta)' (described as 'Fine-grained, repo-scoped') and 'Generate new token (classic)' (described as 'For general use'). The 'Generate new token (classic)' option is highlighted with a yellow box and a red arrow pointing to it.

Name- **kunal-key**

No expiration

A screenshot of the 'New personal access token (classic)' creation form. The left sidebar shows the same navigation as before. The main form has a title 'New personal access token (classic)'. It includes a note about token usage, a 'Note' field containing 'kunal-key', and an 'Expiration *' dropdown set to 'No expiration' (with a note that the token will never expire). A yellow callout box at the bottom left of the form area says 'GitHub strongly recommends that you set an expiration date for your token to help keep your information secure.' and links to 'Learn more'. At the bottom, there's a section for 'Select scopes' with two options: 'repo' (selected) and 'repo:status'.

The screenshot shows the GitHub 'Personal access tokens (classic)' page. On the left, there's a sidebar with 'GitHub Apps', 'OAuth Apps', and 'Personal access tokens' (which is expanded, showing 'Fine-grained tokens' and 'Tokens (classic)'). A red arrow points from the text 'Copy or click on that url and paste into password here' to the token 'ghp_6A1y1VLJiMYVRGGsioueNbWRK72Ljz2dThUw'. At the top right, there are 'Generate new token' and 'Revoke all' buttons.

Copy or click on that url and paste into password here

```
kunal@ubuntu-machines:~/Documents/test$ git push
Username for 'https://github.com': Kunalshiwarkar
Password for 'https://Kunalshiwarkar@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 224 bytes | 224.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Kunalshiwarkar/test.git
 * [new branch]      main -> main
```

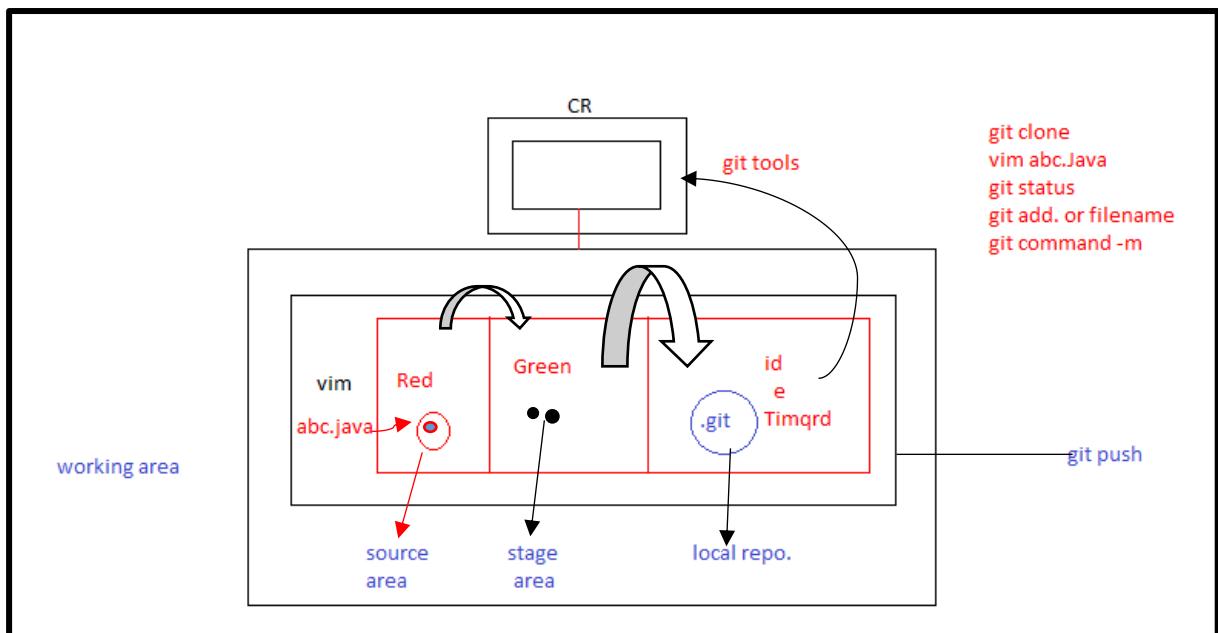
After , click on test.

The screenshot shows a GitHub repository named 'test'. The repository is public. It contains one commit by 'Java' updated 2 hours ago. The commit message is 'Java'.

Click on abc.java file

The screenshot shows the details of the 'abc.java' file commit. The commit was made by 'Java' 3 hours ago. The commit message is 'abc.java file created'. The file was created 3 hours ago. There is a note at the bottom encouraging the addition of a README file.

Screenshot of a GitHub repository page showing a file named abc.java. The code contains a single line: "1 hello kunal". The GitHub interface includes navigation links like Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings.



Note- git config - -global user.name "Kunalshiwarkar" and
git config - - global user.email "kunalshiwarkar786@gmail.com"
used one time because after execution this git config command name and email store in
.gitconfig".
So don,t need of .gitconfig command again.

Q.what is difference between tracked and untracked files?

Ans- tracked files→after git add . Command and see git status the file is in green colour means file tracked because file move frm source area to stage area.

Untracked files→ after file creation show git status it shows file is in red colour means file in source area and untracked .

git pull

If we want to download latest data from existing repository , when other developer working on that existing repo.

Pull downloaded file stored in working area.

git fitch

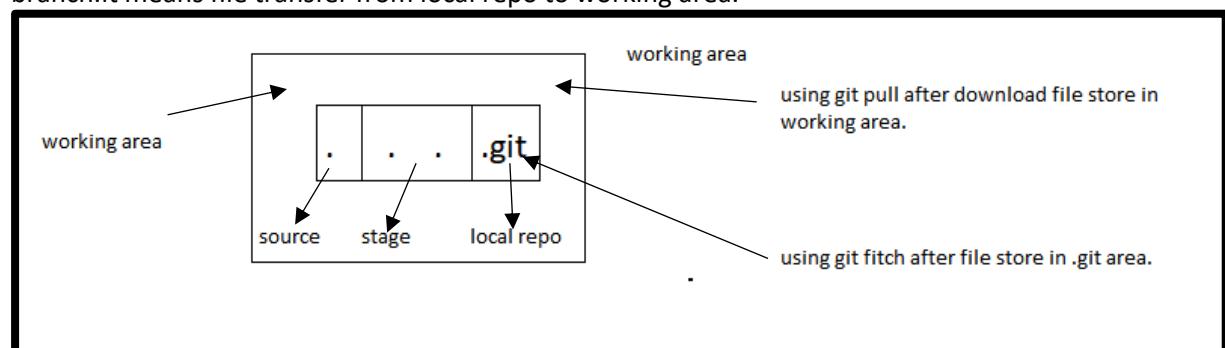
When used?

If we want to just download then used git fetch command.

This file stored in .git file means local repository.

git merge

Git merge is a version control operation that combines changes from one branch into another branch.it means file transfer from local repo to working area.



File created in test repository

```
kunal@ubuntu-machines:~$ cd Documents/
kunal@ubuntu-machines:~/Documents$ cd test
kunal@ubuntu-machines:~/Documents/test$ vim test2
kunal@ubuntu-machines:~/Documents/test$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    test2

nothing added to commit but untracked files present (use "git add" to track)
kunal@ubuntu-machines:~/Documents/test$ git init
Reinitialized existing Git repository in /home/kunal/Documents/test/.git/
kunal@ubuntu-machines:~/Documents/test$ git add .
kunal@ubuntu-machines:~/Documents/test$ git commit -m "done"
[main 157abb4] done
 1 file changed, 1 insertion(+)
 create mode 100644 test2
kunal@ubuntu-machines:~/Documents/test$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

```
kunal@ubuntu-machines:~/Documents/test$ git push
Username for 'https://github.com': Kunalshiwarkar
Password for 'https://Kunalshiwarkar@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 276 bytes | 92.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Kunalshiwarkar/test.git
  c6bb391..157abb4 main -> main
kunal@ubuntu-machines:~/Documents/test$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

test (Public)

main 1 branch 0 tags

Kunalshiwarkar done 157abb4 12 minutes ago 2 commits

abc.java file created 4 days ago

test2 done 12 minutes ago

Help people interested in this repository understand your project by adding a README.

Add a README

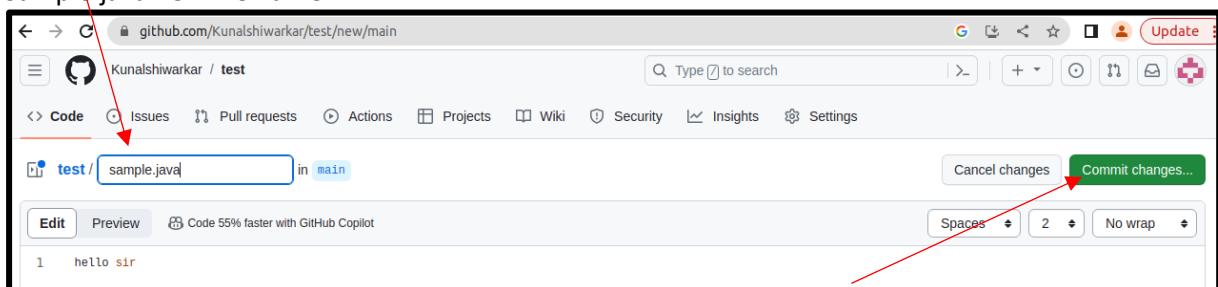
1.Git pull and Git push

Type 1 → using git pull download file from existing repository

```
kunal@ubuntu-machines:~$ cd Documents/
kunal@ubuntu-machines:~/Documents$ cd test
kunal@ubuntu-machines:~/Documents/test$ ls -a
. .. abc.java .git test2
kunal@ubuntu-machines:~/Documents/test$ git status
On branch main
Your branch is up to date with 'origin/main'.

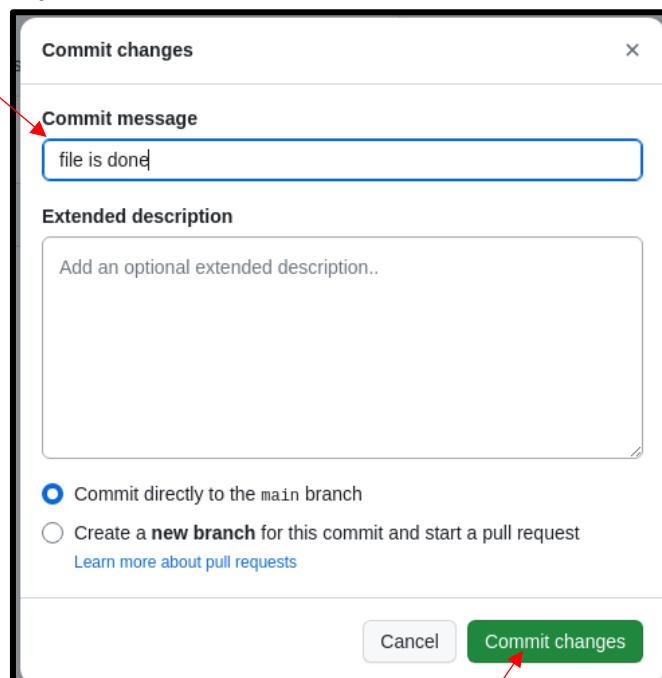
nothing to commit, working tree clean
```

Now, click on add file then click on create a new file in test repository using git hub to36
Sample.java new file name .



Then click on commit changes

Write message in commit



Then click on commit changes

Name	Last commit message	Last commit date
abc.java	file created	4 days ago
sample.java	file is done	now
test2	done	35 minutes ago

```
kunal@ubuntu-machines:~/Documents/test$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 689 bytes | 689.00 KiB/s, done.
From https://github.com/KunalShiwarkar/test
  157abb4..aaa69eb main      -> origin/main
Updating 157abb4..aaa69eb
Fast-forward
 sample.java | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 sample.java
kunal@ubuntu-machines:~/Documents/test$ git status
On branch main
Your branch is up to date with 'origin/main'.

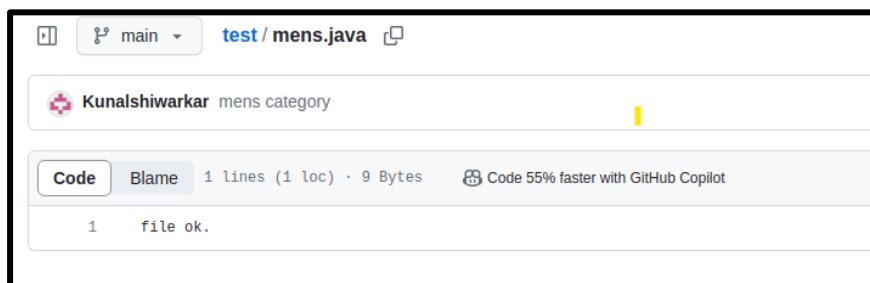
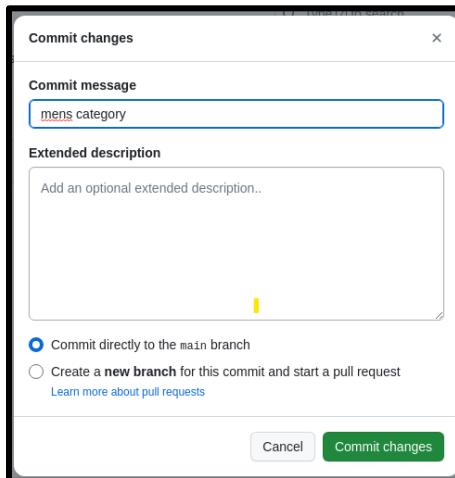
nothing to commit, working tree clean
kunal@ubuntu-machines:~/Documents/test$ ls -a
.  ..  abc.java .git sample.java test2
```

Sample.java in working area using git pull

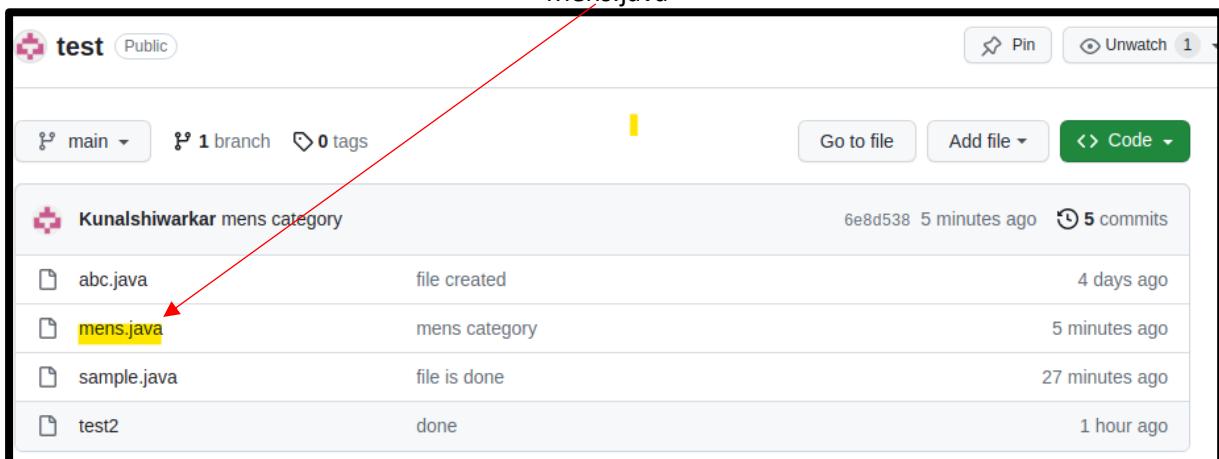
2.Git fetch

Type 2→just download file in local repository i.e “.git” by using git fetch

Create new file mens.java



Mens.java



File store in .git

```
kunal@ubuntu-machines:~/Documents/test$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
kunal@ubuntu-machines:~/Documents/test$ ls -a
. .. abc.java .git sample.java test2
kunal@ubuntu-machines:~/Documents/test$ git fetch
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), 1.29 KiB | 330.00 KiB/s, done.
From https://github.com/KunalShiwarkar/test
  aaa69eb..6e8d538  main      -> origin/main
kunal@ubuntu-machines:~/Documents/test$ ls -a
. .. abc.java .git sample.java test2
kunal@ubuntu-machines:~/Documents/test$ git status
On branch main
Your branch is behind 'origin/main' by 2 commits, and can be fast-forwarded.
(use "git pull" to update your local branch)

nothing to commit, working tree clean
```

3. Transferring to Working area using merge.

local repo i.e. .git into working area

```
kunal@ubuntu-machines:~/Documents/test$ git merge
Updating aaa69eb..6e8d538
Fast-forward
  mens.java | 1 +
  1 file changed, 1 insertion(+)
  create mode 100644 mens.java
kunal@ubuntu-machines:~/Documents/test$ ls -a
. .. abc.java .git mens.java sample.java test2
kunal@ubuntu-machines:~/Documents/test$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

4.create file in test repository i.e shirts.java and create jeans.java in git hub centralized location

```
kunal@ubuntu-machines:~/Documents/test$ vim shirts.java
kunal@ubuntu-machines:~/Documents/test$ git status
On branch main
Your branch is up to date with 'origin/main'.

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

nothing added to commit but untracked files present (use "git add" to track)
kunal@ubuntu-machines:~/Documents/test$ git add .
kunal@ubuntu-machines:~/Documents/test$ git status
On branch main
Your branch is up to date with 'origin/main'.

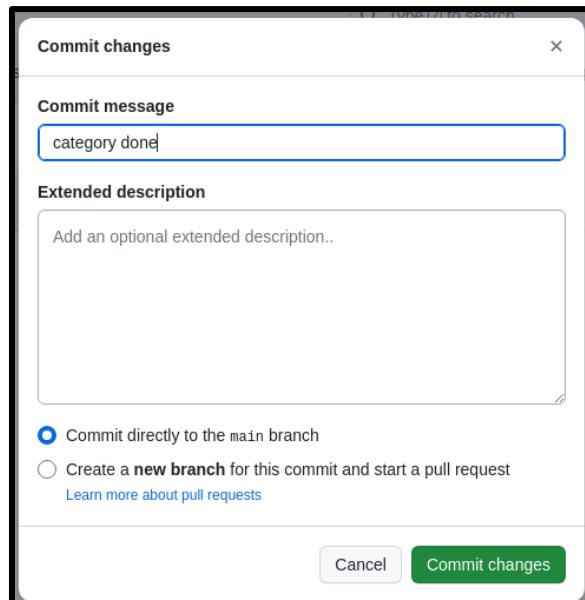
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   shirts.java

kunal@ubuntu-machines:~/Documents/test$ git commit -m "sample"
[main 647d617] sample
 1 file changed, 1 insertion(+)
  create mode 100644 shirts.java
kunal@ubuntu-machines:~/Documents/test$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)
```

1. Git fetch

Eg. Create new file jeans.java





```
kunal@ubuntu-machines:~/Documents/test$ git fetch
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), 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), 650 bytes | 43.00 KiB/s, done.
From https://github.com/Kunalshiwarkar/test
  6e8d538..141659a main      -> origin/main
kunal@ubuntu-machines:~/Documents/test$ ls -a
. .. abc.java .git mens.java sample.java shirts.java test2
```

After git fetching file stored in .git i.e. jeans.java

Then we used git merge command

```
kunal@ubuntu-machines:~/Documents/test$ git merge
Merge made by the 'ort' strategy.
 jeans.java | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 jeans.java
```

After git merge enter this message show drag the msg and then press ctrl+x

```
GNU nano 6.2
Merge remote-tracking branch 'refs/remotes/origin/main'
# Please enter a commit message to explain why this merge is necessary,
# especially if it merges an updated upstream into a topic branch.
#
# Lines starting with '#' will be ignored, and an empty message aborts
# the commit.
/home/kunal/Documents/test/.git/MERGE_MSG
```

Then merge made by the 'ort' strategy. <-- msg show

Jeans.java from .git into working area

```
kunal@ubuntu-machines:~/Documents/test$ ls -a
.  .. abc.java  .git jeans.java  mens.java  sample.java  shirts.java  test2
kunal@ubuntu-machines:~/Documents/test$ git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)
```

Then git push

```
kunal@ubuntu-machines:~/Documents/test$ git push
Username for 'https://github.com': Kunalshiwarkar
Password for 'https://Kunalshiwarkar@github.com':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 531 bytes | 531.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/Kunalshiwarkar/test.git
  141659a..2691aef  main -> main
```

```
kunal@ubuntu-machines:~/Documents/test$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Kunalshiwarkar Merge remote-tracking branch 'refs/remotes/origin/main'		
	2691aef 43 minutes ago	8 commits
abc.java	file created	4 days ago
jeans.java	category done	44 minutes ago
mens.java	mens category	1 hour ago
sample.java	file is done	1 hour ago
shirts.java	sample	53 minutes ago
test2	done	2 hours ago

5.Modify existing user, how to skip stage area

- Modify sample.java file using vim <-- file is already present so we modify file sample.java
- We check status , the file is modified and in source area which is in red colour.
- Changes commit → use git add <filename> to update what will be committed.

```
kunal@ubuntu-machines:~/Documents/test$ vim sample.java
kunal@ubuntu-machines:~/Documents/test$ git status
On branch main
Your branch is up to date with 'origin/main'.

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:   sample.java
```

- We create new file cycle.java in test repository , this file in source area.

```
kunal@ubuntu-machines:~/Documents/test$ vim cycle.java
kunal@ubuntu-machines:~/Documents/test$ git status
On branch main
Your branch is up to date with 'origin/main'.

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:   sample.java

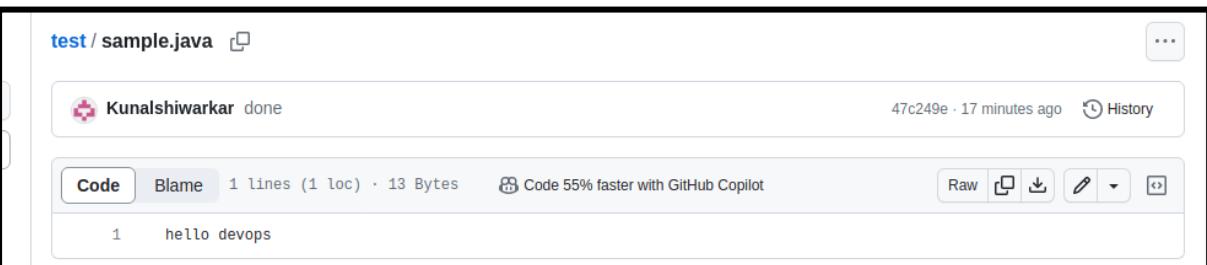
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    cycle.java

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

- After git add one is modify and other file is created for push i.e. cycle.java

```
kunal@ubuntu-machines:~/Documents/test$ git add .
kunal@ubuntu-machines:~/Documents/test$ git commit -m "done"
[main 47c249e] done
 2 files changed, 2 insertions(+), 1 deletion(-)
  create mode 100644 cycle.java
kunal@ubuntu-machines:~/Documents/test$ git push
Username for 'https://github.com': Kunalshiwarkar
Password for 'https://Kunalshiwarkar@github.com':
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 326 bytes | 326.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Kunalshiwarkar/test.git
  2691aef..47c249e main -> main
kunal@ubuntu-machines:~/Documents/test$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```



- Only modify area skip stage area

```
kunal@ubuntu-machines:~/Documents/test$ git add .
kunal@ubuntu-machines:~/Documents/test$ git commit -am "sample file"
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
kunal@ubuntu-machines:~/Documents/test$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
kunal@ubuntu-machines:~/Documents/test$ ls -a
.: .. abc.java cycle.java .git jeans.java mens.java sample.java shirts.java test2
```

Don't used git fetch command in this situation.

Git init

Init means initialization

What is init?

To create local empty repository using init .

After init .git show using ls -a.

Using git.init we can convert normal folder into local folder using git init.

```
kunal@ubuntu-machines:~$ cd Documents/
kunal@ubuntu-machines:~/Documents$ mkdir Mens← creating folder
kunal@ubuntu-machines:~/Documents$ cd Mens/
kunal@ubuntu-machines:~/Documents/Mens$ ls -a
. .. ← .git option not show
kunal@ubuntu-machines:~/Documents/Mens$ git status← git command not work
fatal: not a git repository (or any of the parent directories): .git
kunal@ubuntu-machines:~/Documents/Mens$ git init← this command used for git
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/kunal/Documents/Mens/.git/
kunal@ubuntu-machines:~/Documents/Mens$ ls -a ← after git init normal folder convert
. .. .git ← into local folder and ".git" show
kunal@ubuntu-machines:~/Documents/Mens$ cd ..
```

```
kunal@ubuntu-machines:~/Documents$ mkdir Womens
kunal@ubuntu-machines:~/Documents$ git init Womens/ ← making folder convert into .git by
using git init.
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/kunal/Documents/Womens/.git/
kunal@ubuntu-machines:~/Documents$ cd Womens/
kunal@ubuntu-machines:~/Documents/Womens$ ls -a
. .. .git
kunal@ubuntu-machines:~/Documents/Womens$ cd ..
```

```

kunal@ubuntu-machines:~/Documents$ cd Mens/
kunal@ubuntu-machines:~/Documents/Mens$ ls -a
. .. .git
kunal@ubuntu-machines:~/Documents/Mens$ cd .git← inside git 3 files i.e config,description and HEAD
kunal@ubuntu-machines:~/Documents/Mens/.git$ ls -a
. .. branches config description HEAD hooks info objects refs
kunal@ubuntu-machines:~/Documents/Mens/.git$ cd ..
kunal@ubuntu-machines:~/Documents/Mens$ ls -a
. .. .git
kunal@ubuntu-machines:~/Documents/Mens$ vim D1.java
kunal@ubuntu-machines:~/Documents/Mens$ vim D2.java
kunal@ubuntu-machines:~/Documents/Mens$ ls -a
. .. D1.java D2.java .git
kunal@ubuntu-machines:~/Documents/Mens$ git status
On branch master

No commits yet

Untracked files:
(use "git add <file>..." to include in what will be committed)
  D1.java ← file in source area.
  D2.java

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

```

```

kunal@ubuntu-machines:~/Documents/Mens$ git add .
kunal@ubuntu-machines:~/Documents/Mens$ git status
On branch master

No commits yet

Changes to be committed:
(use "git rm --cached <file>..." to unstage)
  new file: D1.java ← now file in stage group.
  new file: D2.java

kunal@ubuntu-machines:~/Documents/Mens$ git commit -m "init repository"
[master (root-commit) ee9a25b] init repository
 2 files changed, 2 insertions(+)
  create mode 100644 D1.java
  create mode 100644 D2.java
kunal@ubuntu-machines:~/Documents/Mens$ git status
On branch master
nothing to commit, working tree clean
kunal@ubuntu-machines:~/Documents/Mens$ git log
commit ee9a25bff195195d8ffc0c012f2285fe23342d9f (HEAD -> master)
Author: Kunalshiwarkar <Kunalshiwarkar786@gmail.com>
Date:   Fri Oct 13 16:24:20 2023 +0530

  init repository

```

Go to git hub account and create new repo Mens ← repo name is same as folder name “Mens”
Take url.

Syntax \$ git remote <add repo> <url paste>
\$ git push <repo name>

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (*).

Owner *



Repository name *

Mens

Mens is available.

Great repository names are short and memorable. Need inspiration? How about [vigilant-telegram](#) ?

Description (optional)

Public

Anyone on the internet can see this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

copy this url and paste after repo name

Quick setup — if you've done this kind of thing before

or

HTTPS

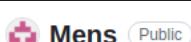
SSH

<https://github.com/Kunalshiwarkar/Mens.git>



Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

```
kunal@ubuntu-machines:~/Documents/Mens$ git remote add Mens https://github.com/Kunalshiwarkar/Mens.git
kunal@ubuntu-machines:~/Documents/Mens$ git remote -v
Mens    https://github.com/Kunalshiwarkar/Mens.git (fetch)
Mens    https://github.com/Kunalshiwarkar/Mens.git (push)
kunal@ubuntu-machines:~/Documents/Mens$ git push Mens
Username for 'https://github.com': Kunalshiwarkar
Password for 'https://Kunalshiwarkar@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 227 bytes | 227.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Kunalshiwarkar/Mens.git
 * [new branch]      master -> master
```



Mens

Public



Unwatch 1

master

1 branch

0 tags

Go to file

Add file

Code

Kunalshiwarkar init repository

83d81c2 14 minutes ago 1 commit

D1.java

init repository

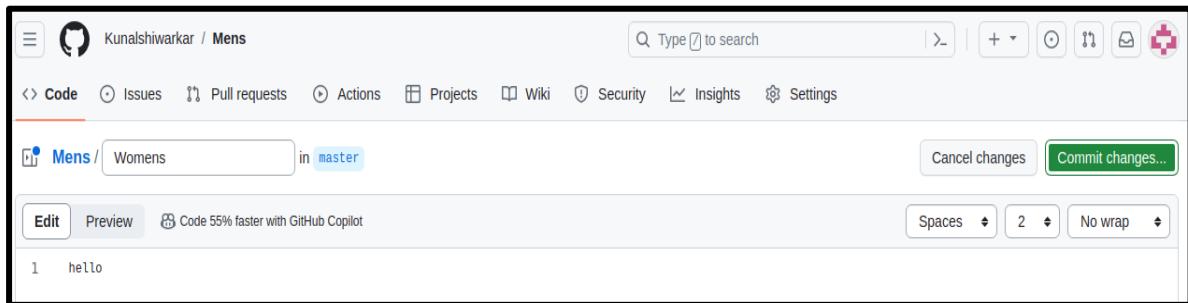
14 minutes ago

D2.java

init repository

14 minutes ago

Now file created in Mens git hub account



```
kunal@ubuntu-machines:~/Documents/Mens$ git pull  
There is no tracking information for the current branch.  
Please specify which branch you want to merge with.  
See git-pull(1) for details.
```

`git pull <remote> <branch>` used to download file from git hub account.

If you wish to set tracking information for this branch you can do so with:

```
git branch --set-upstream-to=Mens/<branch> master
```

```
kunal@ubuntu-machines:~/Documents/Mens$ git pull Mens master  
remote: Enumerating objects: 4, done.  
remote: Counting objects: 100% (4/4), done.  
remote: Compressing objects: 100% (2/2), done.  
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0  
Unpacking objects: 100% (3/3), 661 bytes | 661.00 KiB/s, done.  
From https://github.com/KunalShiwarkar/Mens  
 * branch master      -> FETCH_HEAD  
   83d81c2..5b4ec54  master      -> Mens/master  
Updating 83d81c2..5b4ec54  
Fast-forward  
 Womens | 1 +  
 1 file changed, 1 insertion(+)  
 create mode 100644 Womens  
kunal@ubuntu-machines:~/Documents/Mens$ ls -a  
 . .. D1.java D2.java .git Womens  
kunal@ubuntu-machines:~/Documents/Mens$ git status  
On branch master  
nothing to commit, working tree clean
```

Date -15/10/2023

How to check last to commit log? History?

Commit id ,meta,data store in git hub .

Even user history send on git hub.

Options:

- git log <filename>
- git log → complete history of log show history show
Repository history only commit structure
Show commit id time stamp
- git log -n2
- git log --author="swapnil"
- git log --since="2016/12/13"
- git log --until="2016/12/13"
- git log --grep="DB code"
- git log --oneline
(if we want to see logs on high level that time we can use this command ie. only commit ids and msg)

#Qn.

Show me all logs committed by user "Ram" AND has a commit message "math function" in it.

```
$ git log --author="Ram" --grep="commit msg" -n2
```

```
kunal@kunalsh:~$ cd Documents/
kunal@kunalsh:~/Documents$ cd test/
kunal@kunalsh:~/Documents/test$ ls -a
. abc.java .git mens.java shirts.java
.. cycle.java jeans.java sample.java test2
kunal@kunalsh:~/Documents/test$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
kunal@kunalsh:~/Documents/test$ git log → complete history of log show
commit 47c249e166600e6410053fbac9b3ad418d8b3282 (HEAD -> main, origin/main, origin/HEAD)
Author: Kunalshiwarkar <kunalshiwarkar786@gmail.com>
Date:   Thu Oct 12 16:54:58 2023 +0530

done

commit 2691aefe68ba09565806cf2d486179936475b7ea
Merge: 647d617 141659a
Author: Kunalshiwarkar <kunalshiwarkar786@gmail.com>
```

```
kunal@kunalsh:~/Documents/test$ ls -a
. .. abc.java cycle.java .git jeans.java mens.java sample.java shirts.java test2
kunal@kunalsh:~/Documents/test$ git log abc.java → specific history of file
commit c6bb3915c9483300cc980da6b18f73c866719fdf
Author: kunal <kunalshiwarkar786@gmail.com>
Date:   Sun Oct 8 13:11:50 2023 +0530

file created
kunal@kunalsh:~/Documents/test$ git log -n2 → if we want to see latest 2 commit
commit 47c249e166600e6410053fbac9b3ad418d8b3282 (HEAD -> main, origin/main, origin/HEAD)
Author: Kunalshiwarkar <kunalshiwarkar786@gmail.com>
Date:   Thu Oct 12 16:54:58 2023 +0530

done

commit 2691aefe68ba09565806cf2d486179936475b7ea
Merge: 647d617 141659a
Author: Kunalshiwarkar <kunalshiwarkar786@gmail.com>
Date:   Thu Oct 12 14:50:57 2023 +0530

Merge remote-tracking branch 'refs/remotes/origin/main'
kunal@kunalsh:~/Documents/test$ git log --author="Kunalshiwarkar" → author show the commit
commit 47c249e166600e6410053fbac9b3ad418d8b3282 (HEAD -> main, origin/main, origin/HEAD)
```

```
kunal@kunalsh:~/Documents/test$ git log --author="Kunalshiwarkar" -n3
commit 47c249e166600e6410053fbac9b3ad418d8b3282 (HEAD -> main, origin/main, origin/HEAD)
Author: Kunalshiwarkar <kunalshiwarkar786@gmail.com>
Date: Thu Oct 12 16:54:58 2023 +0530

done

commit 2691aefe68ba09565806cf2d486179936475b7ea
Merge: 647d617 141659a
Author: Kunalshiwarkar <kunalshiwarkar786@gmail.com>
Date: Thu Oct 12 14:50:57 2023 +0530

    Merge remote-tracking branch 'refs/remotes/origin/main'

commit 141659ade883d405220e8fba4915fd304c17d745
Author: Kunalshiwarkar <147299707+Kunalshiwarkar@users.noreply.github.com>
Date: Thu Oct 12 14:49:57 2023 +0530

category done
```

```
kunal@kunalsh:~/Documents/test$ git log --grep=sample.java
kunal@kunalsh:~/Documents/test$ git log --author="Kunalshiwarkar" --grep=test
kunal@kunalsh:~/Documents/test$ git log --oneline
47c249e (HEAD -> main, origin/main, origin/HEAD) done
2691aef Merge remote-tracking branch 'refs/remotes/origin/main'
141659a category done
647d617 sample
6e8d538 mens category
b9e7a9a mens category
aaa69eb file is done
157abb4 done
c6bb391 file created
```

Git log -- until and git log --since are latest problem in github.

```
kunal@kunalsh:~/Documents/test$ git log --until=2023/10/15
commit 47c249e166600e6410053fbac9b3ad418d8b3282 (HEAD -> main, origin/main, origin/HEAD)
Author: Kunalshiwarkar <kunalshiwarkar786@gmail.com>
Date: Thu Oct 12 16:54:58 2023 +0530

done

commit 2691aefe68ba09565806cf2d486179936475b7ea
Merge: 647d617 141659a
Author: Kunalshiwarkar <kunalshiwarkar786@gmail.com>
Date: Thu Oct 12 14:50:57 2023 +0530

    Merge remote-tracking branch 'refs/remotes/origin/main'

commit 141659ade883d405220e8fba4915fd304c17d745
Author: Kunalshiwarkar <147299707+Kunalshiwarkar@users.noreply.github.com>
Date: Thu Oct 12 14:49:57 2023 +0530

category done
```

```
kunal@kunalsh:~/Documents/test$ git log --since=2023/10/8
commit 47c249e166600e6410053fbac9b3ad418d8b3282 (HEAD -> main, origin/main, origin/HEAD)
Author: Kunalshiwarkar <kunalshiwarkar786@gmail.com>
Date: Thu Oct 12 16:54:58 2023 +0530

done

commit 2691aefe68ba09565806cf2d486179936475b7ea
Merge: 647d617 141659a
Author: Kunalshiwarkar <kunalshiwarkar786@gmail.com>
Date: Thu Oct 12 14:50:57 2023 +0530

    Merge remote-tracking branch 'refs/remotes/origin/main'

commit 141659ade883d405220e8fba4915fd304c17d745
Author: Kunalshiwarkar <147299707+Kunalshiwarkar@users.noreply.github.com>
Date: Thu Oct 12 14:49:57 2023 +0530

category done
```

If I modified the files

```
kunal@kunalsh:~/Documents/test$ vim jeans.java →modified the jeans.java which is existing
kunal@kunalsh:~/Documents/test$ git status
On branch main
Your branch is up to date with 'origin/main'.

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:   jeans.java →we modified this file so it is in source area.

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

What I modified i.e. add in this file and all command show the status of modified files.

1. It works on source area using git diff

syntax

```
$ git status
$ git diff <filename>
$ git diff -staged <filename>
$ git diff 089871935..fc894a6bf6d2e (if we want to see difference between two files)(use commit
ids).
```

```
kunal@kunalsh:~/Documents/test$ git diff jeans.java →work on source area
diff --git a/jeans.java b/jeans.java
index 0b5d6ed..f467597 100644
--- a/jeans.java
+++ b/jeans.java
@@ -1 +1,6 @@
-hi
+test
+hello
+hi
+welcome
+to devops
```

+ sign show add message in modified
message

```
kunal@kunalsh:~/Documents/test$ vim cycle.java →modified existing file
kunal@kunalsh:~/Documents/test$ git status
On branch main
Your branch is up to date with 'origin/main'.

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:   cycle.java
    modified:   jeans.java →modified files into source area.

no changes added to commit (use "git add" and/or "git commit -a")
kunal@kunalsh:~/Documents/test$ git diff cycle.java
diff --git a/cycle.java b/cycle.java
index 46afab7..3781319 100644
--- a/cycle.java
+++ b/cycle.java
@@ -1 +1 @@
-Hello morning
+god fever
```

- sign show exact remove message and + sign
show add message in file

```

kunal@kunalsh:~/Documents/test$ git add .
kunal@kunalsh:~/Documents/test$ git diff --staged --stat
--staged --stat
modified file add into stage area
from source area i.e. cycle.java

kunal@kunalsh:~/Documents/test$ git diff --staged cycle.java
diff --git a/cycle.java b/cycle.java
index 46afab7..3781319 100644
--- a/cycle.java
+++ b/cycle.java
@@ -1 +1 @@
-Hello morning
+god fever
check the difference
add file name

kunal@kunalsh:~/Documents/test$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
(use "git restore --staged <file>..." to unstage)
    modified:   cycle.java
    modified:   jeans.java      stage area

```

If difference .git check in modified file

```

kunal@kunalsh:~/Documents/test$ git commit -m "diff file"
[main 92d89ba] diff file
Commit message: kunal@kunalsh:~/Documents/test$ git commit -m "diff file"

```

```

kunal@kunalsh:~/Documents/test$ git log
commit 92d89ba5313a89bdd2d6a23240cc5a59e7ecd (HEAD -> main)
Author: kunal <kunal@kunalsh.myguest.virtualbox.org>
Date:  Sun Oct 15 19:44:23 2023 +0530

    diff file

commit 47c249e166600e6410053fbac9b3ad418d8b3282 (origin/main, origin/HEAD)
Author: Kunalshiwarkar <kunalshiwarkar786@gmail.com>
Date:  Thu Oct 12 16:54:58 2023 +0530

done

```

```

kunal@kunalsh:~/Documents/test$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

```

```

kunal@kunalsh:~/Documents/test$ git log -n2
commit 92d89ba5313a89bdd2d6a23240cc5a59e7ecd (HEAD -> main)
Author: kunal <kunal@kunalsh.myguest.virtualbox.org>
Date:  Sun Oct 15 19:44:23 2023 +0530

    diff file

commit 47c249e166600e6410053fbac9b3ad418d8b3282 (origin/main, origin/HEAD)
Author: Kunalshiwarkar <kunalshiwarkar786@gmail.com>
Date:  Thu Oct 12 16:54:58 2023 +0530

done

```

```
kunal@kunalsh:~/Documents/test$ git diff 2691aefef68ba09565806cf2d486179936475b7ea
diff --git a/cycle.java b/cycle.java
new file mode 100644
index 0000000..3781319
--- /dev/null
+++ b/cycle.java
@@ -0,0 +1 @@
+god fever
diff --git a/jeans.java b/jeans.java
index 0b5d6ed..f467597 100644
--- a/jeans.java
+++ b/jeans.java
@@ -1 +1,6 @@
+hi
```

commit id check difference

```
Kunal@Kunalsh:~/Documents/test$ git push
Username for 'https://github.com': Kunalshiwarkar
Password for 'https://Kunalshiwarkar@github.com':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 354 bytes | 354.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Kunalshiwarkar/test.git
  47c249e..92d89ba main -> main
```

The screenshot shows a GitHub repository named 'test'. The commit history for the 'main' branch is displayed, showing 10 commits. The most recent commit, 'kunal diff file' (commit ID 92d89ba, 48 minutes ago), includes a note about a file named 'cycle.java' being created. Below the commit list, there is a detailed view of the 'cycle.java' file, showing its content: 'god fever'.

File	Description	Time
abc.java	file created	last week
cycle.java	diff file	48 minutes ago
jeans.java	diff file	48 minutes ago
mens.java	mens category	3 days ago
sample.java	done	3 days ago
shirts.java	sample	3 days ago
test2	done	3 days ago

The screenshot shows the content of the 'cycle.java' file from the 'test' repository. The file contains a single line of code: 'god fever'. There are tabs for 'Code', 'Blame', and 'History'. A note at the bottom indicates that the code is 55% faster with GitHub Copilot.

A screenshot of a GitHub repository page for a file named "jeans.java". The file contains the following code:

```
1 test
2 hello
3 hi
4 welcome
5 to devops
```

The GitHub interface shows basic file statistics (6 lines, 5 loc, 33 Bytes) and a note about GitHub Copilot. There are standard file operations buttons at the top right.

2. Deleting a file

A.)
\$ git rm <filename>
\$ git commit <filename> -m "comment"
\$ git push

*If we want to remove Unnecessary file

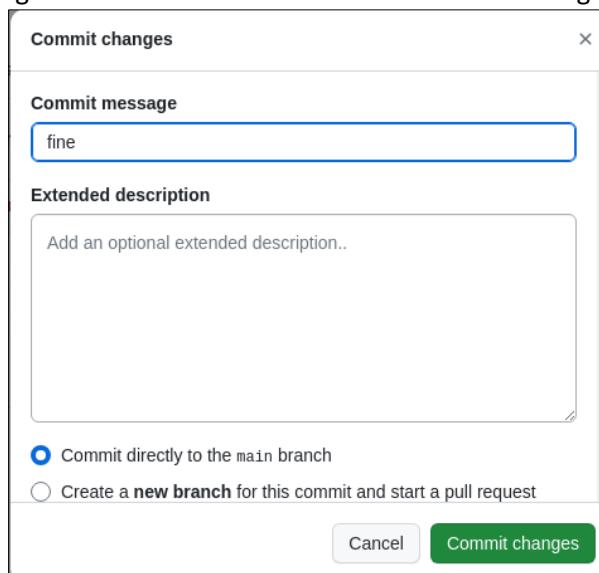
Go to git hub server and delete file from repository.

A screenshot of a GitHub repository page for a file named "mens.java". The file contains the following code:

```
1 file ok.
```

A context menu is open on the right side of the file content area. The "Delete file" option is highlighted with a red arrow. Other options in the menu include "Copy" (with keyboard shortcut Ctrl + Shift + C), "View options" (with checkboxes for "Show code folding buttons", "Wrap lines", "Center content", and "Open symbols on click"), and "Permalink".

Then write commit message for delete in that file then click commit changes.



File successfully deleted.

History

Name	Last commit message	Last commit date
abc.java	file created	last week
cycle.java	diff file	1 hour ago
jeans.java	diff file	1 hour ago
mens.java		
sample.java	done	3 days ago
shirts.java	sample	3 days ago
test2	done	3 days ago

Kunalshiwarkar fine

bd9f47a 2 minutes ago 11 commits

abc.java	file created	last week
cycle.java	diff file	1 hour ago
jeans.java	diff file	1 hour ago
sample.java	done	3 days ago
shirts.java	sample	3 days ago
test2	done	3 days ago

But not delete from terminal then

Git pull command used to delete file from terminal

```
kunal@kunalsh:~/Documents/test$ ls -a
. . . abc.java cycle.java .git jeans.java mens.java sample.java shirts.java test2
kunal@kunalsh:~/Documents/test$ git pull
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.                                but mens.java file show in terminal
remote: Compressing objects: 100% (2/2), done.                            then we used git pull command
remote: Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (2/2), 598 bytes | 598.00 KiB/s, done.
From https://github.com/Kunalshiwarkar/test
  92d89ba..bd9f47a main      -> origin/main
Updating 92d89ba..bd9f47a
Fast-forward
  mens.java | 1 -
  1 file changed, 1 deletion(-)
  delete mode 100644 mens.java
kunal@kunalsh:~/Documents/test$ ls -a                                         when mens.java file not show.
. . . abc.java cycle.java .git jeans.java sample.java shirts.java test2
```

If file delete from local then used git push

```
kunal@kunalsh:~/Documents/test$ ls -a
. .. abc.java cycle.java .git jeans.java sample.java shirts.java test2
kunal@kunalsh:~/Documents/test$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
kunal@kunalsh:~/Documents/test$ git rm sample.java → rm command used to delete file.
rm 'sample.java'
kunal@kunalsh:~/Documents/test$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:   sample.java

kunal@kunalsh:~/Documents/test$ git commit -m "sample.java"
```

```
kunal@kunalsh:~/Documents/test$ git push
Username for 'https://github.com': Kunalshiwarkar
Password for 'https://Kunalshiwarkar@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 231 bytes | 231.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Kunalshiwarkar/test.git
  bd9f47a..2bd33df  main -> main
kunal@kunalsh:~/Documents/test$ ls -a → sample.java file not show
. .. abc.java cycle.java .git jeans.java shirts.java test2
```

Now, sample.java is not show in test repository.

The screenshot shows a GitHub repository interface. At the top, it says 'test' (Public). Below that, it shows 'main' (branch), '1 branch', and '0 tags'. On the right, there are buttons for 'Pin', 'Unwatch', 'Go to file', 'Add file', and 'Code'. The commit history is listed:

Author	File	Description	Time
kunal	sample.java	2bd33df 8 minutes ago	12 commits
	abc.java	file created	last week
	cycle.java	diff file	1 hour ago
	jeans.java	diff file	1 hour ago
	shirts.java	sample	3 days ago
	test2	done	3 days ago

4. Renaming a file/folder

```
A.) $ git mv Login.java <new-name>
      $ git commit -m "rename Login"
      $ git push
```

Mv command is used to rename

```
kunal@kunalsh:~/Documents/test$ git mv test2 grras.java
kunal@kunalsh:~/Documents/test$ ls -a
. .. abc.java cycle.java .git grras.java jeans.java shirts.java
kunal@kunalsh:~/Documents/test$ git commit -m "atest file"
```

Then used git push.

```
kunal@kunalsh:~/Documents/test$ git push
Username for 'https://github.com': Kunalshiwarkar
Password for 'https://Kunalshiwarkar@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 266 bytes | 266.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Kunalshiwarkar/test.git
  2bd33df..26765d4  main -> main
```

A screenshot of a GitHub repository page for a project named 'test'. The repository is public. It shows 1 branch and 0 tags. The main branch has 13 commits. A specific commit by 'kunal' titled 'atest file' is highlighted, showing it was created 5 minutes ago. Other files listed are abc.java, cycle.java, jeans.java, and shirts.java, with their creation times: abc.java (last week), cycle.java (1 hour ago), jeans.java (1 hour ago), and shirts.java (3 days ago).

Carry the history for new file from old file

→ \$ git log --follow <new-name>

```
kunal@kunalsh:~/Documents/test$ git log --follow grras.java
commit 26765d48b59e705bd5b813a91a47cba44470c076 (HEAD -> main, origin/main, origin/HEAD)
Author: kunal <kunal@kunalsh.myguest.virtualbox.org>
Date:   Sun Oct 15 21:29:13 2023 +0530

  atest file

commit 157abb4e06ec79f17120a7d8ee6ed574a348252e
Author: Kunalshiwarkar <kunalshiwarkar786@gmail.com>
Date:   Thu Oct 12 13:11:11 2023 +0530

done
```

```
# Undoing the changes:  
# undo/revert local changes from source area in exits files. ie. present in remote repo.  
$ git checkout <filename>
```

Existing file

1. Source to working

```
kunal@kunalsh:~/Documents/test$ ls -a  
. . . abc.java cycle.java .git grras.java jack.txt jeans.java shirts.java test.java  
kunal@kunalsh:~/Documents/test$ git status  
On branch main  
Your branch is up to date with 'origin/main'.  
  
nothing to commit, working tree clean  
kunal@kunalsh:~/Documents/test$ vim grras.java  
kunal@kunalsh:~/Documents/test$ git status  
On branch main  
Your branch is up to date with 'origin/main'.  
  
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: grras.java  
  
no changes added to commit (use "git add" and/or "git commit -a")  
kunal@kunalsh:~/Documents/test$ cat grras.java  
motu  
patlu  
kunal@kunalsh:~/Documents/test$ git checkout grras.java  
Updated 1 path from the index  
kunal@kunalsh:~/Documents/test$ git status  
On branch main  
Your branch is up to date with 'origin/main'.  
  
nothing to commit, working tree clean
```

Git checkout for
source to working
area

2. # Unstage the changes from STAGE area

```
$ git reset HEAD <filename>  
I.e. file convert from stage area to source area using rest HEAD
```

```
kunal@kunalsh:~/Documents/test$ ls -a  
. . . abc.java cycle.java .git grras.java jack.txt jeans.java shirts.java test.java  
kunal@kunalsh:~/Documents/test$ vim test.java  
kunal@kunalsh:~/Documents/test$ git status  
On branch main  
Your branch is up to date with 'origin/main'.  
  
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: test.java  
  
no changes added to commit (use "git add" and/or "git commit -a")  
kunal@kunalsh:~/Documents/test$ git add .  
kunal@kunalsh:~/Documents/test$ git reset HEAD test.java  
Unstaged changes after reset:  
M test.java  
kunal@kunalsh:~/Documents/test$ git status  
On branch main  
Your branch is up to date with 'origin/main'.  
  
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: test.java  
  
no changes added to commit (use "git add" and/or "git commit -a")
```

► file convert from stage area to source area using
git rest HEAD <file name>

3. # Move changes from .git to source area

```
$ git reset --mixed HEAD~1
```

File is in .git

```
kunal@kunalsh:~/Documents/test$ git add .
kunal@kunalsh:~/Documents/test$ git status
On branch main
Your branch is up to date with 'origin/main'.

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

kunal@kunalsh:~/Documents/test$ git commit -m "test.java"
```

```
kunal@kunalsh:~/Documents/test$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
kunal@kunalsh:~/Documents/test$ git reset --mixed HEAD~1
Unstaged changes after reset:
M      test.java
kunal@kunalsh:~/Documents/test$ git status
On branch main
Your branch is up to date with 'origin/main'.

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:   test.java

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

File is transfer from .git into
source area using
git reset --mixed HEAD~1

4. Move changes from .git to stage area

```
$ git reset --soft HEAD~1
```

```
kunal@kunalsh:~/Documents/test$ git status
On branch main
Your branch is up to date with 'origin/main'.

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:   test.java

no changes added to commit (use "git add" and/or "git commit -a")
kunal@kunalsh:~/Documents/test$ git add .
kunal@kunalsh:~/Documents/test$ git commit -m "test"
```

File transfer from .git to stage
area using
\$ git reset --soft HEAD~1

5. Remove changes from everywhere

```
$ git reset --hard HEAD~1
```

```
kunal@kunalsh:~/Documents/test$ git status
On branch main
Your branch and 'origin/main' have diverged,
and have 1 and 1 different commits each, respectively.
(use "git pull" to merge the remote branch into yours)

nothing to commit, working tree clean
kunal@kunalsh:~/Documents/test$ git reset --hard HEAD~1
HEAD is now at 1f12718 add
kunal@kunalsh:~/Documents/test$ git status
On branch main
Your branch is behind 'origin/main' by 1 commit, and can be fast-forwarded.
(use "git pull" to update your local branch)

nothing to commit, working tree clean
kunal@kunalsh:~/Documents/test$ git pull
Updating 1f12718..38157e8
Fast-forward
 test.java | 4 +---
 1 file changed, 1 insertion(+), 3 deletions(-)
kunal@kunalsh:~/Documents/test$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
kunal@kunalsh:~/Documents/test$ ls -a
. . . abc.java cycle.java .git grras.java jack.txt jeans.java shirts.java test.java
```

.git to working area
Using
git reset --hard HEAD~!

6. # Show all the files with content difference that are modified as part of a commit.

```
git show <SHA-VALUE>
```

```
kunal@kunalsh:~/Documents/test$ git show 38157e8f98d1ea3e0bfdca752e5937ee37b78dab
commit 38157e8f98d1ea3e0bfdca752e5937ee37b78dab (HEAD -> main, origin/main, origin/HEAD)
Author: kunal <kunal@kunalsh.myguest.virtualbox.org>
Date:   Sun Oct 15 22:46:05 2023 +0530

    test file

diff --git a/test.java b/test.java
index 937f40a..318de95 100644
--- a/test.java
+++ b/test.java
@@ -1,5 +1,3 @@
 hello
 all
-welcome
-to
-gitt
+
```

7. # List only files that are committed as part of a revision.

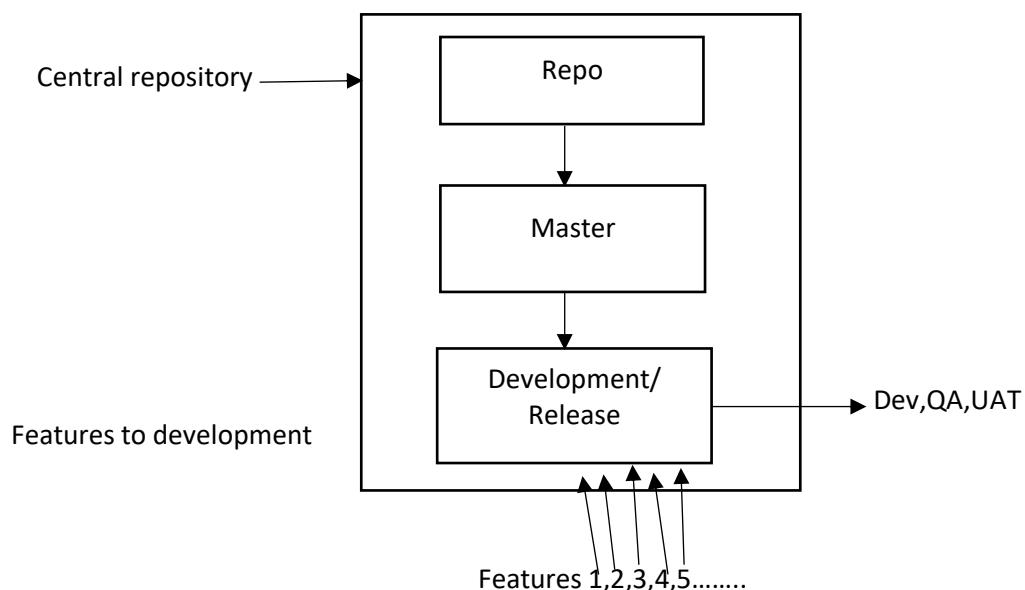
```
$ git diff-tree --no-commit-id --name-only -r <SHA-VALUE>
```

```
kunal@kunalsh:~/Documents/test$ git diff-tree --no-commit-id --name-only -r 38157e8f98d1ea3e0bfdca752e5937ee37b78dab
test.java
```

What is branching?

- Branches allow you to develop features, fix bugs or safely experiment with new ideas in a contained area of your repository.
- You always create a branch from an existing branch.
- Typically, you might create a new branch from the default branch of your repository.
- Branching is facility to every member independently create data and working on data .
- Eg: all developers working on admin's repository.
- 1st developer code should not get spoiled because of others so all developers seperated in groups to do work independently separate area and don't disturb by others, that's separate area all developers work independently is called branching.

Branching structure



[The more files are in development, the more features will be there]

Once all the codes are submitted to the developers, then the complete features branch will go to the central repository.

- After creating code , code can test in sever 3 stages of environment .
 - 1.Development [Dev] → lightly code test
 - 2.Quality Assurance [QA]
 - 3.User Acceptance Test [UAT] → heavily code test
- If any error found in code during testing of any 3 stages in server then error is solved in hot fix or bug fix .
- After problem solving code then transfer into UAT .
- Once code runs on server ie dev,qa,uat.
- Then merge in development.
- Merge Development deploy code in production.
- Add development in master means merge master in defaults.
- After development , Devops admin lock the master.
- Development is replica of Master.
- If 10 repository created in master then replica i.e. there is 10 repository in development.
- Only devops admin can push data in master and there is access to devops admin.

- Features created by developer only permission of pull request.

Raised a pull request → Branches data merge in development.
Master branch defaults display during production.

Q. How many types of branches exist in git?

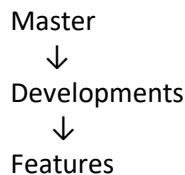
Ans – There are 5 types of branches.

1. Hotfix
2. Bugfix
3. Release
4. Master
5. Development.

Q. Can overwrite possible in code ? what is branching strategy you work?

Ans →

- No, because we used git so no override possible also same file created.
- We can create branches in branch but not from Master branch.
- We can merge a data using pull request only.



Q. What is merging? state its type.

Ans → transferring data from one branch to another branch is called merging.

1. fast forward merge

Code transfer from one branch to another branch.[without conflict]

2. real merge

Whenever we transferring data from one branch to another if we are getting conflicts, we need to resolve the conflicts.

Date-19/10/2023

Conversion of main branch into master branch in default permanently.

Go to setting

The diagram illustrates the process of changing the default branch from 'main' to 'master' on GitHub:

- Main is master default branch**: A callout box pointing to the 'main' branch in the repository settings.
- All repository are made under Mater branch default**: A callout box pointing to the GitHub profile settings.
- Now , default is master branch**: A callout box pointing to the GitHub profile settings after the change.
- Click on repositories**: A callout box pointing to the 'Repositories' link in the GitHub sidebar.
- Rename main into master and click on update**: A callout box pointing to the 'Repository default branch' section in the repository settings.
- Repository default branch**: A callout box pointing to the 'master' dropdown menu in the repository settings.
- Choose the default branch for your new personal rep your integrations still require "master" as the default b more about default branches.**: A callout box pointing to the explanatory text below the dropdown menu.
- Repositories**: A callout box pointing to the 'Repositories' link in the GitHub sidebar.

Owner * Repository name *

Kunalshiwarkar / branching
✓ branching is available.

First of all, I went to GitHub and created a repository named Branching.

Great repository names are short and memorable. Need inspiration? How about [upgraded-octo-couscous](#) ?

Description (optional)

Public Anyone on the internet can see this repository. You choose who can commit.
 Private You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file This is where you can write a long description for your project. [Learn more about READMEs](#).

After that clicked on add readme file because I have to create master in the repository.

Click on create a repository.

Kunalshiwarkar / branching

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

branching (Public) Branch master show

master 2 branches 0 tags Go to file Add file ▾ Code ▾

This branch is up to date with main.

Kunalshiwarkar Initial commit

README.md copy url and paste into terminal Initial commit

README.md Paste url here

Local Codespaces

Clone

HTTPS SSH GitHub CLI

https://github.com/Kunalshiwarkar/branching.git Use Git or checkout with SVN using the web URL.

Download ZIP

```
kunal@kunalsh:~$ cd Documents/
kunal@kunalsh:~/Documents$ git clone https://github.com/Kunalshiwarkar/branching.git
Cloning into 'branching'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
kunal@kunalsh:~/Documents$ cd branching/
kunal@kunalsh:~/Documents/branching$ ls -a
. .git README.md
```

-List all the branch

\$ git branch

-Creating a new branch

\$ git branch <new branch name>

:- Switching to a new created branch

```
$ git checkout <new branch name>
```

:-Creating and switching to a newly created branch

```
$ git checkout -b <new branch name>
```

:-list all remote branches

```
$ git branch -r or -a
```

```
kunal@kunalsh:~/Documents/branching$ git branch * master
* release1.1
kunal@kunalsh:~/Documents/branching$ git branch release1.1
kunal@kunalsh:~/Documents/branching$ git branch
* master
release1.1
kunal@kunalsh:~/Documents/branching$ git checkout release1.1
Switched to branch 'release1.1'
kunal@kunalsh:~/Documents/branching$ git branch
  master
* release1.1
  now branch is release1.1
kunal@kunalsh:~/Documents/branching$ git checkout master
Switched to branch 'master' again login into master
Your branch is up to date with 'origin/master'.
kunal@kunalsh:~/Documents/branching$ git branch
* master
  release1.1
```

all branches have to be listed.
we are present in which current branch can also see using git branch command.

if you want to create new
branch then the command is
\$ git branch <new branch name>

if you to login in
another branch or
enter into new
branch the
command is
\$git checkout
<branch name>

```
kunal@kunalsh:~/Documents/branching$ git checkout -b release1.2 if I have created new
Switched to a new branch 'release1.2' branch and enter in the newly created branch , the command
kunal@kunalsh:~/Documents/branching$ git branch $ git checkout -b <new branch name>
  master
  release1.1
* release1.2
kunal@kunalsh:~/Documents/branching$ git branch -r this command will see all the branches in
  origin/HEAD -> origin/master the central repository will known through this command.
  origin/master
```

```
kunal@kunalsh:~/Documents/branching$ git branch
  master
  release1.1
* release1.2
* means it is in release1.2 branch.
kunal@kunalsh:~/Documents/branching$ vim sample.java
kunal@kunalsh:~/Documents/branching$ git status
On branch release1.2
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    sample.java
      source area
nothing added to commit but untracked files present (use "git add" to track)
kunal@kunalsh:~/Documents/branching$ git add .
```

```
kunal@kunalsh:~/Documents/branching$ git status
On branch release1.2
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   sample1.java
```

stage area

```
kunal@kunalsh:~/Documents/branching$ git commit -m "msg"
```

:-Merge files from source branch to target branch

```
$ git merge <source branch name>
```

```
kunal@kunalsh:~/Documents/branching$ git status local area
On branch release1.2
nothing to commit, working tree clean
kunal@kunalsh:~/Documents/branching$ git checkout release1.1
Switched to branch 'release1.1'
kunal@kunalsh:~/Documents/branching$ git branch      login into release1.1
  master
* release1.1
  release1.2
kunal@kunalsh:~/Documents/branching$ ls -a
.  ..  .git  README.md  there is no file in release1.1
kunal@kunalsh:~/Documents/branching$ git merge release1.2
Updating 10ca98a..75421d6
Fast-forward
  sample.java | 3 +++
  1 file changed, 3 insertions(+)
  create mode 100644 sample.java
kunal@kunalsh:~/Documents/branching$ ls -a
.  ..  .git  README.md  sample.java
kunal@kunalsh:~/Documents/branching$ git status
On branch release1.1
nothing to commit, working tree clean
```

:-First push newly created branch into remote repo

```
$ git push --all
```

```
kunal@kunalsh:~/Documents/branching$ git branch
  main
  master
* release1.1
  release1.2
kunal@kunalsh:~/Documents/branching$ git push --all
Username for 'https://github.com': KunalShiwarkar
Password for 'https://KunalShiwarkar@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 309 bytes | 309.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/KunalShiwarkar/branching.git
 * [new branch]    release1.1 -> release1.1
 * [new branch]    release1.2 -> release1.2
```

currently branch release1.2

push into git hub central repository.

release1.2 had recent pushes 1 minute ago

message of pull request come from push request of local terminal

Compare & pull request

release1.1 had recent pushes 1 minute ago

Compare & pull request

master 3 branches 0 tags

Go to file Add file Code

Switch branches/tags

Find or create a branch...

Initial commit 10ca98a 1 hour ago 1 commit

1 hour ago

Branches Tags default

release1.1 release1.2

click on release1.1 branch show file sample.java then click on release1.2 branch show file sample.java

View all branches Switch branches or tags

release1.1 4 branches 0 tags

Go to file Add file Code

This branch is 1 commit ahead of main.

Contribute

kunal msg 7737d5d 51 minutes ago 2 commits

README.md Initial commit 1 hour ago

sample1.java msg 51 minutes ago

release1.2 4 branches 0 tags

Go to file Add file Code

This branch is 1 commit ahead of main.

Contribute

kunal msg 7737d5d 52 minutes ago 2 commits

README.md Initial commit 1 hour ago

sample1.java msg 52 minutes ago

```
kunal@kunalsh:~/Documents/branching$ git branch -r
origin/HEAD -> origin/master
origin/master
origin/release1.1
origin/release1.2
↑ for verify
kunal@kunalsh:~/Documents/branching$ git branch
  master
* release1.1
  release1.2
```

What is Head of branch?

Head is a pointer in a git which always points to parent of next commit. “*”

```
kunal@kunalsh:~/Documents/branching$ cd .git/
kunal@kunalsh:~/Documents/branching/.git$ ls -a
. .. branches COMMIT_EDITMSG config description HEAD hooks index info logs objects ORIG_HEAD packed-refs refs
kunal@kunalsh:~/Documents/branching/.git$ cat HEAD
ref: refs/heads/release1.1
Will tell which branch you are in
kunal@kunalsh:~/Documents/branching/.git$ cd branches/
kunal@kunalsh:~/Documents/branching/.git/branches$ ls -a
. ..
kunal@kunalsh:~/Documents/branching/.git/branches$ cd ..
kunal@kunalsh:~/Documents/branching/.git$ cd hooks/
kunal@kunalsh:~/Documents/branching/.git/hooks$ ls -a
.           commit-msg.sample    pre-applypatch.sample  prepare-commit-msg.sample  pre-receive.sample
..          fsmonitor-watchman.sample  pre-commit.sample   pre-push.sample        push-to-checkout.sample
applypatch-msg.sample  post-update.sample    pre-merge-commit.sample  pre-rebase.sample      update.sample
```

Eg:

- I am in firstly source or stage.
- Suddenly the boss said, you should immediately close this old branch and create a new branch.
- Unfortunately, the data in the file 1 of old branch that I have created should not be visible in the new branch, so the data will be stored in the background.
- Then focused on boss new branch job.
- After the boss's work is completed, now we will store the background data in file 1 of our old branch
- Background store data in stage area.

```
kunal@kunalsh:~/Documents/branching$ git branch
  master
* release1.1
  release1.2
```

```
kunal@kunalsh:~/Documents/branching$ vim abc.java
kunal@kunalsh:~/Documents/branching$ vim abc1.java
kunal@kunalsh:~/Documents/branching$ echo 'git' > abc2.java
kunal@kunalsh:~/Documents/branching$ echo 'branch' > abc3.java
kunal@kunalsh:~/Documents/branching$ git status
On branch release1.1
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    abc.java
    abc1.java
    abc2.java
    abc3.java
all file in source
nothing added to commit but untracked files present (use "git add" to track)
kunal@kunalsh:~/Documents/branching$ git add abc.java abc1.java
```

```

kunal@kunalsh:~/Documents/branching$ git status
On branch release1.1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file: abc.java
    new file: abc1.java
2 files in stage area and
remaining file in source area

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    abc2.java
    abc3.java

kunal@kunalsh:~/Documents/branching$ git stash
Saved working directory and index state WIP on release1.1: 7737d5d msg
kunal@kunalsh:~/Documents/branching$ git stage
Nothing specified, nothing added.
hint: Maybe you wanted to say 'git add .?
hint: Turn this message off by running
hint: "git config advice.addEmptyPathspec false"

```

command used for store background data

save in stage and working on stage area.

```

kunal@kunalsh:~/Documents/branching$ git checkout release1.2
Switched to branch 'release1.2'
kunal@kunalsh:~/Documents/branching$ ls -a
. .. abc2.java abc3.java .git README.md sample1.java
kunal@kunalsh:~/Documents/branching$ git checkout release1.1
Switched to branch 'release1.1'
kunal@kunalsh:~/Documents/branching$ git status
On branch release1.1
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    abc2.java
    abc3.java
background data in stage
area is not seen.

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

```

```

kunal@kunalsh:~/Documents/branching$ git stash pop
On branch release1.1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file: abc.java
    new file: abc1.java
if background data store in stage and see
this file using git stash pop

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    abc2.java
    abc3.java

Dropped refs/stash@{0} (b9f3b65d2f318c6ede6800b13ecca7a634804ab2)
kunal@kunalsh:~/Documents/branching$ ls -a
. .. abc1.java abc2.java abc3.java abc.java .git README.md sample1.java

```

```
kunal@kunalsh:~/Documents/branching$ git add .
kunal@kunalsh:~/Documents/branching$ git status
On branch release1.1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file: abc.java
    new file: abc1.java      file in stage area
    new file: abc2.java
    new file: abc3.java

kunal@kunalsh:~/Documents/branching$ git commit -m "msg"
```

```
kunal@kunalsh:~/Documents/branching$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'. branch transfer from release1.1
to master
kunal@kunalsh:~/Documents/branching$ git branch
* master
  release1.1
  release1.2
kunal@kunalsh:~/Documents/branching$ ls -a
.  ..  .git  README.md
```

```
kunal@kunalsh:~/Documents/branching$ git merge release1.1
Updating 7a7e014..eacf814
Fast-forward
 abc.java      | 1 +
 abc1.java     | 1 +
 abc2.java     | 1 +
 abc3.java     | 1 +
 sample1.java  | 3 +++
 5 files changed, 7 insertions(+)
 create mode 100644 abc.java
 create mode 100644 abc1.java
 create mode 100644 abc2.java
 create mode 100644 abc3.java
 create mode 100644 sample1.java
kunal@kunalsh:~/Documents/branching$ 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
kunal@kunalsh:~/Documents/branching$ git push --all
```

This is called fast forward merging.

 **branching** Public

[master](#) [3 branches](#) [0 tags](#) [Go to file](#) [Add file](#) [Code](#)

	kunal msg	9bd81c3 7 minutes ago
	README.md	Initial commit
	abc.java	msg
	abc1.java	msg
	abc2.java	msg
	abc3.java	msg
	sample.java	msg

 **branching** Public

[release1.1](#) [3 branches](#) [0 tags](#) [Go to file](#) [Add file](#) [Code](#)

This branch is up to date with master.

[Contribute](#)

	kunal msg	9bd81c3 9 minutes ago
	README.md	Initial commit
	abc.java	msg
	abc1.java	msg
	abc2.java	msg
	abc3.java	msg
	sample.java	msg

 **branching** Public

[release1.2](#) [3 branches](#) [0 tags](#) [Go to file](#) [Add file](#) [Code](#)

This branch is 1 commit behind master.

[Contribute](#)

	kunal msg	75421d6 1 hour ago
	README.md	Initial commit
	sample.java	msg

Now, created branch in GUI
write dev new branch then click on create branch.

The screenshot shows a GitHub repository named 'branching'. In the top left, there's a dropdown for 'master' and a button for '3 branches'. A red arrow points from the text 'write dev new branch' to the 'Create branch' button. Another red arrow points from the text 'then click on create branch.' to the 'Create branch' button. The modal window has 'Branches' selected. It shows a list of commits: 'Initial commit' (17 minutes ago), 'msg' (17 minutes ago), 'msg' (17 minutes ago), and 'msg' (17 minutes ago). Below the commits, there are two files: 'abc3.java' and 'sample.java', both with 'msg' status. At the bottom right of the modal, there's a 'Switch branches or tags' button.

```
kunal@kunalsh:~/Documents/branching$ git branch
* master
  release1.1
  release1.2
kunal@kunalsh:~/Documents/branching$ git branch -r
  origin/HEAD -> origin/master
  origin/master
  origin/release1.1          for verify
  origin/release1.2
kunal@kunalsh:~/Documents/branching$ git pull
From https://github.com/KunalShiwarkar/branching
 * [new branch]      dev      -> origin/dev
Already up to date.
kunal@kunalsh:~/Documents/branching$ git branch -r
  origin/HEAD -> origin/master
  origin/dev
  origin/master
  origin/release1.1          pull dev from central
  origin/release1.2          repository.
kunal@kunalsh:~/Documents/branching$ git branch
* master
  release1.1
  release1.2
```

```
kunal@kunalsh:~/Documents/branching$ git checkout dev
Branch 'dev' set up to track remote branch 'dev' from 'origin'.
Switched to a new branch 'dev'
kunal@kunalsh:~/Documents/branching$ git branch
* dev
  master
  release1.1
  release1.2
kunal@kunalsh:~/Documents/branching$ echo "branch" > dev1.java
kunal@kunalsh:~/Documents/branching$ echo "branch" > dev2.java
kunal@kunalsh:~/Documents/branching$ git status
On branch dev
Your branch is up to date with 'origin/dev'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    dev1.java
    dev2.java

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

```
kunal@kunalsh:~/Documents/branching$ git add .
kunal@kunalsh:~/Documents/branching$ git status
On branch dev
Your branch is up to date with 'origin/dev'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   dev1.java
    new file:   dev2.java

kunal@kunalsh:~/Documents/branching$ git commit -m "msg"
```

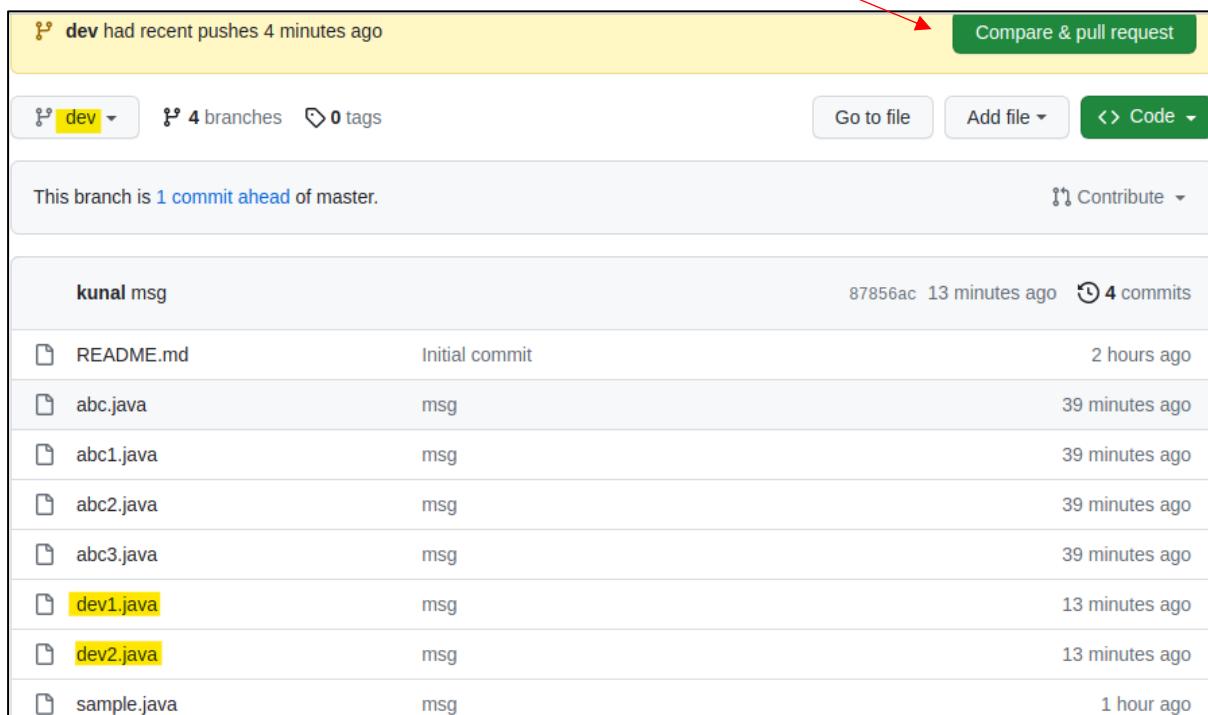
```
kunal@kunalsh:~/Documents/branching$ git status
On branch dev
Your branch is ahead of 'origin/dev' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

```
kunal@kunalsh:~/Documents/branching$ git push --all
Username for 'https://github.com': Kunalshiwarkar
Password for 'https://Kunalshiwarkar@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 237 bytes | 237.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Kunalshiwarkar/branching.git
  9bd81c3..87856ac  dev -> dev
```

Go to central repository , we see dev1 and dev2 files in branch dev.
This done due to push from local terminal .

there is pull request create from push of local terminal.



dev had recent pushes 4 minutes ago		
File	Message	Time
README.md	Initial commit	2 hours ago
abc.java	msg	39 minutes ago
abc1.java	msg	39 minutes ago
abc2.java	msg	39 minutes ago
abc3.java	msg	39 minutes ago
dev1.java	msg	13 minutes ago
dev2.java	msg	13 minutes ago
sample.java	msg	1 hour ago

File	Message	Time
README.md	Initial commit	2 hours ago
abc.java	msg	46 minutes ago
abc1.java	msg	46 minutes ago
abc2.java	msg	46 minutes ago
abc3.java	msg	46 minutes ago
sample.java	msg	2 hours ago

Now we transfer file dev1 and dev2 from dev branch into master branch using merge.

Click on pull request.

Destination source

The screenshot shows the GitHub pull request interface. At the top, it says "base: master" and "compare: dev". A green checkmark indicates "Able to merge. These branches can be automatically merged." The message field contains "msg" and "ok". Below the message field is a red arrow pointing right. A note at the bottom says "Attach files by dragging & dropping, selecting or pasting them." At the bottom right is a green "Create pull request" button.

Then click on merge pull request

The screenshot shows the GitHub merge pull request review interface. It lists three items: 1) "Require approval from specific reviewers before merging" (with a link to "Branch protection rules" and a "Add rule" button). 2) "Continuous integration has not been set up" (with a link to "GitHub Actions" and "several other apps"). 3) "This branch has no conflicts with the base branch" (marked with a green checkmark, stating "Merging can be performed automatically"). At the bottom left is a green "Merge pull request" button, and at the bottom right is a link to "or view command line instructions".

Then click on comment

The screenshot shows the GitHub comment interface. The message field contains "msg" and "ok". Below the message field is a red arrow pointing right. A note at the bottom says "Attach files by dragging & dropping, selecting or pasting them." At the bottom right are two buttons: "Close with comment" and "Comment".

Now in master branching dev1 and dev2 present.

A screenshot of a GitHub repository named 'branching'. The repository is public. The 'master' branch is selected. There are 4 branches and 0 tags. A pull request by Kunalshiwarkar has been merged. The commit history shows several files: README.md (Initial commit, 3 hours ago), abc.java (msg, 1 hour ago), abc1.java (msg, 1 hour ago), abc2.java (msg, 1 hour ago), abc3.java (msg, 1 hour ago), dev1.java (msg, 38 minutes ago), dev2.java (msg, 38 minutes ago), and sample.java (msg, 2 hours ago). The files dev1.java and dev2.java are highlighted in yellow.

Now we transfer data from dev to release1.1

Source to destination i.e. dev to release1.1

Click pull request.

This is called manual pull request.

A screenshot of the same GitHub repository 'branching'. The 'Pull requests' tab is selected. The repository is now public. The 'release1.1' branch is selected. There are 4 branches and 0 tags. The interface shows standard GitHub navigation and search bars.

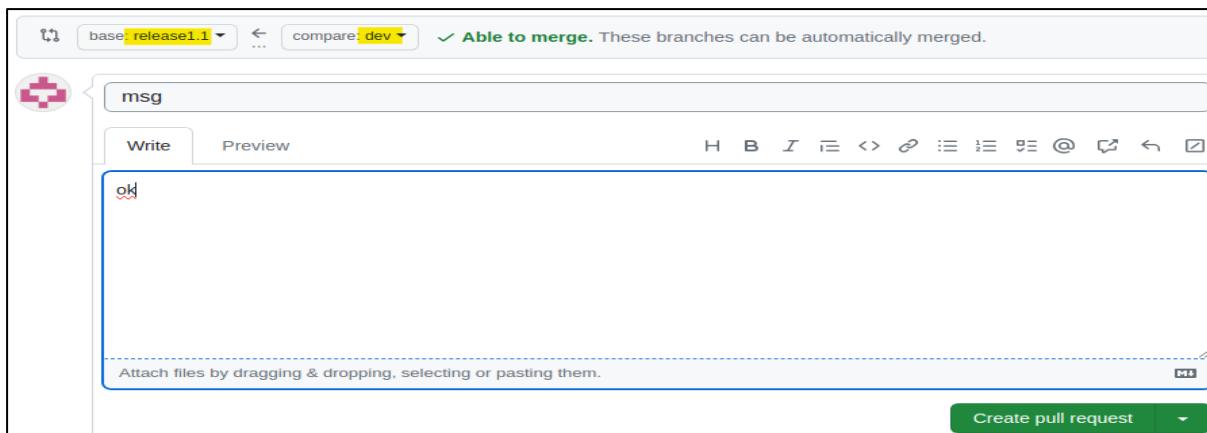
Click on new pull request.

A screenshot of the 'New pull request' dialog. It displays a message about labeling issues and pull requests for new contributors. It includes a 'Dismiss' button, a 'Filters' dropdown set to 'is:pr is:open', a search bar, and a 'New pull request' button. Below the dialog are standard GitHub filtering and sorting options.

Destination

source

click on create pull request.



Merge and confirm commit

A screenshot of a pull request merge history. It shows a purple "Merged" button followed by the text "Kunalshiwarkar merged 1 commit into release1.1 from dev 7 minutes ago". Below this, there's a conversation tab with two messages. The first message is from "Kunalshiwarkar" with the text "No description provided." and a smiley face emoji. The second message is also from "Kunalshiwarkar" with the text "merged commit b68b5fa into release1.1 7 minutes ago". There are tabs for "Conversation" (2), "Commits" (1), "Checks" (0), and "Files changed" (2). A "Revert" button is visible at the bottom right of the message area.

Then message commit.

A screenshot of a repository commit history for the branch "release1.1". The top bar shows "4 branches" and "0 tags". The commit list starts with "Initial commit" for "README.md" by "Kunalshiwarkar" (3 hours ago). Following this are several "msg" commits for files "abc.java", "abc1.java", "abc2.java", "abc3.java", "dev1.java", "dev2.java", and "sample.java", all by "Kunalshiwarkar" (1-2 hours ago). The commit for "dev1.java" and "dev2.java" is highlighted with a yellow background. The commit for "sample.java" is also highlighted with a yellow background.

Date-20/10/23

Conflict merge

At same line no

```
kunal@kunalsh:~/Documents/branching$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
kunal@kunalsh:~/Documents/branching$ git branch
* master
  release1.1
  release1.2
kunal@kunalsh:~/Documents/branching$ git checkout release1.1
Switched to branch 'release1.1'
kunal@kunalsh:~/Documents/branching$ ls -a
. .. .git README.md sample.java
kunal@kunalsh:~/Documents/branching$ cat sample.java
hello all welcome

to

grras
```

```
kunal@kunalsh:~/Documents/branching$ git checkout release1.2
Switched to branch 'release1.2'
kunal@kunalsh:~/Documents/branching$ ls -a
. .. .git README.md sample.java
kunal@kunalsh:~/Documents/branching$ cat sample.java
hello all welcome

to

grras
kunal@kunalsh:~/Documents/branching$ git status
On branch release1.2
nothing to commit, working tree clean
```

```

kunal@kunalsh:~/Documents/branching$ git checkout release1.1
Switched to branch 'release1.1'
kunal@kunalsh:~/Documents/branching$ git status
On branch release1.1
nothing to commit, working tree clean
kunal@kunalsh:~/Documents/branching$ ls -a
. .. .git README.md sample.java
kunal@kunalsh:~/Documents/branching$ vim sample.java ←
kunal@kunalsh:~/Documents/branching$ git status
On branch release1.1
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:   sample.java

no changes added to commit (use "git add" and/or "git commit -a")
kunal@kunalsh:~/Documents/branching$ git add .
kunal@kunalsh:~/Documents/branching$ git status
On branch release1.1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   sample.java

kunal@kunalsh:~/Documents/branching$ git commit -m "msg"

```

Remove "to" and
write "1012" on
exact line no

```

kunal@kunalsh:~/Documents/branching$ git checkout release1.2
Switched to branch 'release1.2'
kunal@kunalsh:~/Documents/branching$ vim sample.java ←
kunal@kunalsh:~/Documents/branching$ git status
On branch release1.2
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:   sample.java

no changes added to commit (use "git add" and/or "git commit -a")
kunal@kunalsh:~/Documents/branching$ git add .
kunal@kunalsh:~/Documents/branching$ git commit -m "msg23"

```

I remove "to" & write
"devops conflicts" at
exact same no line.

```

kunal@kunalsh:~/Documents/branching$ git checkout release1.1
Switched to branch 'release1.1'
kunal@kunalsh:~/Documents/branching$ git merge release1.2
Auto-merging sample.java
CONFLICT (content): Merge conflict in sample.java
Automatic merge failed; fix conflicts and then commit the result.
kunal@kunalsh:~/Documents/branching$ vim sample.java ←
kunal@kunalsh:~/Documents/branching$ git status
On branch release1.1
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   sample.java

no changes added to commit (use "git add" and/or "git commit -a")
kunal@kunalsh:~/Documents/branching$ git add .
kunal@kunalsh:~/Documents/branching$ git commit -m "msg23"

```

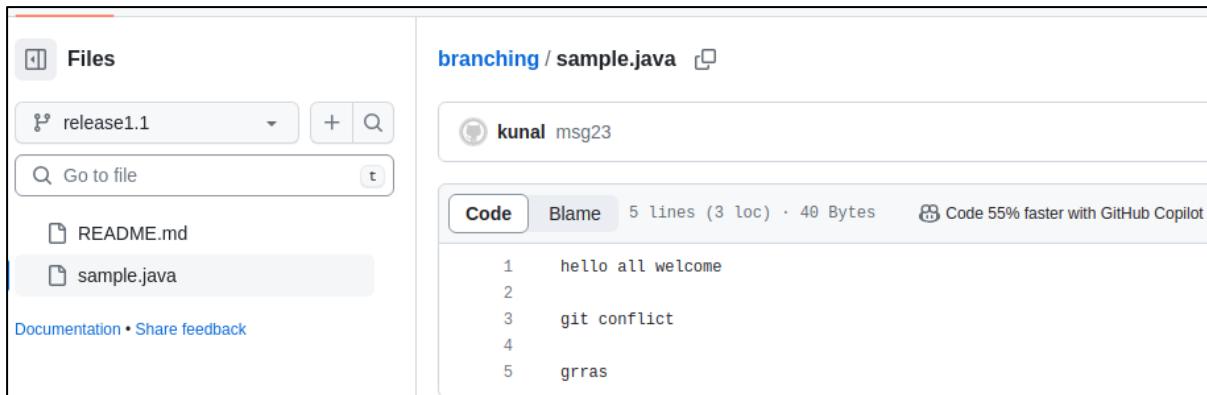
hello all welcome
<<<< HEAD
1012
=====
devops conflict
>>>> release1.2
grras

Remove <<<< HEAD,1012, ===== ,>>>> release1.2
And put devops conflict at exact line no which I want.

Hello all welcome
devops conflict
grras

```
kunal@kunalsh:~/Documents/branching$ git status
On branch release1.1
nothing to commit, working tree clean
```

```
kunal@kunalsh:~/Documents/branching$ git push --all
Username for 'https://github.com': Kunalshiwarkar
Password for 'https://Kunalshiwarkar@github.com':
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 649 bytes | 649.00 KiB/s, done.
Total 7 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/Kunalshiwarkar/branching.git
  c1238ec..f132338  release1.1 -> release1.1
  c1238ec..fb9848e  release1.2 -> release1.2
! [rejected]          master -> master (fetch first)
error: failed to push some refs to 'https://github.com/Kunalshiwarkar/branching.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```



2.New repository example

Owner * Repository name *

 Kunalshiwarkar / conflicts

✓ conflicts is available.

Great repository names are short and memorable. Need inspiration? How about [special-succotash](#) ?

Description (optional)

 **Public**
Anyone on the internet can see this repository. You choose who can commit.

 **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file
This is where you can write a long description for your project. [Learn more about READMEs](#).

Create repository.

 **conflicts** (Public)

 master  1 branch  0 tags

[Go to file](#) [Add file](#) [Code](#)

[Local](#) [Codespaces](#)

Clone [?](#)

[HTTPS](#) [SSH](#) [GitHub CLI](#)

<https://github.com/Kunalshiwarkar/conflic> 

Use Git or checkout with SVN using the web URL.

 [Download ZIP](#)

```
kunal@kunalsh:~/Documents$ git clone https://github.com/Kunalshiwarkar/conflicts.git
Cloning into 'conflicts'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
kunal@kunalsh:~/Documents$ cd conflicts/
kunal@kunalsh:~/Documents/conflicts$ git branch
* master
kunal@kunalsh:~/Documents/conflicts$ ls -a
. .. .git README.md
kunal@kunalsh:~/Documents/conflicts$ vim sample.java
kunal@kunalsh:~/Documents/conflicts$ cat sample.java
hello

a=10

c=20

result=30
kunal@kunalsh:~/Documents/conflicts$ git branch
* master
```

```
kunal@kunalsh:~/Documents/conflicts$ git status
On branch master
Your branch is up to date with 'origin/master'.

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

nothing added to commit but untracked files present (use "git add" to track)
kunal@kunalsh:~/Documents/conflicts$ git add .
kunal@kunalsh:~/Documents/conflicts$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   sample.java

kunal@kunalsh:~/Documents/conflicts$ git commit -m "msg"
```

```
kunal@kunalsh:~/Documents/conflicts$ git checkout -b dev
Switched to a new branch 'dev'
kunal@kunalsh:~/Documents/conflicts$ ls -a
. .. .git README.md sample.java
kunal@kunalsh:~/Documents/conflicts$ cat sample.java
hello

a=10 ←
c=20

result=30
kunal@kunalsh:~/Documents/conflicts$ vim sample.java ←
kunal@kunalsh:~/Documents/conflicts$ 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:   sample.java

no changes added to commit (use "git add" and/or "git commit -a")
kunal@kunalsh:~/Documents/conflicts$ git add .
kunal@kunalsh:~/Documents/conflicts$ git commit -m "msg123"
```

a=10 and result
remove put c=30
at exact line no.

hello
c=30

```
kunal@kunalsh:~/Documents/conflicts$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
kunal@kunalsh:~/Documents/conflicts$ git merge dev
Updating 43492c2..fda42ca
Fast-forward
 sample.java | 4 +---
 1 file changed, 1 insertion(+), 3 deletions(-)
```

```
kunal@kunalsh:~/Documents/conflicts$ git push
Username for 'https://github.com': Kunalshiwarkar
Password for 'https://Kunalshiwarkar@github.com':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 572 bytes | 572.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Kunalshiwarkar/conflicts.git
  9960c03..fda42ca  master -> master
kunal@kunalsh:~/Documents/conflicts$ ls -a
.  ..  .git  README.md  sample.java
```

```
kunal@kunalsh:~/Documents/conflicts$ vim sample.java
kunal@kunalsh:~/Documents/conflicts$ git branch
  dev
* master
kunal@kunalsh:~/Documents/conflicts$ git status
On branch master
Your branch is up to date with 'origin/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:   sample.java

no changes added to commit (use "git add" and/or "git commit -a")
kunal@kunalsh:~/Documents/conflicts$ git add .
kunal@kunalsh:~/Documents/conflicts$ git status
On branch master
Your branch is up to date with 'origin/master'.

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

kunal@kunalsh:~/Documents/conflicts$ git commit -m "msg69"
```



```
kunal@kunalsh:~/Documents/conflicts$ git checkout dev
Switched to branch 'dev'
kunal@kunalsh:~/Documents/conflicts$ ls -a
. .. .git README.md sample.java
kunal@kunalsh:~/Documents/conflicts$ cat sample.java
hello

c=30

kunal@kunalsh:~/Documents/conflicts$ vim sample.java
kunal@kunalsh:~/Documents/conflicts$ 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:   sample.java

no changes added to commit (use "git add" and/or "git commit -a")
kunal@kunalsh:~/Documents/conflicts$ git add .
kunal@kunalsh:~/Documents/conflicts$ git status
On branch dev
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   sample.java

kunal@kunalsh:~/Documents/conflicts$ git commit -m "msg45"
```

```
kunal@kunalsh:~/Documents/conflicts$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
kunal@kunalsh:~/Documents/conflicts$ git merge dev
Auto-merging sample.java
CONFLICT (content): Merge conflict in sample.java
Automatic merge failed; fix conflicts and then commit the result.
kunal@kunalsh:~/Documents/conflicts$ vim sample.java
kunal@kunalsh:~/Documents/conflicts$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

  You have unmerged paths.
    (fix conflicts and run "git commit")
    (use "git merge --abort" to abort the merge)

  Unmerged paths:
    (use "git add <file>..." to mark resolution)
      both modified:   sample.java

no changes added to commit (use "git add" and/or "git commit -a")
kunal@kunalsh:~/Documents/conflicts$ git add .
kunal@kunalsh:~/Documents/conflicts$ git commit -m "msg3"
```

hello
c=660

hello
|<<<<< HEAD
|c=40
=====
|c=660
|>>>>> dev

hello
c=660

```
kunal@kunalsh:~/Documents/conflicts$ git push
Username for 'https://github.com': Kunalshiwarkar
Password for 'https://Kunalshiwarkar@github.com':
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 786 bytes | 786.00 KiB/s, done.
Total 9 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Kunalshiwarkar/conflicts.git
  fda42ca..181dcd2  master -> master
```



How to merge single commit from one branch to another branch is called git cherry-pick.

```
kunal@kunalsh:~/Documents/conflicts$ git checkout dev
Switched to branch 'dev'
kunal@kunalsh:~/Documents/conflicts$ echo "msg123" >> abc.java
kunal@kunalsh:~/Documents/conflicts$ echo "msg123" >> abc1.java
kunal@kunalsh:~/Documents/conflicts$ echo "msg123" >> abc2.java
kunal@kunalsh:~/Documents/conflicts$ echo "msg123" >> abc3.java
kunal@kunalsh:~/Documents/conflicts$ git status
On branch dev
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    abc.java
    abc1.java
    abc2.java
    abc3.java

nothing added to commit but untracked files present (use "git add" to track)
kunal@kunalsh:~/Documents/conflicts$ git add .
kunal@kunalsh:~/Documents/conflicts$ git commit -m "msg12"
```

```
kunal@kunalsh:~/Documents/conflicts$ git log
commit 25dfe89c264c30fa900ab9feaaf013a7bb06ab23 (HEAD -> dev)
Author: kunal <kunal@kunalsh.myguest.virtualbox.org>
Date:   Sun Oct 22 01:01:52 2023 +0530

msg12

commit d99f00c7d6a36cfa8451fd9a302445296730e717
Author: kunal <kunal@kunalsh.myguest.virtualbox.org>
Date:   Sun Oct 22 00:08:24 2023 +0530

msg45

commit fda42ca4a11c3f459a49efb9f7636b2405f0ccdc
Author: kunal <kunal@kunalsh.myguest.virtualbox.org>
Date:   Sat Oct 21 23:59:52 2023 +0530

msg123

commit 43492c2ddb5b905d73addc4ccb16555839492d80
Author: kunal <kunal@kunalsh.myguest.virtualbox.org>
Date:   Sat Oct 21 23:57:00 2023 +0530

msg

commit 9960c03dcd26928fa9d09be9b43b41a83c7454ef
Author: Kunalshiwarkar <147299707+Kunalshiwarkar@users.noreply.github.com>
```

merge single commit from one branch to another branch is called git cherry-pick.

```
kunal@kunalsh:~/Documents/conflicts$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
kunal@kunalsh:~/Documents/conflicts$ git branch
  dev
* master
kunal@kunalsh:~/Documents/conflicts$ git cherry-pick 25dfe89c264c30fa900ab9feaaf013a7bb06ab23
[master fc9e7a7] msg12
Date: Sun Oct 22 01:01:52 2023 +0530
```

```
kunal@kunalsh:~/Documents/conflicts$ ls -a
. .. abc1.java abc2.java abc3.java abc.java .git README.md sample.java
kunal@kunalsh:~/Documents/conflicts$ 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
kunal@kunalsh:~/Documents/conflicts$ git merge dev
Merge made by the 'ort' strategy.
```

Press ctrl+x

```
/home/kunal/Documents/conflicts/.git/MERGE_MSG
Merge branch 'dev'
# Please enter a commit message to explain why this merge is necessary
# especially if it merges an updated upstream into a topic branch.
#
# Lines starting with '#' will be ignored, and an empty message aborts
# the commit.
```

```
^G Help      ^O Write Out   ^W Where Is   ^K Cut       ^T Execute
^X Exit      ^R Read File   ^\ Replace    ^U Paste     ^J Justify
```

```
kunal@kunalsh:~/Documents/conflicts$ git status
On branch master
Your branch is ahead of 'origin/master' by 3 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
kunal@kunalsh:~/Documents/conflicts$ git push
Username for 'https://github.com': Kunalshiwarkar
Password for 'https://Kunalshiwarkar@github.com':
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 653 bytes | 653.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/Kunalshiwarkar/conflicts.git
  181dcd2..e07a44d  master -> master
```

conflicts (Public)

master 1 branch 0 tags

Go to file Add file Code

Commit	Author	Time	Commits
e07a44d	kunal Merge branch 'dev'	6 minutes ago	9 commits
181dcd2	Initial commit	1 hour ago	
abc.java	msg12	17 minutes ago	
abc1.java	msg12	17 minutes ago	
abc2.java	msg12	17 minutes ago	
abc3.java	msg12	17 minutes ago	
sample.java	msg3	1 hour ago	

Git rebase

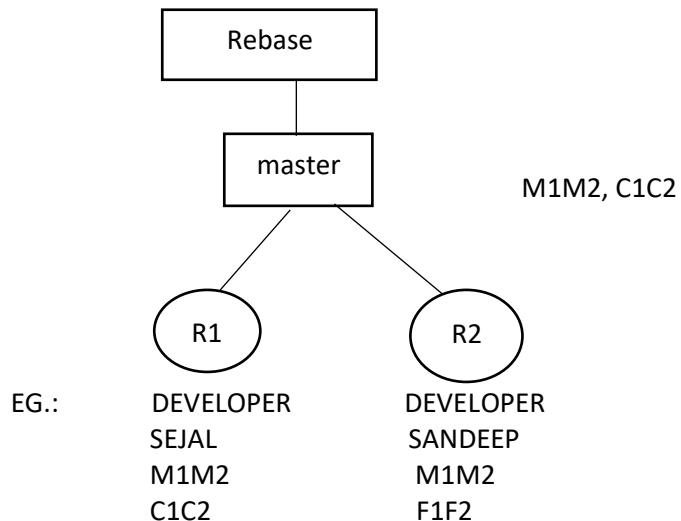
Rebasing is the process of moving or combining a sequence of commits to a new base commit.

The base of master and feature branching is same.

Parent → child

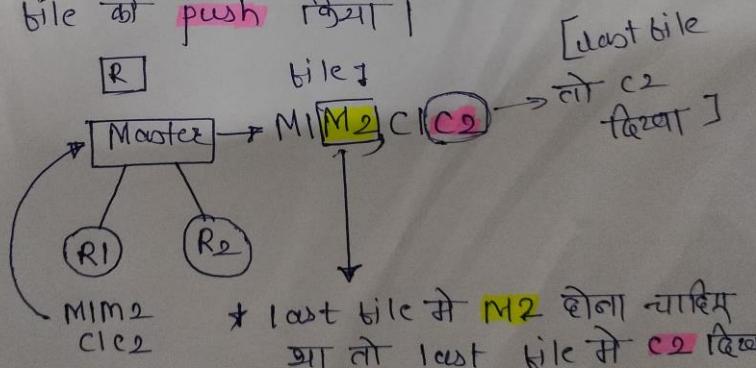
Master base → feature base used rebase for base matching .

There is no need of merge.



What is Rebase

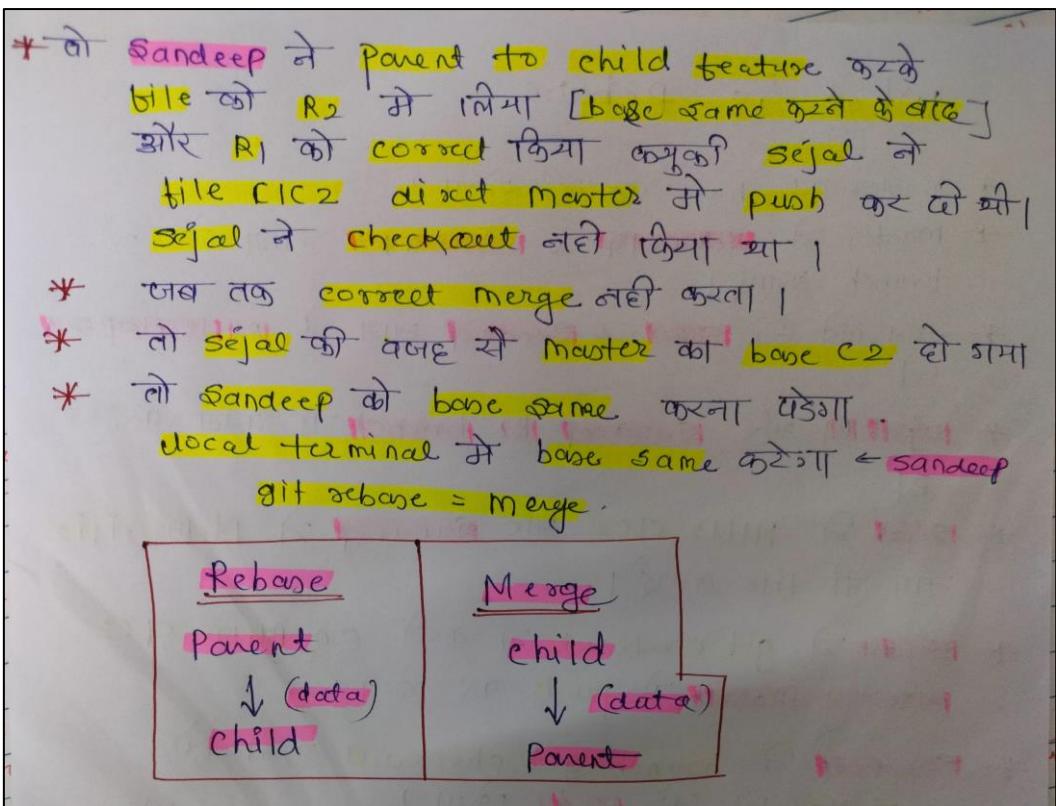
- * Master का base घट की रहेगा।
- * Master ने **release R1** & **select R2** नाम के 2 branch बनाए।
- * मान लेते हैं, **Sejal** & **Sandeep** नाम के 2 **developers** हैं।
- * **Sejal R1** और **Sandeep R2** branch में काम कर रहे हैं।
- * **Sejal** ने **M1 M2, C1 C2** और **Sandeep** ने **M1 M2, F1 F2** नाम की file बनाई।
- * **Sejal** ने **git checkout** वा करते हुए **M1 M2 C1 C2** directly **Master** में **push** कर दिया।
- * **Sandeep** ने **branch git checkout** करते हुए **M1 M2** file को **push** किया।



- * data अलग हो गया | \rightarrow [Sandeep का M2 push दिया गया और Sejal C2]
- * master में मेरे data नहीं चलेगा।

rebase means master & feature branch base is same

- * तो अब Sandeep [base] का use करके, base को same करोगा



EG.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *



Repository name *

conflict-resolve

conflict-resolve is available.

Great repository names are short and memorable. Need inspiration? How about [laughing-octo-pancake](#) ?

Description (optional)

Anyone on the internet can see this repository. You choose who can commit.

You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Create new repository.

conflict-resolve (Public)

master 1 branch 0 tags

Kunalshiwarkar Initial commit

README.md Initial commit

README.md

conflict-resolve

Local Codespaces

Clone

HTTPS SSH GitHub CLI

<https://github.com/Kunalshiwarkar/conflic>

Use Git or checkout with SVN using the web URL.

Download ZIP

```
kunal@kunalsh:~/Documents$ git clone https://github.com/Kunalshiwarkar/conflict-resolve.git
Cloning into 'conflict-resolve'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
kunal@kunalsh:~/Documents$ cd conflict-resolve/
kunal@kunalsh:~/Documents/conflict-resolve$ git branch
* master
kunal@kunalsh:~/Documents/conflict-resolve$ ls -a
.  ..  .git  README.md
kunal@kunalsh:~/Documents/conflict-resolve$ vim sample.java
kunal@kunalsh:~/Documents/conflict-resolve$ cat sample.java
hello

a=10

b=20

c=30

d=40
```

```
kunal@kunalsh:~/Documents/conflict-resolve$ git status
On branch master
Your branch is up to date with 'origin/master'.

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

nothing added to commit but untracked files present (use "git add" to track)
kunal@kunalsh:~/Documents/conflict-resolve$ git add .
kunal@kunalsh:~/Documents/conflict-resolve$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   sample.java

kunal@kunalsh:~/Documents/conflict-resolve$ git commit -m "java"
```

```
kunal@kunalsh:~/Documents/conflict-resolve$ git checkout -b dev
Switched to a new branch 'dev'
kunal@kunalsh:~/Documents/conflict-resolve$ ls -a
. . . .git README.md sample.java
kunal@kunalsh:~/Documents/conflict-resolve$ vim sample.java
kunal@kunalsh:~/Documents/conflict-resolve$ cat sample.java
hello

a=10

b=20

c=30

d=50
kunal@kunalsh:~/Documents/conflict-resolve$ 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:   sample.java

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

```
kunal@kunalsh:~/Documents/conflict-resolve$ git add .
kunal@kunalsh:~/Documents/conflict-resolve$ git status
On branch dev
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   sample.java

kunal@kunalsh:~/Documents/conflict-resolve$ git commit -m "change"
```

```
kunal@kunalsh:~/Documents/conflict-resolve$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
kunal@kunalsh:~/Documents/conflict-resolve$ git branch
  dev
* master
kunal@kunalsh:~/Documents/conflict-resolve$ 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
kunal@kunalsh:~/Documents/conflict-resolve$ git merge dev
Updating d1c1dc7..52a0a7b
Fast-forward
  sample.java | 2 +-
  1 file changed, 1 insertion(+), 1 deletion(-)
```

```
kunal@kunalsh:~/Documents/conflict-resolve$ git push
Username for 'https://github.com': Kunalshiwarkar
Password for 'https://Kunalshiwarkar@github.com':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 586 bytes | 586.00 KiB/s, done
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Kunalshiwarkar/conflict-resolve.git
  5aed40a..52a0a7b master -> master
```

The screenshot shows a GitHub repository page for 'conflict-resolve'. The 'Code' tab is selected. On the left, the 'Files' sidebar shows a dropdown menu set to 'master' and a 'sample.java' file listed. The main area displays the contents of 'sample.java' with the following code:

```
1 hello
2
3 a=10
4
5 b=20
6
7 c=30
8
9 d=50
```

```
kunal@kunalsh:~/Documents/conflict-resolve$ git push --all
Username for 'https://github.com': Kunalshiwarkar
Password for 'https://Kunalshiwarkar@github.com':
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'dev' on GitHub by visiting:
remote:     https://github.com/Kunalshiwarkar/conflict-resolve/pull/new/dev
remote:
To https://github.com/Kunalshiwarkar/conflict-resolve.git
 * [new branch]      dev -> dev
kunal@kunalsh:~/Documents/conflict-resolve$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
kunal@kunalsh:~/Documents/conflict-resolve$ ls -a
. .. .git README.md sample.java
kunal@kunalsh:~/Documents/conflict-resolve$ git branch dev1.1
kunal@kunalsh:~/Documents/conflict-resolve$ git checkout dev
Switched to branch 'dev'
kunal@kunalsh:~/Documents/conflict-resolve$ ls -a
. .. .git README.md sample.java
kunal@kunalsh:~/Documents/conflict-resolve$ vim sample1.1.java
kunal@kunalsh:~/Documents/conflict-resolve$ cat sample1.1.java
hello
kunal@kunalsh:~/Documents/conflict-resolve$ ls -a
. .. .git README.md sample1.1.java sample.java
```

```
kunal@kunalsh:~/Documents/conflict-resolve$ git add .
kunal@kunalsh:~/Documents/conflict-resolve$ git status
On branch dev
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   sample1.1.java

kunal@kunalsh:~/Documents/conflict-resolve$ git commit -m "msg"
[dev 1/1] (1 command)
```

```
kunal@kunalsh:~/Documents/conflict-resolve$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
kunal@kunalsh:~/Documents/conflict-resolve$ vim dev123.java
kunal@kunalsh:~/Documents/conflict-resolve$ cat dev123.java
hello dev
Developer1 directly push into master branch

kunal@kunalsh:~/Documents/conflict-resolve$ ls -a
. .. dev123.java .git README.md sample.java
kunal@kunalsh:~/Documents/conflict-resolve$ git add .
kunal@kunalsh:~/Documents/conflict-resolve$ git commit -m "msg"
```

Git rebase command used

```
kunal@kunalsh:~/Documents/conflict-resolve$ git checkout dev
Switched to branch 'dev'
kunal@kunalsh:~/Documents/conflict-resolve$ git rebase master
Successfully rebased and updated refs/heads/dev.
kunal@kunalsh:~/Documents/conflict-resolve$ ls -a
. .. dev123.java .git README.md sample1.1.java sample.java
kunal@kunalsh:~/Documents/conflict-resolve$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
kunal@kunalsh:~/Documents/conflict-resolve$ ls -a
. .. dev123.java .git README.md sample.java
kunal@kunalsh:~/Documents/conflict-resolve$ git rebase dev
Successfully rebased and updated refs/heads/master.
kunal@kunalsh:~/Documents/conflict-resolve$ ls -a
. .. dev123.java .git README.md sample1.1.java sample.java
kunal@kunalsh:~/Documents/conflict-resolve$ git push --all
Username for 'https://github.com': Kunalshiwarkar
Password for 'https://Kunalshiwarkar@github.com':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 540 bytes | 540.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
```

master 3 branches 0 tags Go to file Add file Code

Your master branch isn't protected Protect this branch

kunal msg c37e8af 8 minutes ago 5 commits

README.md	Initial commit	38 minutes ago
dev123.java	msg	12 minutes ago
sample.java	change	28 minutes ago
sample1.1.java	msg	8 minutes ago

conflict-resolve Public Pin Unwatch

dev 3 branches 0 tags Go to file Add file Code

This branch is up to date with master. Contribute

kunal msg c37e8af 9 minutes ago 5 commits

README.md	Initial commit	39 minutes ago
dev123.java	msg	13 minutes ago
sample.java	change	29 minutes ago
sample1.1.java	msg	9 minutes ago

conflict-resolve Public Pin Unwatch

dev1.1 3 branches 0 tags Go to file Add file Code

This branch is 2 commits behind master. Contribute

kunal change 52a0a7b 30 minutes ago 3 commits

README.md	Initial commit	40 minutes ago
sample.java	change	30 minutes ago

```
kunal@kunalsh:~/Documents/conflict-resolve$ git push --all
Username for 'https://github.com': Kunalshiwarkar
Password for 'https://Kunalshiwarkar@github.com':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 540 bytes | 540.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Kunalshiwarkar/conflict-resolve.git
  52a0a7b..c37e8af  dev -> dev
  52a0a7b..c37e8af  master -> master
* [new branch]      dev1.1 -> dev1.1
kunal@kunalsh:~/Documents/conflict-resolve$ git log --diff-filter=D
kunal@kunalsh:~/Documents/conflict-resolve$ git rm sample.java
rm 'sample.java'
kunal@kunalsh:~/Documents/conflict-resolve$ git status
On branch master
Your branch is up to date with 'origin/master'.
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    sample.java
```

Delete data come
into stage.

```
kunal@kunalsh:~/Documents/conflict-resolve$ git commit -m "file delete"
```

Save data in source.

```
kunal@kunalsh:~/Documents/conflict-resolve$ git push
Username for 'https://github.com': Kunalshiwarkar
Password for 'https://Kunalshiwarkar@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 228 bytes | 228.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Kunalshiwarkar/conflict-resolve.git
  c37e8af..cae4b1e  master -> master
```

```
kunal@kunalsh:~/Documents/conflict-resolve$ git log
commit cae4b1e526b9d663cec355c049dfd64bf452bd45 (HEAD -> master, origin/master, origin/HEAD)
Author: kunal <kunal@kunalsh.myguest.virtualbox.org>
Date:  Mon Oct 23 15:18:47 2023 +0530

  file delete

commit c37e8af2d7a10706edf2232c20c406b9095c4af7 (origin/dev, dev)
Author: kunal <kunal@kunalsh.myguest.virtualbox.org>
Date:  Mon Oct 23 14:56:06 2023 +0530

  msg

commit bbafe823e60bcff74cbbf1e3f127e8590cea51bca
Author: kunal <kunal@kunalsh.myguest.virtualbox.org>
Date:  Mon Oct 23 15:02:01 2023 +0530

  msg

commit 52a0a7be44266c4585722a6c3f911f9845b64b86 (origin/dev1.1, dev1.1)
Author: kunal <kunal@kunalsh.myguest.virtualbox.org>
Date:  Mon Oct 23 14:46:11 2023 +0530

  change

commit d1c1dc78593c940c77952d1d2a0e198c70b24e06
```

```
kunal@kunalsh:~/Documents/conflict-resolve$ git log --diff-filter=D  
commit cae4b1e526b9d663cec355c049dfd64bf452bd45 (HEAD -> master, origin/master, origin/HEAD)  
Author: kunal <kunal@kunalsh.myguest.virtualbox.org>  
Date: Mon Oct 23 15:18:47 2023 +0530  
  
    file delete
```

```
kunal@kunalsh:~/Documents/conflict-resolve$ git checkout cae4b1e526b9d663cec355c049dfd64bf452bd45 sample1.1.java  
Updated 0 paths from a43e05a
```

<Sha value ID > file name

```
kunal@kunalsh:~/Documents/conflict-resolve$ git status  
On branch master  
Your branch is up to date with 'origin/master'.  
  
Changes to be committed:  
(use "git restore --staged <file>..." to unstage)  
  new file:   sample.java
```

```
kunal@kunalsh:~/Documents/conflict-resolve$ git push  
Username for 'https://github.com': Kunalshiwarkar  
Password for 'https://Kunalshiwarkar@github.com':  
Everything up-to-date
```