 **AMITY UNIVERSITY BENGALURU**

SUBJECT:SOURCE CODE MANAGEMENT

****SLOT:L3,L4

NAME:NAVYA.G

ENROLLMENT NUMBER:A86605224256

SUBMITTED TO: DR. MONIT KAPOOR

1.ls – Lists files and directories in the current directory.

Usage: ls or ls -l for details.

2. mkdir – Creates a new directory.

Usage: mkdir project\_folder

3. pwd – Prints the current working directory.

Usage: Just type pwd to see your current path.

4.cd – Changes the directory.

Usage: cd Documents/project

5. vi / vim – Opens a text editor to create or edit files.

Usage: vi main.py (press i to insert,wq to save and exit

6.git init – Initializes a new Git repository.

7. git clone <repo> – Clones an existing repository from remote (like GitHub).

8. git status – Shows the current status of the working directory.

9.git add <file> – Adds file(s) to the staging area.

10.git commit -m "message" – Saves the staged changes with a message.

11. git push – Uploads local commits to a remote repository.

12. git pull – Downloads updates from the remote repository and merges them.

13. git log – Shows the commit history.

**GIT INSTALLATION**

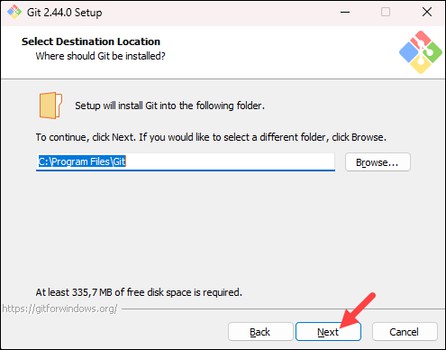
1. wNavigate to the [**official Git downloads page**](https://git-scm.com/download/win)and click the download link for the latest Git version for Windows:



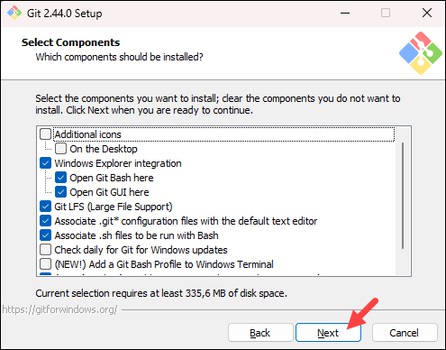
1. Double-click the downloaded [**file**](https://phoenixnap.com/glossary/what-is-a-file)to extract and launch the installer
2. Review the [**GNU General Public License**](https://phoenixnap.com/glossary/gnu-general-public-license), and when you are ready to install, click **Next**.



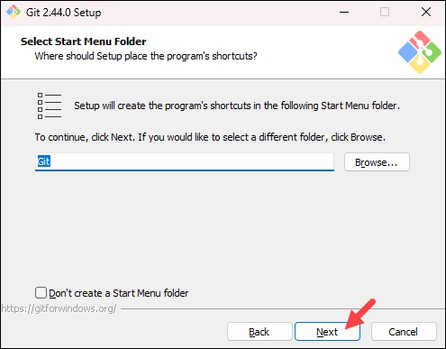
4.The installer prompts you for an installation location. Leave the default one unless you want to change it, and click **Next**.



5.In the component selection screen, leave the defaults unless you need to change them and click **Next**.

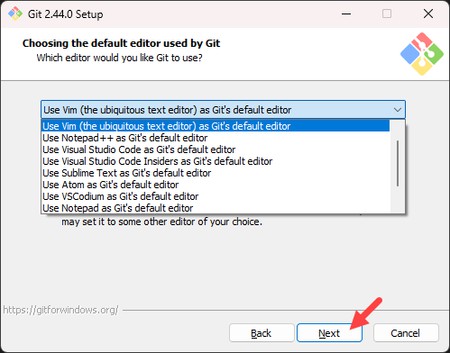


6.The installer offers to create a start menu [**folder**](https://phoenixnap.com/glossary/what-is-a-folder). Click **Next** to accept and proceed to the next step.



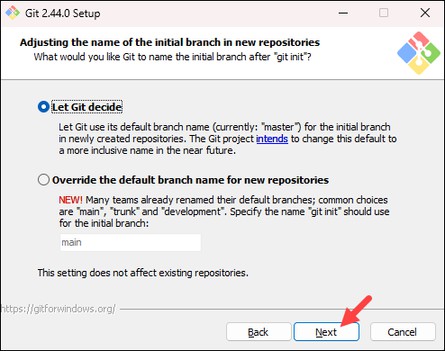
1. Select a text editor you want to use with Git. Use the drop-down menu to select Notepad++ (or whichever text editor you prefer) and click **Next**.

If you prefer to use a CLI text editor in [**Git Bash**](https://phoenixnap.com/kb/what-is-git-bash), select [**nano**](https://phoenixnap.com/kb/use-nano-text-editor-commands-linux)or [**Vim**](https://phoenixnap.com/kb/vim-commands-cheat-sheet)from the list.

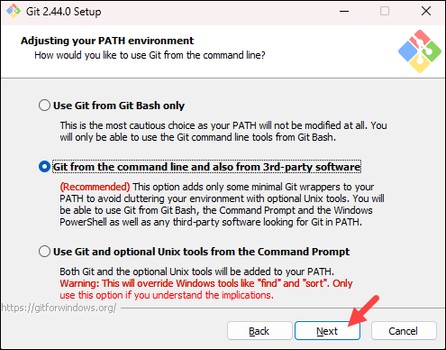


8.The next step allows you to choose a different name for your initial branch. The default is **master**.

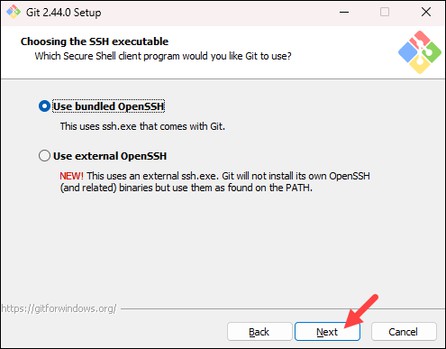
Unless you are working in a team that requires a different name, leave the default option and click **Next.**



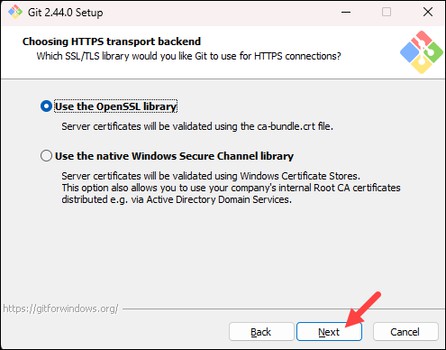
9. The next step allows you to change the **PATH environment**. The **PATH** is the default set of [**directories**](https://phoenixnap.com/glossary/what-is-a-directory)included when you run a command from the command line. Keep the middle (recommended) selection and click **Next**.



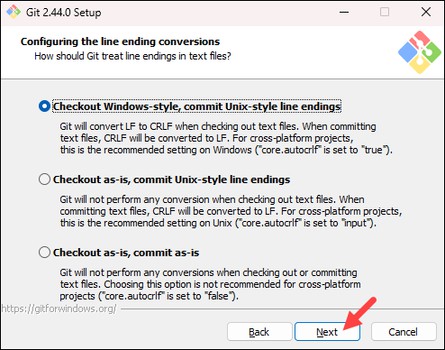
10. The installer prompts you to select the SSH client for Git to use. Git already comes with its own SSH client, so if you don't need a specific one, leave the default option and click **Next.**

****

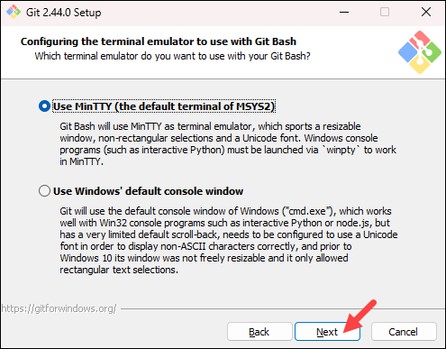
1. The next option relates to server certificates. The default option is recommended for most users. If you work in an Active Directory environment, you may need to switch to Windows Store certificates. Select your preferred option

 and click **Next**.

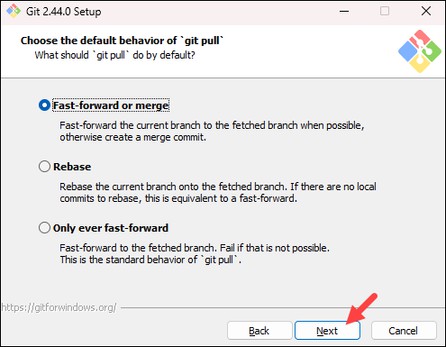
1. The following selection configures line-ending conversion, which relates to the way data is formatted. The default selection is recommended for Windows. Click **Next** to proceed.



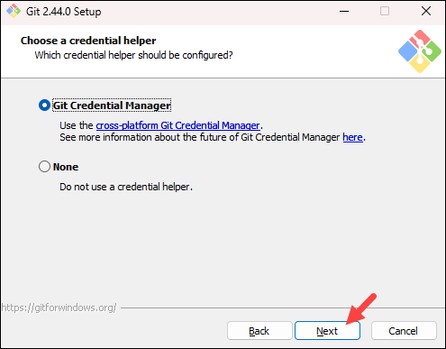
1. Choose the [**terminal emulator**](https://phoenixnap.com/glossary/terminal-emulation)you want to use. The default MinTTY is recommended for its features. Click **Next** to continue.



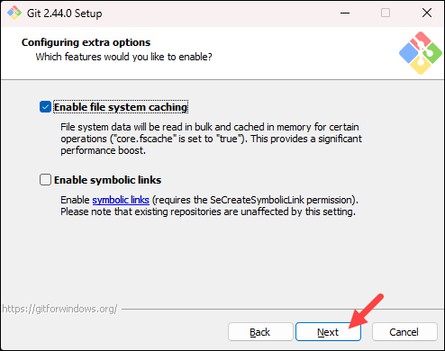
1. The next step allows you to choose what the **git pull** command will do. The default option is recommended unless you specifically need to change its behavior. Click **Next** to continue with the installation.



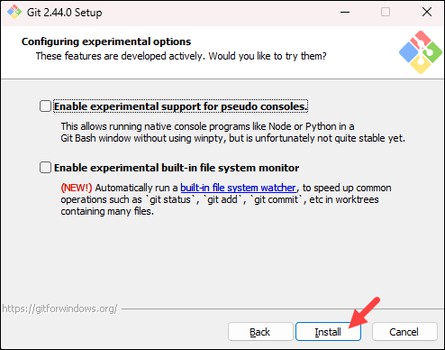
1. The next step is to choose which credential helper to use. Git uses credential helpers to fetch or save credentials. The default option is the most stable one. Select your preferred credential manager and click **Next**.



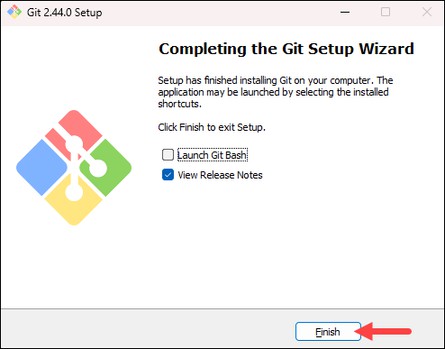
1. The next step lets you decide which extra options to enable. If you use [**symbolic links**](https://phoenixnap.com/kb/symbolic-link-linux), which represent shortcuts for the command line, tick the box. Keep [**file system**](https://phoenixnap.com/glossary/filesystem)caching checked and click **Next**.



1. Depending on which Git version you are installing, it may offer to install experimental features. At the time this article was written, the installer offered options to include support for pseudo controls and a built-in file system monitor. For the most stable operation, do not install experimental features and click **Install**.



1. Once the installation is complete, tick the boxes to view the Release Notes or launch Git Bash if you want to start using Git right away, and click **Finish**.



After opening Git Bash

#command

$git --version

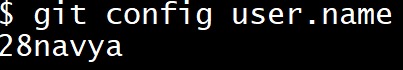


Git Configuration:

#command

$git config user.name “28navya”

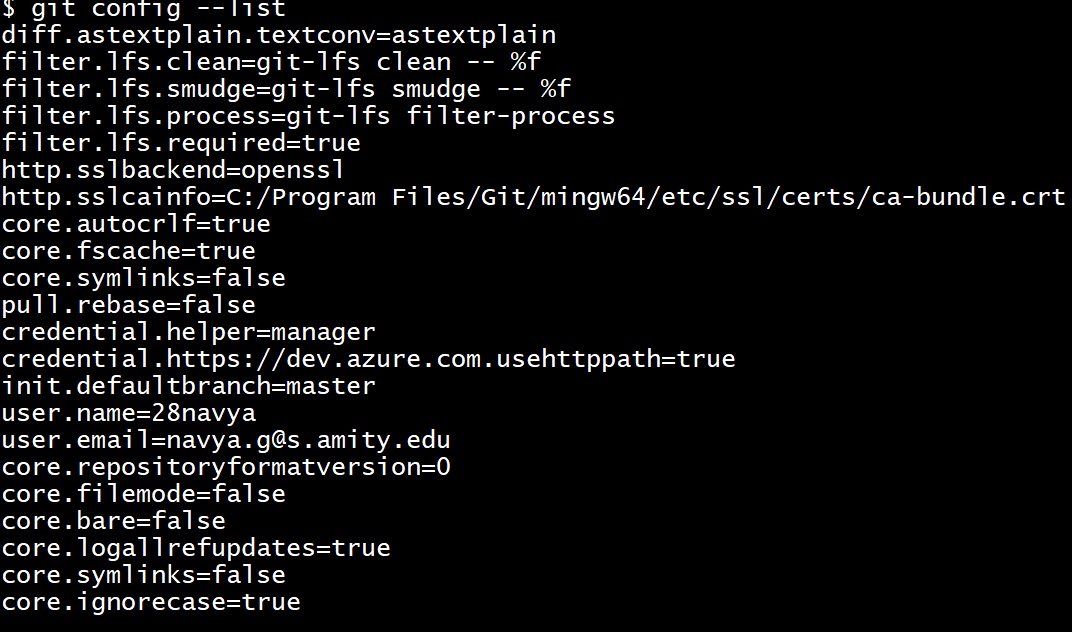
$git config user.email [navya.g@s.amity.edu](mailto:navya.g@s.amity.edu)



$config user.email



$git config --list



**Managing Files and Directories in git**

#commands

$mkdir 124046

$cd 124046

$vi hello.cpp

$cat hello.cpp

$git init

$git status

$git add hello.cpp

$git status

$git commit -m “It is c++ code on hello world”

$git log

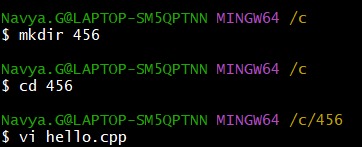
$git log --oneline

#output

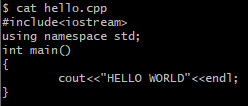
$mkdir 124046

$cd 124046

$vi hello.cpp(//After this command we should press I to type anything into the file.To come out of the file we should press Esc and :wq to exit )



$cat hello.cpp(//It will print or show what ever written in that file)



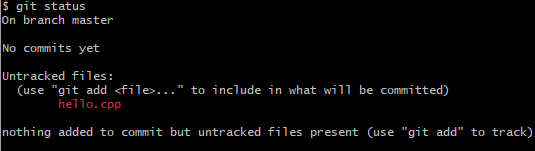
$git init



$ls(It shows the the files)



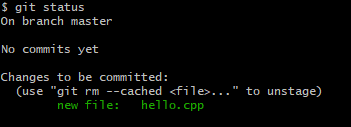
$git status(It shows the files is tracked or untracked but at present it is untracked)



$git add hello.cpp



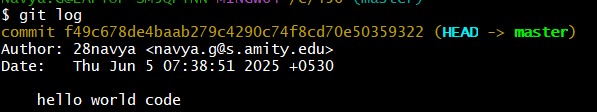
$git status(Now it is tracked)



$git commit -m “code on hello world”



$git log(It shows in long format with authour ,date ,time,email)



$git log --oneline(It display in short format)



**Creating a branch**

#commands

$git branch

$git branch test

$git checkout test

$vi simple.html

$cat simple.html

$git init

$git status

$git add simple.html

$git status

$git commit -m “simple code in html”

$git log --oneline

#output

$git branch(it shows branch, now we have only one branch master)

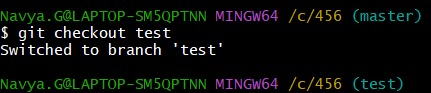


$git branch test(It creats one more branch test)

$git branch

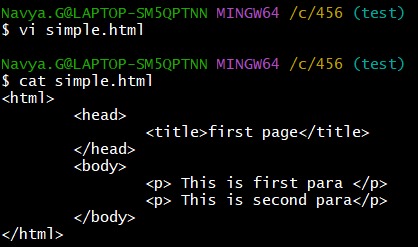


$git checkout test (It changes from master to test)



$vi simple.html

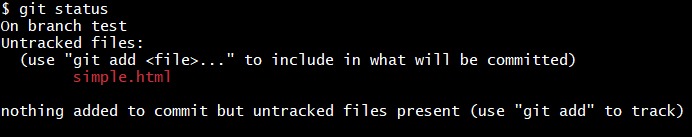
$cat simple.html



$git init



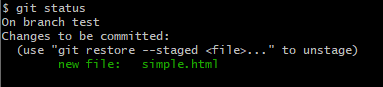
$git status



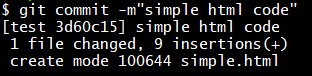
$git add simple.html



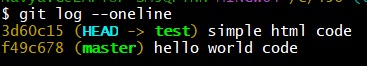
$git status



$git commit -m “simple code in html”



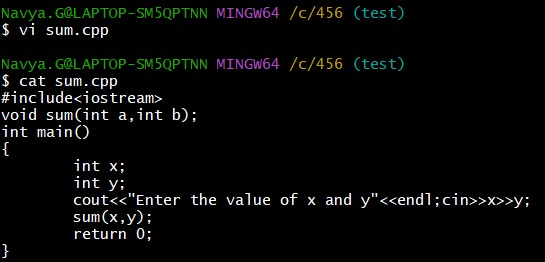
$git log --oneline



Creating one more file in test branch by using the same commands the output will be:

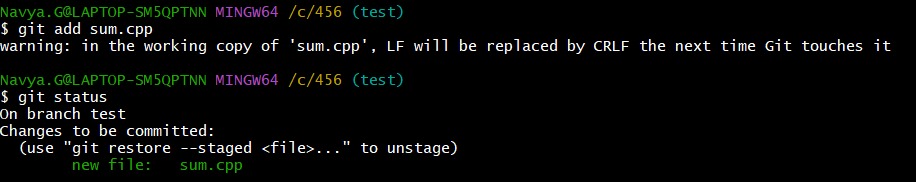
$vi sum.cpp

$cat sum.cpp



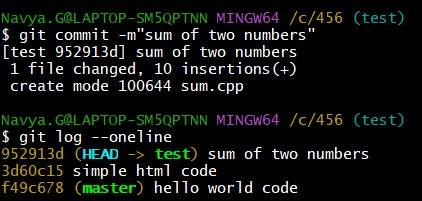
$git add sum.cpp

$git status

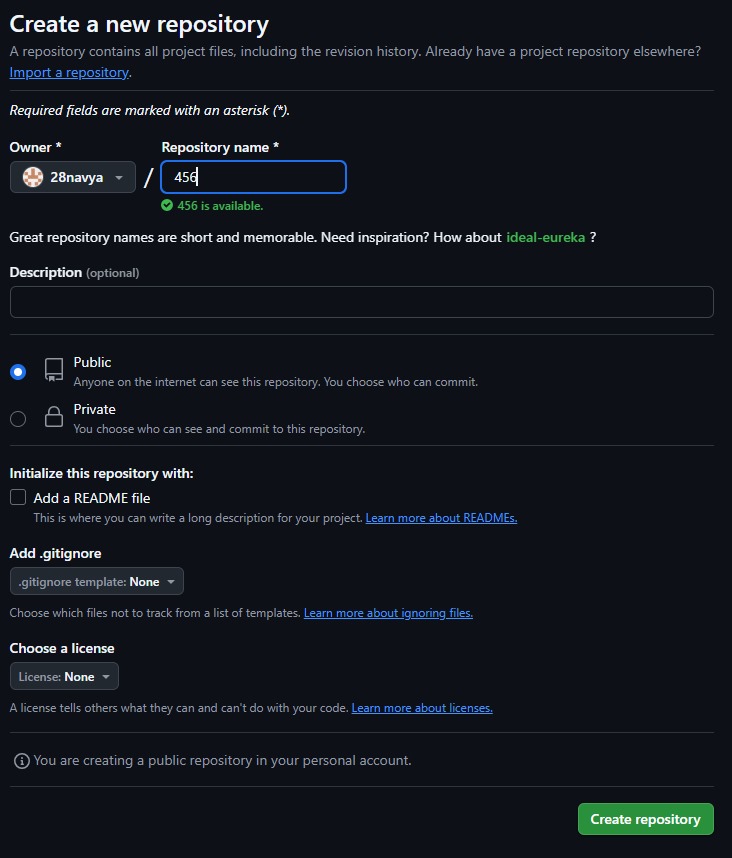


$git commit -m “code on sum of two number”

$git log --oneline



Creating a new repository

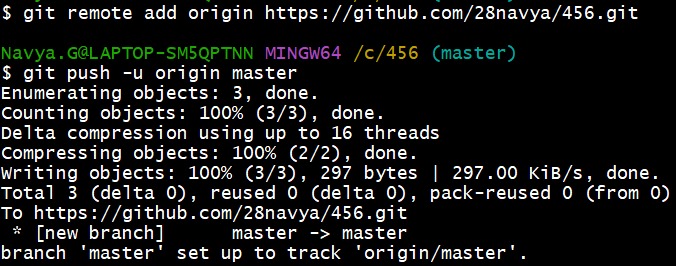


Pushing the files which are in the master and test

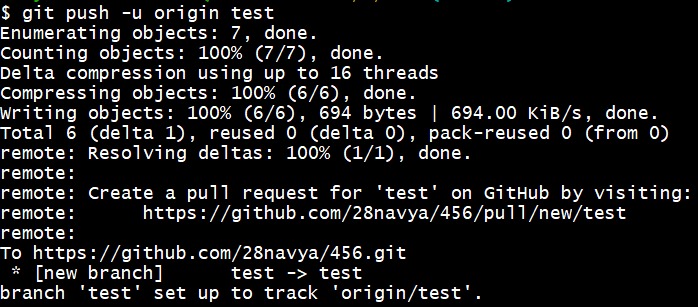
#commands

$git remote add origin <https://github.com/varji123/notes.git>

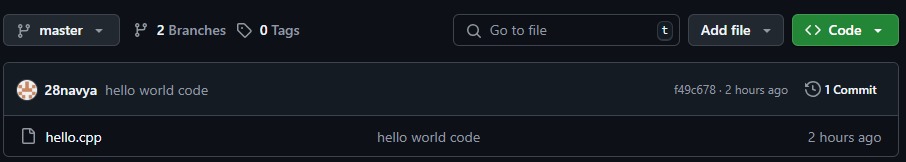
$git push -u origin master(This is from master)

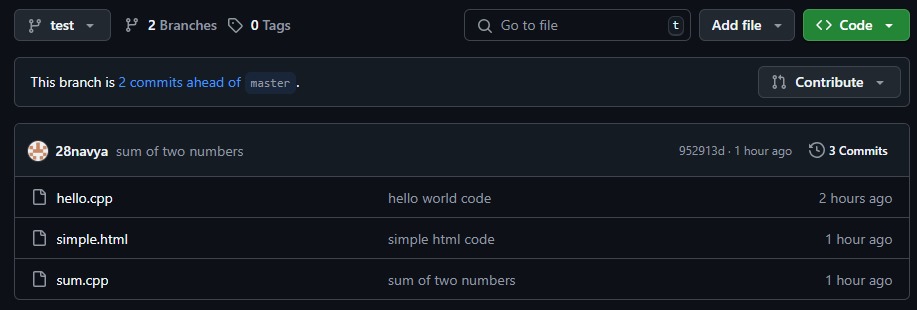


$git push -u origin test



Now the files in master and test will appear on git hub repo in notes after the command git push -u origin test and git push -u origin master





Pull Request:

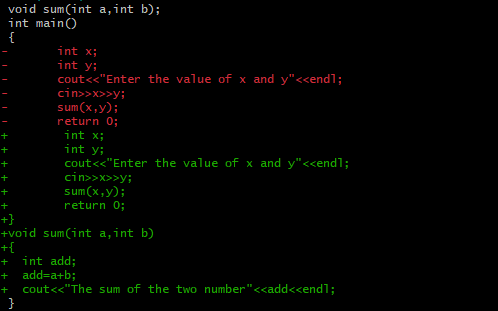
$git pull https://github.com/28navya/456.git



Git diff

It compare the commits or files

$git diff 952913d F49c678



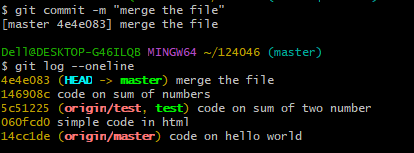
MERGE THE FILES

$git merge

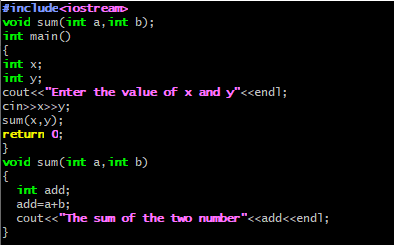
$git conflict master /test

$git mergetool

$git commit -m “merge the file”(After merging the file from two different branch)



This the file which is merged together



Fork and cloning of the repository

In git hub we should search the repo which should to be fork and cloned

After opening the repo we will get the fork option



We cannot fork our own repository

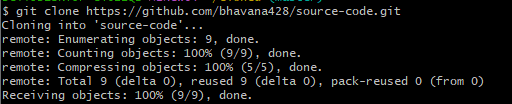
**CLONE:**

#Commands

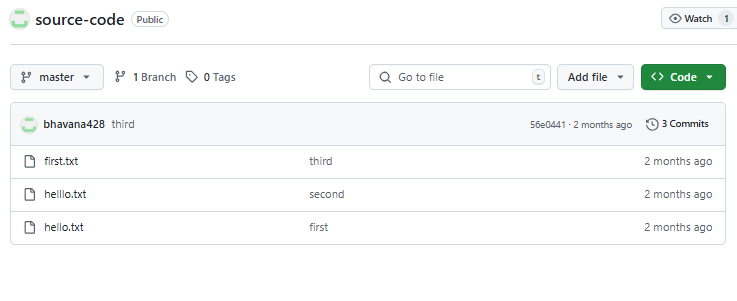
$mkdir cloned

$cd cloned

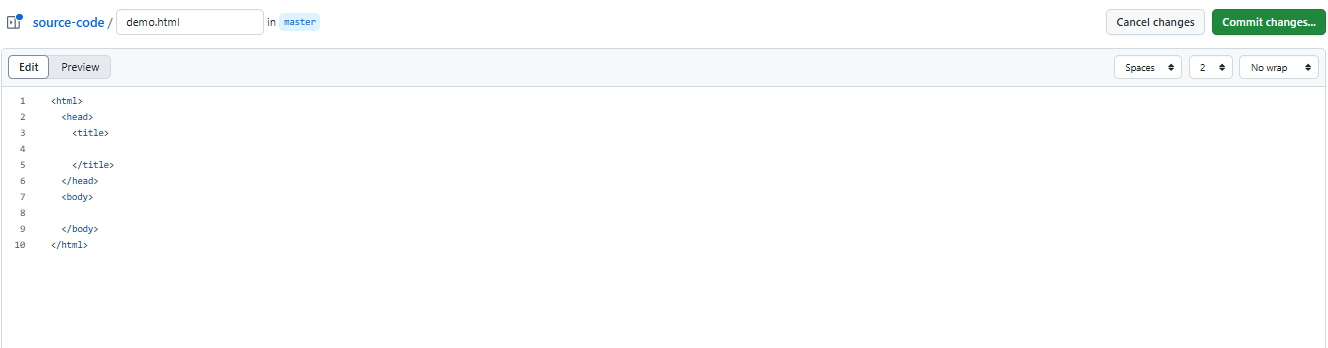
$git clone https://github.com/bhavana428/L34.git



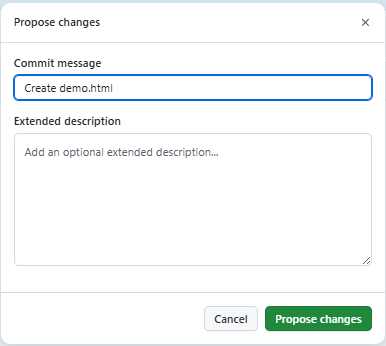
Creating a pull request in git hub:



Add file (adding the file name as demo.html we get)



After typing we should commit changes



After propose changes we will get option of create pull request

