

GIT AND GITHUB TUTORIAL

- **Git** is a distributed version control system that helps us track changes in files and coordinate work on those files among multiple people.
- **GitHub** is a platform that uses Git for version control. It allows us to store our Git repositories remotely, collaborate with others, and manage your codebase.

Git and Github Commands:

1. Initialize a Git repository:

This command initializes a Git repository in the current directory. This will create a .git directory that contains all the necessary metadata for version control.

git init

```
HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT
$ git init
Initialized empty Git repository in C:/Users/HP/Desktop/LOS GIT/.git/

HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT (master)
$ |
```

2. Check the status:

This command shows the status of the files in our repository (whether they're modified, staged for commit, or untracked).

git status

```
HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT (master)
$
```

3. Add changes to staging area:

Before committing changes, we need to add them to the staging area.

git add <filename> # Adds a specific file

git add # Adds all changes in the current directory

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

HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT (master)
$ |
```

4. Commit changes:

We can commit our changes with a message describing the changes.

git commit -m "file updated" # "-m" lets you add a message directly

```
HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT (master)
$ git commit -m "file updated"
[master (root-commit) 6d4a6eb] file updated
1 file changed, 1 insertion(+)
create mode 100644 file.txt

HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT (master)
$ |
```

5. View commit history:

We can view the commit history of our repository using:

git log

```
HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT (master)
$ git log
commit 6d4a6eb0bfb986daab9491d3c48be1ac682d3b25 (HEAD -> master)
Author: Arushi <pandeyarushi369@gmail.com>
Date: Wed Apr 9 12:08:34 2025 +0530

    file updated

HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT (master)
$ |
```

6. Create a new branch:

Git branches let us work on different versions of our project.

git branch <branch-name> # Creates a new branch

```
HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT (master)
$ git branch "wipro"

HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT (master)
$ git branch
HEAD      master    wipro

HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT (master)
$ git branch
HEAD      master    wipro

HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT (master)
$ git branch
* master
  wipro
```

git checkout <branch-name> # Switches to that branch

git checkout -b <branch-name> # Creates and switches to the new branch

```
HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT (master)
$ git checkout wipro
Switched to branch 'wipro'

HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT (wipro)
$ |
```

7. Merge branches:

When we're ready to bring changes from one branch into another (typically after a feature branch), we can merge it.

git checkout main # Switch to the branch you want to merge changes into

git merge <branch-name> # Merge changes from another branch into the current branch

```
HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT (wipro)
$ git merge wipro
Already up to date.

HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT (wipro)
$
```

8. Push changes to remote (GitHub):

Once we're done committing locally, we'll push our changes to a remote GitHub repository.

git push origin <branch-name> # Pushes changes to the remote repository.

9. Pull changes from remote (GitHub):

To get the latest changes from a remote repository (GitHub):

git pull origin <branch-name> # Pulls changes from the remote repository

10. Clone a repository:

To copy a remote repository to our local machine, we use:

git clone <repository-url>

Example:

git clone : <https://github.com/Arushipriya/Wipro.git>

```
HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT (master)
$ mkdir clone

HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT (master)
$ cd clone/

HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT/clone (master)
$ git clone https://github.com/Arushipriya/Wipro.git
Cloning into 'wipro'...
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 13 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (13/13), done.

HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT/clone (master)
$
```

11. Create a new empty file or update the timestamp of an existing file:

touch hello.txt

```
HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT
$ touch file.txt

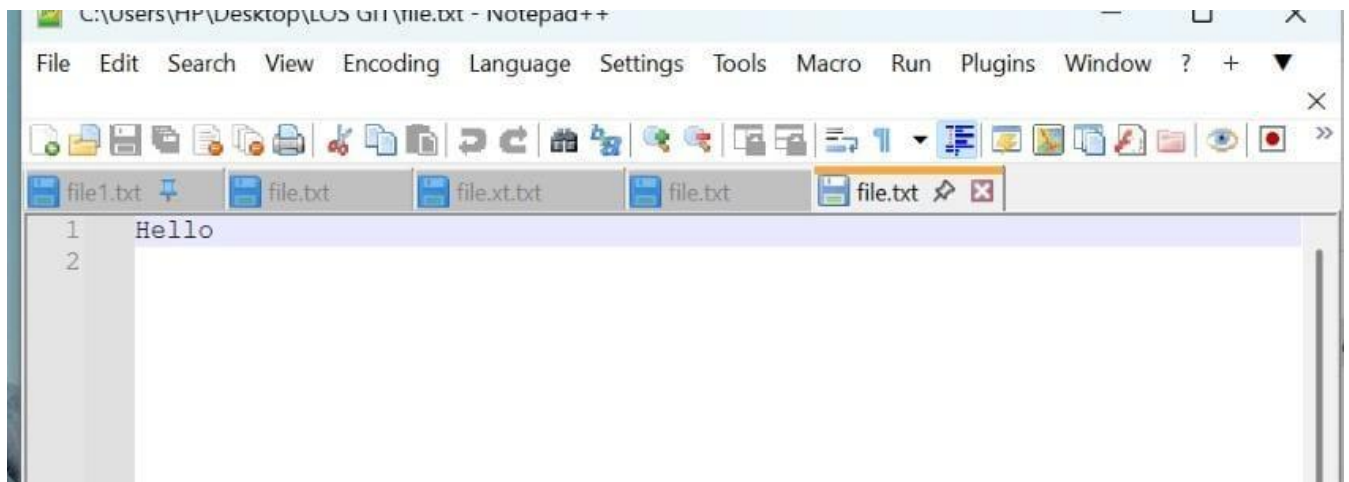
HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT
$
```

12. To print text to the terminal or write it to a file.

`echo "Hello" > file.txt`



```
HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT (master)
$ echo Hello >> file.txt
HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT (master)
$ |
```



GitHub Basics:.

1. Create a new repository on GitHub:

- Go to GitHub and click on **New** repository.
- Fill in the repository name, description, and choose visibility (public or private).
- Click **Create repository**.

2. Link your local Git repository to GitHub:

Once we have a GitHub repository created, we will link it to our local repository by adding a remote URL.

`git remote add origin https://github.com/username/repository.git`

3. Push your local repository to GitHub:

After linking the remote, push your changes to GitHub:

git push -u origin main # Pushes to the main branch on GitHub

4. Fetch changes from GitHub:

To bring the latest changes from the GitHub repository (without merging them into your local files), we use:

git fetch origin

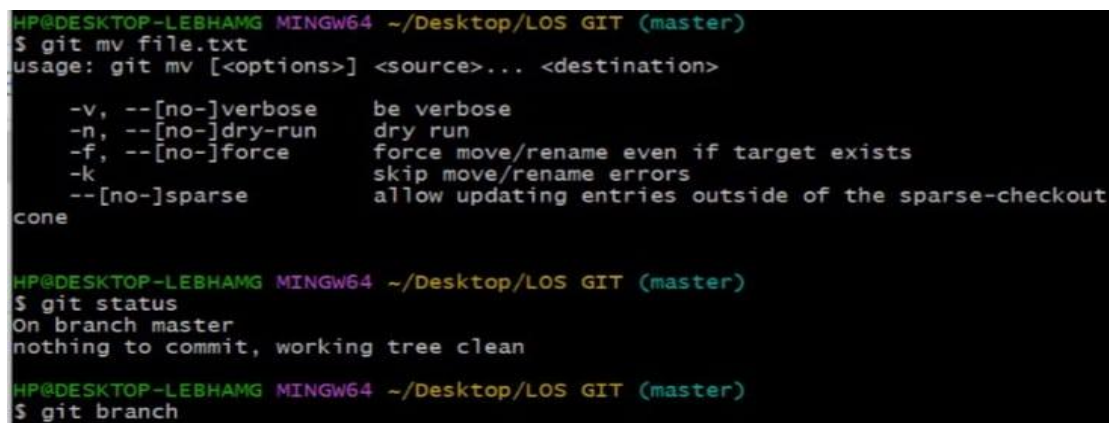
5. Remove a file or folder:

If you want to remove a file or folder from the repository:

git rm <filename> # Removes a file

git rm -r <foldername> # Removes a folder

git commit -m "Removed file/folder"



```
HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT (master)
$ git mv file.txt
usage: git mv [<options>] <source>... <destination>

    -v, --[no-]verbose      be verbose
    -n, --[no-]dry-run      dry run
    -f, --[no-]force        force move/rename even if target exists
    -k, --[no-]sparse       skip move/rename errors
                           allow updating entries outside of the sparse-checkout
                           cone

HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT (master)
$ git status
On branch master
nothing to commit, working tree clean

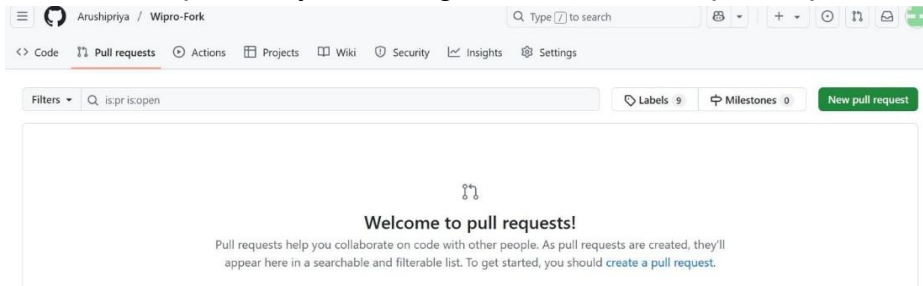
HP@DESKTOP-LEBHAMG MINGW64 ~/Desktop/LOS GIT (master)
$ git branch
```

6. Create a pull request:

After pushing your branch to GitHub, you can open a pull request to merge your changes into the main branch. This is done on GitHub's website, not through the command line:

- Go to the repository on GitHub.
- Select the branch you want to merge and click **New Pull Request**.

- Add a description of your changes and create the pull request.

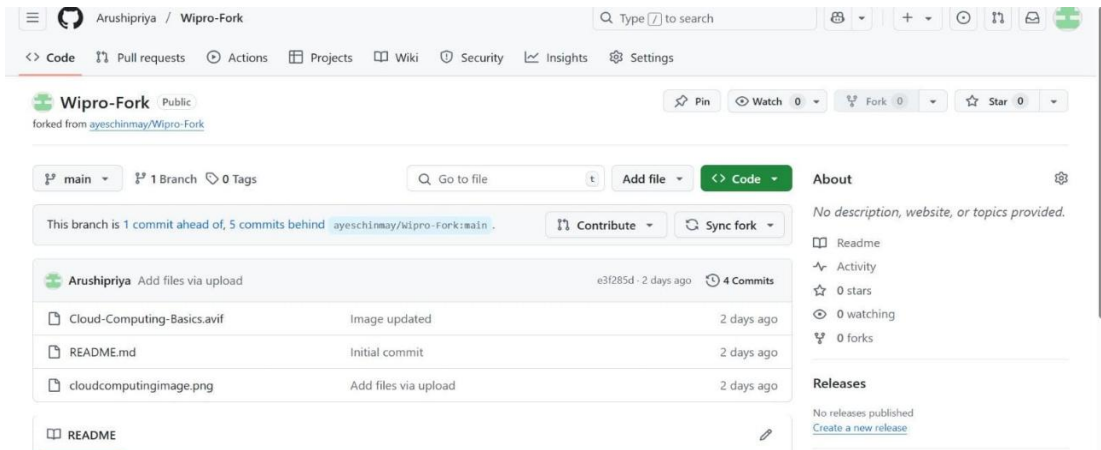


Common GitHub Workflows:

1. Forking a repository:

If you want to contribute to an existing GitHub repository, fork the repository:

- On the GitHub page, click **Fork**.
- This creates a copy of the repository in your account



2. Cloning a fork:

Clone your fork to work on it locally:

```
git clone https://github.com/yourusername/forked-repository.git
```

3. Syncing with the original repository:

To keep your fork up-to-date with the original repository:

```
git remote add upstream https://github.com/original-owner/repository.git # Add the original repository as a remote
```

```
git fetch upstream # Fetch changes from the original repository
```

git merge upstream/main # Merge changes into your local main branch

4. Creating a pull request from a fork:

After making changes to your fork, push them to your GitHub fork and create a pull request to the original repository.

NAME : ARUSHI PRIYA

REGD NO : 2141016384