GIT & GIT HUB

GIT: - GIT is a distributed version controlling system for tracking changes in computer files and coordinating work on those files among multiple peoples. It is created by Linux Torvalds and the OS are POSIX, Linux, Windows, macOS. It's an open-source platform.

Working of git: -

**THREE LAYER ARCHITECTURE**

1. Working directory

* This is where you actively work on your files
* Contains the actual files you can see and edit
* Any changes made here are considered "untracked" until you stage them
* Example: When you create or modify files in your project folder

2**.** Staging area

* Acts as a middle ground between working directory and repository
* Files are prepared here before committing
* Also called the "index" or "cache"
* Used to build up your next commit
* Example: When you use 'git add', files move here from working directory

3. Git repository

* Contains all your committed changes
* Stores the complete history of your project
* Maintains all versions and metadata
* Located in the .git folder of your project
* Example: When you use 'git commit', changes move here from staging area

## COMMANDS & THEIR FUNCTIONS:

1. Basic Commands:

* git init: Initialize a new Git repository
* git clone [url]: Clone/download a repository from remote
* git status: Check status of working directory
* git add [file]: Add file(s) to staging area
* git commit -m "[message]": Commit staged changes with a message
* git push: Upload local repository to remote
* git pull: Download changes from remote to local
* git diff file name: Shows the changed which were made

1. Branch Operations:

* git branch: List all branches
* git branch [name]: Create new branch
* git checkout [branch]: Switch to specified branch
* git merge [branch]: Merge specified branch into current branch
* git branch -d [branch]: Delete a branch

1. Remote Operations:

* git remote add [name] [url]: Add remote repository
* git remote -v: List remote repositories
* git fetch: Download changes from remote without merging
* git push origin [branch]: Push branch to remote
* git pull origin [branch]: Pull branch from remote

1. History & Differences:

* git log: View commit history
* git diff: Show changes between working directory and staging
* git diff --staged: Show changes between staging and last commit
* git show [commit]: Show specific commit details

1. Undoing Changes:

* git reset [file]: Unstage file
* git checkout -- [file]: Discard changes in working directory
* git revert [commit]: Create new commit that undoes specified commit
* git reset --hard [commit]: Reset to specific commit, discarding all changes

1. Stashing:

* git stash: Temporarily store modified files
* git stash pop: Restore most recently stashed files
* git stash list: List all stashed changes

1. Configuration:

* git config --global user.name "[name]": Set username
* git config --global user.email "[email]": Set email
* git config --list: List configuration settings

Checking the status of the file

$ git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: day2.txt

modified: index1.txt

Adding the file using command of git add .

Administrator@a0537ae3a1875ee MINGW64 /d/ATLAS3/day2 (master)

$ git add .

Administrator@a0537ae3a1875ee MINGW64 /d/ATLAS3/day2 (master)

$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: day2.txt

modified: index1.txt

By using the command of add. all the files gets staged to staging area to unstage them we can use the command of (use "git restore --staged <file>..." to unstage)

$git log is used to see the changes made by the user

Administrator@a0537ae3a1875ee MINGW64 /d/ATLAS3/day2 (master)

$ git log

commit 50730f41549aa97637c44c53ca524407bfe9649b (HEAD -> master)

Author: Moinuddin <umoin579@gmail.com>

Date: Tue May 20 15:02:53 2025 +0530

This is the learning of day 2

Administrator@a0537ae3a1875ee MINGW64 /d/ATLAS3/day2 (master)

$ git commit -m "THis is a version 1 of day 1"

[master 8c4565c] THis is a version 1 of day 1

2 files changed, 2 insertions(+)

To connect the local repo to git repo use the command as ($ git remote add origin <https://github.com/Moin145/day-2.git)>

Administrator@a0537ae3a1875ee MINGW64 /d/ATLAS3/day2 (master)

$ git push -u origin master

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 4 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (7/7), 556 bytes | 139.00 KiB/s, done.

Total 7 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

To <https://github.com/Moin145/day-2.git>

\* [new branch] master -> master

branch 'master' set up to track 'origin/master'.

to know the email of the user you can give a command as ($ git config user.email) same for user name

Then add a commit

Administrator@a0537ae3a1875ee MINGW64 /d/ATLAS3/day2 (master)

$ git commit -m "This is the version 2 of day 2"

[master 37ddee4] This is the version 2 of day 2

2 files changed, 2 insertions(+), 2 deletions(-)

then push it to the github repo

Administrator@a0537ae3a1875ee MINGW64 /d/ATLAS3/day2 (master)

$ git push -u origin master

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 4 threads

Compressing objects: 100% (2/2), done.

Writing objects: 100% (4/4), 348 bytes | 174.00 KiB/s, done.

Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

To <https://github.com/Moin145/day-2.git>

8c4565c..37ddee4 master -> master

branch 'master' set up to track 'origin/master'.

Check the changes made using command

Administrator@a0537ae3a1875ee MINGW64 /d/ATLAS3/day2 (master)

$ git log

commit 37ddee49872cc7cad3ffe08b6903f4f812322641 (HEAD -> master, origin/master)

Author: Moinuddin <umoin579@gmail.com>

Date: Tue May 20 15:41:35 2025 +0530

This is the version 2 of day 2

commit 8c4565ce6ea9c0bcc2cd7c2d513f7451bbb233ec

Author: Moinuddin <umoin579@gmail.com>

Date: Tue May 20 15:20:24 2025 +0530

THis is a version 1 of day 1

commit 50730f41549aa97637c44c53ca524407bfe9649b

Author: Moinuddin <umoin579@gmail.com>

Date: Tue May 20 15:02:53 2025 +0530

This is the learning of day 2

Lets say If I changed some content from my local file and want to know the difference between the unchanged file and newly changed file then we can use the command as

$ git diff "filename"

Administrator@a0537ae3a1875ee MINGW64 /d/ATLAS3/day2 (master)

$ git diff index1.txt

diff --git a/index1.txt b/index1.txt

index f37d676..352b990 100644

--- a/index1.txt

+++ b/index1.txt

@@ -1 +1 @@

-This is a version 2 of day2

\ No newline at end of file

+This is a version 3 of day2

\ No newline at end of file

to reset the changes made

Administrator@a0537ae3a1875ee MINGW64 /d/ATLAS3/day2 (master)

$ git reset index1.txt

Unstaged changes after reset:

M index1.txt

Administrator@a0537ae3a1875ee MINGW64 /d/ATLAS3/day2 (master)

$ 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: index1.txt

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

Administrator@a0537ae3a1875ee MINGW64 /d/ATLAS3/day2 (master)

$ git checkout index1.txt

Updated 1 path from the index

Administrator@a0537ae3a1875ee MINGW64 /d/ATLAS3/day2 (master)

$ git checkout .

Updated 2 paths from the index

To add a new file we can use touch command

Administrator@a0537ae3a1875ee MINGW64 /d/ATLAS3/day2 (master)

$ touch .gitignore

To create a new branch

Administrator@a0537ae3a1875ee MINGW64 /d/branching1 (master)

$ git branch login-system

to view all the branches

Administrator@a0537ae3a1875ee MINGW64 /d/branching1 (master)

$ git branch

login-system

\* master

to switch from master to new branch

Administrator@a0537ae3a1875ee MINGW64 /d/branching1 (master)

$ git checkout login-system

Switched to branch 'login-system'

Administrator@a0537ae3a1875ee MINGW64 /d/branching1 (login-system)

$ git branch

\* login-system

master

Administrator@a0537ae3a1875ee MINGW64 /d/branching1 (login-system)

$ git status

On branch login-system

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: branching1.txt

modified: master1.txt

Untracked files:

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

3.txt.txt

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

Administrator@a0537ae3a1875ee MINGW64 /d/branching1 (login-system)

$ git add .

Administrator@a0537ae3a1875ee MINGW64 /d/branching1 (login-system)

$ git status

On branch login-system

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: 3.txt.txt

modified: branching1.txt

modified: master1.txt

Administrator@a0537ae3a1875ee MINGW64 /d/branching1 (login-system)

$ git commit -m "Added login-system"

[login-system abfa8f5] Added login-system

3 files changed, 3 insertions(+), 2 deletions(-)

create mode 100644 3.txt.txt

Administrator@a0537ae3a1875ee MINGW64 /d/branching1 (login-system)

$ git push origin login-system

Enumerating objects: 8, done.

Counting objects: 100% (8/8), done.

Delta compression using up to 4 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (5/5), 485 bytes | 242.00 KiB/s, done.

Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

remote:

remote: Create a pull request for 'login-system' on GitHub by visiting:

remote: <https://github.com/Moin145/branching1/pull/new/login-system>

remote:

To <https://github.com/Moin145/branching1.git>

\* [new branch] login-system -> login-system

Administrator@a0537ae3a1875ee MINGW64 /d/branching1 (login-system)

$ git branch

\* login-system

master

Administrator@a0537ae3a1875ee MINGW64 /d/branching1 (login-system)

$ git checkout master

Switched to branch 'master'

Your branch is up to date with 'origin/master'.

Administrator@a0537ae3a1875ee MINGW64 /d/branching1 (master)

$ git branch

login-system

\* master

to merge the branch to master branch we have to checkout to master branch first

Administrator@a0537ae3a1875ee MINGW64 /d/branching1 (login-system)

$ git checkout master

Switched to branch 'master'

Your branch is up to date with 'origin/master'.

Administrator@a0537ae3a1875ee MINGW64 /d/branching1 (master)

$ git merge login-system

Updating 9fc5ae8..abfa8f5

Fast-forward

3.txt.txt | 1 +

branching1.txt | 2 +-

master1.txt | 2 +-

3 files changed, 3 insertions(+), 2 deletions(-)

create mode 100644 3.txt.txt

Administrator@a0537ae3a1875ee MINGW64 /d/branching1 (master)

$ git push -u origin master

Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

To <https://github.com/Moin145/branching1.git>

9fc5ae8..abfa8f5 master -> master

branch 'master' set up to track 'origin/master'.

to delete the branch

Administrator@a0537ae3a1875ee MINGW64 /d/branching1 (master)

$ git branch -d login-system

Deleted branch login-system (was abfa8f5).

Administrator@a0537ae3a1875ee MINGW64 /d/branching1 (master)

$ git branch

\* master

to delete the branch from the github repo

Administrator@a0537ae3a1875ee MINGW64 /d/branching1 (master)

$ git push origin --delete login-system

To <https://github.com/Moin145/branching1.git>

- [deleted] login-system

Cloning in GIT

Administrator@a0537ae3a1875ee MINGW64 /d/ATLAS3/day2/cloning (master)

$ git clone git@github.com:Moin145/moinportfolio.github.ai.git

Cloning into 'moinportfolio.github.ai'...

The authenticity of host 'github.com (20.207.73.82)' can't be established.

ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.

This key is not known by any other names.

Are you sure you want to continue connecting (yes/no/[fingerprint])? y

Please type 'yes', 'no' or the fingerprint: yes

Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.

remote: Enumerating objects: 19, done.

remote: Counting objects: 100% (19/19), done.

remote: Compressing objects: 100% (17/17), done.

remote: Total 19 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)

Receiving objects: 100% (19/19), 915.92 KiB | 1002.00 KiB/s, done.

Resolving deltas: 100% (4/4), done.