**EXPERIMENT NUMBER 2**

**Aim:** To understand Version Control System / Source Code Management, install git and create a GitHub account

**THEORY:**

Git is a free and open-source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. Git is easy to learn and has a tiny footprint with lightning-fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

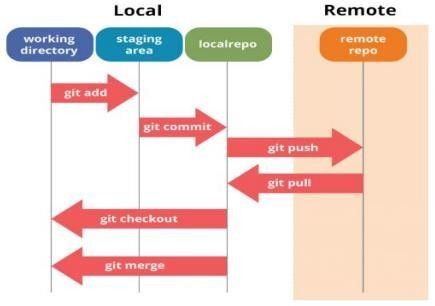
Some of the basic operations in Git are:

1. Initialize
2. Add
3. Commit
4. Pull
5. Push

Some advanced Git operations are:

1. Branching
2. Merging
3. Rebasing

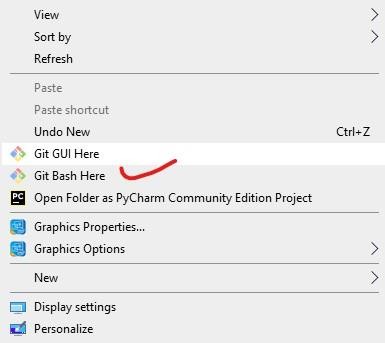
The following diagram depict the all supported operations in GIT



Installation of GIT

1. In windows, download GIT from https://git-scm.com/ and perform the straightforward installation.
2. In Ubuntu, install GIT using $sudo apt install git, Confirm the version after installation

$git --version

Once installation is done, open the terminal in Ubuntu and perform the following steps or in windows Right click and select Git bash here.

**EXECUTION:**

To perform version control, let us create a directory dvcs (Distributed version control system) and change directory to dvcs.

trish@Trishs-MacBook-Air ~ % mkdir gitty

trish@Trishs-MacBook-Air ~ % cd gitty

trish@Trishs-MacBook-Air gitty % git config --global user.name "TrishankV"

trish@Trishs-MacBook-Air gitty % git config --global user.email tvashikar@gmail.com

trish@Trishs-MacBook-Air gitty % git config --global --list

user.name=TrishankV

[user.email=tvashikar@gmail.com](mailto:user.email=tvashikar@gmail.com)

Let us create a repository for version control named ”git-demo-project”

$ mkdir git-demo-project

$ cd git-demo-project/

Now, initialize the repository using following command

$ git init

trish@Trishs-MacBook-Air gitty % cat ~/.gitconfig

[user]

name = TrishankV

email = [tvashikar@gmail.com](mailto:tvashikar@gmail.com)

trish@Trishs-MacBook-Air gitty % mkdir gitties

trish@Trishs-MacBook-Air gitty % cd gitties

trish@Trishs-MacBook-Air gitties % git init

Initialized empty Git repository in /Users/trish/gitty/gitties/.git/

If you have existing repository, then simply delete .git file and reinitialize it. $ rm -rf .git/

trish@Trishs-MacBook-Air gitties % rm -rf .git/

trish@Trishs-MacBook-Air gitties % ls -al

total 0

drwxr-xr-x 2 trish staff 64 Jan 26 13:51 .

drwxr-xr-x 3 trish staff 96 Jan 26 13:50 ..

trish@Trishs-MacBook-Air gitties % git init

Initialized empty Git repository in /Users/trish/gitty/gitties/.git/

trish@Trishs-MacBook-Air gitties % git add.

git: 'add.' is not a git command. See 'git --help'.

The most similar command is

add

trish@Trishs-MacBook-Air gitties % git add .

trish@Trishs-MacBook-Air gitties % git init

Reinitialized existing Git repository in /Users/trish/gitty/gitties/.git/

trish@Trishs-MacBook-Air gitties % git status

On branch main

No commits yet

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

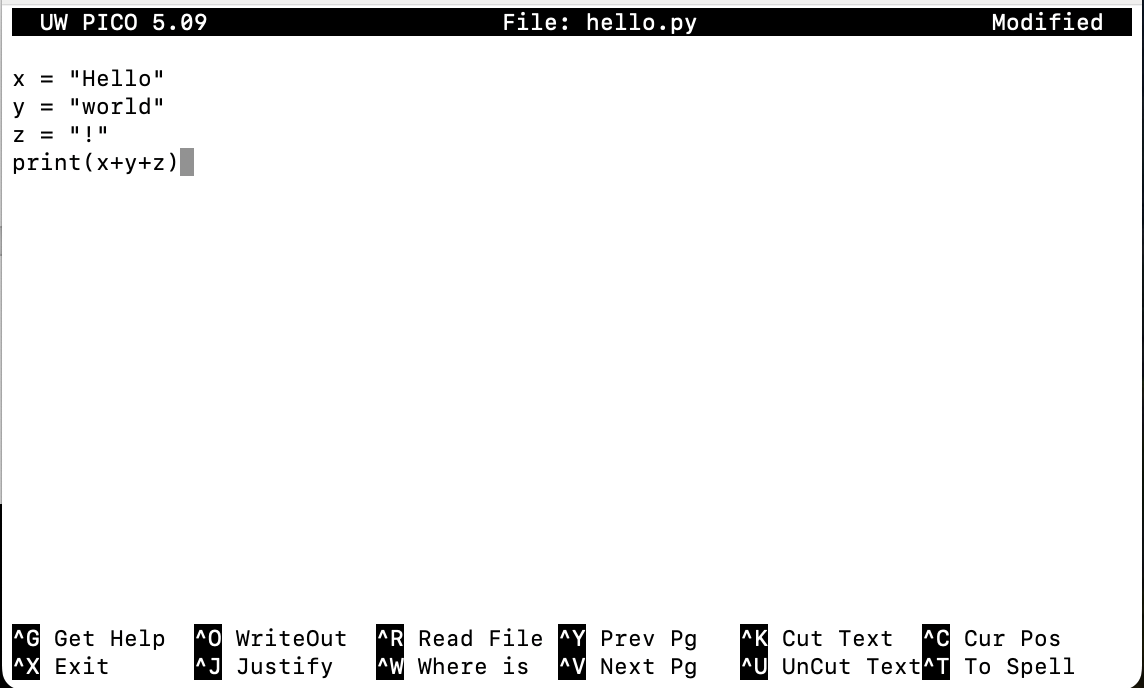
trish@Trishs-MacBook-Air gitties % git commit -m "First Comment"

On branch main

Initial commit

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

trish@Trishs-MacBook-Air gitties % nano hello.py



trish@Trishs-MacBook-Air gitties % git add .

trish@Trishs-MacBook-Air gitties % git commit -am "Express Commit"

[main (root-commit) bb33dfa] Express Commit

1 file changed, 4 insertions(+)

create mode 100644 a.py

trish@Trishs-MacBook-Air gitties % git status

On branch main

nothing to commit, working tree clean

trish@Trishs-MacBook-Air gitties % touch ts

trish@Trishs-MacBook-Air gitties % git status

On branch main

Untracked files:

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

ts

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

trish@Trishs-MacBook-Air gitties % git add ts

trish@Trishs-MacBook-Air gitties % git add a.py

trish@Trishs-MacBook-Air gitties % git status

On branch main

Changes to be committed:

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

new file: ts

trish@Trishs-MacBook-Air gitties % git commit -am "Express Commit"

[main f0e3d11] Express Commit

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 ts

trish@Trishs-MacBook-Air gitties % git log

commit f0e3d1188ce7ac3d78254cd1b2706d9be74d816e (**HEAD -> main**)

Author: TrishankV <tvashikar@gmail.com>

Date: Fri Jan 26 13:56:08 2024 +0530

Express Commit

commit bb33dfaa63864f3959219a07b44350c85f6ae913

Author: TrishankV <tvashikar@gmail.com>

Date: Fri Jan 26 13:54:22 2024 +0530

Express Commit

trish@Trishs-MacBook-Air gitties % git log --online

fatal: unrecognized argument: --online

trish@Trishs-MacBook-Air gitties % git log --oneline

f0e3d11 (**HEAD -> main**) Express Commit

bb33dfa Express Commit

trish@Trishs-MacBook-Air gitties % git clone https://github.com/MSid01/TriviaQuiz.git

Cloning into 'TriviaQuiz'...

remote: Enumerating objects: 12, done.

remote: Counting objects: 100% (12/12), done.

remote: Compressing objects: 100% (11/11), done.

remote: Total 12 (delta 1), reused 8 (delta 0), pack-reused 0

Receiving objects: 100% (12/12), 80.51 KiB | 8.95 MiB/s, done.

Resolving deltas: 100% (1/1), done.

trish@Trishs-MacBook-Air gitties % ls

TriviaQuiz a.py ts

trish@Trishs-MacBook-Air gitties % git remote add origin https://github.com/MSid01/TriviaQuiz.git

trish@Trishs-MacBook-Air gitties % git remote show origin

\* remote origin

Fetch URL: https://github.com/MSid01/TriviaQuiz.git

Push URL: https://github.com/MSid01/TriviaQuiz.git

HEAD branch: main

Remote branch:

main new (next fetch will store in remotes/origin)

Local ref configured for 'git push':

main pushes to main (local out of date)

trish@Trishs-MacBook-Air gitties % git remote rm origin

trish@Trishs-MacBook-Air gitties % git push -u origin master

error: src refspec master does not match any

error: failed to push some refs to 'origin'

trish@Trishs-MacBook-Air gitties % git push -u https://github.com/TrishankV/Sem-6.git

fatal: The current branch main has no upstream branch.

To push the current branch and set the remote as upstream, use

git push --set-upstream https://github.com/TrishankV/Sem-6.git main

To have this happen automatically for branches without a tracking

upstream, see 'push.autoSetupRemote' in 'git help config'.

trish@Trishs-MacBook-Air gitties % git push --set-upstream https://github.com/TrishankV/Sem-6.git main

Create Personal Access Token on GitHub

From your GitHub account, go to **Settings** → **Developer Settings** → **Personal Access Token** → **Tokens (classic)** → **Generate New Token** (Give your password) → **Fillup the form** → click **Generate token** → **Copy the generated Token**, it will be something like ghp\_sFhFsSHhTzMDreGRLjmks4Tzuzgthdvfsrta

Use this PAT code instead of Password as the Password authentication has been restricted since 2021

To https://github.com/TrishankV/Sem-6.git

! [rejected] main -> main (fetch first)

error: failed to push some refs to 'https://github.com/TrishankV/Sem-6.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

trish@Trishs-MacBook-Air gitties % git pull origin main

remote: Enumerating objects: 3, done.

remote: Counting objects: 100% (3/3), done.

remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (3/3), 891 bytes | 178.00 KiB/s, done.

From https://github.com/TrishankV/Sem-6

\* branch main -> FETCH\_HEAD

\* [new branch] main -> origin/main

hint: You have divergent branches and need to specify how to reconcile them.

hint: You can do so by running one of the following commands sometime before

hint: your next pull:

hint:

hint: git config pull.rebase false # merge

hint: git config pull.rebase true # rebase

hint: git config pull.ff only # fast-forward only

hint:

hint: You can replace "git config" with "git config --global" to set a default

hint: preference for all repositories. You can also pass --rebase, --no-rebase,

hint: or --ff-only on the command line to override the configured default per

hint: invocation.

fatal: Need to specify how to reconcile divergent branches.

trish@Trishs-MacBook-Air gitties % git pull origin main --rebase

From https://github.com/TrishankV/Sem-6

\* branch main -> FETCH\_HEAD

Successfully rebased and updated refs/heads/main.

trish@Trishs-MacBook-Air gitties % git pull origin main

From https://github.com/TrishankV/Sem-6

\* branch main -> FETCH\_HEAD

Already up to date.

trish@Trishs-MacBook-Air gitties % git push --set-upstream https://github.com/TrishankV/Sem-6.git main

Enumerating objects: 7, done.

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

Delta compression using up to 8 threads

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

Writing objects: 100% (6/6), 537 bytes | 537.00 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/TrishankV/Sem-6.git

804b34c..a6cb554 main -> main

branch 'main' set up to track 'https://github.com/TrishankV/Sem-6.git/main'.

trish@Trishs-MacBook-Air gitties % git pull

From https://github.com/TrishankV/Sem-6

\* branch main -> FETCH\_HEAD

Already up to date.

trish@Trishs-MacBook-Air gitties % git log --oneline

a6cb554 (**HEAD -> main**) Express Commit

5a5ddbd Express Commit

804b34c (**origin/main**) Initial commit

trish@Trishs-MacBook-Air gitties % git fetch

From https://github.com/TrishankV/Sem-6

\* branch main -> FETCH\_HEAD

trish@Trishs-MacBook-Air gitties % git merge

fatal: No remote-tracking branch for refs/heads/main from https://github.com/TrishankV/Sem-6.git

trish@Trishs-MacBook-Air gitties % cat a.py

x = "hello"

y = "world"

z = "!"

print(x+y+z)

