user@DESKTOP-C38B6GS MINGW64 ~/OneDrive/Desktop/python\_1

$ git init

Initialized empty Git repository in C:/Users/user/OneDrive/Desktop/python\_1/.git/

user@DESKTOP-C38B6GS MINGW64 ~/OneDrive/Desktop/python\_1 (master)

$ git add .

user@DESKTOP-C38B6GS MINGW64 ~/OneDrive/Desktop/python\_1 (master)

$ git status

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: oop.py

new file: py\_1.py

new file: py\_2.py

new file: py\_3.py

new file: py\_4.py

user@DESKTOP-C38B6GS MINGW64 ~/OneDrive/Desktop/python\_1 (master)

$ git commit -m "first\_commit\_data\_excel\_file"

[master (root-commit) 1898aad] first\_commit\_data\_excel\_file

5 files changed, 165 insertions(+)

create mode 100644 oop.py

create mode 100644 py\_1.py

create mode 100644 py\_2.py

create mode 100644 py\_3.py

create mode 100644 py\_4.py

user@DESKTOP-C38B6GS MINGW64 ~/OneDrive/Desktop/python\_1 (master)

$ git branch -M main

user@DESKTOP-C38B6GS MINGW64 ~/OneDrive/Desktop/python\_1 (main)

$ git remote add origin https://github.com/aminul-12/python\_1.git

user@DESKTOP-C38B6GS MINGW64 ~/OneDrive/Desktop/python\_1 (main)

$ git push -u origin main

Enumerating objects: 7, done.

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

Delta compression using up to 8 threads

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

Writing objects: 100% (7/7), 1.66 KiB | 424.00 KiB/s, done.

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

To https://github.com/aminul-12/python\_1.git

\* [new branch] main -> main

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

user@DESKTOP-C38B6GS MINGW64 ~/OneDrive/Desktop/python\_1 (main)

$ git add python.txt

user@DESKTOP-C38B6GS MINGW64 ~/OneDrive/Desktop/python\_1 (main)

$ 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: python.txt

user@DESKTOP-C38B6GS MINGW64 ~/OneDrive/Desktop/python\_1 (main)

$ git commit -m "python.txt\_file"

See 'git help <command>' to read about a specific subcommand

git push -u origin main

eta type korthe hbe

Main Porcelain Commands

add Add file contents to the index

am Apply a series of patches from a mailbox

archive Create an archive of files from a named tree

backfill Download missing objects in a partial clone

bisect Use binary search to find the commit that introduced a bug

branch List, create, or delete branches

bundle Move objects and refs by archive

checkout Switch branches or restore working tree files

cherry-pick Apply the changes introduced by some existing commits

citool Graphical alternative to git-commit

clean Remove untracked files from the working tree

clone Clone a repository into a new directory

commit Record changes to the repository

describe Give an object a human readable name based on an available ref

diff Show changes between commits, commit and working tree, etc

fetch Download objects and refs from another repository

format-patch Prepare patches for e-mail submission

:...skipping...

See 'git help <command>' to read about a specific subcommand

Main Porcelain Commands

add Add file contents to the index

am Apply a series of patches from a mailbox

archive Create an archive of files from a named tree

backfill Download missing objects in a partial clone

bisect Use binary search to find the commit that introduced a bug

branch List, create, or delete branches

bundle Move objects and refs by archive

checkout Switch branches or restore working tree files

cherry-pick Apply the changes introduced by some existing commits

citool Graphical alternative to git-commit

clean Remove untracked files from the working tree

clone Clone a repository into a new directory

commit Record changes to the repository

describe Give an object a human readable name based on an available ref

diff Show changes between commits, commit and working tree, etc

fetch Download objects and refs from another repository

format-patch Prepare patches for e-mail submission

gc Cleanup unnecessary files and optimize the local repository

gitk The Git repository browser

grep Print lines matching a pattern

gui A portable graphical interface to Git

init Create an empty Git repository or reinitialize an existing one

log Show commit logs

maintenance Run tasks to optimize Git repository data

merge Join two or more development histories together

mv Move or rename a file, a directory, or a symlink

notes Add or inspect object notes

pull Fetch from and integrate with another repository or a local branch

push Update remote refs along with associated objects

range-diff Compare two commit ranges (e.g. two versions of a branch)

rebase Reapply commits on top of another base tip

reset Reset current HEAD to the specified state

restore Restore working tree files

revert Revert some existing commits

rm Remove files from the working tree and from the index

scalar A tool for managing large Git repositories

shortlog Summarize 'git log' output

show Show various types of objects

sparse-checkout Reduce your working tree to a subset of tracked files

stash Stash the changes in a dirty working directory away

status Show the working tree status

submodule Initialize, update or inspect submodules

survey EXPERIMENTAL: Measure various repository dimensions of scale

switch Switch branches

tag Create, list, delete or verify a tag object signed with GPG

worktree Manage multiple working trees

Ancillary Commands / Manipulators

config Get and set repository or global options

fast-export Git data exporter

fast-import Backend for fast Git data importers

filter-branch Rewrite branches

mergetool Run merge conflict resolution tools to resolve merge conflicts

pack-refs Pack heads and tags for efficient repository access

:

See 'git help <command>' to read about a specific subcommand

Main Porcelain Commands

add Add file contents to the index

am Apply a series of patches from a mailbox

archive Create an archive of files from a named tree

backfill Download missing objects in a partial clone

bisect Use binary search to find the commit that introduced a bug

branch List, create, or delete branches

bundle Move objects and refs by archive

checkout Switch branches or restore working tree files

cherry-pick Apply the changes introduced by some existing commits

citool Graphical alternative to git-commit

clean Remove untracked files from the working tree

clone Clone a repository into a new directory

commit Record changes to the repository

describe Give an object a human readable name based on an available ref

diff Show changes between commits, commit and working tree, etc

fetch Download objects and refs from another repository

format-patch Prepare patches for e-mail submission

gc Cleanup unnecessary files and optimize the local repository

gitk The Git repository browser

grep Print lines matching a pattern

gui A portable graphical interface to Git

init Create an empty Git repository or reinitialize an existing one

log Show commit logs

maintenance Run tasks to optimize Git repository data

merge Join two or more development histories together

mv Move or rename a file, a directory, or a symlink

notes Add or inspect object notes

pull Fetch from and integrate with another repository or a local branch

push Update remote refs along with associated objects

range-diff Compare two commit ranges (e.g. two versions of a branch)

rebase Reapply commits on top of another base tip

reset Reset current HEAD to the specified state

restore Restore working tree files

revert Revert some existing commits

rm Remove files from the working tree and from the index

scalar A tool for managing large Git repositories

shortlog Summarize 'git log' output

show Show various types of objects

sparse-checkout Reduce your working tree to a subset of tracked files

stash Stash the changes in a dirty working directory away

status Show the working tree status

submodule Initialize, update or inspect submodules

survey EXPERIMENTAL: Measure various repository dimensions of scale

switch Switch branches

tag Create, list, delete or verify a tag object signed with GPG

worktree Manage multiple working trees

Ancillary Commands / Manipulators

config Get and set repository or global options

fast-export Git data exporter

fast-import Backend for fast Git data importers

filter-branch Rewrite branches

mergetool Run merge conflict resolution tools to resolve merge conflicts

pack-refs Pack heads and tags for efficient repository access

: