1 pwd

2 cd project

3 touch a

touch b

ls

4 git init // to create local repository

5 git status

6 git add a b //adding file to staging area

7 git status

git config --global user.email "[pranali.wagh@gmail.com](mailto:pranali.wagh@gmail.com)" //before commit

17 git config --global user.name "pranali"

18 git commit -m "first commit" // committed files to LR

19 git status

20 git log // how many commit is done earlier

21 Make changes in file git add a b (git commit -a -m “files changed”)

22 /\*git commit -m "second version"

23 git status

24 git log

26 history

27 git clone

28 git clone "https://github.com/pranalishelar12/test.git"

29 cd test

30 ls

31 cat f1

33 cat >> f1

34 ls

35 cat f1

36 git add .

37 git commit -"m "updation in f1"

git commit -m "updation in f1"

38 git commit -m "udation in f1"

39 git log --oneline

40 git log

41 git push origin main

42 git push origin main

GIT output

Administrator@204-029 MINGW64 /c/project

$ git init

Initialized empty Git repository in C:/project/.git/

Administrator@204-029 MINGW64 /c/project (master)

$ git status

On branch master

No commits yet

Untracked files:

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

a.txt.docx

b.txt.docx

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

Administrator@204-029 MINGW64 /c/project (master)

$ git add .\*

Administrator@204-029 MINGW64 /c/project (master)

$ git status

On branch master

No commits yet

Untracked files:

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

a.txt.docx

b.txt.docx

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

Administrator@204-029 MINGW64 /c/project (master)

$ git add a.txt b.txt

fatal: pathspec 'a.txt' did not match any files

Administrator@204-029 MINGW64 /c/project (master)

$ git add -a

error: unknown switch `a'

usage: git add [<options>] [--] <pathspec>...

-n, --dry-run dry run

-v, --verbose be verbose

-i, --interactive interactive picking

-p, --patch select hunks interactively

-e, --edit edit current diff and apply

-f, --force allow adding otherwise ignored files

-u, --update update tracked files

--renormalize renormalize EOL of tracked files (implies -u)

-N, --intent-to-add record only the fact that the path will be added later

-A, --all add changes from all tracked and untracked files

--ignore-removal ignore paths removed in the working tree (same as --no-all)

--refresh don't add, only refresh the index

--ignore-errors just skip files which cannot be added because of errors

--ignore-missing check if - even missing - files are ignored in dry run

--sparse allow updating entries outside of the sparse-checkout cone

--chmod (+|-)x override the executable bit of the listed files

--pathspec-from-file <file>

read pathspec from file

--pathspec-file-nul with --pathspec-from-file, pathspec elements are separated with NUL character

Administrator@204-029 MINGW64 /c/project (master)

$ ls

a.txt.docx b.txt.docx

Administrator@204-029 MINGW64 /c/project (master)

$ ls

a.txt b.txt

Administrator@204-029 MINGW64 /c/project (master)

$ git a.txt b.txt

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

Administrator@204-029 MINGW64 /c/project (master)

$ git add a.txt b.txt

Administrator@204-029 MINGW64 /c/project (master)

$ git satus

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

The most similar command is

status

Administrator@204-029 MINGW64 /c/project (master)

$ git status

On branch master

No commits yet

Changes to be committed:

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

new file: a.txt

new file: b.txt

Administrator@204-029 MINGW64 /c/project (master)

$ git config --global user.email "pranali.wagh@gmail.com"

Administrator@204-029 MINGW64 /c/project (master)

$ git config --global user.name "pranali"

Administrator@204-029 MINGW64 /c/project (master)

$ git commit -m "first commit"

[master (root-commit) 57cc53b] first commit

2 files changed, 0 insertions(+), 0 deletions(-)

create mode 100644 a.txt

create mode 100644 b.txt

Administrator@204-029 MINGW64 /c/project (master)

$ git log

commit 57cc53b94ac94e86e919bbaaad21ddd9168e8658 (HEAD -> master)

Author: pranali <pranali.wagh@gmail.com>

Date: Fri Jul 14 12:23:42 2023 +0530

first commit

Administrator@204-029 MINGW64 /c/project (master)

$ 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: a.txt

modified: b.txt

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

Administrator@204-029 MINGW64 /c/project (master)

$ git add a.txt b.txt

Administrator@204-029 MINGW64 /c/project (master)

$ git commit -m "second version"

[master 82f7fd1] second version

2 files changed, 4 insertions(+)

Administrator@204-029 MINGW64 /c/project (master)

$ git log

commit 82f7fd110828c699b1f297ae05b490e4c7ea312e (HEAD -> master)

Author: pranali <pranali.wagh@gmail.com>

Date: Fri Jul 14 12:28:15 2023 +0530

second version

commit 57cc53b94ac94e86e919bbaaad21ddd9168e8658

Author: pranali <pranali.wagh@gmail.com>

Date: Fri Jul 14 12:23:42 2023 +0530

first commit

Administrator@204-029 MINGW64 /c/project (master)

$ git status

On branch master

nothing to commit, working tree clean

Administrator@204-029 MINGW64 /c/project (master)

$ git history

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

Administrator@204-029 MINGW64 /c/project (master)

$ history

1 pwd

2 cd project

3 clear

4 git init

5 git status

6 git add .\*

7 git status

8 git add a.txt b.txt

9 git add -a

10 ls

11 ls

12 git a.txt b.txt

13 git add a.txt b.txt

14 git satus

15 git status

16 git config --global user.email "pranali.wagh@gmail.com"

17 git config --global user.name "pranali"

18 git commit -m "first commit"

19 git log

20 git status

21 git add a.txt b.txt

22 git commit -m "second version"

23 git log

24 git status

25 git history

26 history

Administrator@204-029 MINGW64 /c/project (master)

$ ^C

Administrator@204-029 MINGW64 /c/project (master)

$ git clone

fatal: You must specify a repository to clone.

usage: git clone [<options>] [--] <repo> [<dir>]

-v, --verbose be more verbose

-q, --quiet be more quiet

--progress force progress reporting

--reject-shallow don't clone shallow repository

-n, --no-checkout don't create a checkout

--bare create a bare repository

--mirror create a mirror repository (implies bare)

-l, --local to clone from a local repository

--no-hardlinks don't use local hardlinks, always copy

-s, --shared setup as shared repository

--recurse-submodules[=<pathspec>]

initialize submodules in the clone

--recursive ... alias of --recurse-submodules

-j, --jobs <n> number of submodules cloned in parallel

--template <template-directory>

directory from which templates will be used

--reference <repo> reference repository

--reference-if-able <repo>

reference repository

--dissociate use --reference only while cloning

-o, --origin <name> use <name> instead of 'origin' to track upstream

-b, --branch <branch>

checkout <branch> instead of the remote's HEAD

-u, --upload-pack <path>

path to git-upload-pack on the remote

--depth <depth> create a shallow clone of that depth

--shallow-since <time>

create a shallow clone since a specific time

--shallow-exclude <revision>

deepen history of shallow clone, excluding rev

--single-branch clone only one branch, HEAD or --branch

--no-tags don't clone any tags, and make later fetches not to follow them

--shallow-submodules any cloned submodules will be shallow

--separate-git-dir <gitdir>

separate git dir from working tree

-c, --config <key=value>

set config inside the new repository

--server-option <server-specific>

option to transmit

-4, --ipv4 use IPv4 addresses only

-6, --ipv6 use IPv6 addresses only

--filter <args> object filtering

--also-filter-submodules

apply partial clone filters to submodules

--remote-submodules any cloned submodules will use their remote-tracking branch

--sparse initialize sparse-checkout file to include only files at root

--bundle-uri <uri> a URI for downloading bundles before fetching from origin remote

Administrator@204-029 MINGW64 /c/project (master)

$ git clone "https://github.com/pranalishelar12/test.git"

Cloning into 'test'...

remote: Enumerating objects: 3, done.

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

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

Receiving objects: 100% (3/3), done.

Administrator@204-029 MINGW64 /c/project (master)

$ cd test

Administrator@204-029 MINGW64 /c/project/test (main)

$ ls

f1

Administrator@204-029 MINGW64 /c/project/test (main)

$ cat f1

hi this is cloudy day!!!!!!!!

hi

Administrator@204-029 MINGW64 /c/project/test (main)

$ gedit f1

bash: gedit: command not found

Administrator@204-029 MINGW64 /c/project/test (main)

$ cat >> f1

hi sunny day tomroww

[1]+ Stopped cat >> f1

Administrator@204-029 MINGW64 /c/project/test (main)

$ ls

f1

Administrator@204-029 MINGW64 /c/project/test (main)

$ cat f1

hi this is cloudy day!!!!!!!!

hi

hi sunny day tomroww

Administrator@204-029 MINGW64 /c/project/test (main)

$ git add .

warning: in the working copy of 'f1', LF will be replaced by CRLF the next time Git touches it

Administrator@204-029 MINGW64 /c/project/test (main)

$ git commit -"m "updation in f1"

> git commit -m "updation in f1"

>

> ^C

Administrator@204-029 MINGW64 /c/project/test (main)

$ git commit -m "udation in f1"

[main f73cdbe] udation in f1

1 file changed, 1 insertion(+)

Administrator@204-029 MINGW64 /c/project/test (main)

$ git log --onefile

fatal: unrecognized argument: --onefile

Administrator@204-029 MINGW64 /c/project/test (main)

$ git log

commit f73cdbe11cd86a01ed4ce657df37f96014ffcc07 (HEAD -> main)

Author: pranali <pranali.wagh@gmail.com>

Date: Fri Jul 14 14:26:16 2023 +0530

udation in f1

commit cb7317755887ac1bf6e5eeb3c67b29637416deea (origin/main, origin/HEAD)

Author: pranalishelar12 <139542969+pranalishelar12@users.noreply.github.com>

Date: Fri Jul 14 14:17:21 2023 +0530

Create f1

Administrator@204-029 MINGW64 /c/project/test (main)

$ git push origin main

fatal: User cancelled dialog.

error: unable to read askpass response from 'C:/Program Files/Git/mingw64/bin/git-askpass.exe'

Username for 'https://github.com':

remote: No anonymous write access.

fatal: Authentication failed for 'https://github.com/pranalishelar12/test.git/'

Administrator@204-029 MINGW64 /c/project/test (main)

$ git push origin main

Enumerating objects: 5, done.

Counting objects: 100% (5/5), done.

Delta compression using up to 4 threads

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

Writing objects: 100% (3/3), 276 bytes | 276.00 KiB/s, done.

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

To https://github.com/pranalishelar12/test.git

cb73177..f73cdbe main -> main