**GIT User Manual**

pwd => Location

ls => all Lists in the folder

cd => change directory

# Go to GIT Folder

# User Name Add/ Change / Check

Add Name => $ git config --global user.name "My Name"

Change Name => $ git config --global user.name "My New Name"

Check Name => $ git config –-global user.name

# User Email Add/ Change / Check

Add Email => $ git config --global user.email My Email

Change Email => $ git config --global user.email My New Email

Check Email => $ git config –-global user.email

# Check All List ( Details of all )

List => $ git config –list

# Add Default Branch

Default Branch => $ git config --global init.default branch main

# Make GIT init

Making git init => $ git init

* Check a git hidden folder created into Folder

# git status

Check Status => $ git status

* git status will show branch name ( we add main )
* git status will show commits details
* git status will show add on not add file list

# git add file

Add file => $ git add fileName

* Check $ git status for check file add or not

# git remove file

Remove file => $ git rm –-cached FileName

* Check $ git status for add file remove or not

# Don’t add file or ignore any file

=> Creat text doc change name as .gitignore file

# Add all file at a time

Add all file => $ git add –all

Or => $ git add –A

Or => $ git add .

* This 3 ways to add all files
* Check $ git status all file are Added

# $ git diff when file some change then

File change check => $ git diff

* Git diff will show previous file and new file’s change

# git diff Status

diff A file to A change file => $ git status

* git status will show change between A and changed A
* Old file will red and changed file will be in green color
* Git status will show modified in RED color

# git diff restore

Restore Change => $ git restore FileName

* For single file by name and restore all file use . (dot )

# git diff All restore change file together once

Restore Change all => $ git restore .

# git diff add to Staged ( for commit )

Add diff Staged => $ git add FileName

* Adding to Staged then we can Commit it ..
* Git status will show modified in GREEN color

#git diff staged Restore

Restore staged => $ git restore –staged FileName

#git diff staged Restore all changed file once

Restore staged all => $ git restore –staged .

* For single file by file name and restore all file use . (dot )

# git Commit file

Commit file => $ git commit –a –m “ commit Details ”

# Delete A file by git

Delete file => $ git rm “FileName.EX”

# Restore Delete file

Restore Delete file => $ git restore “FileName.EX”

# Check Delete Status => $ git status

* If deleted file is GREEN it is on Staged
* If deleted file is RED it is on Working Mood

# Rename FileName

Rename => $ git mv “FileName.Ex” “New FileName.Ex”

# Commit Changes

Commit => $ git commit –m “Changed Details”

# Commit Check Details

Commit Log => $ git log

* Details of all Commit Log

# Commit Log Online => $git log --oneline

* All Log details by single line

# Change last Commit Details

Last Commit Change => $ git commit –m “ Change last Commit” --amend

# All Log check

Log All Details => $ git log –p

* Quit from log press q

# Log Help ( all Details log ) => $ git help log

# Change Any commit

Log Change => $ git reset

Multiple branch Use

# ADD new Branch => $ git branch BranchName

* Branch will Show on Comment box pasha@DESKTOP-FEUFOO2 MINGW64 ~/Desktop/Work/GIT/Git\_TEST (main)

# All Branch Check => $ git branch

Will show like this

\* FixTemp

main

# Switch Branch => $ git switch ‘BranchName’

* We show that in comment box as pasha@DESKTOP-FEUFOO2 MINGW64 ~/Desktop/Work/GIT/Git\_TEST (FixTemp)
* Each branch have individual activity
* A branch can keep 5 file and also B branch can keep 3 file
* Each branch can keep their won content save …

# Merge branch => $ git merge –m “Merging FixTemp to main ” FixTemp

* A is main and B is FixTemp
* A can Marge B and also B can Marge A

# Delete Branch => $ git branch –d BranchName

# Switch and Add New Branch Together => $ git switch –c NewBranchName

# All Branch work end need commit => $ git commit –a –m “Comment End of Work”

Git Hub Upload

# Tp PUSH GitHub => $ git remote add origin GitHub\_HTTPS\_Link

Then => $ git branch –M main

Then => $ git push –u origin main

* My $ git push –u origin main is not working then => $ git push -u origin main

Oala Done