**Basics of Git:**

Install the latest **Git version for windows** from ([Here](https://git-scm.com/download/win)). During the installation, accept the defaults and finish the setup.

Run PowerShell and try the following basic (Git-) commands.

PS C:\WINDOWS\system32> **git --version**

*git version 2.39.1.windows.1*

PS C:\WINDOWS\system32> **git config --list**

*diff.astextplain.textconv=astextplain*

*filter.lfs.clean=git-lfs clean -- %f*

*filter.lfs.smudge=git-lfs smudge -- %f*

*filter.lfs.process=git-lfs filter-process*

*filter.lfs.required=true*

*http.sslbackend=openssl*

*http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt*

*core.autocrlf=true*

*core.fscache=true*

*core.symlinks=false*

*pull.rebase=false*

*credential.helper=manager*

*credential.https://dev.azure.com.usehttppath=true init.defaultbranch=master*

PS C:\WINDOWS\system32> **git config --global user.name "Mulugeta M. Gessei"**

PS C:\WINDOWS\system32> **git config --global user.email "gsmuma@gmail.com"**

PS C:\WINDOWS\system32> **git config user.name**

*Mulugeta M. Gessei*

PS C:\WINDOWS\system32> **git config user.email**

*gsmuma@gmail.com*

PS C:\WINDOWS\system32> **git config --list**

*diff.astextplain.textconv=astextplain*

*filter.lfs.clean=git-lfs clean -- %f*

*filter.lfs.smudge=git-lfs smudge -- %f*

*filter.lfs.process=git-lfs filter-process*

*filter.lfs.required=true*

*http.sslbackend=openssl*

*http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt*

*core.autocrlf=true*

*core.fscache=true*

*core.symlinks=false*

*pull.rebase=false*

*credential.helper=manager*

*credential.https://dev.azure.com.usehttppath=true*

*init.defaultbranch=master*

*user.name=Mulugeta M. Gessei*

*user.email=gsmuma@gmail.com*

PS C:\WINDOWS\system32> **git -help**

*unknown option: -help*

*usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]*

*[--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]*

*[-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]*

*[--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]*

*[--super-prefix=<path>] [--config-env=<name>=<envvar>]*

*<command> [<args>]*

PS C:\WINDOWS\system32> **git help config**

*(Opens more* ***Git help*** *in a web browser)*

**Navigate to local directory and initializing Git**

PS C:\WINDOWS\system32> **cd D:\Git\_Pra**

PS D:\Git\_Pra> **git init**

*Initialized empty Git repository in D:/Git\_Pra/.git/*

PS D:\Git\_Pra> **cd .git**

PS D:\Git\_Pra\.git> **ls**

*Directory: D:\Git\_Pra\.git*

*Mode LastWriteTime Length Name*

*d----- 2/6/2023 8:08 PM hooks*

*d----- 2/6/2023 8:08 PM info*

*d----- 2/6/2023 8:08 PM objects*

*d----- 2/6/2023 8:08 PM refs*

*-a---- 2/6/2023 8:08 PM 130 config*

*-a---- 2/6/2023 8:08 PM 73 description*

*-a---- 2/6/2023 8:08 PM 23 HEAD*

PS D:\Git\_Pra\.git> **cd..**

**Creating/Adding “ReadMe” file/s inside the working directory**

PS D:\Git\_Pra> **echo "# Git\_Pra" >> Readme.md**

* So far the newly created file/s aren’t in the staging location yet. These need to be in a staging state before “pushed” to the remote repo. To add these files to the staging location, use:

PS D:\Git\_Pra> **git add .**

* Now that Git is authorized to control the versions of files and contents inside the working/local directory, it is ready to be committed to a remote repo. Use:

PS D:\Git\_Pra> **git commit -m "First commit"**

*[master (root-commit) c1f2875] First commit*

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

*create mode 100644 Readme.md*

**Creating a New Remote Repository, example in GitHub**: and sync the locally committed repo controlled by Git. To do so, login to the GitHub page and copy the “https” url of the repo. Paste the url after the following commands …

PS D:\Git\_Pra> **git remote add origin** https://github.com/mulugetagessei/Git\_Pra.git

PS D:\Git\_Pra> **git push -u origin master**

*info: please complete authentication in your browser...*

*Enumerating objects: 3, done.*

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

*Writing objects: 100% (3/3), 241 bytes | 120.00 KiB/s, done.*

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

*To https://github.com/mulugetagessei/Git\_Pra.git*

*\* [new branch] master -> master*

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

**NB**: When pushing files from a local to remote repos, authentication via username/pwd is a must!

Checking the status of the git using: This confirms what’s on the branch master, staging area,…

PS D:\Git\_Pra> **git status**

*On branch master*

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

*nothing to commit, working tree clean*

Tracked files in Git has three states: *Committed, Modified and Staged.* Right now, there is nothing modified in the branch mater since the last “commit”. Let’s *modify* the repo by adding a file and run the “add” command again to move the branch to the *Staging* location. For illustration purposes, copy this guide into the local repo.

PS D:\Git\_Pra> **git add .**

Add a file to the local repo and run the “status” again using:

PS D:\Git\_Pra> **git status**

*On branch master*

*Your branch is ahead of 'origin/master' by 1 commit.*

*(use "git push" to publish your local commits)*

*Changes to be committed:*

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

*modified: Basics of Git.docx*

When the status returns a GREEN line (which is still in a staging area), it means Git is ready to commit. Run the commit command to move the modified repo to the “committed” location.

PS D:\Git\_Pra> git commit -m "2nd commit"

*[master 6082ce8] 2nd commit*

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

Time to “push” the committed repo to remote (GitHub). Use:

PS D:\Git\_Pra> **git push -u origin master**

*Enumerating objects: 9, done.*

*Counting objects: 100% (9/9), done.*

*Delta compression using up to 16 threads*

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

*Writing objects: 100% (7/7), 15.93 KiB | 2.65 MiB/s, done.*

*Total 7 (delta 3), reused 0 (delta 0), pack-reused 0*

*remote: Resolving deltas: 100% (3/3), completed with 1 local object.*

*To https://github.com/mulugetagessei/Git\_Pra.git*

*88b634a..6082ce8 master -> master*

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

After some modification, if we want to “Restore” the changes made, we can use the “restore” command. This action restores back the modified file and status of the repo remains as before.

PS D:\Git\_Pra> **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: Basics of Git.docx*

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

PS D:\Git\_Pra> **git restore '.\Basics of Git.docx'**

Open the target file from the terminal. The changes made should be restored. Use:

PS D:\Git\_Pra> **& '.\Basics of Git.docx'**