

GIT AND GITHUB TUTORIAL

What is Git?

Git is a popular, free and open source version control system that is fast, scalable intelligently tracks changes in files and source code.

What is GitHub?

GitHub is a website that allows developers to store, manage and collaborate on their code using Git, in addition to hosting repositories and testing services.

GitHub Link - <https://github.com>

Using Git with command Line :

To start using Git, we are first going to open up our Command shell.

Setting up Git :

- Visual Studio Code
- Windows (Git Bash)
- Mac / Linux (Terminal)

Git Installation Commands :

- **For Linux :**
- `sudo apt update` - Updates the list of available packages and their versions from the repositories.

`sudo apt install git` - Installs Git, a version control system, on your system.

- **For MacOS :**
- `brew install git` –
- **brew:** This is the Homebrew package manager, used primarily on macOS to install, update, and manage software packages.
- **install:** Tells Homebrew you want to install something.

`git:` This is the name of the software package you want to install.

COMMANDS :-

Command: git --version

Checks the installed Git version to ensure Git is installed properly.

```
puspa@PUSPANJALI MINGW64 ~/OneDrive/Desktop/2141020006/2141020006_LOS_GIT TUTORIAL
$ git --version
git version 2.49.0.windows.1
```

Command: pwd

Displays the current working directory path.

```
puspa@PUSPANJALI MINGW64 ~/OneDrive/Desktop/2141020006/2141020006_LOS_GIT TUTORIAL
$ pwd
/c/Users/puspa/OneDrive/Desktop/2141020006/2141020006_LOS_GIT TUTORIAL
```

Command: ls -a

Lists all files in the directory, including hidden ones like `.git`.

```
puspa@PUSPANJALI MINGW64 ~/OneDrive/Desktop/2141020006/2141020006_LOS_GIT TUTORIAL
$ ls -a
./  ../
```

Command: git init

Creates a new local Git repository in the current directory.

```
puspa@PUSPANJALI MINGW64 ~/OneDrive/Desktop/2141020006/2141020006_LOS_GIT TUTORIAL
$ git init
Initialized empty Git repository in C:/Users/puspa/OneDrive/Desktop/2141020006/2141020006_LOS_GIT TUTORIAL/.git/
```

Command: git config --global user.name "Puspanjali Tarai"

Sets the global Git username (used in all repos).

Command: git config --global user.email "puspanjalitarai389@gmail.com"

Sets the global Git email (used in commits).

Command: git config --global user.name

Displays the current global Git username.

Command: git config --global user.email

Displays the current global Git email.

```

puspa@PUSPANJALI MINGW64 ~/OneDrive/Desktop/2141020006/2141020006_LOS_GIT TUTORI
AL (main)
$ git config --global user.name "Puspanjali Tarai"

puspa@PUSPANJALI MINGW64 ~/OneDrive/Desktop/2141020006/2141020006_LOS_GIT TUTORI
AL (main)
$ git config --global user.email "puspanjalitarai389@gmail.com"

puspa@PUSPANJALI MINGW64 ~/OneDrive/Desktop/2141020006/2141020006_LOS_GIT TUTORI
AL (main)
$ git config --global user.name
Puspanjali Tarai

puspa@PUSPANJALI MINGW64 ~/OneDrive/Desktop/2141020006/2141020006_LOS_GIT TUTORI
AL (main)
$ git config --global user.email
puspanjalitarai389@gmail.com

```

Command: git config --list

Lists all the Git configuration settings including user info, editor, etc.

```

puspa@PUSPANJALI MINGW64 ~/OneDrive/Desktop/2141020006/2141020006_LOS_GIT TUTORI
AL (main)
$ 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=schannel
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=main
user.name=Puspanjali Tarai
user.email=puspanjalitarai389@gmail.com
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true

```

Command: mkdir <directory_name>

Creates a new directory with the specified name.

Command: cd <directory_name>

Changes the current working directory to the specified one.

```

puspa@PUSPANJALI MINGW64 ~/OneDrive/Desktop/2141020006/2141020006_LOS_GIT TUTORI
AL (main)
$ mkdir Git

puspa@PUSPANJALI MINGW64 ~/OneDrive/Desktop/2141020006/2141020006_LOS_GIT TUTORI
AL (main)
$ cd Git

```

Command: echo "<content>" > <filename>

Creates a new file with the specified content using echo command.

```
puspa@PUSPANJALI MINGW64 ~/OneDrive/Desktop/2141020006/2141020006_LOS_GIT TUTORIAL/Git (main)
$ echo "Wipro Training" >>file1.txt
```

Command: cat <filename>

Displays the content of the specified file.

```
puspa@PUSPANJALI MINGW64 ~/OneDrive/Desktop/2141020006/2141020006_LOS_GIT TUTORIAL/Git (main)
$ cat >file1.txt
Puspanjali Tarai
ITER
WIPRO TRAINING
```

Command: git add .

Adds all modified and new files in the current directory to the staging area.

```
puspa@PUSPANJALI MINGW64 ~/OneDrive/Desktop/2141020006/2141020006_LOS_GIT TUTORIAL/Git (main)
$ git add .
```

Command: git add <filename>

Adds the specified file to the staging area in Git.

Command: git commit -m "message"

Commits the staged changes with a descriptive commit message.

```
puspa@PUSPANJALI MINGW64 ~/OneDrive/Desktop/2141020006/2141020006_LOS_GIT TUTORIAL/Git (main)
$ git add file1.txt
warning: in the working copy of 'Git/file1.txt', LF will be replaced by CRLF the next time Git touches it

puspa@PUSPANJALI MINGW64 ~/OneDrive/Desktop/2141020006/2141020006_LOS_GIT TUTORIAL/Git (main)
$ git commit -m "file updated"
[main (root-commit) 4882daf] file updated
1 file changed, 4 insertions(+)
create mode 100644 Git/file1.txt
```

Command: ls

Lists all files and directories in the current working directory.

```
puspa@PUSPANJALI MINGW64 ~/OneDrive/Desktop/2141020006
$ ls
2141020006_LOS_Ass3/ '2141020006_LOS_GIT TUTORIAL' file1.txt
```

Command: git status

Displays the status of the working directory and staging area.

```
puspa@PUSPANJALI MINGW64 ~/OneDrive/Desktop/2141020006/2141020006_LOS_GIT TUTORI
AL/Git (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:   file1.txt
```

Command: git log

Shows a list of all commits made in the current branch.

```
puspa@PUSPANJALI MINGW64 ~/OneDrive/Desktop/2141020006/2141020006_LOS_GIT TUTORI
AL/Git (main)
$ git log
commit 4882dafa03a9bb32dec5fb7c94f14d95d220329b (HEAD -> main)
Author: Puspanjali Tarai <puspanjalitarai389@gmail.com>
Date:   Wed Apr 9 00:39:22 2025 +0530

    file updated
```

Command: git commit --help

Displays help information about the `git commit` command.

```

puspa@PUSPANJALI MINGW64 ~/OneDrive/Desktop/2141020006/2141020006_LOS_GIT TUTORIAL/Git (main)
$ git commit -help
usage: git commit [-a | --interactive | --patch] [-s] [-v] [-u[<mode>]] [--amend |
--dry-run] [(--c | -C | --squash) <commit> | --fixup [(amend|reword):]<commit>]
[-F <file> | -m <msg>] [--reset-author] [--allow-empty]
[--allow-empty-message] [--no-verify] [-e] [--author=<author>]
[--date=<date>] [--cleanup=<mode>] [--no-status]
[-i | -o] [--pathspec-from-file=<file>] [--pathspec-file-nul]
[(->trailer <token>[(=|:)<value>])...] [-S[<keyid>]]
[--] [<pathspec>...]

-q, --[no-]quiet          suppress summary after successful commit
-v, --[no-]verbose        show diff in commit message template

Commit message options
-F, --[no-]file <file>   read message from file
--[no-]author <author>   override author for commit
--[no-]date <date>        override date for commit
-m, --[no-]message <message>
                           commit message
-c, --[no-]reedit-message <commit>
                           reuse and edit message from specified commit
-C, --[no-]reuse-message <commit>
                           reuse message from specified commit
--[no-]fixup [(amend|reword):]<commit>
                           use autosquash formatted message to fixup or amend/reword
                           specified commit
--[no-]squash <commit>    use autosquash formatted message to squash specified commit
--[no-]reset-author       the commit is authored by me now (used with -C/-c/--amend)
--trailer <trailer>       add custom trailer(s)
-s, --[no-]signoff        add a Signed-off-by trailer
-t, --[no-]template <file>
                           use specified template file
-e, --[no-]edit            force edit of commit
--[no-]cleanup <mode>     how to strip spaces and #comments from message
--[no-]status             include status in commit message template
-S, --[no-]gpg-sign [=<key-id>]

```

Command: git branch

Lists all local branches in the repository.

```

puspa@PUSPANJALI MINGW64 ~/OneDrive/Desktop/2141020006/2141020006_LOS_GIT TUTORIAL/Git (main)
$ git branch
* main

puspa@PUSPANJALI MINGW64 ~/OneDrive/Desktop/2141020006/2141020006_LOS_GIT TUTORIAL/Git (main)
$ git branch
HEAD main

```

Command: git branch <branch-name>

Creates a new branch with the specified name.

```

puspa@PUSPANJALI MINGW64 ~/OneDrive/Desktop/2141020006/2141020006_LOS_GIT TUTORIAL/Git (main)
$ git branch "wipro"

```

Command: `git merge <branch-name>`

Merges the specified branch into the current branch.

```
puspa@PUSPANJALI MINGW64 ~/OneDrive/Desktop/2141020006/2141020006_LOS_GIT TUTORIAL/Git (main)
$ git branch "wipro"
```

Command: `git checkout <branch-name>`

Switches to the specified branch.

```
puspa@PUSPANJALI MINGW64 ~/OneDrive/Desktop/2141020006/2141020006_LOS_GIT TUTORIAL/Git (main)
$ git checkout wipro
D      Git/file1.txt
Switched to branch 'wipro'
```