

# GIT SETUP

- **The Git init** -command creates a new Git repository. It can be used to convert an existing, unversioned project to a Git repository or initialize a new, empty repository.
- **The git status**-command provides a summary of the current state of a Git repository. It displays information about the working directory and the staging area, but not the commit history.
- The **Git add**- command is used in Git to add changes from the working directory to the staging area (also known as the index). The staging area acts as a buffer between your working directory and the project's history, allowing you to selectively prepare changes for the next commit.
- The **git commit** -command in Git is used to record a snapshot of the project's currently staged changes in the local repository. It essentially creates a "save point" in your project's history.
- **git add filename.txt**-It tells Git that you intend to include the current state of `filename.txt` in the upcoming commit.
- **Git branch**- in Git are independent lines of development. When working locally, changes are committed to a specific branch.
- **Git push**- is used to synchronize these local branch changes with a remote repository.



```
MINGW64/c/Users/Admin/OneDrive/Desktop/bhu
Admin@DESKTOP-MTI78B7 MINGW64 ~/OneDrive/Desktop/bhu (master)
$ git init
Reinitialized existing Git repository in C:/Users/Admin/OneDrive/Desktop/bhu/.git/

Admin@DESKTOP-MTI78B7 MINGW64 ~/OneDrive/Desktop/bhu (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   bhu.txt

Admin@DESKTOP-MTI78B7 MINGW64 ~/OneDrive/Desktop/bhu (master)
$ git add bhu.txt

Admin@DESKTOP-MTI78B7 MINGW64 ~/OneDrive/Desktop/bhu (master)
$ git commit -m "first commit"
Author identity unknown

*** Please tell me who you are.

Run

    git config --global user.email "you@example.com"
    git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'Admin@DESKTOP-MTI78B7.(none)')
Admin@DESKTOP-MTI78B7 MINGW64 ~/OneDrive/Desktop/bhu (master)
$ ^C

Admin@DESKTOP-MTI78B7 MINGW64 ~/OneDrive/Desktop/bhu (master)
$ git config --global user.email "bhumika.varshney_cs24@gl.a.ac.in"

Admin@DESKTOP-MTI78B7 MINGW64 ~/OneDrive/Desktop/bhu (master)
$ git config --global user.name "bhumis567"

Admin@DESKTOP-MTI78B7 MINGW64 ~/OneDrive/Desktop/bhu (master)
$ git commit -m "first commit"
[master (root-commit) 7ed811c] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 bhu.txt
```

```
Admin@DESKTOP-MTI78B7 MINGW64 ~/OneDrive/Desktop/bhu (master)
$ git config --global user.email "bhumi.ka.varshney_cs24@gl.a.ac.in"

Admin@DESKTOP-MTI78B7 MINGW64 ~/OneDrive/Desktop/bhu (master)
$ git config --global user.name "bhumi5567"

Admin@DESKTOP-MTI78B7 MINGW64 ~/OneDrive/Desktop/bhu (master)
$ git commit -m "first commit"
[master (root-commit) 7ed811c] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 bhu.txt

Admin@DESKTOP-MTI78B7 MINGW64 ~/OneDrive/Desktop/bhu (master)
$ git branch -m main

Admin@DESKTOP-MTI78B7 MINGW64 ~/OneDrive/Desktop/bhu (main)
$ git remote add origin https://github.com/bhumi5567/bhumi.git
git: 'remote' is not a git command. See 'git --help'.

The most similar command is
remote

Admin@DESKTOP-MTI78B7 MINGW64 ~/OneDrive/Desktop/bhu (main)
$ git remote add origin http://github.com/bhumi5567/bhumi.git

Admin@DESKTOP-MTI78B7 MINGW64 ~/OneDrive/Desktop/bhu (main)
$ git push -m origin main
error: unknown switch 'm'
usage: git push [<options>] [<repository> [<refspec>...]]

-v, --[no-]verbose      be more verbose
-q, --[no-]quiet        be more quiet
--[no-]repo <repository>
                        repository
--[no-]all              push all branches
--[no-]branches        alias of --all
--[no-]mirror          mirror all refs
-d, --[no-]delete      delete refs
--[no-]tags            push tags (can't be used with --all or --branches or --mirror)
-n, --[no-]dry-run      dry run
--[no-]porcelain        machine-readable output
-f, --[no-]force        force updates
--[no-]force-with-lease[=<refname>:<expect>]
                        require old value of ref to be at this value
--[no-]force-if-includes
                        require remote updates to be integrated locally
--[no-]recurse-submodules (check|on-demand|no)
                        control recursive pushing of submodules
```