University of Tehran
Engineering college

Git tutorial

Mehrshad Hekmatara

بسم الله الرحمن الرحيم

What is git, github, gitlab?

Git is a version control system — a tool that helps you keep track of changes in your files (usually code), go back to earlier versions if needed, and collaborate with others without overwriting each other's work.

👉 Think of it like a "time machine + collaboration manager" for your projects.

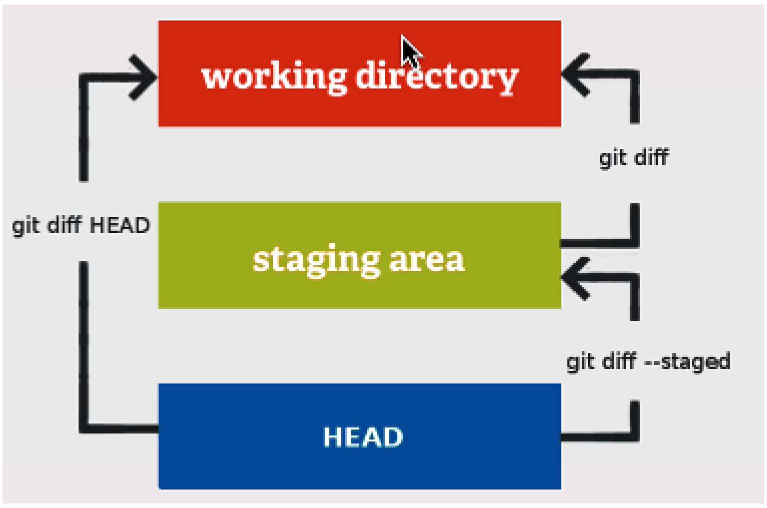
GitHub → An online platform (owned by Microsoft) that hosts Git repositories. It makes it easy to share your projects, collaborate, and use extra tools like pull requests, issues, and actions.

GitLab → Another platform for hosting Git repositories. It's similar to GitHub but offers built-in features for things like continuous integration (CI/CD) and project management. It can be used both online (like GitHub) or installed on your own servers.

👉 In short:

Git = the tool

GitHub / GitLab = places you can store and collaborate on Git projects



1. Working Directory

This is your project folder on your computer.

- It's where you create, edit, and delete files.
- Changes here are not yet tracked by Git until you add them.

2. Staging Area (also called *Index*)

Think of this as a waiting room for changes.

- When you run git add, you move changes from the working directory → to the staging area.
- Only the changes in the staging area will be included in the next commit.

3. HEAD

HEAD is like a pointer that tells you *where you are in your project's history*.

- Usually, HEAD points to the latest commit in the current branch.
- If you switch branches or commits, HEAD moves to that location.

What is repository?

Repository

A repository (repo) is the whole Git project.
It includes:

- The working directory (your project files)
- The .git folder (where Git stores history, commits, branches, and configurations)
- The staging area

👉 You can think of a repository as a complete box of your project with both your current files and their entire version history.

⚡ Tip: A common analogy

- Working Directory → Your notebook where you're writing drafts
- Staging Area → The table where you put pages you want to submit
- Repository → The archive where all submitted pages are permanently stored

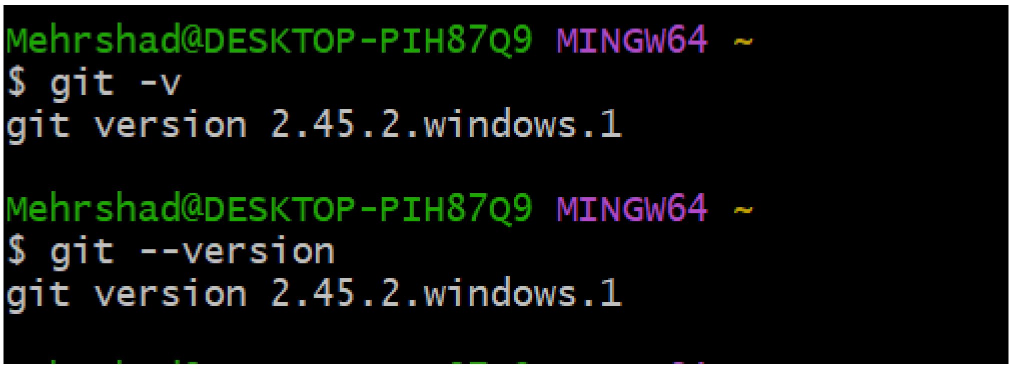
How to install git?

Windows

1. Go to **git-scm.com**
2. Download the Windows installer
3. Run the installer → keep default options (unless you know what to change)

4. After installation, open Git Bash or Command Prompt
5. Check installation:

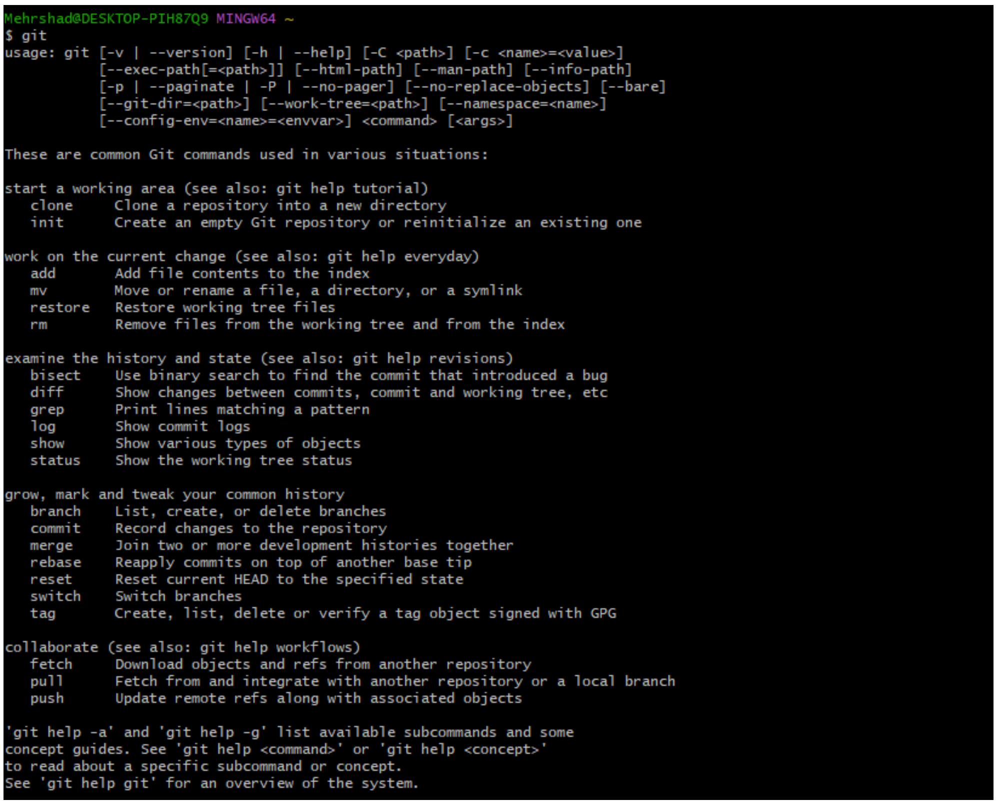
```
git -v
git --version
```



```
Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~
$ git -v
git version 2.45.2.windows.1

Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~
$ git --version
git version 2.45.2.windows.1
```

Checking all the commands :

```
git
```



```
Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~
$ git
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>]
           [--config-env=<name>=<envvar>] <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
   clone     Clone a repository into a new directory
   init      Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
   add       Add file contents to the index
   mv        Move or rename a file, a directory, or a symlink
   restore   Restore working tree files
   rm        Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
   bisect    Use binary search to find the commit that introduced a bug
   diff      Show changes between commits, commit and working tree, etc
   grep      Print lines matching a pattern
   log       Show commit logs
   show      Show various types of objects
   status    Show the working tree status

grow, mark and tweak your common history
   branch    List, create, or delete branches
   commit    Record changes to the repository
   merge     Join two or more development histories together
   rebase    Reapply commits on top of another base tip
   reset     Reset current HEAD to the specified state
   switch    Switch branches
   tag       Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
   fetch     Download objects and refs from another repository
   pull      Fetch from and integrate with another repository or a local branch
   push      Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.
```

Basic Command Line Commands in Git Bash

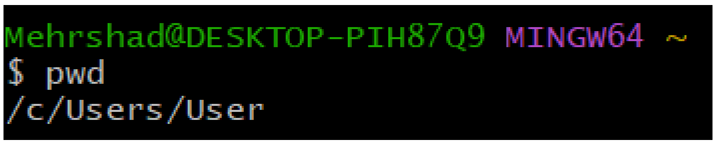
Git Bash is a command-line interface that emulates a Unix-like Bash shell on Windows, making it ideal for working with Git and other development tools. It supports many standard Bash commands for file navigation, manipulation, and more. This tutorial serves as a lesson plan (or "lesson note") to help you document and learn the essential basic and practical commands. I'll cover them in categories for clarity, with explanations, syntax, examples, and tips.

The commands are case-sensitive, and paths use forward slashes (/) like in Unix (e.g., C:/Users/YourName). You can run these in Git Bash by typing them and pressing Enter.

Navigation Commands

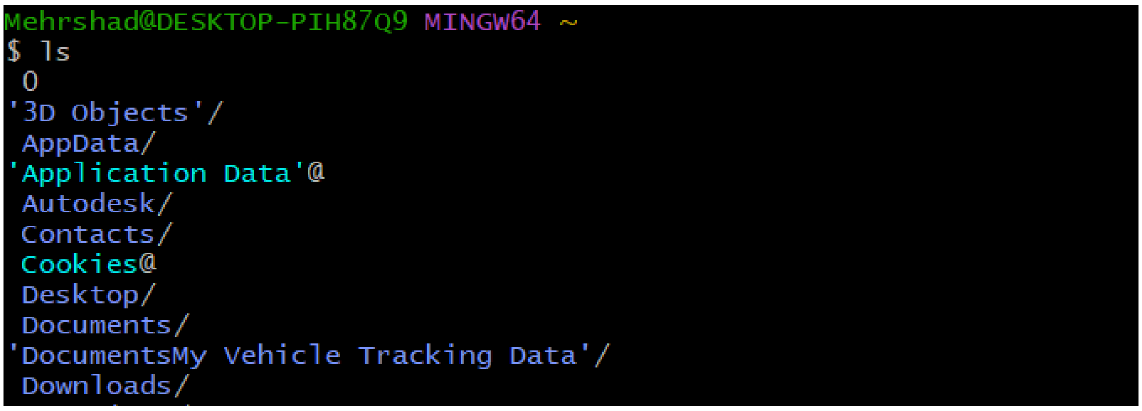
These help you move around directories and see where you are.

- **pwd** (Print Working Directory): Shows the current directory path.
    - Syntax: pwd
    - Tip: Useful to confirm your location before running other commands.



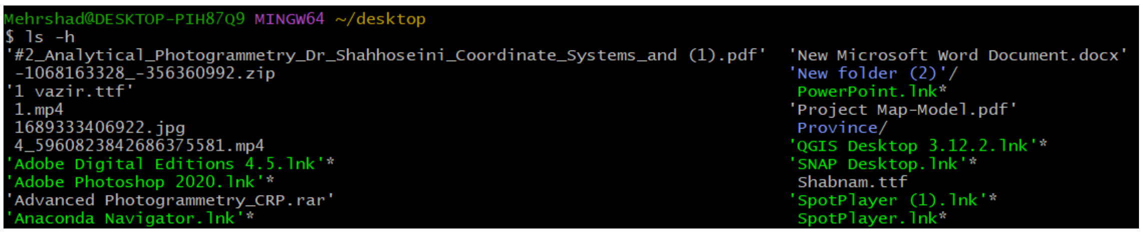
```
Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~
$ pwd
/c/Users/User
```

- **ls** (List): Lists files and directories in the current directory.
    - Syntax: ls [options] [directory]

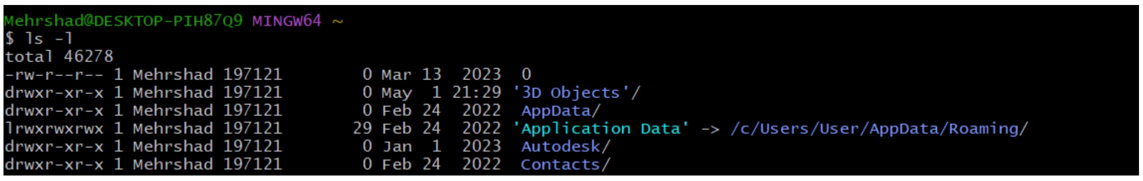


```
Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~
$ ls
 0
'3D Objects'/
 AppData/
'Application Data'@
 Autodesk/
 Contacts/
 Cookies@
 Desktop/
 Documents/
'DocumentsMy Vehicle Tracking Data'/
 Downloads/
```

    - Common options: -l (long format with details), -a (show hidden files), -h (human-readable sizes).

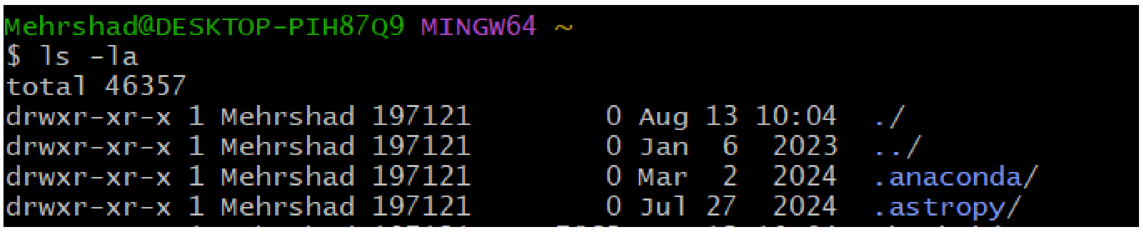


```
Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/desktop
$ ls -h
'#2_Analytical_Photogrammetry_Dr_Shahhoseini_Coordinate_Systems_and (1).pdf'  'New Microsoft Word Document.docx'
 -1068163328_-356360992.zip                                                   'New folder (2)'/
'1 vazir.ttf'                                                                  PowerPoint.lnk*
 1.mp4                                                                        'Project Map-Model.pdf'
 1689333406922.jpg                                                             Province/
 4_5960823842686375581.mp4                                                    'QGIS Desktop 3.12.2.lnk'*
'Adobe Digital Editions 4.5.lnk'*                                             'SNAP Desktop.lnk'*
'Adobe Photoshop 2020.lnk'*                                                    Shabnam.ttf
'Advanced Photogrammetry_CRP.rar'                                             'SpotPlayer (1).lnk'*
'Anaconda Navigator.lnk'*                                                      SpotPlayer.lnk*
```



```
Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~
$ ls -l
total 46278
-rw-r--r-- 1 Mehrshad 197121        0 Mar 13  2023  0
drwxr-xr-x 1 Mehrshad 197121        0 May  1 21:29 '3D Objects'/
drwxr-xr-x 1 Mehrshad 197121        0 Feb 24  2022  AppData/
lrwxrwxrwx 1 Mehrshad 197121       29 Feb 24  2022 'Application Data' -> /c/Users/User/AppData/Roaming/
drwxr-xr-x 1 Mehrshad 197121        0 Jan  1  2023  Autodesk/
drwxr-xr-x 1 Mehrshad 197121        0 Feb 24  2022  Contacts/
```

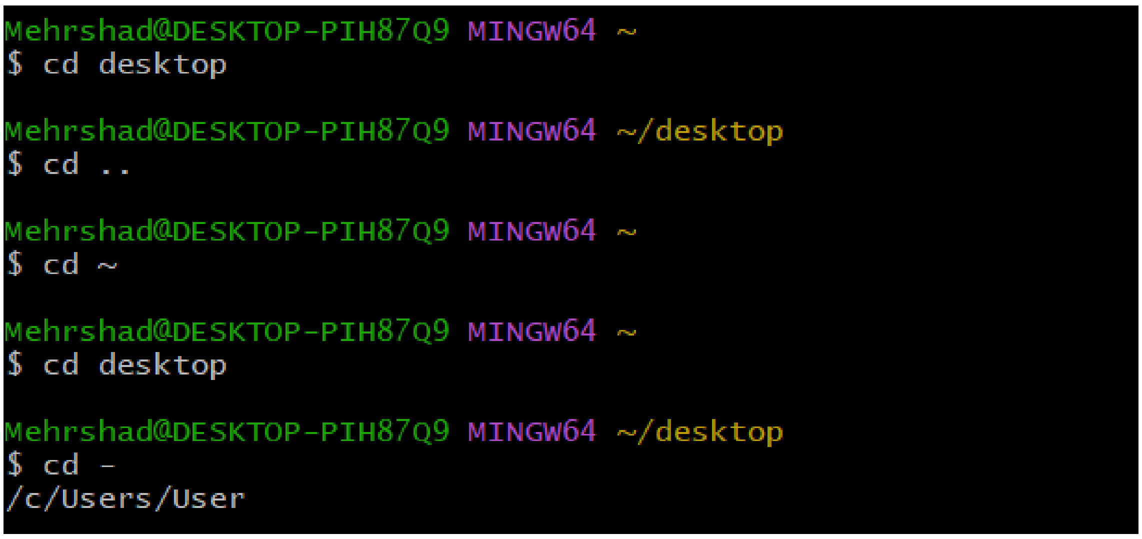
- Example: ls → Lists items like file.txt folder/.
- Example: ls -la → Shows detailed list including hidden files (e.g., .git).



```
Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~
$ ls -la
total 46357
drwxr-xr-x 1 Mehrshad 197121        0 Aug 13 10:04  ./
drwxr-xr-x 1 Mehrshad 197121        0 Jan  6  2023  ../
drwxr-xr-x 1 Mehrshad 197121        0 Mar  2  2024  .anaconda/
drwxr-xr-x 1 Mehrshad 197121        0 Jul 27  2024  .astropy/
```

- Tip: Use ls /path/to/dir to list a specific directory without changing to it.

- **cd** (Change Directory): Moves to a different directory.
  - Syntax: cd [directory]
  - Special shortcuts: cd ~ (go to home directory), cd .. (go up one level), cd - (go to previous directory), cd / (go to root).
  - Example: cd Documents → Changes to the "Documents" folder.
  - Example: cd ../Projects → Goes up one level, then into "Projects".
  - Tip: Use quotes if the folder name has spaces: cd "My Folder". Tab-completion helps auto-fill paths.



```
Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~
$ cd desktop

Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/desktop
$ cd ..

Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~
$ cd ~

Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~
$ cd desktop

Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/desktop
$ cd -
/c/Users/User
```

File and Directory Creation Commands

These create new files or folders.

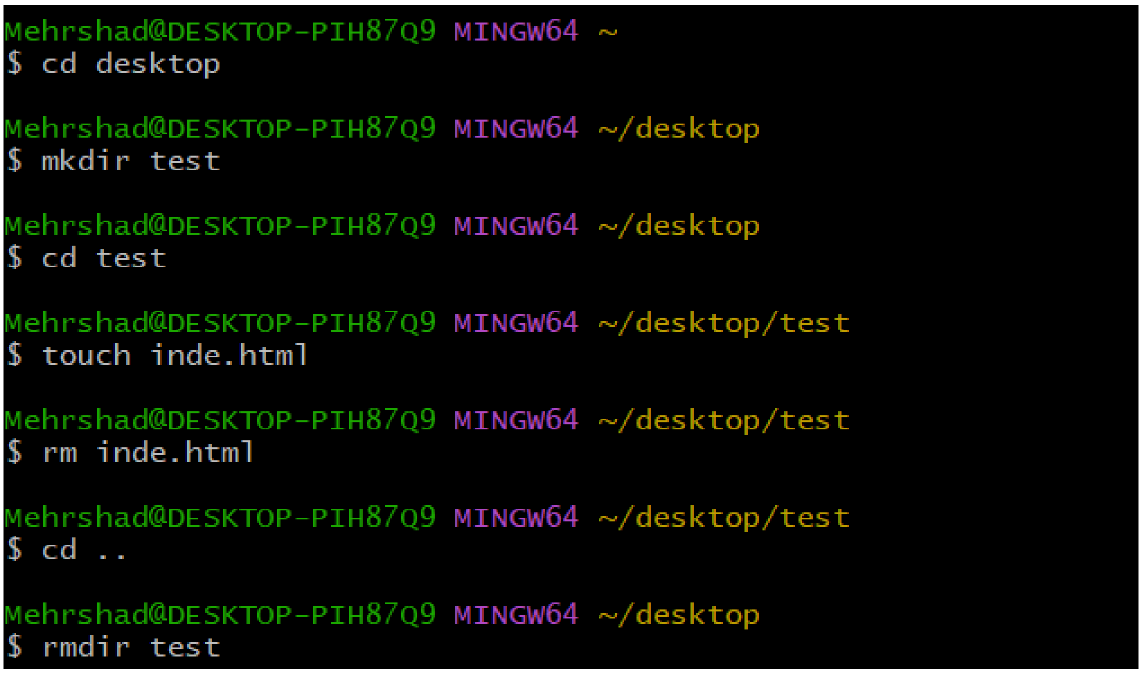
- **touch**: Creates a new empty file (or updates timestamp of an existing one).
  - Syntax: touch filename
  - Tip: Can create multiple files: touch file1.txt file2.txt.
- **mkdir** (Make Directory): Creates a new directory.
  - Syntax: mkdir [options] directory
  - Tip: Use quotes for names with spaces.

File and Directory Removal Commands

Be careful—these can delete data permanently!

- **rm** (Remove): Deletes files or directories.
  - Syntax: rm [options] file-or-dir
  - Options: -r (recursive, for directories), -f (force, no prompts), -i (interactive, prompt before delete).
  - Example: rm file.txt → Deletes "file.txt".
  - Example: rm -r folder → Deletes "folder" and its contents.
  - Tip: Avoid rm -rf /—it can wipe your system! Use with caution.

- **rmdir** (Remove Directory): Deletes an empty directory.
    - Syntax: rmdir directory
    - Tip: Safer than rm -r for directories, as it won't work if not empty.

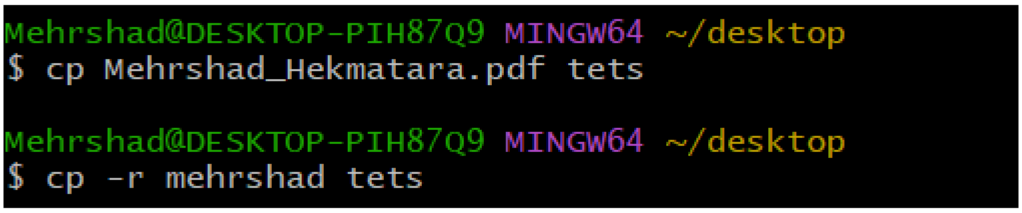


```
Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~
$ cd desktop

Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/desktop
$ mkdir test

Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/desktop
$ cd test

Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/desktop/test
$ touch inde.html

Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/desktop/test
$ rm inde.html

Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/desktop/test
$ cd ..

Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/desktop
$ rmdir test
```

Copying and Moving Commands

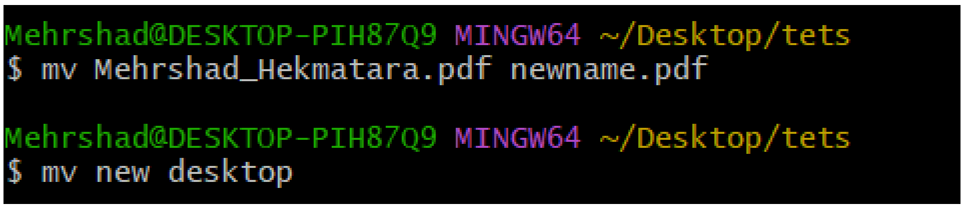
These handle duplicating or relocating files/directories.

- **cp** (Copy): Copies files or directories.
    - Syntax: cp [options] source destination
    - Options: -r (recursive for directories)
    - Example: cp file.txt backup/ → Copies "file.txt" to "backup" folder.
    - Example: cp -r folder newlocation/ → Copies entire folder



```
Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/desktop
$ cp Mehrshad_Hekmatara.pdf tets

Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/desktop
$ cp -r mehrshad tets
```

- **mv** (Move): renames files/directories.
    - Syntax: mv [options] source destination
    - Example: mv file.txt newname.txt → Renames the file.
    - Tip: Great for renaming: no separate "rename" command needed.

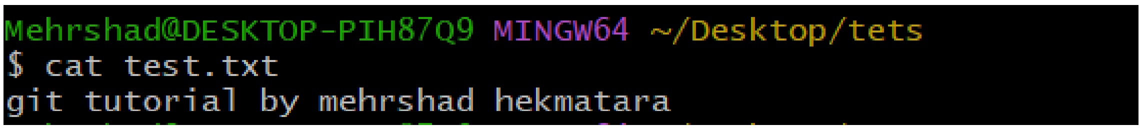


```
Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/Desktop/tets
$ mv Mehrshad_Hekmatara.pdf newname.pdf

Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/Desktop/tets
$ mv new desktop
```

Viewing and Editing File Content

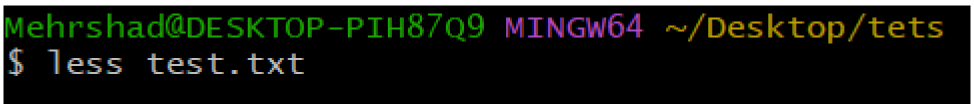
These let you inspect or modify files without opening a full editor.

- **cat** (Concatenate): Displays file content (or combines files).
    - Syntax: cat [file]
    - Example: cat file.txt → Prints the contents to the terminal.



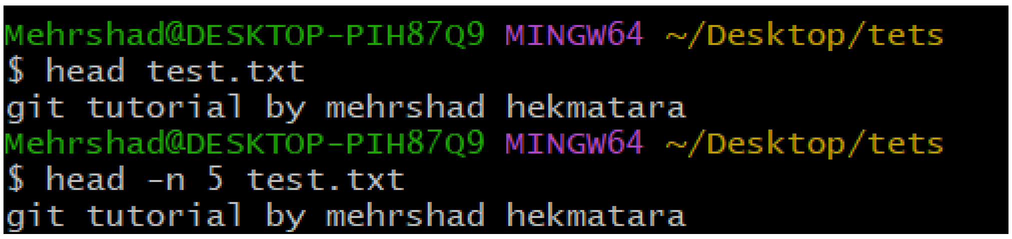
```
Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/Desktop/tets
$ cat test.txt
git tutorial by mehrshad hekmatara
```

- **less** or **more**: Views file content one page at a time.
    - Syntax: less file.txt file.txt
    - Controls: Space (next page), q (quit), /search (find text).
    - Example: less log.txt → Scrolls through the file.



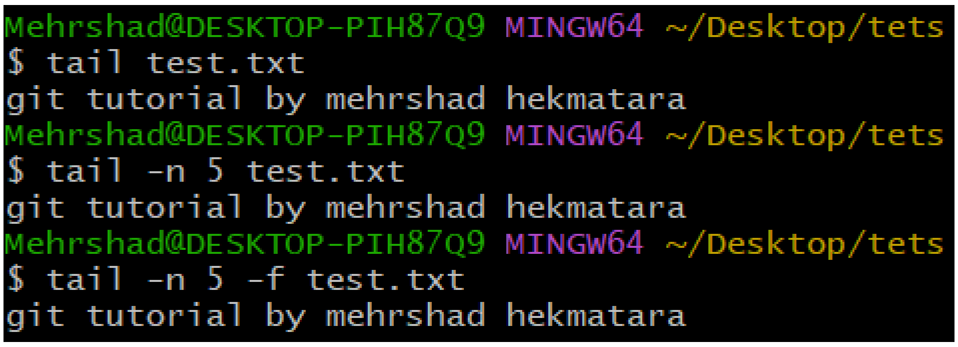
```
Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/Desktop/tets
$ less test.txt
```

- **head**: Shows the first few lines of a file.
    - Syntax: head [options] file
    - Option: -n N (first N lines).
    - Example: head -n 5 file.txt → Shows first 5 lines.



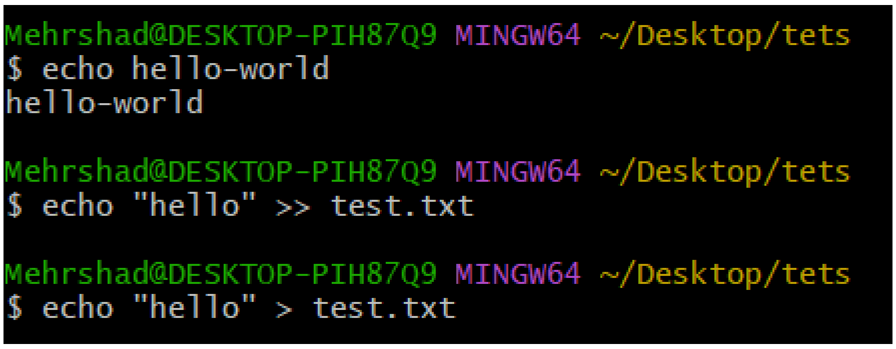
```
Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/Desktop/tets
$ head test.txt
git tutorial by mehrshad hekmatara
Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/Desktop/tets
$ head -n 5 test.txt
git tutorial by mehrshad hekmatara
```

- **tail**: Shows the last few lines of a file.
  - Syntax: tail [options] file
  - Option: -n N (last N lines), -f (follow for live updates).
  - Example: tail -n 10 log.txt → Shows last 10 lines.
  - Tip: Useful for logs: tail -f access.log watches in real-time.



```
Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/Desktop/tets
$ tail test.txt
git tutorial by mehrshad hekmatara
Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/Desktop/tets
$ tail -n 5 test.txt
git tutorial by mehrshad hekmatara
Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/Desktop/tets
$ tail -n 5 -f test.txt
git tutorial by mehrshad hekmatara
```

- **echo**: Prints text to the terminal or a file.
  - Syntax: echo "text" [> file]
  - Example: echo "Hello World" → Prints "Hello World".
  - Example: echo "Hello" > greeting.txt → Writes to file (overwrites).
  - Example: echo "Append" >> greeting.txt → Appends to file.
  - Tip: Use for quick file creation or testing.

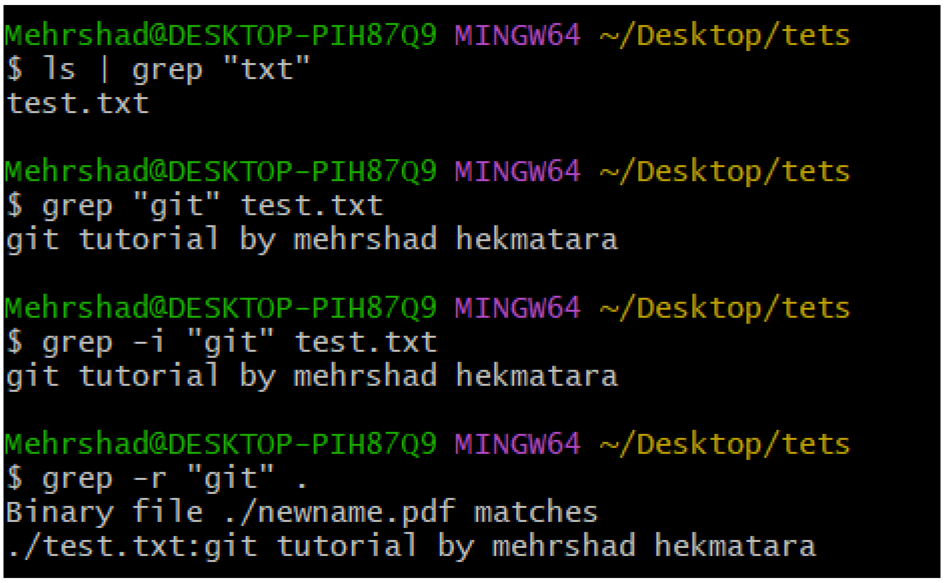


```
Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/Desktop/tets
$ echo hello-world
hello-world

Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/Desktop/tets
$ echo "hello" >> test.txt

Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/Desktop/tets
$ echo "hello" > test.txt
```

Searching and Finding Commands

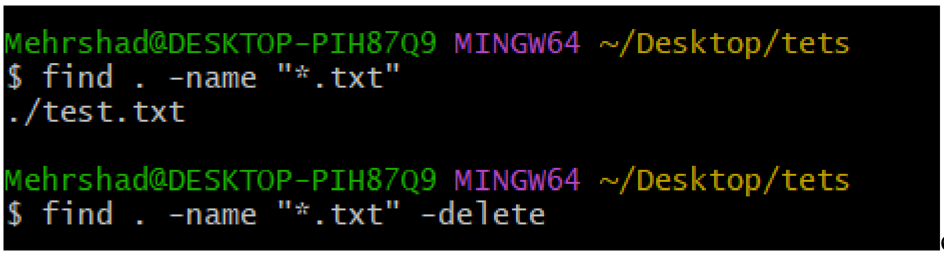
These help locate files or text.

- **grep** (Global Regular Expression Print): Searches for text in files.
    - Syntax: grep [options] "pattern" file
    - Options: -i (case-insensitive), -r (recursive in directories), -n (line numbers).
    - Example: grep "error" log.txt → Finds lines with "error".
    - Example: grep -r "TODO" . → Searches all files in current directory.
    - Tip: Pipe with other commands: ls | grep "txt" (lists only .txt files).



```
Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/Desktop/tets
$ ls | grep "txt"
test.txt

Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/Desktop/tets
$ grep "git" test.txt
git tutorial by mehrshad hekmatara

Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/Desktop/tets
$ grep -i "git" test.txt
git tutorial by mehrshad hekmatara

Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/Desktop/tets
$ grep -r "git" .
Binary file ./newname.pdf matches
./test.txt:git tutorial by mehrshad hekmatara
```

- **find**: Searches for files/directories by name, type, etc.
    - Syntax: find [path] [options]
    - Example: find . -name "*.txt" → Finds all .txt files in current dir and subdirs.
    - Tip: Combine with actions: find . -name "*.tmp" -delete (deletes matches).



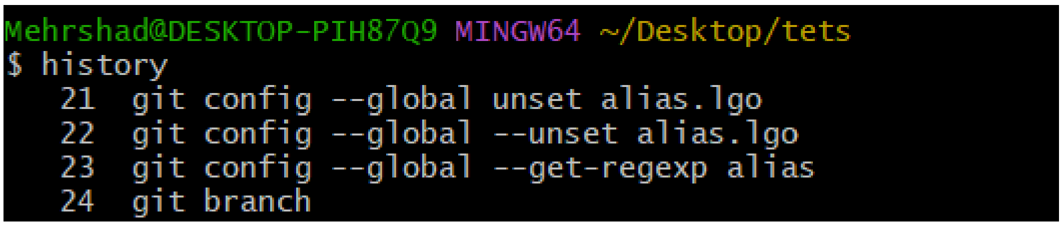
```
Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/Desktop/tets
$ find . -name "*.txt"
./test.txt

Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/Desktop/tets
$ find . -name "*.txt" -delete
```

c

Other Practical Commands

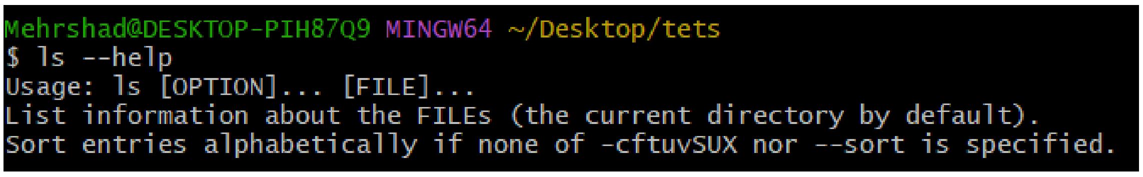
These are handy for everyday use.

- **clear**: Clears the terminal screen.
  - Syntax: clear
  - Tip: Shortcut: Ctrl+L.
- **history**: Shows command history.
  - Syntax: history
  - Tip: Use !n to rerun command #n from history.



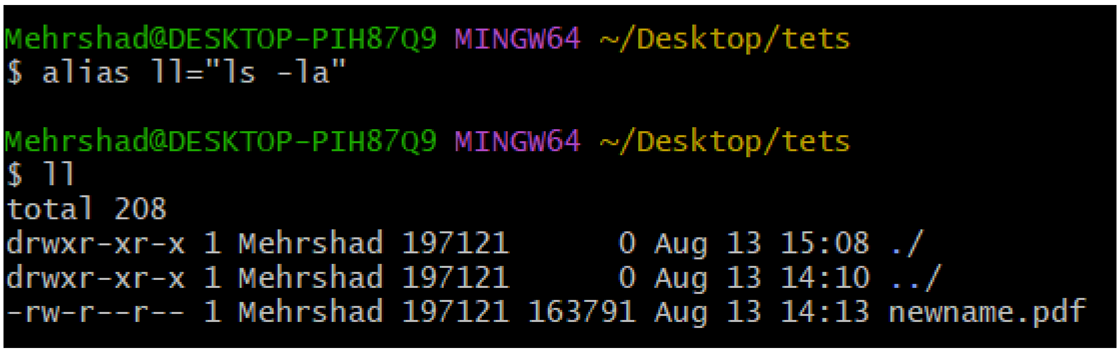
```
Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/Desktop/tets
$ history
   21  git config --global unset alias.lgo
   22  git config --global --unset alias.lgo
   23  git config --global --get-regexp alias
   24  git branch
```

- **--help** (Manual): Shows help for a command.
  - Syntax: command --help
  - Example: man ls → Displays ls manual.



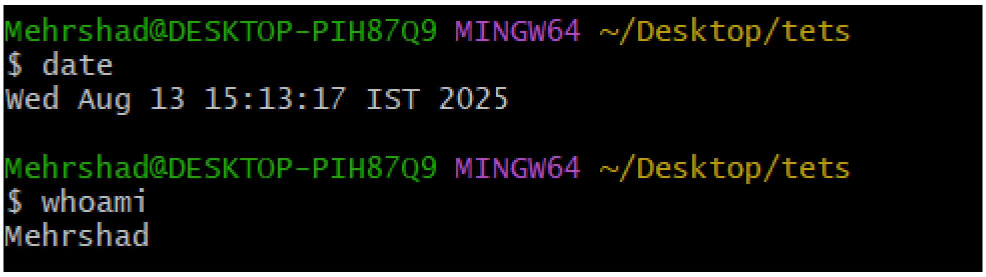
```
Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/Desktop/tets
$ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.
```

- **alias**: Creates shortcuts for commands.
  - Syntax: alias name='command'
  - Example: alias ll='ls -la' → Now ll runs detailed ls.



```
Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/Desktop/tets
$ alias ll="ls -la"

Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/Desktop/tets
$ ll
total 208
drwxr-xr-x 1 Mehrshad 197121      0 Aug 13 15:08 ./
drwxr-xr-x 1 Mehrshad 197121      0 Aug 13 14:10 ../
-rw-r--r-- 1 Mehrshad 197121 163791 Aug 13 14:13 newname.pdf
```

- **pwd**, **whoami**, **date**: Quick info commands.
    - whoami: Shows current username.
    - date: Shows current date/time.



```
Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/Desktop/tets
$ date
Wed Aug 13 15:13:17 IST 2025

Mehrshad@DESKTOP-PIH87Q9 MINGW64 ~/Desktop/tets
$ whoami
Mehrshad
```