

1) Navigation Commands (Moving Around Folders)

1. `pwd`

Print Working Directory — shows where you currently are.

`pwd`

Example output:

/home/ahmed/projects

2. `ls`

List files/folders in the current directory.

`ls`

Useful variants:

`ls -l # detailed list (permissions, size, date)`

`ls -a # show hidden files (.git, .config)`

3. `cd <filename>`

Go into a folder.

`cd Documents`

`cd myproject`

4. `cd ..`

Go one step back (to parent directory).

`cd ..`

5. `cd ../../..`

Go two steps back.

`cd ../../..`

6. `cd ~`

Go to home directory.

`cd ~`

7. `cd /path/to/folder`

Go to a specific location (absolute path).

```
cd /Users/Ahmed/Desktop
```

2) File/Folder Creation

1. **mkdir <foldername>**

Make a new folder.

```
mkdir projects
```

```
mkdir git-practice
```

2. **touch <filename>**

Create an empty file.

```
touch file.txt
```

```
touch main.py
```

3. **echo "text" > file.txt**

Create a file with content (or replace content).

```
echo "Hello World" > readme.txt
```

4. **echo "more text" >> file.txt**

Append text (add without removing old content).

```
echo "New line" >> readme.txt
```

3) File/Folder Operations

1. **rm <file>**

Remove a file.

WARNING: No recycle bin. It's gone.

```
rm readme.txt
```

2. **rm -r <folder>**

Remove a folder and everything inside.

Use carefully.

```
rm -r myproject
```

3. **cp <src> <dest>**

Copy a file.

```
cp a.txt b.txt
```

Copy folder:

```
cp -r folder1 folder2
```

4. mv <old> <new>

Move or rename file/folder.

```
mv oldname.txt newname.txt
```

```
mv file.txt folder/
```

4) Dots Meaning in Terminal

. (single dot)

Means current directory.

Used in many commands:

```
git add . # add everything in the current folder
```

```
./program.sh # run a program in current folder
```

.. (double dot)

Means parent directory (one level above).

```
cd ..
```

```
ls ..
```

../filename

A folder inside your parent directory.

```
cd ../Documents
```

5) Practical Example

Copy and follow:

```
mkdir git-basics
```

```
cd git-basics
```

```
touch a.txt
```

```
echo "hello" >> a.txt
```

```
ls  
pwd  
mkdir notes  
cd notes  
pwd  
cd ..  
rm a.txt  
cd ..  
pwd
```

This small sequence will make everything click.

6) Bonus Commands You Will Use with Git

clear
Clears the terminal screen.

cat <file>
Show file content.

cat readme.txt

code . (if using VS Code)
Open current folder in VS Code.

code .