Useful Command Prompt and Git Commands

# Command Prompt Commands

**1. File & Folder Operations**

|  |  |
| --- | --- |
| Command | Purpose |
| dir | List files and folders in current directory |
| cd foldername | Change directory into a folder |
| cd .. | Go back one folder |
| cd \ | Go to root of the current drive |
| md foldername / mkdir foldername | Create a new folder |
| rd foldername / rmdir foldername | Remove a folder |
| del filename | Delete a file |
| copy file1 file2 | Copy file1 to file2 |
| move file1 foldername\ | Move a file to a folder |
| rename oldname newname | Rename a file or folder |

**2. Create & Edit Files**

|  |  |
| --- | --- |
| Command | Purpose |
| echo Hello > file.txt | Create a file with text |
| echo. > file.txt | Create a blank file |
| type nul > file.txt | Another way to make a blank file |
| copy con file.txt | Create a file and type content (end with Ctrl+Z + Enter) |
| notepad file.txt | Open file in Notepad (creates if not exists) |
| type file.txt | Display contents of the file |
| echo This is my first line > newFile.txt | This **creates** or **overwrites** newFile.txt with the line This is my first line |
| echo This is another line >> newFile.txt | This **adds a new line** to the file without removing existing content. |
| Using **copy con** to type directly into the file  copy con newFile.txt | type your content line by line.  When you're done, press:  Ctrl + Z  (then press Enter) |

**3. Disk & Drive Commands**

|  |  |
| --- | --- |
| Command | Purpose |
| cd /d D:\folder | Change to another drive and folder |
| chkdsk | Check disk for errors |
| format X: | Format a drive (erases everything!) |
| diskpart | Open disk partition tool (admin only) |

**4. System Info & Control**

|  |  |
| --- | --- |
| Command | Purpose |
| cls | Clear the screen |
| exit | Exit the command prompt |
| ver | Show Windows version |
| hostname | Display your computer's name |
| systeminfo | Show system information |
| tasklist | List running processes |
| taskkill /IM notepad.exe /F | Kill a process by name |
| ipconfig | Show IP/network details |
| ping google.com | Test internet connection |
| shutdown /s /t 0 | Shut down the PC |
| shutdown /r /t 0 | Restart the PC |

**5. User & Permissions (Admin)**

|  |  |
| --- | --- |
| Command | Purpose |
| net user | List all users |
| net user username | View details of a user |
| net user username /add | Add a new user |
| net user username /del | Delete a user |

**6. Batch Scripting / Advanced**

|  |  |
| --- | --- |
| Command | Purpose |
| && | Run multiple commands if first succeeds |
| || | Run second command if first fails |
| > | Redirect output to a file (overwrite) |
| >> | Append output to a file |
| set var=value | Set a variable |
| %var% | Access variable |
| for %%x in (\*.txt) do echo %%x | Loop through files |

# Git Commands

**1. Git Setup**

|  |  |
| --- | --- |
| Command | Purpose |
| git config --global user.name "Your Name" | Set your name |
| git config --global user.email "you@example.com" | Set your email |
| git config --global core.editor notepad | Set default text editor |
| git config --list | View all current settings |

**2. Git Repository Basics**

|  |  |
| --- | --- |
| Command | Purpose |
| git init | Create a new Git repository |
| git clone <repo-url> | Copy a remote repo to your computer |
| git status | Check current branch and changes |
| git add <file> | Stage file for commit |
| git add . | Stage all changes |
| git commit -m "message" | Commit staged changes |
| git log | Show commit history |
| git show | Show details of latest commit |
| git diff | See unstaged changes |
| git diff --staged | See staged but uncommitted changes |

**3. Branching and Merging**

|  |  |
| --- | --- |
| Command | Purpose |
| git branch | List all branches |
| git branch branch-name | Create a new branch |
| git checkout branch-name | Switch to a branch |
| git switch branch-name | Switch to a branch (modern way) |
| git merge branch-name | Merge branch into current |
| git branch -d branch-name | Delete a branch |
| git checkout -b new-branch | Create & switch to new branch |

**4. Remote Repositories**

|  |  |
| --- | --- |
| Command | Purpose |
| git remote -v | Show remote URLs |
| git remote add origin <url> | Link local repo to remote |
| git push -u origin main | Push main branch to GitHub |
| git push | Push commits to remote |
| git pull | Pull changes from remote |
| git fetch | Fetch changes without merging |

**5. Undoing Changes**

|  |  |
| --- | --- |
| Command | Purpose |
| git restore <file> | Undo working directory changes |
| git checkout -- <file> | Old way to restore files |
| git reset <file> | Unstage a file |
| git reset --hard | Reset all changes to last commit |
| git revert <commit> | Revert a commit with a new commit |

**6. Stashing**

|  |  |
| --- | --- |
| Command | Purpose |
| git stash | Temporarily save uncommitted changes |
| git stash list | Show stashes |
| git stash apply | Reapply stash |
| git stash drop | Delete the latest stash |

**7. Tagging**

|  |  |
| --- | --- |
| Command | Purpose |
| git tag | List tags |
| git tag v1.0 | Create lightweight tag |
| git tag -a v1.0 -m "Version 1" | Create annotated tag |
| git push origin v1.0 | Push tag to remote |

**8. Other Useful Commands**

|  |  |
| --- | --- |
| Command | Purpose |
| git clean -f | Remove untracked files |
| git blame <file> | See who changed each line |
| git archive | Create archive of repo |
| git reflog | View history of HEAD changes |

**Step 1-** Initialize git -> **git init**

**Step 2-** Use **git status** to see what is committed (tracked) and what is not.

**Step 3-** Put untracked files on the stag

**git add .** (to put all the untracked file in that folder on the stage)

**git fileName.txt** (to put a particular file on the stage)

**Step 4-** Click the picture

**git commit -m “fileName.txt is added”** (commit with any message)

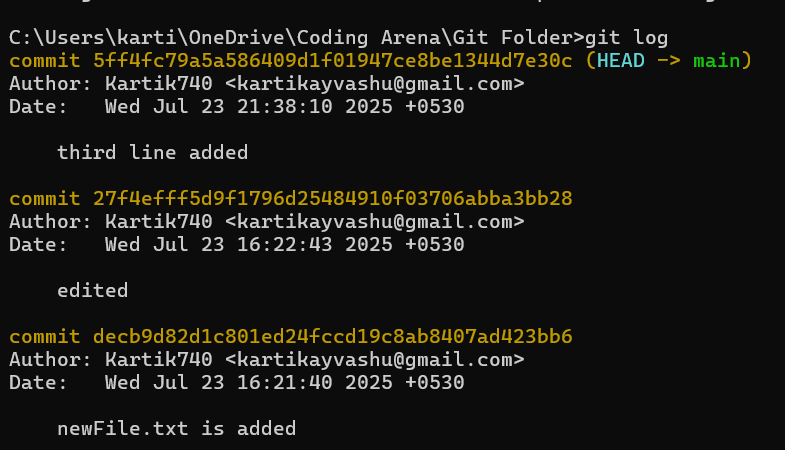
* If any file is staged and you want to unstage it-

**git restore –staged fileName.txt**

* To see the history –

**git log**

* If want to delete any history copy the id just below it (all the history above this id will be deleted, you can’t delete any history in between )

****

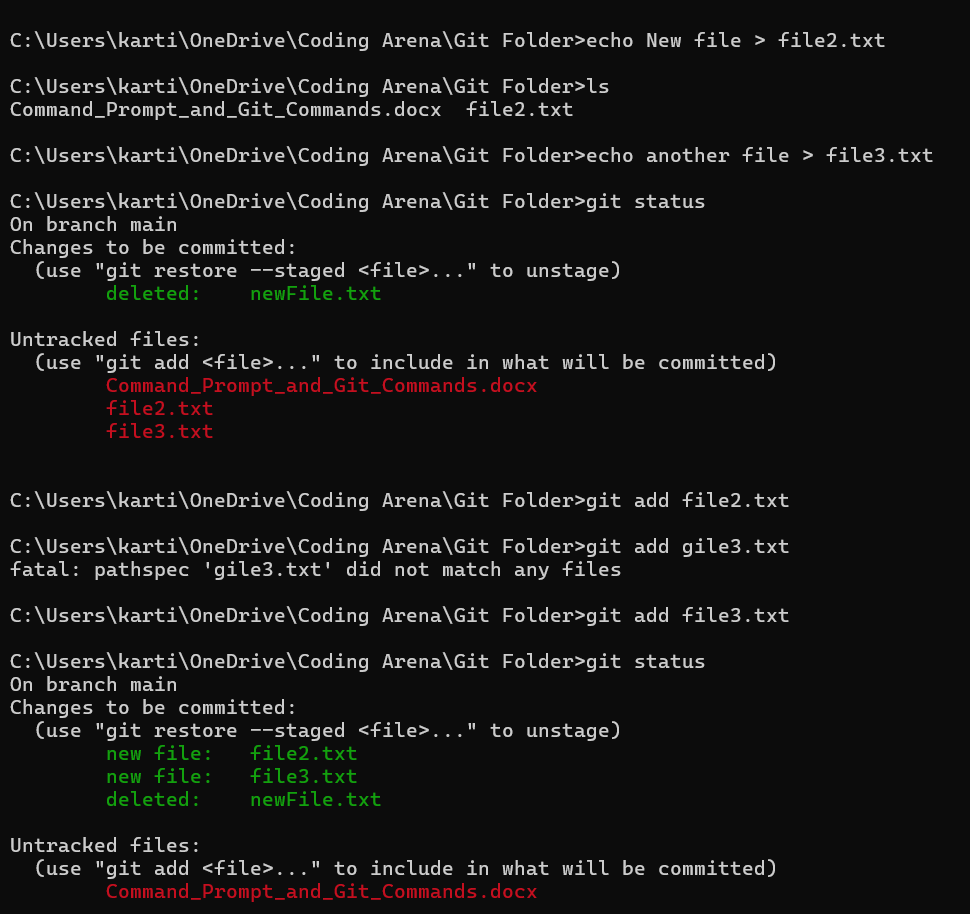
All these history will be deleted

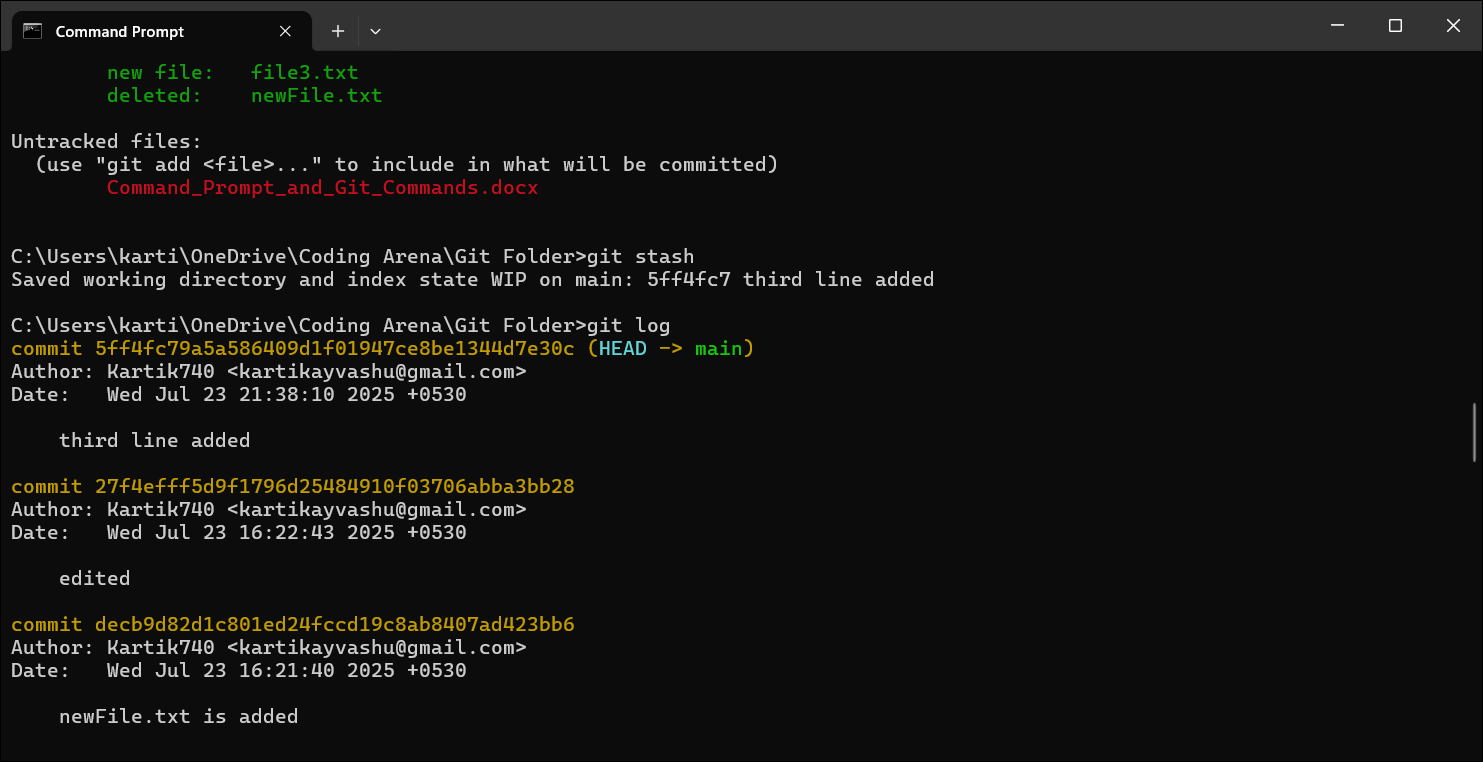
If want to delete all history above it, copy this id



**git reset decb9d82d1c801ed24fccd19c8ab8407ad423bb6**

* If you have done some changes and don’t want to commit it (add to history) but also don’t want to loose it, then you can bring those files on stage and send them to backstage, so when you want you can bring them back

****

****

**git stash**

* To bring back all that who are on backstage (not committed) use-

**git stash pop**

* To remove the file on the backstage (on stash), use-

**git stash clear**

* To attach a GitHub repository to local project –

**git remote add origin** [**https://github.com/Kartik740/First-repo.git**](https://github.com/Kartik740/First-repo.git)

* To push the changes to GitHub repository-

**git push origin main**

* To clone any repository on local device use –

**git clone** [**https://github.com/Kartik740/First-repo.git**](https://github.com/Kartik740/First-repo.git)

* To make new branch –

**git branch branchName**

* To change head to new branch use -

**git checkout branchName**

* To push this branch –

**git push origin branchName**

* **To work on open source project fork the project, clone on local device, make branch, change head to new branch, edit, commit, and push changes, them make a pull request.**
* If you have commited then want to undo **-> git reset id (unstaged) -> git add . (staged) -> git stash (back staged) -> git push origin branchName -f** ( f for forced push as the online repository already has commits and we want to change it, then we have to forced push)