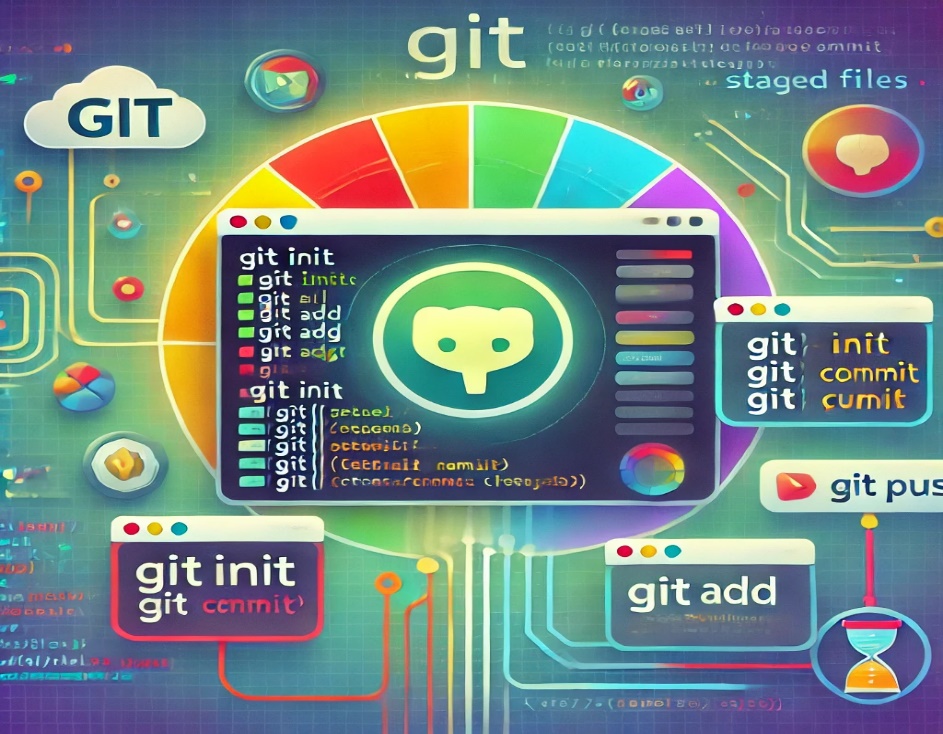
**A Step by Step Guide to Essential Git Commands**

****

What is Git?

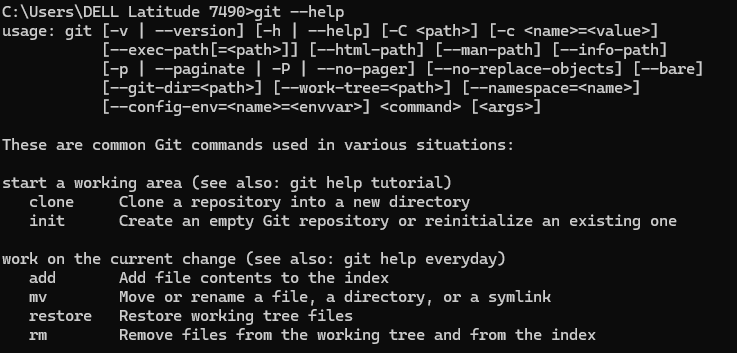
* Git is the open source distributed version control system that facilitates GitHub activities on your laptop or desktop. It is powerful version control system used by developers around the world to manage code and track changes. Whether you are working on a project or collaborating with a team, Git simplifies the process of managing the code files. In this article, I will walk through essential Git commands, providing examples to help you understand how they work.

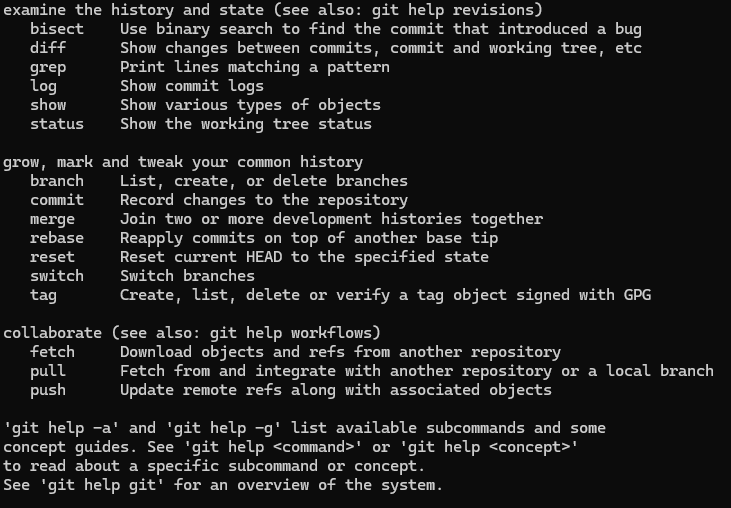
**Git Commands**

$git –version - Prints the Git suite version that the **git** program came from.



$git –help - Prints the synopsis and a list of the most commonly used commands



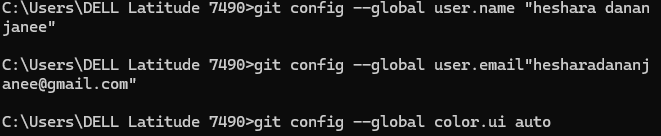


**Configure Tooling**

$git config --global user.name "[name]" - Sets the name you want attached to your commit transactions

$git config --global user.email "[email address]" -Sets the email you want attached to your commit transactions

$git config --global color.ui - auto Enables helpful colorization of command line output



**Branches**

Branches are an important part of working with Git. Any commits you make will be made on the branch you're currently “checked out” to. Use git status to see which branch that is.

$git branch [branch-name] - Creates a new branch

$git checkout [branch-name] - Switches to the specified branch and updates the working directory

$git merge [branch] - Combines the specified branch’s history into the current branch. This is usually done in pull requests, but is an important Git operation.

$git branch -d [branch-name] -Deletes the specified branch

**Create repositories**

$ git init - Turn an existing directory into a git repository

$git clone [url] - Clone (download) a repository that already exists on GitHub, including all of the files, branches, and commits



**Synchronize changes**

$git fetch -Downloads all history from the remote tracking branches

$git merge- Combines remote tracking branch into current local branch

$git push -Uploads all local branch commits to GitHub

$ git pull -Updates your current local working branch with all new commits from the corresponding remote branch on GitHub. git pull is a combination of git fetch and git merge

**Make Changes**

Browse and inspect the evolution of project files

$git log - Lists version history for the current branch

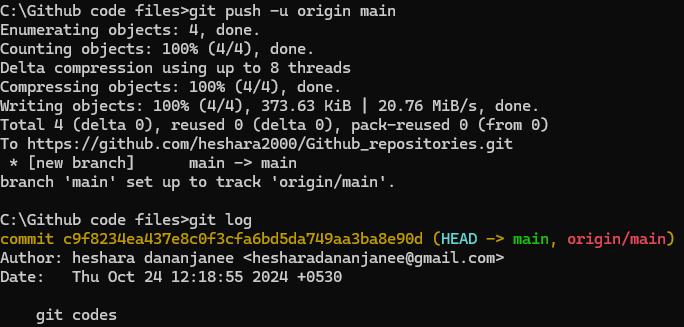
$ git log --follow [file] - Lists version history for a file, including renames

$git diff [first-branch]...[second-branch] -Shows content differences between two branches

$git show [commit] - Outputs metadata and content changes of the specified commit

$git add [file] -Snapshots the file in preparation for versioning

$git commit -m "[descriptive message]" -Records file snapshots permanently in version history



**Redo Commits**

Erase mistakes and craft replacement history

$ git reset [commit] -Undoes all commits after [commit], preserving changes locally

$ git reset --hard [commit] -Discards all history and changes back to the specified commit

**Conclusion**

Mastering Git commands is crucial for developers who want to collaborate efficiently and maintain control over their codebase. By understanding these core commands, you’ll be better equipped to manage your projects and contribute to others. I hope this guide helped you get started with Git, and I encourage you to try these commands on your own projects.

**Resources:**

Git Official Documentation <https://git-scm.com/docs/git>

Video Tutorial: <https://www.youtube.com/watch?v=8JJ101D3knE>