



ASCII Pre-Requisite :

Configuration

- `git config --global user.name "Your Name"` : Sets the name you want attached to your commit messages.
- `git config --global user.email "you@example.com"` : Sets the email you want attached to your commit messages.
- `git config --list` : Lists all the Git configurations.

Repository Setup

- `git init` : Initializes a new Git repository.
- `git clone <repo>` : Clones a repository into a new directory.

Basic Snapshotting

- `git add <file>` : Adds a file to the staging area.
- `git add .` : Adds all changes in the current directory to the staging area.
- `git commit -m "message"` : Commits the staged snapshot with a message.
- `git status` : Displays the state of the working directory and staging area.
- `git diff` : Shows the differences between files in the working directory and the staging area.

- `git reset <file>` : Unstages a file without discarding changes.
- `git rm <file>` : Removes a file from the working directory and stages the deletion.
- `git mv <oldfilename> <newfilename>` : Renames a file and stages the change.

Branching and Merging

- `git branch` : Lists all branches in the repository.
- `git branch <branchname>` : Creates a new branch.
- `git checkout <branchname>` : Switches to the specified branch.
- `git checkout -b <branchname>` : Creates and switches to a new branch.
- `git merge <branchname>` : Merges the specified branch into the current branch.
- `git branch -d <branchname>` : Deletes the specified branch.

Remote Repositories

- `git remote add <name> <url>` : Adds a remote repository.
- `git remote -v` : Lists the remote connections.
- `git fetch <remote>` : Fetches the branches and their commits from the remote repository.
- `git pull <remote>` : Fetches and merges changes from the remote repository into the current branch.
- `git push <remote> <branch>` : Pushes the changes from the local branch to the remote repository.
- `git remote rm <name>` : Removes a remote repository.

Inspecting and Comparing

- `git log` : Shows the commit history for the current branch.
- `git log --oneline` : Shows a compressed view of the commit history.
- `git show <commit>` : Shows details of a specific commit.
- `git diff <branch1> <branch2>` : Shows differences between two branches.
- `git blame <file>` : Shows who made changes to each line of a file.

Undoing Changes

- `git revert <commit>` : Reverts a specific commit.
- `git reset --hard <commit>` : Resets the current branch to a specific commit, discarding all changes.
- `git clean -f` : Removes untracked files from the working directory.
- `git stash` : Temporarily stores changes in the working directory.
- `git stash pop` : Restores the stashed changes.

Tagging

- `git tag` : Lists all tags.
- `git tag -a <tagname> -m "message"` : Creates an annotated tag.
- `git push <remote> --tags` : Pushes all tags to the remote repository.

Advanced Commands

- `git rebase <branch>` : Applies commits from one branch onto another.
- `git cherry-pick <commit>` : Applies a specific commit from another branch.
- `git bisect` : Uses binary search to find the commit that introduced a bug.
- `git reflog` : Shows a log of all reference updates.

Collaboration

- `git pull request` : (Platform-specific) Creates a pull request (on GitHub, GitLab, etc.).
- `git submodule add <url> <path>` : Adds a submodule to your repository.
- `git submodule update --init --recursive` : Clones all submodules and checks out the appropriate commits.

GitHub/GitLab Specific

- `gh repo clone <repo>` : Clones a GitHub repository using GitHub CLI.
- `glab repo clone <repo>` : Clones a GitLab repository using GitLab CLI.

Oh My Git!

An open source game about learning Git!

 <https://blinry.itch.io/oh-my-git>



How to level up your GitHub profile README (Basic to Pro!)

✨ To try everything Brilliant has to offer—free—for a full 30 days, visit <https://brilliant.org/MagdelineHuang/> or click on the link in the description. The first 200 people to use the link will get 20% off

📺 <https://youtu.be/DWFs6aqknqw?si=7ssFab2TuSVTzjib>



<https://code.visualstudio.com/download>

Extensions :

Live Server

Prettier (Format) On save ,Default Formatter

Code Spell Checker

Babel Javascript

Error Lens

ESLint

AutoOpen MarkDown Preview

Git History and Git Graph