Git help

The Three States

`Git Directory`	The .git directory is where Git stores the metadata and object database for the repository
`Working Directory`	A copy of one version of the git project, taken from compressed database in the .git directory
`Staging Area/Index`	File that stores information about what will next be committed into the git repository

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 colorizations of command line input

Create Repositories

`git init [project-name]`	Creates a new local repository with the specified name
`git clone [url]`	Downloads a project and its entire version history

Make Changes

`git status`	Lists all new or modified files to be committed
`git diff`	Shows file differences not yet staged
`git add [file]`	Snapshots the file in preparation for versioning
`git diff --staged`	Shows file differences between staging and the last file version
`git reset [file]`	Unstages the file, but preserves its contents
`git commit -m "[descriptive message]"`	Records the file snapshots permanently in version history

Group Changes

`git branch`	Lists all local branches in the current repository
`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
`git branch -d [branch-name]`	Deletes the specified branch
`git remote add [remote-name] [url]`	Add a new remote git repository as a shortname
`git remote -v`	Lists all remote git repositories

Refactor Filenames

`git rm [file]`	Deletes the file from the working directory and stages the deletion
`git rm --cached [file]`	Removes the file from version control but preserves the file locally
`git mv [file-original] [file-renamed]`	Changes the file name and prepares it for commit

Suppress Tracking

`*.log build/ temp-*`	A text file named .gitignore suppresses accidental versioning of files and paths matching the specified patterns
`git ls-files --other --ignored --exclude-standard`	Lists all ignored files in this project

Save Fragments

`git stash`	Temporarily stores all modified tracking files
`git stash save [message]`	Save local modifications to a new stash
`git stash pop`	Restores the most recently stashed files
`git stash list`	Lists all stashed changesets
`git stash show`	Show the changes recorded in the stash
`git stash drop`	Discards the most recently stashed changeset

Review History

`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

Redo Commits

`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

Synchronize Changes

`git fetch [bookmark] [branch]`	Downloads all history from the repository bookmark, optionally specifying branch
`git merge [bookmark]/[branch]`	Combines bookmark's branch into current local branches
`git push [alias] [branch]`	Uploads all local branch commits to GitHub
`git push [alias] :[branch]`	Deletes remote branch
`git pull`	Downloads bookmark history and incorporates changes
`git pull --rebase`	Downloads bookmark history and incorporates your changes on top of remote changes
`git rebase --interactive --autosquash HEAD~N`	Squash N last commits
`git cherry-pick -n <sha>`	Cherry-pick a commit
`git revert -n <sha>`	Revert a commit