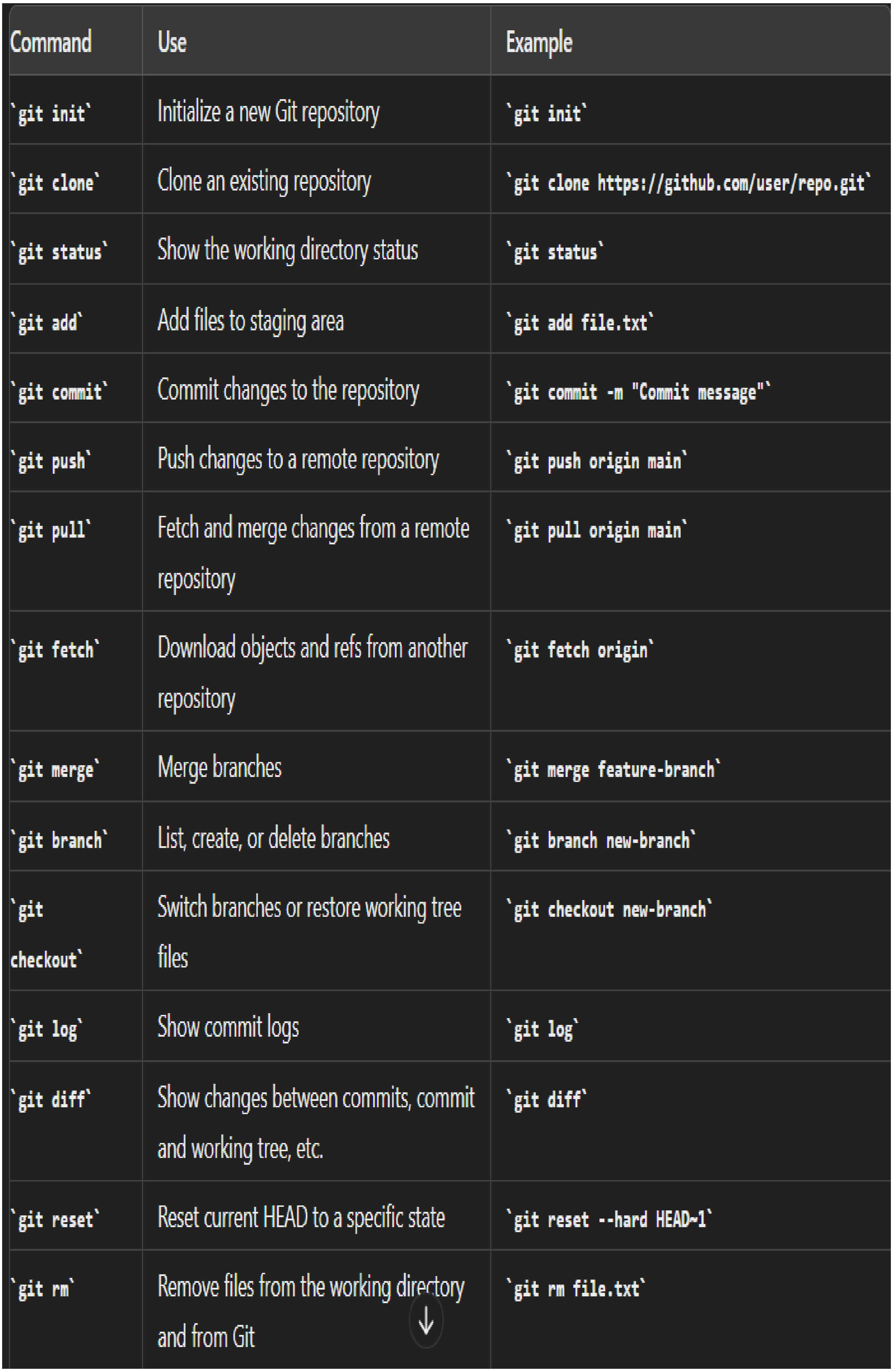
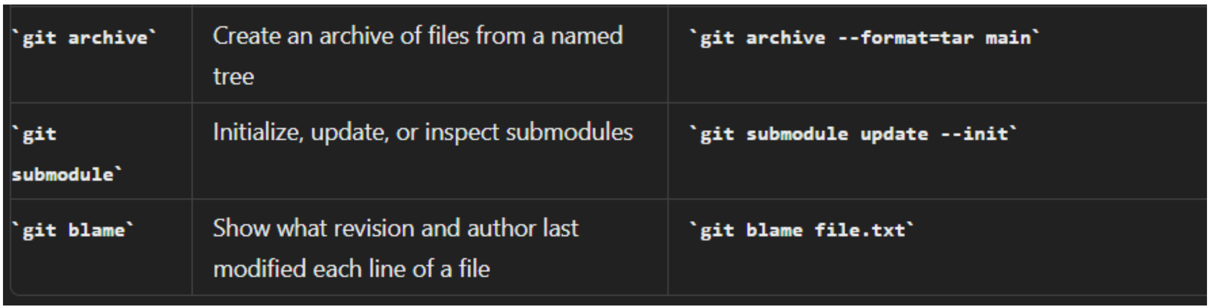
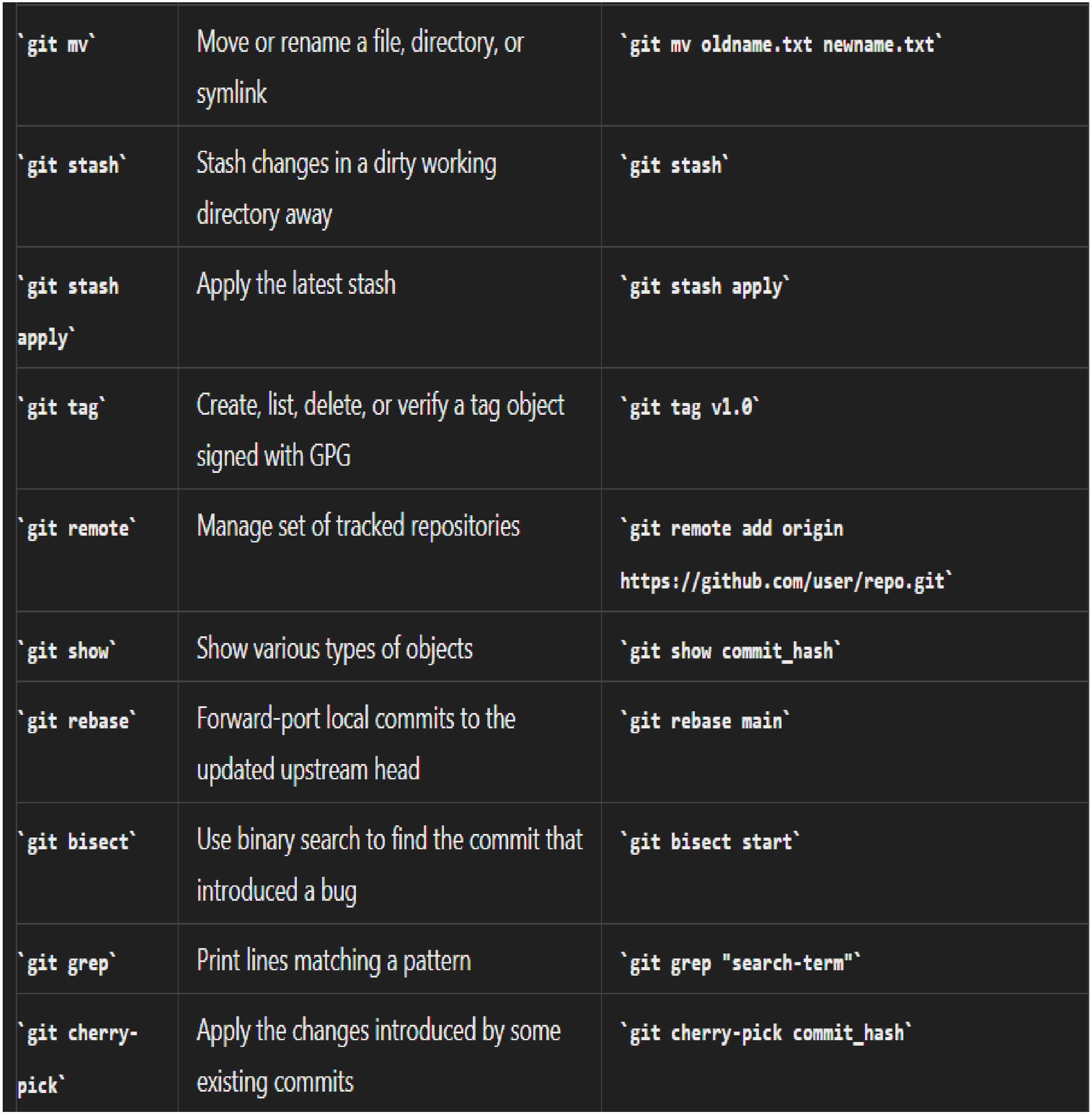


Command	Use	Example
`git init`	Initialize a new Git repository	`git init`
`git clone`	Clone an existing repository	`git clone https://github.com/user/repo.git`
`git status`	Show the working directory status	`git status`
`git add`	Add files to staging area	`git add file.txt`
`git commit`	Commit changes to the repository	`git commit -m "Commit message"`
`git push`	Push changes to a remote repository	`git push origin main`
`git pull`	Fetch and merge changes from a remote repository	`git pull origin main`
`git fetch`	Download objects and refs from another repository	`git fetch origin`
`git merge`	Merge branches	`git merge feature-branch`
`git branch`	List, create, or delete branches	`git branch new-branch`
`git checkout`	Switch branches or restore working tree files	`git checkout new-branch`
`git log`	Show commit logs	`git log`
`git diff`	Show changes between commits, commit and working tree, etc.	`git diff`
`git reset`	Reset current HEAD to a specific state	`git reset --hard HEAD~1`
`git rm`	Remove files from the working directory and from Git	`git rm file.txt`



`git archive`	Create an archive of files from a named tree	`git archive --format=tar main`
`git submodule`	Initialize, update, or inspect submodules	`git submodule update --init`
`git blame`	Show what revision and author last modified each line of a file	`git blame file.txt`



`git mv`	Move or rename a file, directory, or symlink	`git mv oldname.txt newname.txt`
`git stash`	Stash changes in a dirty working directory away	`git stash`
`git stash apply`	Apply the latest stash	`git stash apply`
`git tag`	Create, list, delete, or verify a tag object signed with GPG	`git tag v1.0`
`git remote`	Manage set of tracked repositories	`git remote add origin https://github.com/user/repo.git`
`git show`	Show various types of objects	`git show commit_hash`
`git rebase`	Forward-port local commits to the updated upstream head	`git rebase main`
`git bisect`	Use binary search to find the commit that introduced a bug	`git bisect start`
`git grep`	Print lines matching a pattern	`git grep "search-term"`
`git cherry-pick`	Apply the changes introduced by some existing commits	`git cherry-pick commit_hash`