Local Changes

Display the status of modified files

```
git status
```

Add a file to staging as it looks right now

```
git add [file_name]
```

Add a folder to staging as it looks right now

```
git add [folder_name]
```

Commit staged files in a new commit

```
git commit -m "descriptive_message"
```

Add all files to staging and commit them at once

```
git commit -am "descriptive_message"
```

Unstage a file while retaining the changes

```
git reset [file_name]
```

Diff of what is changed but not staged

```
git diff
```

Diff of what has changed between staged changes and the last commit

```
git diff --staged
```

Branches

List all branches. The current one is marked with *

```
git branch
```

Create a new branch

```
git branch [branch_name]
```

Switch to a branch

```
git checkout [branch_name]
```

Create a new branch and switch to it

```
git checkout -b [branch_name]
```

Switch to the previously checked out branch

```
git checkout -
```

Rename a branch

```
git checkout -m [new_branch]
```

Delete a branch, locally

```
git branch -d [branch_name]
```

Merge another branch into the current one

```
git merge [branch_name]
```

Working with a Remote Repository

Fetch and merge all commits from the tracked remote branch

```
git pull
```

Fetch and merge all commits from a specific remote branch

```
git pull [alias] [branch_name]
```

Fetch recent changes from the tracked remote branch but don't merge them

```
git fetch
```

Fetch all branches, tags, and references from the remote

```
git fetch --all
```

Create and switch to a local branch based on a remote branch

```
git switch -c [local_branch_name] [remote_name/remote_branch_name]
```

Push all local branch commits to the tracked remote branch

```
git push
```

Push all local branch commits to a specific remote branch

```
git push [alias] [branch_name]
```

Add a new remote repository with the given alias

```
git remote add [alias] [repo_url]
```

Display a list of remote repositories and their URLs

```
git remote -v
```

Commit History

Show all commits in the current branch?s history

```
git log
```

Show all commits in the current branch?s history by printing each commit on a single line

```
git log --oneline
```

Show number of commits per author on all branches, excluding merge commits.

```
git shortlog -s -n --all --no-merges
```

Show number of commits per author on a branch, excluding merge commits.

```
git shortlog -s -n [branch_name] --no-merges
```

Show number of commits per author on all branches, including merge commits.

```
git shortlog -s -n --all
```

Show number of commits per author on a branch, including merge commits.

```
git shortlog -s -n [branch_name]
```

Rebase

Reapply commits from the current branch on top of another base

```
git rebase [branch_name]
```

Abort a rebase

```
git rebase ?-abort
```

Continue a rebase after resolving conflicts

```
git rebase ?-continue
```

Undo

Revert the changes in a commit and record them in a new commit

```
git revert [commit]
```

Reset to a previous commit and preserve the changes made since [commit] as unstaged

```
git reset [commit]
```

Reset to a previous commit and discard the changes made since the [commit]

```
git reset --hard [commit]
```

Stash

Stash modified and staged changes

```
git stash
```

Stash modified and staged changes with a custom message

```
git stash push -m "message"
```

Stash a selected file by specifying a path

```
git stash push src/custom.css
```

List all stashed changesets

```
git stash list
```

Restore the most recently stashed changeset and delete it

```
git stash pop
```

Delete the most recently stashed changeset

```
git stash drop
```

Tags

Create a new tag

```
git tag "tagname"
```

List all tags

```
git tag
```

Delete a tag

```
git tag -d "tagname"
```

Repository Setup

Create an empty repository in the current folder

```
git init
```

Create an empty repository in a specific folder

```
git init [folder_name]
```

Clone a repository and add it to the current folder

```
git clone [repo_url]
```

Clone a repository to a specific folder

```
git clone [repo_url] [folder_name]
```

Global Config

Set the username

```
git config --global user.name "user_name"
```

Set the user email

```
git config --global user.email "user_email"
```

Set automatic command line coloring

```
git config --global color.ui auto
```