

# Git Command Cheat Sheet

---

## 1. Initial Setup

Before starting with Git, you need to configure your identity (name and email):

Set your name:

```
git config --global user.name "Your Name"
```

Set your email:

```
git config --global user.email "you@example.com"
```

## 2. Creating a Repository

Initialize a new Git repository in your current directory:

```
git init
```

Clone an existing remote repository:

```
git clone <repository-url>
```

## 3. Working with Files

Check the status of your working directory and staging area:

```
git status
```

Track a specific file or changes in the working directory:

```
git add <file>
```

Add all files to the staging area:

```
git add .
```

## 4. Committing Changes

Commit the staged changes with a message:

```
git commit -m "Commit message here"
```

Commit all tracked changes (bypassing the staging area):

```
git commit -a -m "Commit message here"
```

## 5. Branching and Merging

Create a new branch:

```
git branch <branch-name>
```

Switch to another branch:

```
git checkout <branch-name>
```

Create and switch to a new branch in one command:

```
git checkout -b <branch-name>
```

Merge a branch into the current branch:

```
git merge <branch-name>
```

## 6. Working with Remote Repositories

Add a remote repository:

```
git remote add origin <remote-repository-url>
```

View remote URLs:

```
git remote -v
```

Push changes to the remote repository:

```
git push origin <branch-name>
```

Pull the latest changes from the remote repository:

```
git pull origin <branch-name>
```

## 7. Stashing Changes

Save your changes temporarily (stash them):

```
git stash
```

Apply the last stashed changes:

```
git stash apply
```

## 8. Viewing Commit History

View the commit history:

```
git log
```

View commit history in a simplified format:

```
git log --oneline
```

## 9. Undoing Changes

Unstage a file while retaining its changes:

```
git reset <file>
```

Revert changes in the working directory to the last commit:

```
git checkout -- <file>
```

Undo the last commit but keep the changes:

```
git reset --soft HEAD~1
```

## 10. Tagging Releases

Create a new tag for a specific commit:

```
git tag <tag-name>
```

Push a tag to the remote repository:

```
git push origin <tag-name>
```