



Essential Git Commands

git init

Initialize a new Git repository in your current directory.

```
git init
```

git clone

Clone an existing repository from GitHub or another remote location.

```
git clone <repository_url>
```

git status

Check the status of your working directory and staging area.

```
git status
```

git add

Add changes in the working directory to the staging area.y code

```
git add <file_name>
```

Add all changes:

```
git add .
```

git commit

Commit the staged changes to the repository with a descriptive message.

```
git commit -m "Your commit message"
```

git push

Push your changes to a remote repository, such as GitHub.

```
git push origin <branch_name>
```

git pull

Fetch and merge changes from a remote repository to your local repository.

YouTube: www.youtube.com/@deasadiqbal

GitHub: github.com/deasadiqbal

Medium: <https://medium.com/@deasadiqbal>



```
git pull
```

git fetch

Retrieve the latest changes from a remote repository without merging.

```
git fetch
```

git branch

List all branches or create a new branch.

```
git branch
```

Create a new branch:

```
git branch <new_branch_name>
```

git checkout

Switch to a different branch or restore working tree files.

```
git checkout <branch_name>
```

Create and switch to a new branch:

```
git checkout -b <new_branch_name>
```

git merge

Merge changes from one branch into your current branch.

```
git merge <branch_name>
```

git log

View the commit history.

```
git log
```

git diff

Show changes between commits, branches, or your working directory and the staging area.

```
git diff
```

git reset

YouTube: www.youtube.com/@deasadiqbal

GitHub: github.com/deasadiqbal

Medium: <https://medium.com/@deasadiqbal>



Undo changes by resetting the staging area and/or the working directory.

```
git reset <file_name>
```

Soft reset:

```
git reset --soft <commit_hash>
```

Hard reset:

```
git reset --hard <commit_hash>
```

git rm

Remove files from the working directory and the staging area.

```
git rm <file_name>
```

git stash

Stash changes in a dirty working directory to clean up your working directory.

```
git stash
```

git stash pop

Apply the most recently stashed changes and remove them from the stash list.

```
git stash pop
```

git remote

Manage remote repository connections.

```
git remote -v
```

Add a remote repository:

```
git remote add origin <repository_url>
```

git tag

Create, list, delete, or verify tags.

```
git tag
```

Create a new tag:

YouTube: www.youtube.com/@deasadiqbal

GitHub: github.com/deasadiqbal

Medium: <https://medium.com/@deasadiqbal>



```
git tag -a v1.0 -m "Version 1.0"
```

git show

Show various types of objects (e.g., commits, tags).

```
git show <object>
```

git rebase

Reapply commits on top of another base tip.

```
git rebase <branch_name>
```

YouTube: www.youtube.com/@deasadiqbal

GitHub: github.com/deasadiqbal

Medium: <https://medium.com/@deasadiqbal>