Here's a list of the **important Git commands** categorized by functionality, along with a brief description for each:

Setup and Configuration

`git config`
Configure user details and settings.

Set global username:
```
git config --global user.name "Your Name"
```

Set global email:
```
git config --global user.email "youremail@example.com"
```

View configuration:
```
git config --list
```

Repository Initialization

Initialize a new Git repository:
`git init`

Clone an existing repository:
```
git clone <repository_url>
```

Staging and Committing Changes

Add changes to the staging area:

Add a specific file:
```
git add <file_name>
```

Add all files:
```
git add .
```

Commit staged changes:

With a message:
```
git commit -m "Commit message"
```

Amend the last commit:
```
git commit --amend
```

Branching and Merging

Manage branches:

List all branches:
```
git branch
```

Create a new branch:
```
git branch <branch_name>
```

Delete a branch:
```
git branch -d <branch_name>
```

Switch branches or restore files:

Switch to a branch:
```
git checkout <branch_name>
```

Create and switch to a new branch:
```
git checkout -b <branch_name>
```

Merge branches:
```
git merge <branch_name>
```

Viewing Changes and History

View the status of the working directory:
```
git status
```

Show differences between files:

Compare working directory with the staging area:
```
git diff
```

Compare staged changes with the last commit:
```
git diff --staged
```

View commit history:

Default log:
```
git log
```

Compact log with one line per commit:
```
git log --oneline
```

Remote Repositories

Manage remote repositories:

Add a remote repository:
```
git remote add origin <repository_url>
```

View remote repositories:
```
git remote -v
```

Push changes to a remote repository:
```
git push origin <branch_name>
```

Fetch and merge changes from a remote repository:
```
git pull origin <branch_name>
```

Fetch changes from a remote repository without merging:
```
git fetch origin
```

Undoing Changes

Undo changes in the staging area or commit history:

Unstage a file:
```
git reset <file_name>
```

Reset to a previous commit:
```
git reset --hard <commit_hash>
```

`git restore`
Discard changes in the working directory:
`git restore <file_name>`

`git revert`
Create a new commit that undoes changes from a previous commit:
`git revert <commit_hash>`

Stashing

Save changes temporarily:

Stash changes:
`git stash`

Apply stashed changes:
`git stash apply`

View stashed changes:
`git stash list`

Essential Tips for Beginners

`git help`: Get help for any command: