

Git Commands Cheat Sheet

1. Git Setup

- Set up Git user:

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

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

- Check your current Git configuration:

```
git config --list
```

- Set your default text editor for Git:

```
git config --global core.editor "nano" # Replace with your editor (e.g., vim, code)
```

2. Creating and Cloning Repositories

- Initialize a new Git repository:

```
git init
```

- Clone an existing repository:

```
git clone <repository_url>
```

3. Working with Files

- Check the status of files in the repository:

```
git status
```

- Stage files for commit:

```
git add <file_name> (specific file) or git add . (all changes)
```

- Commit changes with a message:

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

- View changes before committing:

git diff

- Remove a file from staging (unstage it):

git reset <file_name>

- Remove a file from the repository and the working directory:

git rm <file_name>

4. Branching and Merging

- Create a new branch:

git branch <branch_name>

- Switch to a different branch:

git checkout <branch_name>

- Create and switch to a new branch in one command:

git checkout -b <branch_name>

- Merge another branch into the current branch:

git merge <branch_name>

- List all branches:

git branch

- Delete a branch:

git branch -d <branch_name>

5. Remote Repositories

- View remote repositories:

git remote -v

- Add a remote repository:

git remote add origin <repository_url>

- Push changes to a remote repository:

`git push origin <branch_name>`

- Pull changes from a remote repository:

`git pull origin <branch_name>`

- Fetch changes from a remote repository (without merging):

`git fetch origin`

- Remove a remote repository:

`git remote remove origin`

6. Viewing History

- View commit history:

`git log`

- View a summarized commit history:

`git log --oneline`

- View a specific commit:

`git show <commit_hash>`

7. Undoing Changes

- Undo the last commit (keep changes in the working directory):

`git reset --soft HEAD~1`

- Undo the last commit (discard changes in the working directory):

`git reset --hard HEAD~1`

- Unstage a file (undo git add):

`git reset <file_name>`

- Discard changes in a file (revert to the last committed version):

`git checkout -- <file_name>`

- Revert a commit (create a new commit that undoes a previous commit):

`git revert <commit_hash>`

8. Advanced Commands

- Stash changes (temporarily save changes):

`git stash`

- Apply the last stashed changes:

`git stash apply`

- Delete a stash:

`git stash drop`

- Show the current Git configuration:

`git config --list`

- Check which files are being tracked:

`git ls-files`