**# 📝 Git Cheat Sheet**

A beginner-friendly guide to essential Git commands.

---

**## 📦 Project Setup**

**### `git init`**

Initializes a new Git repository in your current directory. It creates a hidden `.git/` folder to start tracking your project. Run this command only once at the beginning of a project.

```bash

git init

📂 File Status & Changes

git status

Displays the state of the working directory and staging area. It shows you which files are modified, staged for the next commit, or untracked.

```bash

git status

git add

Stages changes, preparing them for the next commit. You can add specific files or all changes.

```bash

# Add a specific file

git add <filename>

# Add all changes in the current directory

git add .

git commit -m "message"

Records the staged changes to the repository's history. Each commit is a snapshot of your project at a specific point in time and should have a meaningful message describing the change.

```bash

git commit -m "Your meaningful commit message"

🌱 Branching

git branch

Lists all local branches in the current repository. The active branch will be marked with an asterisk (\*).

```bash

git branch

git branch <branchname>

Creates a new branch. This doesn't switch you to the new branch.

Bash

git branch feature-x

git checkout <branchname>

Switches your working directory to the specified branch.

Bash

git checkout feature-x

🔁 Merging & Restoring

git merge <branchname>

Merges the history of the specified branch into your current branch. Make sure you are on the branch you want to merge into before running this command.

Bash

git merge feature-x

git restore <filename>

Restores a file to the state it was in at the last commit, discarding any changes you've made in your working directory. This does not affect your commit history.

Bash

git restore index.html

🔙 Resetting Changes

git reset <filename>

Unstages a file from the staging area but keeps the modifications in your working directory. This is useful if you accidentally staged a file with git add.

Bash

git reset index.html

🔗 Remote Repositories

git remote -v

Lists the remote repositories connected to your local project, showing their URLs for fetching and pushing.

Bash

git remote -v

🕒 Viewing History

git log

Shows the commit history for the current branch. Press q to exit the log view.

Bash

git log

✅ Quick Reference Table

Command Description

git init    Initialize a new repo

git status  Check the status of the working directory

git add <file>  Stage changes for commit

git commit -m "msg" Commit staged changes with a message

git branch  List all branches

git branch <name>   Create a new branch

git checkout <name> Switch to a different branch

git merge <name>    Merge a branch into the current branch

git restore <file>  Restore a file to its last committed state

git reset <file>    Unstage a file

git remote -v   Show configured remote repositories

git log View the commit history

Export to Sheets

✍️ Tips

Always run git status before adding or committing to see exactly what you're changing.

Write clear and meaningful commit messages.

Use branches to separate work on different features or bug fixes to keep your main branch clean.