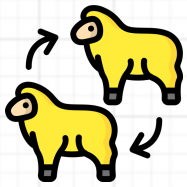
GIT CHEAT SHEET

Git commands in simple words



Clone

This is like downloading a whole folder of files from the internet. You do this when you want to work on a project that someone else has already started. Use: `git clone <repository_url>`

Local usage

Add

Think of it as picking out the files you want to include in your next update. You're getting them ready to be saved.
Use:

- `git add .` to stage all files.
- `git add <file-name OR folder-name>` to stage a specific file or files in a specific folder.

Commit

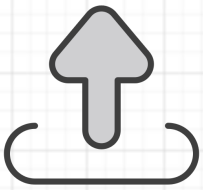
Imagine taking a snapshot of your files at a specific point. You do this when you're done with some changes and want to remember them.
Use: `git commit -m "Your commit message"`

Think of `git add` as preparing the changes you want to save, and `git commit` as actually saving those changes with a descriptive note. This separation allows you to review and organize your changes before permanently recording them in your project's history.

Remote usage

Pull

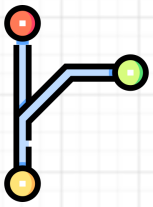
It's like checking if there are any new changes in the project. If others have added things since you last checked, you "pull" those changes into your version. Use: `git pull`



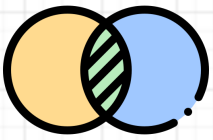
Push

After you've made changes and committed them, you "push" those changes up to a shared place so that others can see and use them. Use: `git push`

Branch



This is like creating a separate copy of your project to experiment with new ideas. You can work on changes without affecting the main version until you're sure they're good. Use: `git branch <branch_name>`



Merge

When you're done experimenting and want to add your changes back to the main project, you "merge" them. It's like putting together puzzle pieces. Use: `git merge <branch_name>`

When using the `git merge <branch_name>` command, you typically need to specify the branch that you want to merge into your current branch. This is important because Git needs to know which branch should receive the changes from the specified branch.

For example, if you're on your main development branch (let's call it **`main`**), and you want to merge changes from a feature branch (let's call it **`feature-branch`**) into the **`main`** branch, you would do:

- `git checkout main` `# Switch to the main branch`
- `git merge feature-branch` `# Merge changes from feature-branch into main`

Status

Think of it as asking Git if there's anything you've changed that you haven't told it about. It gives you a list of files you need to "add" or "commit". Use: `git status`

Log

It's like reading a history book of all the changes made to your project. The "log" command shows you a timeline of what happened. Use: `git log`

Created by Svetlana Lesyukova