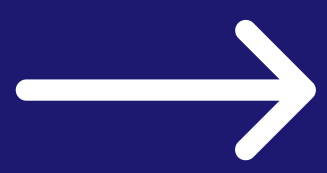
Top 100 Git Commands You Must Know

By Rupendra Ragala

Let’s Swipe



Basic Setup and Configuration

- git --version: Check the installed version of Git.
- git config --global user.name "Your Name": Set your global username for commits.
- git config --global user.email "youremail@example.com": Set your global email for commits.
- git config --global core.editor "editor": Set the default text editor for Git.
- git config --list: List all the current Git configurations.
- git init: Initialize a new Git repository.
- git clone [URL]: Clone a remote repository to your local machine.
- git remote -v: Show the URLs of remote repositories.
- git remote add [name] [URL]: Add a new remote repository.
- git status: Show the status of your working directory and staging area.

Let’s Swipe

Working with Files

- git add [file]: Stage a specific file.
- git add .: Stage all changes in the current directory.
- git commit -m "message": Commit changes with a descriptive message.
- git commit --amend: Modify the most recent commit.
- git diff: View unstaged changes.
- git diff --staged: View staged changes.
- git restore [file]: Discard changes in the working directory.
- git checkout [branch/file]: Switch branches or restore files.
- git rm [file]: Remove a file from the working directory and staging area.
- git mv [old_file] [new_file]: Rename or move a file.

Let’s Swipe

Branching and Merging

- git branch: List all branches.
- git branch [branch_name]: Create a new branch.
- git checkout [branch_name]: Switch to a different branch.
- git checkout -b [branch_name]: Create and switch to a new branch.
- git merge [branch_name]: Merge a branch into the current branch.
- git branch -d [branch_name]: Delete a branch.
- git branch -D [branch_name]: Force delete a branch.
- git log --oneline --graph: Visualize branch history.
- git stash: Save uncommitted changes for later use.
- git stash apply: Reapply the most recent stashed changes.

Let’s Swipe

Collaboration and Synchronization

- git pull: Fetch and merge changes from a remote repository.
- git fetch: Fetch changes from a remote repository without merging.
- git push: Push changes to a remote repository.
- git push -u origin [branch_name]: Push a branch and set it as upstream.
- git rebase [branch_name]: Reapply commits on top of another base tip.
- git pull --rebase: Fetch changes and reapply your changes on top of them.
- git push --force: Forcefully push changes, overwriting conflicts.
- git remote remove [name]: Remove a remote repository.
- git tag [tag_name]: Create a lightweight tag for a specific commit.
- git push origin --tags: Push all tags to the remote repository.

Let’s Swipe

Advanced Usage

- git cherry-pick [commit_hash]: Apply a specific commit from another branch.
- git revert [commit_hash]: Create a new commit that undoes changes from a specific commit.
- git reset [file]: Unstage a file while keeping its changes.
- git reset --soft [commit_hash]: Reset to a commit, keeping changes staged.
- git reset --hard [commit_hash]: Reset to a commit, discarding all changes.
- git log: Show the commit history.
- git blame [file]: Show the commit history line by line for a file.
- git reflog: Show a history of all Git actions.
- git clean -f: Remove untracked files from the working directory.
- git archive --format=zip --output=[file.zip] [branch_name]: Create a zip archive of a branch.

Let’s Swipe

Working with Logs and History

- git show [commit_hash]: Show details of a specific commit.
- git log -p: Show commit history with the changes made in each commit.
- git log --stat: Show commit history with statistics on files changed.
- git log --author="[name]": Show commits by a specific author.
- git log --since="2 weeks ago": Show commits made in a specific time range.
- git log --grep="keyword": Search commits by message.
- git shortlog: Summarize the commit history by author.
- git bisect: Find a commit that introduced a bug using binary search.
- git diff [branch1]...[branch2]: Compare changes between two branches.
- git whatchanged: View changes history like git log but more focused on files.

Let’s Swipe

Stashing Advanced Commands

- git stash list: View the list of stashed changes.
- git stash pop: Reapply and remove the most recent stash.
- git stash drop: Delete a specific stash.
- git stash save "message": Save changes with a descriptive message.
- git stash branch [branch_name]: Create a new branch with the stashed changes.
- git stash clear: Clear all stashes.
- git stash show -p: View the changes in the latest stash.
- git stash apply [stash_id]: Apply a specific stash.
- git stash create: Create a stash without storing it in the stash list.
- git stash store: Store a stash created with git stash create.

Let’s Swipe

Working with Remotes

- git remote show [name]: Display detailed information about a remote.
- git push origin :[branch_name]: Delete a remote branch.
- git push --all: Push all branches to the remote repository.
- git fetch --prune: Remove references to deleted remote branches.
- git remote rename [old_name] [new_name]: Rename a remote.
- git remote set-url [name] [new_URL]: Change the URL of a remote repository.
- git pull –prune: Fetch and remove references to deleted remote branches.
- git push origin --delete [tag_name]: Delete a remote tag.
- git fetch origin [branch_name]: Fetch a specific branch.
- git branch --set-upstream-to=origin/[branch_name]: Link a local branch to a remote branch.

Let’s Swipe

Ignoring and Cleaning Files

- git check-ignore -v [file]: Check if a file is ignored and why.
- git rm --cached [file]: Remove a file from tracking but keep it locally.
- git clean -fd: Remove untracked files and directories.
- git clean -n: Show a preview of untracked files and directories to be removed.
- git clean -x: Remove untracked and ignored files.
- git add --patch: Interactively stage parts of a file.
- git update-index --assume-unchanged [file]: Ignore local changes to a tracked file.
- git update-index --no-assume-unchanged [file]: Stop ignoring local changes.
- git ls-files --ignored: List all ignored files.
- git rm -r --cached .: Remove all files from the index (useful for updating .gitignore).

Let’s Swipe

Advanced Git Features

- git rebase -i: Interactively rebase and edit commits.
- git cherry [branch1] [branch2]: List commits in one branch not in another.
- git subtree add --prefix=[dir] [repo] [branch]: Add a subdirectory from another repo.
- git subtree push --prefix=[dir] [repo] [branch]: Push changes in a subdirectory to another repo.
- git filter-branch: Rewrite commit history (deprecated in favor of git filter-repo).
- git worktree add [dir] [branch]: Work on multiple branches simultaneously.
- git submodule add [URL]: Add a submodule to the repository.
- git submodule update --init --recursive: Initialize and update all submodules.
- git notes add -m "message": Add notes to a commit.
- git blame -L [start,end] [file]: Blame only specific lines in a file.

Let’s Swipe

Thank You

Save