Getting Started

Merge a remote branch into your current branch to bring

\$ git merge [alias]/[branch]
No fast-forward

Only fast-forward

\$ git merge --no-ff [alias]/[branch]

Transmit local branch commits to the remote repository

it up to date

branch

Create a new local repository Show modified files in working directory, staged for your Set the name that will be attached to your commits and \$ git init [project name] \$ git status \$ git config --global user.name "name" Stages the file, ready for commit Set an email address that will be attached to your Clone a repository commits and tags \$ git add [file] \$ git clone git_url \$ git config --global user.email "email" Stage all changed files, ready for commit Enable some colorization of Git output Clone a repository into a specified directory \$ git add . \$ git config --global color.ui auto \$ git clone git url my directory Commit all staged files to versioned history Edit the global configuration file in a text editor \$ git commit -m "commit message" \$ git config --global --edit Commit all your tracked files to versioned history \$ git commit -am "commit message" List all local branches Show the commit history for the currently active branch Discard changes in working directory which is not staged \$ git branch \$ git log \$ git restore [file] List all branches, local and remote Unstage a stagged file or file which is staged Show the commits on branchA that are not on branchB \$ git branch -av \$ git restore --staged [file] \$ git log branchB..branchA Switch to my_branch, and update working directory Unstages file, keeping the file changes Show the commits that changed file, even across renames \$ git checkout my_branch \$ git reset [file] Create a new branch called new branch Revert everything to the last commit \$ git log --follow [file] \$ git checkout -b new branch \$ git reset --hard Show the diff of what is in branchA that is not in branchB Delete the branch called my_branch Diff of what is changed but not staged \$ git diff branchB...branchA \$ git branch -d my branch Show any object in Git in human-readable format Merge branchA into branchB Diff of what is staged but not yet committed \$ git show [SHA] \$ git diff --staged \$ git checkout branchB \$ git merge branchA Apply any commits of current branch ahead of specified Tag the current commit \$ git tag my tag \$ git rebase [branch] Fetch down all the branches from that Git remote Add a git URL as an alias Save modified and staged changes \$ git fetch [alias] \$ git remote add [alias] [url] \$ git stash

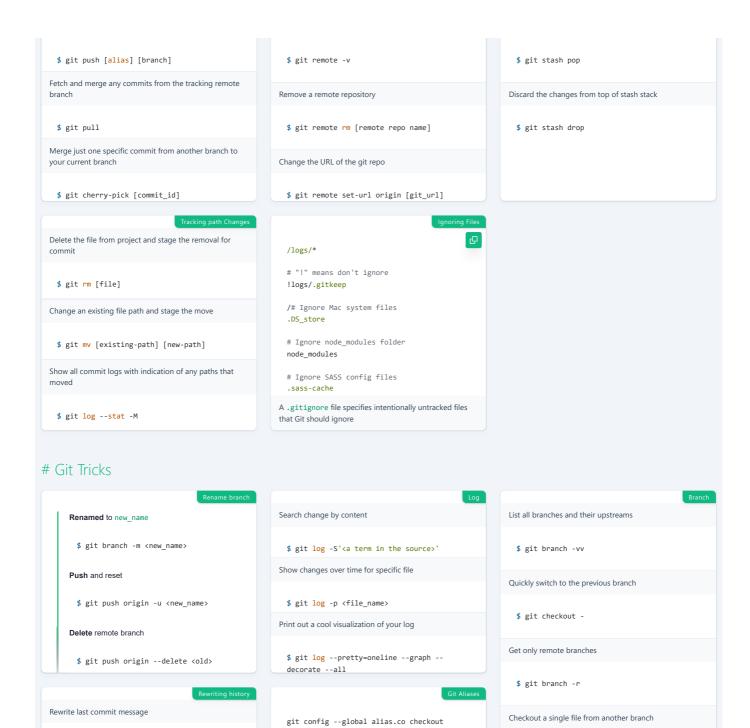
Show the names of the remote repositories you've set up

Show the names and URLs of the remote repositories

List stack-order of stashed file changes

Write working from top of stash stack

\$ git stash list



git config --global alias.br branch

git config --global alias.ci commit

git config --global alias.st status

See also: More Aliases

\$ git checkout <branch> -- <file>

\$ git commit --amend -m "new message"

See also: Rewriting history