

Git Cheat Sheet

15

Minutes

WHAT IS GIT?

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

INSTALLATION

Git for All Platforms:
<http://git-scm.com>

Debian/Ubuntu:
In terminal type,
sudo apt-get install git

Commands

15

Minutes

CONFIGURATION/ SETUP

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

Set the name that will be visible with your commits & tags

git config --global user.email "you@example.com"

Set the email that will be visible with your commits & tags

git config --global color.ui auto

Set color for git output/ command line

START A PROJECT

git init <directory>

Create empty git repository in the specified directory. Run with no arguments to initiate the current directory as a git repository.

git clone <url>

Set the email that will be visible with your commits & tags

GIT PULL

git pull --rebase <remote>

Fetch the remote's copy of current branch and rebases it into the local copy. Uses git rebase instead of merge to integrate the branches.

GIT BASICS

git add <file>

Add the file to the staging area for next commit. Use . in place of file to add all changed files from the current directory to the staging area.

git status

List the staged, unstaged & untracked files

git log

List current branch's commit history

git branch

List your branches. a* will appear next to currently active branch.

git checkout -b <branch>

Create a new branch named "branch". -b flag is used to checkout the existing branch & go to new branch.

DAY TO DAY USE

git merge <branch>

Merge branch named "branch" into current branch

git rm <file>

Remove file from working directory and staging area

git diff

Show changes between working directory and staging area

git add

Add the file to the staging area for next commit. Use . in place of file to add all changed files from the current directory to the staging area.

git checkout -d <branch>

Remove selected branch, if it is already merged into any other. -D instead of -d forces deletion.

GIT DIFF

git diff HEAD	Show difference between working directory and last commit.
git diff --cached	Show difference between staged changes and last commit.

UNDOING CHANGES

git revert <commit>	Create new commit that undoes all of the changes made in , then apply it to the current branch.
git clean -n	Shows which files would be removed from working directory. Use the -f flag in place of the -n flag to execute the clean.
git commit --amend	Replace the last commit with the staged changes and last commit combined. Use with nothing staged to edit the last commit's message.

TEMPORARY COMMITS

git stash	Put current changes in your working directory into stash for later use.
git stash pop	Apply stored stash content into working directory, and clear stash.
git stash drop	Delete a specific stash from all your previous stashes.

GIT REBASE

git rebase -i <base>	Interactively rebase current branch onto . Launches editor to enter commands for how each commit will be transferred to the new base.
-----------------------------------	---

GIT PUSH

git push <remote> --force	Forces the git push even if it results in a non-fast-forward merge. Do not use the --force flag unless you're absolutely sure you know what you're doing
git push <remote> --all	Push all of your local branches to the specified remote.
git push <remote> --tags	Tags aren't automatically pushed when you push a branch or use the --all flag. The --tags flag sends all of your local tags to the remote repo

REMOTE REPOSITORIES

git remote add <name> <url>	Create a new connection to a remote repo
git fetch <remote> <branch>	Fetch specific branch from remote repo. Leave off branch to fetch all remote refs
git fetch --prune <remote>	Delete remote refs that were removed from the remote repo
git pull <remote>	Fetch the copy of remote repo and merge it into local copy
git push <remote> <branch>	Push the branch to <remote>, create a branch named "branch" if it doesn't exist in remote repo
git push -u <remote> <branch>	Push local branch to remote repo & sets its copy as an upstream

REWRITE HISTORY

git reflog	Show a log of changes to the local repo's HEAD
git rebase <base>	Rebase the current branch onto <base>. Base can be a commit ID, branch name, tag or a relative reference to HEAD
git reset <file>	Remove from the staging area, but leave the working directory unchanged. This unstages a file without overwriting any changes.
git reset --hard <commit>	Clear staging area, rewrite working tree from specified commit.

