



Git Cheat Sheet

A distributed version control system.

Download: <https://git-scm.com/>

Documentation: <https://git-scm.com/docs>

YouTube Tutorials Playlist: <https://bit.ly/git-playlist>



General

```
// show docs for git command
man git-clone
// clone repo
git clone URL
// show git status
git status
// show differences
git diff BRANCH, COMMIT or FILE
// diff between working directory and last commit
git diff HEAD
git log -- file.txt // only show commits with file.txt
git log -p -- file.txt // print full diff of each commit
```

Config

```
// list all git configs
git config --list or -l
// list only systes/global/local
git config --global -l
// add name to global config
git config --global user.name NAME
// edit global config
git config --global --edit or -e
// remove record from git config
git config --global --unset name
```

Commit

```
// stage changes for commit
git add --all or -a or .
// create commit with message
git commit -m "MESSAGE"
// combine the above in one command
git commit -a -m "MESSAGE"
// amend new changes to last commit
git commit --amend
```

Fetch, pull and push changes

```
git fetch // fetch from remote repo
git pull // fetch and merge changes to local repo
git push // push local changes to remote repo
```

Branches

```
// list all branches
git branch
// create new branch
git branch NAME
// checkout branch
git checkout NAME
// create and checkout
git checkout -b NAME
// switch to previous branch
git checkout -
```

Managing Branches

```
// show merged or unmerged branches
git branches --no-merged
git branches --merged
// rename branch
git branch -m OLD-NAME NEW-NAME
// push local 'new-branch' to remote repo
git push remote new-branch
// delete local branch
git branch -d new-branch
// delete remote branch
git push origin --delete remote-branch
```

Undoing changes

```
// unstage one file
git reset readme.me
// unstage all changes
git reset HEAD
// unstage all changes are discards them!
git reset --hard HEAD
// reset to a specific commit and keep changes
git reset cb5f4f0
// same as above + discard changes, commits
git reset --hard cb5f4f0
```

Diff

```
// use Diff3 for merge conflicts
git config merge.conflictstyle diff3
```

Reset file to specific commit

```
// reset file to a version from a specific commit
git checkout cb5f4f0 file.txt
// reset file using the interactive mode
git checkout cb5f4f0 file.txt -p
// revert commit by creating new commit
git revert cb5f4f0
```

Merge

```
// merge branch-name to current branch
git merge branch-name
```

Rebase

```
// rebase current branch
// all commits in current branch will be
// replayed on top of changes from new base
// and new commits will be created
git rebase branch-name
// rebase in interactive mode
git rebase -i branch-name
```

Squash commits

```
// squash last 2 commits
git rebase -i HEAD~2
// you will get this:
pick a6817d7 AA
pick 3db9646 BB
// update to this:
pick a6817d7 AA
squash 3db9646 BB
```

Stash changes

```
git stash // put current changes in new stash
git stash save -u // include untracked files
git stash list // show list of stashed changes
git stash apply stash@{0} // apply changes
git stash drop stash@{0} // drop these changes
git stash pop // apply and drop last stash entry
// create new branch from choosen stash
git stash branch new-branch stash@{0}
git stash clear // remove all stash entries
```

Advanced commands

```
// bring changes from commit to my branch
git cherry-pick a6817d7
// bring changes without committing
git cherry-pick -n a6817d7
git blame file.txt // who has changed this file?
git blame -L 3,10 file.txt // from line 3 to10?
git blame --since=1.week file.txt // last week
```

Git bisect - find bad commit

```
git bisect start // start bisect process
git bisect bad // mark commit as bad
git bisect good // mark commit as good
git bisect reset // complete bisect process
```

Alias

```
// create alias
git config --global alias.last 'log -1 HEAD'
// use above alias
git last
```

Useful aliases

```
// git a adds all and shows status
alias.a=!git add . && git status
// git au adds untracked and show status
alias.au=!git add -u . && git status
// git acm = add and commit with message
alias.acm !git add . && git commit -m
```

Advanced Aliases

```
// git clean-branches removes unused branches
alias.clean-branches "!git branch | grep -v
master | xargs git branch -D"
// git alias shows all aliases in all configs
alias.alias=!git config --list | grep 'alias\' | sed
's/alias\.\([^\=]*\)=\(.*/\1\ => \2/' | sort
```

Tools and resources

YouTube tutorials - <https://bit.ly/git-playlist>
 Visualizing Git Concepts with D3:
<http://onlywei.github.io/explain-git-with-d3/>

