

Basic Setup

configure settings

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

git config --global user.email "my-name@school.edu"

git config --global color.ui "auto"

git config --list see all settings

initialize

git init initialize repository in current directory

ignore

touch .gitignore setup txt file with list of files and directories to ignore (one per line)

*.pdf ignore pdf files everywhere

model/* ignore everything in models directory

!models/scripts don't ignore the scripts directory underneath models

git rm --cached file1 file1 was tracked at one point and keeps showing up in git status - this removes it from tracking (as long as file1 is also in .gitignore)

History

log

git log show all commits

q escape long readout and return to command line

git log --oneline returns just one line per commit (commit hash + message)

git log -n view n most recent commits

git log file show file commits

git log --before date date specified as yyy-mm-dd or 2.days.ago

git log --after date -- before date specify date range

git log -p view entire diff of changes for each commit found

git log --stat summary of changes in each commit (# lines added, removed, etc.)

History (cont)

checkout

git checkout abcde file checkout specific version of file

Local changes

special bash

git mv oldfile newfile preserves git history when renaming. git also figures it out if git add newfile and git rm oldfile, but that takes longer.

status

git status shows all files that have been changed since last commit

compare

git diff file1 compare local file1 to last commit of file1

whoops - undo!

git checkout -- file1 go back to previous committed version, toss all changes out the window

add

git add file1 file2 file3 stage files for commit

whoops - undo!

git rm -- cached file1 unstage file1

git rm -r -- cached . unstage all files recursively

commit

git commit -m "My message" commit staged files with a message

git commit -a -m "My message" stages all modified or removed files first (combines add and commit)

whoops - undo!

git commit -- amend edit the last message (if file has changed, changes will also be added to last commit)

git commit -- amend -m "Better message" ditto

git reset HEAD~1 undo entire commit, changes are still there but are now unstaged



Local changes (cont)

`git reset --hard HEAD~1` undo entire commit, revert to previous version
(changes all gone)

`git reset --hard abcde` return to specified version

`git revert HEAD~2` go back to earlier version by making new commit
(rather than undoing commits)

`git rm` totally deletes the file (+ git tracking). To only remove file from
git tracking, `git rm --cached`

Forks and Branches

Fork

`fork` button On the repository you want to contribute to

C

By **[deleted]**
cheatography.com/deleted-30394/

Not published yet.
Last updated 2nd December, 2016.
Page 2 of 2.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>