

PANIC Commands

<code>git reset --soft HEAD^</code>	Undo last commit, put changes into staging
<code>git commit --amend -m "New Message"</code>	Change the last commit
<code>git reset --hard HEAD^</code>	Undo last commit and all changes
<code>git reset --hard HEAD^^</code>	Undo last 2 commits and all changes

Deleting and untracking files

Delete from file system and git	<code>git rm FILE.txt</code>
Delete from git ONLY	<code>git rm --cached FILE.txt</code>

After that, you may commit to save the delete change.

Better ways to show git log

Colorizing Log	<code>git config --global color.ui true</code>
One commit per line	<code>git log --pretty=oneline</code>
Output log the way you want	<code>git log --pretty=format:"%h %ad- [%an] %s"</code>
Show log and modified lines	<code>git log --oneline -p</code>
Show log and insertions and deletes	<code>git log --oneline --stat</code>
Show log with a tree visual representation	<code>git log --oneline --graph</code>
Limit git log until some date	<code>git log --until=1.minute.ago</code>
Limit git log since some date	<code>git log --since=1.day.ago</code>
Range git log	<code>git log --since=1.month.ago --until=2.weeks.ago</code>
Range git log with specific date	<code>git log --since=2000-01-01 --until=2012-12-21</code>

Custom git log output

<code>%ad</code>	author date
<code>%an</code>	author name
<code>%h</code>	SHA hash
<code>%s</code>	subject
<code>%d</code>	ref names

Ex: `git log --pretty=format:"%h %ad- [%an] %s"`

Pushing Faster

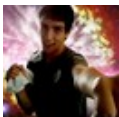
First time	<code>git push -u origin master</code>
Then	<code>git push</code>

Git will memorize the location you want to push by first using the command `-u`

Git Aliases

Git lol (log with some colors and visual tree)	<code>git config --global alias.lol "log --pretty=oneline --abbrev-commit --graph --decorate"</code>
Custom git output	<code>git config --global alias.mylog "log --pretty=format:'%h [%an] %s' --graph"</code>
Git status via <code>git st</code>	<code>git config --global alias.st status</code>
Git checkout via <code>git co</code>	<code>git config --global alias.co checkout</code>
Git branch via <code>git br</code>	<code>git config --global alias.br branch</code>
Git commit via <code>git ci</code>	<code>git config --global alias.ci commit</code>

Create your own alias using the format:
`git config --global alias.youralias "your command, ommiting 'git'`



By **wdfelippe**
cheatography.com/wdfelippe/

Published 17th March, 2013.
 Last updated 5th June, 2014.
 Page 1 of 1.

Sponsored by **Readability-Score.com**
 Measure your website readability!
<https://readability-score.com>