

Git Cheat Sheet

Universal Commands
Anytime <hash> is used any valid git format for specifying a commit can be used.
--graph #Show with branch graph...useful with git log
-i #Interactive mode...type ? for help. Used with almost all git cmd

git add
--dry-run --ignore-missing #Check if any files would be ignored
--patch #See Universal commands -i

git branch
<branch> <parent tag/branch> #Create new <branch>
-d <branch> #Delete only if merged
-D <branch> #CAUTION!! force delete branch
-m <oldname> <newname> #Rename branch
-v #Show last commit on each branch
-u <remote> /<branch> #Set remote branch as upstream of current

git checkout
-- <file> #Discard all changes..can't be undone
-b <new branch> <parent tag/branch> #Create new <branch>

git cherry-pick
<commit hash 1> <commit hash n> #Pick n commits to current

git clean (all flags are optional)
Careful Now!! Run git stash --all first or have .gitignore set up well
-n -d #dry run of files and empty dirs clean will remove
-f -d -x -i #-f force, -x will remove all the files in your .gitignore

git clone
-o <remote> <url> #Specify remote

git commit
--amend #Amend commit

git describe
<branch> #give nearest tag and no. commits on top

git diff
--check #Show possible whitespace issues
<parent branch>...<child branch> #Show changes child made

git fetch
<remote> -p #Update refs and remove outdated refs

git grep
By default it will search all files in your working dir
--break --heading #Combine all other options with these
-n #Print line numbers of matches
--count #Prints no. matches per file
-p #See function match was found in

git log
-p -cc <-> #Show diff of last <-> commits, -cc for merge commits
<-> #Show logs of last <-> commits
--pretty=format:"%h - %an, %ad : %s" #Custom log format
--since="<time>" #relative eg. 2 weeks or 2015-12-25
--until="<time>" #Same as above
--<author> #limit to commits by author...use --all-match with --grep
-S<string> #search commits that a/m/d <string>
^<parent> <child branch 1> <child branch n> #list changes in <cb1> and <cbn> not in parent...works with single <cb>
--left-right <branch 1>...<branch 2> #Commits from common parent

git merge
All -X options can be used with git pull as well
<branch> #branch to merge into current
--no-ff #creates merge commit
--abort #dive, dive, dive
-Xignore-all-space #ignore whitespace(ws) when merging
-Xignore-space-change #treats sequences of ws as 1ws
-s ours/theirs #Choose version of conflicting files

git mergetool
#Fires up appropriate gui merge tool

git merge-base
<branch 1> <branch 2> #Shows most recent common parent

git push
-u <remote> <branch> #Set remote branch as upstream of current
<remote> <tag> #Push <tag> to <remote>
remote --delete <branch> #delete remote branch

git pull
See git merge for -X options

git rebase
<branch> #Applies changes from current branch onto common parent of specified <branch>
--onto <parent> <child branch 1> <child branch n> #replay commits from all cb's onto parent

git reset
HEAD --<hard/soft> <fileA(optional)> #soft to undo & keep changes
--patch #Partially reset files
HEAD~<-> --<hard/soft> #Go back <-> commits

git remote
-v #All remotes
add <remote> <url>
show <remote>
prune <remote> #Update refs
rename <remote> <new remote name>
rm <remote>

git revert
HEAD~<n> #Revert <n>th commit from head (n=3 is fourth last)
<branch>~<->..<<branch>~<n> #from xth to nth cmt && x > n
--abort #Abandon ship

git shortlog
--no-merges <branch> --not <branch or tag> #commits by author
--merges <branch> --not <branch or tag> #merges by author

git show
<commit hash> #Details of commit
<commit hash>^ #Show parent of commit
<commit hash>^2 #Show other parent of commit...merges
<commit hash>~<n> #Details of commit <n> commits before

git stash
apply --index(optional) <stash>(optional) #apply a stashed object. --index will try to reapply staged changes. If no <stash> is specified then it takes the top stash on the stack
list #list all stashed changes
save --patch(optional) --keep-index(optional) #k-i won't stash stgd
create #create a stash that is a regular commit object

git status
-b #show branch info
-u #Show untracked files
--ignored #List all ignored files

git tag
-a <tag> -m"msg" #Create tag with a message
-a <tag> <commit hash> #Tag a previous commit
-l "pattern" #List all tags with optional pattern match
<tag> #Lightweight tag same as commit

Extra Stuff
for branch in `git branch -r --merged grep -v HEAD` ; do echo -e `git show --format="%ci %cr %an" \$branch head -n 1` \\\t \$branch; done sort -r #All live branches merged into current branch
for branch in `git branch -r --no-merged grep -v HEAD` ; do echo -e `git show --format="%ci %cr %an" \$branch head -n 1` \\\t \$branch; done sort -r #All live branches not merged into current branch

git aliases
git config --global alias.<alias> <stuff to alias> #Create a git alias
git config --get-regexp alias\ ---Shows all git aliases
git last
git log -1 HEAD
git unstage
git reset HEAD
git co
git checkout
git br
git branch
git c
git commit -m
git st
git status
git sba
git show-branch --color=always -a --more=10 --no-name
git sban
git show-branch --color=always -a --more=10
git a
git add
git aa
git add .
git po
git pull origin
git puo
git push origin
git rl
git reflog
git lp
git log --pretty=format:"%h - %an, %ad : %s"
git lpfh
git log --pretty=format:"%H - %an, %ad : %s"
git fo
git fetch origin -p