Git basics Cheat Sheet

by cmok1996 via cheatography.com/136331/cs/28455/

Create Git Repository

From existing directory	cd proj_dir ; git init ; git add . ; git commit -m <message>; git remote add origin <url>; git push -u origin <url>
From other repository	mkdir proj_dir; cd proj_dir; git clone <url>; cd <repo>

Local changes

Changed in working directory	git status
Show tracked file changes between index and HEAD	git diff <file>
Add changed files / Track changes in working tree	git add file1 file2 ...
Stage all files in directory	git add <dir/>
Add all files in current directory to staging area (>2.0)	git add .
Add all files to staging area	git add -A
Add files deleted/modified +not created	git add -u
Add new and modified files only + not deleted	git add --ignore-removal .
See all changes ready for commit	git diff --cached

Local changes (cont)

Restore staged changes but still track in working tree / Undo git add modified file	git restore <file> / git checkout -- <file> / git checkout -- .
Unstage / Untrack / Undo git add new file	git restore --staged <file> / git reset --mixed HEAD <file> / git rm --cached <file>
Track remove file	git rm <file>
Track remove directory (recursively remove files)	git rm <dir/> r
Commit	git commit -m "message"
Change commit message	git commit --amend -m "message"
Remove last n local commit (changes reset in index but still preserved in working tree)	git reset HEAD~n
Soft reset local commits (changes in index are preserved and ready to commit	git reset --soft HEAD~n
Hard reset local commits (changes completely removed in both index and working tree -> mixed reset + restore	git reset --hard HEAD~n

Local changes (cont)

Revert both local and remote commits (Creates a new commit to revert changes, ready to be pushed)	git revert <commit_id>
Reset changes back to specific branch	git reset --hard <remote/branch>
Permanently clear untracked files / Clear working tree	git clean -xdf

History

Show all commits	git log
Simplified logs	git log --pretty=oneline / git log--pretty=reference
Show file commits	git log <file>
Show directory commits	git log <dir/>
Show commit stats changes	git log --stat
Show commits of branch	git log <branch> / git log <remote> <branch>
Who changed file (walkthrough of code history)	git blame <file>

Branching

List branches (-a to include remote branches)	git branch -a

Branching (cont)

Switch to branch (Requires a clean slate to change branch)	git checkout <branch_name>
Create new branch	git branch <new_branch>
Create new branch and checkout to new branch	git checkout -b <new_branch>
Delete local branch	git branch -d <branch_name>
Stash tracked changes	git stash save "message"
Show stash list	git stash list
Apply stash to another branch	git checkout <branch> ; git stash apply <stash_id>
Drop stash	git stash drop <stash_id>
Merge feature_branch into master_branch (master-feature-HEAD)	git checkout master ; git merge feature
Rebase master_branch into feature_branch (master-feature-HEAD)	git checkout feature; git rebase master
Resolving conflicts	git mergetool ; git commit -m "message"
Checkout previous commits (creates a detached head state - no branch attached to it)	git checkout <commit_id> ; git checkout -b <new_branch> (if want to attach head and save commits)

Remote

List remotes	git remote -v
Show tracking information	git remote show <remote>
Add remote	git remote add <remote> <repo_url>
Fetch changes from remote repo	git fetch <remote>
Fetch remote branch changes + Merge into current branch	git pull <remote> <remote_branch>
Pull rebase (-remote-changes-HEAD; to avoid conflicts if local commits are already performed)	git pull --rebase <remote> <branch>
Create local branch to track remote branch	git checkout <remote_branch> / git checkout --track <remote_branch>
Check tracking branches	git branch -vv
Set local branch to track remote branch	git branch -u <remote/branch>
Set upstream branch using push (local branch to track remote branch)	git push -u <remote> <remote_branch>
Push local commits to upstream remote	git push <remote>
Push local commits to remote branch with same name	git push <remote> HEAD

Remote (cont)

Push local commits to remote branch with different remote tracking / Push to someone else's branch	git push <remote> HEAD:<remote_branch>

Others

Documentation	git <command> --help
Ignore files	Edit .gitignore
Set username	git config --global user.name "John Doe"
Set email address	git config --global user.email johndoe@example.com
Set external text editor	git config --global core.editor emacs
Set external merge tool	git config --global merge.tool extMerge
Set external diff tool	git config --global diff.external extDiff