Create Change Update From existing data Using your favorite editor / IDE Fetch latest changes from origin cd ~/my project dir git fetch git init this does not merge them git add . Pull latest changes from origin From existing repo Revert git pull git clone ~/existing/repo ~/new/repo does a fetch followed by a merge git clone you@host.org:dir/project.git Apply a patch that someone sent you Return to the last committed state default protocol is ssh git am -3 patch.mbox git checkout -f | git reset --hard In case of conflict, resolve the conflict and you cannot undo a hard reset git am --resolve Revert the last commit git revert HEAD **Browse** Creates a new commit Revert specific commit Files changed in working directory git revert \$id Commit git status Creates a new commit Changes to tracked files Fix the last commit git diff Commit all local changes git commit -a --amend git commit -a Changes between ID1 and ID2 after editing the broken files git diff <ID1> <ID2> Checkout the ID version of a file History of changes git checkout <ID> <file> git log Who changed what and when in a file **Publish** git blame <file> A commit identified by ID Branch Prepare a patch for other developers git show <ID> git format-patch origin A specific file from a specific ID List all branches Push changes to origin git diff <ID>:<FILE> git branch git push [origin] [branch] Search for patterns Switch to the BRANCH branch Make a version or milestone git grep <pattern> [path] git checkout <BRANCH> git tag <version name> Merge branch B1 into branch B2 git checkout <B2> git merge <B1> Create branch based on HEAD Useful tips git branch <BRANCH> Create branch based on another Get help git branch <new> <base> Configuration git help [command] Delete a branch Create empty branch git branch -d <branch> git symbolic-ref HEAD git config [--global] global is stored in ~/.gitconfig refs/heads/newbranch rm .git/index git clean -fdx Resolve merge conflicts user.name \$name <do work> user.email \$email git add your files color View merge conflicts color.ui auto git commit -m 'Initial commit' git diff **Graphical log** github View merge conflicts against base file github.user \$user git log --graph git diff --base <FILE> git log --graph --pretty=oneline -github.token \$token View merge conflicts against other changes

git diff --theirs <FILE>

git diff --ours <FILE>

git rebase --continue

After resolving conflicts, merge with

git add <CONFLICTING FILE>

View merge conflicts against your changes

git push <origin> <branch>

git push <origin> :<branch>

Delete remote branch and locally

git branch -d <branch>

abbrev-commit

Push branch to remote

do not use on low memory p

optimisation

pack.threads 0

diff.renamelimit 0

core.autocrlf true