|  |  |  |
| --- | --- | --- |
| ADD | git add <file>  git add .  git add -A :/ ili git add :/  git add <dir>  git add -u | # dodaje promjene file-a u staging  # sve file-ove od trenutnog direktorija  # dodaje promjene svih file-ove u staging. (git add -A za novije verzije git-a)  # dodaje sve promjene unutar direktorija.  # dodaje samo tracked file-ove |
| BLAME | git blame <file> | # prikazuje tko i kada i u kojem komitu je promijenio neku liniju u file-u |
| BRANCH | git branch  git branch -r  git branch -a  git branch --list g\*  git branch -m grana1 grana2  git branch <grana> <sha1>  git branch -d <grana>  git branch -a --contains SHA1  git branch -v  git branch -vv  git checkout -b <grana> <sha1> | # lista sve lokalne grane, zvjezdica na checkedout grani ili na deteched HEAD  # lista sve remote grane  # lista sve grane, lokalne i remote grane  # lista sve lokalne grane koje počinju sa g (g\* je reg izraz)  # preimenuje grana1 u grana2 (grana1 može biti checkoutirana)  # stvara grana na SHA1  # briše lokalnu granu (HEAD ne može biti na toj grani)  # lista sve grane koje sadrže komit SHA1  # lista grane, sha1 i naslov commit poruke  # lista grane, sha1, naslov commit poruke i upstream granu.  # stvara novu granu na commitu sa sha1 i checkoutira tu granu |
| CHECKOUT | git checkout <sha1>  git checkout <grana>  git checkout <tag>  git checkout master <file>  git checkout -b <grana> <sha1>  git checkout -B <grana> <sha1>  git merge-base <grana1> <grana2> | # pomiče HEAD na <sha1> (konflikt ako u radnom direktoriju/stage-u imamo promjene koje su u konfliktu sa promjenama u <sha1>)  # pomiče HEAD na <grana>  # pomiče HEAD na <tag>  # file s grane master stavlja u staging  # stvara novu granu na commitu sa sha1 i checkoutira tu granu  # stvara novu granu na commitu sa sha1 i checkoutira tu granu, ako grana postoji onda je pomiče na taj <sha1>  # nalazi sjecište dvije grane |
| CHERRY-PICK | git checkout test && git cherry-pick SHA1 | # na test grani dodaje commit SHA1, komit poruka identična |
| CLEAN | git clean -xfd  git clean -xffd | # briše untracked file-ove (ne briše dodane subrepo-e)  # briše untracked file-ove (briše dodane subrepo-e) |
| CLONE | git clone ssh://gerrit.ericsson.se:29418/platform/example  git clone ssh://gerrit.ericsson.se:29418/platform/example &&  scp -p -P 29418 gerrit.ericsson.se:hooks/commit-msg example/.git/hooks/  git clone --branch master --depth 200 ssh://gerrit.ericsson.se:29418/platform/example  git clone --signle --branch master ssh://gerrit.ericsson.se:29418/platform/example  git clone -b test ssh://gerrit.ericsson.se:29418/platform/example  git clone --recursive ssh://gerritmirror-ha.rnd.ki.sw.ericsson.se:29418/platform/cpp.git | # kloniranje remote repozitorija example  # kloniranje remote repozitorija example + dodavanje hook koja dodaje automatsku Change-Id u komit poruku  # klonira samo 200 zadnjih komitova na grani master  # klonira samo granu master  # klonira sve grane i kreira lokalnu granu test. HEAD je na lokalnoj grani test koja prati remote granu test  # klonira cpp repo i njegove subrepo-e |
| COMMIT | git commit  git commit --allow-empty  git commit <file>  git commit --amend  git commit --amend --no-edit  git cimmit -m “naslov komit poruke“  git commit -F <file>  git commit -C <sha1>  git commit -s | # stvara novi komit  # komita prazni komit  # komita samo jedan file iz indexa  # stvara novi komit na istoj bazi kao što je bio i prethodni komit, npr. želimo izmijeniti postojeću komit poruku.  # bez editiranja commit msg.  # komitanje bez otvaranja prozora za unos komit poruke  # čita commit msg iz file-a  # stvara novi komit, komit poruka je jednaka kao i na komitu sa <sha1>  # dodaje Signed-off-by u commit msg. Moguće da Gerrit ne dopušta pushanje bez Signed-off-by. |
| DIFF | git diff --name-only HEAD~ HEAD  git diff --name-only master test  git diff master test –stat  git diff master test <file>  git difftool -t kdiff3 master test <file>  git diff <file>  git diff <file1> <file2>  git diff HEAD <file>  git diff --cached <file>  git diff <branchA>:<fileA> <branchB>:<fileB>  git diff --diff-filter=M --stat  git diff HEAD --diff-filter=M --stat  git diff --staged --diff-filter=M --stat | # svi fileovi koji su se mijenjali u zadnjem komitu  # svi fileovi koji su se mijenjali između master i test grane  # diff file-a između master i test grane  # diff file-a između master i test grane, grafički prikaz  # diff radni direktorij vs. Stage za file  # diff radni direktorij vs. Stage za file1 i file2  # diff radni direktorij vs. Repozitorij  # stage vs. repozitorij  # diff između fileA na grani A i fileB na grani B  # modified file-ovi u radnom direktoriju (ne vrijedi za untrack file-ove).  # --diff-filter=A za added. --diff-filter=D za deleted.  # modified file-ovi u radnom direktoriju i stage-u (ne vrijedi za untrack file-ove u random direktoriju)  # modified fileovi u stage-u |
| DIFFTOOL | git difftool --ext-diff SHA1 SHA1 FILE  git difftool -t kdiff3 SHA1 SHA1 FILE  git difftool -t meld SHA1 SHA1 FILE | # grafički diff |
| FETCH | git fetch  git tech --all  git fetch -p  git fetch -t  git fetch origin master | # dohvat iz remote (origin) repozitorija  # dohvat iz svih remote repozitorija  # ako su remote grane bile obrisane, briše ih i iz lokal repozitorija  # update tag-ova ako su bili premještani  # dohvat iz remote (origin) repozitorija ali samo promjena sa master grane. |
| GITK | gitk | # browser i diff tool |
| INIT | cd <dir> && git init | # pretvara obični direktorij u git direktorij |
| LOG | git log  git log master  git log <SHA1>  git log master test  git log --all  git log master -10  git log master..test  git log master...test  git log ..test  git log --grep TR12345  git log --author Marko  git log -1 --pretty=format:%an HEAD  git log -1 --pretty=format:%B HEAD  git log -1 --stat  lg1 | # lista history od HEAD  # lista history od master grane  # lista history od SHA1  # lista history od master i test grane  # lista history sa svih grana  # lista history zadnjih 10 commitova sa master grane  # commiti sa test grane, ne sa master grane  # commiti sa master ili test grane, ali ne zajednički  # komiti sa test grane, ali ne i od HEAD  # commiti koji sadrže TR u commit poruci  # commiti od Marka  # autor od komita na koji pokazuje HEAD  # komit poruka od komita na koji pokazuje HEAD  # komit poruka i mijenjani fileovi trenutnog komita  # alias lg1 'git --no-pager log --graph --decorate --pretty=oneline --abbrev-commit' |
| LS-REMOTE | git ls-remote  git ls-remote ssh://gerrit.ericsson.se:29418/platform/example | # lista remote reference (origin)  # lista remote reference od example repozitorija (ne moramo biti u kloniranom repozitoriju example) |
| MAN PAGES | git <naredba> --help | # man-pages |
| MERGE | git checkout master  git merge test  git merge-base <grana1> <grana2> | # promjene sa test grane merga u master granu.  # nalazi sjecište dvije grane |
| MERGETOOL | Git mergetool | # grafički tool za rješavanje konflikata |
| PULL | git pull  git pull -r | # git fetch + git merge. Dohvat iz remote (origin) repozitorija i merge sa remote granom. Moramo biti sa HEAD na lokalnoj grani koja prati remote granu  # git fetch + git rebase. |
| PUSH | git push origin HEAD:refs/for/master  git push origin HEAD:refs/for/master/TOPIC | # pusha promjene sa HEAD u master granu u Gerrit  # pusha promjene sa HEAD u master granu u Gerrit i dodaje TOPIC za taj push (više pusheva može imati isti TOPIC) |
| REFLOG | git reflog show master  (git log -g --abbrev-commit --pretty=oneline master) | # lista što se radilo na master grani |
| REBASE | git checkout test && git rebase master  git rebase -i HEAD~2 | # promjene sa test grane rebase-a na bazi master  # rebase interactive, npr. spojimo dva commit u jedan commit |
| REMOTE | git remote -v  git remote show origin  git remote add <remote name> <path to repo> | # putanja remote repozitorija za fetch i push  # putanja remote repozitorija za fetch i push, koje grane prate remote grane…  # dodaje dodatni remote repozitorij |
| RESET | git reset <SHA1>  git reset --soft <SHA1>  git reset --hard <SHA1> | # stavlja HEAD na SHA1, promjene sa starog SHA-1 vidljive u radnom direktoriju  # stavlja HEAD na SHA1, promjene sa starog SHA-1 vidljive u stagingu  # stavlja HEAD na SHA1, čisti radni direktorij i staging |
| REV-LIST | git rev-list | # lista komitove u lokalnom repozitoriju (git rev-list master --not test) |
| REV-PARSE | git rev-parse HEAD  git rev-parse --abbrev-ref HEAD  git rev-parse --show-toplevel | # daje SHA1 na koji pokazuje HEAD  # daje granu ili HEAD ako nismo na grani  # daje path git top levela |
| REVERT | git revert <SHA1> | # stvara novi commit, revert SHA1 |
| RM | git rm <file> | # briše file iz staginga, promjena spremna za komitanje. |
| SHOW-REF | git show-ref | # lista lokalne reference (git show-ref master) |
| SHOW | git show <TAG>  git show -s --pretty=format: <TAG> | # prikazuje tag poruku, komit poruku, diff  # prikazuje samo tag poruku |
| STASH | git stash  git stash pop | # čisti radni direktorij i staging, promjene u radnom direktoriju i stagingu će biti sačuvane.  # vraća sačuvane promjene |
| STATUS | git status  git status –ignored  git status --porcelain | # prikazuje status promjena  # prikazuje i ignored fileove  # prikazuje A za Added, M za Modified, D za Deleted, ?? za untracked fileove. |
| SUBMODULE | git submodule update | # update-a submodule u skladu sa HEAD u top repo |
| TAG | git tag  git tag --points-at HEAD  git tag <TAG> <SHA1>  git tag -a <TAG> <SHA1>  git tag –d <TAG> | # lista sve tagove  # lista sve tagove na HEAD komitu  # kreira TAG na SHA1  # kreira TAG s porukom na SHA1  # briše TAG |