GITHUB COMMANDS

NAJBOLJ UPORABLJENI GIT UKAZI:

|  |  |  |
| --- | --- | --- |
| UKAZ | RAZLAGA | PRIMER |
| git add | Doda spremembe datotekam v trenutni delovni directory na vaš indeks. | *git add* |
| git commit | Vzame vse spremembe, zapisane v indeksu, ustvari nov commit objekt, ki kaže nanj in določi vejo, ki kaže na novo objavo (commit). | *git commit -m "informacijo o plačilu doda sprememb" git commiting -a -m "zavezoval vse spremembe, enak git add in git commit '* |
| git push | Potisne vse spremenjene lokalne objekte/elemente na oddaljeno repository in nadaljuje svoje podveje (branches). | *git push origin master* |
| git status | Prikaže stanje datotek v verziji indeksa v delovni directory. | *git status* |
| git merge | Združi eno ali več vej v trenutno glavno vejo in avtomatično ustvari nov commit, če ni nobenih konfliktov. | *git merge newbranchversion* |
| git checkout | Odjavi drugačno vejo – izvede switch (izbira) med različnimi vejami projekta z ažurniranjem indeksa, delavnega drevesa in glava (HEAD) da prezrcali izbrano vejo. | *git checkout newbranch* |

OSTALI UPORABLJENI GIT URAZI:

* **git config**  
  Settira konfiguracijske vrednosti za vaše uporabniško ime, e-pošta, gpg key, prednostni diff algoritmov, format datotek in več.

*Primer:*

*git config --global user.name "My Name" git config --global user.email "user@domain.com" cat ~ / .gitconfig [user] name = My Name email = user@domain.com*

* **git init**  
  Inicializira novo git repozitorij - ustvari začetno ".git" mapo/directory v novem ali obstoječem projektu.

*Primer:*

*cd / home / user / my\_new\_git\_folder / git init*

* **git clone**  
  Ustvari git repository kopijo iz oddaljenega vira. Doda tudi izvirno lokacijo kot daljinski upravljalnik, tako da lahko uporabnik odvzame od njega ga spet potisne ven, če ima dovoljenja (permissions).

*Primer:*

*git clone git@github.com: uporabnik / test.git*

* **git rm**  
  Zbriše datoteke iz indeksa in iz vaše trenutne delovni directory, tako da jih ne bo mogoče slediti.

*Primer:*

*git rm ime\_datoteke*

* **git branch**  
  Naredi seznam vseh obstoječih podvej, vključno oddaljene (remote) podveje, če je '-a' na voljo. Ustvari novo podvejo (novo vejo po kateri teče razvoj programa), če je ime podveje na voljo.

*Primer:*

*git branch -a \* master remotes / origin / master*

* **git reset**  
  Resetira tvoj indeks in trenutno delovno directory na stanje vašega zadnjega commit-a. *Primer:*

*git reset --hard HEAD*

* **git stash**  
  Začasno shrani spremembe, ki jih ne želite takoj commit-ati. Pozneje lahko uporabite/aplicirate spremembe.

*Primer:*

*git stash Saved working directory and index state "WIP on master: 84f241e first commit" HEAD is now at 84f241e first commit (To restore them type "git stash apply")*

* **git tag**  
  Tagira en poseben commit s preprostim tekstom (comments). (Dodajanje tag-a enemu commitu).

*Primer:*

*git tag -a v1.0 -m "To je različica 1.0 tag"*

* **git fetch**  
  Pušča vse objekte iz oddaljene repository, ki niso prisotni v tisti lokalni.

*Primer:*

*git fetch origin*

* **git pull**  
  Pušča datoteke iz oddaljene repository in jih združi (merge) z vašo lokalno directory.

Ta ukaz je enak ukazu: git fetch in git merge sequence.

*Primer:*

*git pull origin*

* **git remote**  
  Prikazuje vse oddaljene različice/verzije svoje repository.

*Primer:*

*git remote origin*

* **git log**  
  Prikaže seznam objav (commits) na podveji, vključno z ustreznimi potajli (details). *Primer:*

*git log commit 84f241e8a0d768fb37ff7ad40e294b61a99a0abe Author: User <user@domain.com> Date: Mon May 3 09:24:05 2010 +0300 first commit*

* **git show**  
  Prikazuje informacije o določenem git objektu.

*Primer:*

*git show commit 84f241e8a0d768fb37ff7ad40e294b61a99a0abe Author: User <user@domain.com> Date: Mon May 3 09:24:05 2010 +0300 first commit diff --git a/README b/README new file mode 100644 index 0000000..e69de29*

* **git ls-tree**  
  Prikazuje drevo objektov (object), vključno modaliteto (mode) in ime vsake točke in SHA-1 vrednost medeža (blob) ali drevo, na katerega kaže.

*Primer:*

*git ls-tree master^{tree} 100644 blob e69de29bb2d1d6434b8b29ae775ad8c2e48c5391 README*

* **git cat-file**  
  Uporabljen za ogled/prikaz vrsto določenega objekta (object) skozi vrednost SHA-1.

*Primer:*

*git cat-file -t e69de29bb2d1d6434b8b29ae775ad8c2e48c5391 blob*

* **git grep**Vam omogoča iskanje po vaših drevesnih vsebin za besede in stavke.

*Primer:*

*git grep "www.siteground.com" - \* .php*

* **git diff**  
  Generira popravki datoteke ali statistike razlik med poti ali datotekami v vaši git repository ali v indeksu, ali v vaši trenutni delovni repository.

*Primer:*

*git diff*

* **gitk**  
  Grafični Tcl/Tk, temeljen vmesnik (based interface) za lokalno Git repository.

*Primer:*

*gitk*

* **git archive**

Ustvari en .tar ali .zip file, upoštevajoč usebin posameznega drevesa iz vaše repository.

*Primer:*

*git archive --format=zip master^ README >file.zip*

* **git gc**

Garbage collector za vašo repository. Optimizira vašo repository. Se izvaja naklučno (occasionally).

*Primer:*

*git gc Counting objects: 7, done. Delta compression using up to 2 threads. Compressing objects: 100% (5/5), done. Writing objects: 100% (7/7), done. Total 7 (delta 1), reused 0 (delta 0)*

* **git fsck**

Izvede eno integritetno kontrolo (integrity check) od Git file system, ter identificira korruptirane objekte (corrupted objects).

*Primer:*

*git fsck*

* **git prune**

Zbriše objekte, ki niso dosti uporabljeni oz. pointed od nobenega objekta v nobeni dosegljivi veji (branch).

*Primer:*

*Git prune*