

Konfigurointi, git config

```
git config user.name "Etunimi Sukunimi"
```

Asettaa käyttäjänimen joka näkyy esim. versiohistoriassa muutoksen tekijänä.

```
git config user.email "etunimi.sukunimi@v-erkkoasema.fi"
```

Kuten yllä, mutta sähköpostiosoitteelle.

```
git config --list
```

Näyttää nykyisen konfiguraation.

Asetukset ovat repositorykohtaisia. --global flagilla voit asettaa esim. käyttäjänimen kerralla globaalisti, git config --global user.name "Etunimi Sukunimi"

Aloittaminen

```
git init
```

Luo nykyiseen hakemistoon uuden repositoryn. Käytetään silloin, kun luodaan ensimmäinen repository uudelle projektille.

```
git status
```

Näyttää nykyisen repositoryn tilanteen; esim. nykyisen haaran (branch), onko se ajan tasalla yms.

```
git clone user@domain.xyz:path/to/repo.git
```

Kopioi olemassaolevan repositoryn nykyiseen hakemistoon (SSH:n yli), ja lisää sen automaattisesti remoteksi.

Muutoshistoria

```
git log
```

Näyttää kaikki commitit, alkaen uusimmasta

```
git log -p tiedostonimi.xyz
```

Näyttää ks. tiedoston muutoshistorian

```
git diff
```

Näyttää kaikki commitoimattomat muutokset

```
git diff HEAD ./path/to/file.xyz
```

Näyttää viimeisimmän commitin jälkeen tiedostoon tehdyt muutokset.

```
git blame ./path/to/file.xyz
```

Kuka töhöili ja mitä... ;) Eli näyttää kuka on viimeksi muokannut tiedostoa, ja mitä on muokattu

Muutosten tekeminen

```
git add tiedostonimi.xyz
```

Lisää kyseisen tiedoston commitoitavaksi

```
git add --all
```

Asettaa kaikki muuttuneet tiedostot commitoitavaksi

```
git commit -m "Tiivistelmä muutoksista"
```

Commitoi add-komennolla lisätyt muutokset

```
git commit -am "Tiivistelmä muutoksista"
```

Commitoi automaattisesti kaikki muuttuneet tiedostot

```
git commit --amend -m "Päivitetty tiivistelmä muutoksista"
```

Muuttaa/päivittää edellistä committia

Committia odottavat tiedostot ovat ns. staging arealla.

HUOM! Remoteen **julkaistuja committeja ei saa muokata**, vaan muutokset on tehtävä uusina committeina.

Muutosten kumoaminen

```
git reset --hard HEAD
```

Kumoo kaikki paikalliset muutokset

```
git reset --soft HEAD~1
```

Kumoo edellisen commitin

```
git reset --hard 0ad5a7a6
```

Palauttaa tiedoston annetun commit-tunnisteen mukaiseen tilaan

```
git checkout tiedostonimi.xyz
```

Palauttaa annetun tiedoston edellisen commitin mukaiseen tilaan

--hard flagi tuhoaa kaikki tehdyt muutokset, joten sen kanssa on syytä olla varovainen. --soft flagi säilyttää muutokset, ja ne ovat komennon jälkeen commitoimattomina muutoksina staging arealla

Haarat (branch)

Haarat (branch) (cont)

```
git branch -D haaran_nimi
```

Poistaa haaran

```
git rebase haaran_nimi
```

Hakee masteriin tehdyt muutokset annettuun haaraan.

```
git rebase --continue
```

Jos rebasen aikana tulee ristiriitoja, tämä komento jatkaa rebasea. Suorita vasta kun olet ratkaissut ristiriidat.

Oletushaara on nimeltään master

Muutokset repositoryjen välillä

```
git pull
```

"Vetää" muutokset remote-repositorystä

```
git push
```

"Työntää" muutokset remote-repositoryyn

```
git pull origin haaran_nimi
```

"Vetää" muutokset origin-repositorystä annetusta haarasta.

```
git push origin haaran_nimi
```

"Työntää" muutokset origin-repositoryyn annettuun haaraan.

```
git remote -v
```

Listaa kaikki remotet

Remote-repositoryjä voi olla useita. Jos repository on kloonattu, git asettaa automaattisesti lähderepositoryn aliakseksi `origin`, ja käyttää sitä pull/push komentojen oletuksena. Komennoissa voi käyttää aliaksen tilalta myös suoraan SSH:ta; `user@domain.xyz:path/to/repo.git` master

git branch haaran_nimi

| Luo uuden haaran valitulla nimellä

git checkout haaran_nimi

| Vaihtaa aktiivisen haaran

git checkout -b haaran_nimi

| Kuten yllä, mutta luo ja vaihtaa haaran yhdellä komennolla

git merge haaran_nimi

| Liittää haaran master-haaraan



By **Urbanproof**

cheatography.com/urbanproof/

Not published yet.

Last updated 28th December, 2018.

Page 1 of 2.

Sponsored by **ApolloPad.com**

Everyone has a novel in them. Finish Yours!

<https://apollopad.com>