

Mika Huttunen ja Silja Polvi

Mikä Git?

Git on laajalti käytetty versionhallinnan työkalu, jota tarvitaan muunmuassa harjoitustöitä tehdessä. Versionhallinnan avulla esimerkiksi tiedostoihin tekemäsi muutokset ovat aina tallessa, ja voit palata edelliseen versioon, jos jokin menee pieleen.

Yksityiskohtainen “kädestä pitäen” -ohje Gitin käyttöön

Huomaa: Lyhyemmät ohjeet löydät [OhTu-kurssin harjoituksista](#).

Aloittaminen

1. Jos et ole koulun koneella ja koneessasi ei ole valmiiksi Gitä, voit ladata sen täältä: <http://help.github.com/win-set-up-git/>. Ohje on Windowsille, mutta sen alussa on linkit myös muita käyttöjärjestelmiä varten.
2. Mene osoitteeseen [Github.com/plans](https://github.com/plans)
3. Paina oikealla ylhäällä olevaa nappia “Create a free account”
4. Täytä vaadittavat neljä kenttää ja klikkaa “Create an account”
5. Paina oikealla olevaa harmaata nappia “New repository”
6. Laita repositorion nimeksi vaikka työsi nimi, OhHa tai Javalabra. Valitse pallura “Public” ja ruutu “Initialize this repository with a README”. Klikkaa lopuksi “Create repository”.

SSH-avaimen luominen omalla koneella

Katso ohje [tästä](#). Mikäli latsit Gitin yllä olevan linkin takaa Githubin sivuilta, käytä komentorivinä Git Bashia!

SSH-avaimen luominen laitoksen koneella

Huomaa: Laitoksen sivuilla on tähänkin vähän lyhyemmät [ohjeet](#).

SSH-avaimen luominen ei ole pakollista, mutta helpottaa versionhallinnan käyttöä.

1. Avaa komentorivi (Konsole tai Terminal)
2. Kirjoita komentoriville komento `ssh käyttäjätunnuksesi@melkki.cs.helsinki.fi`
3. Anna salasanasasi (laitoksen salanasana, sama jolla kirjaudut pajaluokassa koneelle)
4. Anna komento `ssh-keygen`
5. Ohjelma kysyy, mihin tiedostoon avain tallennetaan. Oletuksena tuleva paikka on hyvä, joten hyväksy painamalla enter.
6. Ohjelma pyytää asettamaan salasanan avaimellesi. Salasanaa ei tarvitse antaa, ja siten kirjautuminen on vaivattominta. Hyväksy salasananottomuus painamalla pari kertaa enter. Salasanan käyttäminen on hieman turvallisempaa mutta vaatii lisävaivaa.
7. Nyt komentorivin alussa pitäisi olla (edelleen) `käyttäjätunnuksesi@melkki~$`
8. Aseta ssh-hakemiston oikeudet antamalla ensin komento `chmod u=rwx,go= .ssh` ja antamalla seuraavaksi komento `chmod u=rwx,go= .ssh/*`
9. Lisää avaimen julkinen pari Githubiin: anna ensin komento `less .ssh/id_rsa.pub` niin pääset tarkastelemaan tiedostoa

10. Kopioi koko tiedoston sisältö (eli teksti)
11. Avaa nettiselaimella Github ja kirjaudu sisään (jos et jo ole kirjautuneena)
12. Paina oikeassa yläkulmassa olevaa jokoavaimen ja ruuvimeisselin kuvaa ("Account Settings")
13. Klikkaa vasemmalta "SSH Keys" ja sitten oikealta nappia "Add SSH key"
14. Anna nimeksi (Title) vaikka "Melkin avain" ja liitä Key-kenttään kohdassa 10 kopioimasi teksti. Poista tekstin seasta mahdolliset rivinvaihdot ja välilyönnit, jotta avaimesi sekaan ei jää "white spaceja". Paina lopuksi nappia "Add key".
15. Anna confirm-kentälle Githubin salasanasasi
16. Avaimesi on nyt lisätty Githubiin!

HUOM. Joudut tekemään tämän jokaisella koneella, josta haluat käyttää Githubia. Laitoksella riittää, että ssh-avaimen luomisen tekee kerran yhdellä koneella.

Repositorion valmisteleminen käyttöä varten

HUOMAA: Tässä ohjeessa komentorivillä tarkoitetaan Windowsin tapauksessa Git Bashia! Windowsin omat komentorivit eivät tunnista git-alkuisia komentoja, mikäli otit Gitin käyttöön yllä olevan linkin takana olevan Windows-ohjeen mukaan. Silloin sinun on siis käytettävä Git Bashia Gitin hallinnointiin.

1. Hankkiudu Githubin sivulla luomasi repositorion (esimerkiksi OhHa) näkymään. Sinne pääsee vaikkapa etusivulta kun olet kirjautunut, klikkaamalla repositorion nimeä.
2. Kopioi melko ylhäällä olevassa kentässä oleva osoite, joka on suunnilleen muotoa `git@github.com:käyttäjätunnuksesi/OhHa.git`
3. Avaa komentorivi ja anna komento tyyliin `git clone git@github.com:käyttäjätunnuksesi/OhHa.git`
4. Seuraavaksi ohjelma pyytää vahvistamaan äskeisen komennon (yes/no) - vastaa `yes`
5. Mene komentorivillä äsken kloonaamaasi kansioon (esim. OhHa) komennolla `cd OhHa` tms. Kaikki tiedostot voit listata komennolla `ls`.
6. Komennon `cd OhHa` jälkeen olet kansiossa, johon tulet tallentamaan projektisi. Nyt komennolla `ls` tulisi näkyä tiedosto README.

Pieni repotreeni

1. Varmista, että olet komentorivillä repositoriokansiossasi (esim. OhHa)
2. Avaa README-tiedosto vaikkapa komennolla `notepad README.md` (Windowsissa), `nano README` (koulun koneella), tai ihan millä tekstieditorilla haluat
3. Tiedoston pitäisi aueta valitsemassasi editorissa. Kirjoita tiedostoon jotakin ja tallenna se. Nanossa tallenna tiedosto painamalla `Ctrl + o`, ja vahvista tallennus painamalla `enter`.
4. Anna komentorivillä komento `git status`. Nyt näet luettelon tiedostoista, joita olet muokannut (modified), tässä tapauksessa README-tiedosto.
5. Anna komento `git commit -am "eka muokkaus"`. Jos viesti (tässä `eka muokkaus`) unohtuu, avautuu editori. Poistu editorista komennolla `:q`. Anna antamasi commit-komento uudelleen, tällä kertaa viestin kera.

6. Nyt tekemäsi muokkaukset (README-tiedoston muokkaus) ovat paikallisessa repositoriossasi. Sieltä ne pitää vielä työntää verkossa olevaan Githubin repoosi komennolla `git push`.
7. Mene Githubin sivulla olevaan repositoriosi näkymään (kuten edellisen ohjeen kohdassa 1) ja päivitä näkymä. Huomaat, että README-tiedostoon tekemäsi muokkaukset ovat nyt Githubissa asti, antamasi viestin kera!

Toinen repotreeni

1. Hankkiudu komentorivillä repositorion kansioosi (esim. OhHa). Anna komento `mkdir testikansio` luodaksesi uuden kansion.
2. Lisää testikansioosi jokin tiedosto: etsi tietokoneeltasi luomasi kansio ja luo tai siirrä sinne vaikkapa tekstitiedosto.
3. Lisää luomasi kansio versionhallinnoitavaksi antamalla komentoriville komento `git add testikansio`. Nyt versionhallinta seuraa kansiotasi siinä mielessä, että se näkyy `status`-komennolla, ja sen voi siirtää paikalliseen versionhallintaan (`commit`). Huomaa, että työpöytä kansio ei tule hallinnoitavaksi `add`-komennolla.
4. Anna `commit`-komento (`git commit -am "uusi testitiedosto ja -kansio lisattu"`).
5. Anna `push`-komento (`git push`). Tarkista Githubin nettisivulta, että uusi kansiosi ja sen sisältämä tiedosto ilmestyvät repositorionäkymääsi.

Nyt osaat lisätä Githubiin tiedostoja ja kansioita: `add` - `commit` - `push`! Tee samaan malliin harjoitustyöllesi dokumentaatiokansio ja sinne ensimmäisellä viikolla tarvittava dokumentointi.

Kolmas repotreeni: pull

Jos työskentelet useammalla kuin yhdellä koneella, voit versionhallinnan avulla pitää projektin ajan tasalla ilman että joudut siirtämään tiedostoja koneelta toiselle esimerkiksi muistitikulla. Sehän on yksi versionhallinnan hyödyistä! Tässä harjoituksessa pääset harjoittelemaan tiedostojen kopioimista Githubista koneellesi toisen repositoriokloonikansion avulla. Todellisessa tilanteessa harjoituksessa luotava toinen kloonikansio vastaa eri koneella olevaa kloonikansiota. Et siis tarvitse toista kloonia (varjorepo) muuhun kuin tähän harjoitukseen.

1. Luo uudella nimellä (esim. varjorepo) toinen klooni repositoriosta antamalla komentorivillä komento tyyliin `git clone git@github.com:käyttäjätunnuksesi/OhHa.git varjorepo`. Klooni luodaan siis muuten kuten kohdassa "Repositorion valmistelu käyttöä varten", paitsi että nyt kloonikansio nimetään itse.
2. Muokkaa **alkuperäisessä** repossa (esim. OhHa) olevaa tiedostoa README. Anna `commit`-komento ja `push`-komento alkuperäisessä repositoriokloonikansiossasi (OhHa), jotta äsken tehty muutos päättyy Githubiin asti.
3. Mene tämän harjoituksen alussa luomaasi varjorepoon (`cd ..` ja tämän jälkeen `cd varjorepo`). Avaa nyt README-tiedosto tässä uudessa repossa, jolloin huomaat että se ei ole muuttunut.

4. Vedä alkuperäisessä kansiossasi (OhHa) tekemäsi muutokset varjorepoon Githubista komennolla `git pull`. Huomaat, että README-tiedosto päivittyy niiden muutosten mukaan, jotka teit alkuperäisessä kloonissa (OhHa)! Tarkista tämä vielä varjorepossa olevasta README-tiedostosta.

Git & Netbeans

Git ja Netbeans ovat hyviä kavereita. Käytä Netbeans-projektin tallennuspaikkana koneellasi olevaa repositoriokloonikansiota: aseta Netbeansissa kohtaan "Project Location" koneella oleva repositorion kansio (esim. OhHa). Nyt tallennat projektiasi koko ajan repositorion kansioon.

1. Anna (kun olet ensin siirtynyt komentorivillä repositoriokansioosi) komentoriville komento `git status`. Huomaat kaksi uutta tiedostoa: harjoitustyösi lisäksi tiedoston `.gitignore`. Viimeksi mainittu sisältää Netbeans-tietoa, josta ei tarvitse välittää, mutta joka kuitenkin pitää lisätä versionhallinnoitavaksi.
2. Anna komento `git add harkkatyöprojektisiNimi` ja sen jälkeen komento `git add .gitignore`.
3. Nyt komennon `git status` pitäisi näyttää sekä harkkatyöprojektisi sisältöineen että `.gitignore`.
4. Anna commit-komento ja push-komento. Nyt harjoitustyösi löytyy Githubistasi!

Nyt voit aloittaa harjoitustyösi tekemisen. Muista **AINA työskentelyn alussa vetää** (pullata) Githubista projektisi, **ja töiden loputtua työntää** (pushata) **kaikki** tekemäsi muutokset Githubiin. Nyt voit jatkaa harjoitustyötäsi miltä tahansa tietokoneelta aina siitä mihin lopetit edellisellä kerralla.