

Sisällys

Gitin käyttö, perusteet	1
1 Lataa Git osoitteesta	1
2. Alusta paikallinen Git-repositorio	1
3. Lisää projektin tiedostot versiohallintaan.....	2
4. Tee ensimmäinen commit	2
4.1 Commit historia	2
5. Luo projektillesi haara.....	3
6. Luo etärepositorio GitHubissa	3
7. Yhdistä paikallinen repositorio etärepositorioon	3
8. Lähetä projektikansio -haaran GitHubiin	4
9. Tiedostojen palauttaminen etärepositoriosta	4

Gitin käyttö, perusteet

Projektin tarkoituksena on opetella Gitin käyttöä. Tuotos on tässä tiedostossa. Tarkoituksena on luoda ohjeistus, joka ei vaadi aiempaa osaamista tai kokemusta. Oletuskäyttöjärjestelmänä Windows

1 Lataa Git osoitteesta

<https://git-scm.com/downloads>

2. Alusta paikallinen Git-repositorio

Mene powershellillä tai komentokehoteella projektikansioon ja alusta Git:

```
cd kansio1/kansio2//projektikansio
```

suorita komento:

```
git init
```

Tämä komento alustaa Git-repositorion nykyiseen kansioon. Se luo piilotetun .git-kansion, joka sisältää kaikki versionhallintaan tarvittavat tiedot. Repositorio on se paikka, johon Git tallentaa kaiken

versionhallinnan kannalta tärkeän tiedon. Se tallentaa kaikki projektin muutokset ja mahdollistaa paluun aiempiin versioihin.

3. Lisää projektin tiedostot versiohallintaan

Suorita komento:

```
git add .
```

Jos haluat lisätä vain yhden tiedoston version hallintaan, käytetään komentoa `git add tiedostonimi.txt`

Tämä lisää kaikki projektikansion tiedostot versionhallinnan staging alueelle. **HUOM.** Tämän jälkeen tehdyt muutokset eivät siirry staging alueella oleviin tiedostoihin automaattisesti vaan pitää suorittaa uusi `add` komento. Esimerkki:

Tiedostonimi.txt sisältö "moi"

```
git add .
```

Staging alueen tiedostonimi.txt sisältö = "moi"

Tiedostonimi.txt sisältö "moi kaikki", paikallinen tallennus

Staging alueen tiedostonimi.txt sisältö = "moi"

Tässä tulee olla tarkkana, että versionhallintaan lähtevät halutut muutokset. Yksin työskennellessä potentiaalisesti ärsyttävää, tiimissä potentiaalisesti kriittistä.

4. Tee ensimmäinen commit

Tallenna muutokset paikalliseen repositorioon `commit`-komennolla:

Suorita komento:

```
git commit -m "Ensimmäinen commit"
```

Kommentti on pakollinen.

4.1 Commit historia

Suorita komento:

```
git log
```

Tulos esimerkiksi:

```
commit 5972c9b91ea81460b1f4f7e6249601e5725fd759
```

```
Author: Juha Riikonen <riikonen85@hotmail.com>
```

```
Date: Sat Oct 12 18:29:53 2024 +0300
```

Django_tutoriaali tiedostot lisätty

Suorittamalla komennon

```
git checkout commit_tunnus
```

Projektikansio siirtyy siihen tilaan jossa se oli kyseisen commit tunnuksen jälkeen. **HUOM.** Voit tehdä muutoksia, mutta näitä muutoksia ei liitetä mihinkään haaraan, ellet luo uutta haaraa tai teet jatkotoimia.

5. Luo projektillesi haara

Suorita komento:

```
git checkout -b projektin_nimi
```

-b viittaa haaraan (branch)

6. Luo etärepoitorio GitHubissa

Mene GitHubiin ja luo uusi repositorio.

1. Siirry GitHubiin ja valitse New Repository.
2. Anna repositoriolle nimi (esim. Projects).
3. Jätä muut asetukset ennalleen ja valitse Create Repository.
4. Kopioi GitHub-repositorion osoite, perusmuoto: <https://github.com/kayttajanimi/Projects.git>

Luonnollisesti etärepoitorio on elintärkeä tiimin kanssa työskenneltäessä.

7. Yhdistä paikallinen repositorio etärepoitorioon

Suorita komento

```
git remote add origin https://github.com/kayttajanimi/Projects.git
```

Tämä komento kertoo Gitille, että etärepoitorio sijaitsee osoitteessa <https://github.com/kayttajanimi/Projects.git> ja sen nimi on origin.

Mikäli unohdat etärepoitorion nimen, suorita komento

```
git remote -v
```

8. Lähetä projektikansio -haaran GitHubiin

Nyt voit lähettää ("pushata") projektikansion etärepositorioon.

Suorita komento:

```
git push -u origin projektikansio
```

-u origin projektikansio tarkoittaa, että Git muistaa jatkossa, että haluat työskennellä tämän haaran ja origin-etärepositorion kanssa, joten sinun ei tarvitse kirjoittaa haaraa joka kerta.

9. Tiedostojen palauttaminen etärepositoriosta

Suorita alla oleva komento paikallisen projektikansion sijainnista:

```
git pull
```

Tämä päivittää paikalliset tiedostot etärepositorion mukaisiksi. Komento voidaan esittää täsmällisempänä esimerkiksi muodossa

```
git origin main
```

 jossa haetaan origin nimisen etärepositorion main haarasta uusimmat tiedot.