

Git-kurs

Versjonkontroll med WebKom

19. februar 2020

<https://github.com/kaprests/Gitkurs>

Hva er git?

GitHub

Git \neq GitHub!

- Git
 - Versjonkontrollprogram
- GitHub
 - Vertstjeneste for git-repoer
 - Finnes flere alternativer (GitLab, Bitbucket etc.)

Hva er git?

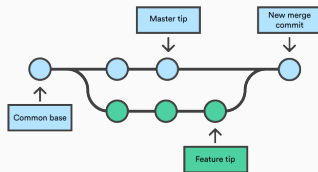
- Tar “bilde” av koden når du vil
- Gjør det enkelt å bla mellom ulike “bilder”
- Man kan dele “bildene” med andre

Terminologi

Noen viktige begreper

- Commit
- Branch
- Repository
 - Lokalt/remote

Om en
forstår disse begrepene forstår en git.



Et slags “snapshot” av koden/prosjektet.

En commit består av:

- Endring fra forrige commit
- Forelder
- Hashkode
- Metadata

En oppretter commits på logiske/taktiske tidpunkter, f.eks. etter å ha fikset en bug eller implementert en ny funksjon.

En kan alltid returnere til en tidligere commit

Alle commits tilhører en branch

Hovedbranch: master

Head: referanse til nåverende branch

Merging

- Sammenslåing av branches
- Mergekonflikter

For større endringer (mer enn en commit) oppretter en gjerne en egen branch som man commiter endringene på, før en merger den inn i master.

Som regel prosjektmappen

Inneholder:

- Alle filene du vil holde styr på og deres historikk
- Alle commits

Kan lagres på f.eks. GitHub

- cloning
- pull/push

Hvordan bruke git

Det finnes GUI

- ikke standard

Standard å bruke kommandolinjen

Ulike IDE-er har ofte git-integrasjon og mulighet for å gjøre ting med GUI

Oversikt over de viktigste kommandoene

- `git init`
- `git add`
- `git commit`
- `git pull`
- `git push`
- `git branch`
- `git checkout`
- `git merge`
- `git clone`

Git arbeidsflyt - lokalt repo på din pc

1. Gjør endringer
2. Stage endringer med `git add`
3. Lag commit med `git commit`

Git arbeidsflyt - repo med kun deg, med remote

1. Gjør endringer
2. Stage endringer med `git add`
3. Lag commit med `git commit`
4. Push endringer til origin (GitHub) med `git push`

Git arbeidsflyt - remote med flere bidragsyttere, liten endring

1. Gjør endringer
2. Stage endringer med `git add`
3. Lag commit med `git commit`
4. Hent nyeste versjon fra origin med `git pull`
5. Push endringer til origin (GitHub) med `git push`

Git arbeidsflyt - remote med flere bidragsyttere, større endring

1. Lag ny branch med git branch <navn på ny branch>
2. Bytt til din nye branch med git checkout <navn på branch>
3. Gjenta til endringen er klar
 - 3.1 Gjør endringer
 - 3.2 Stage endringer med git add
 - 3.3 Lag commit med git commit
4. Bytt til master branch med git checkout master
5. Hent nyeste versjon fra origin med git pull
6. Merge din branch med endringer inn i master med git merge <navn på din branch>
7. Push endringer til origin (GitHub) med git push

Oppgave 1: Opprett et git-repository

- Lag prosjektmappe (ev. bruk noe du har fra før)
- Bytte til mappen/åpne en terminal i mappen
- `git init`

Oppgave 2: Commit noen endringer

- Gjør endringer (opprett/endre/slette file(er))
- Stage ønskede endringer (git add)
- Commit (git commit)

Gjenta dette noen ganger og skriv git logfor å se commit-loggen.

Oppgave 3: Få det på GitHub

- Opprett nytt, tomt repo på GitHub (uten README)
- Set remote
- Push til origin master

Når du har gjort dette kan fremtidige commits lastes opp til GitHub med bare `git push`.

-

Får du til dette kan du nok til å bruke git med dine egne prosjekter!

Oppgave 4: Samarbeid - prosjekt med flere bidragsyttere

- ...

Terminal cheat sheet (bash og mac)

- bytte mappe: cd
 - cd .. - (en mappe "opp")
 - cd Documents/gitkurs/ - (Bytt til mappen gitkurs)
- mkdir <mappenavn>- Opprett ny mappe

GitHub Pro

Som student har du rett på GitHub Pro gjennom GitHub Education, dette er helt gratis! Se <https://education.github.com/pack>.

Grafisk tutorial på avansert bruk

<https://learngitbranching.js.org>