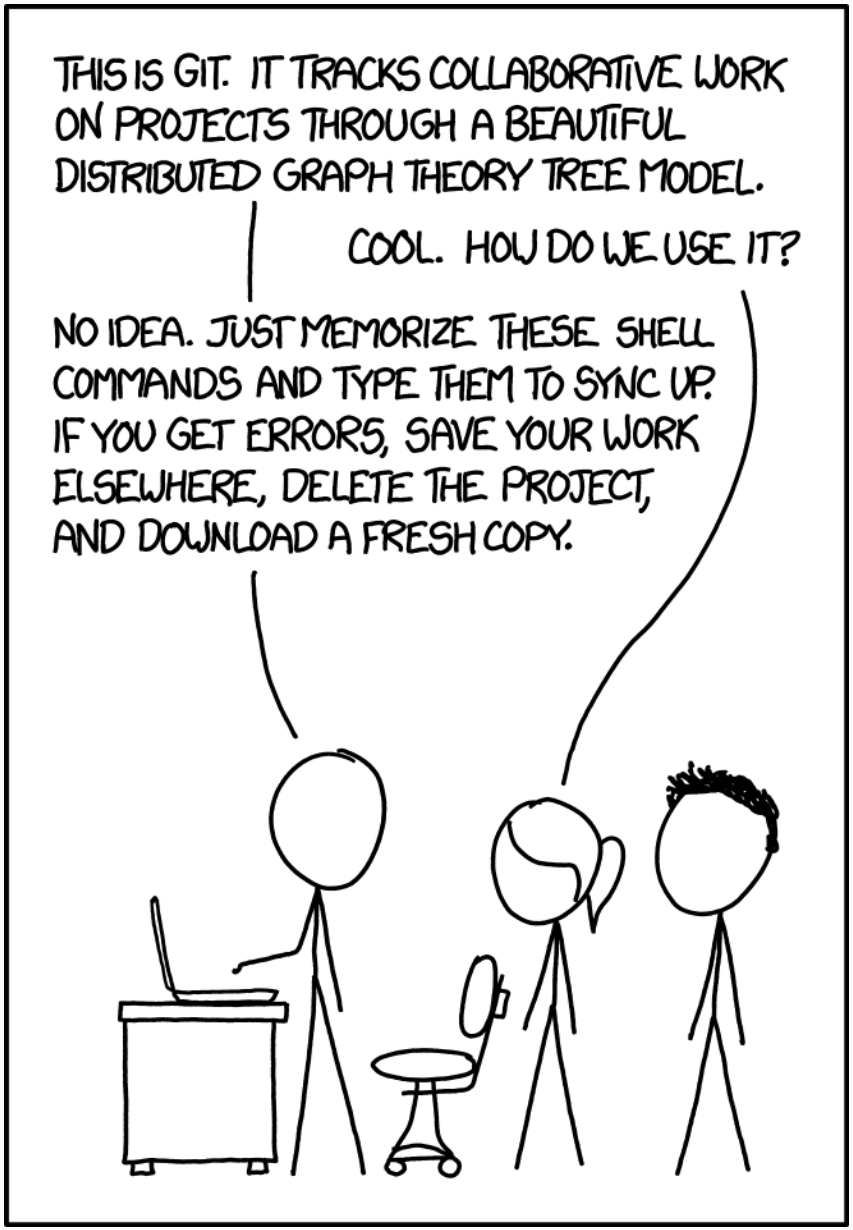
Git-kurs

Versjonkontroll med WebKom

11. mars 2020

`https://github.com/kaprests/Gitkurs`


THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

Hva er git?

GitHub

$$\text{Git} \neq \text{GitHub!}$$

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

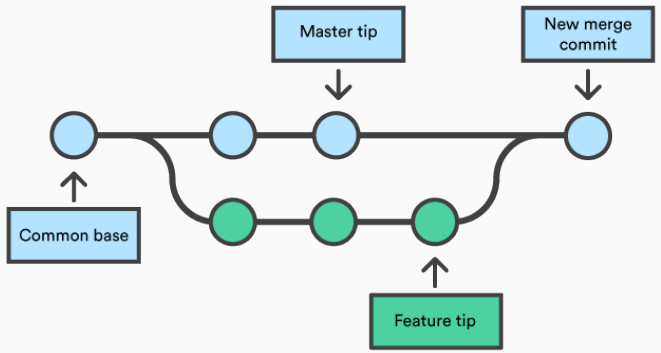
Hva er git?

- 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.

Commit

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

Branch

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.

Repository

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

Intro til kommandolinjen

Grunnleggende bruk, antar Linux eller Mac -terminal.
For Windowsbrukere: git-bash
Kommandoer:

- ls : list - printer innholdet i nåværende mappe
- cd <mappenavn >: change directory - bytt til mappe
  - . betyr denne mappen .. betyr mappen over
  - cd .. bytter altså til mappen over nåværende mappe
- mkdir <nytt mappenavn >: Oppretter ny mappe
- cat <filnavn >: skriver ut innholdet til en fil

Dersom du kun bruker terminalen til git, så trenger du egentlig ikke å kunne mer enn de to første kommandoene, i tillegg til gitkommandoer da, såklart.

Oppgave: Bruke kommandolinjen

1. Opprett en ny mappe med en eller flere filer
2. Naviger til mappen i kommandolinjen med cd
3. list innholdet i mappen med ls
4. Opprett enda en ny mappe inne i mappen med mkdir
5. Gå inn i den nye mappen med cd
6. Gå tilbake til den første mappen med cd ..

Kommandoer

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 bidragsytere, 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 bidragsytere, 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

Oppgaver

Oppgave 1: Opprett et git-repositorry

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

Oppgaver

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.

Oppgaver

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!

Oppgaver

Oppgave 4: Samarbeid - prosjekt med flere bidragsytere

- ...

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

Avsluttende tips

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`