

git

GitHub

INTRODUKSJON TIL GIT OG GITHUB

Presentasjon av [Kudakwashe Kidwell Chambwe](#)

DAGENS AGENDA

1. Hva er **versjonshåndtering**?
2. **Hva** er git?
3. **Hvorfor** git?
4. **Hvordan** git?
 - git "språket"
 - git basics
 - git arbeidsflyt
 - git verktøy
 - tips og tricks
5. GitHub

HVA ER VERSJONSHÅNDTERING?

Versjonshåndtering (Version Control) er et verktøy som lar oss..

SAMARBEIDE

Samarbeide med andre på ett eller flere prosjekter

HÅNDTERE OG OVERVÅKE ENDRINGER

Vi alle gjør feil.. Tanken med versjonshåndtering er at vi skal kunne se alt som er blitt gjort, og ha muligheten til å tilbake stille ved eventuelle feil

HVA ER GIT?

VCS

Et **versjonshåndteringssystem** (VCS) for å håndtere og administrere endringer av filer og eventuelt koordinere når flere samarbeider

DVCS

Et **distribuert** versjonshåndteringssystem (DVCS), hvor brukerne/klienter speiler hele repoet istedet for å stadig hente ut det nyste filene (i.e. sentralisert). M.a.o alle brukere har en kopi av repoet.



HVORFOR GIT?

- **Raskt!** Henter informasjon om filer og prosjektet raskt og effektivt, siden nesten alt er lokalt.
- **Distribuert!** Alle har en kopi av "repoet"
- **Skalerbart!** Gir mulighet for tusenvis av utviklere til å samarbeide på samme prosjektet og samme koden.

HVORFOR GIT?

- **Lokalt!** Trenger ikke nettverk. Alt kan gjøres offline, med mindre man skal jobbe med andre (i.e. **GitHub**).
- **Branches!** Eksperimenter med koden og prosjektet i et separat miljø. Enkelt å lage, endre og slette "features".
- **Håndterbart!** Alle en lokal kopi av repoet, som betyr at alle har oversikt over filene og historien til filene til enhver tid.

HVORDAN GIT?

GIT "SPRÅKET"

REPOSITORY?

Ett "**oppbevaringssted**" eller "**mappe**" hvor alle filene som er i git-prosjektet befinner seg

BRANCH?

En **gren**, eller en uavhengig serie med commits in versjonsdatabase. Kan brukes som en "lekeplass" for å eksperimentere med features uten å påvirke hovedfilene

COMMIT?

Et **objekt** som holder på informasjonen om en spesifikk endring. Ofte har en **melding**.

GIT BASICS

```
git clone [url]
```

Kloner repositoret du henter fra.

```
git pull
```

Henter nyeste versjon av repoet du befinner deg i.

```
git push
```

Laster opp nyeste comittede versjonen du har av repoet du befinner deg i.

```
git add [file]
```

Legge til, nye eller endrede filer som du vil laste opp til repoet.

GIT BASICS

```
git commit -m [message]
```

Før filer kan lastes opp, må de committes. Dette må gjøres med en melding.

```
git branch [branch-name]
```

Lager en branch/gren.

```
git merge [branch]
```

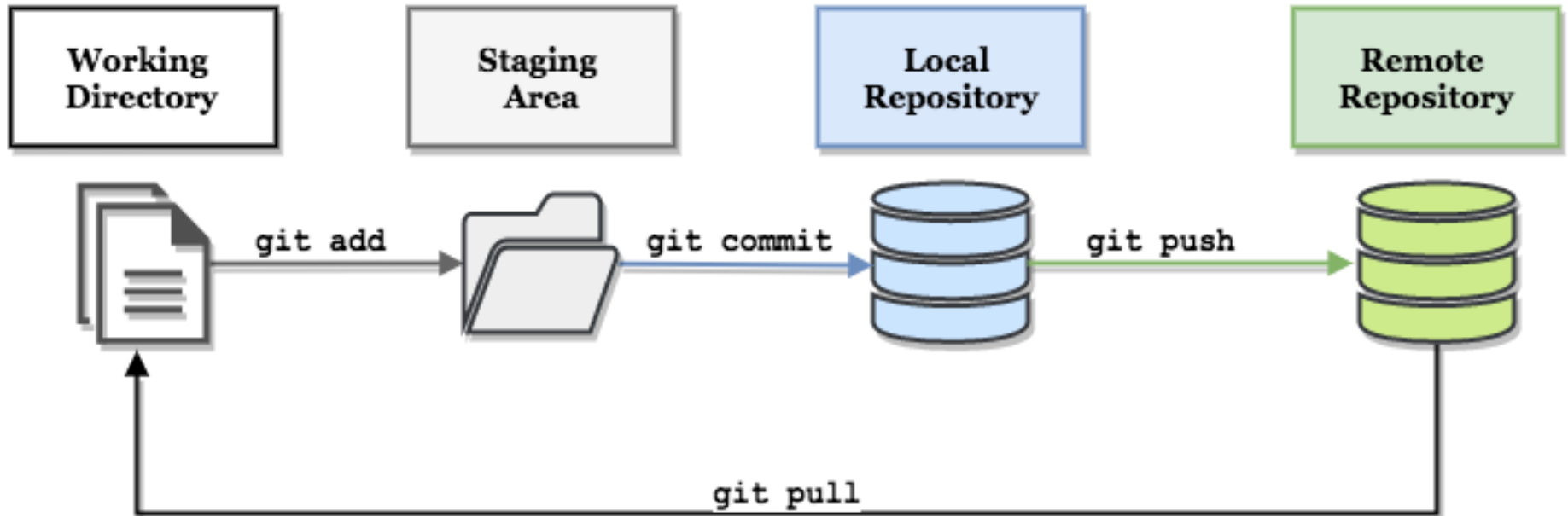
Laster opp alle endringer (i.e. **slå sammen**) fra en gren til den du står på (ved kall). NB: Her kan det oppstå konflikter(!).

```
git checkout [branch-name]
```

Skifter gren du står på. Brukes for å navigere seg fra/til ulike grener.

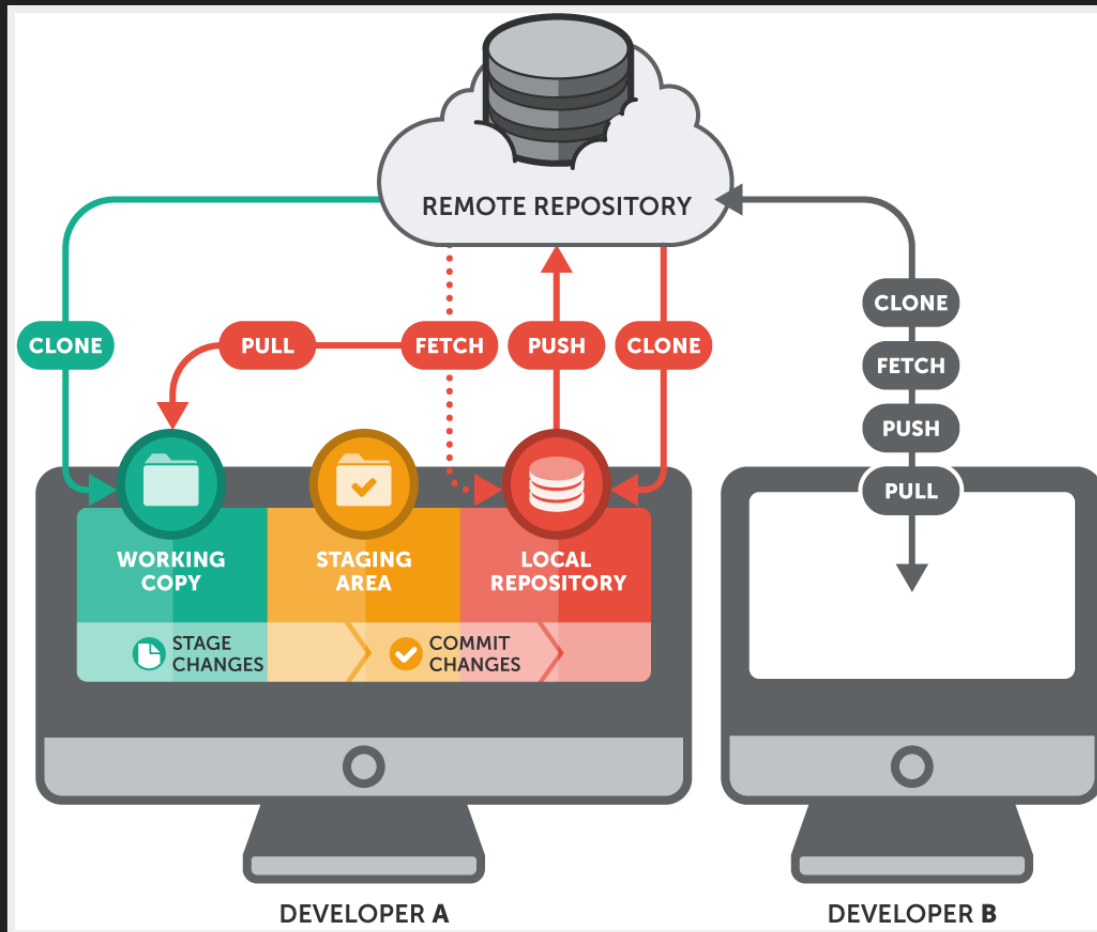
GIT ARBEIDSFLYT

EN FORENKLET ILLUSTRASJON



GIT ARBEIDSFLYT

DET STORE BILDET



GIT ARBEIDSFLYT

SPESIELLE FILER

`README.md`

Les-meg-fil på GitHub er denne filen som vises på github-siden deres.

`.gitignore`

Listen av filer/mapper som ikke skal committes (e.g. libraries, api-nøkler, etc..) [.gitignore generator](#)

`LICENSE.txt`

Lisensen du publiseres under. [Oversikt her](#)

GIT VERKTØY

GUI KLIENTER

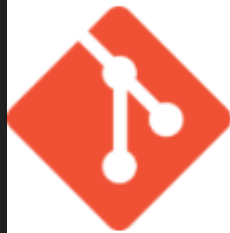
Det finnes et bredt utvalg av **git-applikasjoner** med grafiske grensesnitt. De ulike applikasjonene gir et visuelt oversikt av bl.a. git-kommandoer og repoet. [Oversikt her](#)

GIT WEBTJENESTER

Det finnes også ulike **webtjenester** som tar seg av å "hoste" git-repoer. E.g. Gitlab, Bitbucket, GitHub (mest aktuelt), etc...



GIT VERKTØY



git

GitHub



ATOM



eclipse



IntelliJ IDEA

TIPS OG TRICKS

#1 RESPEKTÉR MASTER

ALDRI last opp kode som ikke kjører til master. Worst case; bruk branch.



TIPS AND TRICKS

#2 BRUK GIT!

Det kan være skremmende.. men det er virkelig nyttig. Dere sparer mye tid i det lange løp. When in doubt, google it!

#3 SLACK!

More code, more problems.. Viktig på kunne kommunisere for eks. ved mergekonflikter. Viktig verktøy for kommunikasjon er slack. Kan integreres med **git**



Training

Git & GitHub Training (External)

GitHub

