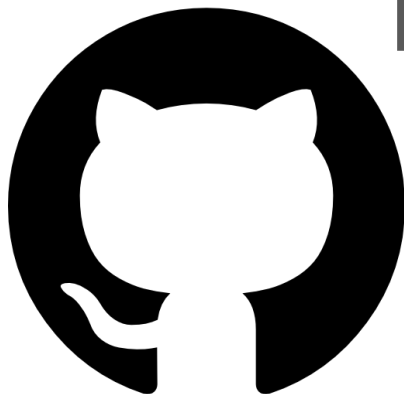


Versjonskontroll og GitHub

Lars og Sigurd IN2000 V26

slides stjeltish fra Marius og Victor V25



Agenda

- Hvem
 - hva
 - hvor
 - hvorfor
- Viktige kommandoer
 - tips and tricks
- Ditt første repo med Sigurd :)
- LAB?

In case of fire



1. git commit



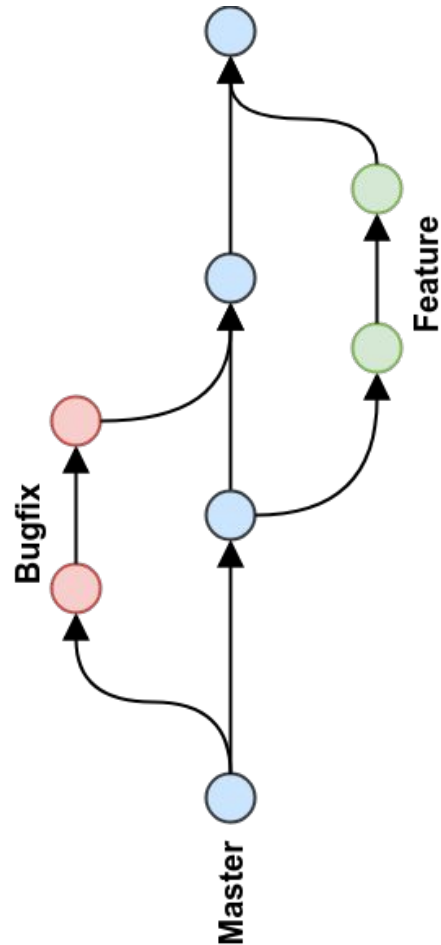
2. git push



3. leave building

Hva er git?

- Open source versjonskontrollsystem (VCS)
- VCS: Tracke endringer i kodebaser
- VCS: Lar flere utviklere samarbeide om kode
- Brukes fra CLI eller integrert i IDE



Hvorfor bruke et VCS?



One Devloper Army @OneDevloperA... · 3d

When you delete a block of code that you thought was useless



9

138

632



```
oblig1_ferdig_2.java    oblig1FERDIG_FAKTISK.java    oblig1FERDIG.java
oblig1_ferdig_denne.java oblig1_ferdig.java           oblig1.java
```



Git og GitHub er *ikke* det samme



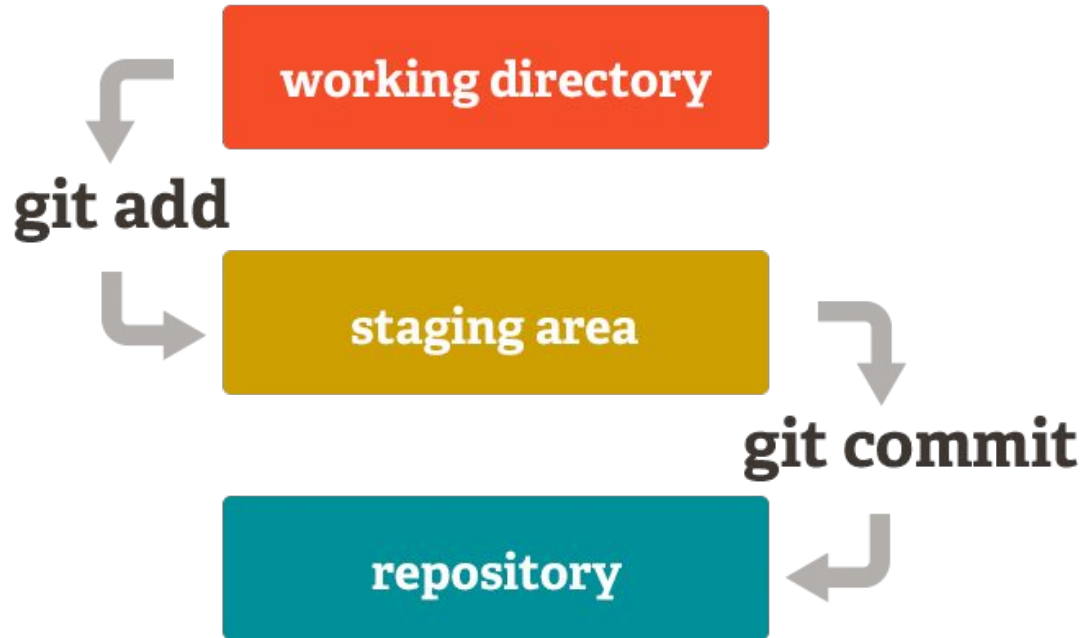
- Utviklerplattform på web
- “OneDrive for kode”
- Distribuert versjonskontroll
- Tilbyr tilgangskontroll / bug tracking / task management osv.



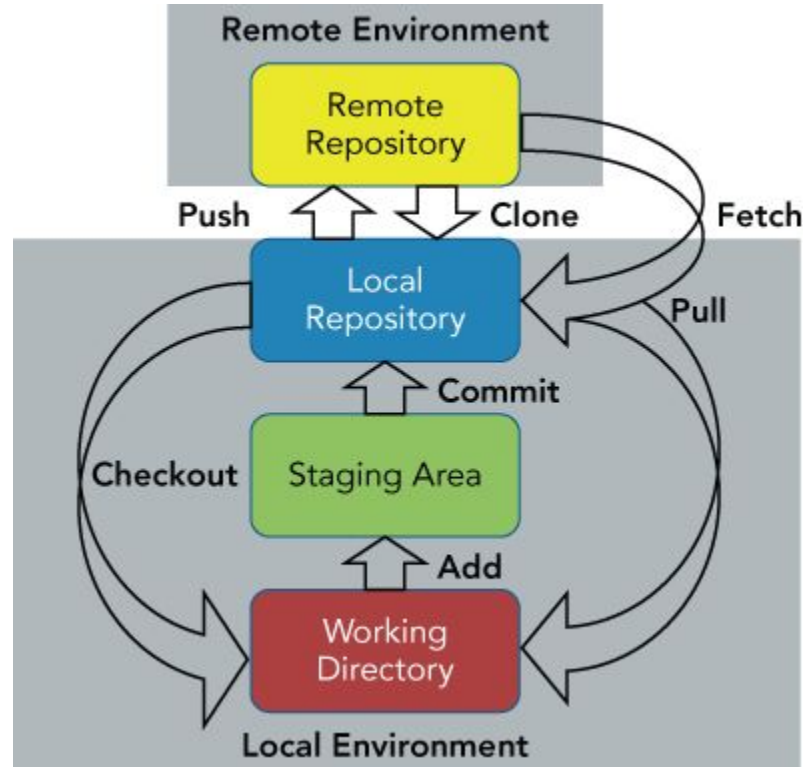
- Lokalt verktøy
- Lokal versjonskontroll
- Samarbeider enkelt med GitHub eller andre liknende tjenester

Viktig å forstå forskjellen!

Arbeidsflyt i *git*



Arbeidsflyt i *GitHub*



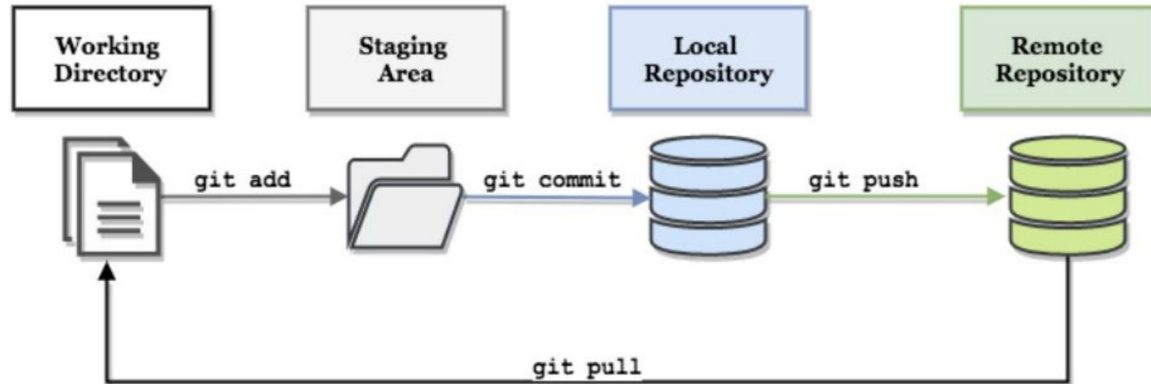
Kommandoer i git



git add [file]

Gjør: Flytter endringer fra *working directory* til *staging area*

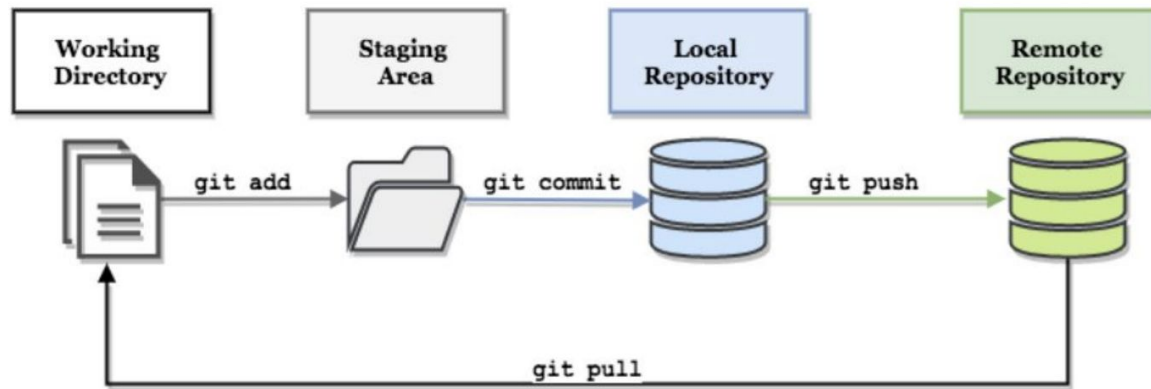
Formål: Lar deg forberede et *snapshot* av koden før den legges inn i commit-historikken



git commit

git commit -m “[descriptive message]”

Gjør: Flytter et staget *snapshot* til prosjekthistorikken





```
% git add .  
% git commit -m '1'
```

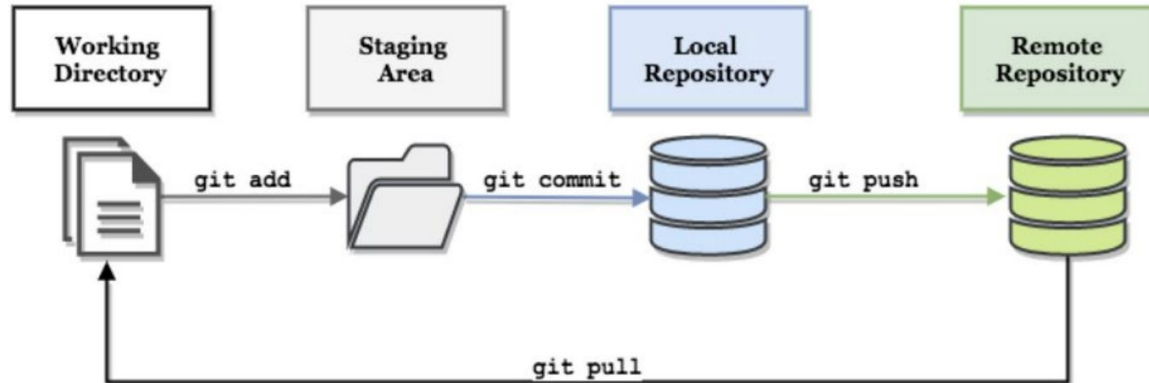


Git commit
message
actually summarizes
what you
changed in the code

"Small
changes"

git push

Gjør: Dytter endringer fra *lokalt repository* til f.eks GitHub



git fetch

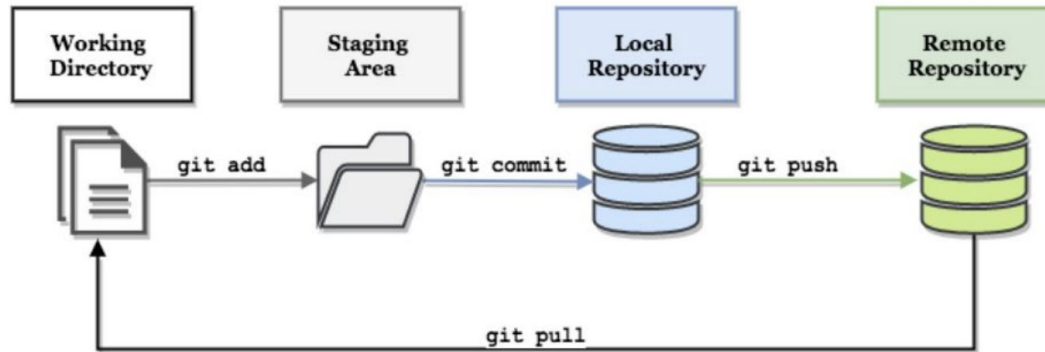
Gjør: Laster ned endringer fra et *remote repository*, men integrerer **ikke** endringene i ditt *working directory*

Formål: Du får muligheten til å inspisere endringene før du merger disse inn i ditt lokale prosjekt

git pull

Gjør: Laster ned endringer fra *remote repository* og *merger* inn i ditt lokale prosjekt.

NB: Ja, her overskriver du ulagrede endringer. Bruk git *aktivt* for å lagre unna endringer *før* du bruker git pull.

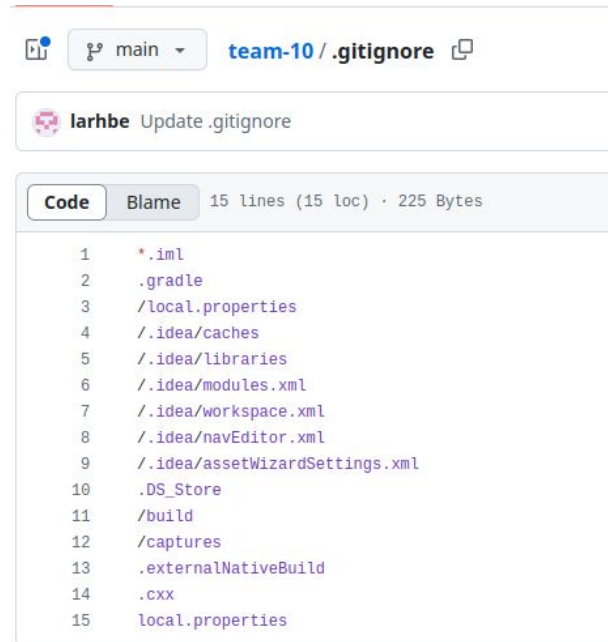


Andre nyttige kommandoer

- **git log**
 - Viser commit-hisotikk
- **git diff**
 - Viser hva som er endret i filer
- **git status**
 - Filer som er endret, filer som er staged, filer som ikke spores
- **git stash**
 - Lagrer endringer midlertidig.

Tracked vs. untracked files

- Du behøver ikke laste opp *alle* filene i prosjektet ditt på GitHub
- Vi *tracker* typisk:
 - Nødvendige filer
 - Filer vi ønsker å la andre endre på
- Vi tracker **ikke**
 - Environment variabler
 - Cache / build-relaterte filer
 - API-nøkler!!? 🧠
 - DB-endepunkt
- .gitignore-fil



The screenshot shows a GitHub repository interface for a file named `.gitignore` in the `team-10` repository, specifically on the `main` branch. The file is 15 lines long and 225 bytes. The content of the file is as follows:

```
1 *.iml
2 .gradle
3 /local.properties
4 /.idea/caches
5 /.idea/libraries
6 /.idea/modules.xml
7 /.idea/workspace.xml
8 /.idea/navEditor.xml
9 /.idea/assetWizardSettings.xml
10 .DS_Store
11 /build
12 /captures
13 .externalNativeBuild
14 .cxx
15 local.properties
```

Branching

- En *branch* er et uavhengig *working directory*, *staging area* og *commit history*
- Commits er altså *lokale* for en branch
- Branches har navn!
 - Hovedbranchen kalles *main* (tidligere: *master*)
 - Andre branches navngis etter formål (development, test, feature, fix m.m)
- En branch grener ut fra et *snapshot* av koden i ditt *working directory*
- Gir deg muligheten til å branche ut - gjøre endringer på kode - revert om det ikke fungerer

Merge

- Merge kombinerer endringer fra en branch inn i en annen
- Som regel når en feature er ferdig utviklet
- Typisk: feature-branch -> main
- Merge conflicts :(
- Pull-requests på GitHub
 - Lettere å resolve konflikter

Keyboard fix #240

Merged piaoka merged 2 commits into dev from keyboard-fix on May 9, 2025

Conversation 1 · Commits 2 · Checks 0 · Files changed 5

hennion (Henning Osmo Nordha... commented on May 9, 2025) ...

No description provided.

hennion added 2 commits 10 months ago

- fixed keyboard and made networkbanner prettier 889c885
- merged with dev 8fc2371

plaoa (Pia Ottar Karlßen) commented on May 9, 2025 ...

lgtn

plaoa merged commit 7ab6432 into dev on May 9, 2025 Revert



Kode i fellesskap: Branching-strategier

- Ofte mye kaos i starten
- Strategiene er smidige, kan alltid tilpasses
- Det viktigste er felles enighet om arbeidsflyt

Prinsipper for samarbeid om kode

- Én person i én fil
- Ikke jobb på samme features - men gjerne parprogrammér
- Commit ofte
- Én ting per commit, ikke endre mange store deler av kodebasen samtidig
- Avhengig av strategi: oppdater branchen din ofte
- Pull før push ^
- LAG PULL REQUEST, IKKE PUSH TIL MAIN

Git er tidkrevende å lære seg...

Så her er noen hjelpemidler:

- <https://education.github.com/git-cheat-sheet-education.pdf> 🐮
- <https://learngitbranching.js.org/> 🧩
- ChatGPT / KI / språkmodeller (forslag skal tas med en klype 🧂)