
Velkommen til gruppetime i IN2000!

Gruppe 5 Jamila og Lars

Viktige beskjeder!

- Underkjent oblig? :(
- Dobbeltsjekk feedback og status + evt. ny frist



Oppsummering oblig 2

Det viktigste

- Jobbe med API
 - Bli kjent med (viktigheten av) arkitektur
 - Jobbe med en litt større App
-

Git / Github

Git

vs.

GitHub



Git is installed and maintained on your local system (rather than in the cloud)



Git is a high quality version control system

First developed in 2005



One thing that really sets Git apart is its branching model

GitHub is designed as a Git repository hosting service



You can share your code with others, giving them the power to make revisions or edits

GitHub is a cloud-based hosting service



GitHub is exclusively cloud-based



Nødvendige kommandoer

tre steg, i denne rekkefølgen, for å legge dine endringer til i branchen

| | |
|----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| <code>git clone <link></code> | henter ned et eksisterende prosjekt fra github, til din pc (lokalt) |
| <code>git pull</code> | henter inn eventuelle endringer og oppdateringer som har blitt gjort på branchen |
| <code>git status</code> | kan se status på filene, slik at du f.eks. vet hva du skal add'e |
| <code>git add .</code> | legger til *alt* |
| <code>git add <filename></code> | legger til de filene som ramses opp |
| <code>git commit -m "melding"</code> | setter sammen alle dine endringer og forbereder til et eventuelt push. Husk alltid meldingen (gjørne en beskrivende en)! |
| <code>git push <branchname></code> | pusher dine commits, og de ligger nå på den oppgitte branchen på github |
| | |
| <code>git branch</code> | for å se listen over alle branchene i ditt repo, og se hvilken branch du er på nå |
| <code>git branch <branchname></code> | for å lage en ny branch |
| <code>git checkout <branchname></code> | for å gå til den nevnte andre branchen |

Legger brevet i konvolutten



`git add <filnavn>`

Setter på et frimerke
og gjøres klart til...



`git commit -m "melding"`

Sendes!



`git push`

Alle passasjerer går ombord



git add <filnavn>

Boarding completed,
dørene lukkes, setebeltene festes...



git commit -m "melding"

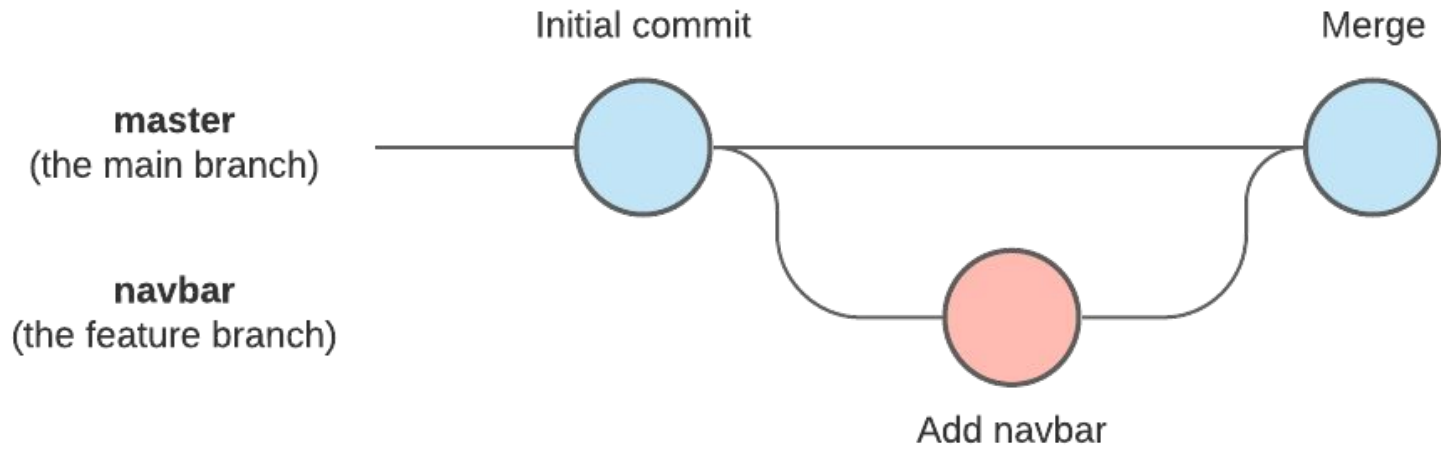
Takeoff!

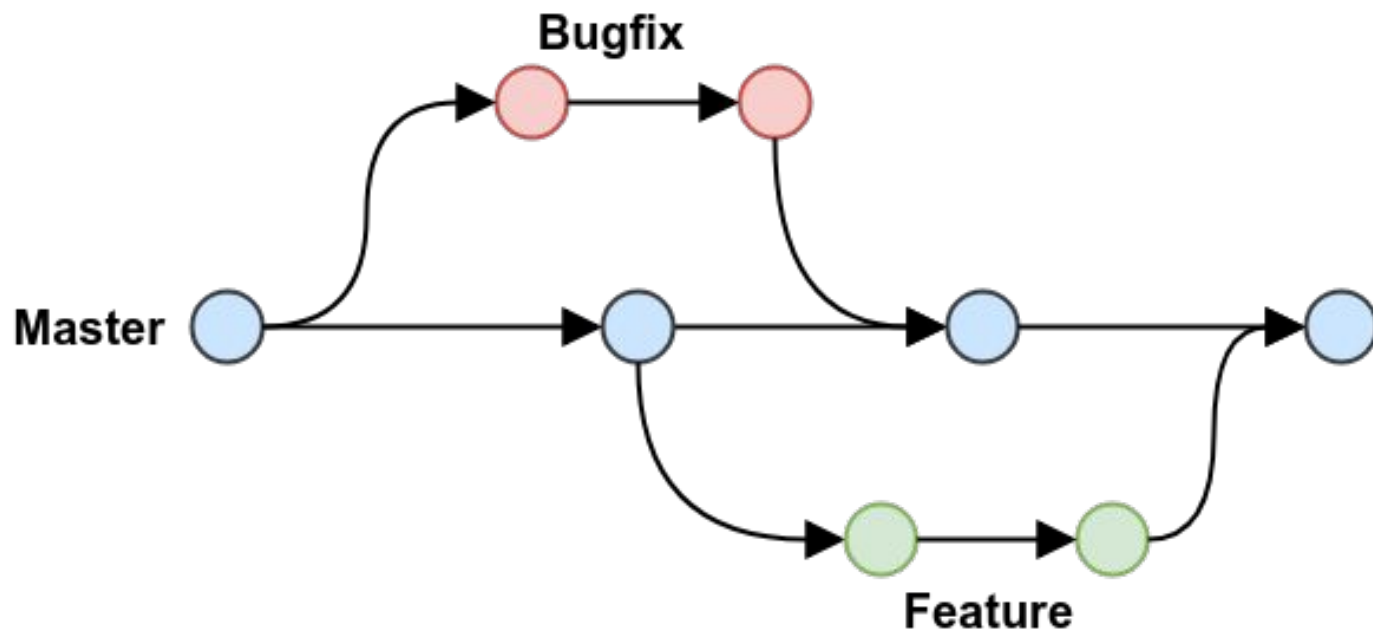


git push

...og ikke sånn 😞







Prøv selv!

Fin ressurs for å lære mer om git (datamodellen til git):

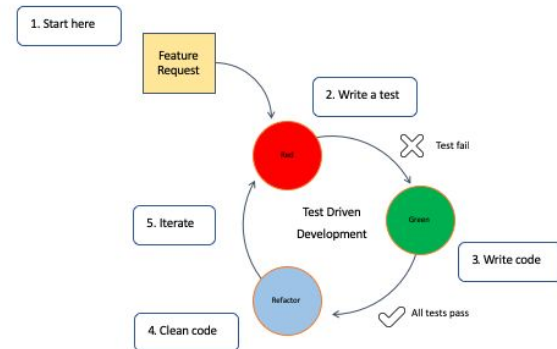
<https://missing.csail.mit.edu/2020/version-control/>

Morsomt git-spill for å øve seg på git: <https://learngitbranching.js.org/>

Testing

Hvorfor tester vi?

- Mål med testingen
 - oppdage mulige defekter
 - foreslå eventuelle forbedringer
 - løse utfordringene før de utarter seg til å bli faktisk problemer
 - gjøre applikasjonen mer robust
- Selv små feil kan være veldig kostbare
 - sinker eller irriterer vårt normale arbeid
 - kan forårsake alvorlige følgefeil og defekter



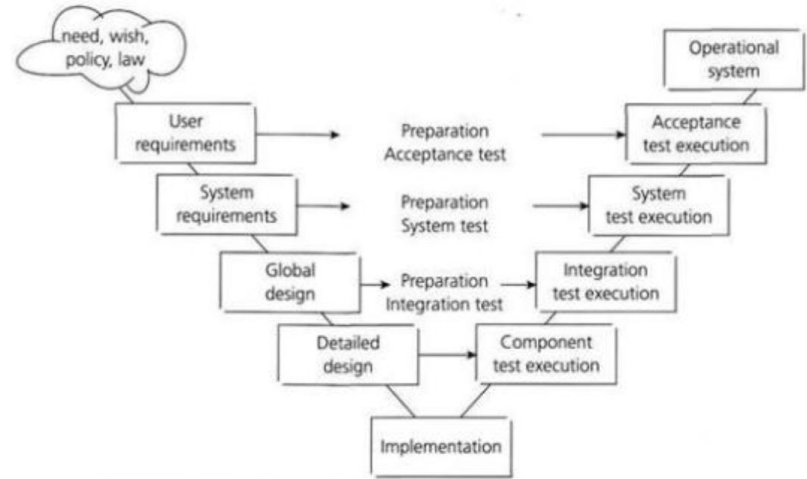
Hvordan tester vi?

- Uformell testing underveis i utviklingen

- kodegjennomgang
- bruk av logg-meldinger
- utskrift til terminal
- sjekk i emulatoren

- Kontinuerlig testing (TDD)

- enhetstesting (testing av individuelle komponenter, f.eks. JUnit5 bibliotek)
- integrasjonstesting (testing av hvordan komponentene fungerer sammen)
- systemtesting (testing av hele systemet, ende-til-ende testing)
- akseptansetesting (for å bekrefte at programvaren oppfyller kravene som ble definert i starten av prosjektet)



Vi skriver noen tester!!

Nyttig: <https://junit.org/junit5/docs/current/user-guide/>

Send oss en mail:
camilltk@uio.no
jamilakm@uio.no
larshaas@uio.no
