## Git og Github – notater for å huske hvordan man gjør det:

Beste tutorial: Kevin Stratvert: <https://www.youtube.com/watch?v=tRZGeaHPoaw>

Første gangs oppsett av repository osv:

Når vi bestemmer directory for vår Git så må vi skrive cd og adressen til mappa på pcn vår, men husk, som David Hasselø sa: vi må ha "hermetegn først og hermetegn (quote/end quote)" på slutten. Fredrik Strømsholm kom med et godt alternativ også: Når man har mellomrom i adressen kan man også skrive \ etter mellomrom, da er ikke " nødvendig.

Eksempel:

cd "C:\Users\jokane\Dropbox\JornGit2023" # cd står for change directory, deretter

git init som initierer repositoryn, da får du samtidig ei skjult mappe i gitrepositorymappa di som heter .git og da vet du at dette er en git-repository.

### Branches1

jokane@LT-VKRI-416 MINGW64 ~/Dropbox/JornGit2023 (master)

$ git branch -a

\* master

Det som står over her er min master branch. Om jeg skriver git branch -a får jeg svar på hvilken branch jeg befinner meg i.

Et bilde som inneholder tekst

Automatisk generert beskrivelse

Tracking files (filene som ikke er tracked registreres ikke i git)

git add navnpå.fil

clear fjerner alt fra skriveloggen på en måte.

### Daglig oppstart:

Once you have navigated to the correct folder, you can initialize Git on that folder:

Start Git bash og tast inn:

cd "C:\Users\jokane\Dropbox\JornGit2023"

Eller høyreklikk og lim det inn

Du kan også finne mappa i filutforsker, høyreklikk og velg

Git Bash here!

.gitignore er navnet du kan gi til en tekstfil og der kan du legge inn filnavnene på de filene som skal ignoreres i gitbash. Om du adder alle filene med git add . kan du ikke skjule dem med .gitignore etterpå...

### Staging:

git add . Når du har valgt git add minfil.py så er den fila staged

Om du angrer staging så kan du bruke denne kommandoen:

git restore –staged navnpåfil

### Commiting:

git add –all, eller stor A, eller bare .

git commit -m «forklaring på hva som er committa» .Når du velger git commit -m så lagrer vi en save av fila slik den er akkurat nå. Husk å ta med beskrivelse også

git commit -m "Dette er første versjon av programmet, kun hello world nå"

git commit -a -m «denne kommandoen hopper over staging og går rett på commit»

git diff . For å sjekke hva som er forandra:

git rm sletter ei fil (du kan slette den på vanlig måte i mappa også) og

git restore henter den tilbake

### VS-code:

Om du åpner ei fil fra repo kan du velge å åpne hele mappa

git mv «navnpåfil» «nyttnavn» det står for git move og kan for eksempel brukes til å forandre navn på fil som kan commites slik at man vet når den forandra navn

Avsluttet 13.04 19:12 i filmen.

### Log / sammendrag

git log gir en oversikt over alt som har blitt gjort fra jeg oppretta brukernavn og epostadresse til siste oppdatering

du kan også skrive git log –oneline og få et sammendrag

NB! Q eller q er tasten du må presse for å komme ut av log-view!

Du kan make en amend i stedet for å gjøre alt på nytt om det er noen skrivefeil eller noe sånt, men jeg tror du må adde først uansett

Eks. oppdatere kommentar til regnearket jeg oppdaterte:

git add regneark.xlsx

git commit -m «sdfasdf»

git add regneark.xlsx

git commit -m «redigert tekst» --amend (husk dash dash)

### Branches2

For å opprette en ny branch taster vi inn git branch og navn

git branch navnpågreina

git switch navnpågreina gjør at jeg nå befinner meg i den branchen

Opprett en ny branch gå direkte til den branchen:

git switch -c navnpånybranch

oppdater ei fil og commite den i den branchen

git commit -a -m «beskrivelse av forandringer»

git switch main (switched to branch ‘main’)

git merge -m «merger det som skjedde i navnpågreina» navnpågreina

Nå er forandringene også med i hovedbranchen, dermed kan den greina slettes:

git branch -d navnpågreina Trykk enter og branchen er slettet

### Konflikt om ei fil har blitt oppdatert i både main og den nye branchen

Om jeg prøver med merge da får jeg bare opp feilmelding. Om jeg åpner fila i filutforsker ser jeg hvordan fila ser ut opprinnelig og hva som er nytt – slett det jeg ikke vil ha, lagre og prøv å commite den på nytt igjen.

Avsluttet 32:18

Github

Etter å ha opprettet konto kan du enten starte ny repository eller koble til en eksisterende mappe.

git remote add origin <https://www.w3schools.com/git/git_new_files.asp?remote=github> 36:36 i filmen

git branch -M main

git push -u origin main