Vejledning til github

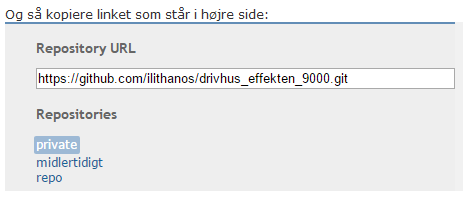
Denne vejledning skulle gerne giver jer mulige for at får en bedre forståelse af git, samt gøre der pisse nemt at dele file med hinanden 😊 guiden jeg er lagt ved brugt jeg på 3 semester hvor den blev skrevet af en af mine gruppe medlemmer, som havde rigtig godt styr på det. Der for kan guiden nok nogen steder være lidt overkil med information, så vi ikke skal bruge.

1. Cloning a Repository

Som det første når man skal til at arbejde på projekt, som deles på et git repository. Er at få det klonet så man har det lokalt.

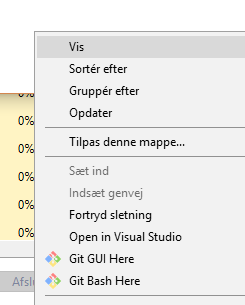
Dette gøres ved først at gå ind på repository-fanebladet: !http://redmine.t-follmann.dk/projects/drivhus-effekten-3000/repository/revisions/master/raw/Pictures\_for\_redmine/GitWikiSetup/repo\_tab.PNG!

Og så kopiere linket som står i højre side:



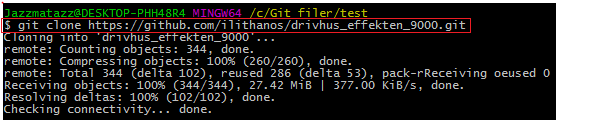
Lav nu en ny mappe det sted hvor du ønsker at projekt repo'et lokalt. Et forslag er at lave en mappe på C-drevet kaldet fx. Git filer, så kan du også gemme, andre repo'er der i fremtiden og holde orden på de ting som er delt henover git.

Nå mappen er oprettet, højerklik nu på denne mappe og vælg:

Git Bash here:

Nu for du så git terminalen frem, og du kopierer nu repo-linket fra før ind i terminalen ved at trykke på @Shift + Insert@, så det står

Sådan her ”git clone” [her skrives repo linket som du kopierede før]\*, og trykker så på enter. Så skulle det gerne se ca. sådan her ud:



Nu har du så repoet liggende lokalt og det du fremover laver ligger kun lokalt indtil du har "commited" dine ændringer, og "pushed" dem til git.

1. Working with the repository - Basic Linux commands and little extra.

Nu kan vi så begynde at arbejde med repo'et. Du kan lave mapper, tilføje filer til mapperne, flytte rundt på filerne, og omdøbe dem derefter, præcis som du plejer igennem windows normale GUI. Men du kan også gøre det via. terminalen, jeg vil vende til disse ting senere.

Men først tager vi de mest essentielle commands som i for brug for når i arbejder med git.



Når i starter på en ny arbejdsdag eller arbejder videre efter et par timers pause. Så start med at "pull" filerne fra clouden.

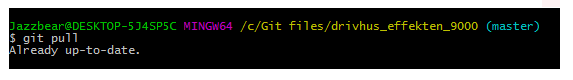
Det kan være der er filer essentielle for at du kan fortsætte med din opgave, eller der er lavet ændringer i forhold til en opgave.

Dette gør du ved commanden ”git pull”, da det er et private repository bliver formentlig bedt om at skrive dit bruger navn og password.

Disse er de samme som dem du bruger på \*github.com\*. Nu du har indtastet disse skulle den gerne hente de nye/ændrede filer,

og merge det med de ting du har liggende, med mindre der ikke er nogen nye/ændrede filer så får du beskeden:

"already up to date". Nu er du så klar til at fortsætte arbejdet.





Når du så har lavet et stykke arbejde og mener det er færdigt og ønsker at "pushe" det til repo'et i clouden.

Så skal du først fortælle git at den skal tilføje og "tracke" de filer du har tilføjet til repoet lokalt.

Det gør du ved brug af \*git add\* og så foræller du hvilke filer der skal med, skrevet i følgende syntax:



så tilføjer git den ene fil du ønsker skal med.

Men i de fleste tilfælde så er det jo kun de filer, som du havde tænkt dig skulle med i projektet, som du smidder ind i mapperne på dit lokale repo. Så i langt de fleste tilfælde, kan spare dig en masse besvær, og gøre det lidt nemt med en af de 2 følgende commands:



Denne command "add'er" alle nye/ændrede filer, som står i den mappe som du i terminalen befinder dig i, samt alle filer og mapper som ligger i underligende mapper, fra din postion. Med mindre du har flyttet dig rundt, med change directory commanden så står du altid der hvor du "bashed" ind i git i starten.



Er nemmere og mere overordnet MEN! Vigtigt!!! Jo nemmere det er, desto mindre har du også selv styr over hvad der egentlig sker. Så det kræver at du selv holder styr på hvilke filer du har flyttet ind i diverse mapper osv. Så kort sagt den tilføjer ALT! nyt i jeres lokale repo.

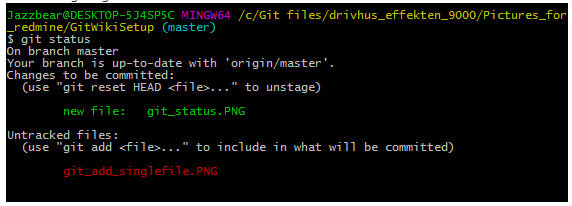


Hvis man kommer til at tilføje filer ved et uheld med \*@git add\*@\*, så kan man behændigt omgøre (undo) denne beslutning med git reset \*.  
man må dog under ingen omstændigheder lave commit, før man laver reset! (da reset i så fald ingen virkning har)



Når i har addet de filer i ønsker at gemme til repoet, så er det en god ide lige at tjekke om alt er som det skal være, er de rigtige filer blevet ændret/oprettet og tilføjet (add'et) til gits file tracker. Det gør i med commanden: git status

Så får i noget lignende det her:

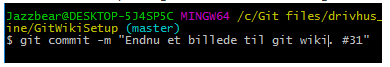


Når i så mener i færdige med det i har lavet, og vil til at "pushe" det til clouden, så skal i først "commite" det i har lavet. Det er på sin vis den måde i fortæller git, "Jeg ønsker at gemme det jeg har lavet nu", så opretter git en ny virtuel version af projektet, medfølgende et unikt branch-id, så vis i laver ged i noget senere.

Så kan vi gå tilbage til et tidligere stadie hvor der er mindre "ged" i projektet.



Måden i commiter på er: \*git commit -m "skriv noget her" så det ser sådan her ud:





Nu er i så klar til at smide det op på det online clouden, det gøres med commanden: git push.

Så for at summere op. Når skal ligge filer op, så bruger i følgende række følge af commandore:

git pull

git add -A eller git add \*

git commit -m ”teskt med kommentar til hvad i har laver eller ændret”

git push

Når skal til at lave noget projekt arbejde så sørger for altid at have ”pullede” før i går i gang med det, så chancen for at der kunne op står merge fejl mindre. Der er også vigtig at når i færdig og gemt med at lave noget så lukke programmet som i har brugt, ellers kan den også godt brokke sig en del.

Hvis der sprøgsmål udvidet information så kan jeg godt skaffe det 😊