## Hvordan bruke Github sammen:

Det har allerede vært en lang leksjon om bruk av verktøy for å jobbe med Git samt hvordan bruke Git. Her er en annen oppskrift for deg som vil bruke terminalen / bash.

### **Setup:**

1. En person i gruppen oppretter et **private** repository i Github ([link](https://docs.github.com/en/github/getting-started-with-github/create-a-repo), [link](https://docs.github.com/en/github/administering-a-repository/setting-repository-visibility)) og inviterer sine gruppedeltakere
2. Samme person lager en mappe på sin maskin hvor koden til eksamen skal være
3. Samme person åpner VSCode og åpner den nye mappen i VSCode
4. Samme person henter ned starter-templaten fra BASE ([link](https://github.com/mariuswallin/webapp_eksamen_21)) (husk punktum etter urlen nedenfor)
   1. Git clone <https://github.com/mariuswallin/webapp_eksamen_21> .
5. Samme person sletter gammelt GIT repo via terminalen.
   1. Kommandoen må skrives der .git ligger.
      1. Enten i root-mappen (den du laget i steg 2) eller mappen\_du\_laget/webapp\_eksamen\_21 (hvis du ikke husket punktum i steg 4)
   2. rm -rf .git
6. Samme person rydder bort det som ikke trengs i starter-templaten
7. Samme person følger oppskriften Github har gitt i punkt 1
8. Samme person sjekker at koden har kommet til Github
9. Samme person lager seg en ny branch (erstatt student\_navn med ditt fornavn)
   1. git checkout -b student\_navn
10. De andre i gruppen kloner det nye prosjektet som er laget (husk punktum)
    1. git clone DERES\_URL\_FRA\_GIT .
11. De andre i gruppen lager sin egen branch (erstatt student\_navn med ditt fornavn)
    1. git checkout -b student\_navn
12. Lag en test komponent og commit denne til github. Github kommer til å anbefale en git kommando du må bruke for å legge til din branch og kode i GitHub
13. Sjekk at du finner igjen din branch og kode i Github

### **Lagre det du har gjort i Github**

Bør gjøres ofte og ihvertfall før du avslutter for dagen. Sørger for at du har sikkerhetskopi av koden din.

1. git add .
   1. “Stager” alle filer slik at de er klare til å bli sendt
2. git commit -m “Beskjed”
   1. Lager en beskjed som du kan se etter når du skal finne igjen endringene i ettertid
3. git push
   1. Sender koden til Github

### **Merge din kode inn til master / main:**

Gjør dette når du vil at hovedbranchen (der alt sammen skal flettes sammen) skal inneholde dine endringer.

1. Gå ut av den branchen du står i og inn i master (eller main)
   1. git checkout master (eller main)
2. Sørg for at du har fått alle nye endringer
   1. git fetch
   2. git pull
3. Merge din branch inn i master
   1. git merge navn\_på\_branch
   2. Løs eventuelle konflikter

### **Merge en annen sin kode inn i din:**

Gjør dette når du trenger en medstudents kode. Før du kan gjøre dette må du commite din kode.

1. Gå ut av den branchen du står i og inn i student\_navn sin branch (erstatt student\_navn med fornavnet til den du jobber sammen med)
   1. git checkout student\_navn
2. Sørg for at du har fått alle nye endringer
   1. git fetch
   2. git pull
3. Gå tilbake til din branch
   1. git checkout -
4. Merge din student sin branch inn i din (erstatt student\_navn med fornavnet til den du jobber sammen med)
   1. git merge student\_navn
   2. Løs eventuelle konflikter