Dokumentationsrapport

https://JakobEsp.github.io/Mit-CV-site/

Indhold

[Oprettelse af repository 3](#_Toc112323503)

[Git spørgsmål 4](#_Toc112323504)

[Hjemmeside proces 6](#_Toc112323505)

[Logbog og opgaveforløb 7](#_Toc112323506)

[17/8 7](#_Toc112323507)

[18/8 8](#_Toc112323508)

[19/8 9](#_Toc112323509)

[22/8 9](#_Toc112323510)

[23/8 9](#_Toc112323511)

[24/8 9](#_Toc112323512)

[25/8 9](#_Toc112323513)

# Oprettelse af repository

**For at oprette et repository på min pc brugte jeg følgende linjer kode/kommandoer**

Cd + fil placereing;

Git init; opretter repositoriet localt i valgte mappe.

Touch ”index.html”; opretter html fil i valgt mappe.

Git status; Viser ændringer i repo.

Git add index.html; tilføjer filen til staging.

Git commit –m ’kommentar’; tilføjer ændringer til Main/Master.

**Tilføjer repository til Github**

git remote add origin <https://github.com/JakobEsp/Mit-CV-site.git>;

git branch –M main;

git push -u origin main;

koden oven over får først fat på lokation på github. Derefter laver den laver den main branch og skubber den til github.

## Git spørgsmål

1. Du skal forklar hvorfor git findes?

* git blev lavet for bedre versions håndtering både på større og mindre projekter, sådan at alle includeret kunne arbejde på samme produkt samme steder i koden på samme tid.

1. Hvad gør git og hvorfor er det brugbart?

* Git holder øje med alle opdateringer og ændringer i et repository. Og det gør det muligt at arbejde i branches. Så hvis man opdatere en branch eller merger ind i Main/master og der er et problem. Kan man gå en version tilbage og fejlfinde uden at arbejde direkte på main branch.

1. Hvordan bruger man git på et projekt, når man er flere udvikler?

* Så bruger man en tjeneste som fx github for at tilføje collaborators. Hvilket gør det muligt for dem også at lave deres egen branch og arbejde på branchen i repo’et.

1. Når vi arbejder med git, hvad betyder det så, når vi har et projekt eller Repository?

* Et repository er den gruppe filer som git tracker og holder øje med.

1. Hvordan arbejder git?

* Git tracker filer ud fra deres SHA-1 hashes. Ellers referere den til metadata til filerne og tracker det.

1. Hvad er et Commit?

* Et commit er måden man sender stagede filer til den nuværende version af en branch på enheden man arbejder på.

1. Hvad er en Branching og Hvad er Master/Main?

* Maser/Main siger lidt sig selv, main/hoved branchen er der alt det man arbejde på bliver pushed eller merged til når der er en ny addition til repositoriet.
* I en ekstern branch arbejder man som reelt med nye features som senere når de er færdige kan blive merged ind til main branch.

1. Hvad er Merging?

* Mergning er når man sammensætter to branches i sit repository. Som regelt er det en ekstern branch der bliver merged ind til main branch.

1. Hvad er git Clone?

* git clone er at tage en kopi af et repository fra en anden kilde som fx github og downloade den ned på din egen enhed.

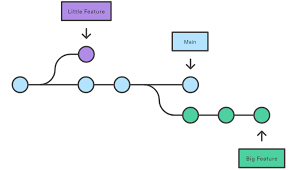
1. Hvorfor bruger man som udvikler versionskontrol ->> git?

* Sådan at alle på projektet kan arbejde på samme projekt med den samme Master/main version til rådighed, i at branche ud fra. Det gør at man ikke bliver nødt til at manuelt sende filer og kode til hinanden, og sammensætte det på en enhed. Det sammensætter selv filer og holder øje med ændringer.

1. Forklar om Developer Workflow med git – hvordan det forløber?

* Efter oprettelse af repository lokalt og på github er der en række commands jeg jeg har brugt mest i mit workflow i forbindelse med denne opgave.
* Git add –p
  + for at se ændrede filer og hvad der er blevet ændret, derefter et y for ja til at add til staging.
* Git add filnavn
  + Tilføjer en bestemt fil til staging
* Git add –A
  + Tilføjer alle ændrede filer til staging
* Git commit –m ’besked notat’
  + Commiter alle stagede filer til det lokale repository.
* Git status
  + Viser status på redigerede filer (om de er staged eller ikke)
* git push origin
  + Den skubber commitede ændringer til den branch på github man arbejder på. Man kan også bruge git push –u origin main, men der skulle den gerne skubbe ændringerne til main branch og ikke den branch man sidder på.
* Git checkout –b branchnavn
  + Skifter til en anden branch i repositoriet.
* Git merge branchnavn
  + Merger den givne branch til den man er i.

1. Vis visuelt hvordan Master og flere branchs vil se ud for et givet projekt.



# Hjemmeside proces

# Logbog og opgaveforløb

## 17/8

Startede med at logge ind på diverse programmer på min pc.

Jeg har set videoerne & prøvet det af selv:

- Git Tutorial Part 1: What is Version Control?

- Git Tutorial Part 2: Vocab (Repo, Staging, Commit, Push, Pull)

- Git Tutorial Part 3: Installation, Command-line & Clone

- Git Tutorial Part 4: GitHub (Pushing to a Server)

Git noter af min nuværende forståelse af koden.

” cd + stifinder vej til mappe” – placere git der

”pwd” – viser hvor git er i dine filer fx /c/Opgaver

”mkdir” – makedirectory

”Touch + navn på fil man vil oprette.filtype” - opret fil i repository

”git status” – viser status på filer. Om de der er foretaget ændringer på, er staged eller ikke.

”git add + navn på fil der skal stages” – stager fil så den kan commites

”git commit –m + ’kommentar’” – commiter stagede filer til repository

”git checkout + <navn på branch>” – hopper til anden branch

”git clone + link til github repository” – cloner repository fra online server.

”git remote set-url + url på git” – gør den valgte destination til indholdet i set git

”Git remote –v” viser url på det repository destinationen tilhøre

”git add –A” tilføjer alle ændringer til staging

## 18/8

Først lavede jeg overskrifter og indholdsfortegnelse på min Rapport.

Derefter gik jeg i gang med at øve mig og få bedre forståelse for Git, ved at lave de 3 første niveauer af –

<https://learngitbranching.js.org/>

Efter har jeg siddet of læst lidt Html kode for at prøve at danne en grundforståelse da jeg ikke har arbejdet med det før. Og læst lidt op på Bootstrap.

Jeg er blevet enig med mig selv om at få opsat mit repo før jeg begynder på hjemmeside og bootstrap.

Git notater.

Git branch; viser hvilken branch man er i.

Git checkout –b <navn på branch>-branch; laver en ny branch

Git merge <navn på branch>-branch; merge til den branch man er på.(husk at være på main branch.

Git push –u origin main; skubber nuværende commit til github.

Derefter har jeg eksperimenteret lidt med git.

Til sidst har jeg set de første 6 afsnit af ”Bootstrap 5 Crash course” imens jeg har prøvet dele af det af inde i mit program.

### 19/8

De første 2 timer her jeg siddet og øvet mig på html & en smule bootstrap på w3schools.

Op til pausen kl 11. har jeg fået lavet layout på min startside på min hjemmeside.

Det sidste af dagen sad vi på lærepladsen.dk

### 22/8

Om formiddagen var lærepladsen.dk nede, så jeg brugte tiden på at lave et nyt CV. Da det i starten af eftermiddagen kom online igen fik jeg researched en masse læresteder jeg har tænkt mig at søge i fremtiden.

### 23/8

Op til kl 9 har jeg gjort sådan at mine knapper har en destination, opsat github på min desktop og oprettet 3 nye html dokumenter som hver skal være en side på mit CV.

Op indtil kl 11 har jeg siddet og arbejdet på mit card på resume siden. Har haft lidt udfordringer med position & billede destination. Positionen viste det sig at jeg havde skrevet min container klasse ind forkert. Og billedet prøvede jeg i starten at hive fra en mappe i mit repo. Det kunne jeg godt gjorde det bare forkert.

Det sidste af dagen har jeg lavet layout til alle siderne i mit CV.

### 24/8

Det første af dagen brugte jeg på at importere tekst til min hjemmeside fra mit CV. Derefter brugte jeg resten af tiden på at se på lærepladser og sende ansøgning.

### 25/8

Indtil kl 9 lavede jeg opdatering på side knapperne på min side. Da jeg lagde mærke til at knapperne ikke rykkede sig med siden. Løsningen var at putte knappen i sin egen <div> og bruge klassen ”fixed-bottom”.

Op til kl 11 svarede jeg på spørgsmålene til git.

Så lavede jeg de git spørgsmål og begyndte at importere mere af mit CV fra word til hjemmesiden.