

GitHub og Brackets

Hvordan vi kan lave versionsstyring fra Brackets

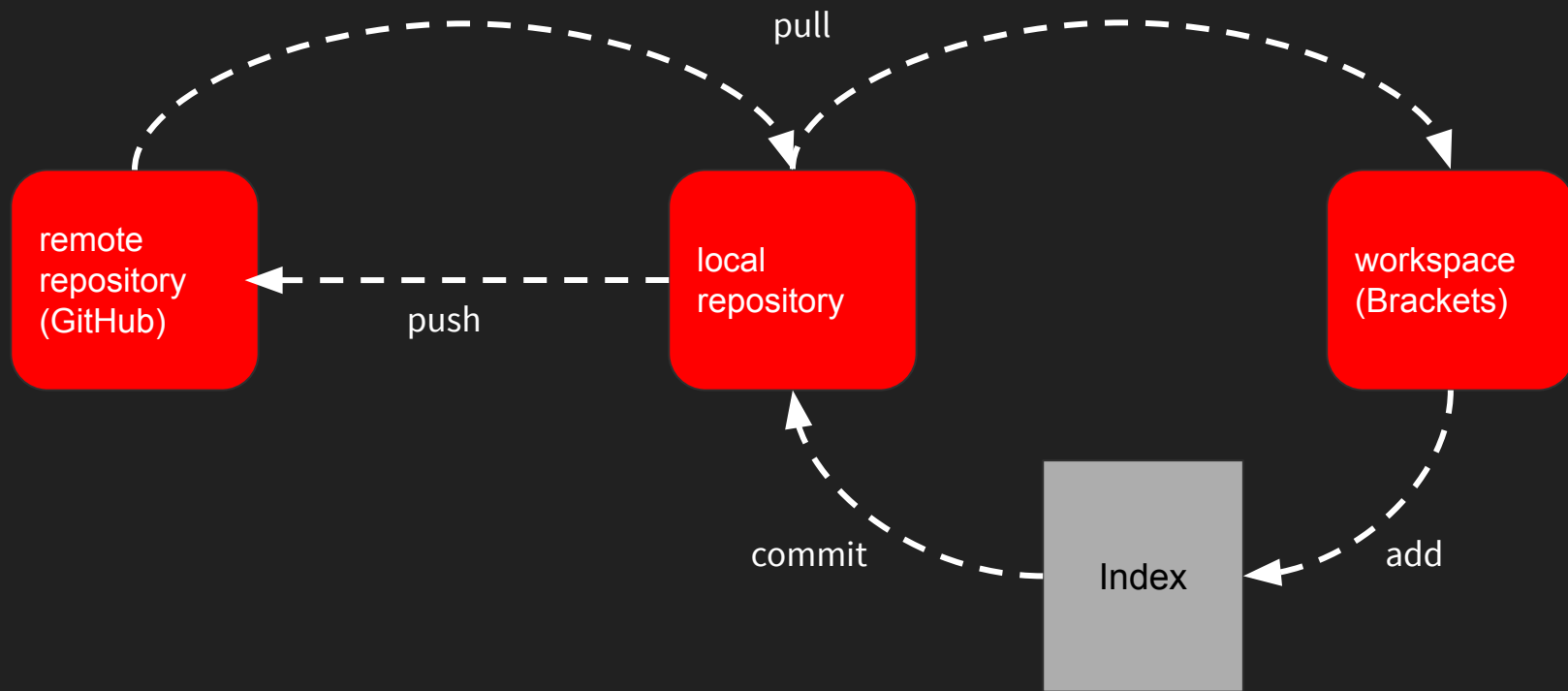
Git - et recap

Hvad er versionsstyring?

Hvorfor lave versionsstyring?

Hvordan fungerer Git / Github

Git basis flow



GitHub - terminologi #1

Repository (repo): a repo tracks all the changes you made in that directory. In tangible form, it is a .git/ folder that lives inside the directory.

Branch: independent line of development typically used to develop new features without disturbing the rest of the repository. Delete branch when it is no longer needed!
The default branch is called the “**Master**” branch.

Fork: within the context of GitHub, to fork is to copy someone’s GitHub repository and put it in your GitHub account. Under the hood, it is just cloning their repository.

Clone: the concept of a “clone” just means to copy the entire repository.

<https://gitbookdown.site/index.html>

GitHub - terminologi #2

Commit: a “commit” adds your latest changes to your repository. If a change is not committed, Git would not recognize the change.

Push: upload your commits to a remote repository (like GitHub!)

Pull: fetches the changes that is in your remote repository GitHub.

<https://gitbookdown.site/index.html>

Github og Brackets workflow

1. Opret Repo på gitHub
2. Opret lokal mappe
3. Clon Repo via Brackets
4. Opret Branch
5. Rediger filer
6. Commit
7. Merge ind i Master - slet evt. branch
8. Commit
9. Push

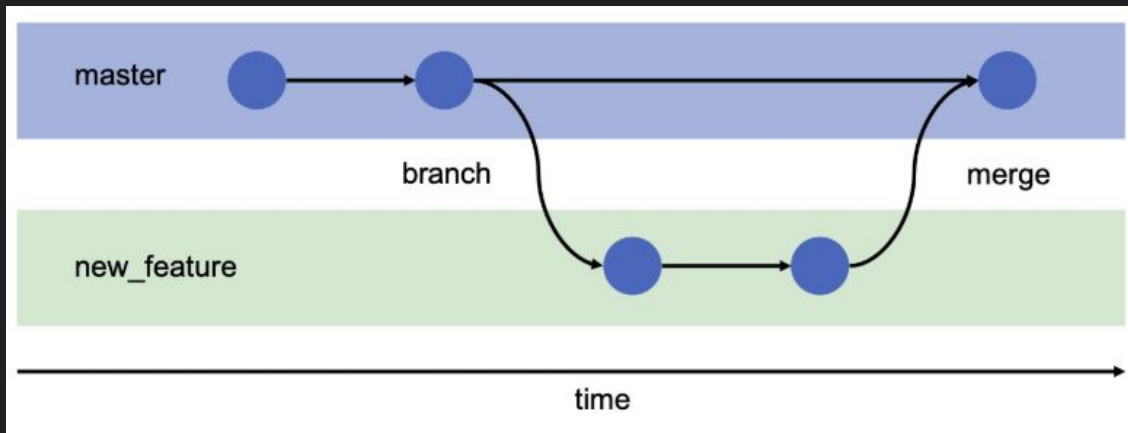
Github og Brackets workflow - hvis du allerede har lavet et lokalt repository (init)

1. (init Repo lokalt- Brackets)
2. Opret repo på GitHub - UDEN README.MD og .gitignore!
3. *Create New Remote* i Brackets med det nye github repo som Remote
4. Commit
5. Push

Nu kan du arbejde som vanligt...

Branches

- Hvad er en Branch?
 - **Branch:** independent line of development typically used to develop new features without disturbing the rest of the repository. The default branch is called the “**Master**” branch.
- Hvordan skal de bruges?
- Merging Branches



Litteratur

Rigtig god manual og tutorials:

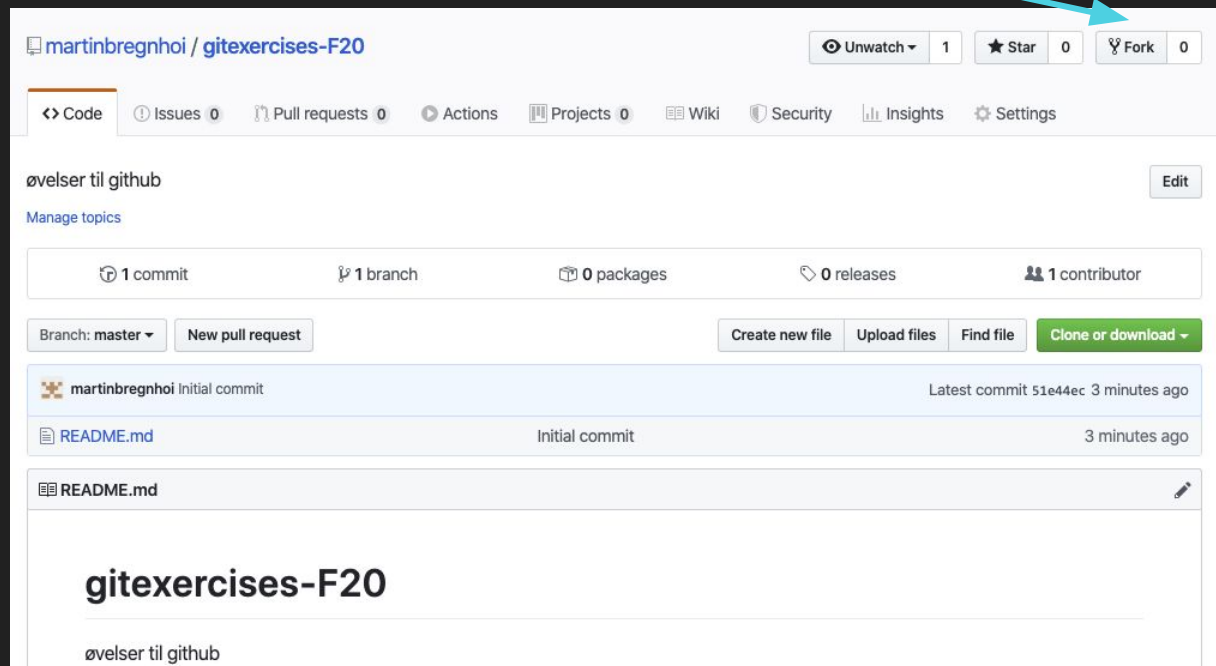
<https://practicalseries.com/1002-vcs/index.html>

Github og Brackets video (made for you by me):

<https://youtu.be/UKF0VhwhfYuY>

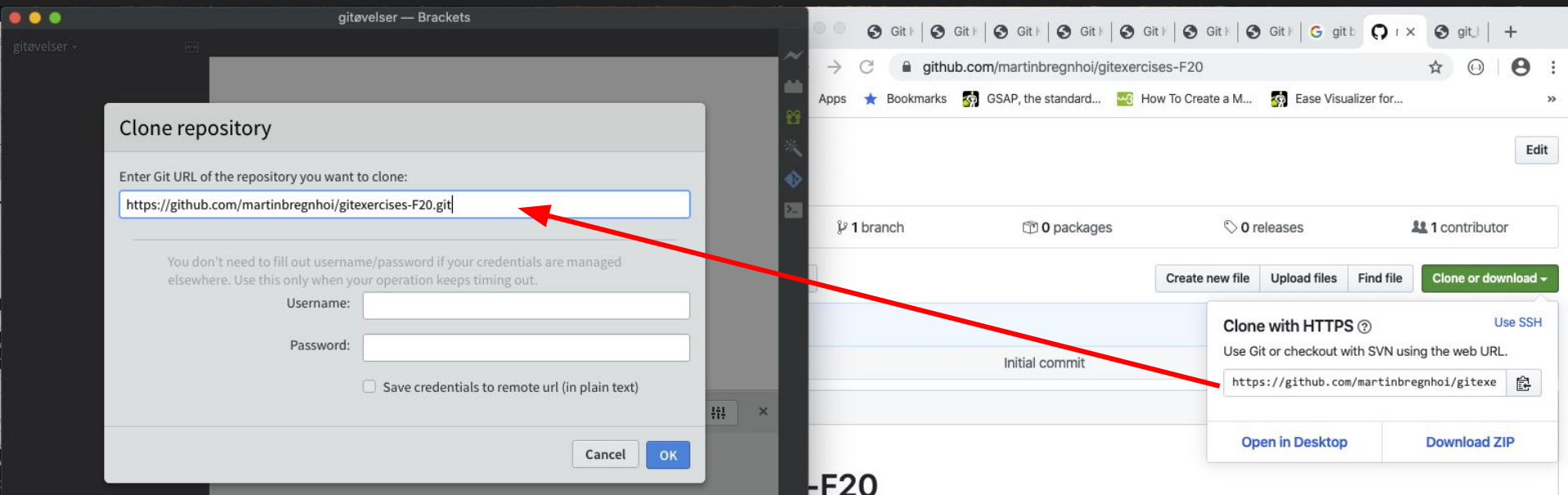
Øvelse 1 - Fork repository

gå til: <https://github.com/martinbregnhoi/gitexercises-F20> og Fork repo til din egen GitHubkonto



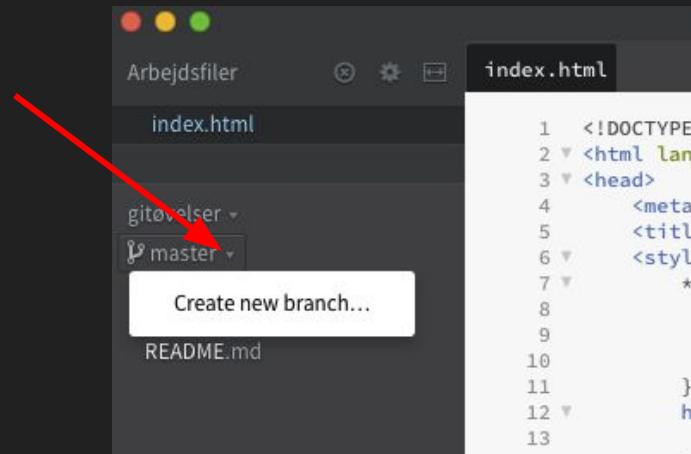
Øvelse 2 - Clon repository i Brackets

1. Åbn Brackets
2. Opret en ny tom mappe på din computer (ikke på skrivebordet!)
3. I Brackets: Clon repository'et fra øvelse 1 - indsæt Username og Password



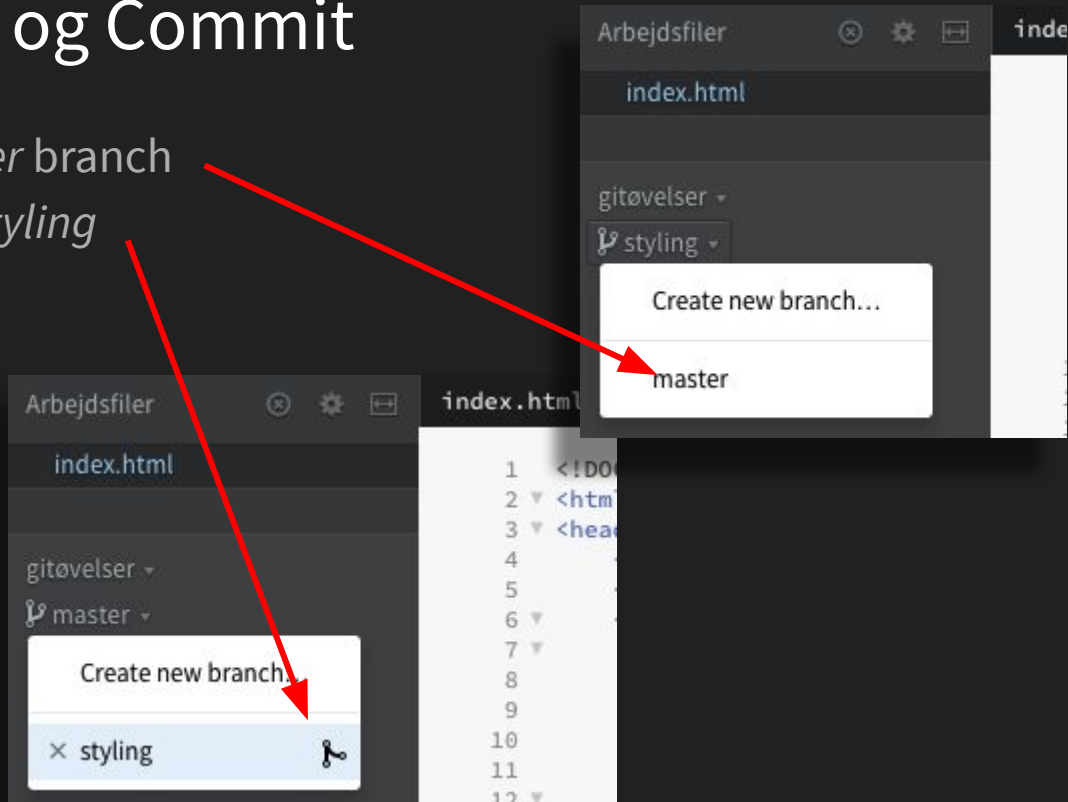
Øvelse 3 - Opret Branch og ret filer

1. Opret en ny Branch "*styling*" til eksperiment med styling
2. Tilføj valgfri styling til *header* og *footer*
3. Gem og Commit - husk fornuftig Commit message!



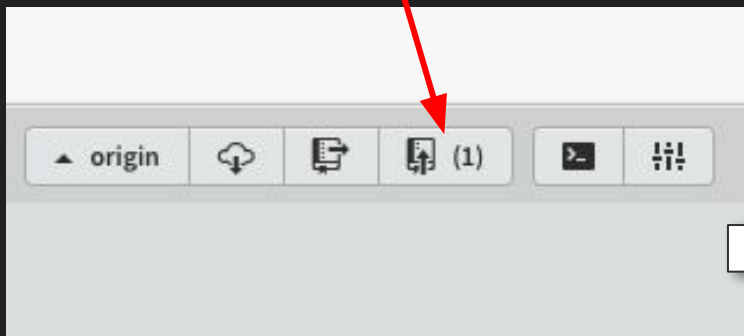
Øvelse 4 - Merge og Commit

1. Skift tilbage til *master* branch
2. Merge *master* med *styling*



Øvelse 5 - Push

1. Commit og Push til GitHub



Push to remote — origin

Current tracking branch: **origin/master**

Action: ☒ Push to current tracking branch
☐ Push to another branch

☒ Default push
☐ Forced push
☐ Delete remote branch

☐ Send tags

You don't need to fill out username/password if your credentials are managed elsewhere. Use this only when your operation keeps timing out.

Username:

Password:

☐ Save credentials to remote url (in plain text)

Cancel OK

Øvelse 7 - mere Merge og Commit

1. Skift tilbage til *styling* branch
2. flyt css til selvstændigt css dokument
 - a. opret css fil
 - b. klip al css indenfor style tag og indsæt i den nye css fil
 - c. lav link til css fil i index.html
 - d. test at det virker
3. gem og Commit
4. skift til *master* branch, og merge ændringer
5. commit og push til GitHub
6. slet *styling* branch

Github for teams - Samarbejds strategi

- Opret et nyt Repository til hvert nyt projekt
 - Tilføj gruppemedlemmer som Collaborators. Det gøres under *Settings > Collaborators*
 - Hvert gruppemedlem laver en tom mappe UDENFOR de mapper der i forvejen er knyttet til Github - det er vigtigt for at kunne knytte projektet til det nye Repository.
 - Hvert gruppemedlem kloner det nye projekt repository.
 - Hvert gruppemedlem laver en BRANCHE som de arbejder i.
 - Før man pusher - hent (pull) Master branchen, og merge egen branch ind i denne. Vigtigt!
 - Push Master branch tilbage til Github
-
- Aftal hvem der arbejder på hvilke filer, så i undgår at arbejde på de samme. **DETTE SKABER MERGE KONFLIKTER, og skal undgås!!!**
 - Commit og push når du har lavet nogle ændringer i en fil
 - Pull før du starter, så du er sikker på at have den nyeste versioner af projektet!!!

Øvelse 8 - gruppeøvelse

1. Arbejd sammen to og to (A - B)
2. Opret et nyt repo på den enes GitHub. Den anden skal oprettes som collaborator
3. Clon begge det nye repo
4. A opretter et html dokument i master branch. Gem, commit, push
5. B - henter (pull) de nyeste filer
6. B - opret en ny Branch *tests* - og lav nogle ændringer
7. skift til *master* branch, og merge ændringer
8. commit og push til GitHub
9. A gør det samme: pull > opret branch > rediger > skift tilbage til master > merge, commit, push

Øvelse 9 - gruppeøvelse

Arbejd videre fra øvelse 8. Prøv at se hvad der sker hvis i arbejder på den samme fil samtidig.

1. hent (pull) de nyeste filer
2. skift til *test* branch - og lav nogle ændringer
3. skift til *master* branch, og merge ændringer
4. tjek resultatet
5. undersøg hvad der skal til for at undgå mergekonflikter