

# De vil give en GIT workshop

**Men de har ingen undervisere til at gøre det**

**Nu står vi her**

---

## Nemt at komme i gang

- Downloade git:
    - <https://git-scm.com/>
  - Downloade github desktop:
    - <https://desktop.github.com/>
  - GIT Bogen
    - <https://git-scm.com/book/en/v2>
- 

## git - a short intro

### GOALS

- 1. Create and share repositories**
  - 2. Know the 5 basic commands**
  - 3. Know how these translate to a GUI**
  - 4. Resolve a merge conflict**
  - 5. Branching**
- 

## Creating and ignoring

**.gitignore is your best friend!**

```
*.o  
*.d  
*.DS_store
```

## lets create a repository

---

## git commands

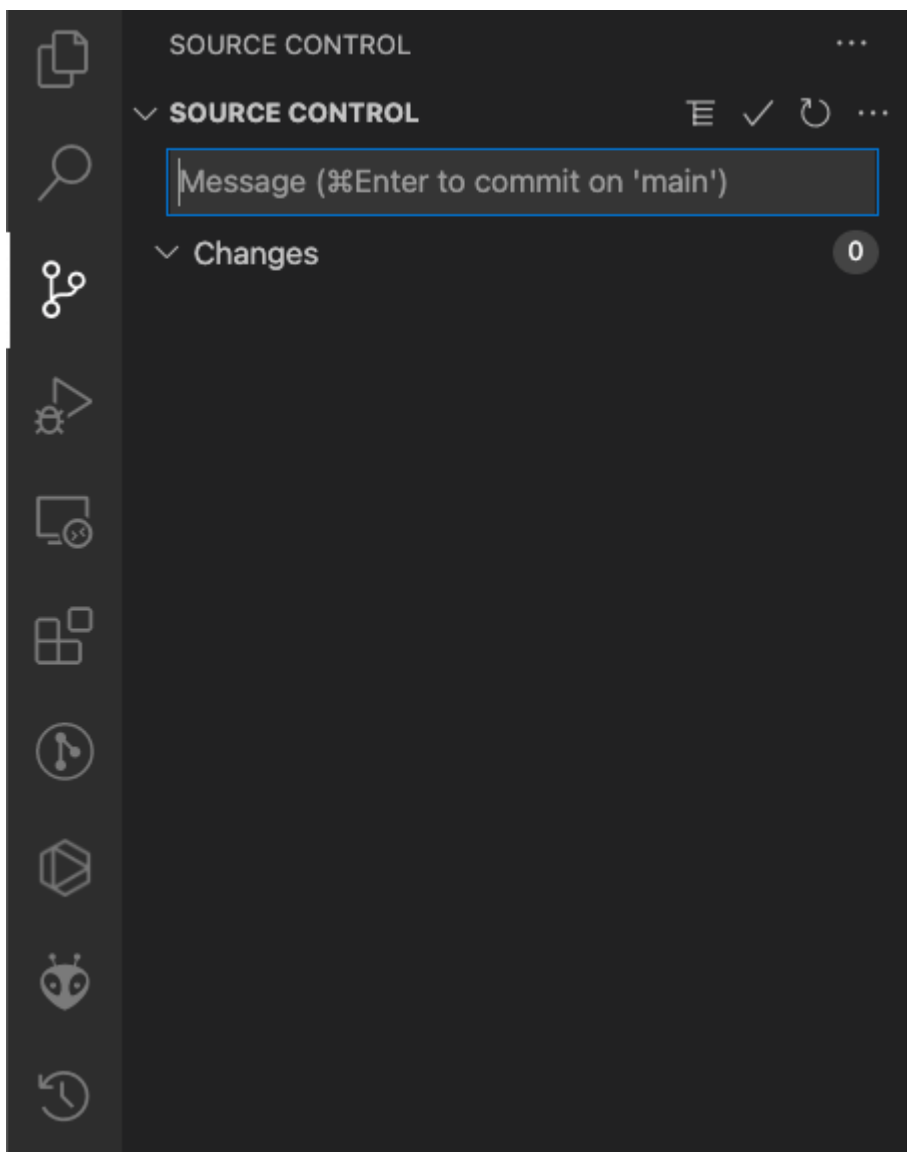
1. git clone <link>- clones a repository to a local repository on your machine
  2. git status - shows the current status of your repository
  3. git add -A add all untracked files and changes in your repository
  4. git commit -m "commit message" - commits your changes to the repository
  5. git pull - pulls the changes from origin to your
  6. git push - pushes your changes to the origin
- 

## Lets demo that!

---

## A GUI for all your machines

### Visual studio code!



---

## Lets try something

1. Brug github. Lav et repo på git hub.

Clone det fra terminalen.

Ændre en fil i repoet på din computer.

Commit det fra terminalen.

Push det fra terminalen.

det er din basis for at kunne bruge git alle steder altid.

hvis du sidder på windows skal du installere git.

Og bruge programmet "git bash" til overstående.

---

2. efter overstående, Overvej, i hvilket environment du skal arbejde mest i.

Er det visual studio, vs code, Psoc creator eller noget andet?

I visual og vs code er der integrated løsninger.

Når du opretter et repo på github kan du vælge en designet til visual studio.

Lav en .gitignore som for os minimum skal håndtere .gitignore fil til c eller c++

I psoc er det smartest at bruge github desktop mender jeg.

der er notorisk mange gennerede filer.

Jeg har en .gitignore fil hvis du er intresseret.

gennem løb øvelse 1 i dit valgte enviornment

---

3. Merge konflikts.

Jeg anbefaler at du enten selv laver en merge konflikt,  
eller sætter dig med en makker og fucker med nogle filer for hinanden.

---

4. Branching

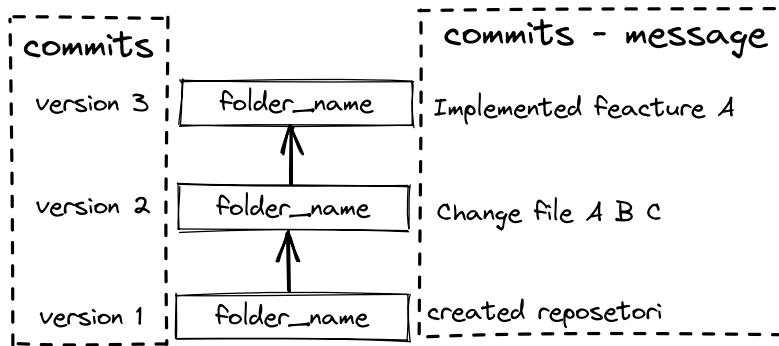
I er selv i stand til at finde frem til hvordan man brancher

hints:

1. git branch <new branch name> brancher ud fra nu værende.
  2. git checkout <branch name> skifter til anden brach
- 

## Hvad er git?

1. en folder, som er vertions styret.
2. en folder heder et repo
3. du har din egen version af repoet på din maskine
4. Du kan synce den til remote repoet, repoet i skyen.
5. Så dan kan man holde alting assure.



## .gitignore som løser de fleste problemer

```
# Prerequisites
*.d

# Compiled Object files
*.slo
*.lo
*.o
*.obj

# Precompiled Headers
*.gch
*.pch

# Compiled Dynamic libraries
*.so
*.dylib
*.dll

# Fortran module files
*.mod
*.smod

# Compiled Static libraries
*.lai
*.la
*.a
*.lib

# Executables
*.exe
```

```
*.out
*.app

# DS_Store files
.DS_Store

# other stange temps
~$*

# obsidian
*.obsidian

### PSoCCreator ###
# Modified from https://www.gitignore.io/?templates=psoccreator
# You may need to add additional directories, e.g. CortexM3/

# Project Settings
*.cywrk.*
*.cyprj.*

# Generated Assets and Resources
Debug/
Release/
Export/
CortexM0/
CortexM3/
codegentemp
Generated_Source
*_datasheet.pdf
*_timing.html
*.cycdx
*.cyfit
*.rpt
*.svd
*.log
*.zip
#.swp
*._*
*.DS_Store

# clang tidy and format
*.format
```

\*.tidy