

○ Før vi starter...

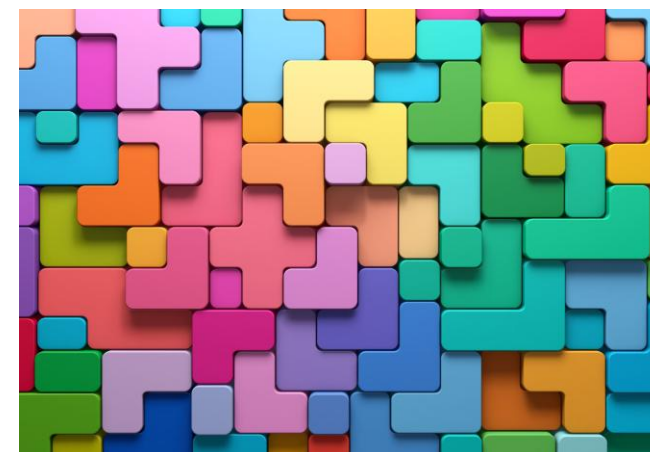
- Logg inn på github.uio.no med UiO-bruker og passord (hvis du ikke har gjort det før, eller er usikker)
- Når du har gjort det, vil vi kunne opprette et **repository** for prosjekt 0 til deg
- Dette må være på plass til livekodingen etterpå



GITHUB ER DIN VENN

FORELESNING 2

FREDAG 22/8



○ Læremål: Versjonskontroll og testing

- Bruke GitHub til å holde styr på forskjellige versjoner av programmet (og gå tilbake til noe som fungerte de gangene ting ikke virker eller det bare blir rot)
- Samarbeide uten å ødelegge for hverandre (hver utvikler jobber med sin egen testversjon av moderprogrammet)
- (testing kommer vi tilbake til neste uke)



○ Motivasjon: Hvorfor GitHub

- Du vil ikke miste filene dine hvis harddisken ryker
- Du vil kunne gå tilbake til en versjon av koden som du vet virker hvis en feil har oppstått
- Det er viktig å vite hva som ble endret for hver ny versjon av koden (spesielt når andre bruker koden)
- I et større prosjekt er det også viktig å vite hvem som gjorde hva når
- Og at ikke flere jobber i samme fil samtidig → kaos



○ Github

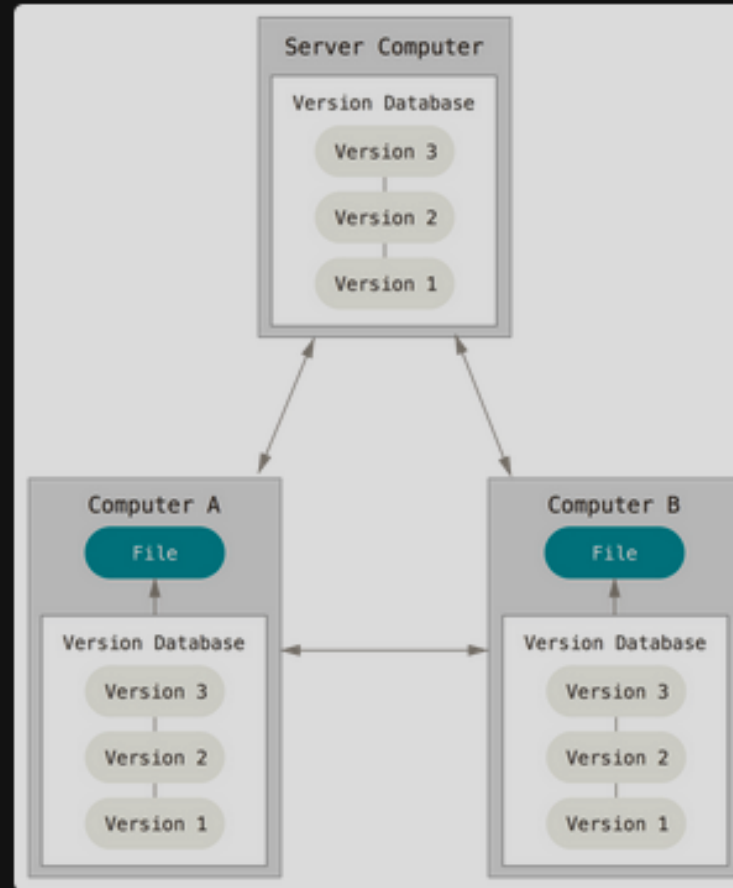


Fig. 13 A distributed VCS. Source: Chacon & Straub, ProGit (2014)

Også nyttig om du jobber på ulike maskiner i løpet av dagen



- GitHub vs. (for eksempel) OneDrive

	GitHub	OneDrive
Enkel backup	Ja	Ja
Versjonskontroll	Ja	Tja...
Laget for koding	Ja	Nei
Forgrening (branching)	Ja	Nei
Dokumentasjons- vennlig	Ja	Nei
Direkte i VS Code	Ja	Nei



○ “Men jeg har GitHub-konto fra før”

- Om du har konto på github.com må du likevel jobbe i UiO sin github-løsning med alle prosjektene
- Dere blir vurdert på hvordan dere har brukt GitHub underveis
- Hvis dere kun har én commit med det ferdige resultatet til github.uio.no blir ikke prosjektet godkjent (vi må se prosessen, ikke bare resultatet)
- **TLDR: Bruk UiO sin GitHub til alt du gjør i IN1910**



○ Hva skjer hvor?

github.uio.no (server)

Din datamaskin

GitHub Desktop

VS Code (+git)

- Alle versjoner av filene (med historikk) er lagret her
- Administrere prosjektet ditt
- Lese dokumentasjon (README)

- Opprette lokal kopi av repository på disken (clone)
- Endre aktiv branch som du jobber i
- Inkludere (stage) endringer til en ny commit
- Lage commit med melding om hva som er endret
- Dytte (push) commit opp til server



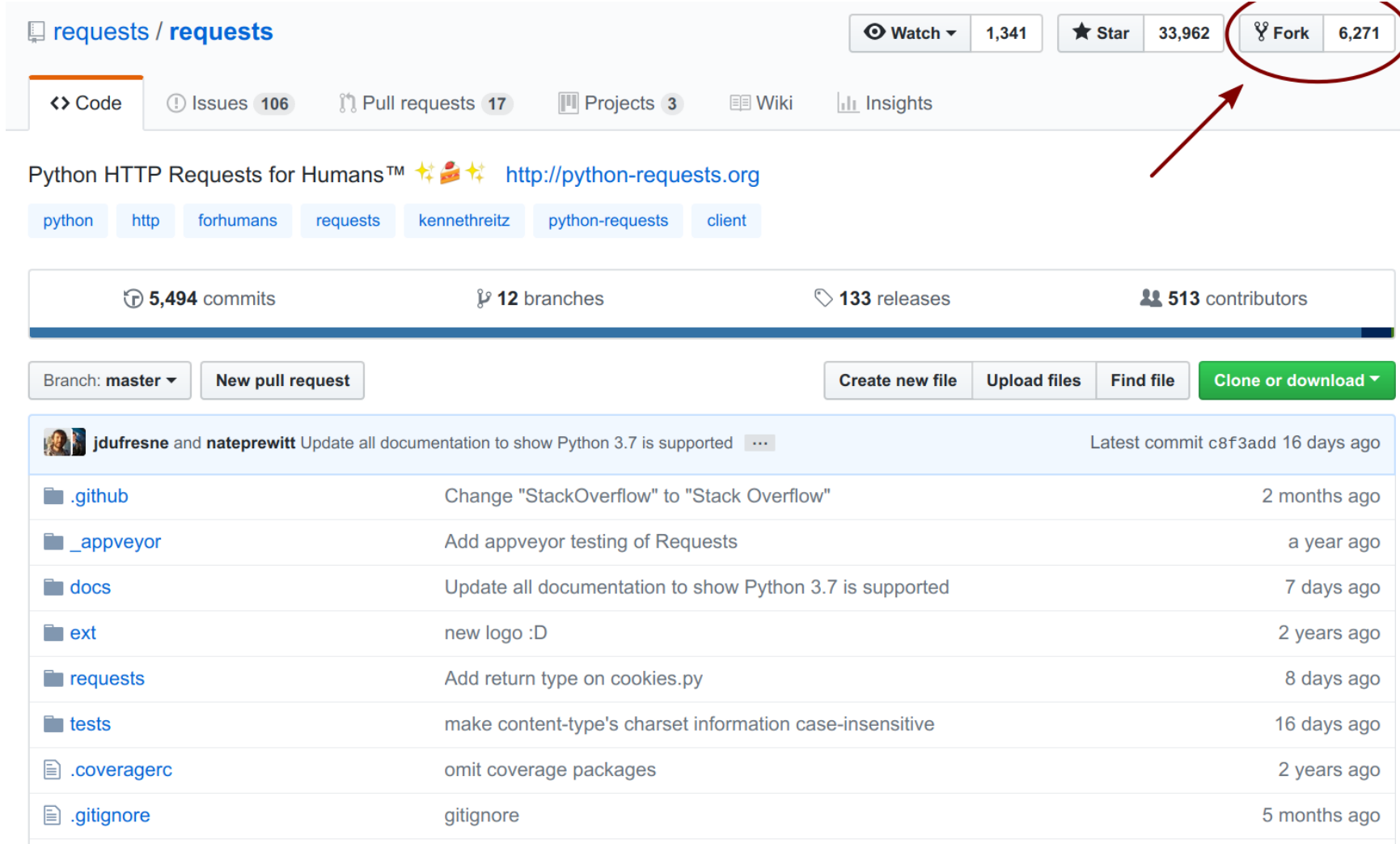
- Når du har skrevet litt kode...



Fig. 16 Source: Chacon & Straub, ProGit (2014)



○ Forking: Lage egen versjon av noen andres repository



The screenshot shows the GitHub repository page for 'requests' by 'requests'. The repository has 1,341 watchers, 33,962 stars, and 6,271 forks. A red circle highlights the 'Fork' button, and a red arrow points to it. Below the repository name, there are tabs for 'Code', 'Issues' (106), 'Pull requests' (17), 'Projects' (3), 'Wiki', and 'Insights'. The repository description is 'Python HTTP Requests for Humans™' with a link to 'http://python-requests.org'. Below the description are tags: 'python', 'http', 'forhumans', 'requests', 'kennethreitz', 'python-requests', and 'client'. The repository statistics show 5,494 commits, 12 branches, 133 releases, and 513 contributors. At the bottom, there is a list of recent commits with their descriptions and timestamps.

requests / requests

Watch 1,341 Star 33,962 Fork 6,271

Code Issues 106 Pull requests 17 Projects 3 Wiki Insights

Python HTTP Requests for Humans™ ✨🍰✨ <http://python-requests.org>

python http forhumans requests kennethreitz python-requests client

5,494 commits 12 branches 133 releases 513 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

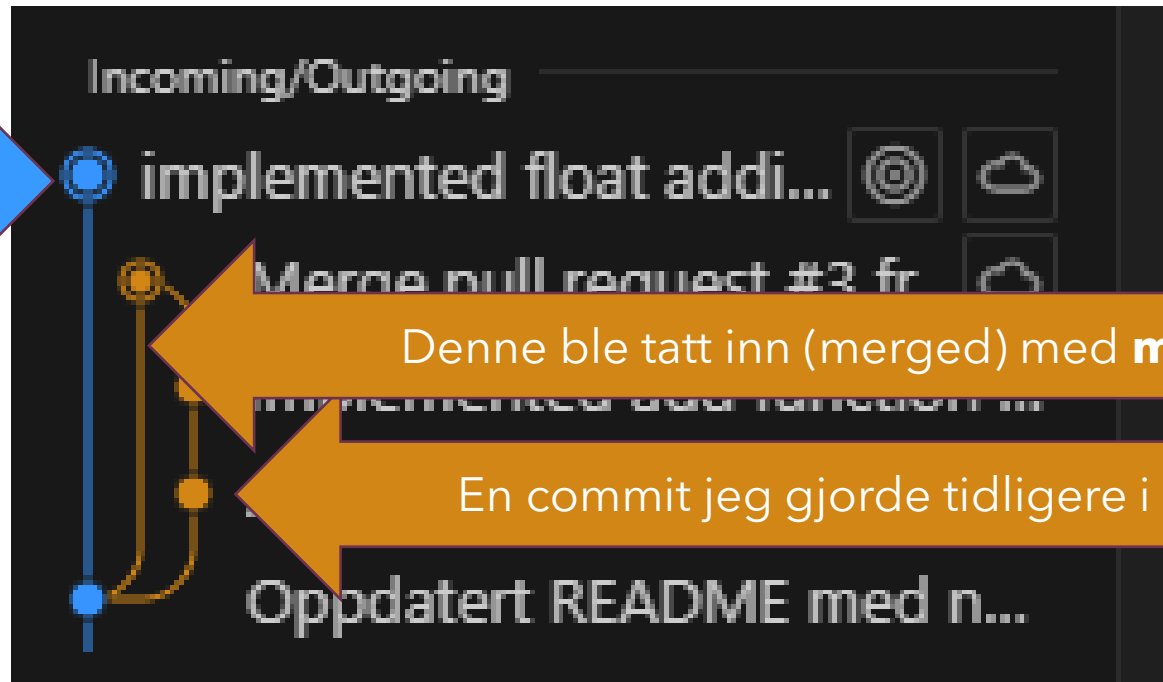
jdufresne and nateprewitt Update all documentation to show Python 3.7 is supported ... Latest commit c8f3add 16 days ago

.github	Change "StackOverflow" to "Stack Overflow"	2 months ago
_appveyor	Add appveyor testing of Requests	a year ago
docs	Update all documentation to show Python 3.7 is supported	7 days ago
ext	new logo :D	2 years ago
requests	Add return type on cookies.py	8 days ago
tests	make content-type's charset information case-insensitive	16 days ago
.coveragerc	omit coverage packages	2 years ago
.gitignore	gitignore	5 months ago



- Forgrening (branching): Jobbe parallelt i forskjellige grener av prosjektet

En commit fra meg i branchen jeg jobber i nå

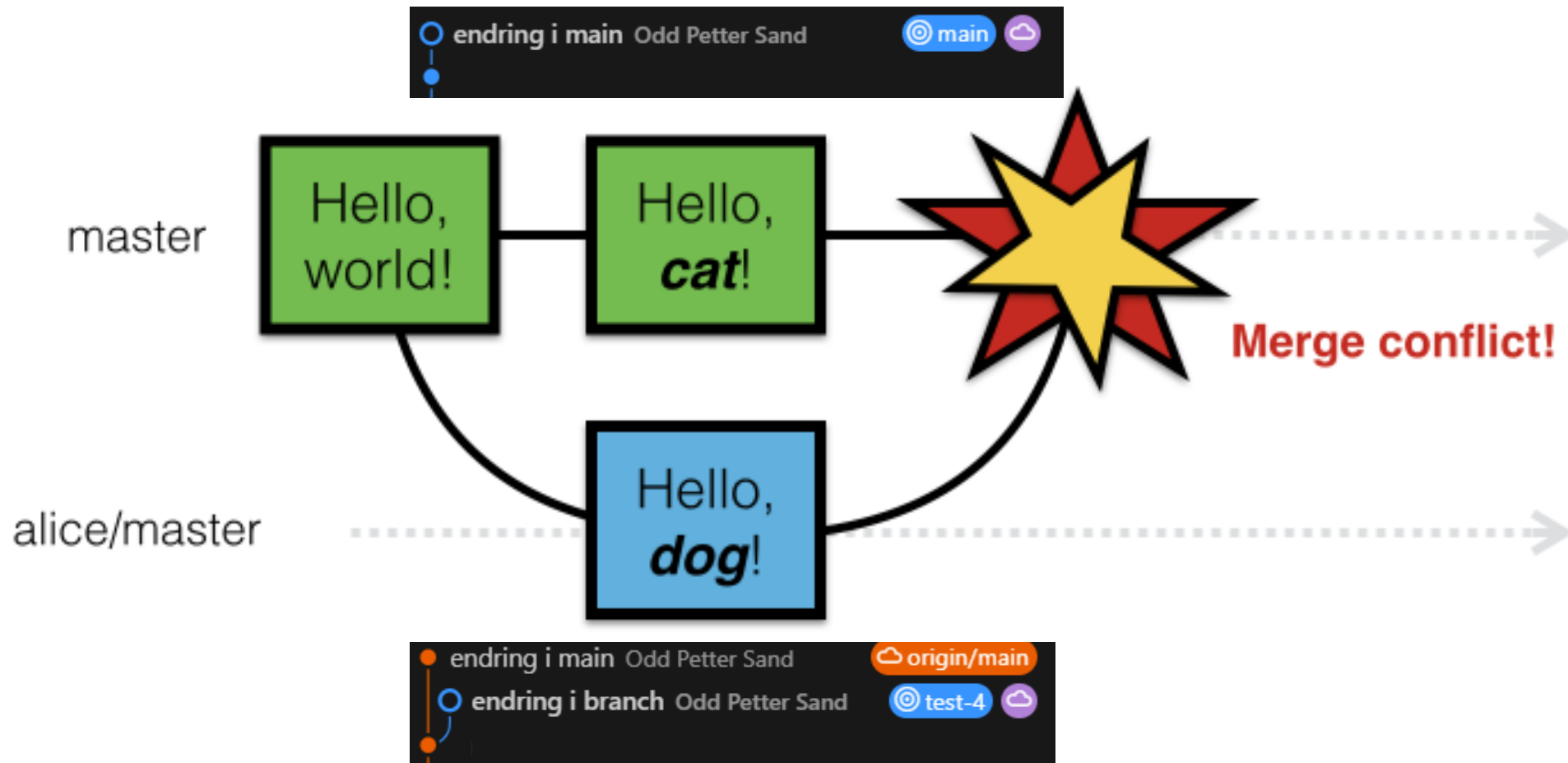


Denne ble tatt inn (merged) med **main**-branchen

En commit jeg gjorde tidligere i en annen branch

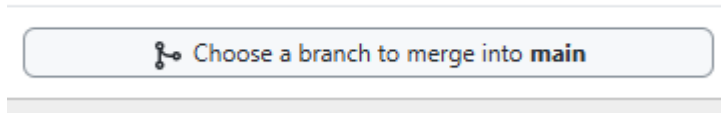


- Konflikter kan oppstå hvis samme kode endres i hver sin branch!

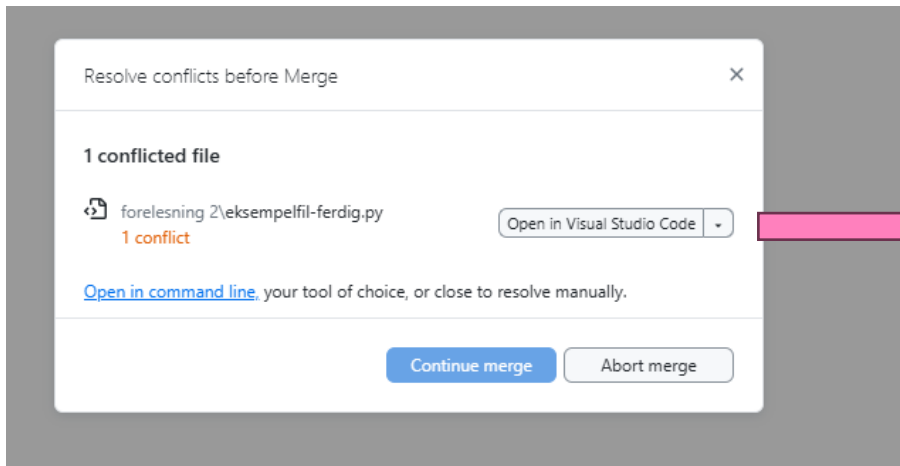
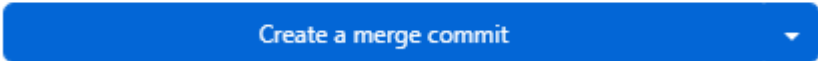




Github Desktop



There will be **1 conflicted file** when merging **test-4** into **main**



VS Code

```
1  print("Her er et program!")
2
3  x = input("x = ")
   Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
4  <<<<<<< HEAD (Current Change)
5  y = 2*x + 4
6  =====
7  y = x + 2
8  >>>>>>> test-3 (Incoming Change)
9
10 print(f"2*{x} + 4 = {y}")
```



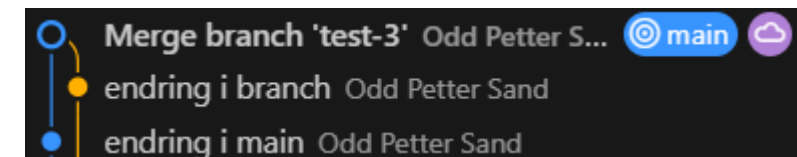
```
1 print("Her er et program!")
2
3 x = input("x = ")
  Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
4 <<<<<< HEAD (Current Change)
5 y = 2*x + 4
6 =====
7 y = x + 2
8 >>>>>> test-3 (Incoming Change)
9
```

Resolve in Merge Editor

Incoming	Current
φ 5d8c9c5 · refs/remotes/origin/test-3, refs/heads/test-3	φ 28ac8aa · refs/heads/main, refs/remotes/origin/main, refs/remotes/origin/HEAD
1 print("Her er et program!")	1 print("Her er et program!")
2	2
3 x = input("x = ")	3 x = input("x = ")
Accept Incoming Ignore	Accept Current Ignore
4 y = x + 2	4 y = 2*x + 4

```
Result C:\GitHub\IN1910-live\forelesning 2\eksempelfil-ferdig.py
1 print("Her er et program!")
2
3 x = input("x = ")
  No Changes Accepted
4 y = 2*x + 1
```

Complete Merge



○ Pull requests

- Når man skal inkludere kode fra en branch inn i **main**, vil man i et større prosjekt lage en *pull request*
- Da må en prosjektleder godkjenne endringene som er gjort, og passe på at andre endringer som er gjort i **main** i mellomtiden ikke forårsaker problemer
- Når pull request'en er godkjent vil endringene merges
- Men dette er også en nyttig arbeidsflyt når man samarbeider med seg selv på et lite prosjekt (derfor gjør vi det allerede i prosjekt 0)



○ Lure tips for GitHub

- Sjekk at du jobber i riktig branch – hvis du sender masse commits til feil branch, blir det fort en del å rydde opp
- Commit ofte! Så kan du gå tilbake til akkurat rett før noe eventuelt går galt / blir rotete
- Skriv tydelige commit-meldinger hver gang, så du selv og andre skjønner hva som ble gjort for hver commit
- Sørg for at koden virker, er grundig testet og godt dokumentert før du lager en pull request



- Livekoding (se [forelesningsopptak](#))
 - Klone Prosjekt 0-repository til lokal maskin med GitHub Desktop
 - Teste en commit i VS Code
 - Lage en ny branch
 - Alt dette er en del av Prosjekt 0, som er publisert på [emnesiden](#) (under beskjeder)



Welcome to GitHub Desktop

GitHub Desktop is a seamless way to contribute to open source and GitHub Enterprise. Sign in below to get started with your projects.

Sign in to GitHub.com

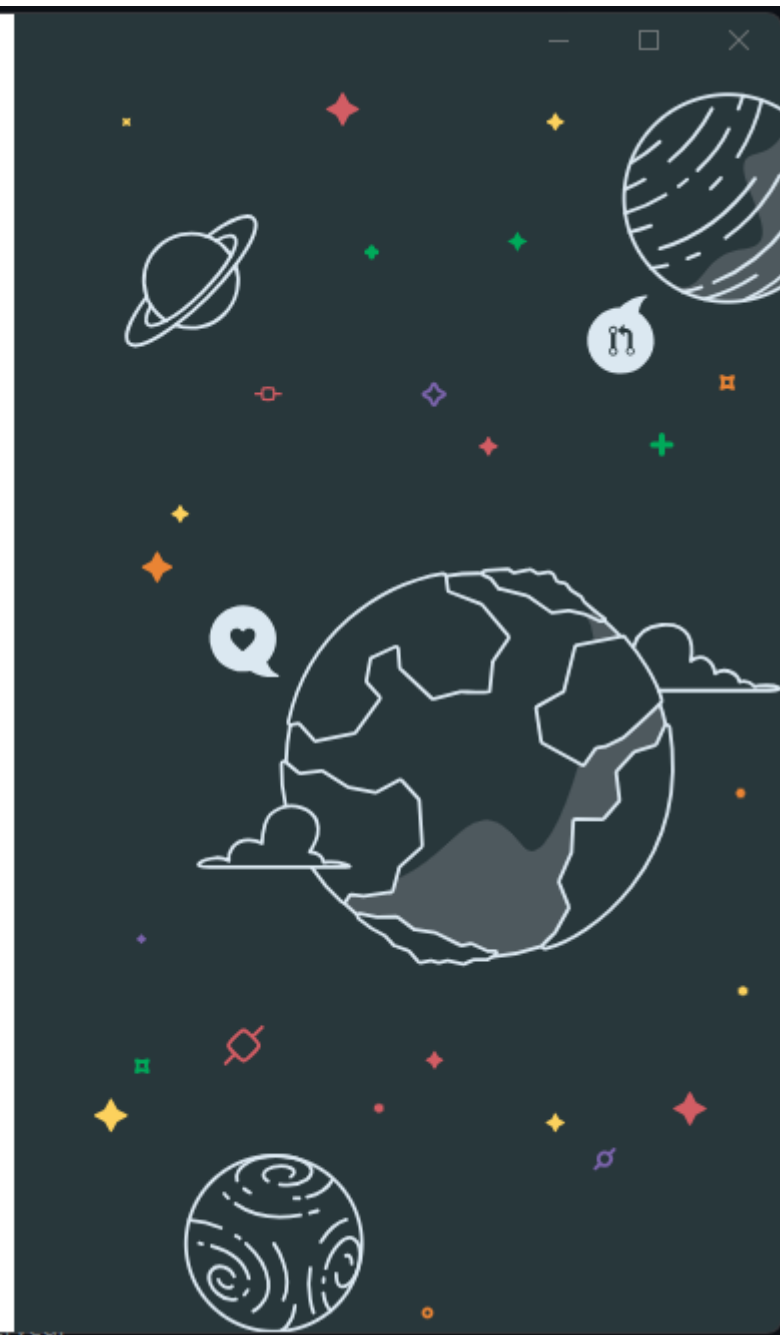
Sign in to GitHub Enterprise

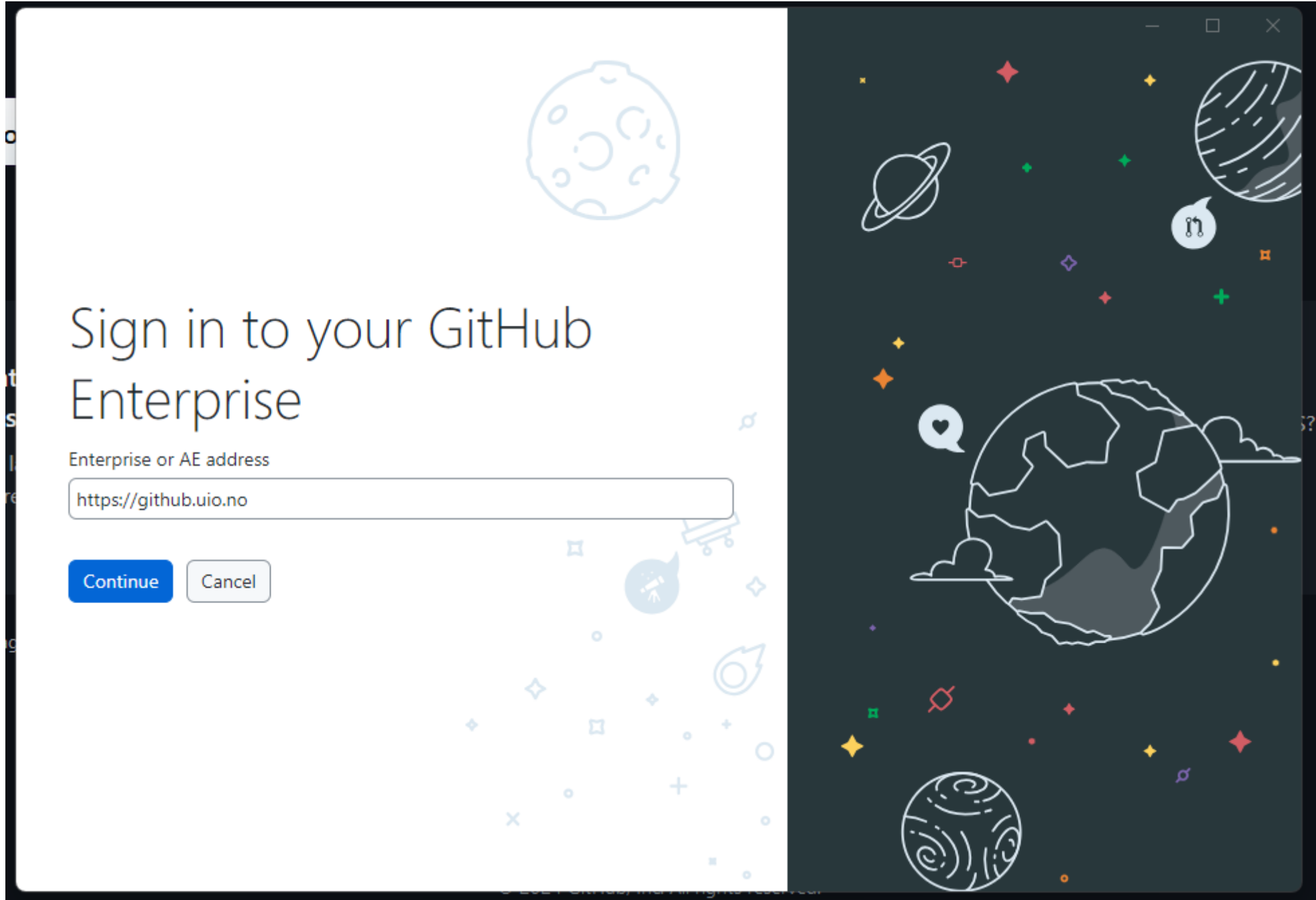
New to GitHub? [Create your free account.](#)

[Skip this step](#)


By creating an account, you agree to the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#).

GitHub Desktop sends usage metrics to improve the product and inform feature decisions. [Learn more about user metrics.](#)





Sign in to your GitHub Enterprise

Sign in using your browser 

or

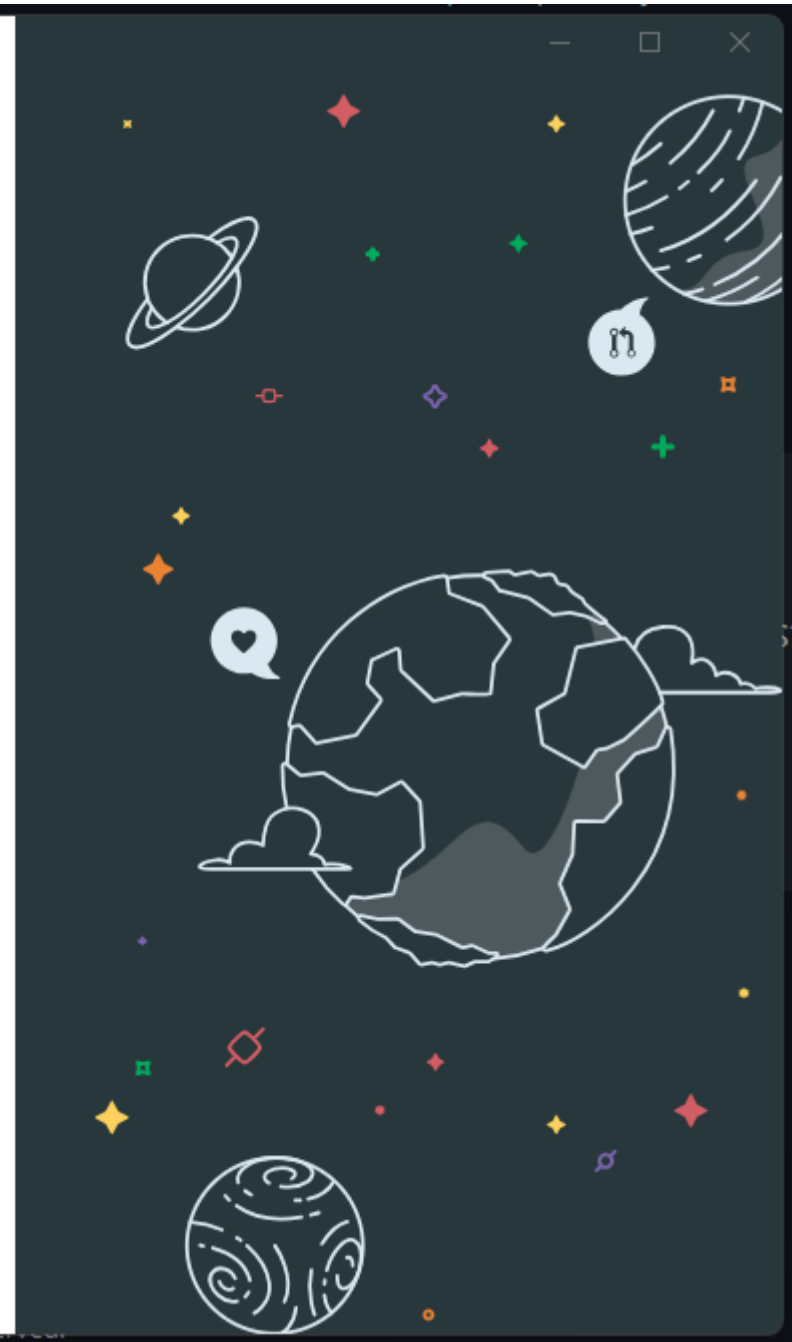
Username or email address

Password

Sign in

Cancel

[Forgot password?](#)





Configure Git

This is used to identify the commits you create. Anyone will be able to see this information if you publish commits.

- ☒ Use my GitHub Enterprise account name and email address
- ☐ Configure manually



Create a tutorial repository...



Clone a repository from the Internet...



Create a New Repository on your local drive...



Add an Existing Repository from your local drive...



ProTip! You can drag & drop an existing repository folder here to add it to Desktop



- Ikke legg repo'et i OneDrive-mappa!
(eller andre steder med auto-backup)

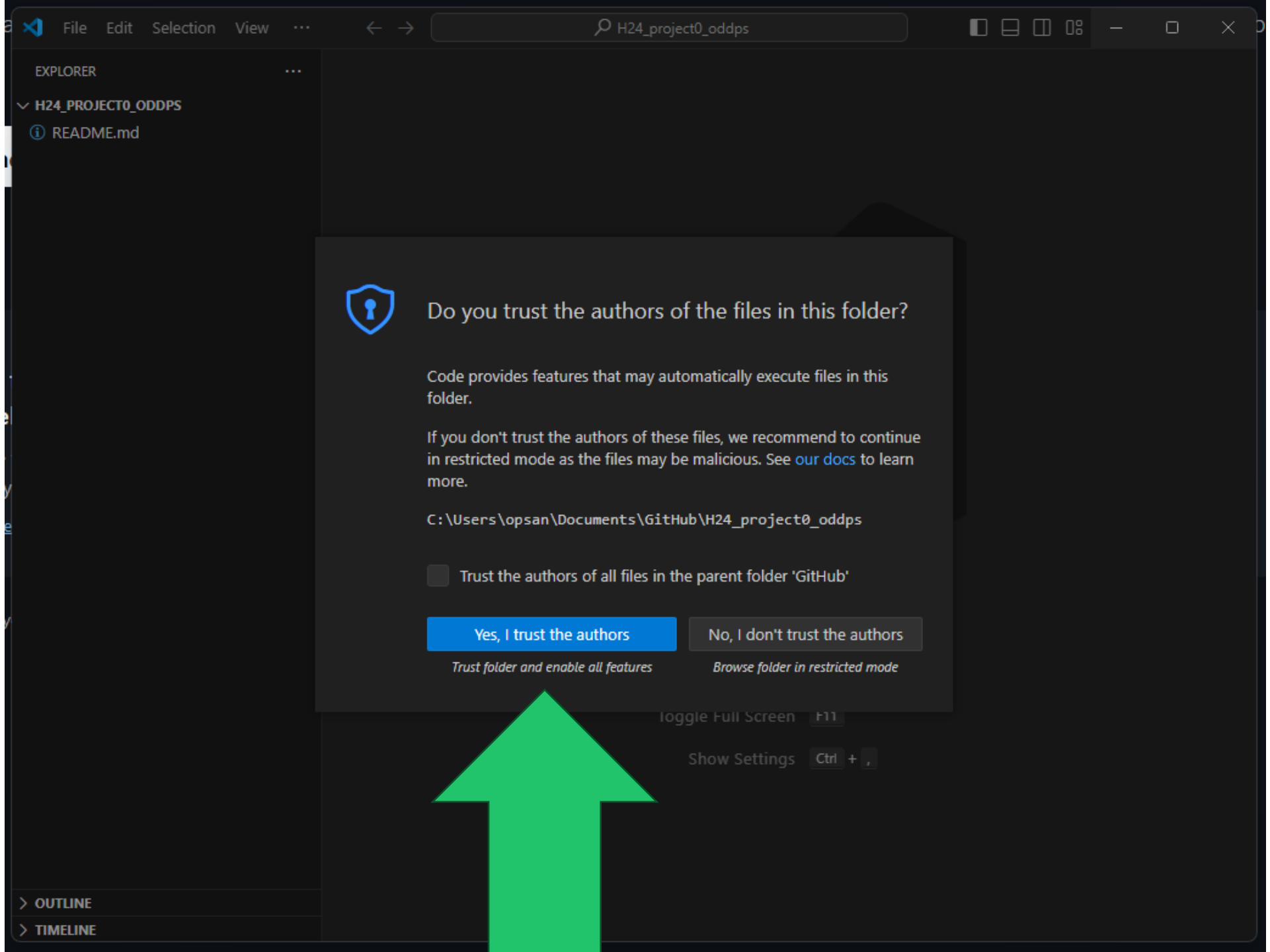


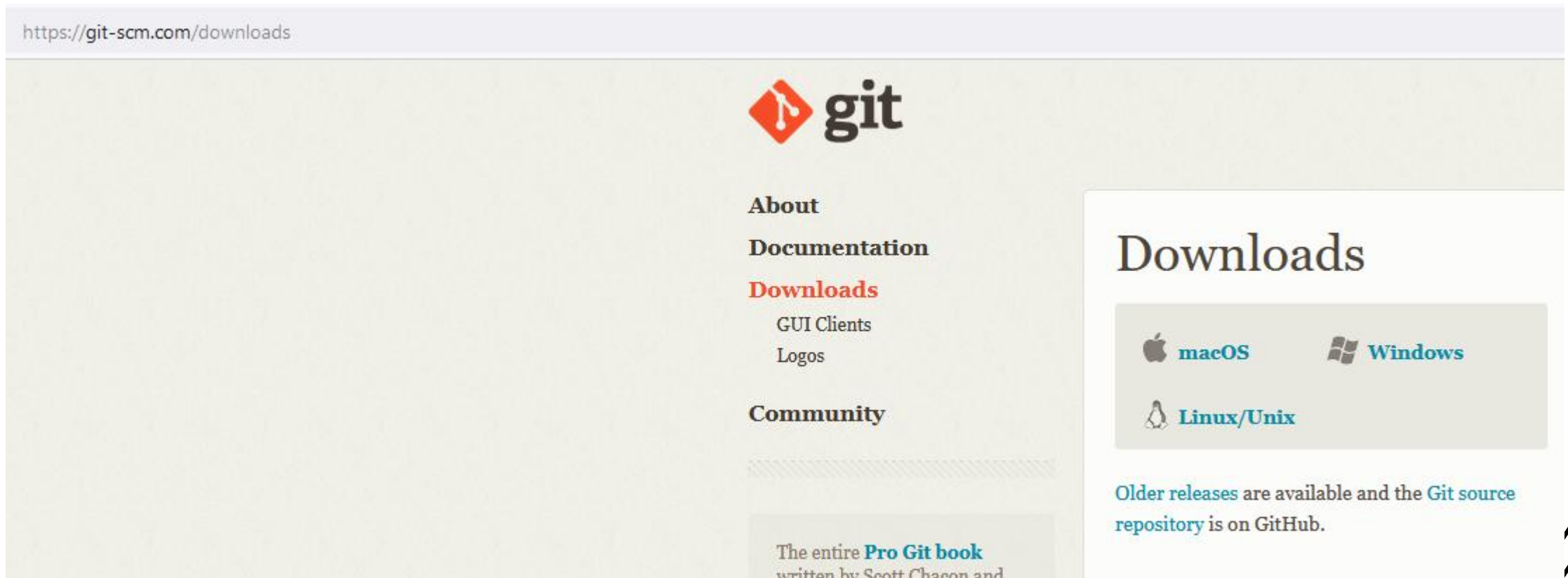
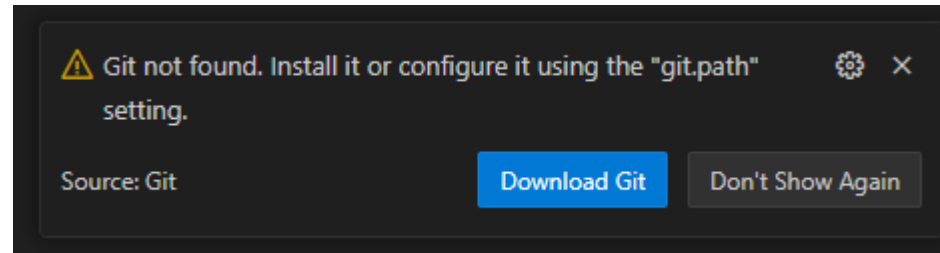
Local path

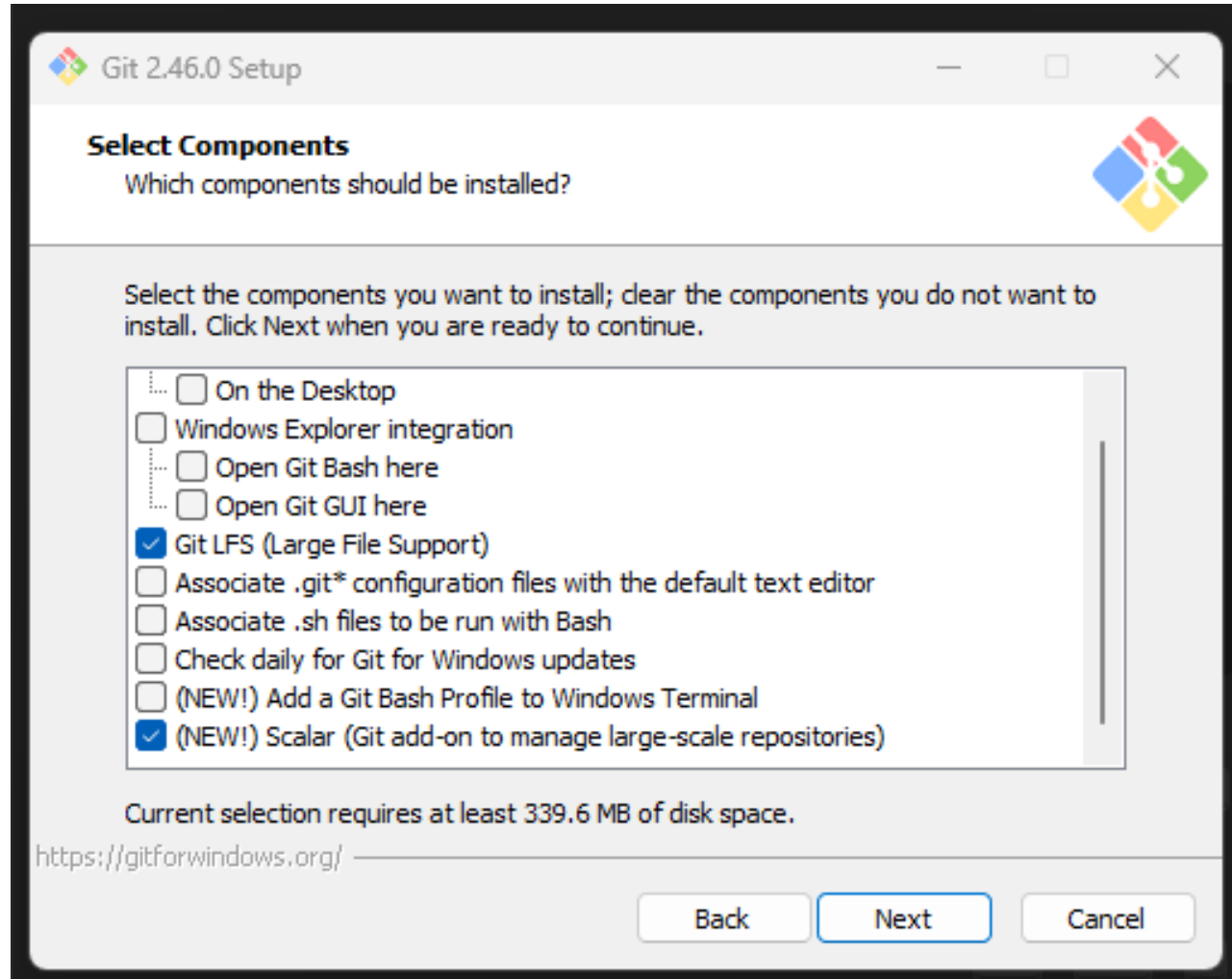


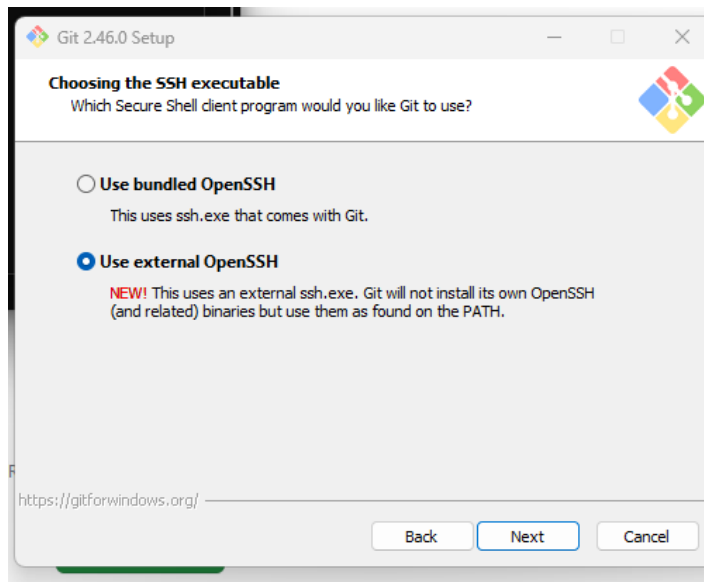
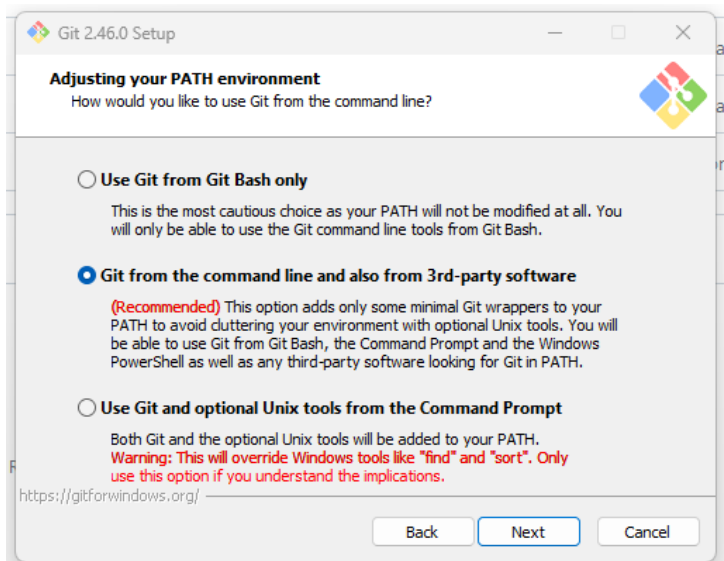
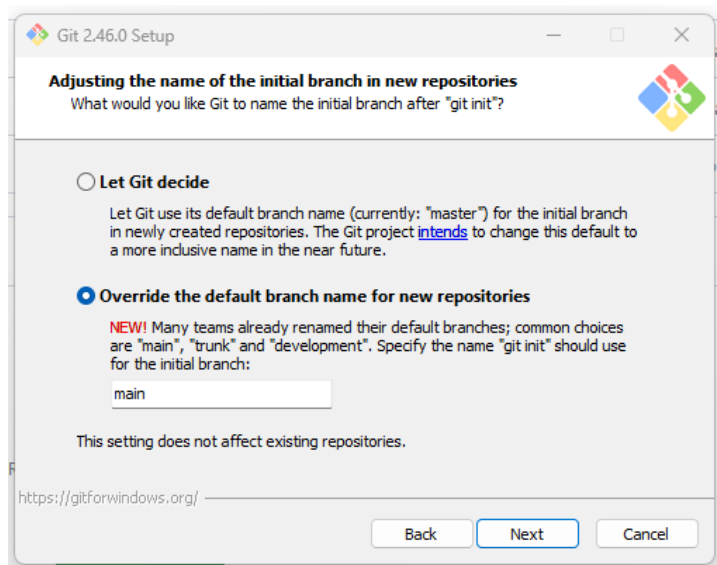
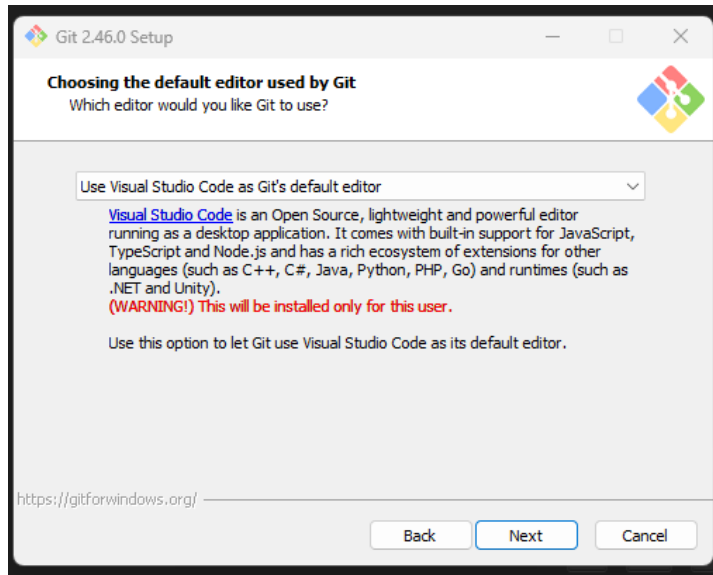
Local path

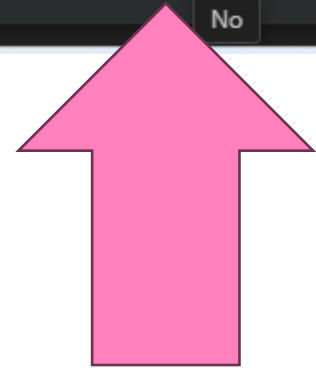
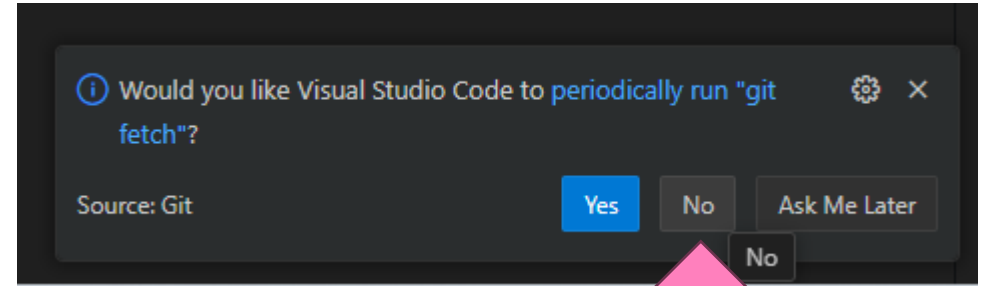
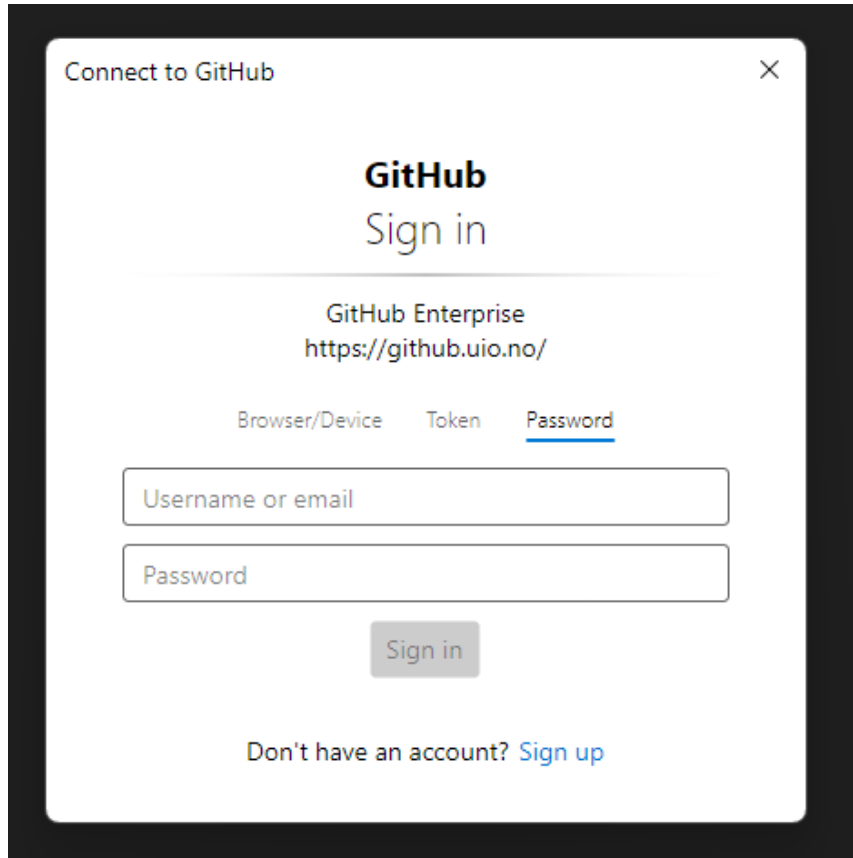
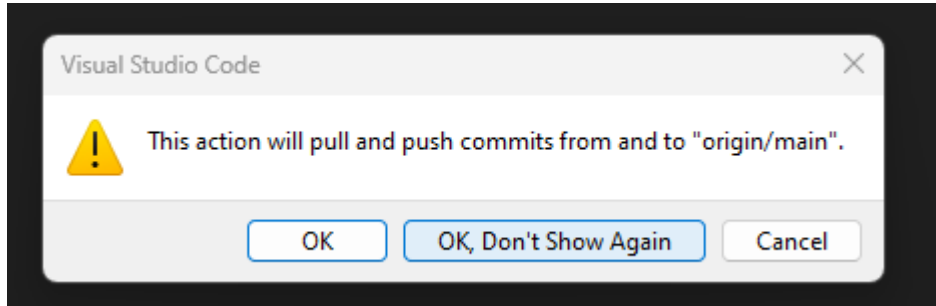












○ Litt GitHumor...



	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT
MESSAGES GET LESS AND LESS INFORMATIVE.

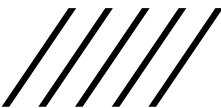




In case of fire



- 🔑 1. **git commit**
- 📁⬆️ 2. **git push**
- 🚶 3. **leave building**



○ Gruppetimene (starter neste uke)

- Her kan du jobbe med og få hjelp med prosjektene
- Du vil ikke få løsninger servert direkte, men få hjelp til å komme videre i riktig retning slik at du/dere lærer noe av det
- Spørsmål om og gjennomgang av viktige konsepter (dere kan be gruppelærere om hjelp til tema dere synes er vanskelig)
- I tillegg kan du jobbe med ukesoppgavene (frivillig, ikke pensum, likevel anbefalt fordi å jobbe med dem ofte vil gjøre prosjektene lettere)



○ Etter forelesningen

- Se på [resten av Prosjekt 0](#)
- Finn ut [hvilken gruppetime](#) du skal på neste uke (hvis du ønsker å bytte, må du spørre gruppelærer på timen du ønsker å dra på om det er plass)
- Installer det du eventuelt mangler av disse (se forelesningsopptak for hvordan de bør installeres):
 - [GitHub Desktop](#) (Windows/Mac)
 - Linux: Enten følg [denne oppskriften](#) eller bruk [git i terminalen](#) istedenfor
 - [Git](#)

