

INF-1400
17.02.2026
Anne Bosch





Plan for dagen

- Oversikt over arbeidskrav 2
 - Kravene til løsningen deres
 - Hva skal levers og hvor
 - Vurdering
- Versjonskontroll med git
- Arbeidsprosess mot GitHub
- Oppgave for denne uka

Arbeidslivsrelevans

Anvende
objektorienterte
prinsipper

AI-verktøy

GUI / pygame

Git og GitHub

Større prosjekt
uten en fasit

Refleksjon

Oblig 2

Arbeidslivsrelevans

Anvende
objektorienterte
prinsipper

AI-verktøy

Oblig 2

GUI / pygame

Git og GitHub

Større prosjekt
uten en fasit

Refleksjon

Objektorienterte prinsipper

- Klasser, instanser og objektrelasjoner
- Arv mellom klasser
- Komposisjon
- Polymorfi
- Abstrakte baseklasser
- Informasjonsgjemming og tilgangskontroll (encapsulation)

Arbeidslivsrelevans

Anvende
objektorienterte
prinsipper

AI-verktøy

Oblig 2

GUI / pygame

Git og GitHub

Større prosjekt
uten en fasit

Refleksjon

Arbeidslivsrelevans

Anvende
objektorienterte
prinsipper

AI-verktøy

GUI / pygame

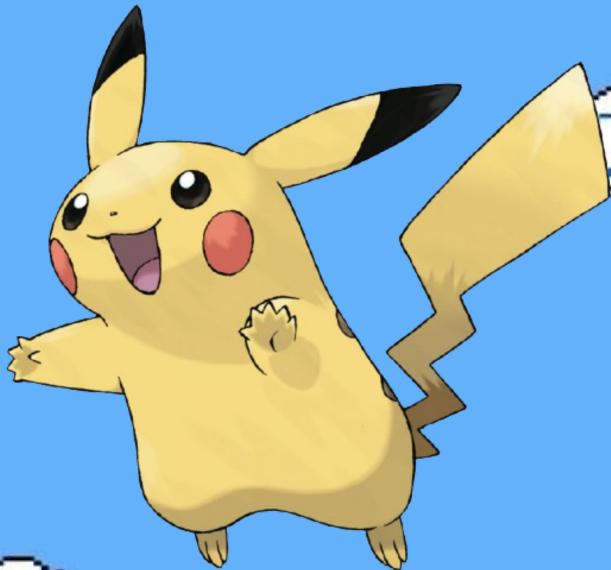
Oblig 2

Git og GitHub

Større prosjekt
uten en fasit

Refleksjon

Krav til løsningen dere skal utvikle

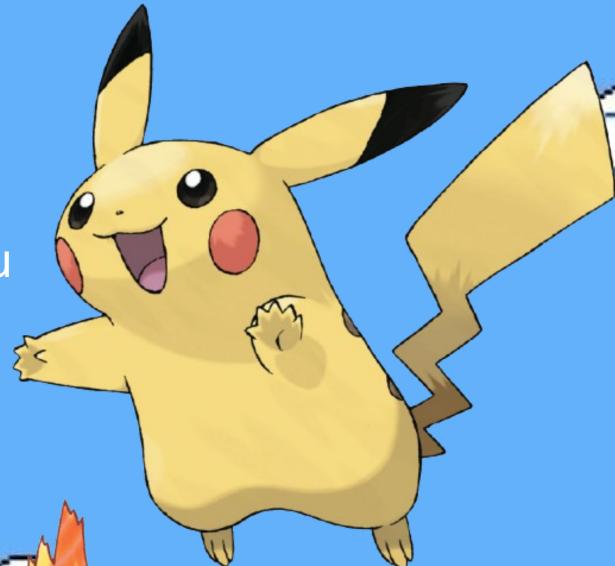




Bulbasaur



Charmander



Pikachu

Arter / species



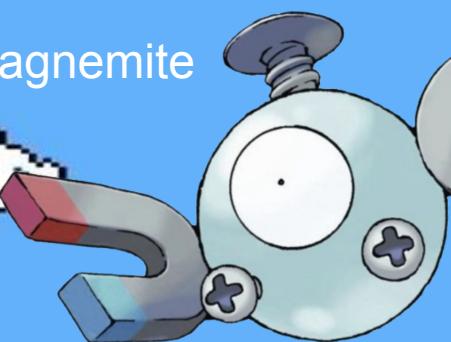
Fred

Bob

Flere покемон av
samme art



Magnemite



Minst 3 ulike arter



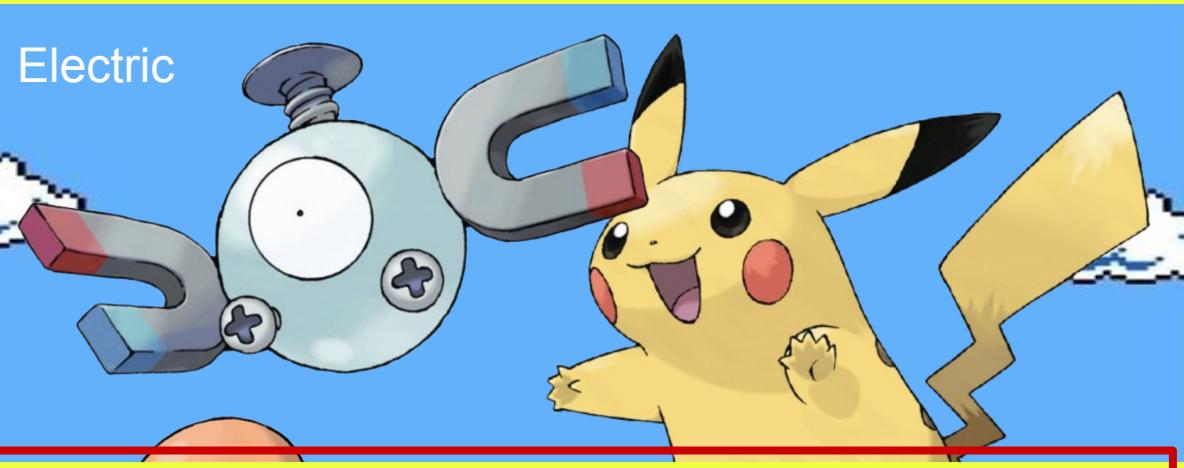
Rapidash





Grass

Minst 3 ulike typer



Electric



Fire

Fred

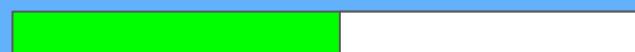
157 sek



Behov



Energy



Nutrition



Happiness

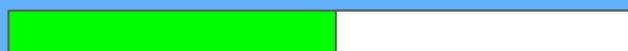
Humør/tilstand: Fornøyd

Fred

157 sek

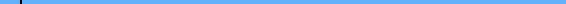
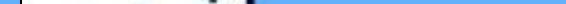


Behov



Humør/tilstand: Fornøyd

Bladder	Fun
Hunger	Social
Energy	Hygiene



Fred

157 sek



Tiden går

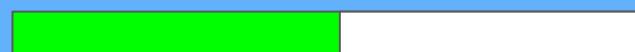


verdien til behovene synker

Behov



Energy



Nutrition



Happiness

Humør/tilstand: Fornøyd

Fred

200 sek



Tiden går



verdien til behovene synker

Behov



Energy



Nutrition



Happiness

Humør/tilstand: Urolig

Fred

200 sek



Tiden går

↓
verdien til behovene synker

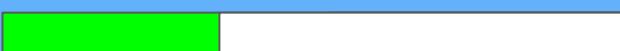
Behov

Utføre handlinger

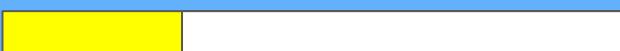
↓
verdien til behovene øker



Energy



Nutrition



Happiness

Humør/tilstand: Urolig

Fred

210 sek



Behov



Energy



Nutrition



Happiness

Humør/tilstand: Urolig

Tiden går



verdien til behovene synker

Utføre handlinger



verdien til behovene øker

`fred.gi_mat()`

Fred

210 sek



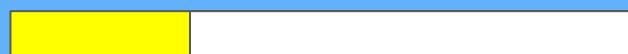
Behov



Energy



Nutrition



Happiness

Humør/tilstand: Nøytral

Tiden går



verdien til behovene synker

Utføre handlinger



verdien til behovene øker

`fred.gi_mat()`

210 sek

Luke



Behov



Humør/tilstand: Tilfreds

Tiden går



verdien til behovene synker

Utføre handlinger



verdien til behovene øker

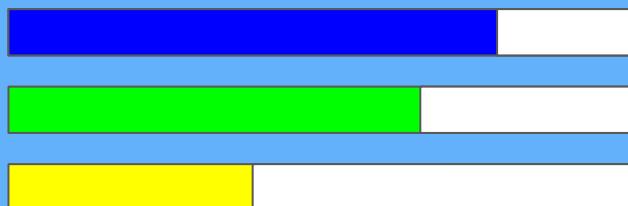
`luke.gi_mat()`

210 sek

Luke



Behov



Humør/tilstand: Tilfreds

Tiden går



verdien til behovene synker

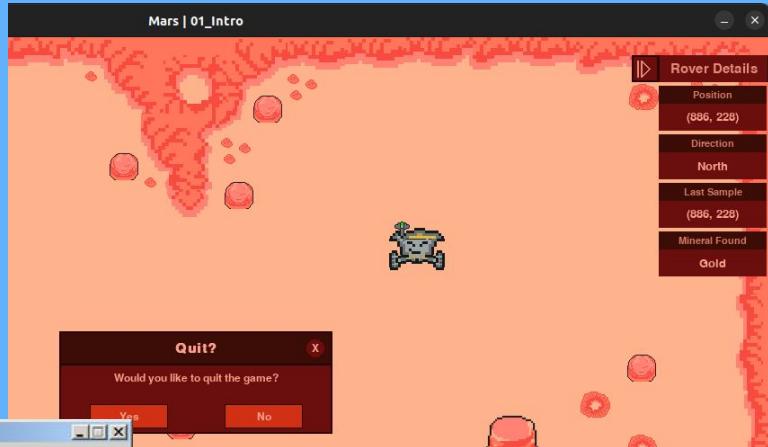
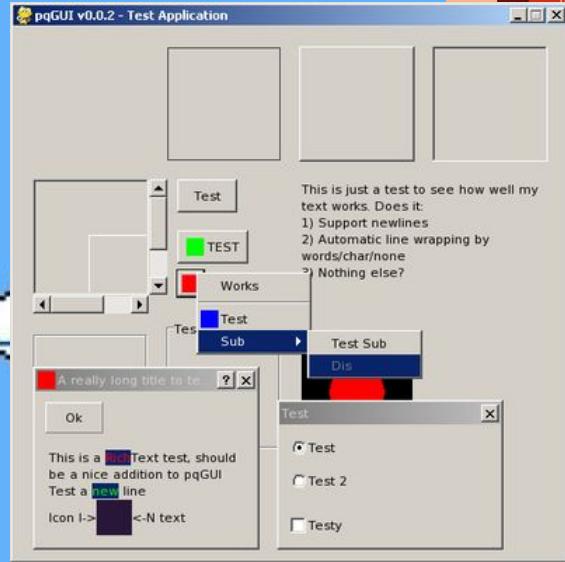
Utføre handlinger



verdien til behovene øker

`luke.gi_mat()`

GUI og Interaksjon



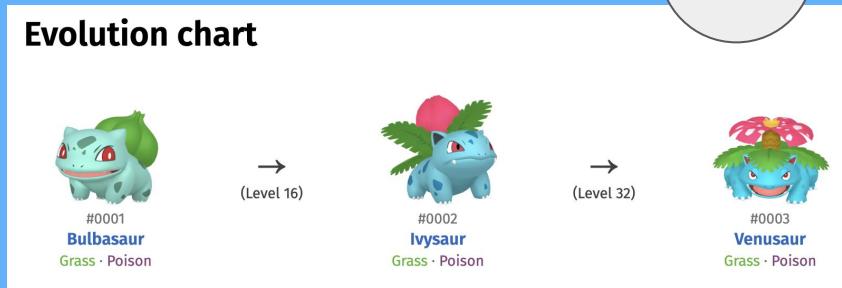
Base case requirements

- Simulator - ta vare på deres Pokémon
- Grafisk brukergrensesnitt - pygame
- Interaksjon
- Minst 3 ulike arter
- Minst 3 ulike typer
- Behov
- Mulig å skille Pokémon-instanser av samme art
- Tidsaspekt
- Pokémon reagerer på interaksjon
- Første versjon
- Felles krav



Utvidelser

Evolution chart



2

	0025	Pikachu	ELECTRIC
	0001	Bulbasaur	GRASS POISON



3

DEFENSE → ATTACK ↴	NOR	FIR	WAT	ELE	GRA	ICE	FIG	POI	GRO	FLY	PSY	BUG	ROC	GHO	DRA	DAR	STE	FAI
NORMAL																		
FIRE	½	½			2	2					2	½	2	½	½	½	2	
WATER	2	½			½	½				2		2		½	½	½	½	
ELECTRIC		2	½	½	½					0	2				½	½	½	
GRASS	½	2			½				½	2	½		½	2	½	½	½	
ICE	½	½			2	½			2	2			2	½	2	½	½	
FIGHTING	2				2				2	½	½	½	½	2	0	2	2	½
POISON					2				½	½	½	½	½	½	0	2	2	

4



Arbeidslivsrelevans

Anvende
objektorienterte
prinsipper

AI-verktøy

Oblig 2

GUI / pygame

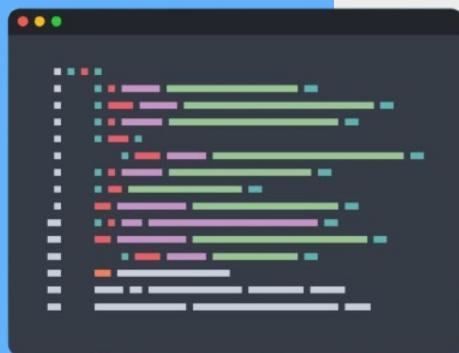
Git og GitHub

Refleksjon

Større prosjekt
uten en fasit

KI-bruk

4. Avansert KI-samarbeid



Du kan deletere tydelig avgrensede oppgaver (f.eks. utkast til avsnitt, skriving av datarense-skript, koding, simuleringer, utforming av visuelle plakater, generering av bilder/video/musikk, oversettelse av eget utkast). Verifisering er obligatorisk.

Ulike løsninger

Fredagens forelesning

Leveranser - 3 deler



Kode - 2 versjoner

10 minutter
fagsamtale

Gjennomføring
23. - 27. mars



Frist: 20. mars
Rapport



Fagsamtaler



- Muntlig vurdering
- Objektorientert programering
- Triller terning tema
- Ikke presentasjon
- Hjelpearereren stiller åpne spørsmål
- Designvalg - forståelse og begrunne
- Drøfting av ulike muligheter
- Lete etter kunnskapen deres

- 10 minutter
- Individuelt
- 23. - 27. mars
- Booking - førstemann til mølla
- Fysisk/digital
- Bestått/ikke bestått
- Info canvas
- Tilrettelegging
- Utfordringer og mestring
- Ute i jobb



Vurdering - Objektorienterte prinsipper

Klasser, instanser og objektrelasjoner	Arv mellom klasser	Komposisjon	Polymorfisme	Abstrakte baseklasser	Informasjons-gjemming og tilgangsstyring (encapsulation)
Design av klasser og prosjektstruktur					
Instansiering av objekter					
Relasjoner mellom klasser					
Tilstandsendring					
Ansvarsfordeling mellom klasser					

Rapporten

- Designvalg
- Drøfte egen AI-bruk
- Profesjonelle utviklerpraksiser - Versjonskontroll
- Ikke AI-genereres

Arbeidslivsrelevans

Anvende
objektorienterte
prinsipper

AI-verktøy

GUI / pygame

Git og GitHub

Større prosjekt
uten en fasit

Refleksjon

Oblig 2

Arbeidslivsrelevans

Anvende
objektorienterte
prinsipper

AI-verktøy

GUI / pygame

Git og GitHub

Oblig 2

Større prosjekt
uten en fasit

Refleksjon

Hvor skal dere levere?

- GitHub (Classroom)
- report.md
- Koden i prosjektet
- Hjemmeoppgave denne uka - sette opp git og GitHub



THIS IS GIT. IT TRACKS COLLABORATIVE WORK
ON PROJECTS THROUGH A BEAUTIFUL
DISTRIBUTED GRAPH THEORY TREE MODEL.

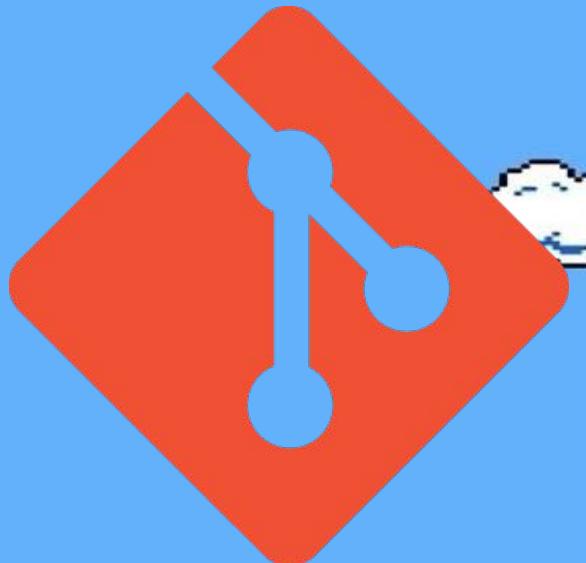
COOL. HOW DO WE USE IT?

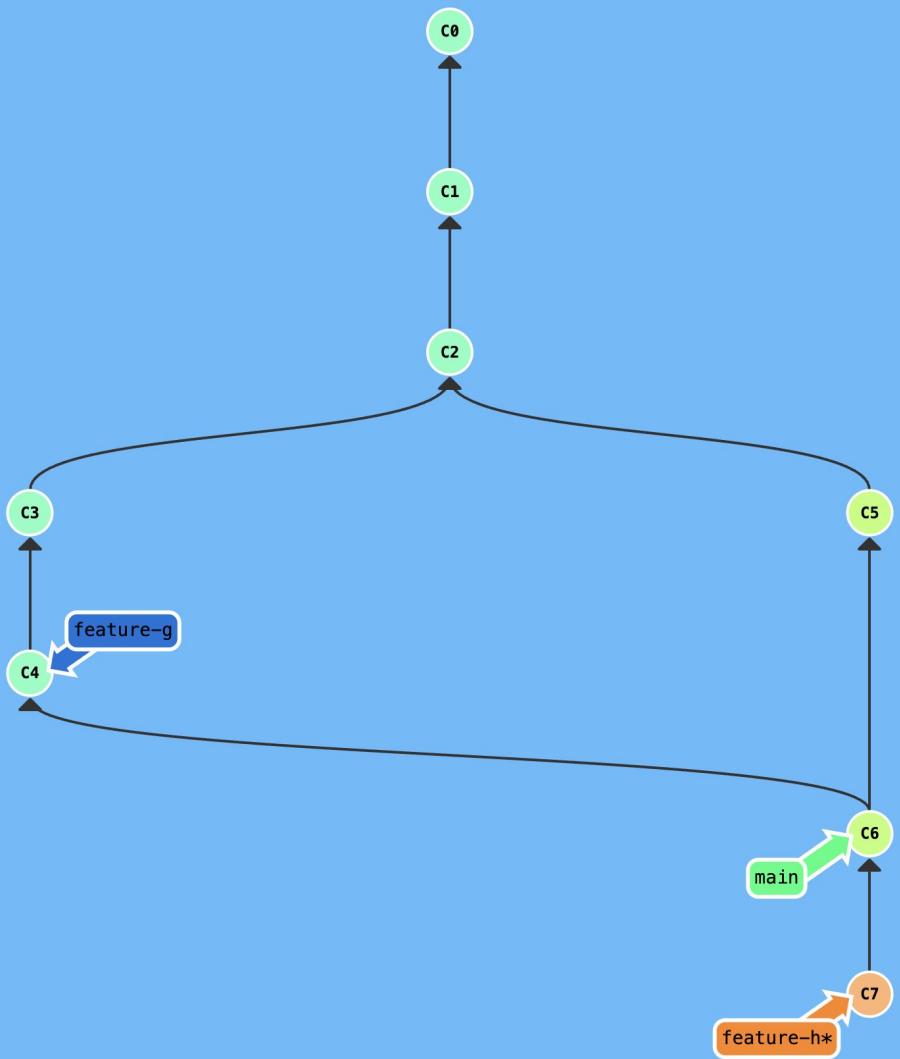
NO IDEA. JUST MEMORIZIZE THESE SHELL
COMMANDS AND TYPE THEM TO SYNC UP.
IF YOU GET ERRORS, SAVE YOUR WORK
ELSEWHERE, DELETE THE PROJECT,
AND DOWNLOAD A FRESH COPY.



Hva er git?

- Versjonskontrollsysteem
- Kontroll over: ulike versjoner av din kode
- Repository
- gitignore
- Oversikt over endringer over tid
- Hoppe mellom versjoner

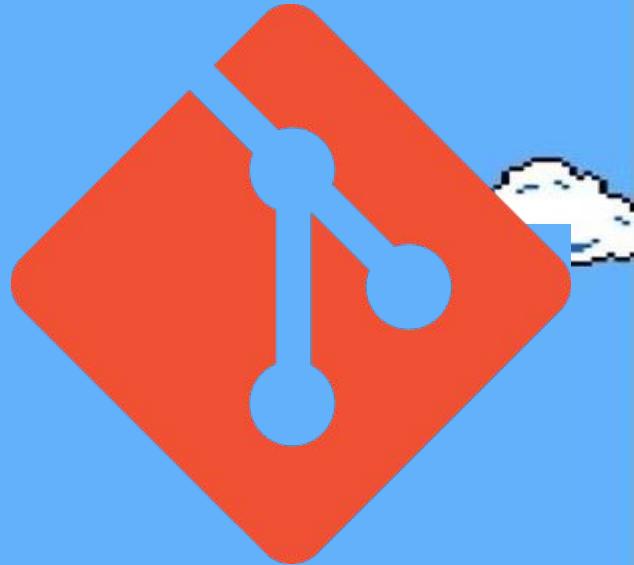




Hva er git?



- Versjonskontrollsysteem
- Kontroll over: ulike versjoner av din kode
- Repository
- gitignore
- Oversikt over endringer over tid
- Hoppe mellom versjoner
- Linus Torvalds
- 2005
- 2008

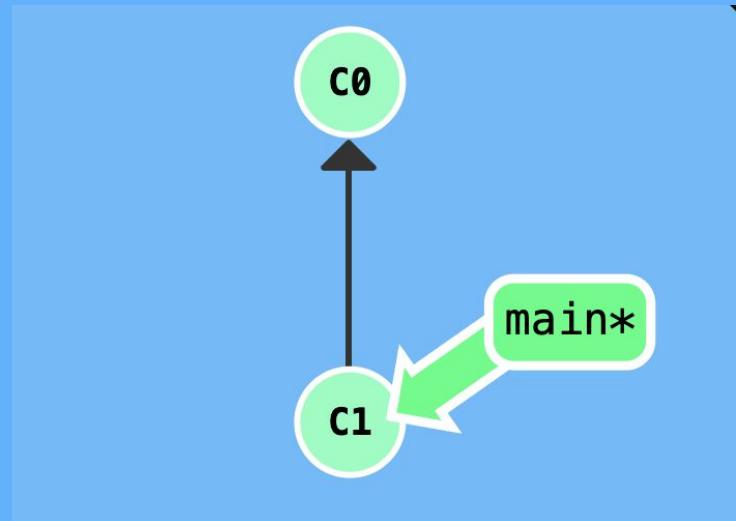


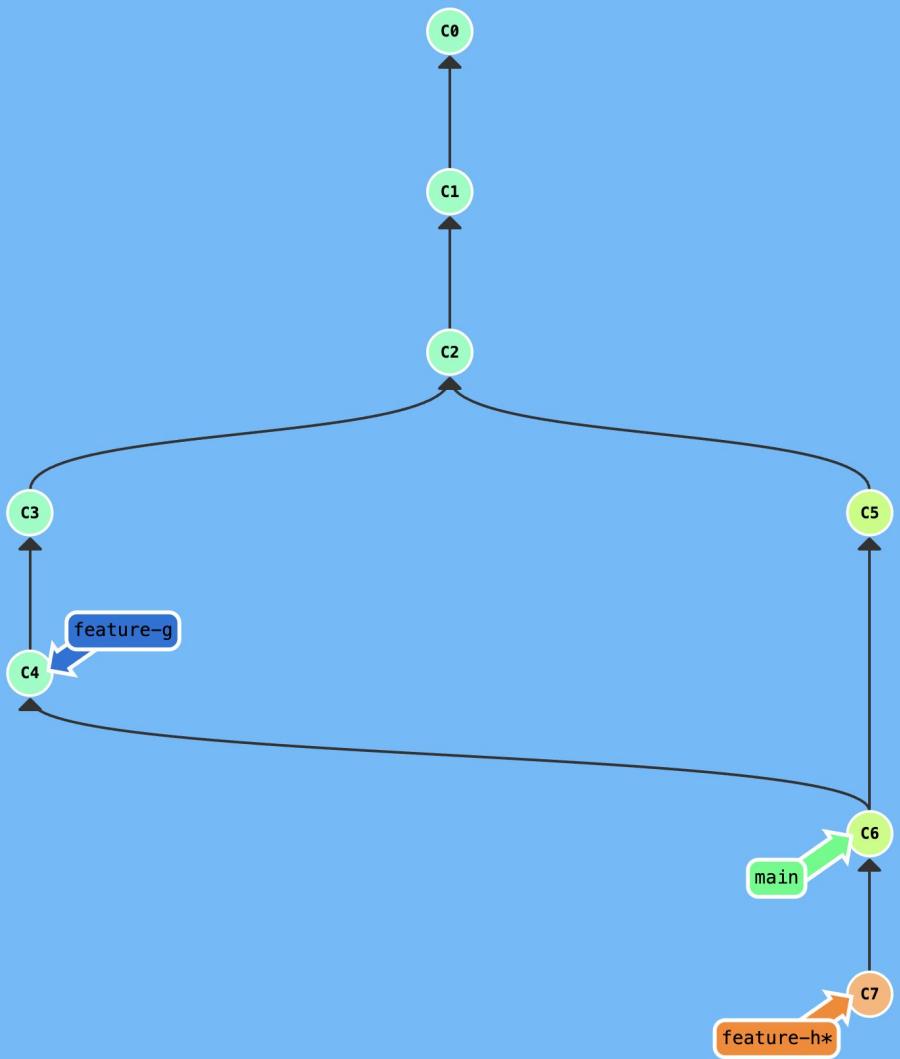
Konsepter

- Commit
- Staging-området /index
- Branch
- HEAD
- Remote
- Kommandoer

Commit

- Snapshot av hele repoet
- Hash / id
- Parent
- Melding
- Tilknyttet forfatter
- Tidsstempel





```
annebosch@ forelesninger % git show 0c96aa19575f1647c88258ec9f317e1d1  
commit 0c96aa19575f1647c88258ec9f317e1d158aaa89 (HEAD -> main, origin/main, origin/HEAD)  
Author: ambosch <anne.bosch@...>  
Date:   Tue Feb 3 15:53:55 2026 +0100
```

update lecture plan

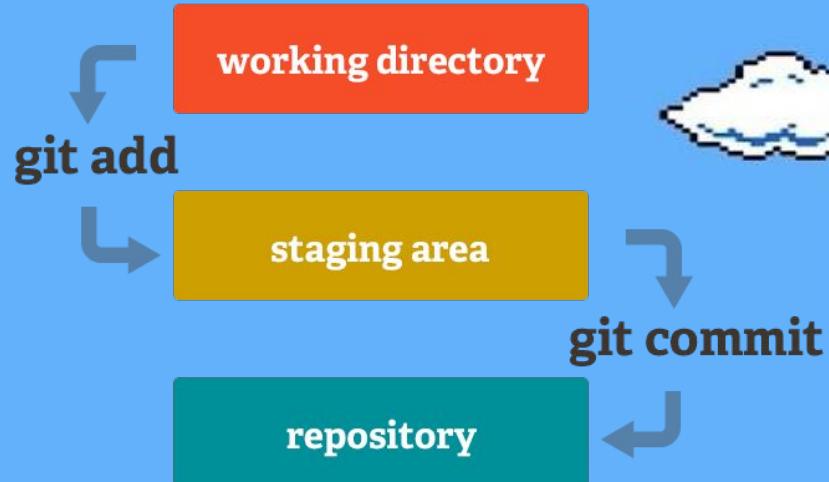
```
diff --git a/semesterplan.MD b/semesterplan.MD  
index f5d5623..21b9c41 100644  
--- a/semesterplan.MD  
+++ b/semesterplan.MD  
@@ -1,5 +1,5 @@  
-Forelesninger  
-----  
+## Forelesninger  
+  
- Tirsdag 14:15-16:00  
-----
```

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

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

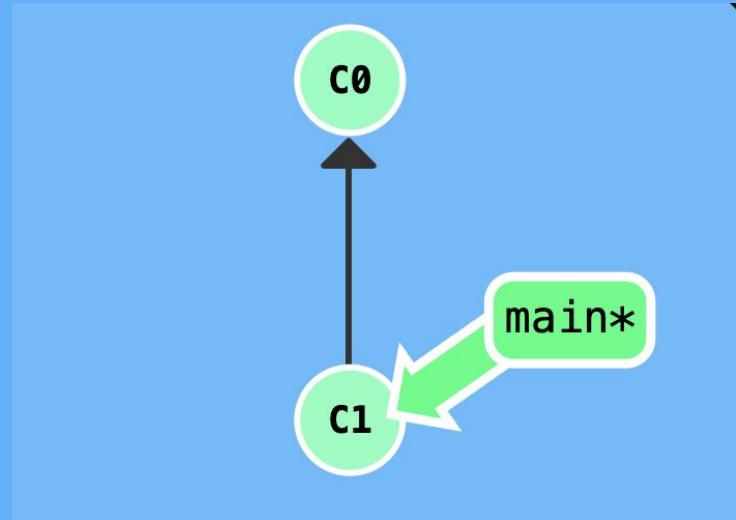
Staging-området

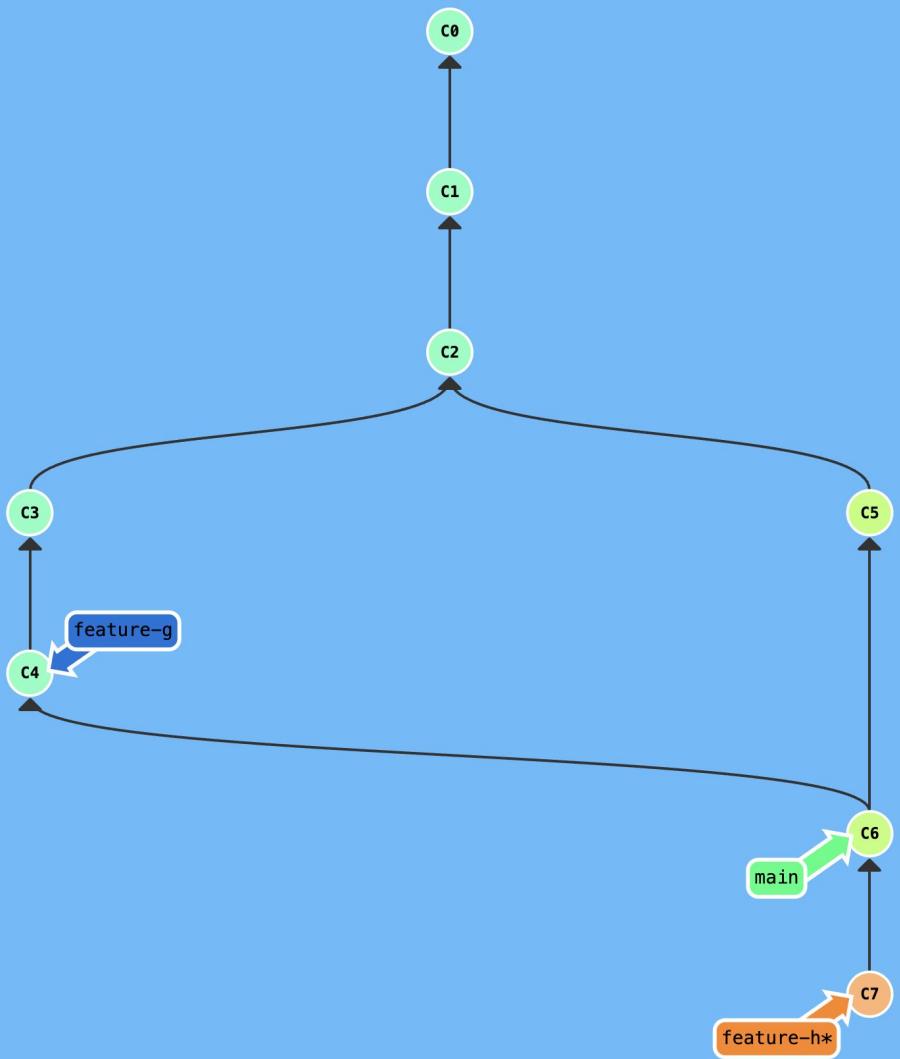
- Sette sammen en commit
- Ikke alle endringer må bli med
- Hele filer
- Enkelte linjer



Branch

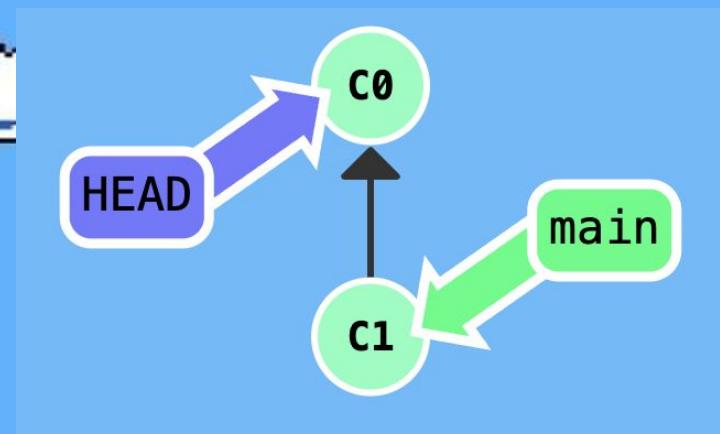
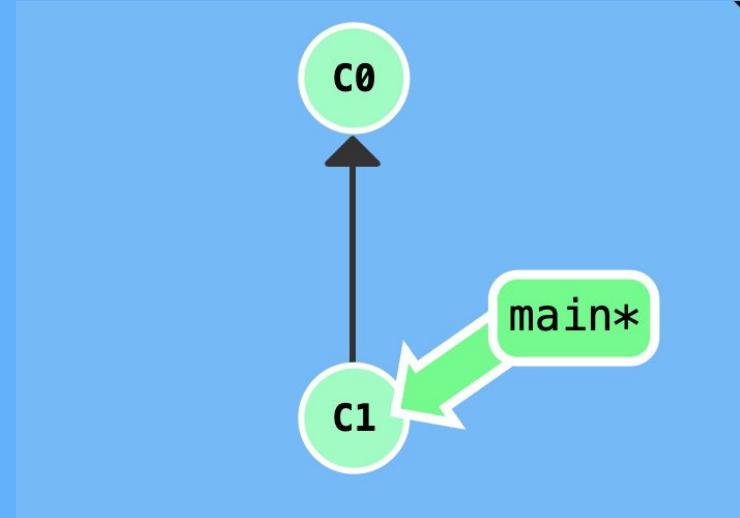
- Peker til en commit
- Pekeren har et navn
- Utforsking
- Systematisering
- Isolering
- Velge hva du ønsker å hente inn i main med en merge
- Teamarbeid





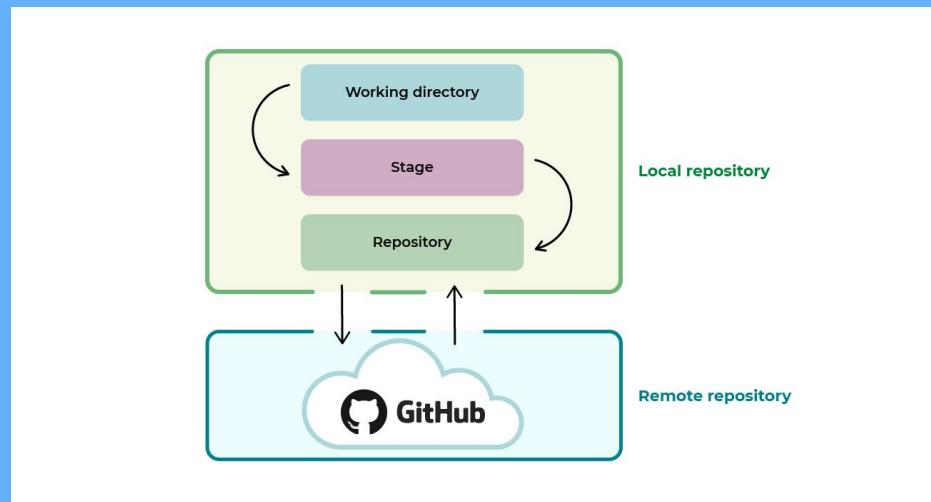
HEAD

- Referanse til nåværende posisjon i historikken
- Hoppe rundt i historikken
- Detatched head



Remote

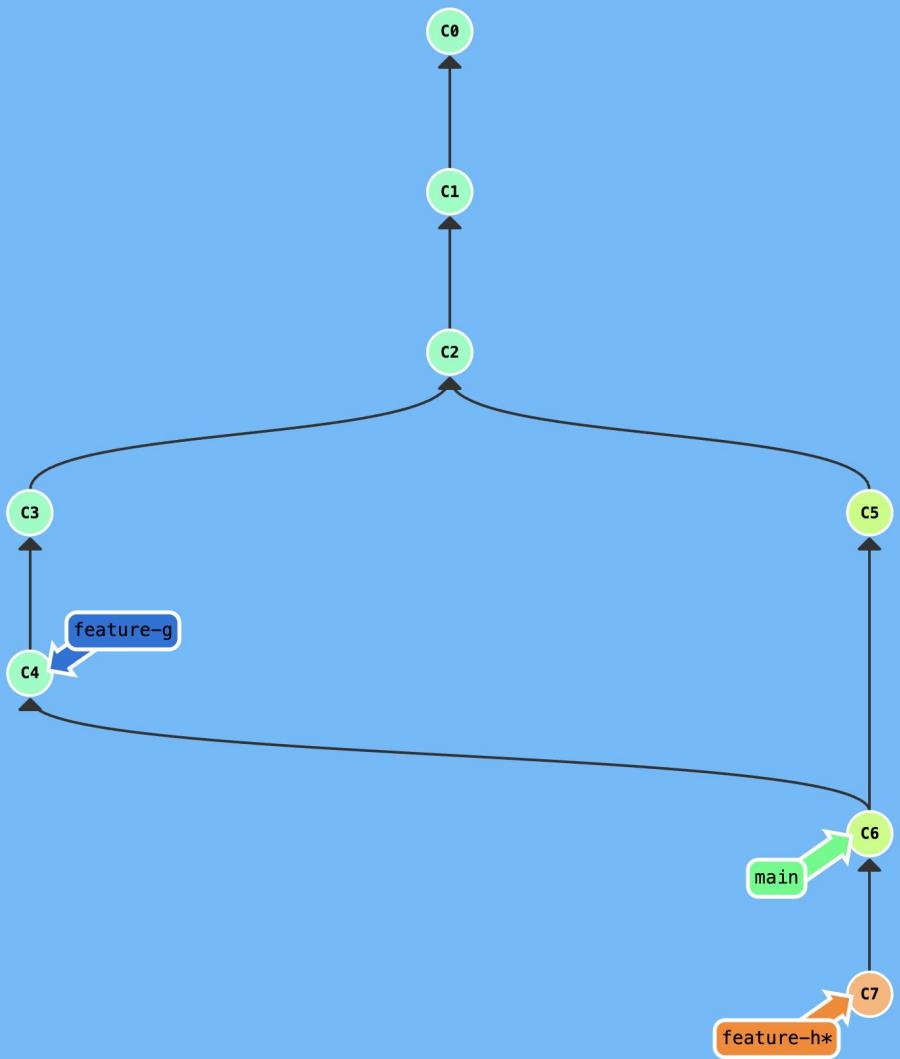
- GitHub
- En kopi av repoet på en annen server (deployet)
- Backup
- Deling med andre
- Pull requests og code reviews
- De-facto sannheten
- Synkronisere lokalt og remote manuelt



Kommandoer

- git init
- git add <file.md>
- git commit -m "<my message>"
- git commit --amend
- git status
- git branch <name>
- git checkout <branchname>
- git merge <branchname>





Kommandoer

- git init .
- git add <file.md>
- git commit -m "<my message>"
- git commit --amend
- git status
- git branch <name>
- git checkout <branchname>
- git merge <branchname>

- git clone <remote url>
- git fetch
- git pull
- git push



Aldri utlært, men kommer langt med kjernen

- tags
- stash
- rebase
- reflog
- reset
- cherry-pick
- octopus-merge
- forks
- ...

Bli komfortabel med git

→ investering som lønner seg

Oblig 3 - teamarbeid



Drar sammen trådene til obligen

- Bruke git i utviklingen
- Teste ut ulike løsninger
- Levere koden på GitHub
- Pushe opp til remote
- 2 brancher
 - Før utvidelsen
 - Etter utvidelsen

Gode praksiser

- Vi jobber sjeldent direkte i main
- Merger PRer inn i main
- Committe ofte
- Små commits
- Gode committ-meldinger
- Pushe ofte

(DEMO)

Oppgave til fredag

- Sette opp git lokalt - installere
- GitHub-bruker med uit-e-posten
- Godta invitasjonen inn i emnet sin GitHub-organisasjon (får e-post med link)
- Godta oppgaven i GitHub classroom (egen link)
- Da har du automagisk fått et repo på github
- Oppgaveteksten
- Klone dette repoet lokalt
- Autentisering mellom GitHub og lokal maskin
- Pushe opp en liten endring - opprette main.py
- Da er du klar til å knekke igang med oblig 2
- Bonus: Søke om GitHub copilot pro-lisens for studenter

Legge ut på Canvas:

- Smørbrødlista
- Guide til autentisering
- Lenker - øve på git-kommandoer

Arbeidslivsrelevans

Anvende
objektorienterte
prinsipper

AI-verktøy

GUI / pygame

Git og GitHub

Oblig 2

Større prosjekt
uten en fasit

Refleksjon

Målet med denne forelesninga

- God innsikt i arbeidskravet
- Hva kreves av dere i arbeidskravet
- Git er ikke skummelt
 - Forståelig verktøy
 - Superkraft
- Hvordan jobbe mot en remote på GitHub
- Klar for å sette igang, første oppsett denne uka



Tilbakemelding

<https://nettskjema.no/a/vibe-check>

