



Git 102



Agenda

- Repetition
- Ångra saker i git...
- Mer repetition, repetition, repetition...
- Brancher
- Github
 - Skapa konto
 - Ladda upp ett repo
 - Skapa repo i github och ladda ner
- Ladda upp er blogg



Repetition

- > `git init`
 - > `git status`
 - > `git add`
 - > `git commit -m "..."`
-
- > `git log`
 - > `git diff`
 - > `git checkout ...`



Repetition

- Demo (gör ett projekt med flera commits, hoppa mellan commits)
- > `git config --global core.editor nano` (lättare editor än vi)
- Uppgift: Skapa en tom mapp, replikera det jag gjorde ungefär...



Frågor

- Vad är git? Använd gärna egna ord, din egna uppfattning....
- Vad gör git init?
- Vad är ett repository?
- Vad är en commit?



Frågor

- Ponera att du har ändrat i en fil, hur committar du ändringen?
- Vad är working directory?
- Vad är staging area?
- Vilka 3 tillstånd kan ett working directory ha?



Frågor

- Vad gör git log?
- Hur kan vi titta på skillnaden mellan olika commits?
- Hur hoppar vi fram och tillbaka i historiken?



Bonus Frågor

- Vad händer om jag gör git status utanför projekt?
- Om jag har en mapp inuti mitt projekt, hur committar jag förändringar där?
- Om jag har filer som inte git ska “tracka”, hur gör jag?
- Hur tar jag bort ett repository från ett projekt?



Få hjälp....

> `git <command> -help`

Ger hjälp med ett commando för att se vilka flaggor som finns

> `git help --all`

Visar alla subcommndon och översikt av dessa

Ångra saker i Git





Ångra saker i git...

Ibland behöver vi ta bort saker i git, tex en fil

```
> rm unnecessary.html
```

Tar bort fil från working dir men om vi stagat filen så ligger den fortfarande på staging area... Hmm....

```
> git rm unnecessary.html (tar bort från både work dir och staging area)
```

Ångra saker i git...

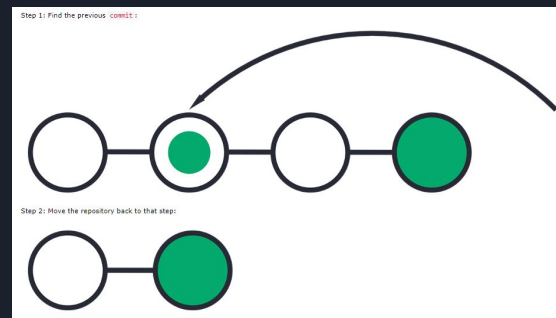
Om vi vill titta på tillstånd i working directory i en commit (Detached HEAD)


```
> git checkout 78ba34...
```

Om vi vill inte bara vill gå tillbaka och kika i tidigare historik utan ta bort commits

```
> git reset 78ba34...
```

Tar bort alla commits efter denna commit

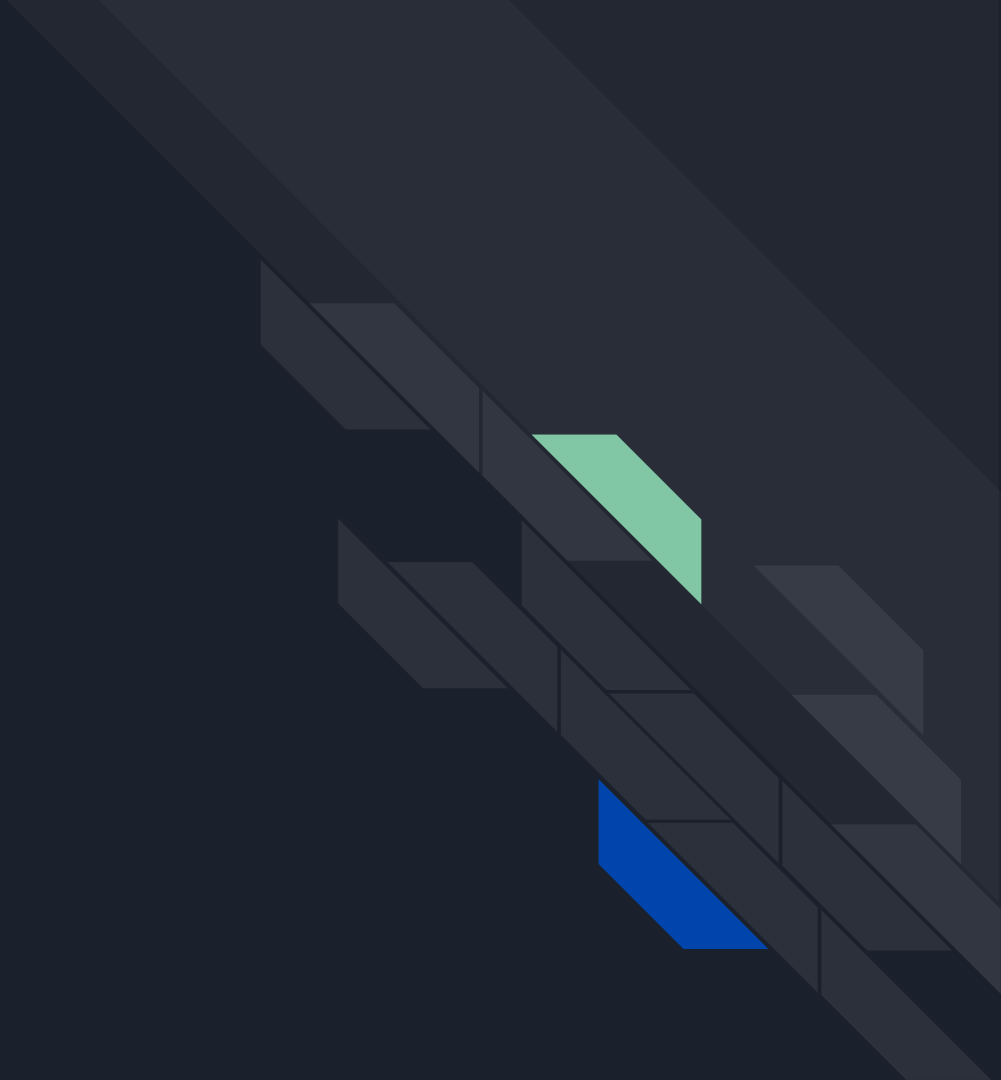




Övning Repetition (inlämningsuppgift)

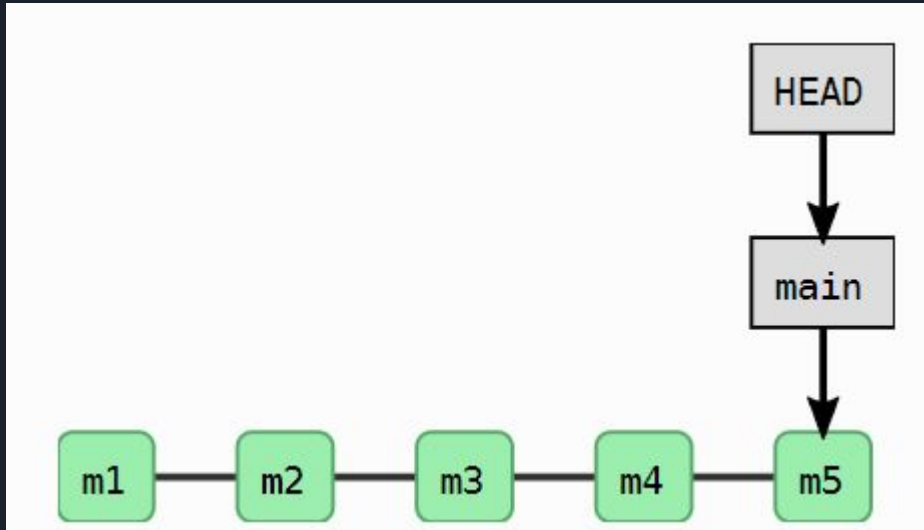
1. Gå in i ditt projekt (din blogg) och initialisera ett git repo
2. Kolla status (`git status`)
3. Committa allt du hittills har gjort (`git add -all`, `git commit`)
4. Gör en liten förändring
5. Kolla status (`git status`)
6. Committa din förändring (`git add`, `git commit`)
7. Titta på historiken (`git log`)
8. Hoppa fram och tillbaka mellan din första och senaste commit (`git checkout`)
9. Skapa en onödig fil och staga den (`git add`)
10. Ta bort filen från working directory och staging area (`git rm`)
11. Ta bort allt du gjort från och med din första commit (`git reset`)

Brancher



Brancher

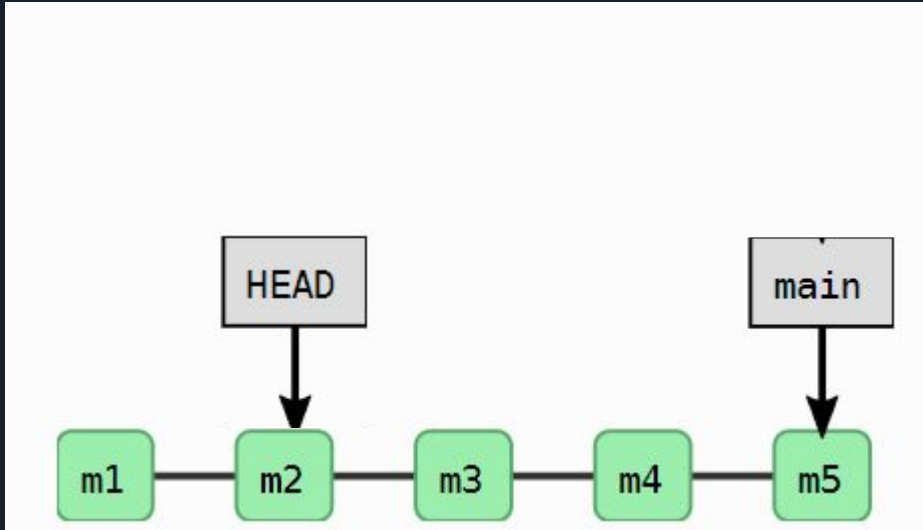
Hittills så har vi bara arbetat på en branch (main/master) - Demo



Brancher

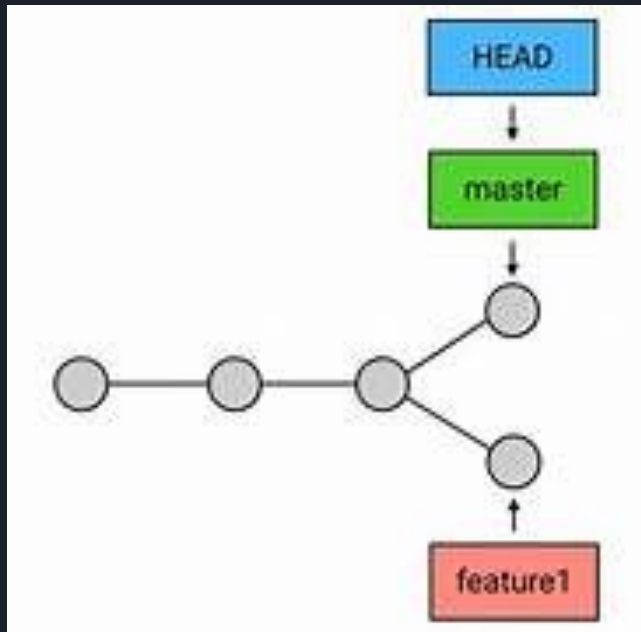
När vi har kört (`git checkout ...`) så har vårt HEAD hamnat i detached state

> `git checkout main` (för att återställa huvudet)



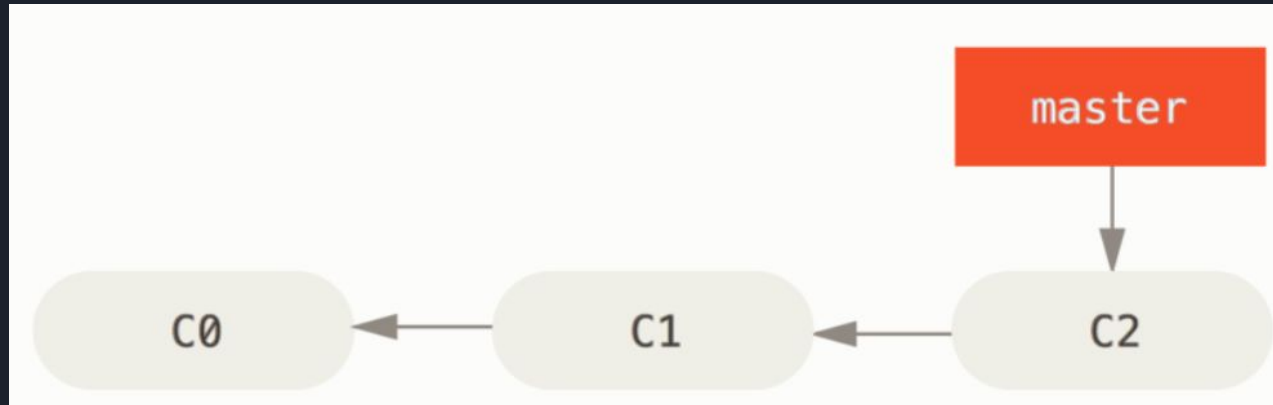
Brancher

Men vi skulle kunna skapa en en ny branch och jobba i olika riktningar i projektet - Demo



Brancher

Ponera att vi har följande commit historik

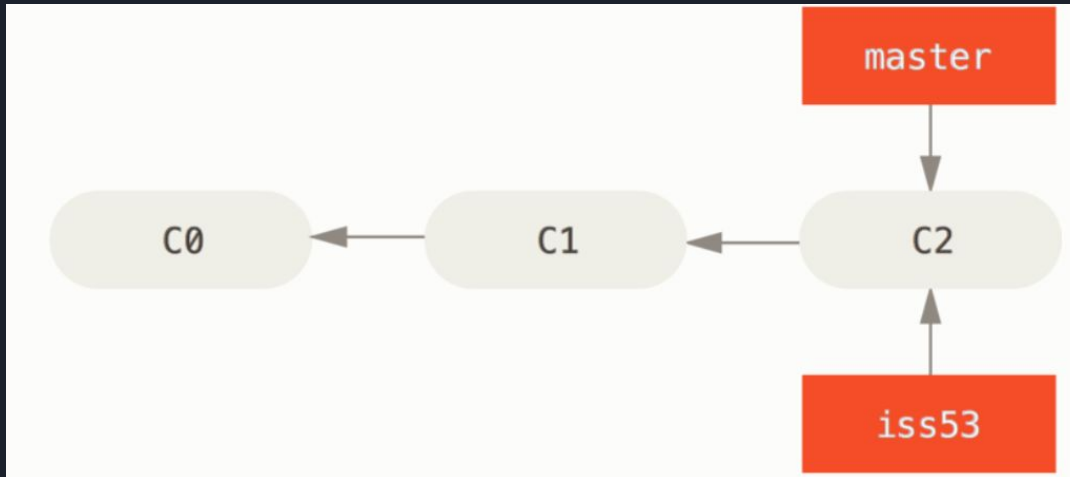


Brancher

Om vi vill skapa en ny branch från master gör vi det med

> `git branch iss53 master`

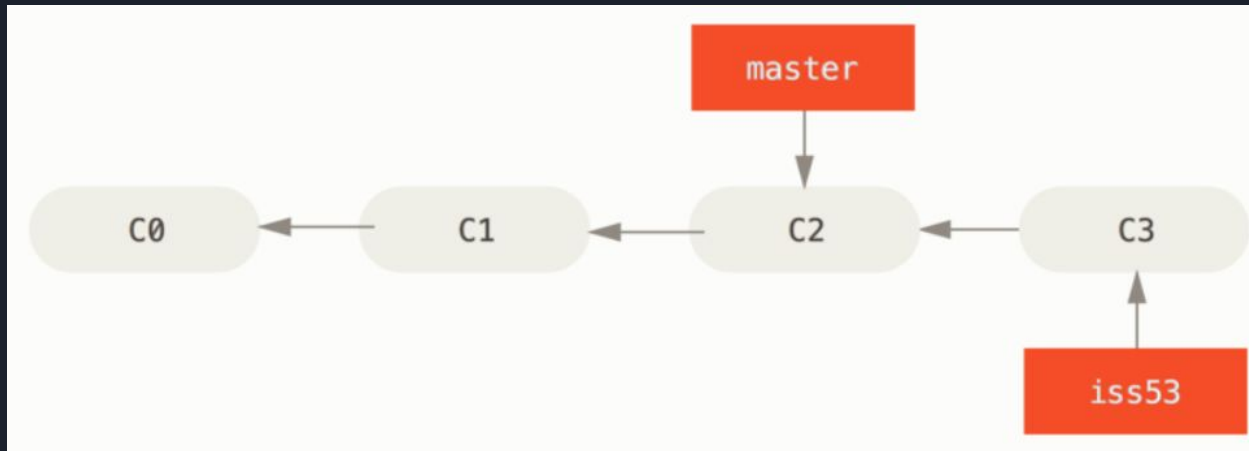
> `git switch iss53`



Brancher

Om vi nu skapar en ny commit gör vi det på den nya branchen (iss53)

> `git commit -a -m "created new footer"`

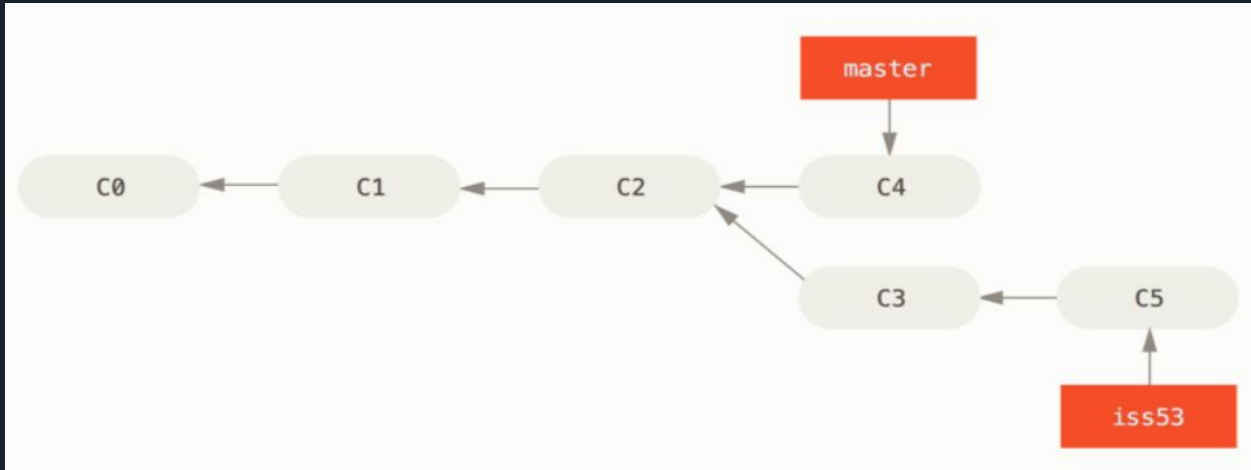


Brancher

Men ingenting stoppar oss från att hoppa tillbaka till master och committa

> `git switch master`

> `git commit -a -m "added navbar"`

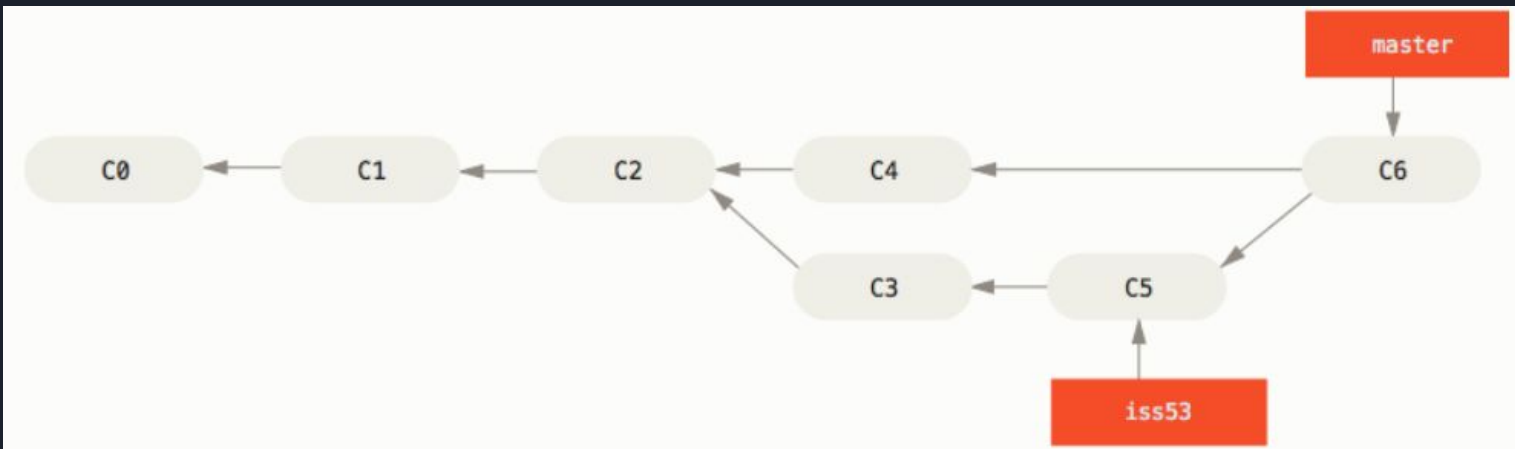


Brancher

Om vi nu vi slå ihop den nya featuren i iss53 branchen med master kan vi

> `git switch master`

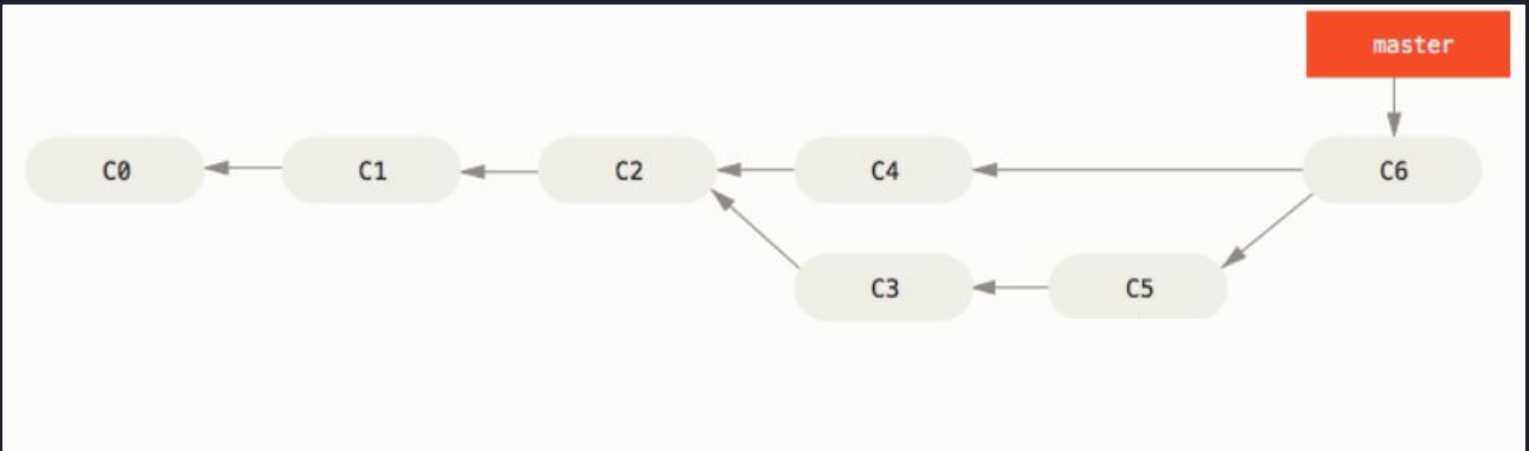
> `git merge iss53`



Brancher

Eftersom vi inte längre behöver iss53 branchen kan vi ta bort den

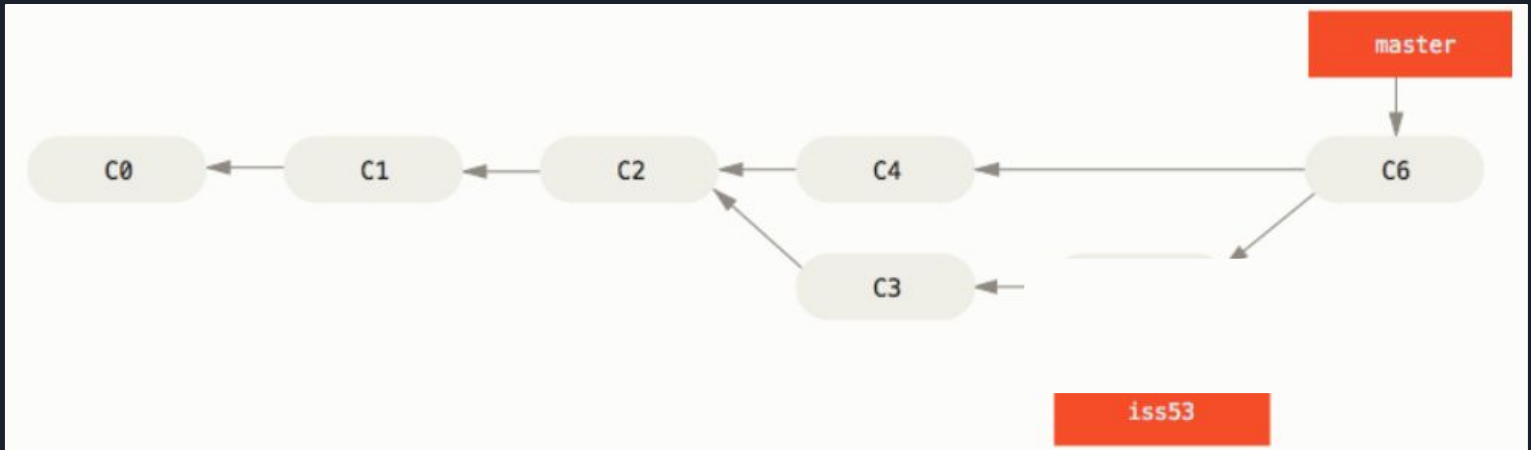
> `git branch -d iss53`



Brancher

För att få en bättre överblick över commit historik och brancher skapar vi ett alias

> `git config --global alias.graph "log --all --graph --decorate --oneline"`





Brancher





Brancher

Dom sub-kommandon som vi har sett gällande brancher är

- > `git branch ny-branch gammal-branch`

- > `git switch ny-branch`

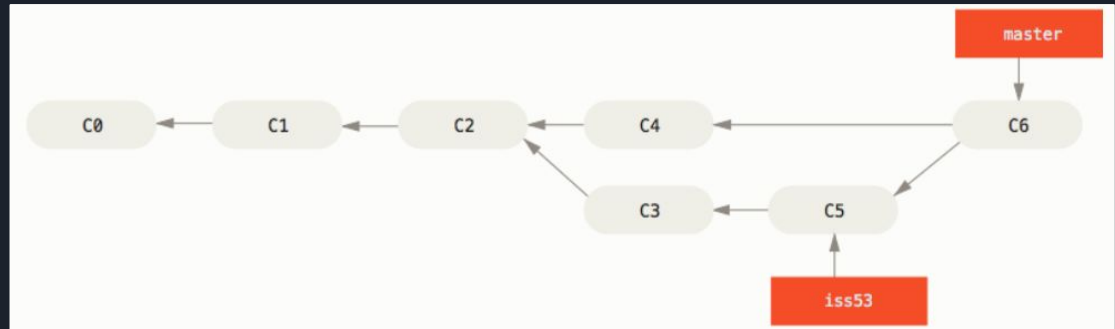
- > `git merge ny-branch`

- > `git branch -d ny-branch`

Brancher

Övning, (försök återskapa denna historik)

- > `git branch ny-branch gammal-branch`
- > `git switch ny-branch`
- > `git merge ny-branch`
- > `git branch -d ny-branch`



Github



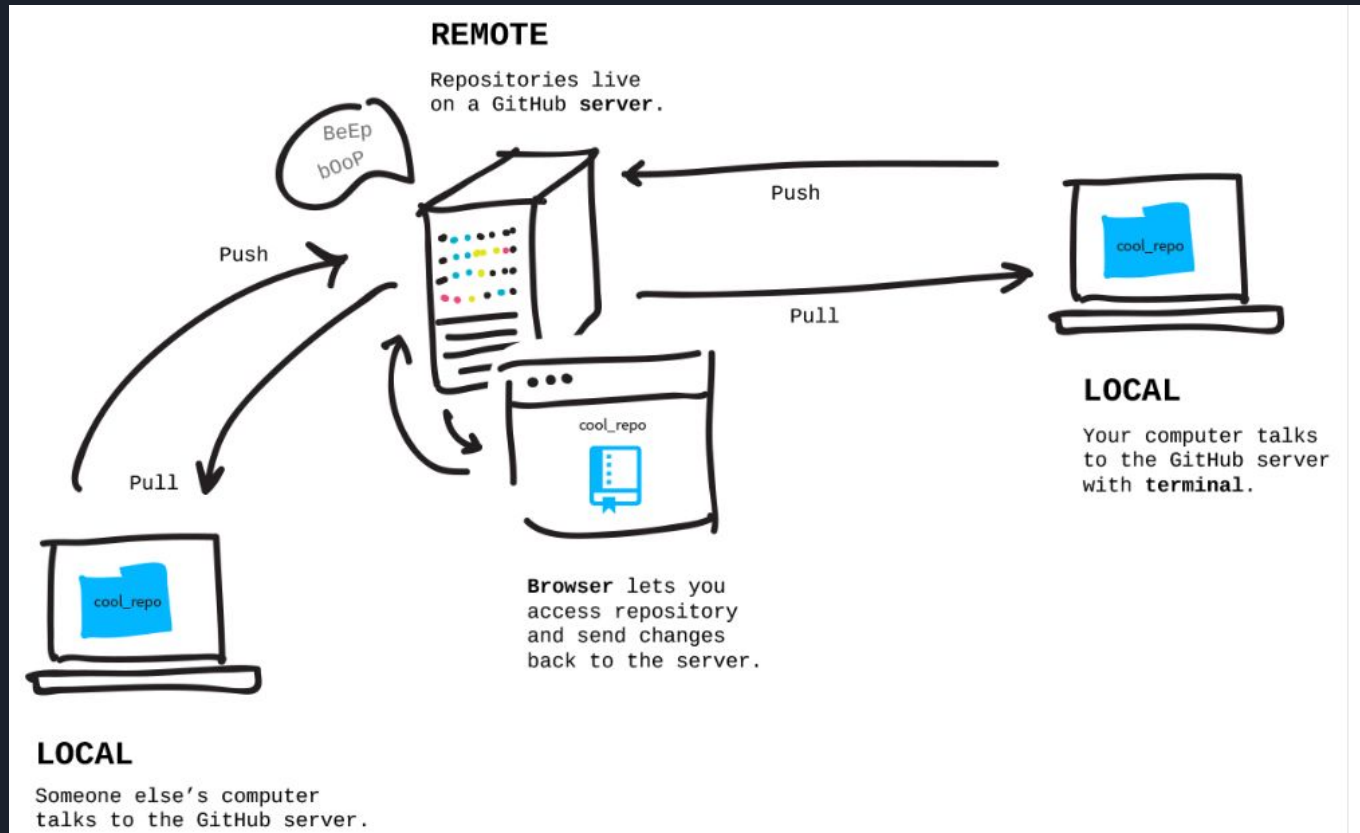


Github

Det finns ett koncept av en remote i Git

- En remote är kopia av repot på en annan plats i filsystem/eller på annan dator

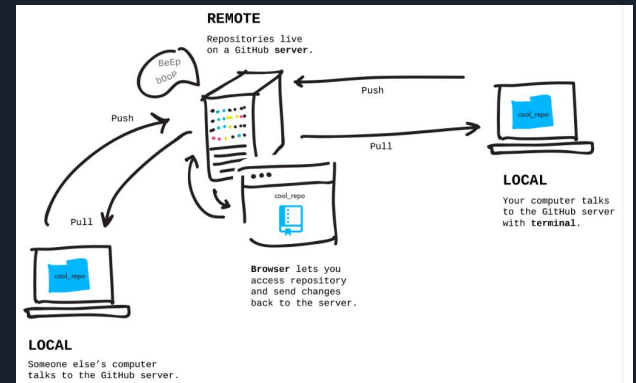
Github



Github

Demo

- Clona repo och committa 2 gngr
- Pusha commits
- Visa att commits skickats till github





Github

Demo (skapa remote på annan plats i **filsystem**)

- > `git init -bare` (skapar ett repo utan working directory)

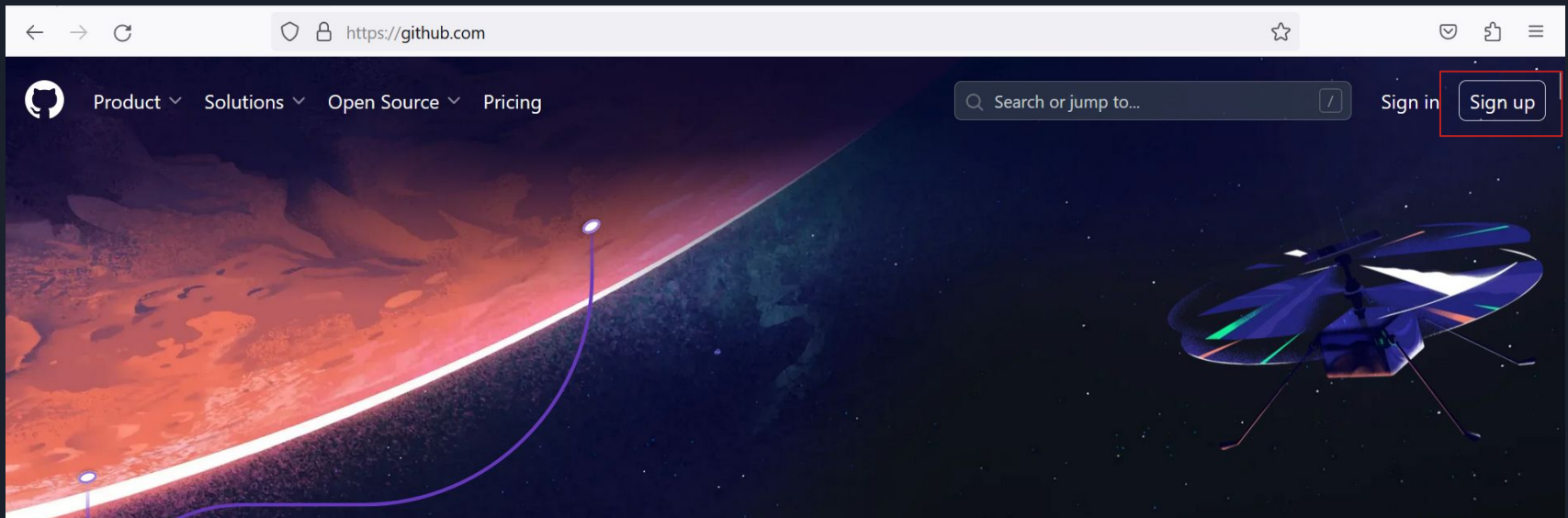
Inuti ditt ursprungliga projekt lägg till remote och skicka alla commits till remote

- > `git remote add min_origin_remote ~/sökväg/till/remote`
- > `git push min_origin_remote main`



GitHub

Skapa konto på Github



Github

Skapa konto på Github

The screenshot shows the GitHub Dashboard interface. At the top, there's a navigation bar with the GitHub logo, the word "Dashboard", a search bar with the placeholder "Type / to search", and several utility icons (plus, refresh, share, mail, profile). Below the navigation bar, the left sidebar contains the "Top Repositories" section with a search input "Find a repository..." and a list of repositories: "patriques82/react-native", "patriques82/webbbramverk-for-JS", and "patriques82/project". A green "New" button with a repository icon is highlighted with a red rectangle. The main content area is titled "Home" and includes a "Send feedback" link and a "Filter" button. A prominent announcement box titled "Updates to your homepage feed" with a close button (X) contains the text: "We've combined the power of the Following feed with the For you feed so there's one place to discover content on GitHub. There's improved filtering so you can customize your feed exactly how you like it, and a shiny new visual design. ✨". A "Learn more" link is at the bottom of the announcement.

Dashboard

Search: Type / to search

Top Repositories

Find a repository...

- patriques82/react-native
- patriques82/webbbramverk-for-JS
- patriques82/project

New

Home

Send feedback

Filter

Updates to your homepage feed

We've combined the power of the Following feed with the For you feed so there's one place to discover content on GitHub. There's improved filtering so you can customize your feed exactly how you like it, and a shiny new visual design. ✨

[Learn more](#)

Github

Ni kan kommer inte kunna göra något av det....

...or create a new repository on the command line

```
echo "# test" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:patriques82/test.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin git@github.com:patriques82/test.git
git branch -M main
git push -u origin main
```





Github

Why????

...or create a new repository on the command line

```
echo "# test" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:patriques82/test.git
git push -u origin main
```



...or push an existing repository from the command line

```
git remote add origin git@github.com:patriques82/test.git
git branch -M main
git push -u origin main
```



Github

Ni har registrerat er användare på github men git programmet på datorn är fortfarande inte kopplat!



Jag känner den killen!



Men vem F** är det!?

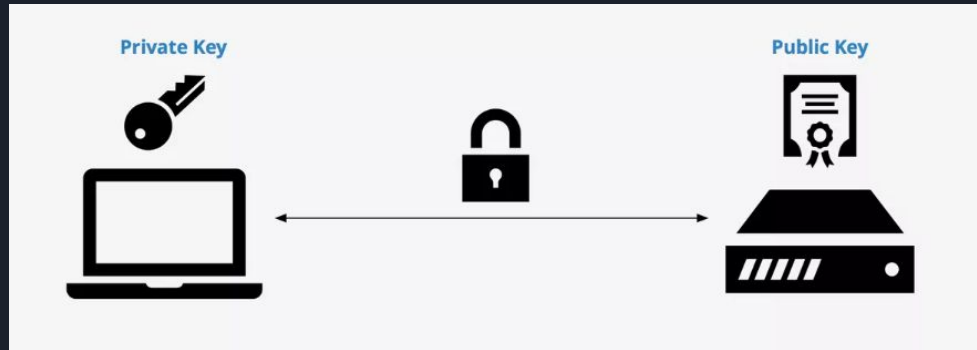
MINGW64:/c/Users/patri

```
patri@LAPTOP-QAG0D7PI MINGW64 ~  
$ git --version  
git version 2.33.1.windows.1
```

Github

SSH-Key

När ni har skapat ett konto behöver vi skapa ssh-nycklar och ge dem till github





Github

Skapa ssh-nyckel och copy-pasta den publika nyckeln till github

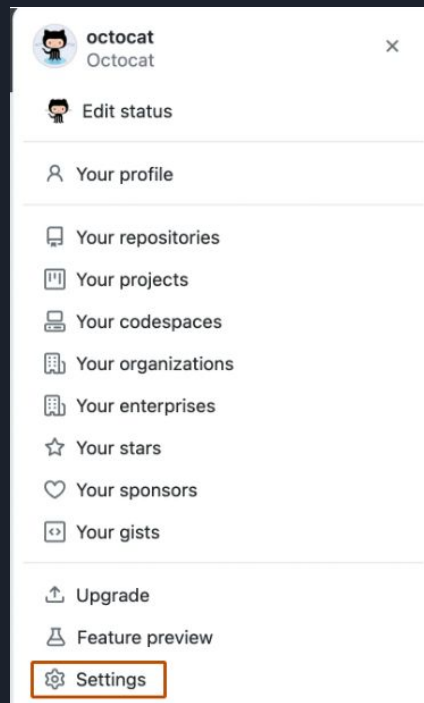
1. Kolla först om du redan har SSH-nyckel
 - a. > `ls -al ~/.ssh`
 - b. Om du redan har hoppa till steg 3
2. Skapa ssh-nyckel
 - a. > `ssh-keygen -t ed25519 -C "din_email@email.com"`
3. Kopiera publik ssh-nyckel
 - a. > `cat ~/.ssh/id_ed25519.pub` (kopiera)
4. Klistra in publik ssh-nyckel i github



Github

Klistra in publik ssh-nyckel på github

 SSH and GPG keys.





Github

Prova pusha nu till er trepo

> `git push -u origin main`





Github

Övning

- Gör en commit lokalt på i ert repo
- Pusha committen till Github

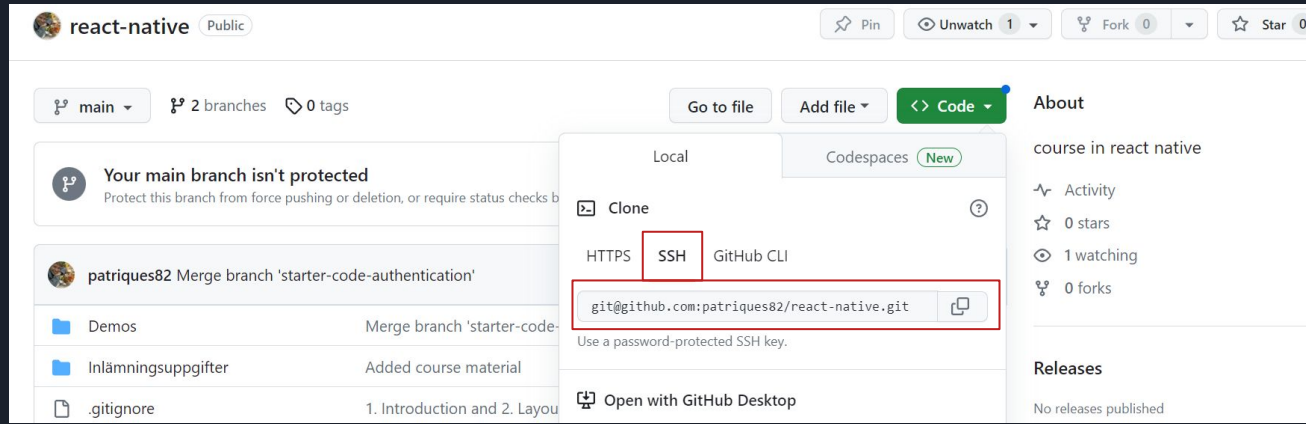
> `git push`

- Gå in på github och se att commit finns där

Github

Övning

- Ta bort projektet lokalt från dator
 - > `rmdir ert_project`
- Clona projektet till er dator och återställ
 - > `git clone git@github.com:er_användare/ert_project.git`



Ladda upp er blogg





Github pages

<https://pages.github.com/>



Läxa

Coderefinery

- <https://coderefinery.github.io/git-intro/branches/>
- <https://coderefinery.github.io/git-intro/remotes/>

Videos (follow along)

<https://www.youtube.com/watch?v=RGQj5yH7evk> (visar hur man jobbar med git från VSCode)

Om man känner: “jag fattar ingenting??”

- <https://www.youtube.com/watch?v=hrTQipWp6co> (del 1)
- <https://www.youtube.com/watch?v=1ibmWyt8hfw> (del 2)