

4. Versionshantering

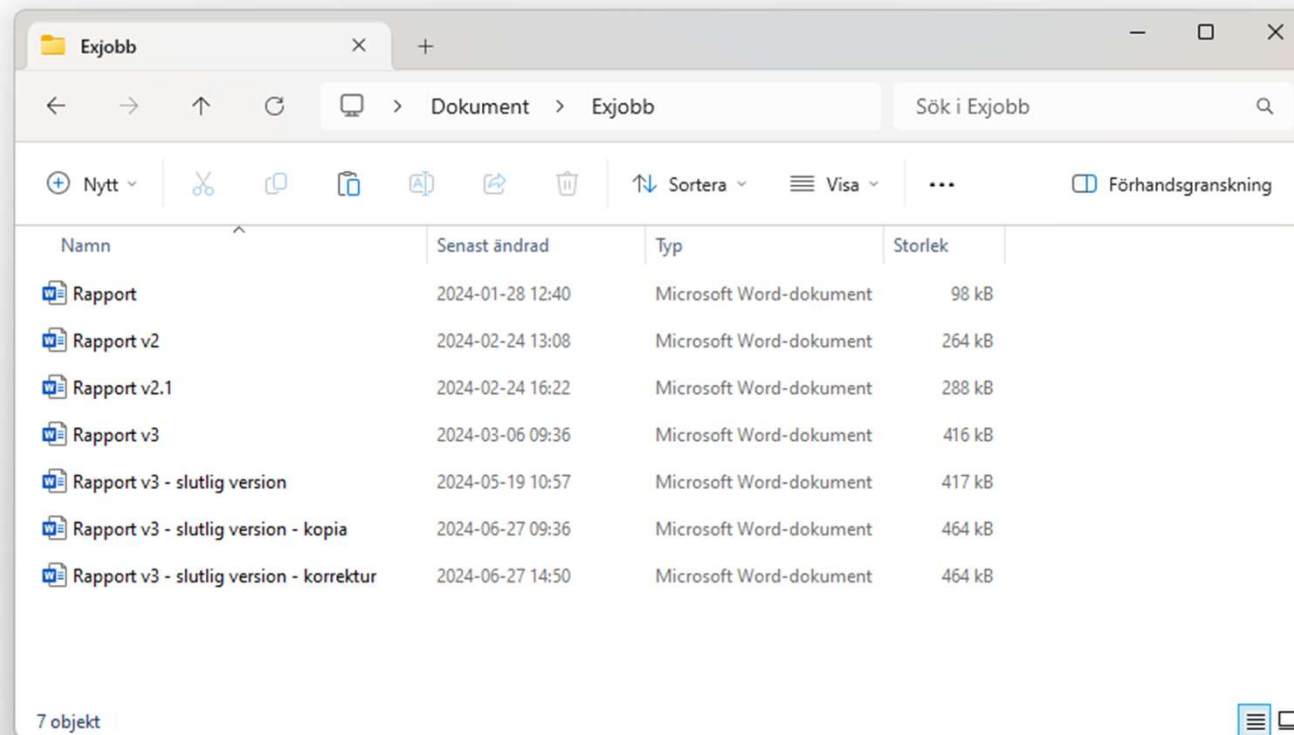
TDDE23 Funktionell och imperativ programmering, del 1

Peter Dalenius

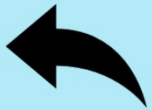
Institutionen för datavetenskap (IDA)

2024-09-19

Varför behövs versionshantering?



Fördelar med versionshantering



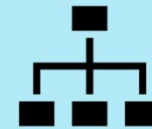
Backa till en tidigare version



Samarbeta flera personer i samma projekt



Spåra vem som gjorde vad och när



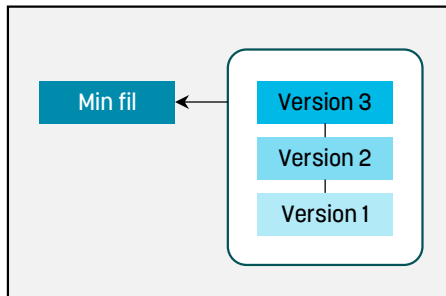
Arbeta med flera parallella versioner



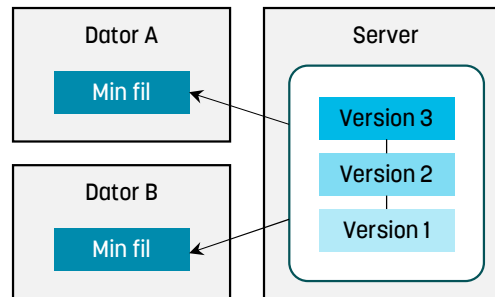
Ha en trygg säkerhetskopia

Olika strategier för versionshantering

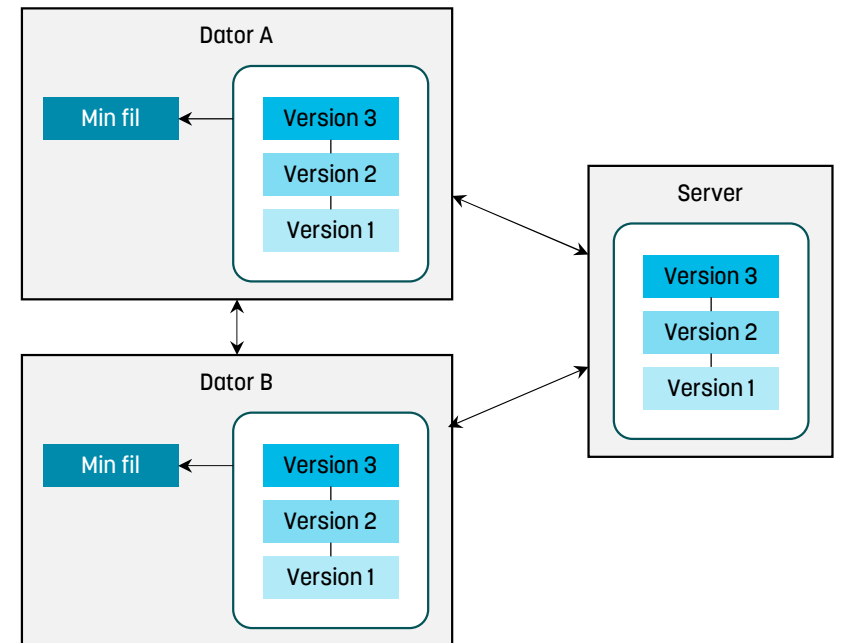
Lokal



Central



Distribuerad





- Distribuerat versionshanterings-system
- Fri och öppen källkod
- Började utvecklas 2005 av Linus Torvalds för att hantera utveckling av Linux-kärnan
- Används av 95% av alla professionella utvecklare (enligt en enkät av Stack Overflow 2022)



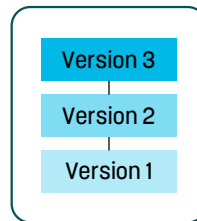
- Webbaserad central lagringsplats för många projekt som använder Git
- Lanserades 2008
- Används både för öppen källkod och för kommersiella projekt
- Över 100 miljoner utvecklare
- Över 400 miljoner projekt



- Webbaserad utvecklingsplattform
- Började utvecklas 2011
- Över 30 miljoner användare
- Linköpings universitet har en egen installation på adressen gitlab.liu.se som vi kommer använda i kursen

Grundläggande begrepp

Repo
Repository
Versionsarkiv



*Datastruktur med alla
filer och all ändrings-
historik i ett projekt*

Grundläggande kommandon i git

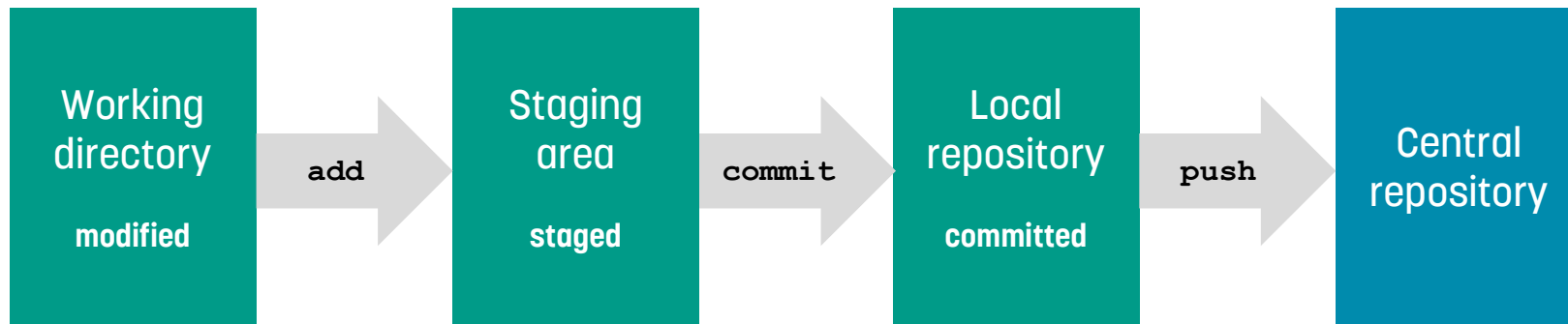
Skapa ett tomt lokalt repo

```
lnx-v-tl-p-1 [~/lab]> mkdir myproject
lnx-v-tl-p-1 [~/lab]> cd myproject
lnx-v-tl-p-1 [~/lab/myproject]> git init
Initialised empty Git repository in /home/petjo96/lab/myproject/.git/
lnx-v-tl-p-1 [~/lab/myproject]> ls
total 0
lnx-v-tl-p-1 [~/lab/myproject]> ls -a
total 34
drwx----- 3 petjo96 employee-liu.se  3 Jun 27 12:22 .
drwx----- 9 petjo96 employee-liu.se 28 Jun 27 12:22 ..
drwx----- 7 petjo96 employee-liu.se 10 Jun 27 12:22 .git
lnx-v-tl-p-1 [~/lab/myproject]>
```

Klona ett repo från en server

```
lnx-v-tl-p-1 [~/lab]> git clone https://gitlab.liu.se/tdde23-2023/tdde23-2023-labbar-12-d1-a-10.git
Cloning into 'tdde23-2023-labbar-12-d1-a-10'...
Username for 'https://gitlab.liu.se': petjo96
Password for 'https://petjo96@gitlab.liu.se':
remote: Enumerating objects: 157, done.
remote: Total 157 (delta 0), reused 0 (delta 0), pack-reused 157
Receiving objects: 100% (157/157), 572.60 KiB | 19.74 MiB/s, done.
Resolving deltas: 100% (65/65), done.
lnx-v-tl-p-1 [~/lab]> cd tdde23-2023-labbar-12-d1-a-10/
lnx-v-tl-p-1 [~/lab/tdde23-2023-labbar-12-d1-a-10]> ls
total 19
drwx----- 2 petjo96 employee-liu.se  3 Jun 27 12:46 lab1
drwx----- 2 petjo96 employee-liu.se  3 Jun 27 12:46 lab2
drwx----- 2 petjo96 employee-liu.se  4 Jun 27 12:46 lab3
drwx----- 2 petjo96 employee-liu.se  7 Jun 27 12:46 lab4
drwx----- 2 petjo96 employee-liu.se 10 Jun 27 12:46 lab5
lnx-v-tl-p-1 [~/lab/tdde23-2023-labbar-12-d1-a-10]>
```

Olika tillstånd för filer i ett projekt



Lägga till filer till ett repo

```
lnx-v-tl-p-1 [~/lab/myproject]> ls
total 1
-rwx----- 1 petjo96 employee-liu.se 91 Jun 27 13:06 intro.py
lnx-v-tl-p-1 [~/lab/myproject]> git status
On branch master
No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    intro.py

nothing added to commit but untracked files present (use "git add" to track)
lnx-v-tl-p-1 [~/lab/myproject]> git add intro.py
lnx-v-tl-p-1 [~/lab/myproject]> git commit -m 'Added introduction'
[master (root-commit) dfe3f75] Added introduction
 1 file changed, 4 insertions(+)
 create mode 100644 intro.py
lnx-v-tl-p-1 [~/lab/myproject]> git push
```

Git-kommandon som vi har sett hittills

Kommando	Vad gör det?
<code>git init</code>	Skapar ett nytt tomt repo i aktuell katalog
<code>git clone</code>	Skapar en lokal kopia av ett repo från en server
<code>git status</code>	Visar statusen för ett lokalt repo
<code>git add</code>	Förbereder en eller flera filer för att sparas i det lokala repot
<code>git commit</code>	Sparar en snapshot av alla förberedda filer i det lokala repot
<code>git push</code>	Skickar in det lokala repots ändringar till en server

T tdde23-2024-labbar-20-d1-c-03

master tdde23-2024-labbar-20-d1-c-03 / +



Updated .gitignore

Jonas Kvarnström authored 1 year ago



Forked from TDDE23-24 / python-tdde23-base

Up to date with the upstream repository.

Name	Last commit
lab1	Updates
lab2	Updates
lab3	Fixed merge conflict
lab4	Fix execution flag
lab5	Added files
.gitignore	Updated .gitignore

History

Find file

Edit

Code

Clone with SSH

git@gitlab.liu.se:tdde23-2024/td

Clone with HTTPS

https://gitlab.liu.se/tdde23-202

Open in your IDE

Visual Studio Code (SSH)

Visual Studio Code (HTTPS)

IntelliJ IDEA (SSH)

IntelliJ IDEA (HTTPS)

Download source code

zip

tar.gz

tar.bz2

tar

Logga in på
<https://gitlab.liu.se/> och
kopiera adressen för att
klona ditt repo.

Mer om git som kan vara bra att veta

När är det dags att göra commit?

- Undvik stora commits som innehåller många olika saker, särskilt i projekt med många utvecklare.
- Gör en commit när du har gjort en ändring som du lätt kan förklara.
- Skriv ett tydligt commit-meddelande så att du själv och dina kollegor förstår vad som ändrats.

```
git commit -m "Refactor subsystem X for readability"  
git commit -m "Update getting started documentation"  
git commit -m "Remove deprecated methods"
```

Läs mer:
<https://cbea.ms/git-commit/>

Hur skriver man ett bra commit-meddelande?

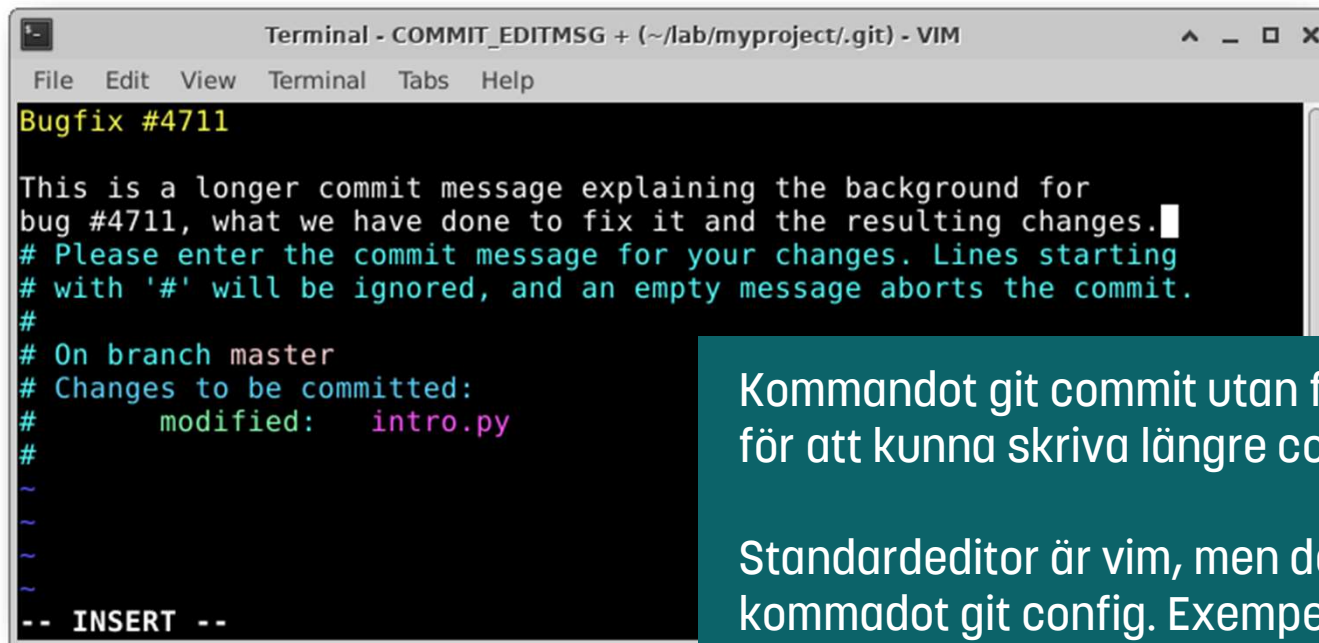
- Följ standard för företaget eller projektet
- Använd imperativ form (uppmaning)
- Inkludera typ av ändring
- Var kort och koncis
- Vad och varför?

Buggfix, uppdatering,
ny funktion, refaktorisering, ...

Skriv kort, helst
inte alls.

Varför har ändringen gjorts?
Vilka effekter får ändringen?
Vilket behov uppfyller ändringen?

Längre commit-meddelande



```
Terminal - COMMIT_EDITMSG + (~/.lab/myproject/.git) - VIM
File Edit View Terminal Tabs Help
Bugfix #4711
This is a longer commit message explaining the background for
bug #4711, what we have done to fix it and the resulting changes.
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# Changes to be committed:
#   modified:   intro.py
#
~
~
~
-- INSERT --
```

Kommandot `git commit` utan flaggan `-m` öppnar en editor för att kunna skriva längre commit-meddelanden.

Standardeditor är vim, men det kan ändras genom kommandot `git config`. Exempel:

```
git config --global core.editor "emacs"
```

Versionshistorik



Visa tidigare commit

```
lnx-v-tl-p-3 [~/lab/myproject]> git log
commit 42149581223d3536e5ded56d1bba4f25cedd66f3 (HEAD -> master)
Author: Peter Dalenius <peter.dalenius@liu.se>
Date: Thu Sep 12 11:29:17 2024 +0200

    Change background color

commit 15902b515eebd1e83c36e1784f738a0af4fdbb97
Author: Peter Dalenius <peter.dalenius@liu.se>
Date: Mon Aug 12 11:21:43 2024 +0200

    Bugfix #4711

commit dfe3f756917a259b1644976637d2f47f5bcf697b
Author: Peter Dalenius <peter.dalenius@liu.se>
Date: Thu Jun 27 13:07:11 2024 +0200

    Added introduction
```

Gå tillbaka till en tidigare commit

```
lnx-v-tl-p-3 [~/lab/myproject]> git checkout 15902b515eebd1e83c36e1784f738a0af4fdbb97  
Note: switching to '15902b515eebd1e83c36e1784f738a0af4fdbb97'.
```

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using `-c` with the switch command. Example:

```
git switch -c <new-branch-name>
```

Or undo this operation with:

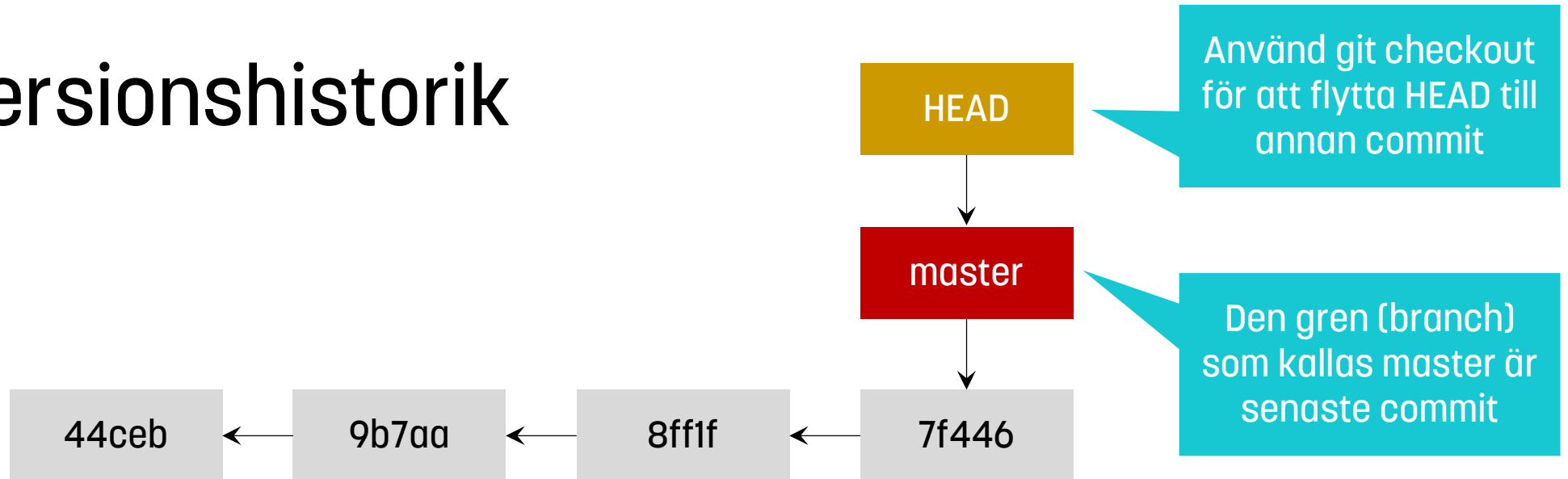
```
git switch -
```

```
HEAD is now at 15902b5 Bugfix #4711
```

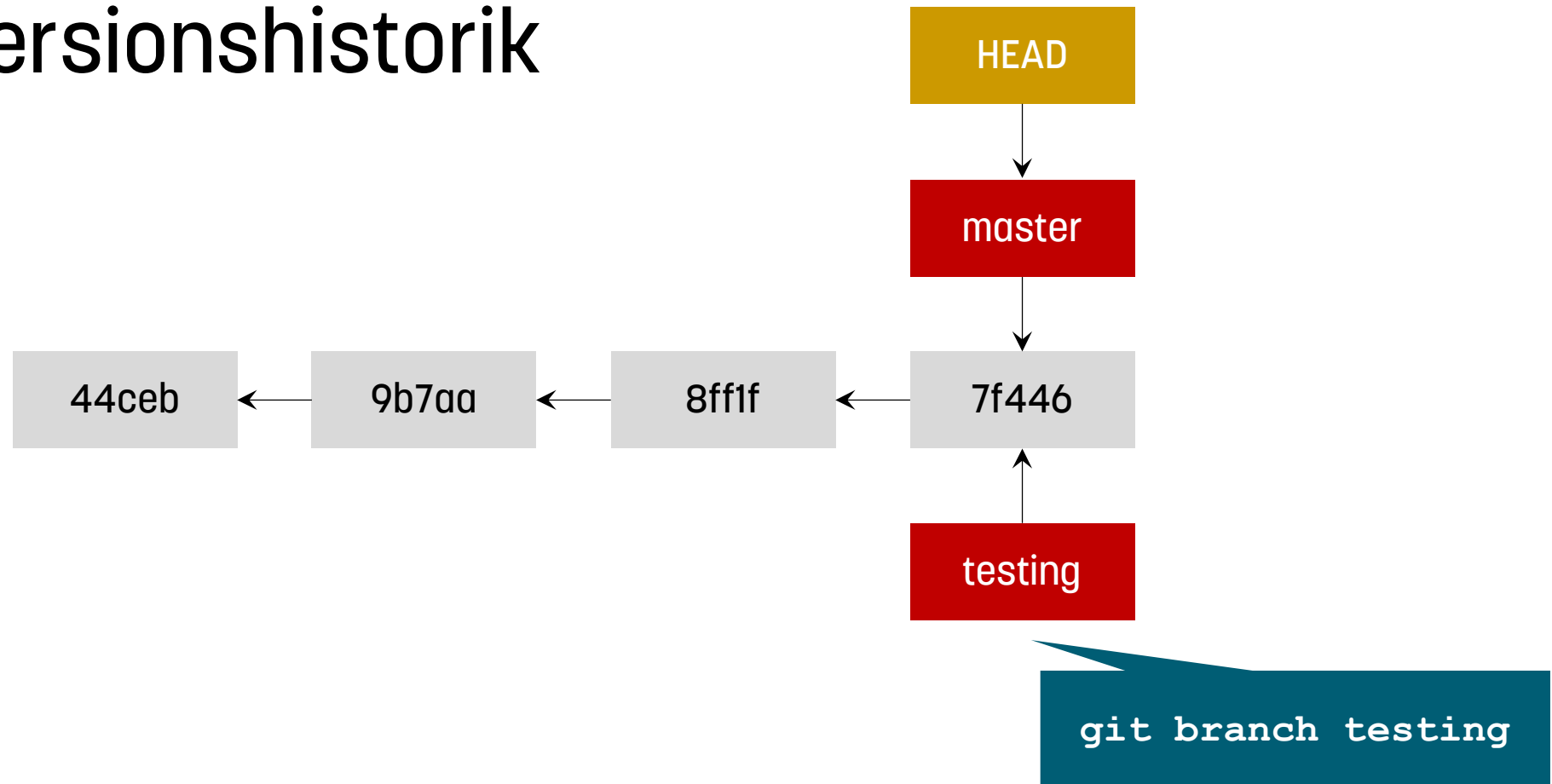
Återgå till senaste commit

```
lnx-v-tl-p-3 [~/lab/myproject]> git checkout master  
Previous HEAD position was 15902b5 Bugfix #4711  
Switched to branch 'master'
```

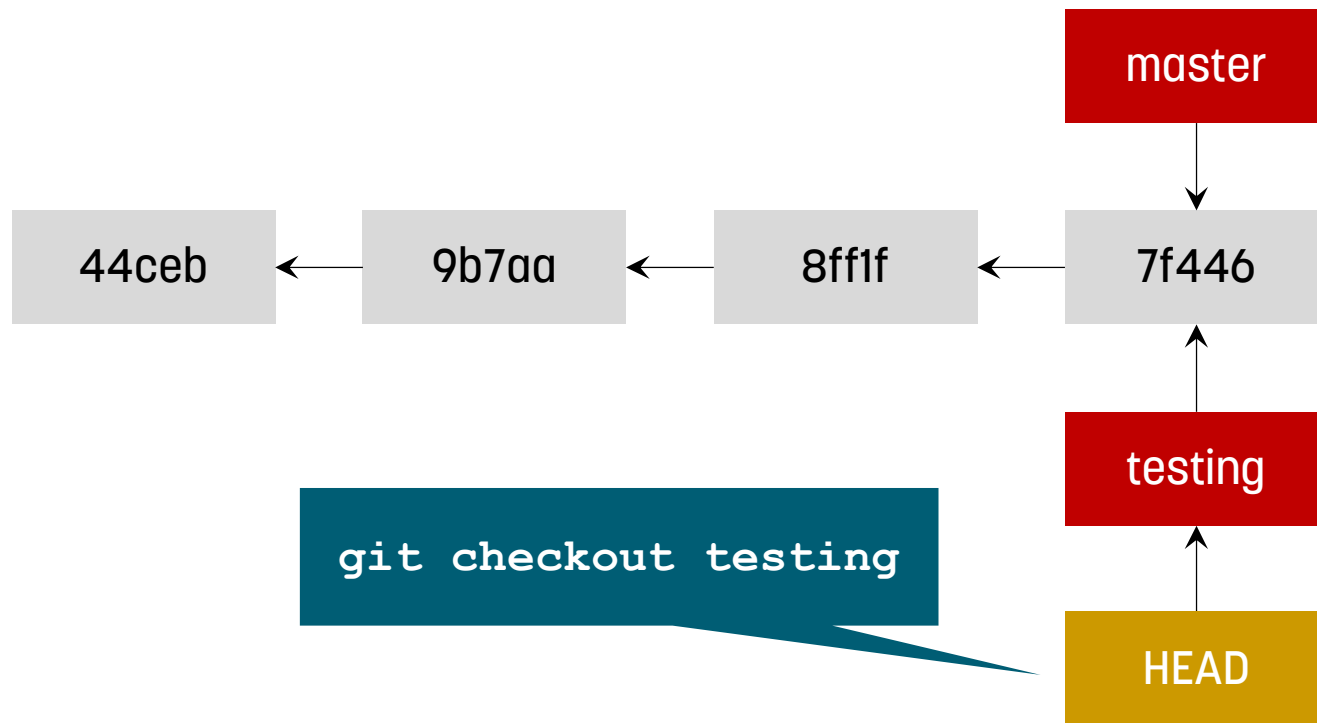
Versionshistorik



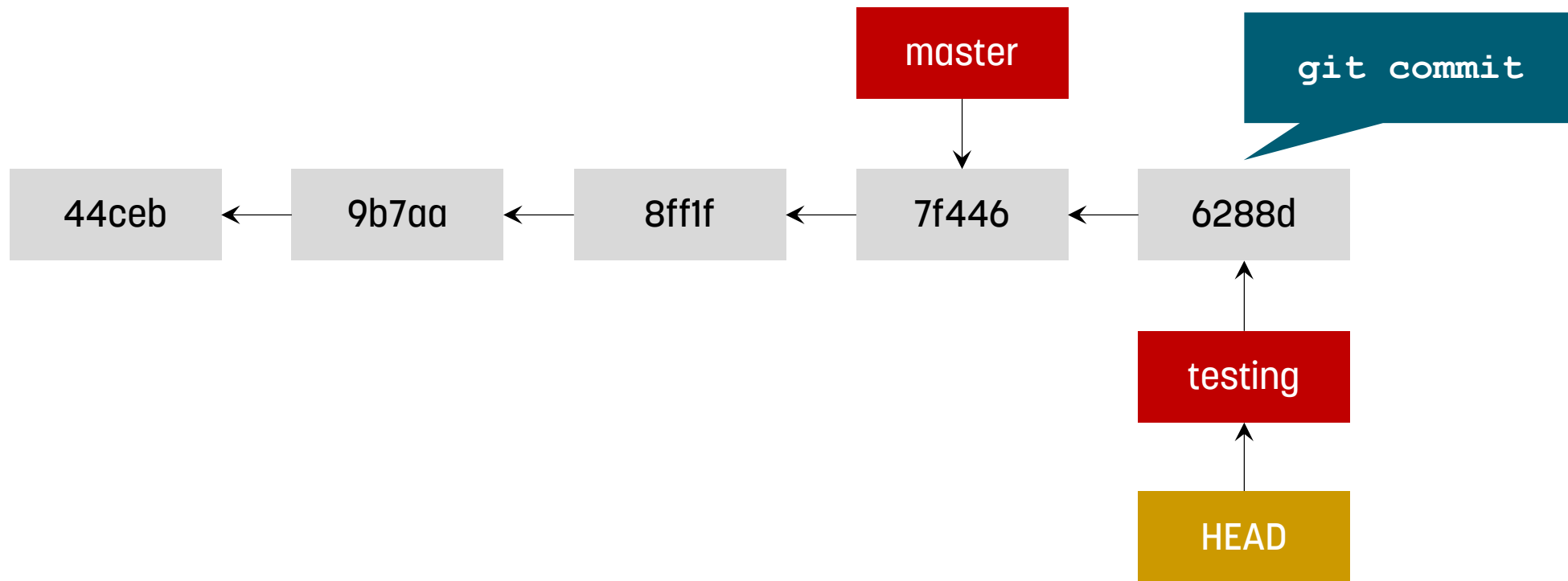
Versionshistorik



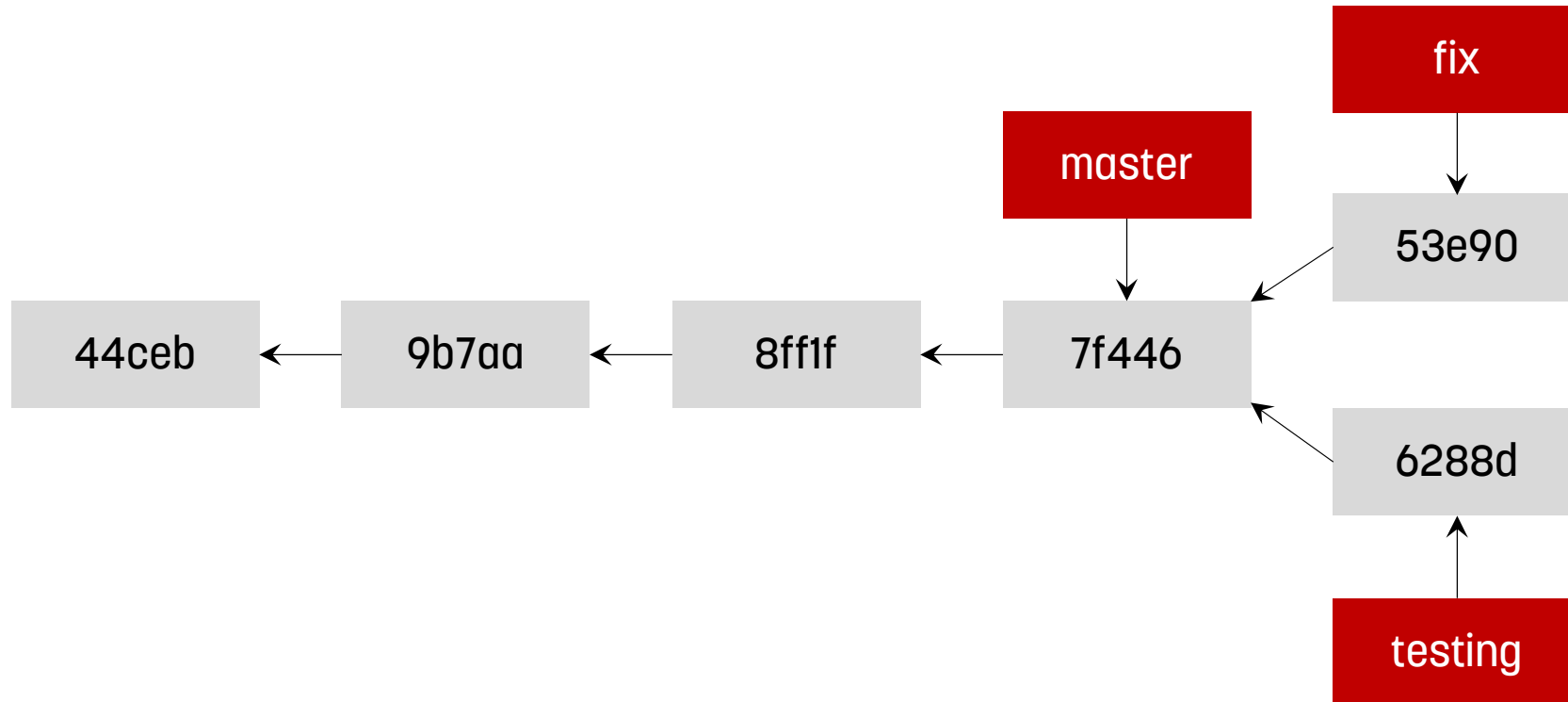
Versionshistorik



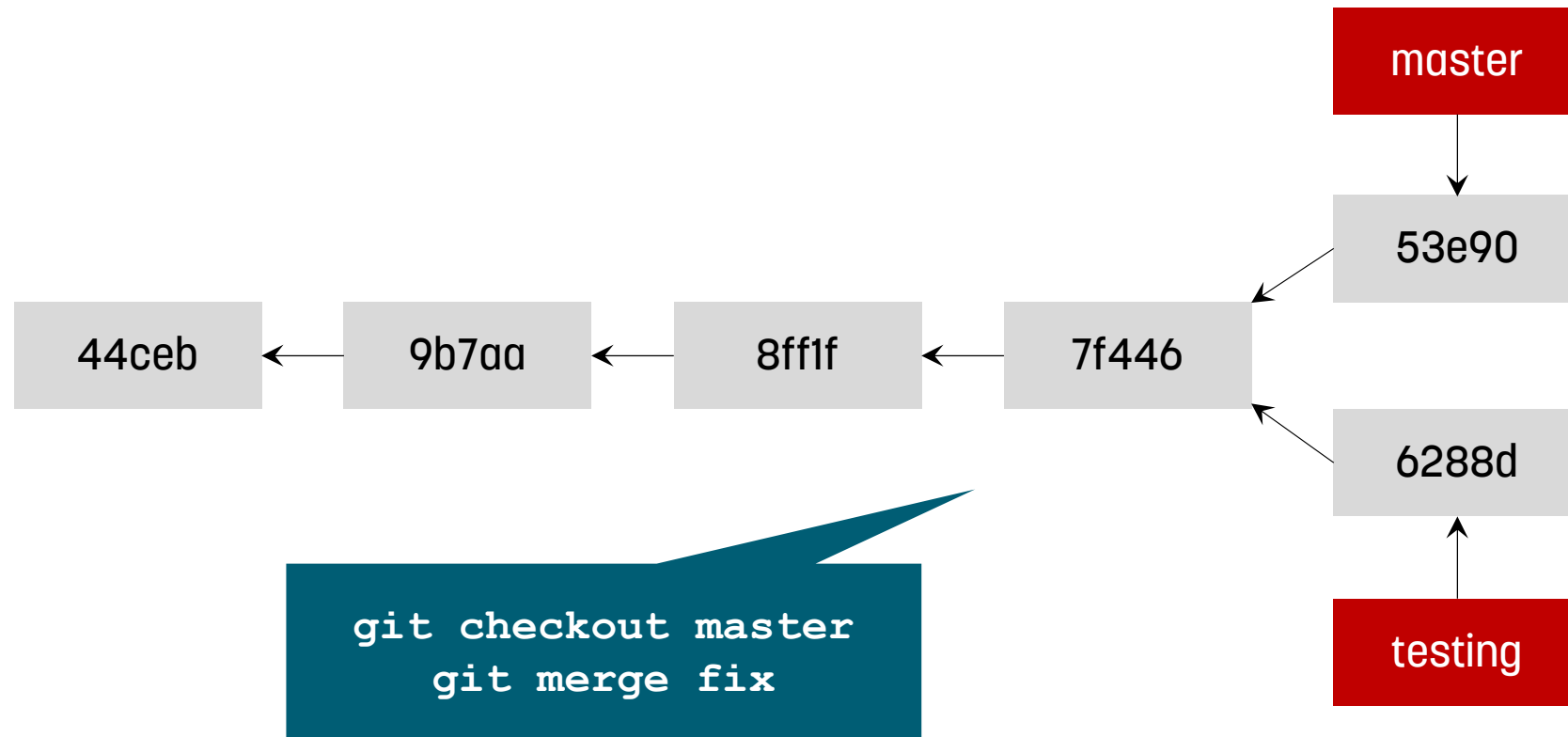
Versionshistorik



Versionshistorik



Versionshistorik



Ytterligare git-kommandon

Kommando	Vad gör det?
<code>git log</code>	Visa alla commits
<code>git checkout</code>	Byt till en annan gren eller commit
<code>git branch</code>	Skapa en ny gren
<code>git switch</code>	Nyare kommando som funkar ungefär som checkout
<code>git merge</code>	Slå samman två grenar

About

Documentation

Reference

Book

Videos

External Links

Downloads

Community

This book is available in [English](#).

Full translation available in

[azərbaycan dili](#),

[български език](#),

[Deutsch](#),

[Español](#),

[Français](#),

[Ελληνικά](#),

[日本語](#),

[한국어](#),

[Nederlands](#),

[Русский](#),

[Slovenščina](#),

[Tagalog](#),

[Українська](#)

[简体中文](#),

Partial translations available in

[Čeština](#),

[Македонски](#),

[Polski](#),

[Српски](#),

[Ўзбекча](#),

[繁體中文](#),

Translations started for

Book

The entire Pro Git book, written by Scott Chacon and Ben Straub and published by Apress, is available here. All content is licensed under the [Creative Commons Attribution Non Commercial Share Alike 3.0 license](#). Print versions of the book are available on [Amazon.com](#).

The version found here has been updated with corrections and additions from [hundreds of contributors](#). If you see an error or have a suggestion, patches and issues are welcome in its [GitHub repository](#).



2nd Edition (2014)

Download Ebook



1. Getting Started

- 1.1 About Version Control
- 1.2 A Short History of Git
- 1.3 What is Git?
- 1.4 The Command Line
- 1.5 Installing Git
- 1.6 First-Time Git Setup
- 1.7 Getting Help
- 1.8 Summary

2. Git Basics

- 2.1 Getting a Git Repository
- 2.2 Recording Changes to the Repository
- 2.3 Viewing the Commit History
- 2.4 Undoing Things
- 2.5 Working with Remotes
- 2.6 Tagging
- 2.7 Git Aliases
- 2.8 Summary

3. Git Branching

- 3.1 Branches in a Nutshell
- 3.2 Basic Branching and Merging
- 3.3 Branch Management
- 3.4 Branching Workflows

En bra introduktion till Git finns
online på adressen
<https://git-scm.com/book>

Fler fördjupningstips



Learn Git in 15 Minutes (Colt Steele)

<https://www.youtube.com/watch?v=USjZcfj8yxE>



Git Tutorial for Beginners: Learn Git in 1 Hour (Programming with Mosh)

<https://www.youtube.com/watch?v=8JJ101D3knE>