

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 MEMORIZE 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.



omega point.



GIT 101

Niklas Hätty

Ziad Patrous

Goals

Familiarize yourself with basic key concepts of git and be able to use it as a tool in your future assignment

Course plan

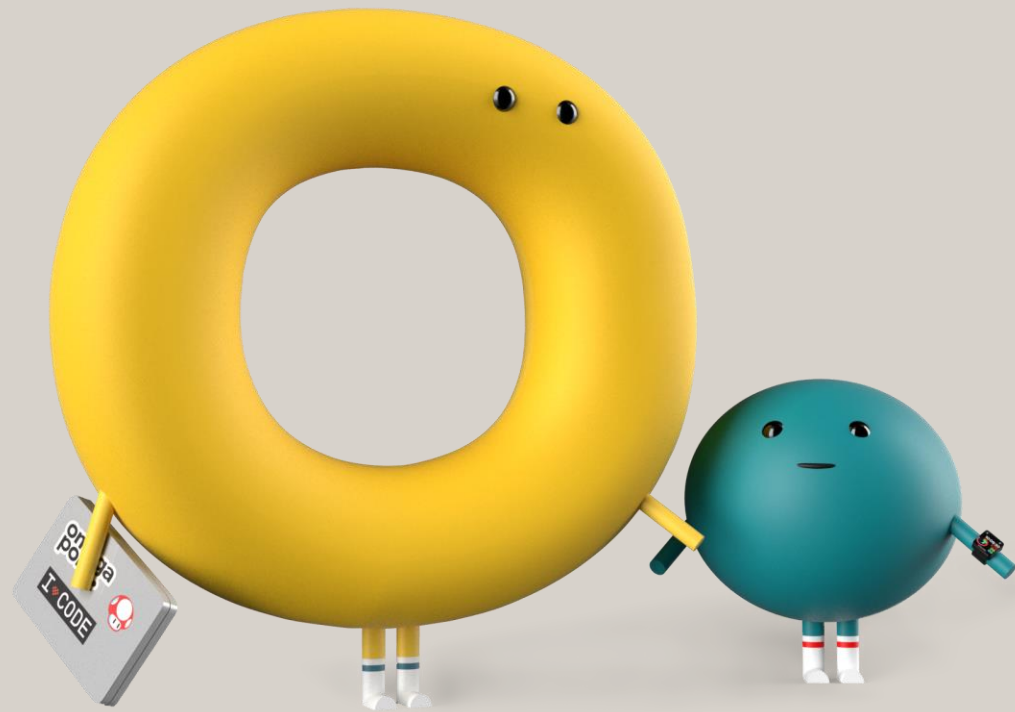
- Part 1 – Basic Git
- Part 2 – Lab
- Part 3 – Branching Strategies
- Part 4 – Scenarios

Preliminary schedule

- 08:30 – 09:00 – Introduction and Git fundamentals
- 10 min break
- 09:10 – 10:30 Github lab
- 15 min break
- 10:45 - 12:00 Branching strategies, and real life scenarios
- 12:00 - ~12:15 Questions? Done :)

Lite ordningsuppmaningar för en digital kurs

- Använd teams-funktioner!
- Fråga gärna, men lättast blir nog om ni använder handuppräkningsfunktionen i teams.
- Vi kommer att hålla paustider så ni vet när ni behöver vara tillbaka
- Använd gärna kameran :)



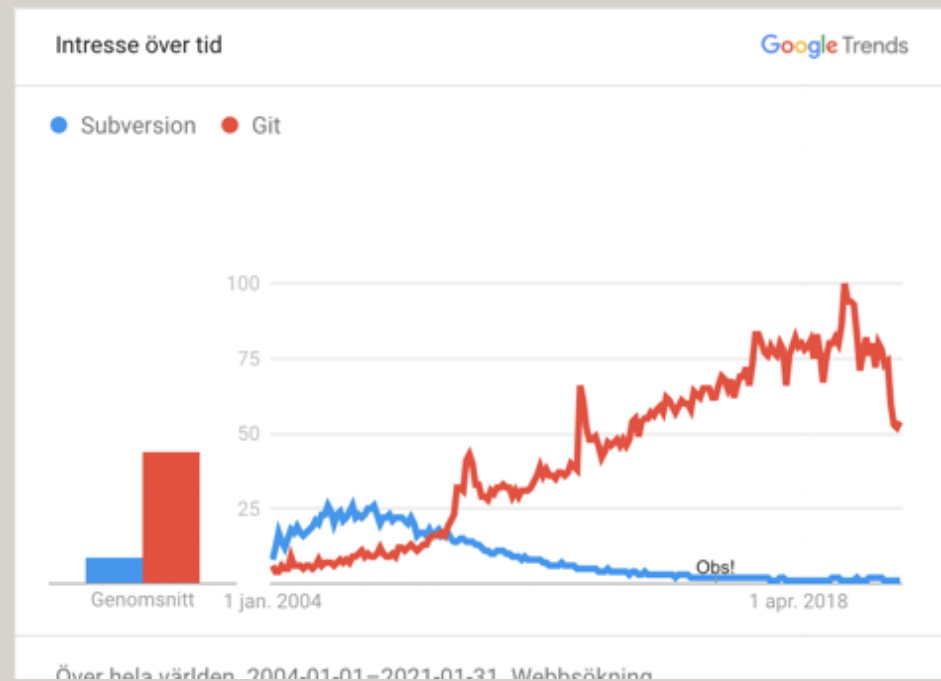
What is git?

What is git?

- An open source version control system (VCS)
- A open source version control system (VCS) created by Linus Torvalds
- It is free and is the foundation of many services such as Github, Bitbucket and GitLab
- We use git to:
 - manage changes to code over time
 - keep history of **all** modifications so that reverting mistakes is easier
 - collaboration between developers

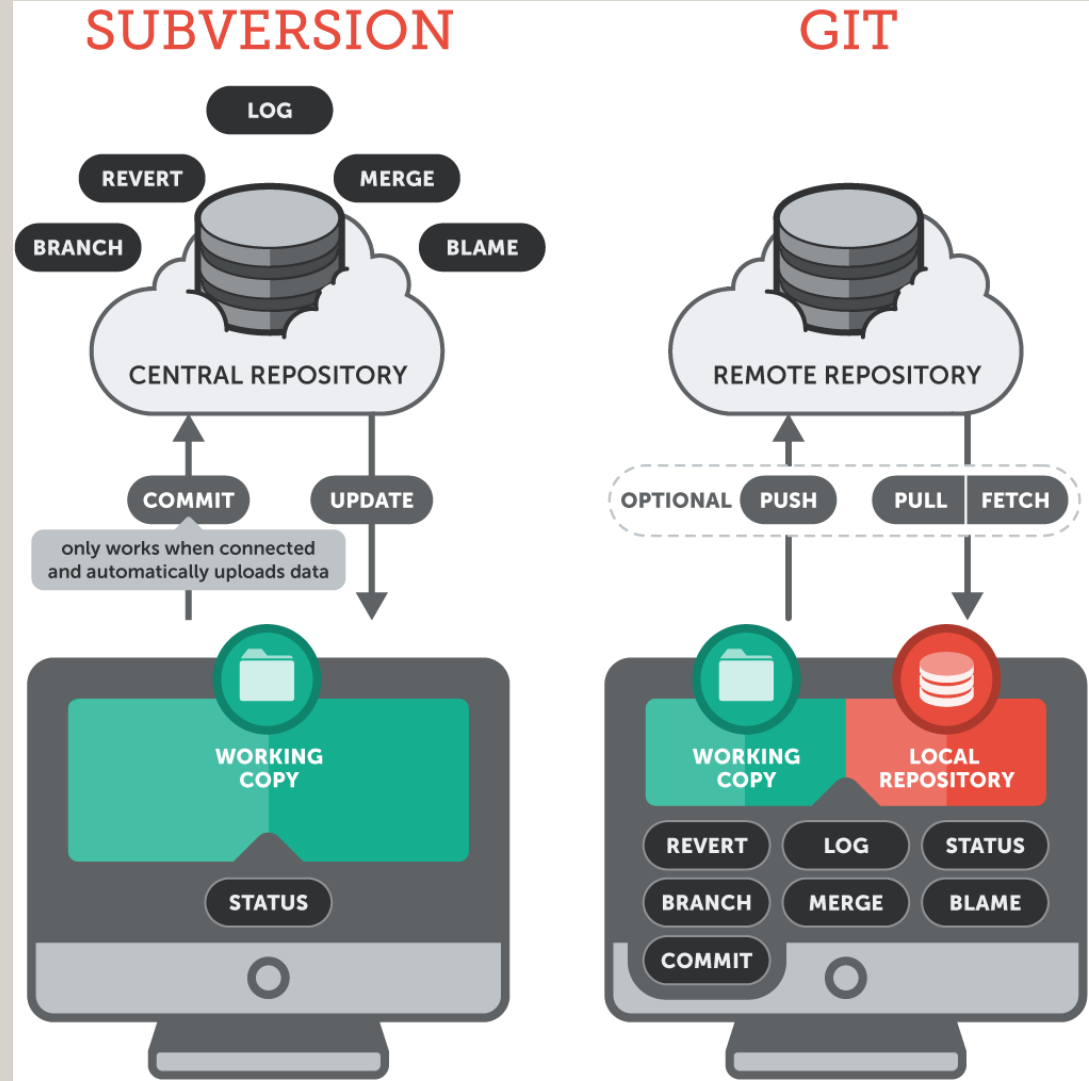
What is git?

- The most widely used VCS



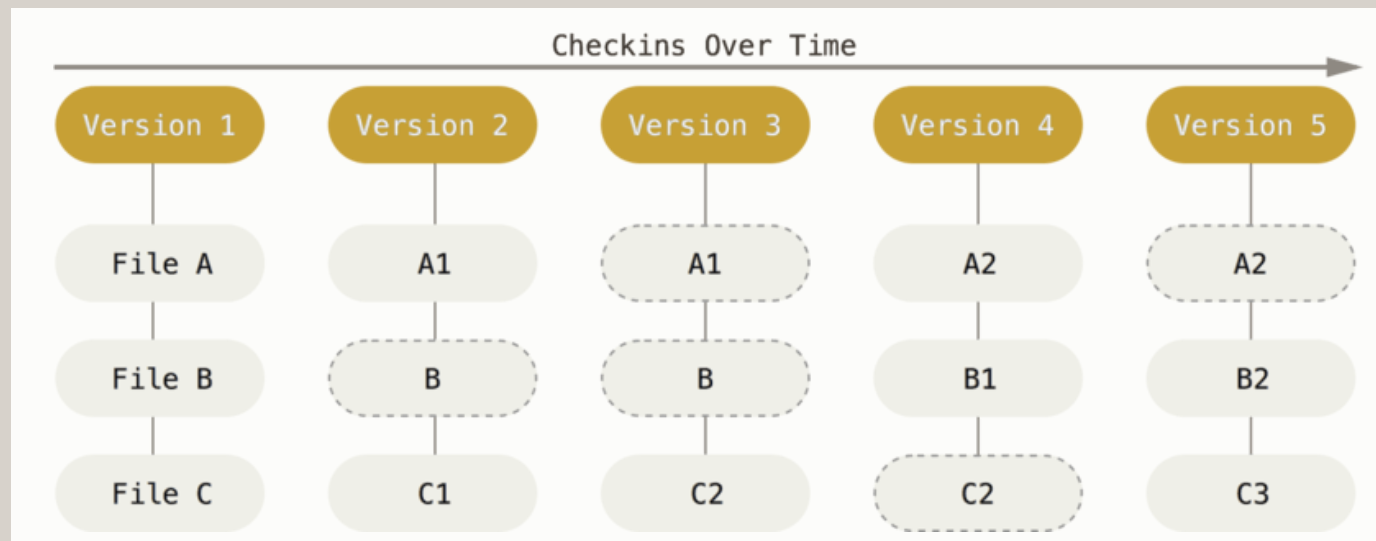
What is git?

- A distributed VCS



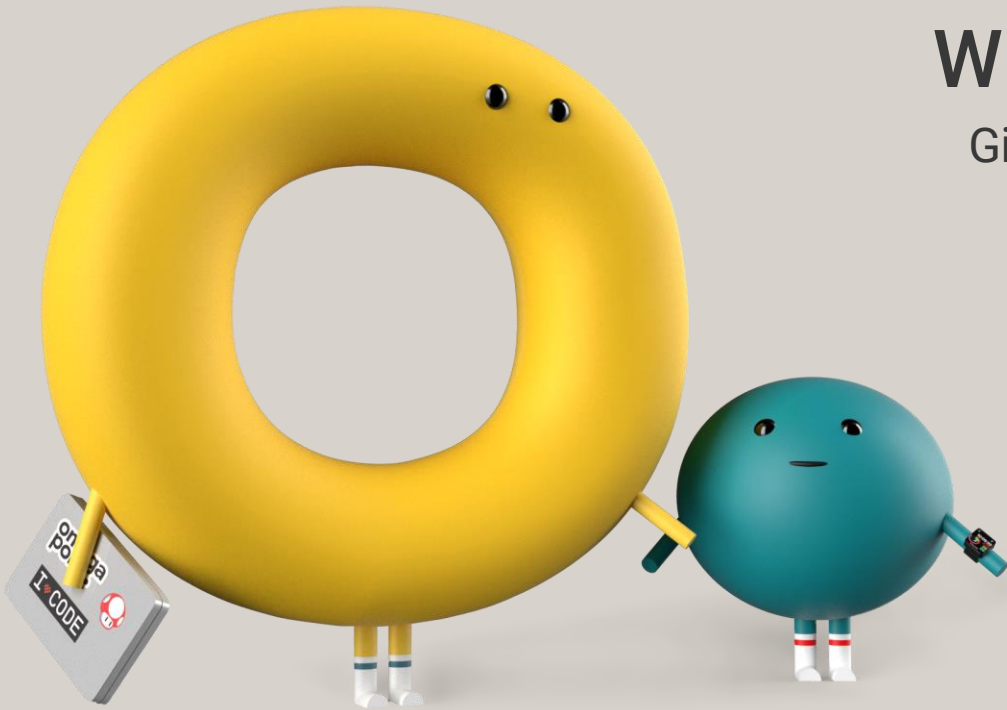
What is git?

- A distributed VCS
- I git hanterar man förändringar med snapshots av filerna.



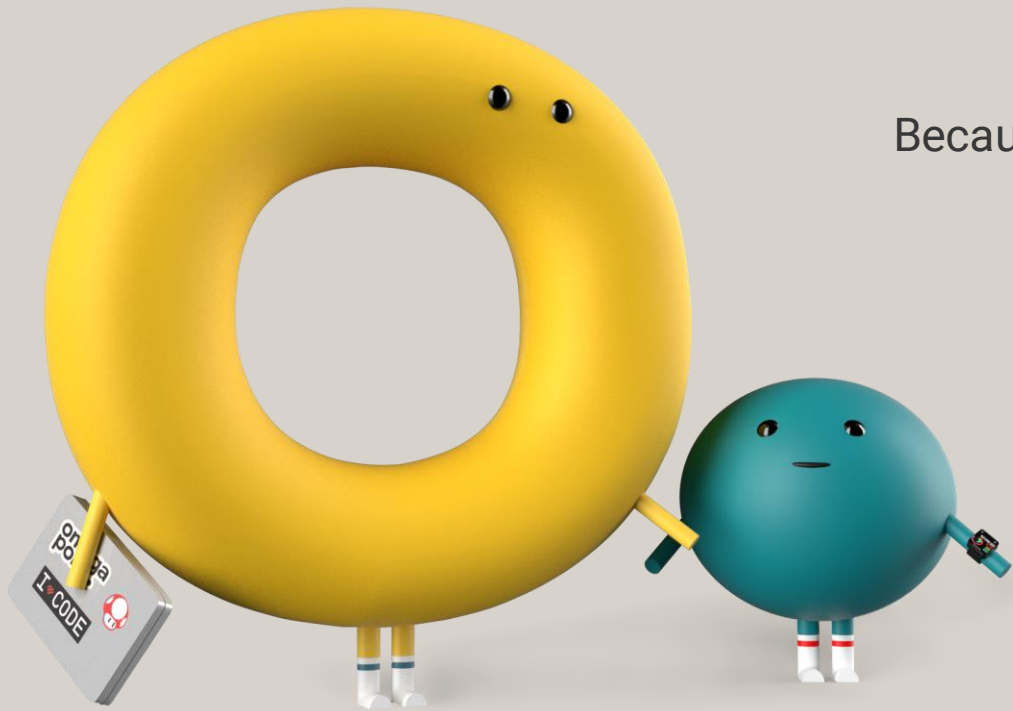
What is git? – in one sentence

Git is a tool used by developers to easily collaborate in a shared code base

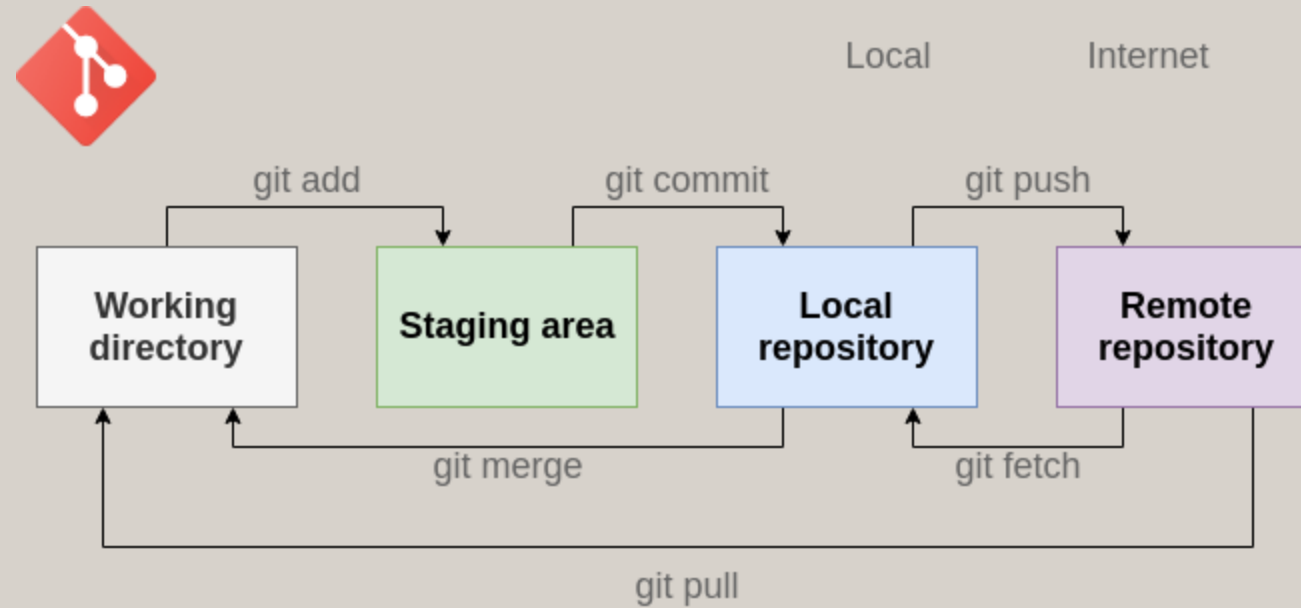


Why should we use git?

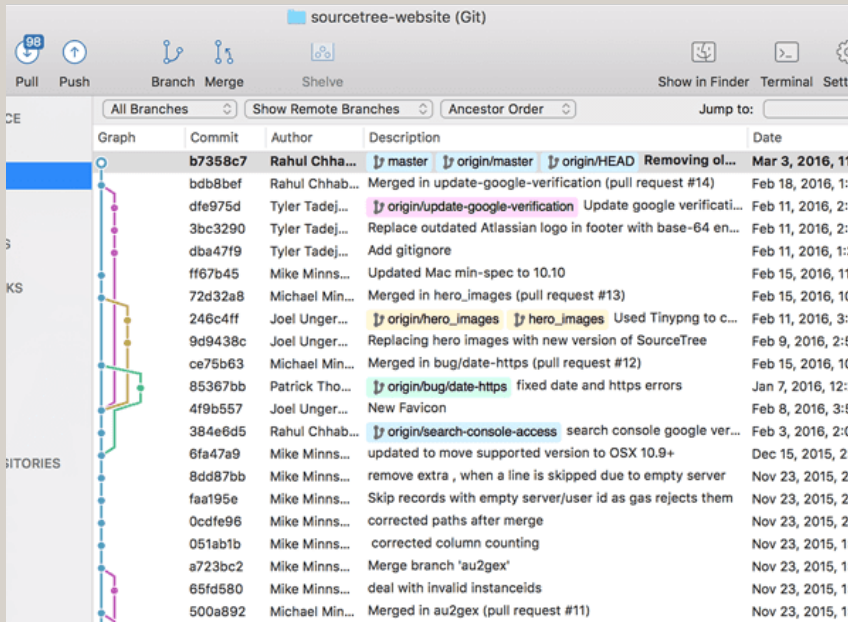
Because every other alternative leads to complete
utter chaos and you will hate your life



Git building stones



GUI client vs terminal

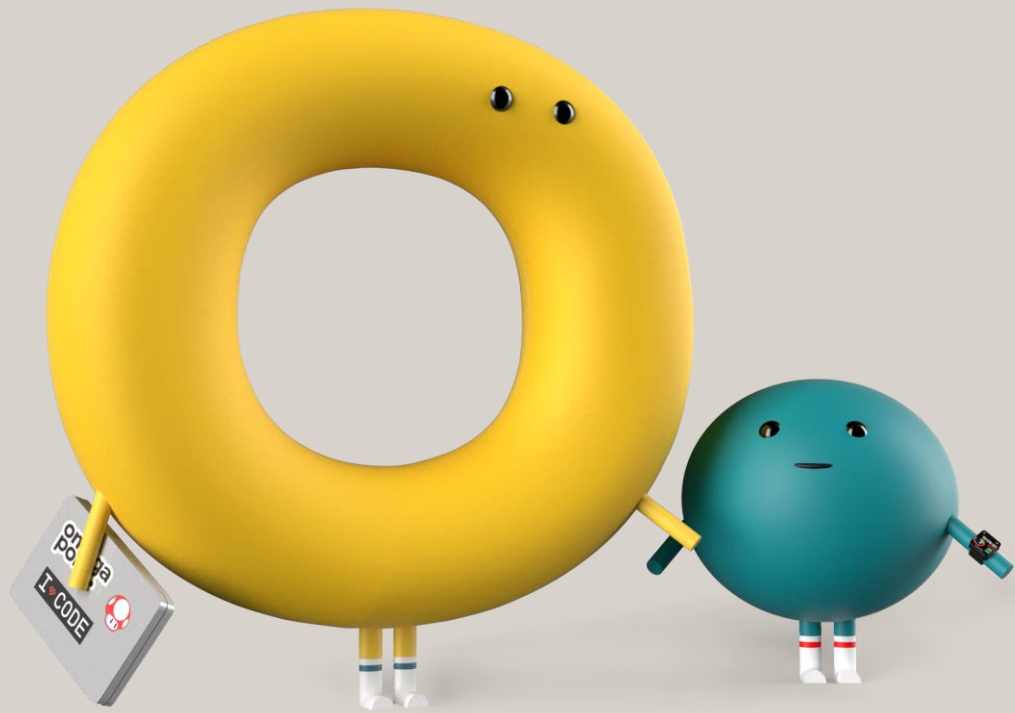


```
as@space dotfiles]$ git status
branch high-dpi
files not staged for commit:
  use "git add <file>..." to update what will be
  use "git checkout -- <file>..." to discard changes
modified:   betterlockscreen/betterlockscreen
modified:   i3/config
modified:   polybar/config

Untracked files:
  (use "git add" to track)
  changes

Changes added to commit (use "git add" and/or "git commit -a"):
  changes
as@space dotfiles]$ git commit -m "Doing some work"
```

Paus!
Vi börjar igen: 09:15



Lab time

Githug --- LAB till 10:30

To start githug:

Write githug in a terminal

To correct the current level

githug play

Commands:

```
githug - -help # display this message
githug help [TASK] # Describe available tasks or one specific task
githug hint      # Get a hint for the current level
githug levels    # List all of the levels
githug play      # Initialize the game
githug reset     # Reset the current level
githug test      # Test a level from a file path
```

Paus!
Vi börjar igen: 10:40

Branches?

- A commit object contains a pointer to the snapshot of the staged content.

9a3f

8cda

3bef

Branches?

- A commit object contains a pointer to the snapshot of the staged content.



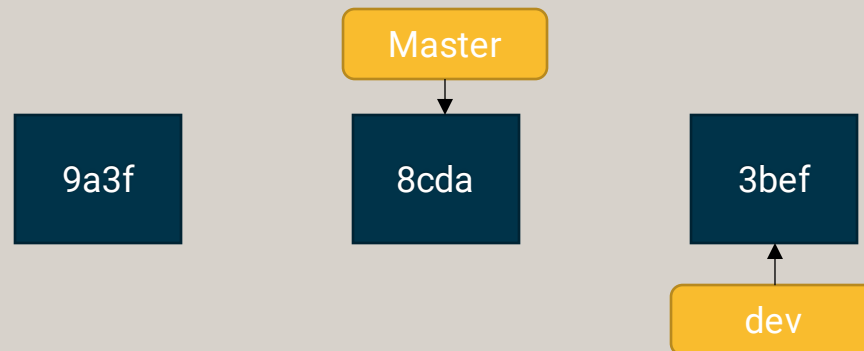
Branches?

- A commit object contains a pointer to the snapshot of the staged content.



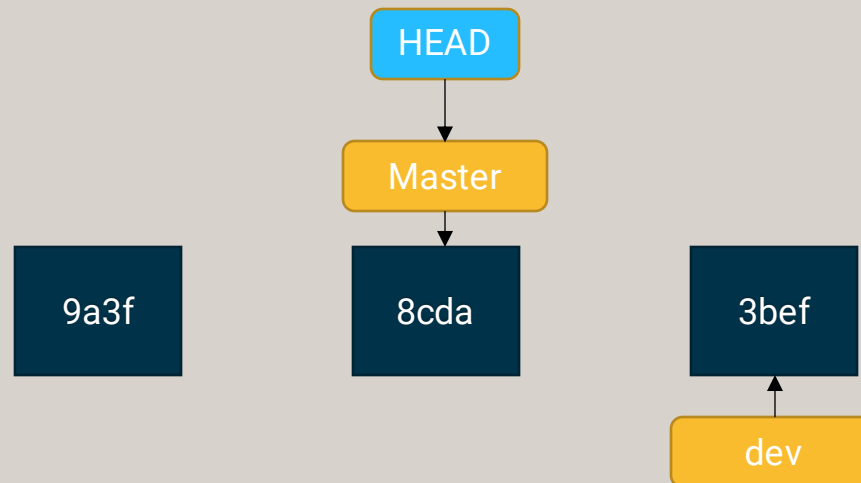
Branches?

- A commit object contains a pointer to the snapshot of the staged content.



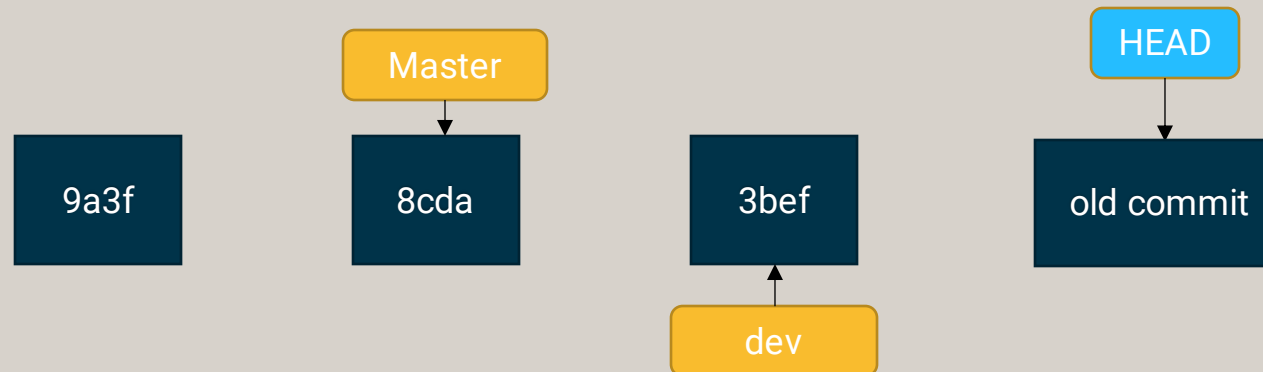
Branches?

- A commit object contains a pointer to the snapshot of the staged content.



Branches?

- A commit object contains a pointer to the snapshot of the staged content.



Branching strategies

- Different strategies in different workplaces
 - Hard to change how people work
 - You can influence a bit
 - Different opinions
 - There is no best way
- Feature branch
 - Short lived
 - Long lived

Branching strategies

One branch to rule them all

- Branching is evil
- En master branch
 - Feature toggle för att kontrollera kod
- Utveckla i par och i mob
- Integrationstester är viktigt

Master



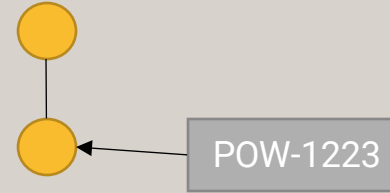
Develop



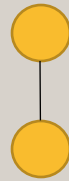
Master



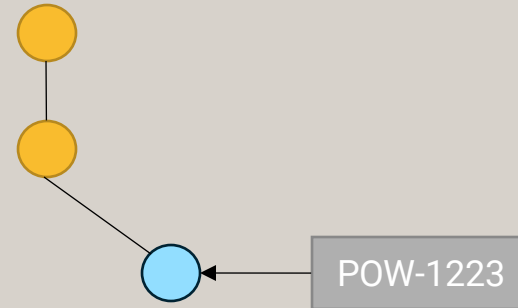
Develop



Master



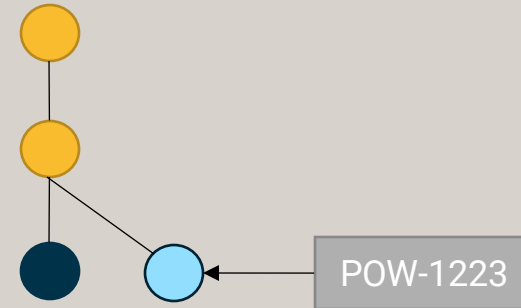
Develop



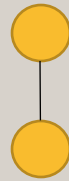
Master



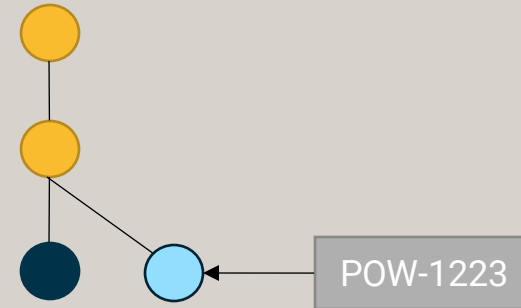
Develop



Master

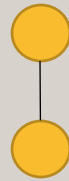


Develop

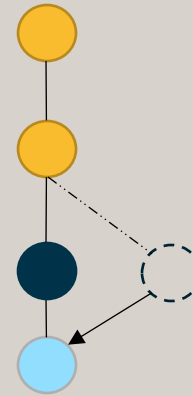


Pull request: pow-1233 -> develop

Master



Develop



Master



Develop



Master



Develop



Master



Develop

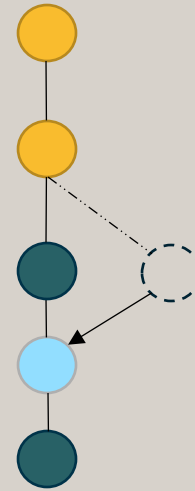


Pull request från release/1.0.0 -> master

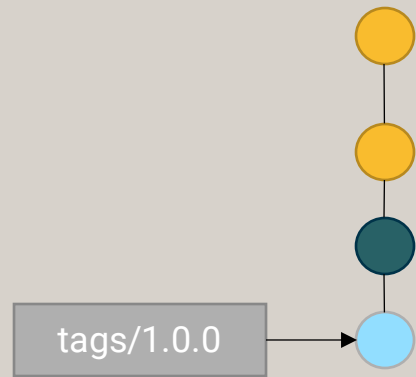
Master



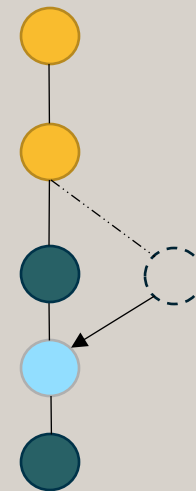
Develop



Master



Develop



Scenario 1 – merge conflict

- A merge conflict can occur when two people change the same file and git can't figure out which of the lines to keep

```
[niklas@space git-repo]$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)

        both modified:   my_code.sh

no changes added to commit (use "git add" and/or "git commit -a")
[niklas@space git-repo]$
```

Scenario 2 – feature branching

- Checkout a new branch
- Push branch
- Change name of branch
- Remove branch

Scenario 3 – A lot of local commits

```
niklas scenario_a_lot_of_local_commits » git logcourse
70d6579 2021-02-02 Niklas Hättö (HEAD -> master) I hate life.
c871912 2021-02-02 Niklas Hättö Isdfjjdsfjkfds
701cd66 2021-02-02 Niklas Hättö stuff.....
5f864b5 2021-02-02 Niklas Hättö Removed whitespace
e251b1d 2021-02-02 Niklas Hättö Typo again...
10810f5 2021-02-02 Niklas Hättö Fixed a typo
9116e5d 2021-02-02 Niklas Hättö Fixed an issue where taxes were calculated incorrectly
e5cbaae 2021-02-02 Niklas Hättö Initial commit
```

"Can you please reword your commits to explain what they are doing?"

- Ziad

"I will accept this pull request if you squash those ugly commits :)"

- Friendly developer

"What the f is going on here?"

- Linus Torvalds (probably)

Scenario 3 – A lot of local commits

```
niklas scenario_a_lot_of_local_commits » git rebase --interactive e5cbaae
[detached HEAD 76a17e8] Fixed an issue where taxes were calculated incorrectly
Date: Tue Feb 2 22:01:54 2021 +0100
1 file changed, 7 insertions(+)
Successfully rebased and updated refs/heads/master.

niklas scenario_a_lot_of_local_commits » git logcourse
76a17e8 2021-02-02 Niklas Hättö (HEAD -> master) Fixed an issue where taxes were calculated incorrectly
e5cbaae 2021-02-02 Niklas Hättö Initial commit
```

"Wow! Never seen a cleaner pull request than this"

- Ziad

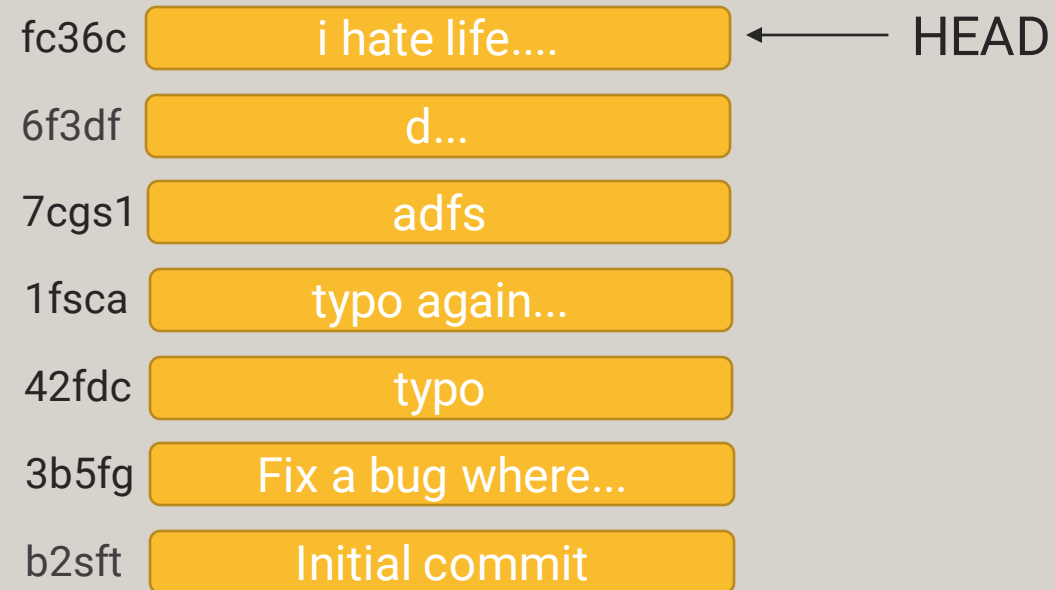
"Great. Accepted! :)"

- Friendly developer

"alright."

- Linus Torvalds (probably)

Scenario 3 – A lot of local commits

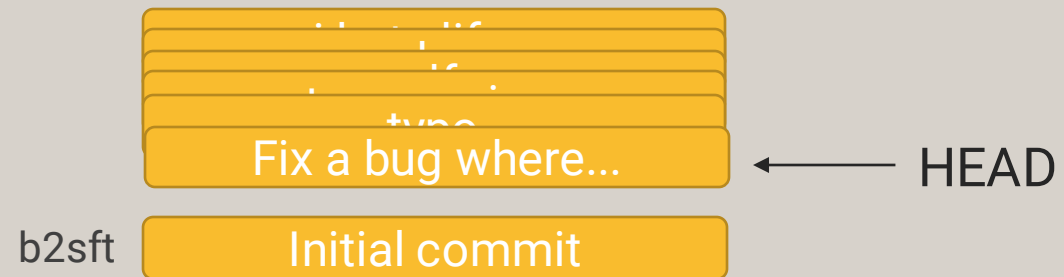


Scenario 3

git rebase --interactive b2sft



Scenario 3 – A lot of local commits



A note on changing the history

"Do not rebase commits that exist outside your repository and that people may have based work on.

If you follow that guideline, you'll be fine. If you don't, people will hate you, and you'll be scorned by friends and family."

- git-scm.com

- Rebasing changes history!
- Doing so may create tricky merge problems for your collaborator
- Avoid using --force flag and you are fine*

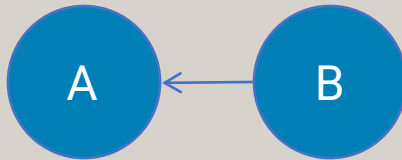
* Exceptions apply :)

Paus!
Vi börjar igen: xx:xx

Scenario 4 – git pull origin master --rebase

- When
 - You are working on a feature branch and want to get the latest changes in a remote branch
- How
 - Git pull with merging strategy: rebase

Scenario 4

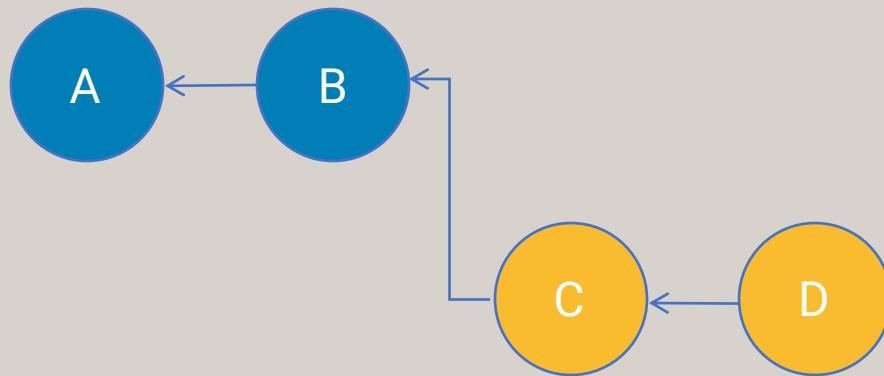


```
git commit -m "A"  
git commit -m "B"  
git push origin master_branch
```

master_branch

feature_branch

Scenario 4

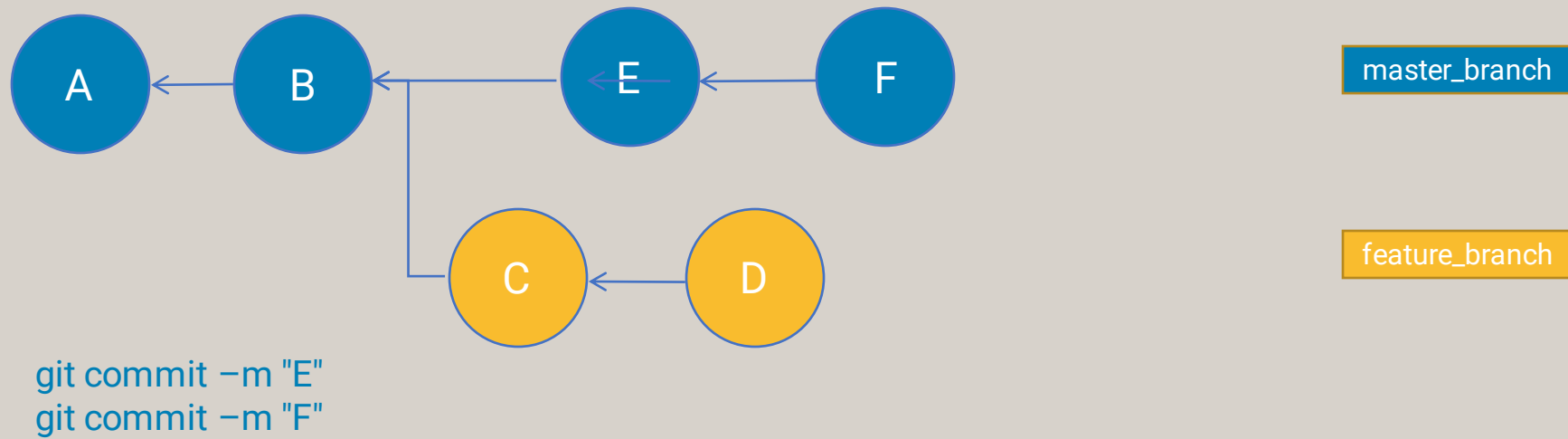


git checkout -b feature_branch
git commit -m "C"
git commit -m "D"

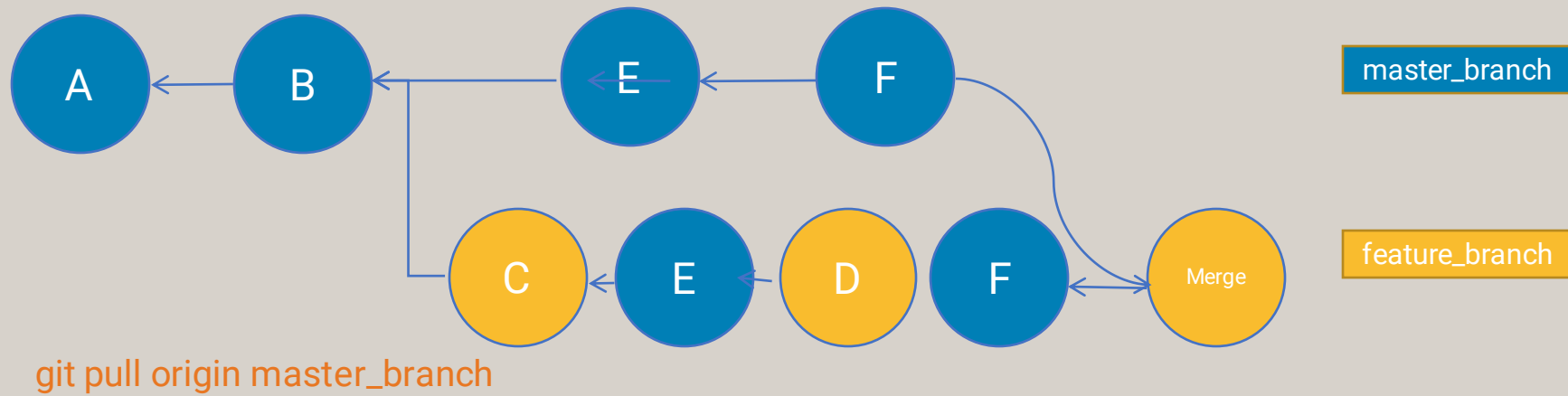
master_branch

feature_branch

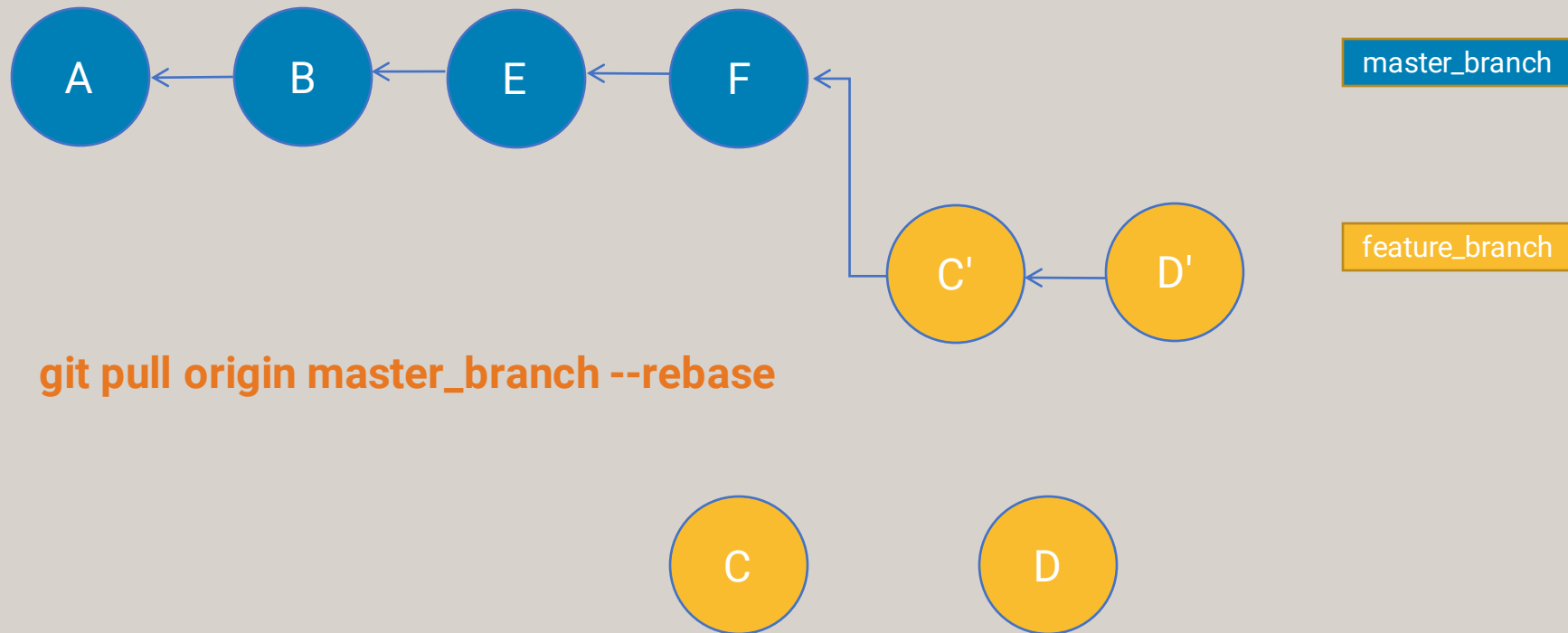
Scenario 4



Scenario 4 – Merging instead of rebasing



Scenario 4

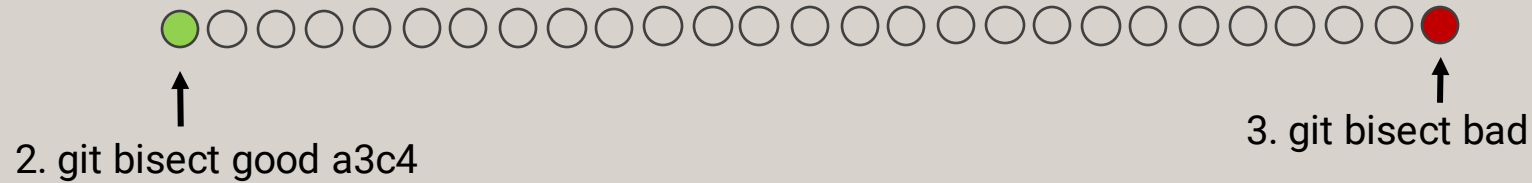


Scenario 5 – code is broken, find the faulty commit

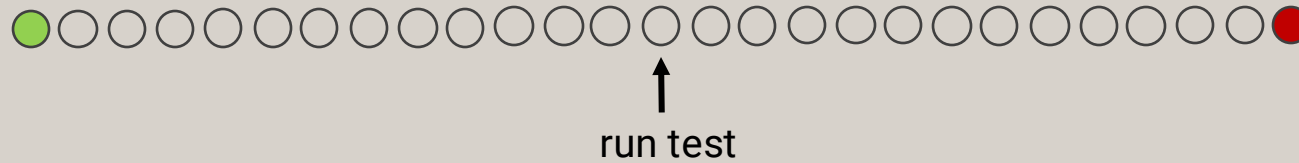
- When
 - A bug was introduced somewhere in the last 100 commits and you have a way to test if the bug is present e.g. a test or visual confirmation
- How
 - Git bisect

Scenario 5 – code is broken, find the faulty commit

1. git bisect start



Scenario 5 – code is broken, find the faulty commit



Scenario 5 – code is broken, find the faulty commit



Scenario 5 – code is broken, find the faulty commit



Scenario 5 – code is broken, find the faulty commit



Scenario 5 – code is broken, find the faulty commit



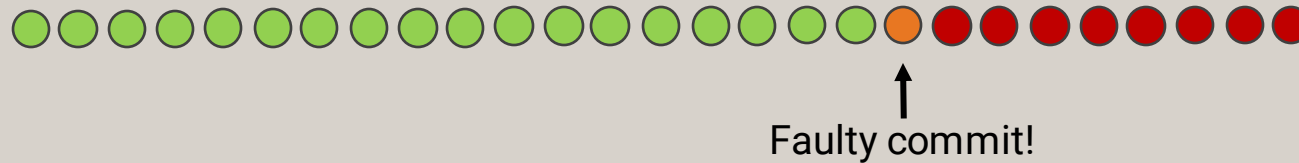
Scenario 5 – code is broken, find the faulty commit



Scenario 5 – code is broken, find the faulty commit



Scenario 5 – code is broken, find the faulty commit



Scenario 6 – nuclear reset aka I did bad

- When
 - The time it takes to solve your git problems exceeds the time to just copy and redo the changes
 - You just want to start over
- How
 - `git reset --hard origin/master`

Mobprogrammering och git under pandemi

- Mob.sh
 - Ett verktyg för att programmera i mob med git
- IntelliJ Code With Me
 - AKA Google Drive fast IntelliJ

**omega
point.**