

Workshop Git

multiplayer notepad

Anthony Clays

21 november 2016

UL  **SSIS**

Inleiding

De basis

Eerste stappen

Branches en samenwerken

Geavanceerde features

Inleiding

- Auteur: Linus Torvalds

- Auteur: Linus Torvalds
- Beter versiecontrolesysteem:

- Auteur: Linus Torvalds
- Beter versiecontrolesysteem:
 - Snel

- Auteur: Linus Torvalds
- Beter versiecontrolesysteem:
 - Snel
 - Gedistribueerd

- Auteur: Linus Torvalds
- Beter versiecontrolesysteem:
 - Snel
 - Gedistribueerd
 - Data-integriteit (“You know you can trust the data”)

- Auteur: Linus Torvalds
- Beter versiecontrolesysteem:
 - Snel
 - Gedistribueerd
 - Data-integriteit (“You know you can trust the data”)
- De naam “git”

The name "git" was given by Linus Torvalds when he wrote the very first version. He described the tool as "the stupid content tracker" and the name as (depending on your mood):

- random three-letter combination that is pronounceable, and not actually used by any common UNIX command. The fact that it is a mispronunciation of "get" may or may not be relevant.
- stupid. contemptible and despicable. simple. Take your pick from the dictionary of slang.
- "global information tracker": you're in a good mood, and it actually works for you. Angels sing, and a light suddenly fills the room.
- "goddamn idiotic truckload of sh*t": when it breaks

Windows `https://git-scm.com/`.

Na installatie: gebruik "Git Bash"

macOS Standaard geïnstalleerd?

Anders: Homebrew of `https://git-scm.com/`

Linux Installeer via package manager

(`sudo apt-get install git`)

Windows `https://git-scm.com/`.

Na installatie: gebruik "Git Bash"

macOS Standaard geïnstalleerd?

Anders: Homebrew of `https://git-scm.com/`

Linux Installeer via package manager

(`sudo apt-get install git`)

- `git config --global user.email "balthasar@boma.org"`
- `git config --global user.name "Balthasar Boma"`
- Windows: `git config --global core.editor wordpad`

De basis



Staat van je repository: drie verschillende “stadia”

Staat van je repository: drie verschillende “stadia”

Working Tree De bestanden zoals ze op je harde schijf staan

Staat van je repository: drie verschillende “stadia”

Working Tree De bestanden zoals ze op je harde schijf staan

HEAD De commit waarop je aan het verderwerken bent

Staat van je repository: drie verschillende “stadia”

Working Tree De bestanden zoals ze op je harde schijf staan

HEAD De commit waarop je aan het verderwerken bent

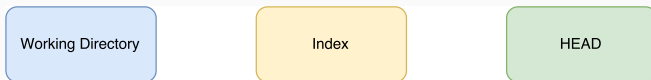
Index De veranderingen die je in de *volgende* commit wilt opslaan

Staat van je repository: drie verschillende “stadia”

Working Tree De bestanden zoals ze op je harde schijf staan

HEAD De commit waarop je aan het verderwerken bent

Index De veranderingen die je in de *volgende* commit wilt opslaan



Eerste stappen

- Voor we beginnen: `git help`

- Voor we beginnen: `git help`
- `git init`: creëer een nieuwe git repository in de huidige map

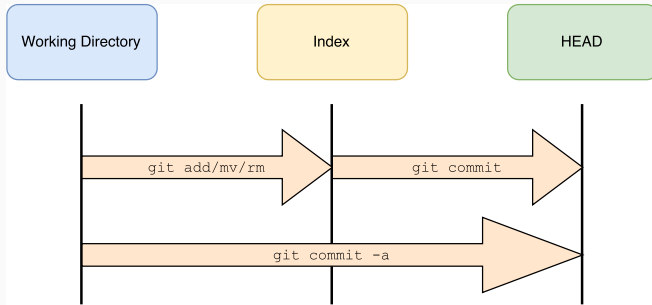
- Voor we beginnen: `git help`
- `git init`: creëer een nieuwe git repository in de huidige map
- Volledig lokaal

- Voor we beginnen: `git help`
- `git init`: creëer een nieuwe git repository in de huidige map
- Volledig lokaal
- Demo: volg mee!

- `git add`: toevoegen aan *index*

- `git add`: toevoegen aan *index*
- `git commit`: nieuwe commit maken verderbouwend op HEAD

- `git add`: toevoegen aan *index*
- `git commit`: nieuwe commit maken verderbouwend op HEAD
- Demo

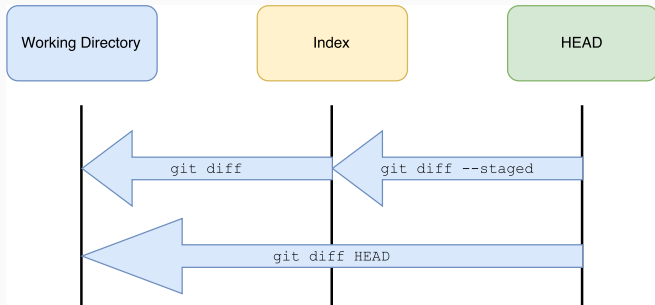


Git can remember it for you wholesale

- `git log` Simpele oplisting
- `git log --oneline` zet commit hash + message op één regel
- `git log --graph` toont de graph tussen branches
- `git log --decorate` toont labels voor de verschillende branches
- `git log identifiser` tot *identifiser* ipv HEAD

- `git diff` Index → Working directory
- `git diff --staged` HEAD → Index
- `git diff id` Commit *id* → Working directory
- `git diff id1 id2` Commit *id1* → Commit *id2*

- `git diff` Index → Working directory
- `git diff --staged` HEAD → Index
- `git diff id` Commit *id* → Working directory
- `git diff id1 id2` Commit *id1* → Commit *id2*



Terugkeren naar een andere commit

Terugkeren naar een andere commit
Meestal HEAD, HEAD~, HEAD~2 ...

Terugkeren naar een andere commit

Meestal HEAD, HEAD~, HEAD~2 ...

- `git reset --soft`

Update HEAD naar een bepaalde commit en stop.

Terugkeren naar een andere commit

Meestal HEAD, HEAD~, HEAD~2 ...

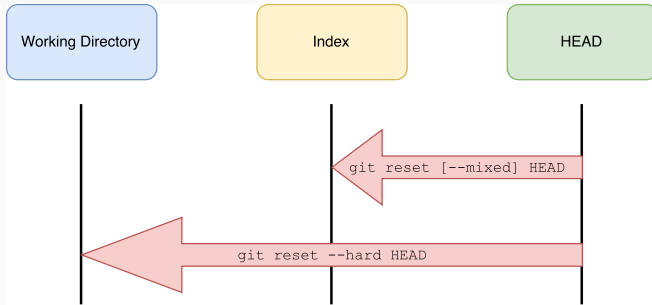
- `git reset --soft`
Update HEAD naar een bepaalde commit en stop.
- `git reset [--mixed]`
Hetzelfde, maar reset ook de index.

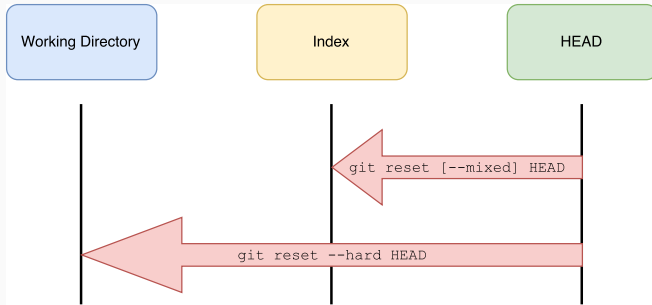
Terugkeren naar een andere commit

Meestal HEAD, HEAD~, HEAD~2 ...

- `git reset --soft`
Update HEAD naar een bepaalde commit en stop.
- `git reset [--mixed]`
Hetzelfde, maar reset ook de index.
- `git reset --hard`
Hetzelfde, maar reset zowel de index als de working directory.

git reset (2)





Demo

git checkout : zet het bestand (in de working directory)
terug naar de versie in de index

- .gitignore vertelt git welke bestanden te negeren

- .gitignore vertelt git welke bestanden te negeren
- Meestal mee in repository

- .gitignore vertelt git welke bestanden te negeren
- Meestal mee in repository
- Generatie: <https://gitignore.io>

Branches en samenwerken

- Branches zijn verwijzingen naar een commit

- Branches zijn verwijzingen naar een commit
- `git branch`: branches beheren

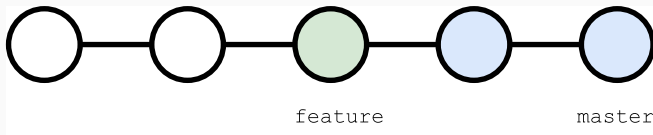
- Branches zijn verwijzingen naar een commit
- `git branch`: branches beheren
- `git checkout branch`: switch naar branch *branch*

- Branches zijn verwijzingen naar een commit
- `git branch`: branches beheren
- `git checkout branch`: switch naar branch *branch*
- `git checkout -b branch`: `git branch branch &&`
`git checkout branch`

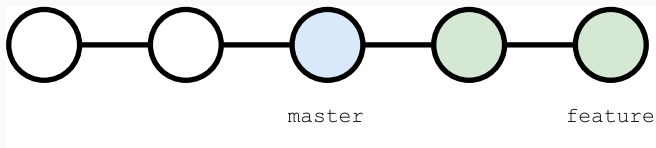
- `git merge identifïer`: merget een commit in de huidige branch

- `git merge identifiser`: merget een commit in de huidige branch
- In praktijk is deze commit (bijna) altijd een branch

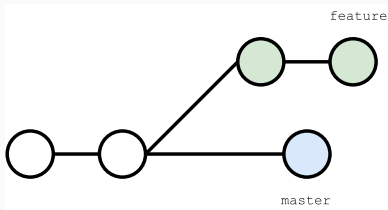
- `git merge identifier`: merget een commit in de huidige branch
- In praktijk is deze commit (bijna) altijd een branch
- Indien *identifier* een voorvader is van de huidige branch: doe niets



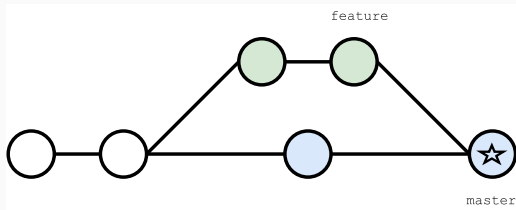
- `git merge identifïer`: merget een commit in de huidige branch
- In praktijk is deze commit (bijna) altijd een branch
- Indien *identifïer* een voorvader is van de huidige branch: doe niets
- Indien deze branch volgt op de huidige: *fast-forward*



- `git merge identifiser`: merget een commit in de huidige branch
- In praktijk is deze commit (bijna) altijd een branch
- Indien *identifiser* een voorvader is van de huidige branch: doe niets
- Indien deze branch volgt op de huidige: *fast-forward*
- Anders: creëert een merge commit met twee parents: één van elke branch



- `git merge identifiser`: merget een commit in de huidige branch
- In praktijk is deze commit (bijna) altijd een branch
- Indien *identifiser* een voorvader is van de huidige branch: doe niets
- Indien deze branch volgt op de huidige: *fast-forward*
- Anders: creëert een merge commit met twee parents: één van elke branch



- Veranderingen in hetzelfde bestand worden slim samengevoegd

- Veranderingen in hetzelfde bestand worden slim samengevoegd
- Veranderingen op dezelfde regel \Rightarrow conflict!

- Veranderingen in hetzelfde bestand worden slim samengevoegd
- Veranderingen op dezelfde regel \Rightarrow conflict!
- Voordat de merge commit gefinaliseerd wordt: los het conflict op.

- Veranderingen in hetzelfde bestand worden slim samengevoegd
- Veranderingen op dezelfde regel \Rightarrow conflict!
- Voordat de merge commit gefinaliseerd wordt: los het conflict op.
- `git merge --abort`

- Tot nu toe: alles lokaal

- Tot nu toe: alles lokaal
- Waar is remote?

- Tot nu toe: alles lokaal
- Waar is remote?
 - `git clone url`: download een remote repository

- Tot nu toe: alles lokaal
- Waar is remote?
 - `git clone url`: download een remote repository
 - `git remote add origin url`: handmatig URL toevoegen

- Tot nu toe: alles lokaal
- Waar is remote?
 - `git clone url`: download een remote repository
 - `git remote add origin url`: handmatig URL toevoegen
 - `git remote set-url origin url`: URL van remote aanpassen

- Tot nu toe: alles lokaal
- Waar is remote?
 - `git clone url`: download een remote repository
 - `git remote add origin url`: handmatig URL toevoegen
 - `git remote set-url origin url`: URL van remote aanpassen
- Veranderingen ophalen die in de remote zijn doorgevoerd:
`git fetch`

- Tot nu toe: alles lokaal
- Waar is remote?
 - `git clone url`: download een remote repository
 - `git remote add origin url`: handmatig URL toevoegen
 - `git remote set-url origin url`: URL van remote aanpassen
- Veranderingen ophalen die in de remote zijn doorgevoerd:
`git fetch`
- Deze veranderingen ook direct lokaal doorvoeren: `git pull`

- Tot nu toe: alles lokaal
- Waar is remote?
 - `git clone url`: download een remote repository
 - `git remote add origin url`: handmatig URL toevoegen
 - `git remote set-url origin url`: URL van remote aanpassen
- Veranderingen ophalen die in de remote zijn doorgevoerd:
`git fetch`
- Deze veranderingen ook direct lokaal doorvoeren: `git pull`
- Pushen naar remote: `git push`


```
git clone https://github.com/anthonyclays/demo2
```

`git checkout identifiek path`: update files in de working directory om overeen te komen met de versie van dit bestand in een gegeven commit

`git checkout identifiser path`: update files in de working directory om overeen te komen met de versie van dit bestand in een gegeven commit

- Indien geen identifier: HEAD

`git checkout identifïer path`: update files in de working directory om overeen te komen met de versie van dit bestand in een gegeven commit

- Indien geen identifïer: HEAD
- Indien geen path: working tree

Geavanceerde features

- rebase = tak verplaatsen naar een nieuwe base commit

- rebase = tak verplaatsen naar een nieuwe base commit
- Als alternatief voor merge: branch “voortbewegen” op base branch

- rebase = tak verplaatsen naar een nieuwe base commit
- Als alternatief voor merge: branch “vooruitbewegen” op base branch
- Voordelen

- rebase = tak verplaatsen naar een nieuwe base commit
- Als alternatief voor merge: branch “vooruitbewegen” op base branch
- Voordelen
 - Lineaire geschiedenis

- rebase = tak verplaatsen naar een nieuwe base commit
- Als alternatief voor merge: branch “vooruitbewegen” op base branch
- Voordelen
 - Lineaire geschiedenis
 - Geen extra *merge commit*

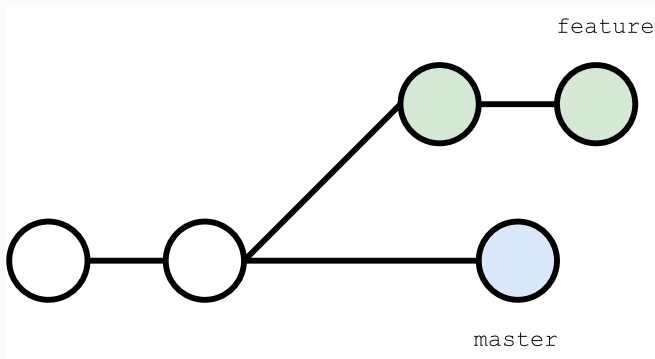
- rebase = tak verplaatsen naar een nieuwe base commit
- Als alternatief voor merge: branch “vooruitbewegen” op base branch
- Voordelen
 - Lineaire geschiedenis
 - Geen extra *merge commit*
- Nadelen

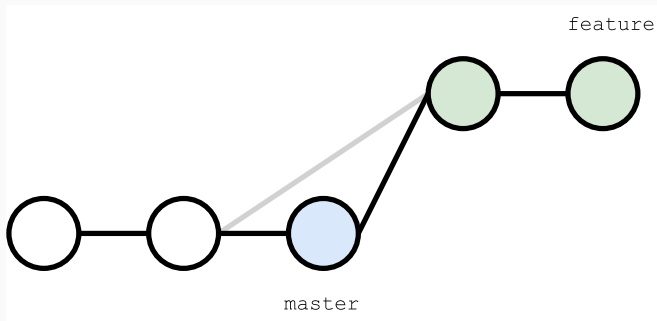
- rebase = tak verplaatsen naar een nieuwe base commit
- Als alternatief voor merge: branch “vooruitbewegen” op base branch
- Voordelen
 - Lineaire geschiedenis
 - Geen extra *merge commit*
- Nadelen
 - Herschrijft geschiedenis (gevaarlijk)

- rebase = tak verplaatsen naar een nieuwe base commit
- Als alternatief voor merge: branch “vooruitbewegen” op base branch
- Voordelen
 - Lineaire geschiedenis
 - Geen extra *merge commit*
- Nadelen
 - Herschrijft geschiedenis (gevaarlijk)
 - Moeilijker

- rebase = tak verplaatsen naar een nieuwe base commit
- Als alternatief voor merge: branch “vooruitbewegen” op base branch
- Voordelen
 - Lineaire geschiedenis
 - Geen extra *merge commit*
- Nadelen
 - Herschrijft geschiedenis (gevaarlijk)
 - Moeilijker
- rebase -i

- rebase = tak verplaatsen naar een nieuwe base commit
- Als alternatief voor merge: branch “vooruitbewegen” op base branch
- Voordelen
 - Lineaire geschiedenis
 - Geen extra *merge commit*
- Nadelen
 - Herschrijft geschiedenis (gevaarlijk)
 - Moeilijker
- `rebase -i`
- `rebase --onto=new-base branch old-base`





```
git commit --amend
```

`git commit --amend`: voeg veranderingen toe aan vorige
commit

the sock drawer of version control

- `git stash [save]` sla huidige veranderingen op
- `git stash (apply|pop) [stash]` pas de laatste stash toe
- `git stash drop [stash]` verwijder deze stash
- `git stash list` toon alle stashes
- `git stash branch branch [stash]` pas toe in een nieuwe branch

- `git tag [-a] tag` tag maken op huidige commit
- `git tag -d tag` tag verwijderen
- `git push --tags` lokale tags naar remote kopiëren

Destroying friendships since 2006

Toon de auteur(s) van een bestand

Destroying friendships since 2006

Toon de auteur(s) van een bestand

Lijn per lijn

Toont een lijst van alle staten waarin de repository zich heeft bevonden

Toont een lijst van alle staten waarin de repository zich heeft bevonden

Vind verloren commits terug

- Git zit logisch in mekaar

- Git zit logisch in mekaar
- Als iets foutloopt: probeer te begrijpen waarom

- Git zit logisch in mekaar
- Als iets foutloopt: probeer te begrijpen waarom
- Als je iets niet snapt: lees de manual, zoek op op google

- Git zit logisch in mekaar
- Als iets foutloopt: probeer te begrijpen waarom
- Als je iets niet snapt: lees de manual, zoek op op google
- Doe nooit een force push naar master `git commit crime`

- Git zit logisch in mekaar
- Als iets foutloopt: probeer te begrijpen waarom
- Als je iets niet snapt: lees de manual, zoek op op google
- Doe nooit een force push naar master `git commit crime`
- ULYSSIS biedt ook GitLab aan (private repositories)

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

