

Git for Developers

Oefening 1: Repositories

Installeer Git

Maak een repository aan

- Open **Git Bash**
- cd c:
- mkdir git
- cd git
- git init demo

Oefening 2: Basic Snapshotting

1. Maak een nieuw tekstbestand aan
2. Bekijk de status (status)
3. Voeg deze toe aan de repository (add)
4. Bekijk de status (status)
5. Commit de wijzigingen (het toegevoegde bestand) (commit)
6. Bekijk de status (status)
7. Wijzig de inhoud van het bestand
8. Bekijk de status (status)
9. Commit de wijzigingen
10. Wat is er gebeurd? Wat had je verwacht?
11. Voeg het gewijzigde bestand toe (add)
12. Bekijk de status (status)
13. Commit de wijzigingen (commit)
14. Bekijk de status (status)
15. Maak nog een nieuw tekstbestand aan
16. Bekijk de status (status)
17. Commit de wijzigingen (commit)
18. Wat is er gebeurd? Wat had je verwacht?

Oefening 3: Going Distributed

Clone de huidige repository

- cd c:/git¹
- git clone demo demo-clone

Bekijk de clone

- cd demo-clone
- git log --oneline

Bekijk de remote tracking branches

- git remote
- git remote -v

Voeg een nieuw bestand toe (add)

Commit deze wijziging (commit)

Navigeer naar de oorspronkelijke repository

- cd c:/git/demo

Controleer of de nieuwe wijzigingen doorgevoerd zijn

- git log --oneline

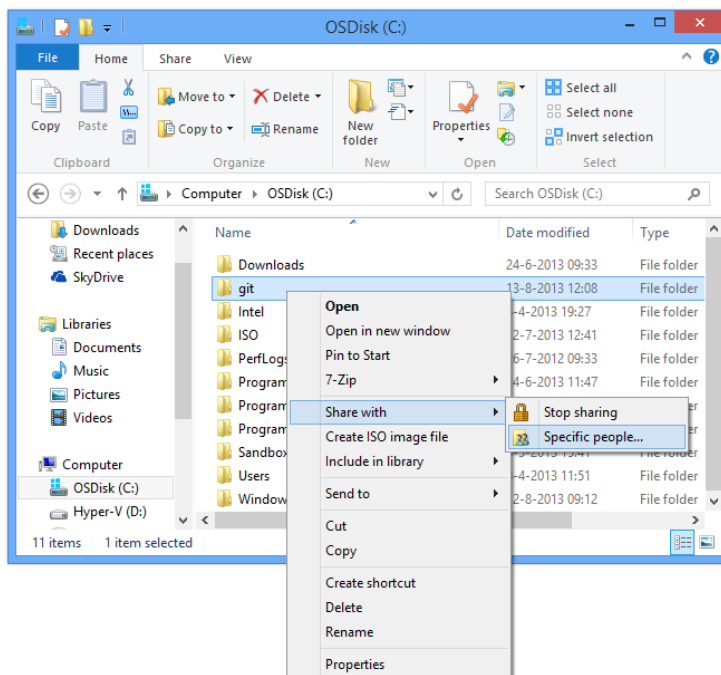
Haal de wijziging binnen (pull)

Controleer of het nieuwe bestand aanwezig is

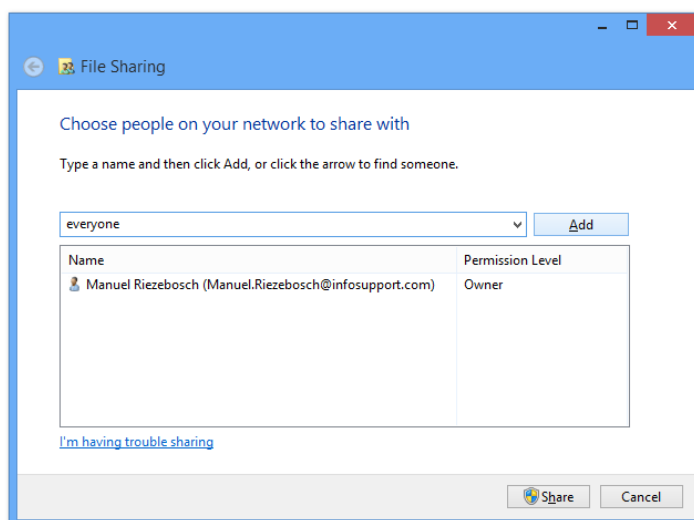
- git log --oneline

Maak een share aan voor de git directory:

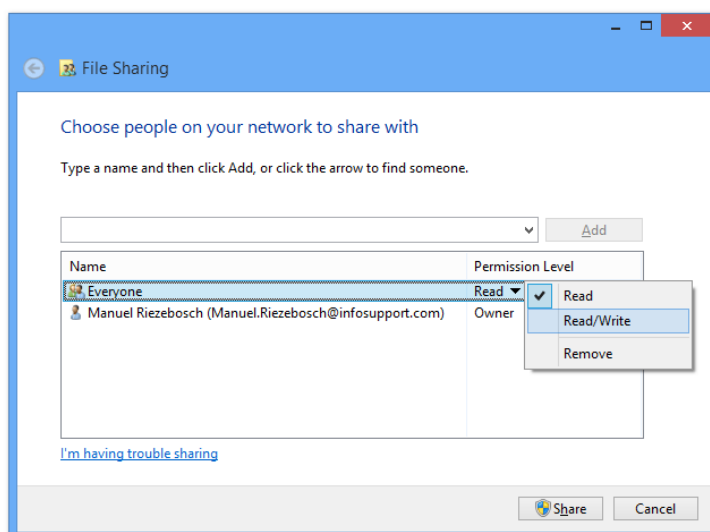
¹ **Let op!** Bash werkt met de forward slash als path separator.



Rechtermuisknop → Share with → Specific people...



Type in "everyone" en klik op "Add"



Everyone → Read → Read/Write

Maak nu een "bare" repository aan in de share van degene die naast je zit:

- `git init //cursistXY2/git/cursistXX3 --bare`

Voeg deze repository toe als remote aan jouw repository:

- `git remote add buurman //cursistXY/git/cursistXX`

Push de commits van master naar deze remote

- `git push buurman master`

Oefening 4: Branching and Merging

1. Clone de bootstrap source code: <https://github.com/twbs/bootstrap.git> (clone)
2. Bekijk de carousel example
3. Maak een branch aan met de naam 'background' (branch)
4. Checkout de nieuwe branch (checkout)
5. Pas de achtergrondkleur aan in de carousel.css


```
body {
    padding-bottom: 40px;
    color: #5a5a5a;
    background-color: red;
}
```
6. Bekijk het resultaat
7. Commit de change.
8. Checkout master
9. Maak een nieuwe branch aan met de naam 'footer'
10. Checkout de nieuwe branch
11. Pas in de index.html de footer aan naar je eigen bedrijfsnaam (op de plaats van "2013 Company, Inc")
12. Bekijk het resultaat
13. Commit de change.
14. Maak een nieuwe branch aan met de naam 'combined'
15. Merge de changes van de andere branches naar deze branch
16. Bekijk het resultaat

² cursistXY is de machinenaam van de PC van je buurman of buurvrouw

³ cursistXX is de machinenaam van je eigen PC

Extra oefening:

De docent deelt een repository met daarin een leeg bestand waarin de namen en de e-mailadressen van de cursisten moeten komen. Iedereen krijgt leesrechten op deze repository en kan deze clonen. Slechts enkelen krijgen schrijfrechten.

De cursisten wisselen onderling commits uit door elkaar schrijfrechten op de repository te geven. Uiteindelijk pusht de cursist met schrijfrechten het resultaat terug naar de repository van de docent.

Oefening 5: Status and Diff

1. Pas een bestand in de workin directory aan.
2. Voeg de wijzigingen toe aan de repository (add).
3. Wijzig hetzelfde bestand nogmaals
4. Vergelijk de
 - Workspace met index
 - Workspace met HEAD
 - Index met HEAD

Oefening 6: Rewriting History

1. Voeg de huidige wijzigingen in de working directory toe aan de laatste commit (commit --amend)
2. Beschrijf wat er net is gebeurd in de repository. Wanneer mag dit nog en wanneer absoluut niet meer?
3. Voeg alle commits samen tot één commit (rebase)