



Git - intro

Web Essentials

DE HOGESCHOOL MET HET NETWERK

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook



Version Control System

- Systeem om veranderingen in files bij te houden
- Samenwerken aan dezelfde code
- Historiek van files
 - Wie heeft aan wat gewerkt?
 - Uitleg bij elke *commit* is belangrijk!
 - Veranderingen ongedaan maken

Version Control System

- Terminologie:
 - **Repository:**
Plaats waar bestanden én informatie worden bewaard
 - **Master repository:**
Meestal op een externe server
Bevat de huidige (gedeelde) versie van de bestanden in het project
 - **Working copy:**
Lokale kopie van de master repository waar de gebruiker wijzigingen in kan brengen
 - **Working directory:**
Lokale map op systeem van de gebruiker waar hij effectief in werkt en test

Version Control System

- Terminologie:
 - **Commit:**
Veranderingen in de lokale bestanden toevoegen aan een repository
 - **Push:**
Veranderingen in de working copy updaten naar de master repository
 - **Pull:**
Veranderingen in de master repository overbrengen naar de working copy
 - Merge:** Veranderingen van de master worden meestal automatisch verwerkt en samengevoegd in de lokale
 - Conflict:** als de veranderingen niet automatisch opgelost kunnen worden resulteert dit soms in een conflict dat opgelost moet worden.

Version Control System

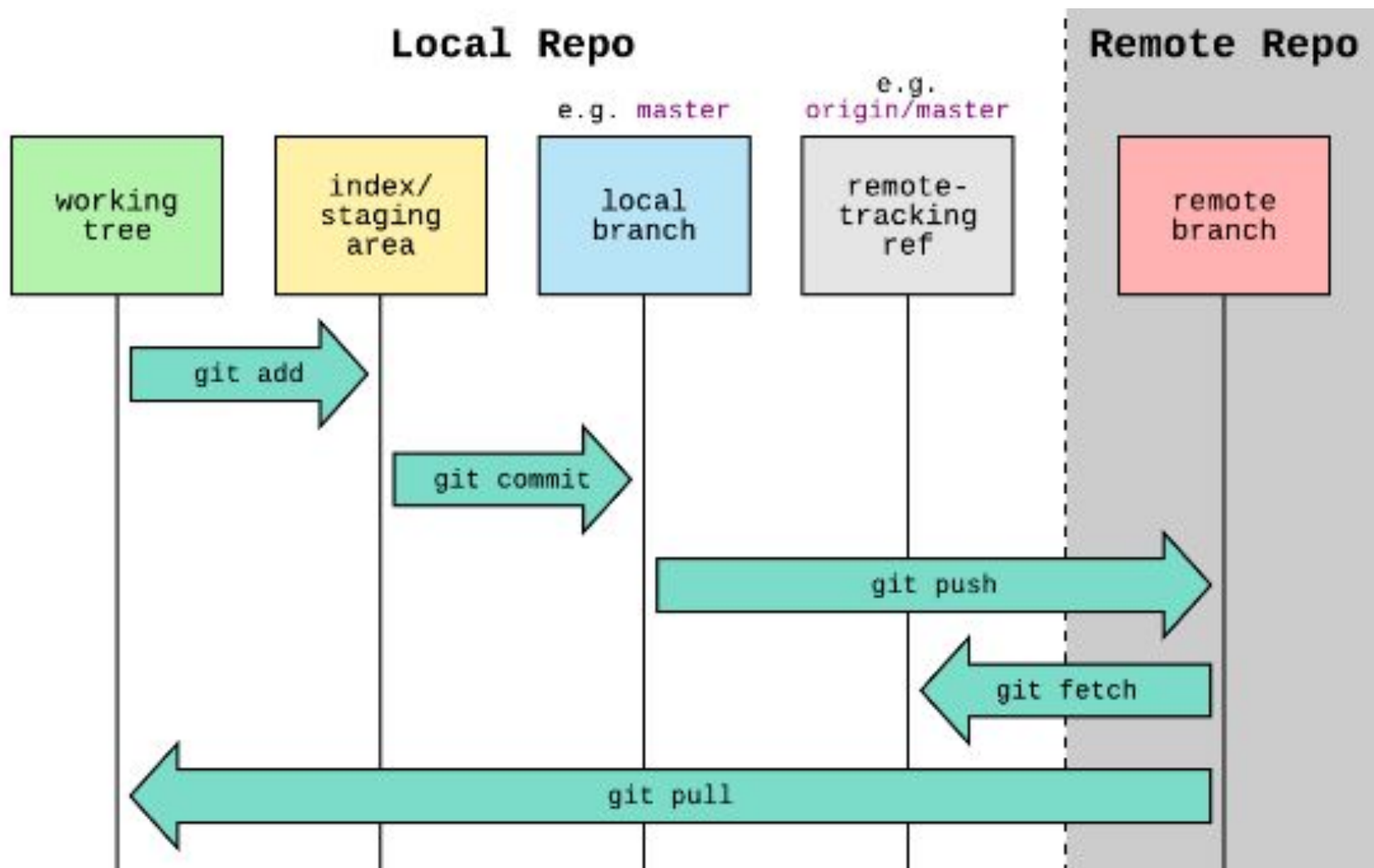
- Verschillende soorten
 - Non-distributed:
 - Server heeft master repository
 - Alle *commits* naar de server
 - Voorbeelden: SVN, CVS
 - Distributed:
 - Server heeft master repository
 - Ontwikkelaars hebben een lokale kopie
 - Voorbeelden: Mercurial, **Git**

A Distributed Revision Control System

Git

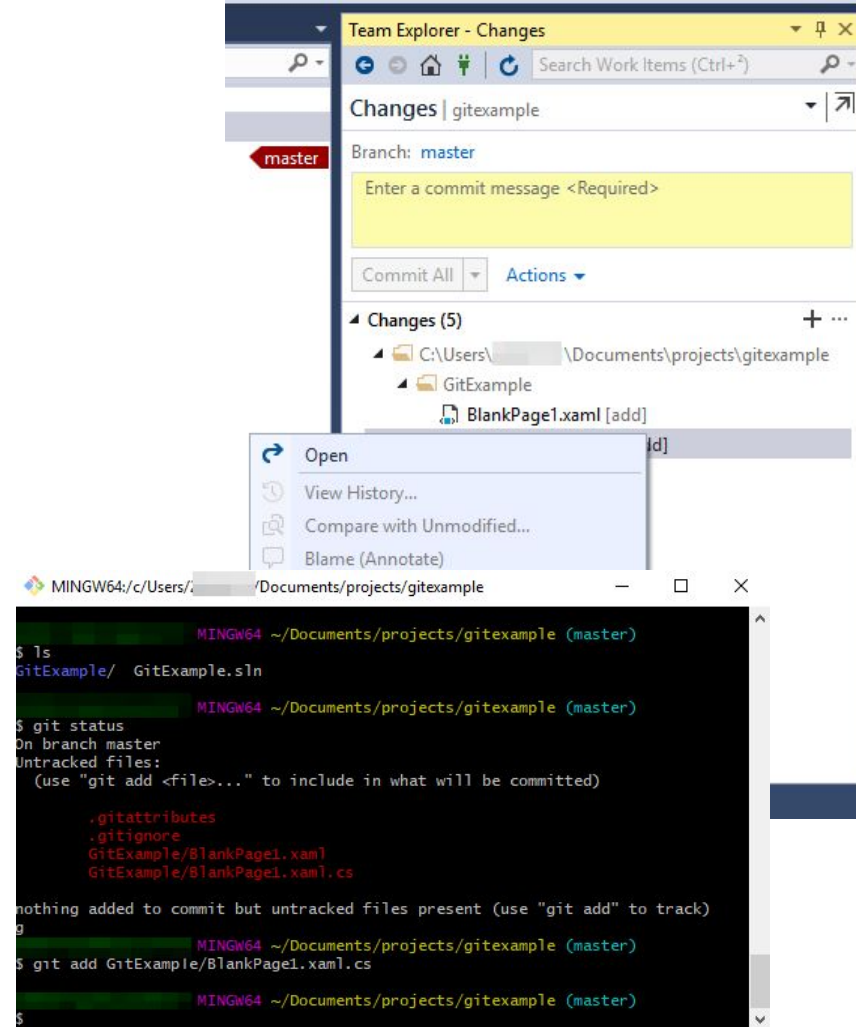
Git

- Open source project gestart door Linus Torvalds
- Heel populair
- Lokale kopie van de master repository
 - Veranderingen lokaal toevoegen
 - Doorsturen naar server kan later



Git operaties

- Interactie met Git
 - Via de command line:
git <command>
 - [Git Bash](#)
 - Via GUI of integratie in development tool (bv: Visual Studio)

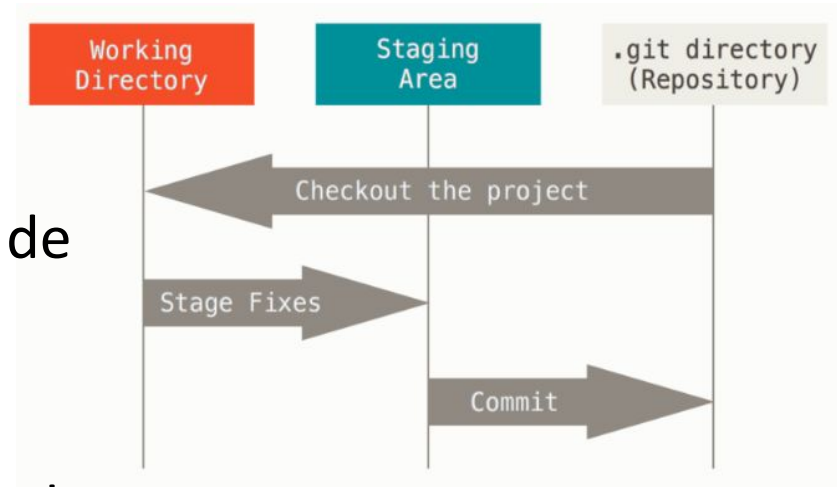


Initialisatie

- Nieuwe repository starten (lokaal):
 - `git init`
 - Eenmalig:
 - `git config --global user.name "naam"`
 - `git config --global user.email "jouwEmailadres"`
- Bestaande repository ophalen:
 - `git clone /path/to/repository`
 - `git clone username@host:/path/to/repository`

Toestand data

- Data bevindt zich in 3 toestanden:
 - Modified:
 - De data is aangepast in de working directory
 - Staged:
 - De data staat klaar voor de volgende commit
 - Committed:
 - Aanpassingen zijn doorgegeven naar de repository (lokaal)



Status

- Toestand van files/data bekijken:
 - git status

```
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   GitExample/BlankPage1.xaml.cs

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   GitExample/BlankPage1.xaml.cs
    modified:   GitExample/MainPage.xaml

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

    .gitattributes
    .gitignore
    GitExample/BlankPage1.xaml
```

The diagram illustrates the output of the `git status` command. It highlights three categories of file changes:

- Staged:** A new file, `GitExample/BlankPage1.xaml.cs`, is ready to be committed.
- Modified:** Two existing files, `GitExample/BlankPage1.xaml.cs` and `GitExample/MainPage.xaml`, have been modified but are not yet staged.
- (untracked):** Three files, `.gitattributes`, `.gitignore`, and `GitExample/BlankPage1.xaml`, are present in the working directory but have not been tracked by Git.

Stage

- File/data klaar zetten voor een commit:
 - `git add <filename>`
- Controleerbaar via `git status`
- Kunnen ook meerdere files tegelijk zijn
- Opnieuw iets gewijzigd? -> opnieuw add!

Commit

- Commando:
 - `git commit -m "Boodschap die in de log getoond wordt"`
- **ALTIJD** een goede message voorzien bij een commit
- Na een commit zitten de veranderingen in de repository (lokaal)
- Controleerbaar via:
 - `git log`

```
$ git log
commit 175c6f10933856ebbd645ff3ddf9d1256c0a4753 (HEAD -> master)
Author: 
Date: Thu Jun 7 12:06:05 2018 +0200

    Initial commit
```

SHA-1 Hash

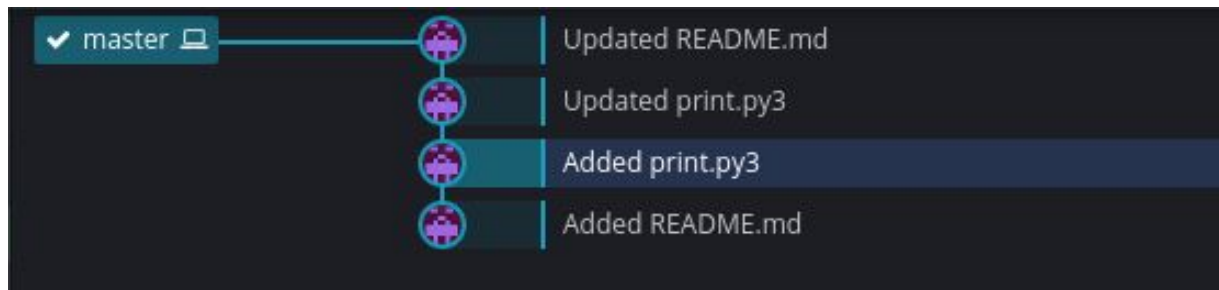
- Elke commit is een toevoeging van veranderingen
- Een change is een reeks bytes
- Elke reeks bytes kan omgezet worden in SHA-1 hash
- Deze hash is uniek en is de 'naam' van de commit en kan bijvoorbeeld gebruikt worden om terug te keren naar een vorige versie
- Bijvoorbeeld:

```
$ git log
commit 175c6f10933856ebbd645ff3ddf9d1256c0a4753 (HEAD -> master)
Author: 
Date: Thu Jun 7 12:06:05 2018 +0200

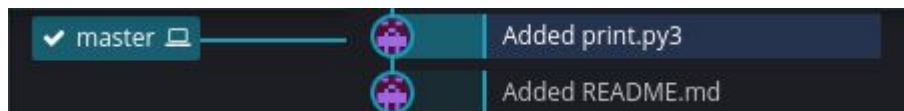
    Initial commit
```

Wijzigingen ongedaan maken

Optie 1: teruggaan in de tijd

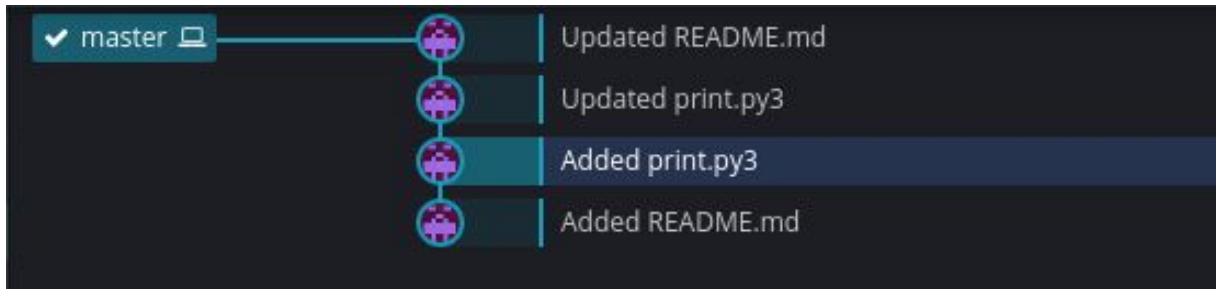


```
$ git log --oneline
9ab387c Updated README.md
aa29753 Updated print.py3
150ffb8 Added print.py3
1037b1d Added README.md
$ git reset --hard 150ffb8
```

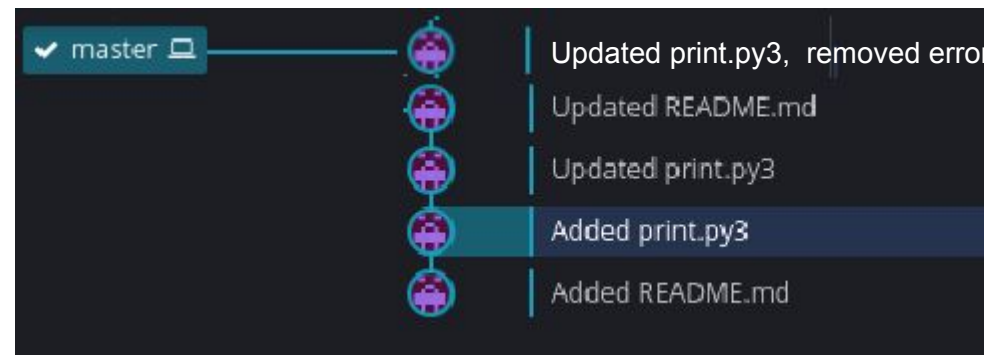


Wijzigingen ongedaan maken

Optie 2: teruggaan naar een oude commit en nieuwe commit met aangebrachte wijzigingen maken



```
$ git log --oneline
9ab387c Updated README.md
aa29753 Updated print.py3
150fffb8 Added print.py3
1037b1d Added README.md
$ git checkout -f 150fffb8
# bekijk de oude toestand
$ git checkout master
# maak wijzigen, add, commit
$ vi print.py3
$ git add print.py3
$ git commit -m "Updated print.py3, removed error"
```



Wijzigingen ongedaan maken

Optie 3: unstaged changes ongedaan maken

```
$ rm -rf *  
# oops :-)  
$ git reset --hard HEAD
```

Push/Pull

- Commando:
 - `git push origin master`
- Brengt de veranderingen van de lokale repository over naar de remote repository
- Commando:
 - `git pull`
- Ophalen van veranderingen op de master repository naar de lokale

Online repository

Create a New Repository x


← → ↻ <https://github.com/new> ☆ ABP

Search GitHub Pull requests Issues Marketplace Explore

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner **Repository name**

 **tim** / ✓

Great repository names are short and memorable. Need inspiration? How about **bookish-lamp**.

Description (optional)

☒ **Public**
Anyone can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

☒ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** | Add a license: **None** ⓘ

Create repository

Online repository

The screenshot shows the GitHub interface for the repository 'tim/zebra'. At the top, there's a dark navigation bar with the GitHub logo, a search bar, and links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below this, the repository name 'tim / zebra' is displayed, along with statistics: 1 Unwatch, 0 Stars, and 0 Forks. A secondary navigation bar includes links for 'Code', 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki', 'Settings', and 'Insights'. The main content area shows a message: 'No description, website, or topics provided.' with an 'Edit' button. Below this, repository statistics are listed: 1 commit, 1 branch, 0 releases, and 1 contributor. A row of buttons includes 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and a green 'Clone or download' button. A dropdown menu is open from the 'Clone or download' button, showing options to 'Clone with HTTPS' (with a question mark icon), 'Use SSH', and 'Download ZIP'. The 'Clone with HTTPS' option is selected, displaying the URL 'https://github.com/tim/zebra.git' with a copy icon. The repository's commit history is visible, showing a single commit by 'tim' with the message 'Initial commit' and a file named 'README.md'. The repository name 'zebra' is also displayed at the bottom left.

tim / zebra

Unwatch 1 Star 0 Fork 0

<> Code Issues 0 Pull requests 0 Projects 0 Wiki Settings Insights

No description, website, or topics provided. Edit

Add topics

1 commit 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

tim committed on GitHub Initial commit

README.md Initial commit

README.md

zebra

Clone with HTTPS ? Use SSH


Use Git or checkout with SVN using the web URL.


https://github.com/tim/zebra.git


Download ZIP


```
$ git clone https://github.com/tim/zebra.git
$ cd zebra
$ ls
README.md
$ vi print.py3
$ git add print.py3
$ git commit -m "Added print.py3"
[master b01039b] Added print.py3
 1 file changed, 1 insertion(+)
 create mode 100644 print.py3
$ git push origin master
Username for 'https://github.com': tim
Password for 'https://tim@github.com':
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 294 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/tim/zebra.git
 a8a3582..b01039b  master -> master
```

Online repository

 2 commits

 1 branch

 0 releases

 1 contributor

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾

 **tim** Added print.py3 Latest commit b01039b 21 minutes ago [README.md](#) Initial commit 24 minutes ago [print.py3](#) Added print.py3 21 minutes ago **README.md**

zebra

Merge conflicts

situatie: na pull wordt er een andere wijziging doorgevoerd in de online repository.
push kan niet automatisch gemerged worden

```
$ git push origin master
Username for 'https://github.com': tim
Password for 'https://tim@github.com':
To https://github.com/tim/zebra.git
 ! [rejected]          master -> master (fetch first)
error: failed to push some refs to
'https://github.com/tim/zebra.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```


Merge conflicts

```
$ git pull origin master
```

```
remote: Counting objects: 3, done.
```

```
remote: Compressing objects: 100% (2/2), done.
```

```
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
```

```
Unpacking objects: 100% (3/3), done.
```

```
From https://github.com/willekeij/zebra
```

```
* branch                master      -> FETCH_HEAD
```

```
   b01039b..a2f40f1  master      -> o/master
```

```
Auto-merging print.py3
```

```
CONFLICT (content): Merge conflict in print.py3
```

```
Automatic merge failed; fix conflicts and then commit the result.
```

Merge conflicts

```
$ cat print.py3
<<<<<<< HEAD
print("ok")
=====
print("OK")
>>>>>>> a2f40f14af51b9167b37bdfc4bc49d02a59d195
```

```
# Editeer print.py3
git add print.py3
git commit -m "Resolved merge conflict"
git push origin master
```

Installatie & tutorials

Installatie:

<https://git-scm.com/downloads>

Informatie:

<http://rogerdudler.github.io/git-guide/>

Oefenen:

<https://try.github.io/>

<https://www.udemy.com/course/git-started-with-github/>

PE

- Maak GitHub account <https://github.com/>
 - VoornaamNaamPXL
 - PXL email adres
- Klik op de link op BlackBoard
 - Join classroom
- Volg de instructies in je persoonlijke repository!

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Security

Insights

Settings

1tina-opdracht-1-git-d-ries created by GitHub Classroom

Edit

[Manage topics](#)

1 commit

1 branch

0 releases

1 contributor

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾



d-ries Initial commit

Latest commit 307cedc 4 days ago



html

Initial commit

4 days ago



README.md

Initial commit

4 days ago



gebruik_git.md

Initial commit

4 days ago

README.md



Opgave GIT week 3

Clone de repository

Clone de repository naar jouw lokaal toestel. Instructies hoe je dit kan doen, kan je terug vinden in de slides op Blackboard onder week 6 of in het bestand gebruik_git.md.

TIP: git clone

Opdrachten

- Maak in je laatst gemaakte oefening een git repository. Voeg alle bestanden van deze oefening toe aan je repo.
- Maak op github een nieuwe repo en push de bestanden.
- Maak een wijziging in je bestanden en gebruik de commando's om de wijzigingen naar github te pushen.
- Maak de wijzigingen ongedaan door terug te springen naar de vorige commit.