

# Statische websites

---

VERSIEBEHEER

# Overzicht

- ↘ Wat is git?
- ↘ Setup
- ↘ Repositories
- ↘ How to use (commando's)
- ↘ Branches
- ↘ .gitignore

# Wat is git?

Git is **een applicatie** die ervoor zorgt dat je **versiebeheer** kan gebruiken op jouw device

Git is **absoluut niet** de enige versiebeheer oplossing die er bestaat, maar wel veruit **de populairste**

Versiebeheer is **een makkelijke manier** om **code te bewaren** en **te delen** tussen meerdere devices of gebruikers

Bij het downloaden van git krijg je toegang tot enkele **commando's** op jouw device

# Wat is git?

GIT != GitHub, GitLab, Bitbucket, SourceForge, ...



# Hoe kan je git gebruiken?

Command-line Interface (CLI)

Graphical User Interface (GUI)

- <https://desktop.github.com/>
- <https://www.sourcetreeapp.com/>
- <https://www.gitkraken.com/>
- <https://www.syntevo.com/smartgit/>
- <https://marketplace.visualstudio.com/items?itemName=eamodio.gitlens> (Visual Studio Code plugin)

# GUI vs CLI

- Desktop applicaties zijn goed voor **ervaren** git gebruikers
- Onervaren git gebruikers die een desktop applicatie gebruiken om versiebeheer te doen gaan **snel fouten maken** en **data verliezen**
- Desktop applicaties doen **eigenlijk niet meer** dan **in de achtergrond de commando's runnen** die we vandaag gaan leren.
- Je kan **absoluut niet méér** met een desktop applicatie dan met **de cli**, integendeel, desktop applicaties hebben vaak **limitaties**.
- De **beste manier** om git te leren is via een cli
- Er zijn maar **enkele commando's** die je moet kennen om git te gebruiken
- Je kan git gebruiken vanuit **een terminal naar keuze**

# Wat is een repository of in't kort repo?

## Dictionary

Definitions from [Oxford Languages](#) · [Learn more](#)

Search for a word



# repository

/rɪˈpɒzɪt(ə)ri/

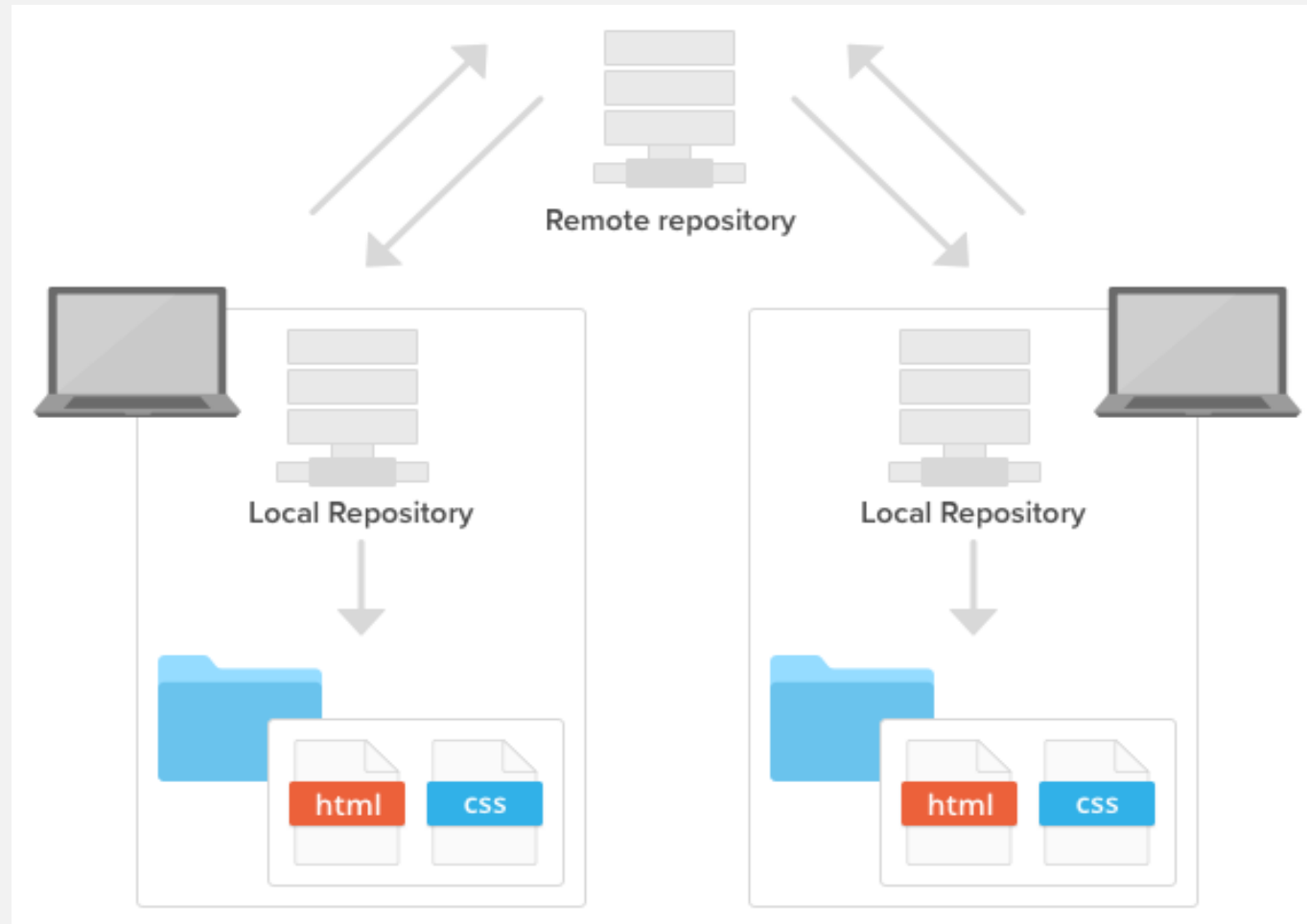
*noun*

a place where or receptacle in which things are or may be stored.

"a deep repository for nuclear waste"

- a place where something, especially a natural resource, is found in significant quantities.  
"accessible repositories of water"
- **COMPUTING**  
a central location in which data is stored and managed.  
"the metadata will be aggregated in a repository"

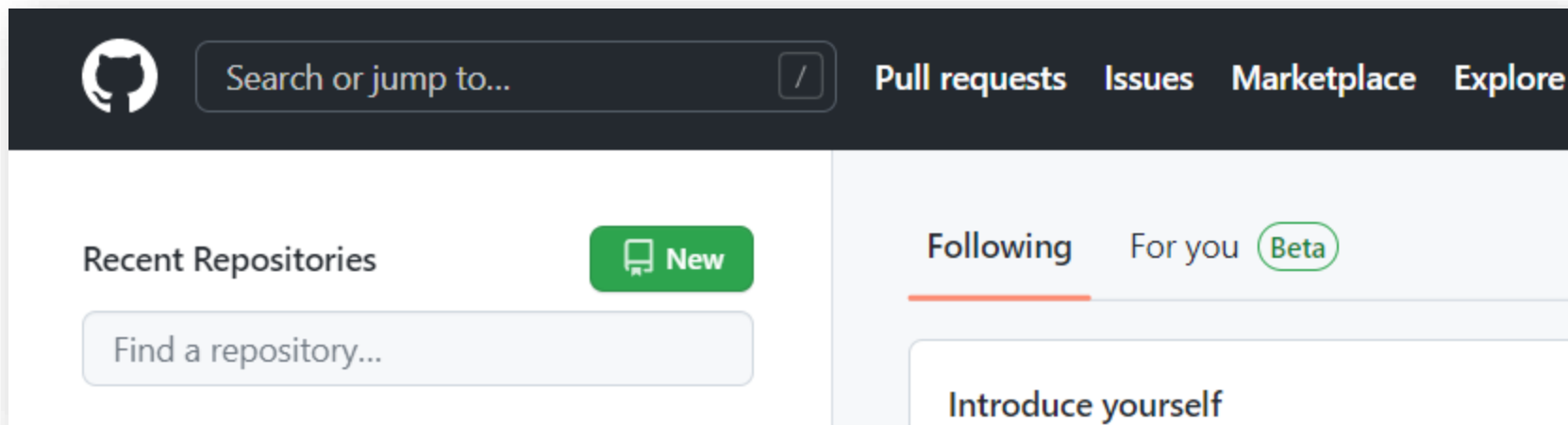
# Welke soorten repo's zijn er?








# Repo aanmaken

- Ga naar <https://github.com/>
- **Sign up** met je Thomas More e-mailadres
- Recent Repositories » New




# Repo aanmaken

- Repository name
- Description (optional)
- Public or private
- README file
- .gitignore
- License

 Search or jump to...  [Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#) 

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


Owner \*  
 niels-tm ▾


Repository name \*  
timetolearngit ✓

Great repository names are short and memorable. Need inspiration? How about [reimagined-broccoli?](#)

Description (optional)

I'm learning git!

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

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

**Initialize this repository with:**  
Skip this step if you're importing an existing repository.


☒ **Add a README file**  
This is where you can write a long description for your project. [Learn more.](#)


**Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: **None** ▾


**Choose a license**  
A license tells others what they can and can't do with your code. [Learn more.](#)

License: **None** ▾

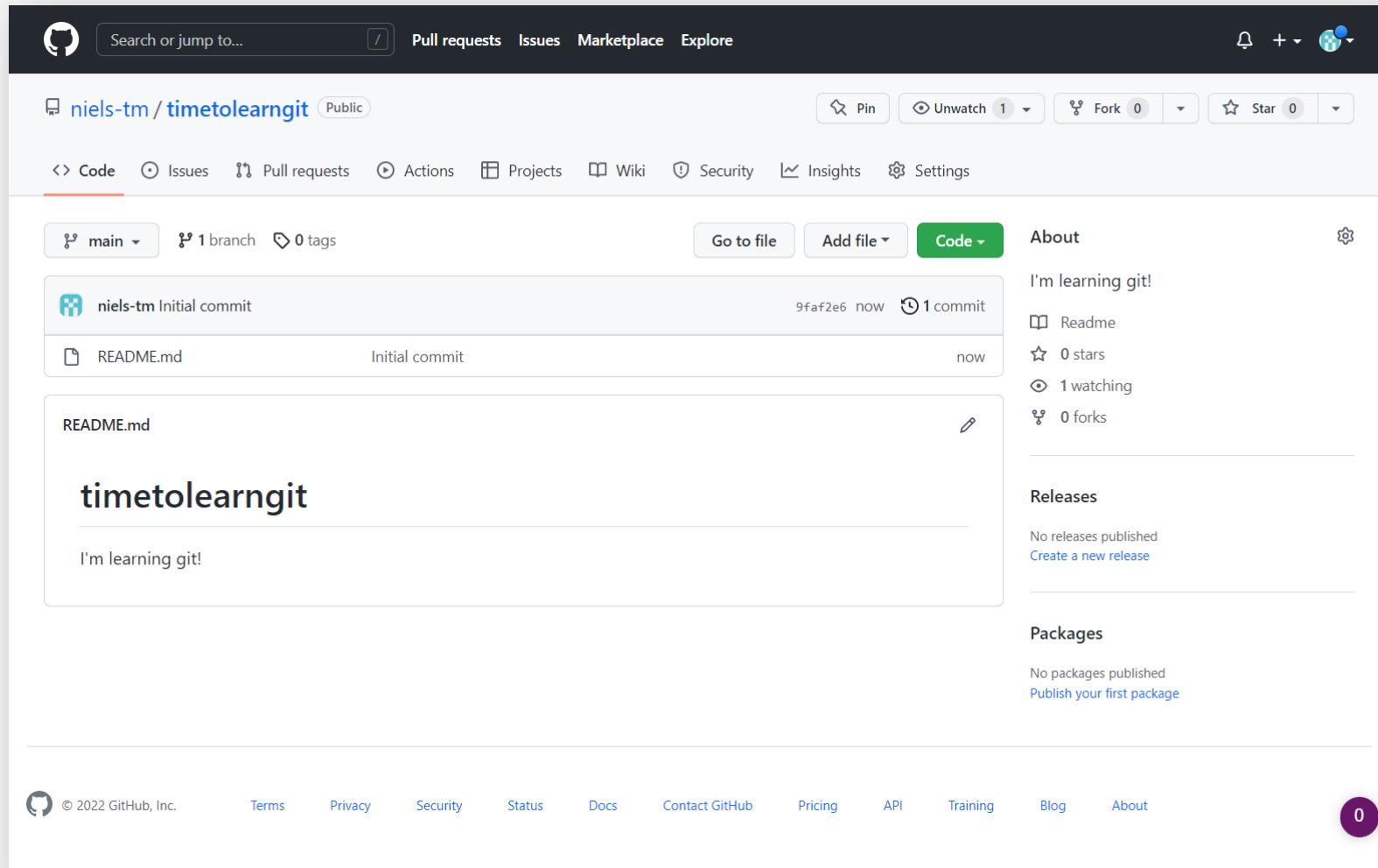
This will set  **main** as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account.

Create repository

 © 2022 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

# Repo aanmaken



# Download git

## Setup

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



Search entire site...

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.



### About

The advantages of Git compared to other source control systems.



### Documentation

Command reference pages, Pro Git book content, videos and other material.



### Downloads

GUI clients and binary releases for all major platforms.



### Community

Get involved! Bug reporting, mailing list, chat, development and more.



**Pro Git** by Scott Chacon and Ben Straub is available to **read online for free**. Dead tree versions are available on [Amazon.com](https://www.amazon.com).



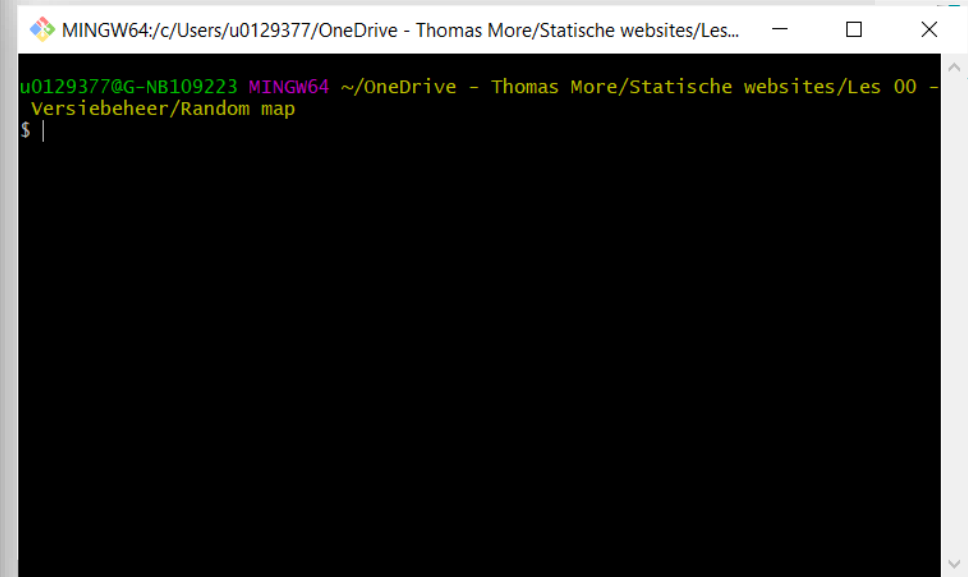
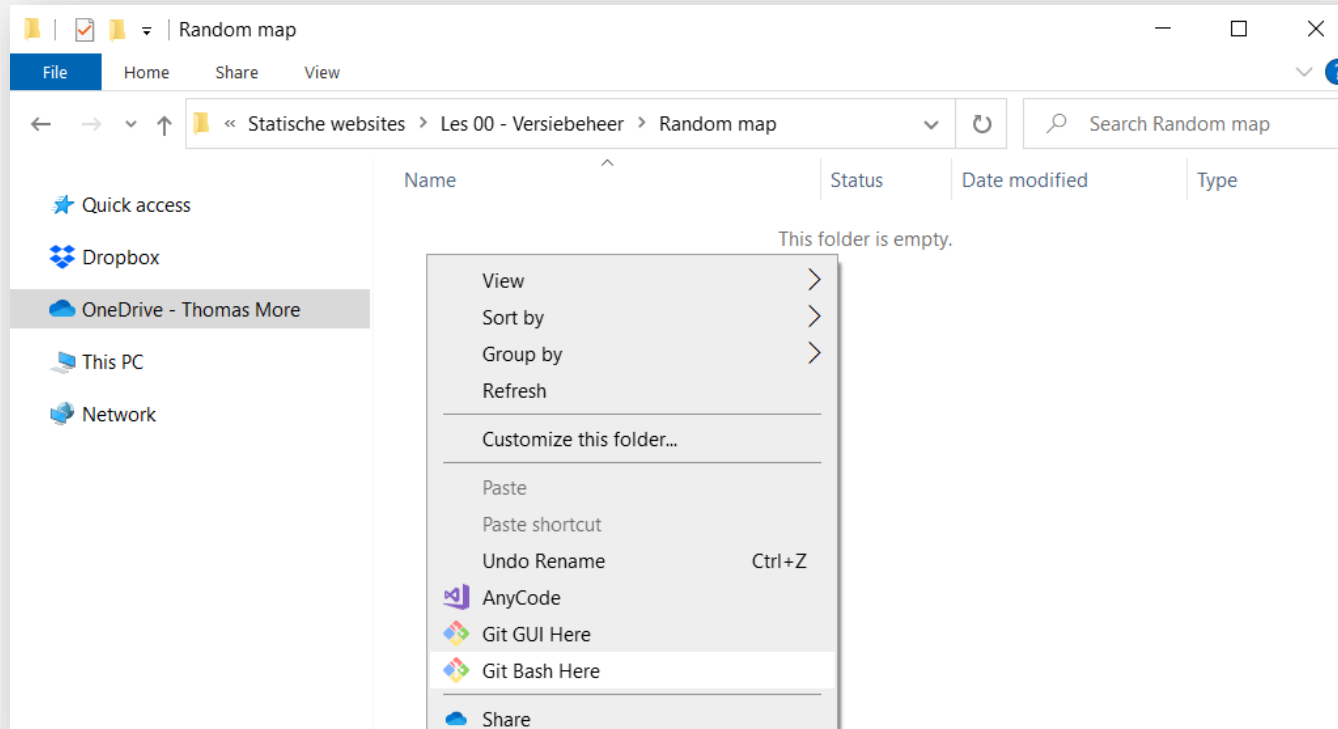
 **Windows GUIs**

 **Mac Build**

 **T**

 **S**

# Hoe begin je best met git?



Git Bash

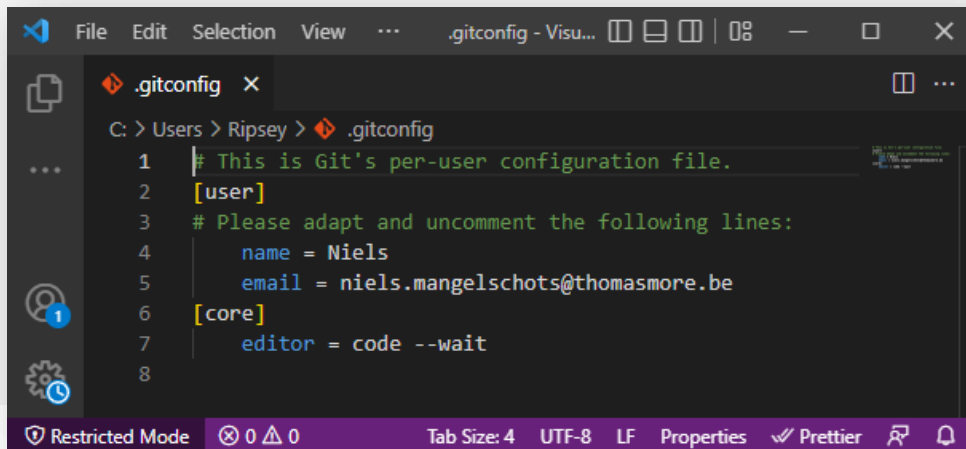
# Git instellen

## Identiteit instellen

1. `git config --global user.name 'Niels'`
2. `git config --global user.email 'niels.mangelschots@thomasmore.be'`

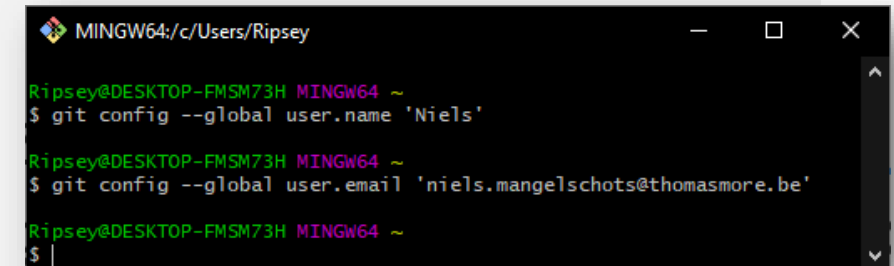
## Editor instellen

3. Zorg ervoor dat [Visual Studio Code](#) is installed
4. `git config --global core.editor "code --wait"`
5. Instellingen bekijken door `git config --global -e`



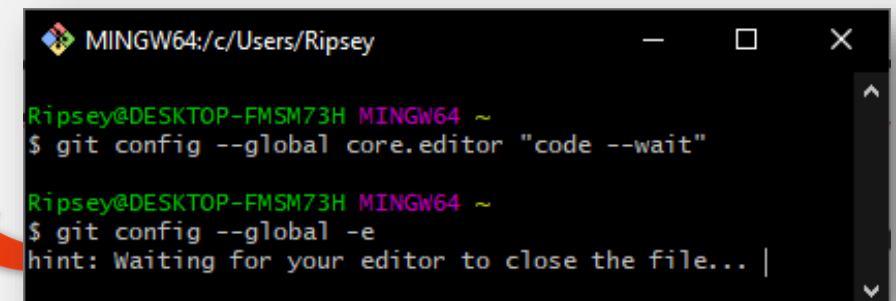
A screenshot of the Visual Studio Code editor interface. The file explorer on the left shows the '.gitconfig' file selected. The main editor area displays the contents of the file, which is a standard Git configuration template. The configuration is set for a user named 'Niels' with the email 'niels.mangelschots@thomasmore.be' and the editor set to 'code --wait'. The status bar at the bottom indicates 'Restricted Mode' and '0 0 0' errors.

```
.gitconfig
C: > Users > Ripsey > .gitconfig
1 # This is Git's per-user configuration file.
2 [user]
3 # Please adapt and uncomment the following lines:
4   name = Niels
5   email = niels.mangelschots@thomasmore.be
6 [core]
7   editor = code --wait
8
```



A screenshot of a Windows Command Prompt terminal window titled 'MINGW64:/c/Users/Ripsey'. It shows the execution of the first two commands from the 'Identiteit instellen' section. The first command sets the global user name to 'Niels', and the second sets the global user email to 'niels.mangelschots@thomasmore.be'. The prompt is ready for the next command.

```
MINGW64:/c/Users/Ripsey
Ripsey@DESKTOP-FMSM73H MINGW64 ~
$ git config --global user.name 'Niels'
Ripsey@DESKTOP-FMSM73H MINGW64 ~
$ git config --global user.email 'niels.mangelschots@thomasmore.be'
Ripsey@DESKTOP-FMSM73H MINGW64 ~
$
```

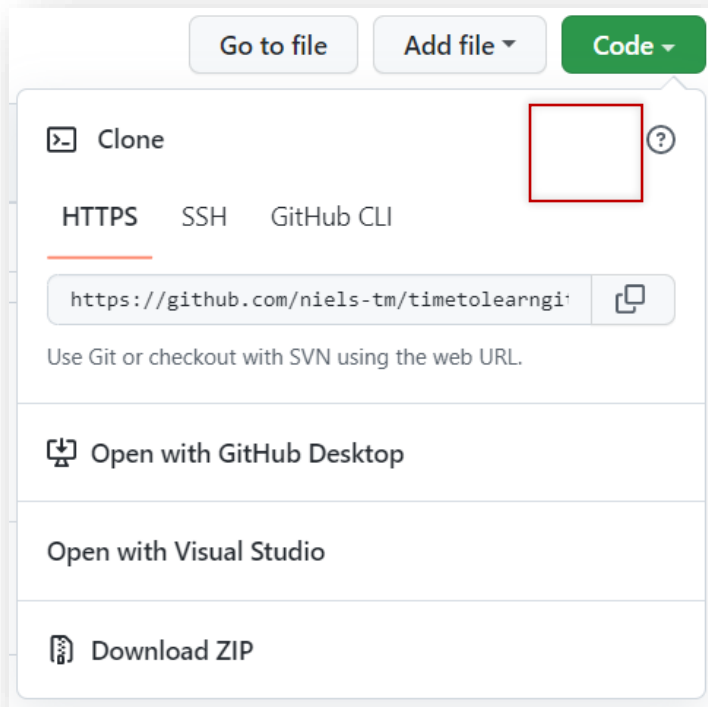


A screenshot of a Windows Command Prompt terminal window titled 'MINGW64:/c/Users/Ripsey'. It shows the execution of the third command from the 'Editor instellen' section. The command sets the global core editor to 'code --wait'. The output shows a hint: 'Waiting for your editor to close the file...'. A red arrow points from the terminal window above to this one, indicating the sequence of steps.

```
MINGW64:/c/Users/Ripsey
Ripsey@DESKTOP-FMSM73H MINGW64 ~
$ git config --global core.editor "code --wait"
Ripsey@DESKTOP-FMSM73H MINGW64 ~
$ git config --global -e
hint: Waiting for your editor to close the file... |
```

# git clone

- Maakt een **exakte kopie** van wat er op een remote repository (GitHub) staat op jouw lokaal systeem
- Eens we dit gedaan hebben kunnen aanpassingen beginnen maken

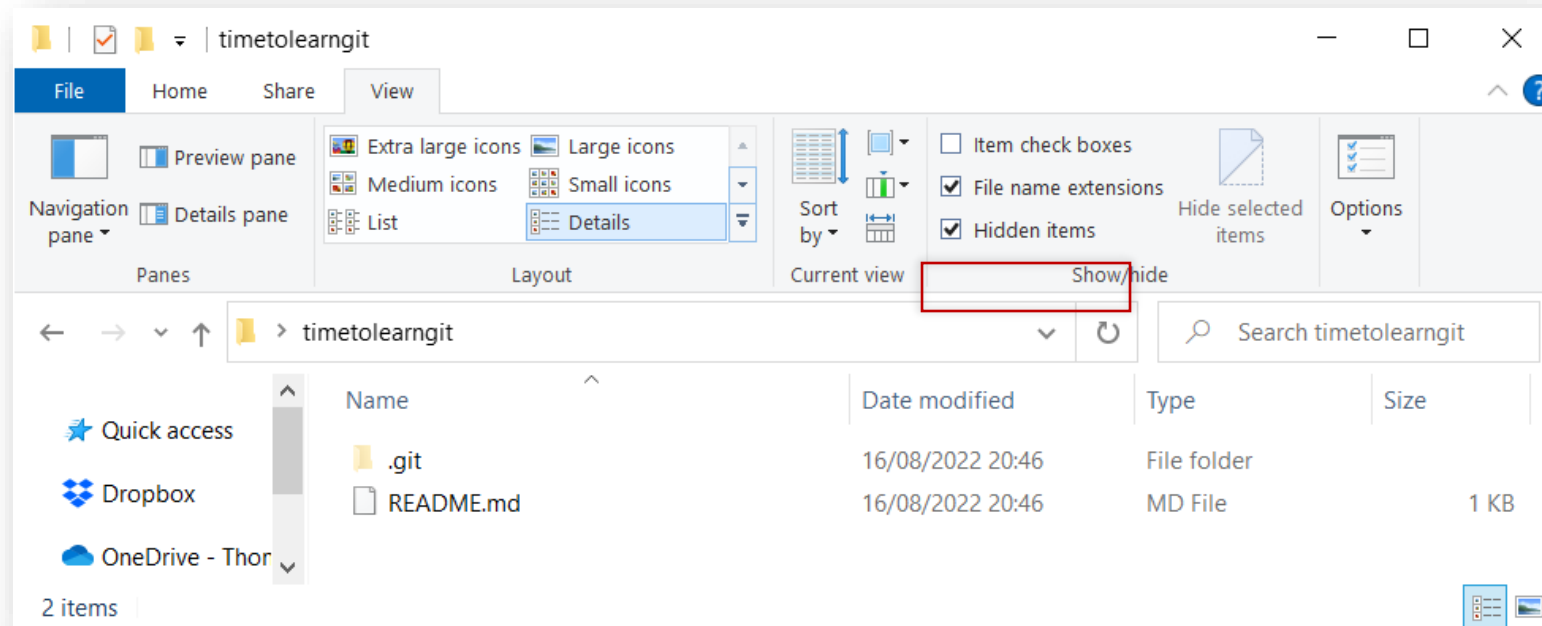


```
MINGW64:/c/Users/u0129377/Desktop
u0129377@G-NB109223 MINGW64 ~/Desktop
$ git clone https://github.com/niels-tm/timetolearngit.git
Cloning into 'timetolearngit'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

u0129377@G-NB109223 MINGW64 ~/Desktop
$ |
```

# git clone

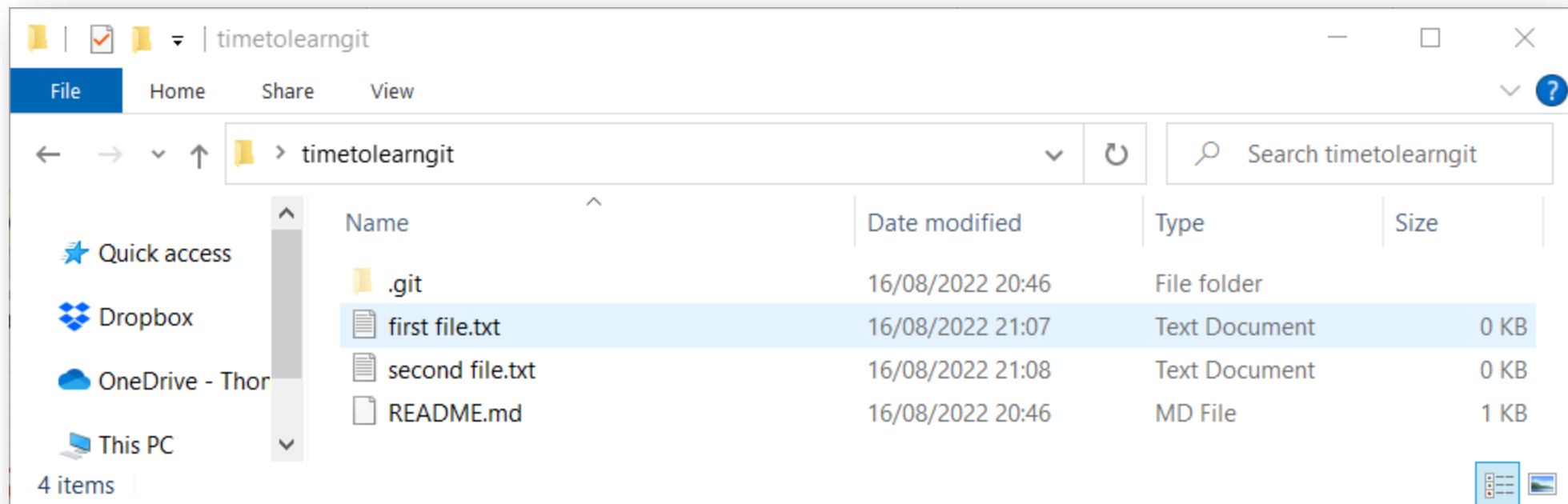
- .git folder » zorgt ervoor dat git **data kan bijhouden** ivm dit project, nooit aankomen
- Alle bestanden die je van **de remote repository** (GitHub) afgehaald hebt, in ons geval dus enkel 'README.md'





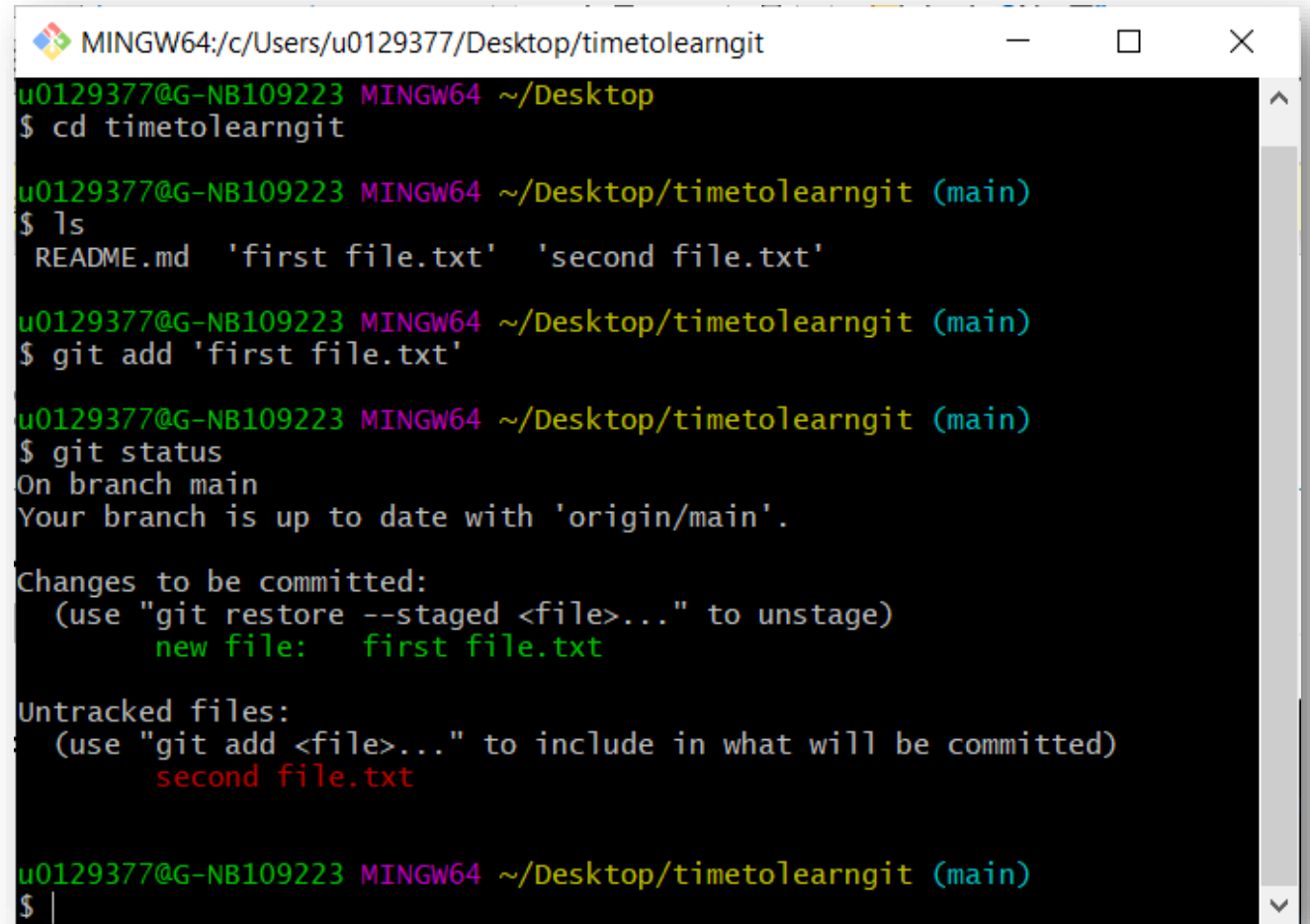
# Bestanden toevoegen

- Voeg twee random bestanden toe aan je map
- Hoe krijgen we nu onze bestanden naar de **remote repository** op GitHub?



# git add

- Voeg één of meerdere bestanden/folders toe aan de staging area
- git add 'first file.txt'
  - × Voegt dit bestand toe aan de staging area



```
MINGW64:/c/Users/u0129377/Desktop/timetolearngit
u0129377@G-NB109223 MINGW64 ~/Desktop
$ cd timetolearngit

u0129377@G-NB109223 MINGW64 ~/Desktop/timetolearngit (main)
$ ls
README.md 'first file.txt' 'second file.txt'

u0129377@G-NB109223 MINGW64 ~/Desktop/timetolearngit (main)
$ git add 'first file.txt'

u0129377@G-NB109223 MINGW64 ~/Desktop/timetolearngit (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

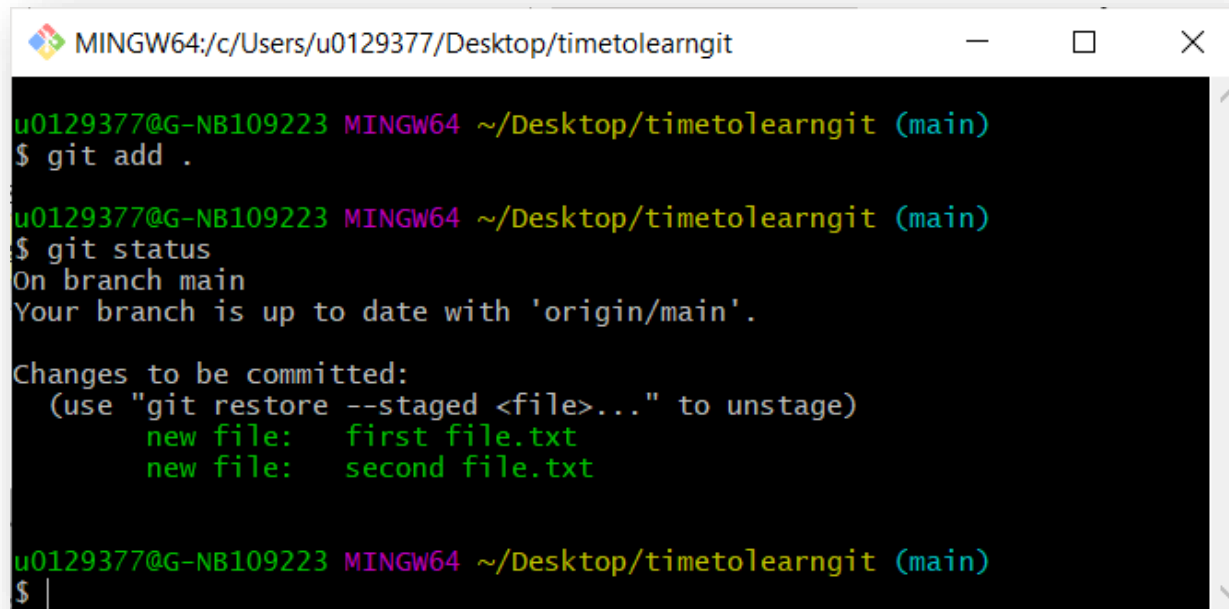
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   first file.txt

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

u0129377@G-NB109223 MINGW64 ~/Desktop/timetolearngit (main)
$ |
```

# git add

- We willen bestanden niet **één voor één** toevoegen aan de staging area
- Voeg ze **allemaal toe** door het commando **'git add.'** te gebruiken



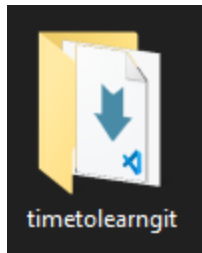
```
MINGW64:/c:/Users/u0129377/Desktop/timetolearngit
u0129377@G-NB109223 MINGW64 ~/Desktop/timetolearngit (main)
$ git add .
u0129377@G-NB109223 MINGW64 ~/Desktop/timetolearngit (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   first file.txt
    new file:   second file.txt

u0129377@G-NB109223 MINGW64 ~/Desktop/timetolearngit (main)
$ |
```

# Flow: staging area

Staging area (device)	Local repo (device)	Remote repo (GitHub)
<p>..is de plaats waar files terechtkomen na <b>git add</b></p> <p><b>git add &lt;filename&gt;</b> voegt één specifiek bestand toe</p> <p><b>git add .</b> voegt alle bestanden toe</p> <p><b>git status</b> controleert welke bestanden er wel of niet in de staging area zitten</p>		

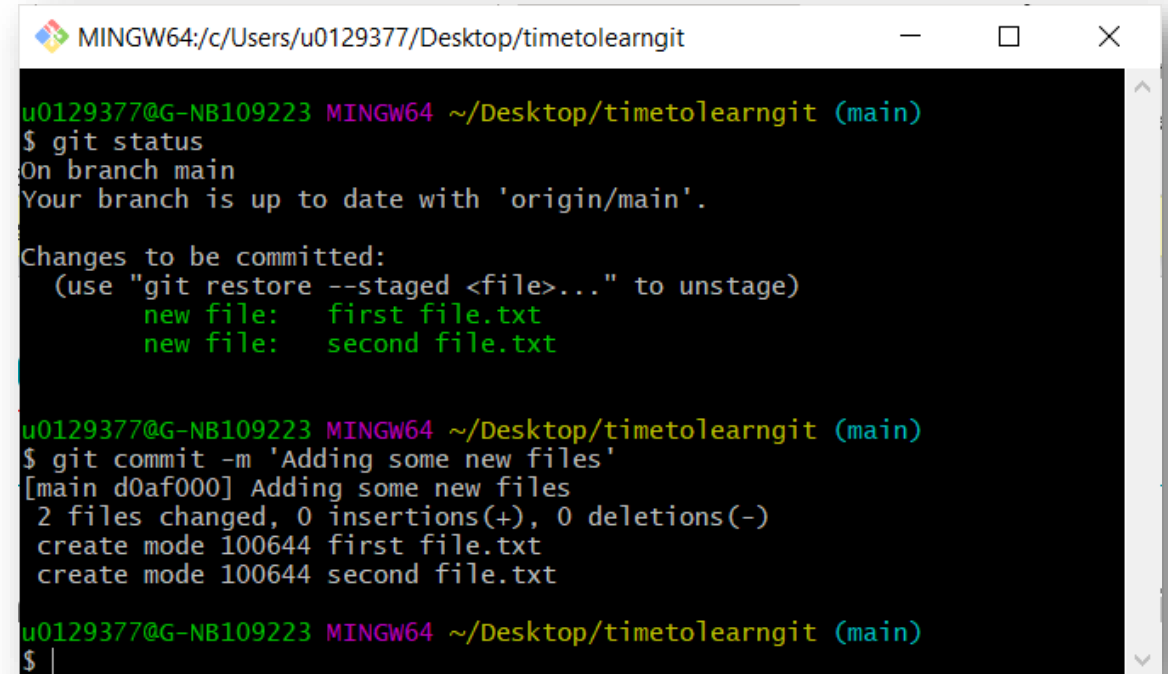


git add

Staging area

# git commit

- Voegt alle bestanden die in de **staging area** staan toe aan **local repository**
- Een commit moet altijd **wat commentaar bevatten** waarin je omschrijft wat je veranderd hebt
- Bijvoorbeeld: **git commit -m 'Adding some new files'**
- Elke commit heeft een random id
  - ✗ Deze commit kreeg **d0af000**



```
MINGW64:/c/Users/u0129377/Desktop/timetolearngit

u0129377@G-NB109223 MINGW64 ~/Desktop/timetolearngit (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

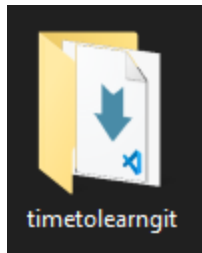
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   first file.txt
        new file:   second file.txt

u0129377@G-NB109223 MINGW64 ~/Desktop/timetolearngit (main)
$ git commit -m 'Adding some new files'
[main d0af000] Adding some new files
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 first file.txt
 create mode 100644 second file.txt

u0129377@G-NB109223 MINGW64 ~/Desktop/timetolearngit (main)
$
```

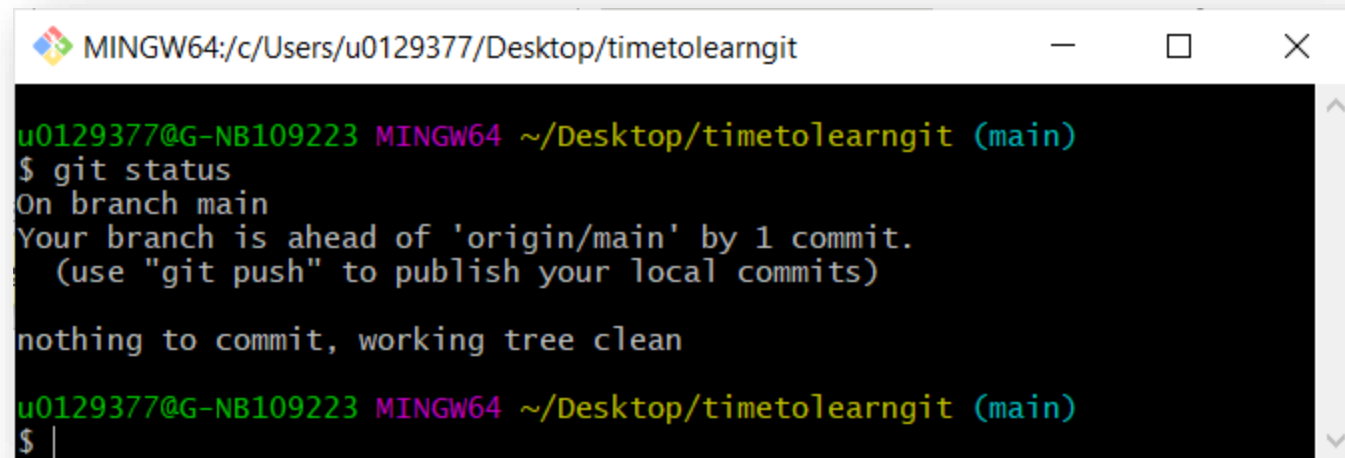
# Flow: local repo

Staging area (device)	Local repo (device)	Remote repo (GitHub)
<p>..is de plaats waar files terechtkomen na <b>git add</b></p> <p><b>git add &lt;filename&gt;</b> voegt één specifiek bestand toe</p> <p><b>git add .</b> voegt alle bestanden toe</p> <p><b>git status</b> controleert welke bestanden er wel of niet in de staging area zitten</p>	<p>..is de plaats waar files terechtkomen na <b>git commit</b></p> <p>altijd met <b>-m</b> en wat commentaar tussen quotes</p> <p>niet verplicht om na elke commit te pushen</p>	



# git status

- Git status geeft **de status** weer van git
- Hieronder vertelt git status ons dat er **een commit gebeurd is** op onze **local repository** maar dat deze veranderingen **nog niet toegepast zijn** op de **remote repository**
- We staan momenteel dus **één commit** voor
- Git status raadt ons aan om **'git push'** te gebruiken om onze veranderingen te publiceren



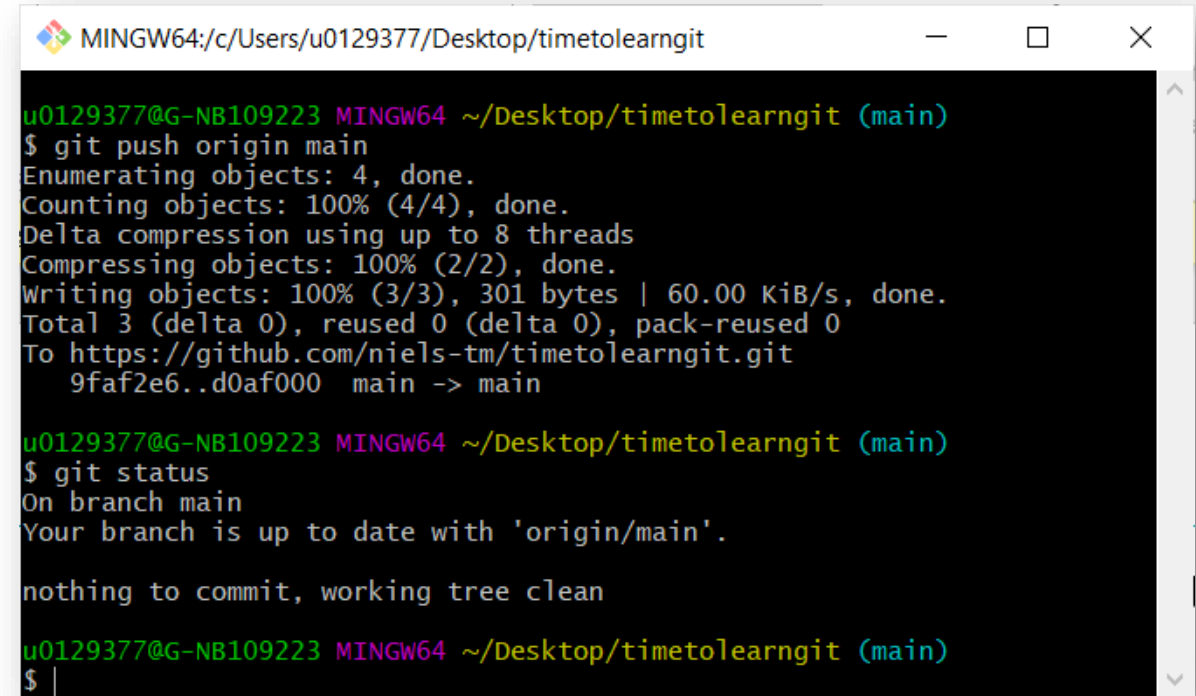
```
MINGW64:/c:/Users/u0129377/Desktop/timetolearngit
u0129377@G-NB109223 MINGW64 ~/Desktop/timetolearngit (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

u0129377@G-NB109223 MINGW64 ~/Desktop/timetolearngit (main)
$
```

# git push

- Pusht, of duwt, de bestanden die in de **local repo** staan naar de **remote repo**
- We moeten altijd **specifiëren naar welke branch** we pushen, hier straks meer over
- Bijvoorbeeld: **git push origin main**
- Git bash **toont altijd** op **welke branch** je op welk moment zit
- Git push geeft onderaan aan welke **commits** er naar welke **branch** zijn **gepusht**
  - × In ons geval maar eentje, **d0af000**



```
MINGW64:/c/Users/u0129377/Desktop/timetolearngit

u0129377@G-NB109223 MINGW64 ~/Desktop/timetolearngit (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 301 bytes | 60.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/niels-tm/timetolearngit.git
   9faf2e6..d0af000  main -> main

u0129377@G-NB109223 MINGW64 ~/Desktop/timetolearngit (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

u0129377@G-NB109223 MINGW64 ~/Desktop/timetolearngit (main)
$ |
```



# Pushing commits

The screenshot shows the GitHub interface for a repository named 'timetolearngit' by user 'niels-tm'. The repository is public and has 1 branch (main) and 0 tags. The commit history shows three commits: an initial commit for README.md 3 hours ago, and two subsequent commits for 'first file.txt' and 'second file.txt' 9 minutes ago, both adding new files. The README.md content is visible, showing the repository name 'timetolearngit' and the text 'I'm learning git!'. The right sidebar contains links to the README, 0 stars, 1 watching, 0 forks, and sections for Releases and Packages, both indicating no published items.

niels-tm / timetolearngit (Public)

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

niels-tm Adding some new files d0af000 9 minutes ago 2 commits

File	Commit Message	Time
README.md	Initial commit	3 hours ago
first file.txt	Adding some new files	9 minutes ago
second file.txt	Adding some new files	9 minutes ago

README.md

## timetolearngit

I'm learning git!

**About**

I'm learning git!

- Readme
- 0 stars
- 1 watching
- 0 forks

**Releases**

No releases published  
[Create a new release](#)

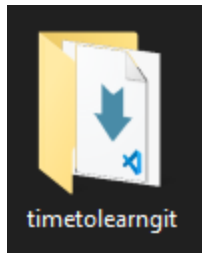
**Packages**

No packages published  
[Publish your first package](#)

© 2022 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

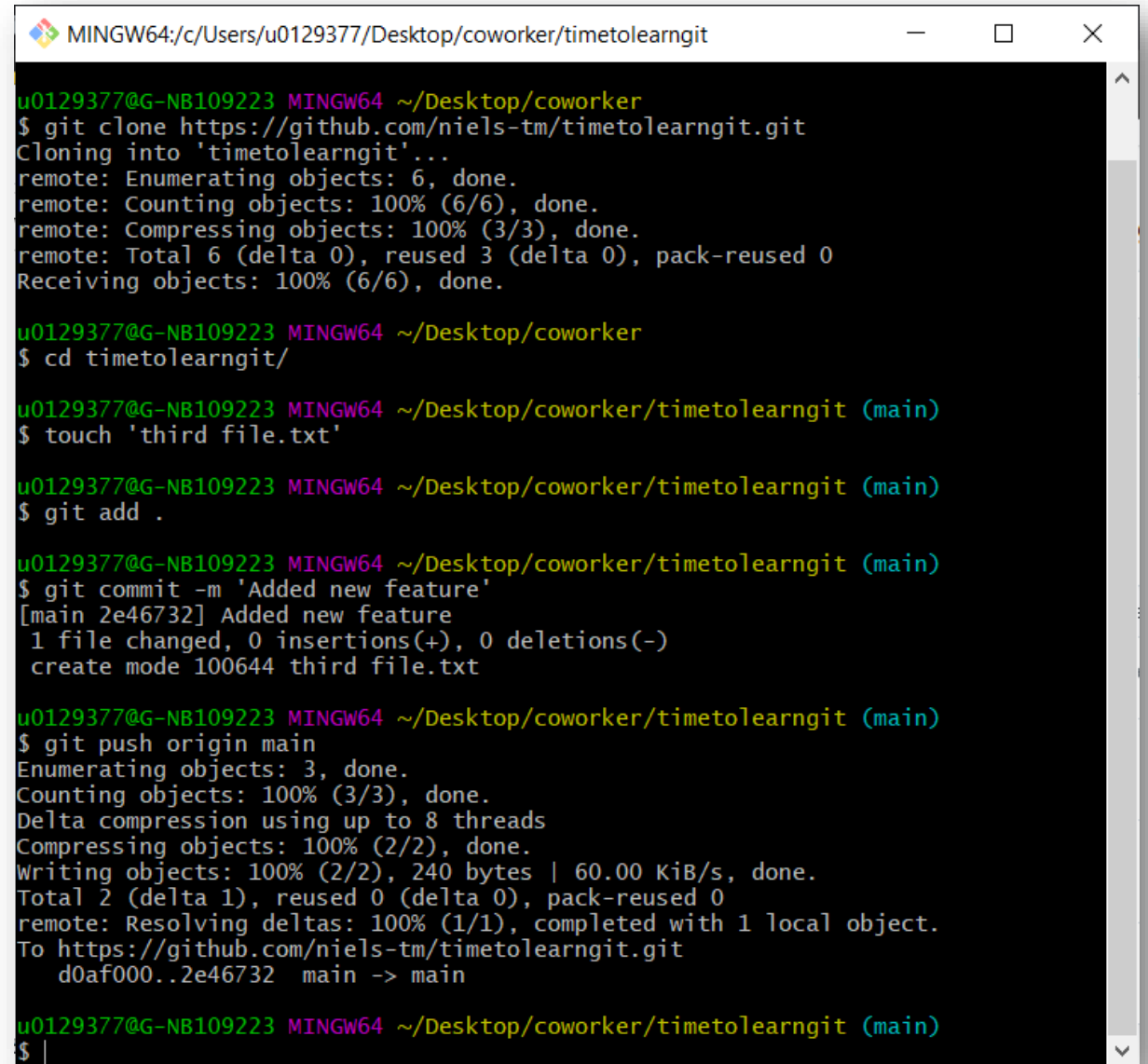
# Flow: remote repo

Staging area (device)	Local repo (device)	Remote repo (GitHub)
<p>..is de plaats waar files terechtkomen na <b>git add</b></p> <p><b>git add &lt;filename&gt;</b> voegt één specifiek bestand toe</p> <p><b>git add .</b> voegt alle bestanden toe</p> <p><b>git status</b> controleert welke bestanden er wel of niet in de staging area zitten</p>	<p>..is de plaats waar files terechtkomen na <b>git commit</b></p> <p>altijd met <b>-m</b> en wat commentaar tussen quotes</p> <p>niet verplicht om na elke commit te pushen</p>	<p>..is de plaats waar files terechtkomen na <b>git push</b></p> <p>verplaatst al onze aanpassingen naar <b>GitHub</b></p> <p>altijd gebruiken op deze manier: <b>git push origin branchnaam</b></p>



# Wat als..

.. een teamlid gebruik maakt van **git clone** om ook mee te werken aan het project en voegt een **belangrijk** bestand toe?



```
MINGW64:/c:/Users/u0129377/Desktop/coworker/timetolearngit

u0129377@G-NB109223 MINGW64 ~/Desktop/coworker
$ git clone https://github.com/niels-tm/timetolearngit.git
Cloning into 'timetolearngit'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.

u0129377@G-NB109223 MINGW64 ~/Desktop/coworker
$ cd timetolearngit/

u0129377@G-NB109223 MINGW64 ~/Desktop/coworker/timetolearngit (main)
$ touch 'third file.txt'

u0129377@G-NB109223 MINGW64 ~/Desktop/coworker/timetolearngit (main)
$ git add .

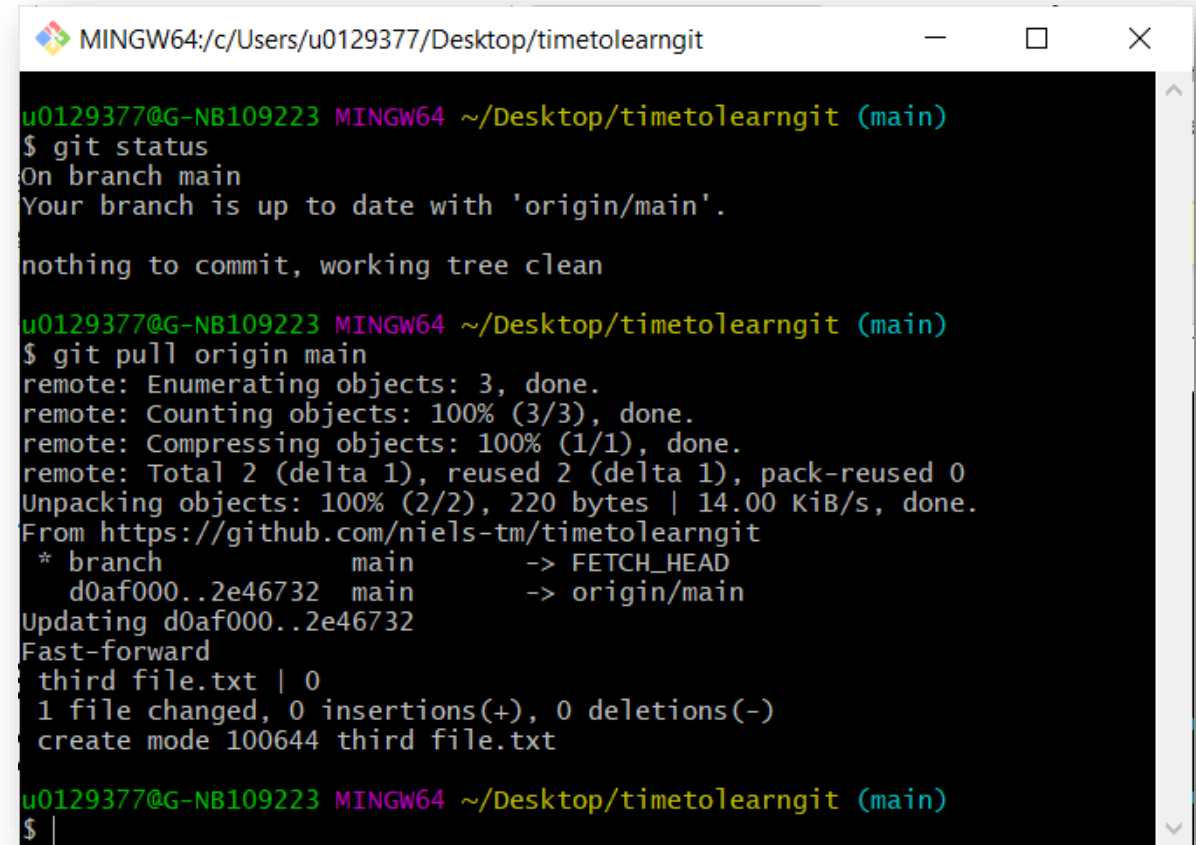
u0129377@G-NB109223 MINGW64 ~/Desktop/coworker/timetolearngit (main)
$ git commit -m 'Added new feature'
[main 2e46732] Added new feature
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 third file.txt

u0129377@G-NB109223 MINGW64 ~/Desktop/coworker/timetolearngit (main)
$ git push origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 240 bytes | 60.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/niels-tm/timetolearngit.git
d0af000..2e46732  main -> main

u0129377@G-NB109223 MINGW64 ~/Desktop/coworker/timetolearngit (main)
$ |
```

# git pull

- Pullt, of trekt, de nieuwe bestanden die in de **remote repository** staan naar je lokale folder aka **local repository**
- We moeten **altijd specificeren** van welke **branch** we pullen
- Bijvoorbeeld: **git pull origin main**

A terminal window titled 'MINGW64:/c:/Users/u0129377/Desktop/timetolearngit' showing the execution of git commands. The first command is 'git status', which reports that the branch 'main' is up to date with 'origin/main' and the working tree is clean. The second command is 'git pull origin main', which performs a fast-forward merge from the remote repository. The output shows the enumeration, counting, and compressing of objects, followed by the unpacking of objects and the creation of a new file 'third file.txt'.

```
MINGW64:/c:/Users/u0129377/Desktop/timetolearngit

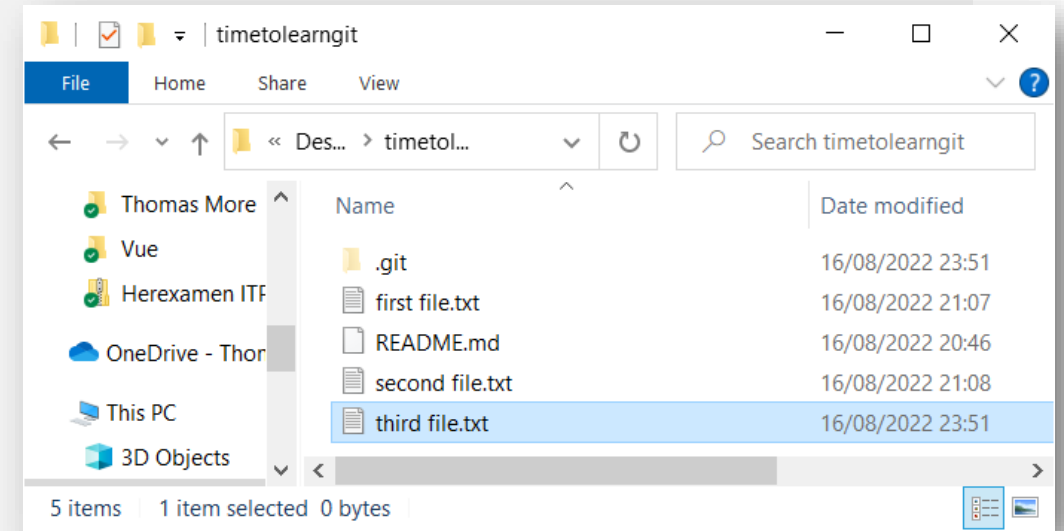
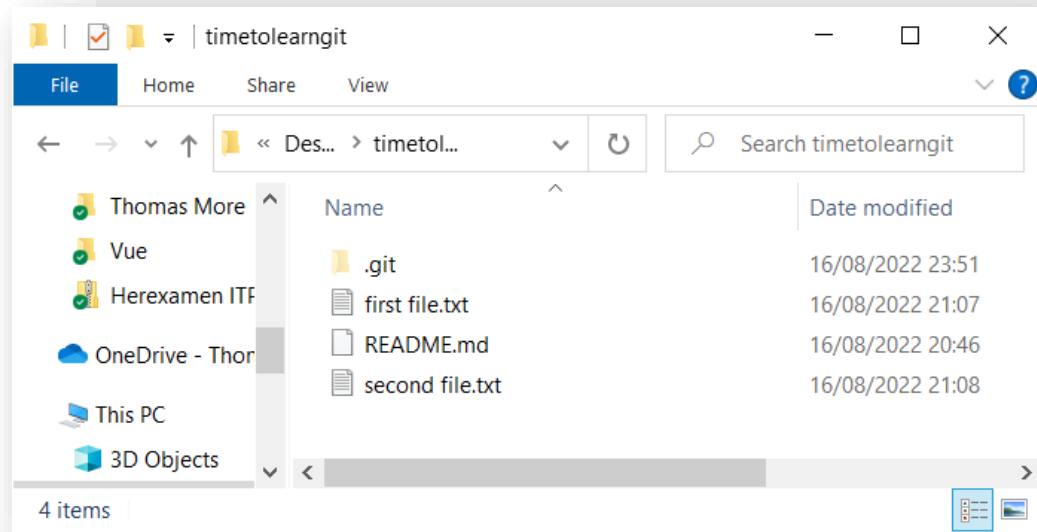
u0129377@G-NB109223 MINGW64 ~/Desktop/timetolearngit (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

u0129377@G-NB109223 MINGW64 ~/Desktop/timetolearngit (main)
$ git pull origin main
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 2 (delta 1), reused 2 (delta 1), pack-reused 0
Unpacking objects: 100% (2/2), 220 bytes | 14.00 KiB/s, done.
From https://github.com/niels-tm/timetolearngit
* branch                main          -> FETCH_HEAD
  d0af000..2e46732      main          -> origin/main
Updating d0af000..2e46732
Fast-forward
 third file.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 third file.txt

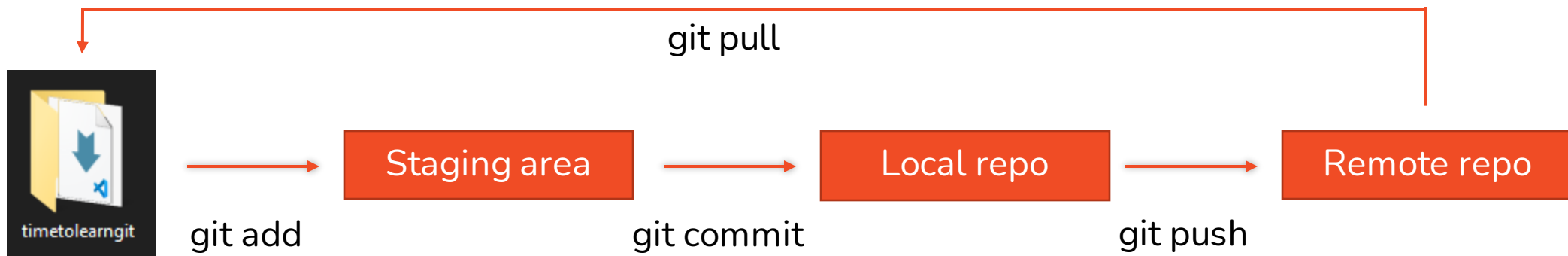
u0129377@G-NB109223 MINGW64 ~/Desktop/timetolearngit (main)
$
```

# Wat heeft dit veranderd?



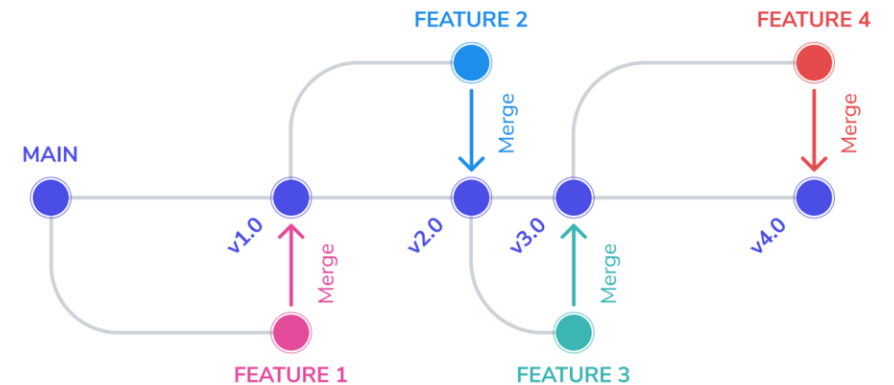
# Complete flow

Staging area (device)	Local repo (device)	Remote repo (GitHub)
<p>..is de plaats waar files terechtkomen na <b>git add</b></p> <p><b>git add &lt;filename&gt;</b> voegt één specifiek bestand toe</p> <p><b>git add .</b> voegt alle bestanden toe</p> <p><b>git status</b> controleert welke bestanden er wel of niet in de staging area zitten</p>	<p>..is de plaats waar files terechtkomen na <b>git commit</b></p> <p>altijd met <b>-m</b> en wat commentaar tussen quotes</p> <p>niet verplicht om na elke commit te pushen</p>	<p>..is de plaats waar files terechtkomen na <b>git push</b></p> <p>verplaatst al onze aanpassingen naar <b>GitHub</b></p> <p>altijd gebruiken op deze manier: <b>git push origin branchnaam</b></p> <p>haal files van de <b>remote repo</b> naar <b>local repo</b></p> <p>altijd gebruiken op deze manier: <b>git pull origin branchnaam</b></p>



# Wat is een branch?

- Een branch, of tak, is **de plaats** waarop je momenteel **aan het werken bent** binnenin git
- Tot nu toe waren we **altijd op main** aan het werken, (vroeger master)
- We kunnen **zo veel branches** aanmaken als we willen
- Wanneer we dit doen wordt er een **kopie van de data gemaakt** van de branch waarop we momenteel zitten
- Deze data wordt dan in de **nieuwe branch** gestopt
- Branches zijn erg interessant om **verschillende versies** te hebben van jouw codebase. Zo kan je **dingen uitproberen** zonder te riskeren dat je de originele codebase aanpast met onafgewerkte code of code waar mogelijk nog bugs in zitten



# git checkout

- `git checkout` is het commando dat je toelaat van **branch te wisselen**
- `git checkout -b branchnaam` maakt **een nieuwe branch** aan, en **wisselt** je **automatisch** naar deze **nieuwe branch**
- In latere modules krijg je meer info over branches!
- Toch nu al meer info over branches?
  - × [Git & GitHub Tutorial for Beginners #8 – Branches](#)
  - × [Git & GitHub Tutorial for Beginners #9 - Merging Branches \(& conflicts\)](#)



# .gitignore

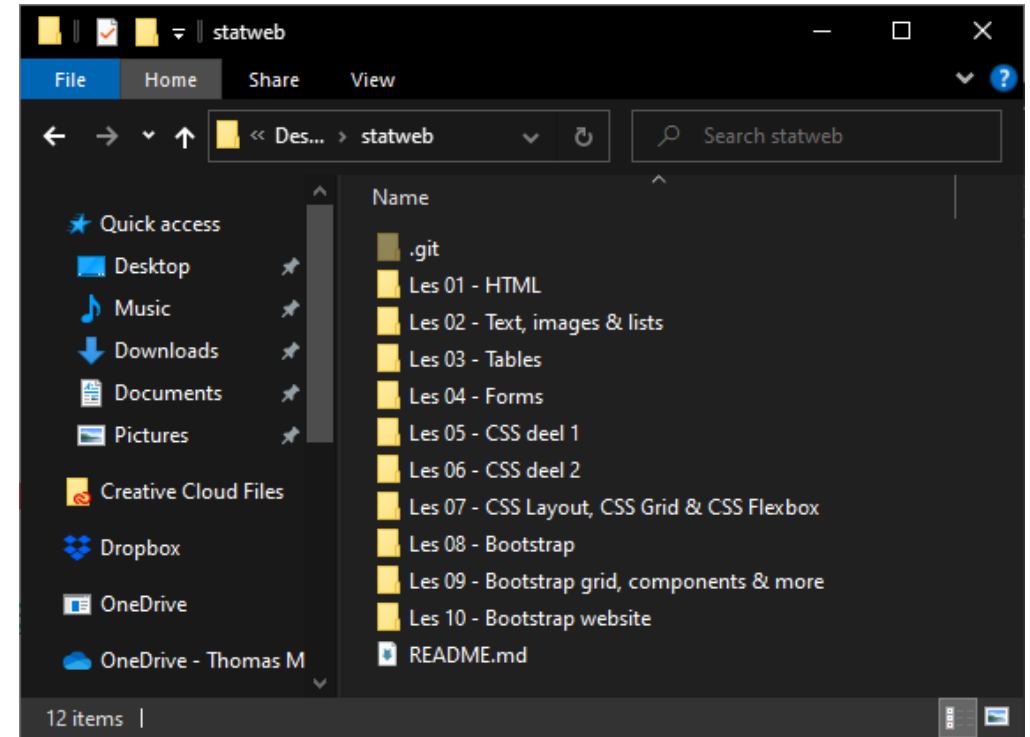
- Een bestand waarin je definieert welke bestanden of directories **niet** in naar de **staging area** gaan bij het gebruik van **git add .** of **git add 'some file name'**
- Logischerwijs komen deze bestanden/directories dan ook **nooit** in de **local repo** met **git commit**
- En dus ook nooit op de **remote repo** met **git push**

## Welke bestanden best niet naar staging area sturen?

- Het gemakkelijkste voorbeeld is de **.vs folder** die **Visual Studio** aanmaakt wanneer je een **nieuw** project aanmaakt
- Deze folder bevat bestanden die **alleen** relevant zijn voor jouw lokale **Visual Studio** installatie, je wilt deze dus **niet** doorgeven aan je collega's
- Ook bestanden waar **secrets, poorten, API namen, database info** etc. in staan geven we niet door
- De **.vs folder** blokkeer je door het volgende te zetten in het **.gitignore** bestand
  - × **.vs/**

# GitHub

- Maak een repo '**statweb**' voor alle oefeningen van deze module
- Push **na elke les** alle oefeningen naar **GitHub**
- Zet je repo op **public** of zorg dat je **docent toegang heeft** door een invite te sturen
  - × [niels.mangelschots@thomasmore.be](mailto:niels.mangelschots@thomasmore.be)
  - × [stijn.willekens@thomasmore.be](mailto:stijn.willekens@thomasmore.be)
  - × [bart.ooms@thomasmore.be](mailto:bart.ooms@thomasmore.be)
- Upload URL van je repo op **Canvas**  
<https://github.com/niels-tm/statweb>






## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner \*

Repository name \*

 niels-tm / statweb 

Great repository names are short and memorable. Need inspiration? How about [ideal-train?](#)

Description (optional)

Statweb oefeningen

☒ Public

Anyone on the internet can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☒ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore


Choose which files not to track from a list of templates. [Learn more.](#)


.gitignore template: None

Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

License: None

This will set  main as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account.

Create repository



Search or jump to...


[Pulls](#) [Issues](#) [Marketplace](#) [Explore](#)





niels-tm / statweb


Public


 Pin


 Unwatch 1


 Fork 0


 Star 0


 Code


 Issues


 Pull requests

 Actions


 Projects

 Wiki

 Security

 Insights



 main

Go to file

Add file

Code

About



niels-tm Initial commit

1 minute ago



1



README.md

Initial commit

1 minute ago


README.md





# statweb


Statweb oefeningen

## Statweb oefeningen

 Readme

 0 stars

 1 watching

 0 forks

## Releases

No releases published














[Create a new release](#)

## Packages

No packages published

[Publish your first package](#)

# Resultaat einde module

	niels-tm maps update	1ae4a0e 35 seconds ago	 4 commits
	Les 01 - HTML	maps	yesterday
	Les 02 - Text, images & lists	maps update	1 hour ago
	Les 03 - Tables	maps update	1 hour ago
	Les 04 - Forms	maps update	1 hour ago
	Les 05 - CSS deel 1	maps	yesterday
	Les 06 - CSS deel 2	maps	yesterday
	Les 07 - CSS Layout, CSS Grid & CSS ...	maps update	35 seconds ago
	Les 08 - Bootstrap	maps update	1 hour ago
	Les 09 - Bootstrap grid, components ...	maps	yesterday
	Les 10 - Bootstrap website	maps	yesterday
	README.md	Initial commit	yesterday

# Lijst van commado's

- `cd <directory naam>` » navigeer van een directory naar een andere
- `cd ..` » een directory terug gaan
- `cd ../../..` » 3 directories terug gaan, etc.
- Gebruik de tab toets om van **autocompletion** te genieten
- `ls` » toon alle bestanden/directories in de map
- `ls -a` » toont ook verborgen bestanden/directories
- `cat` » schrijf de tekst die in een bestand staat neer in het console venster
- `git status` » Bekijk de status van de branch waarop je momenteel zit
- `git clone`
- `git add`
- `git commit`
- `git push`
- `git pull`
- `git checkout` » om te verwisselen van branch
- `git checkout -b` » voor een nieuwe branch

# Vragen

---