

# Appendix

Continuous Integration Basics

## INHOUD

<b>1</b>	<b>WERKEN MET BESTANDEN EN MAPPEN (WINDOWS 10)</b>	<b>3</b>
1.1	Inleiding	3
1.2	Bron	3
1.3	Onduidelijkheden?	3
<b>2</b>	<b>GIT STATUS</b>	<b>4</b>
2.1	Inleiding	4
2.2	Startsituatie simuleren	4
2.3	Verduidelijking	6
2.3.1	Waarom zijn ze wel zichtbaar na onze eerste commit?	7
2.4	Opties op git status	7
<b>3</b>	<b>WERKEN MET COMMANDO'S</b>	<b>8</b>
3.1	Inleiding	8
3.2	Startsituatie	8
3.3	Bestanden toevoegen aan je git repository (lokaal)	10
3.4	Je remote repository updaten met je huidige lokale repository	11
3.5	Je lokale repository updaten met je remote repository	11

# 1 WERKEN MET BESTANDEN EN MAPPEN (WINDOWS 10)

## 1.1 INLEIDING

---

Naar aanleiding van het eerste lesmoment van deze module werden er hier en daar nog vragen gesteld omtrent de basis werking en de concepten omtrent bestanden en mappen binnen Windows 10.

## 1.2 BRON

---

Indien je zelf nog niet meteen volledig mee bent hoe bestanden en mappen op je pc/laptop nu precies werken en waarvoor deze dienen kan je gerust je kennis even gaan bijwerken via onderstaande link.

[Windows 10 – Organizing Files and Folders](#) (*LinkedIn Learning*)

Toegang tot het LinkedIn Learning platform zou beschikbaar moeten zijn voor jou als student via het [Academic Software platform](#).

## 1.3 ONDUIDELIJKHEDEN?

---

Indien je nog steeds onduidelijkheden ondervindt in de werking van bestanden en/of mappen contacteer dan je lector via e-mail.

## 2 GIT STATUS

### 2.1 INLEIDING

Tijdens het recap moment van de eerste les kwamen we terecht op een situatie waar het `git status` commando ons in de subfolder enkel de inhoud toonde van de root folder. Deze situatie vroeg om wat onderzoekwerk en verduidelijking.



#### Opgelet

Dit is **aanvullende informatie** specifiek gericht op een vraag of voorval tijdens een lesmoment. Verdere informatie zal je verduidelijking van dit voorval opleveren en hier dieper op ingaan.

Deze informatie valt echter **buiten de basisscope van de cursus**.

### 2.2 STARTSITUATIE SIMULEREN

Onderstaande commando's leveren een gelijkaardige startsituatie op waar we het (*visuele*) probleem kunnen kaderen. We starten van in een lege folder zonder git repository.

```
git init
touch file1.md
touch file2.md
mkdir folderA
touch folderA/file3.md
git status
```

Na uitvoeren van deze commando's (*in een hiervoor voorziene folder uiteraard*) krijgen we onderstaande status terug.

```
ConsoleFriend@Desktop ~/Desktop/recap (master)
$ git status
On branch master

No commits yet

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

        file1.md
        file2.md
        folderA/

nothing added to commit but untracked files present (use "git add" to track)
```

We zien alle non tracked files, maar we zien inhoudelijk (*nog*) niet wat er precies aan bestanden in onze **folderA** als non tracked files zit.

Hiervoor gingen we tijdens het recap moment even naar deze folder zelf. `cd folderA`

Het uitvoeren van de `git status` gaf ons echter volgende uitkomst.

```
ConsoleFriend@Desktop ~/Desktop/recap/folderA (master)
$ git status
On branch master

No commits yet

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

    ../file1.md
    ../file2.md
    ./

nothing added to commit but untracked files present (use "git add" to track)
```

We zien terug enkel de bestanden in onze **root** folder. En deze keer zelfs helemaal niks die verwijst naar het bestand in **folderA**.

Als we echter eerst alles toevoegen aan onze git repository, denk aan de 2 stappen die hiervoor nodig zijn, door het stagen en commiten kregen we een ander beeld nadien.

```
git add .
git commit -m "Add new files"
touch folderA/file4.md
git status
```

```
ConsoleFriend@Desktop ~/Desktop/recap (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    folderA/file4.md

nothing added to commit but untracked files present (use "git add" to track)
```

Wijzigingen van folder gaf ons deze keer wel waar we in eerste instantie naar op zoek waren.

```
ConsoleFriend@Desktop ~/Desktop/recap/folderA (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    file4.md

nothing added to commit but untracked files present (use "git add" to track)
```

## 2.3 VERDUIDELIJKING

In de eerste fase (voor we bestanden in onze folderA gecommitt hebben) ziet git daadwerkelijk dat er in folderA bestanden aanwezig zijn die tot op heden nog niet getracked zijn.

Echter het `git status` commando heeft van nature de output (met optie `--long`) en deze zal steeds de relatieve paden weergeven t.o.v. onze **root folder**.

Er zijn echter wel een aantal opties die ons meteen de mogelijkheid geven om wel de nodige informatie te zien inclusief de bestanden die in de subfolder aanwezig zijn (voor onze eerste commit).

We vertrekken in deze terug van onderstaande startsituatie.

```
git init
touch file1.md
touch file2.md
mkdir folderA
touch folderA/file3.md
git status -u
```

Door de optie `-u` te gebruiken vragen we aan git om ons ook de naam van **untracked files** mee te geven. Deze zal ervoor zorgen dat **untracked files** die zich niet in de root bevinden ook getoond worden.

```
ConsoleFriend@Desktop ~/Desktop/recap (master)
$ git status -u
On branch master

No commits yet

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

    file1.md
    file2.md
    folderA/file3.md

nothing added to commit but untracked files present (use "git add" to track)
```

Idem dus als we `git status -u` vanuit onze **subfolder** uitvoeren.

```
ConsoleFriend@Desktop ~/Desktop/recap/folderA (master)
$ git status -u
On branch master

No commits yet

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

    ../file1.md
    ../file2.md
    file3.md

nothing added to commit but untracked files present (use "git add" to track)
```

### 2.3.1 WAAROM ZIJN ZE WEL ZICHTBAAR NA ONZE EERSTE COMMIT?

Wel, na onze eerste commit bevat onze **folderA** een bestand die we wel tracken, en werd toegevoegd aan git in de structuur. Vanaf dat punt krijgen we telkens wel de informatie te zien via `git status`.

## 2.4 OPTIES OP GIT STATUS

Het `git status` commando heeft tal van opties welke misschien ook wel even leuk zijn om mee te experimenteren. Bekijk gerust onderstaande screenshot even en experimenteer zelf ook gerust even met de combinatie van de opties! De link naar de algemene documentatie van git over het `git status` commando kan je op het einde van dit deel terugvinden.

```
ConsoleFriend@Desktop ~/Desktop/recap/folderA (master)
$ git status -s -b
## No commits yet on master
?? ../file1.md
?? ../file2.md
?? ../
?? ../

ConsoleFriend@Desktop ~/Desktop/recap/folderA (master)
$ git status -s -b -v
## No commits yet on master
?? ../file1.md
?? ../file2.md
?? ../
?? ../

ConsoleFriend@Desktop ~/Desktop/recap/folderA (master)
$ git status --porcelain
?? file1.md
?? file2.md
?? folderA/

ConsoleFriend@Desktop ~/Desktop/recap/folderA (master)
$ git status -u
On branch master

No commits yet

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

    ../file1.md
    ../file2.md
    file3.md

nothing added to commit but untracked files present (use "git add" to track)

ConsoleFriend@Desktop ~/Desktop/recap/folderA (master)
$ git status -uno
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

ConsoleFriend@Desktop ~/Desktop/recap/folderA (master)
$ git status -uall
On branch master

No commits yet

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

    ../file1.md
    ../file2.md
    file3.md

nothing added to commit but untracked files present (use "git add" to track)
```



#### git documentatie

De volledige documentatie van het `git status` commando kan je terugvinden via onderstaande link.

- [git status docs](#)

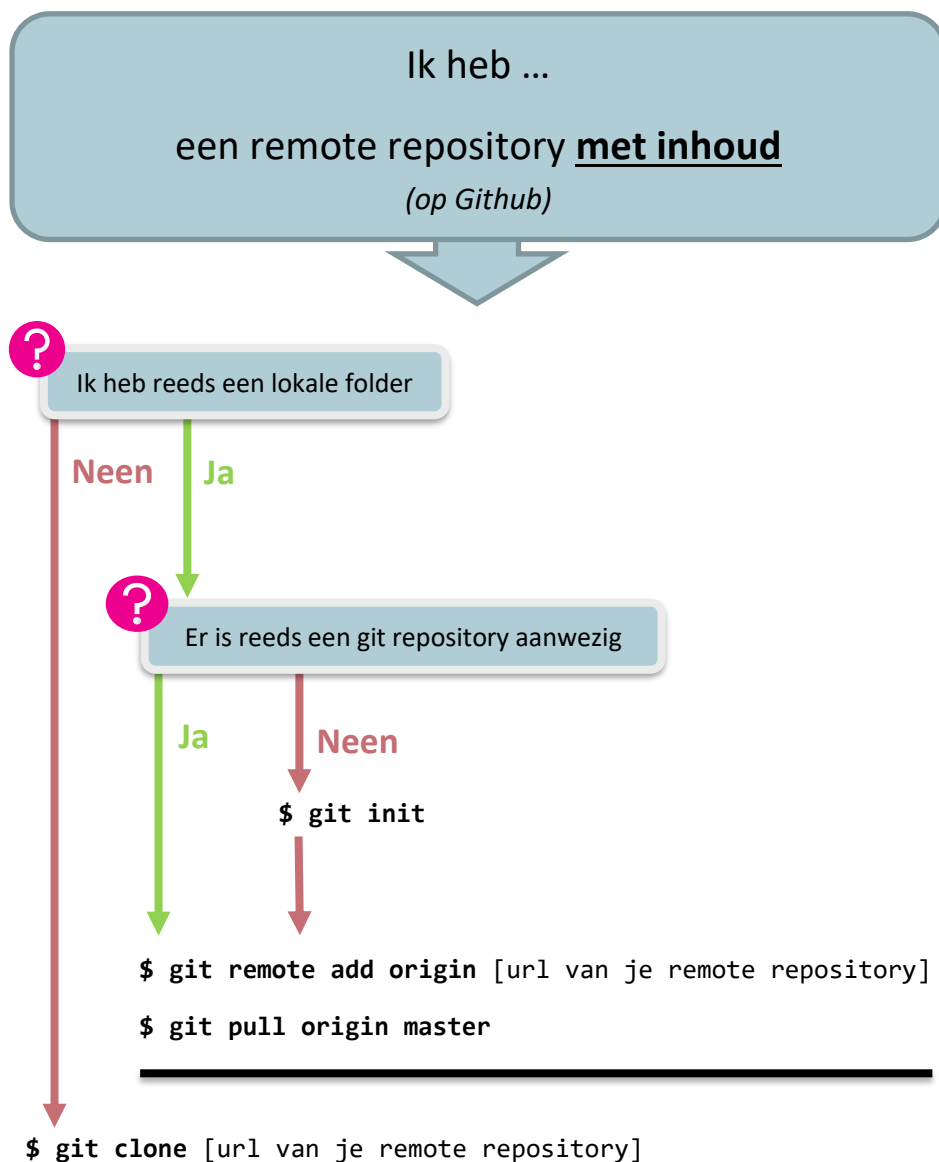
## 3 WERKEN MET COMMANDO'S

### 3.1 INLEIDING

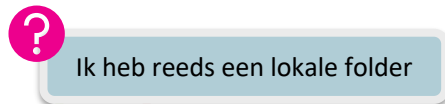
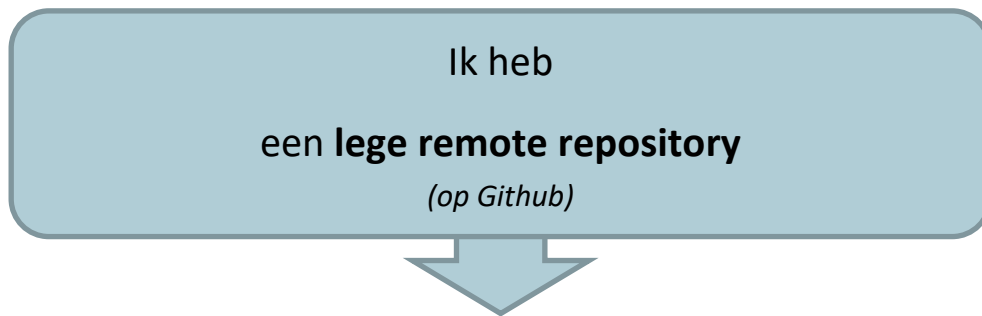
In dit deel trachten we de stappen die je dient te ondernemen tijdens het werken met git via een aantal basisvragen in kaart te brengen. Op deze manier kan je steeds terugvallen op de (*basis*) commando's die je in jouw specifieke situatie kan gebruiken.

### 3.2 STARTSITUATIE

Kies de passende startsituatie welke voor jou van toepassing is en volg aan de hand van de vragen de stappen die je nodig hebt om van start te gaan.



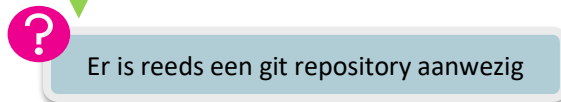




Ja

Neen

```
$ git init NaamVanFolder  
$ cd NaamVanFolder  
$ git remote add origin [url van je remote repository]
```



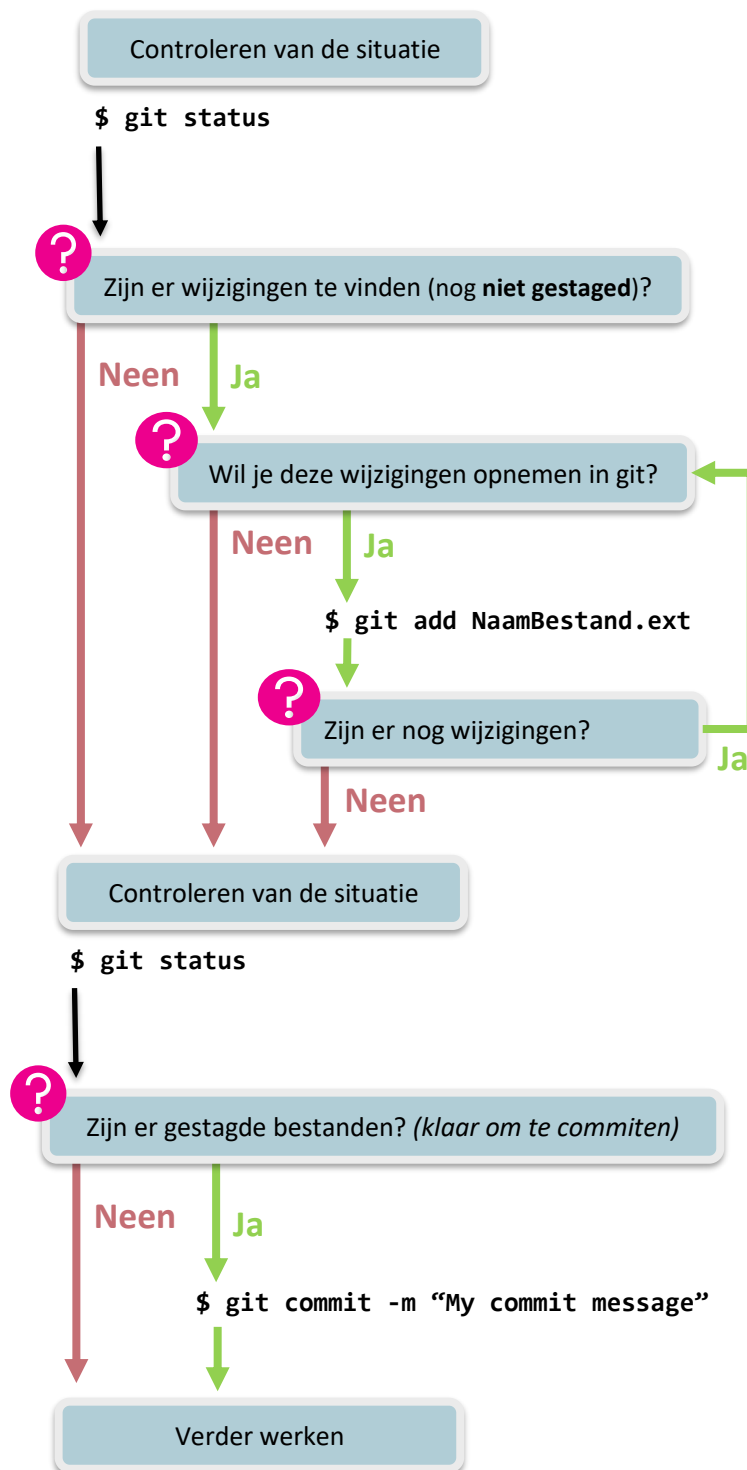
Ja

Neen

```
$ git init
```

```
$ git remote add origin [url van je remote repository]
```

### 3.3 BESTANDEN TOEVOEGEN AAN JE GIT REPOSITORY (LOKAAL)



### 3.4 JE REMOTE REPOSITORY UPDATEN MET JE HUIDIGE LOKALE REPOSITORY

---

Je gemaakte lokale commits richting je remote sturen

```
$ git push [alias van je remote] [naam van je branch]
```

Voorbeeld:

```
$ git push origin master
```

### 3.5 JE LOKALE REPOSITORY UPDATEN MET JE REMOTE REPOSITORY

---

Je lokale repository updaten met de situatie op je remote

```
$ git pull [alias van je remote] [naam van je branch]
```

Voorbeeld:

```
$ git pull origin master
```

