

Introductie in Git en GitHub.

Auteur: Carine Derkoningen

Ondersteuning: Kris Hermans

Nagelezen door: Carina Medats

Wat is Git?

Git is een vorm van versiebeheer. “Versiebeheer is het systeem waarbij veranderingen in een bestand of groep van bestanden over de tijd worden bijgehouden, zodat je later specifieke versies kan opvragen. ... in principe kan elk soort bestand op een computer aan versiebeheer worden onderworpen.” [1]

Je hebt verschillende soorten van versiebeheer, wij beperken ons tot Git, omdat die net iets meer doet, dan de meeste anderen. Git is ontworpen door Linus Torvalds. Bij Git ga je niet enkel de laatste versie van de bestanden bijhouden, maar hou je ook bij wie, wat, wanneer veranderd heeft. Dit heeft als voordeel, dat je altijd naar een vroegere versie kan teruggaan of de juiste persoon vragen stellen bij een verandering.

Een versie is dus de toestand van je project op een bepaald moment. Bij Git gebruik je commit om aan te geven op welk moment je die toestand wil bijhouden. Door je commit wordt er een verwijzing ('pointer') gemaakt (metadata wordt bijgehouden) naar een bepaalde toestand.

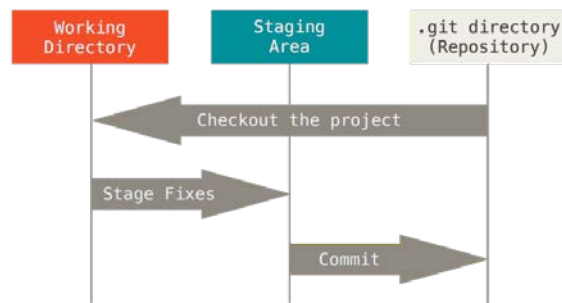
Je kan Git lokaal voor jezelf gebruiken voor elk project dat je maakt. Maar je kan Git ook op een server gebruiken, alleen of voor een groepsproject. In dat laatste geval heb je controle over de veranderingen in een project, ook al werk je er met meerdere mensen aan. Hier komen we later op terug.

Hier volgt een korte omschrijving hoe Git werkt, maar later in deze cursus wordt met voorbeelden dit nog eens duidelijk gemaakt.

Om te starten ga je een map op je computer aanduiden om het Git project in onder te brengen. Hierdoor krijg je 3 ruimtes in de map (dit zie je niet). De eerste ruimte is de werkruimte. Alles wat je doet op je computer in dit project, zoals bestanden toevoegen, wijzigen enz. gebeurt in die werkruimte (= working area). Git ziet dit, maar Git doet hier nog niets mee.

Op een gegeven moment beslis je dat bepaalde bestanden klaar zijn om opgenomen te worden voor versiebeheer, je gaat ze nu klaar zetten. Nu komen deze bestanden in de tweede ruimte terecht, een soort wachtruimte(= staging area), een index. Alleen de dingen die in deze ruimte zitten kan je uiteindelijk aan de derde ruimte toevoegen, de opslagruimte, door een commit uit te voeren. Deze opslagruimte wordt bij Git een 'repository' genoemd. [2]

Visueel ziet dat er zo uit:



Die commits(+metadata) worden bijgehouden in de verborgen map .git. Hierdoor kan je altijd de geschiedenis van je project bekijken, want elke commit krijgt een eigen id.

Op de manier hier beschreven kan je dus in elk project met versiebeheer; Git, werken. Dit is zeker aan te raden, ook al werk je alleen aan je code. Je maakt een stuk, test dit en dat werkt perfect. Je werkt er verder aan en plots crasht alles, dan is het toch interessant als je dat eerste goede stuk ergens terug kan oproepen.

Maar zoals gezegd, kan je ook met meerdere personen aan 1 project werken. Dan ga je je project ergens op een server zetten, zodat iedereen aan de code kan. Interessant is het als je een server hebt, die ook met Git werkt. Enkele voorbeelden hiervan zijn Bitbucket, Visual Studio Online en GitHub. Bij Bitbucket kan je projecten privé aanmaken en is gratis als je met maximum 5 mensen aan hetzelfde project werkt.

GitHub is gratis als je het gebruikt voor publieke en open source projecten, dus iedereen kan je code zien. Als je privé wil werken, hangt er een prijskaartje aan. Voor gebruik tijdens de lessen zullen je docenten private repositories aanmaken voor groepswork.

In deze cursus gaan we ons beperken tot GitHub. De repositories die we hier aanmaken, zijn dus publiek.

Site waar je git commando's kan testen: <https://try.github.io/levels/1/challenges/1>

Wat is GitHub?

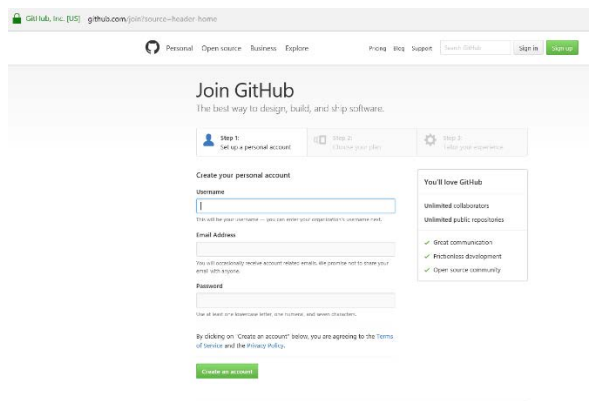
<http://github.com>

GitHub geeft je de mogelijkheid om code in de Cloud te zetten en er als team samen aan te werken. GitHub is gebouwd op het versiebeheer systeem van Git. Dit is dus een uitbreiding van je versiebeheer lokaal. Maar GitHub is veel meer dan alleen Git; er zit een hele workflow achter die je kan gebruiken.

Één van de dingen die wel handig zijn in gebruik zijn de projects, daar kan je een volledige planning i.v.m. je project in onder brengen. Deze kan je dan omzetten in issues, waar je dan de communicatie per onderdeel verder kan zetten.

GitHub in detail uitleggen, zou ons te ver leiden en hiervoor is geen ruimte in deze cursus. Wel gaan we in de loop van de cursus GitHub gebruiken en op die manier sommige dingen wel in detail bekijken.

Als eerste maak je op GitHub een account aan, hiervoor heb je een gebruikersnaam, een e-mailadres en een wachtwoord nodig. Kies een gebruikersnaam, die het voor anderen (met name je docenten) duidelijk maakt over wie het gaat. Na het aanmaken van je account krijg je nog een vragenlijst, maar die mag je skippen of invullen.



Je kan nu de gids doorlopen om kennis te maken met GitHub:

<https://guides.github.com/activities/hello-world/>

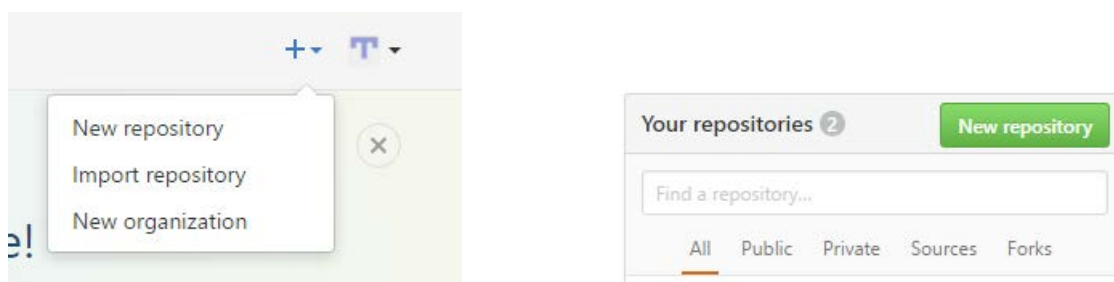
Tot zover voorlopig onze activiteit op GitHub. Nu ga je de computer voorbereiden voor het werk met Git.

Download van <https://git-for-windows.github.io/> de .exe en voer deze daarna uit, laat alles standaard, dit wil zeggen wat aangevinkt is zo laten en niets extra aanvinken. Zo installeren we Git voor Windows. Dit is gratis.

Als je dit installeert kan je git uitvoeren met commando's. Hier heb je 3 mogelijkheden om te werken: het kan in PowerShell ISE, in de gewone CMD (opdrachtterminal) of in de Bash terminal, die zich ook installeert op je computer

Wij gaan zelf een repository creëren in deze cursus, zodat je weet wat er juist allemaal gebeurt. Standaard kom je in het gebruikersaccount terecht als je een terminal opstart.

Als eerste gaan we een repository maken op GitHub. Hiervoor moet je je inloggen op GitHub.com en je kiest dat je een nieuwe repository wil aanmaken. Dit kan met het plusje naast je account afbeelding of met de knop "New repository" in het kader met het overzicht van jouw projecten:



Je krijgt nu een invulveld. We geven de repository de naam 'Cursus Git bij Windows'. Je geeft een korte uitleg bij de beschrijving. ReadMe.md aanvinken.

In dit project, ga je geen .gitignore laten genereren, omdat je hier niet met code bestanden gaat werken. Weet dat je in een repository met code wel altijd een .gitignore bestand opneemt. Later gaan we dieper in op dit soort bestand en waarom het goed is om altijd een .gitignore bestand in je repository te zetten. Je kan dit dan later nog toevoegen.


Nu klik je op de knop 'Create repository' en je remote repo is aangemaakt.

Create a new repository

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

Owner

Repository name

 cd-pxlvm ▾

 /


Cursus Git bij Data analyse ✓

Great repository names


Your new repository will be created as Cursus-Git-bij-Data-analyse **ork**.

Description (optional)

Cursus over het gebruik van Git via de commandline

☒  **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.

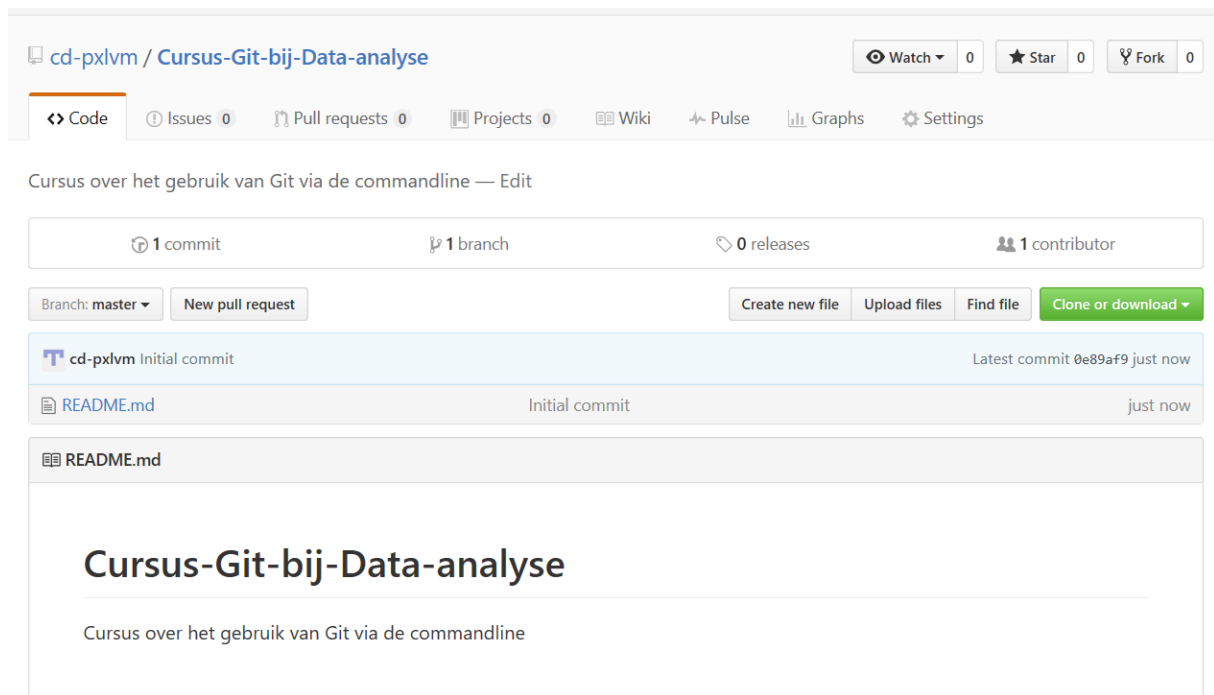
Add .gitignore: **None** ▾

 |

Add a license: **None** ▾ 

Create repository

Resultaat:



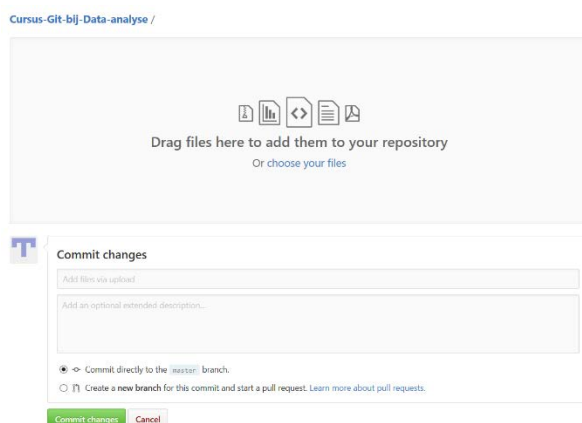
Gebruik Git

Dit onderdeel van de cursus is opgemaakt met Markdown van Github. Je zal tijdens het gebruiken van Github zelf ook op die manier tekstdocumenten maken. Hoe je zelf zo een bestand creëert, zien we later in de cursus.

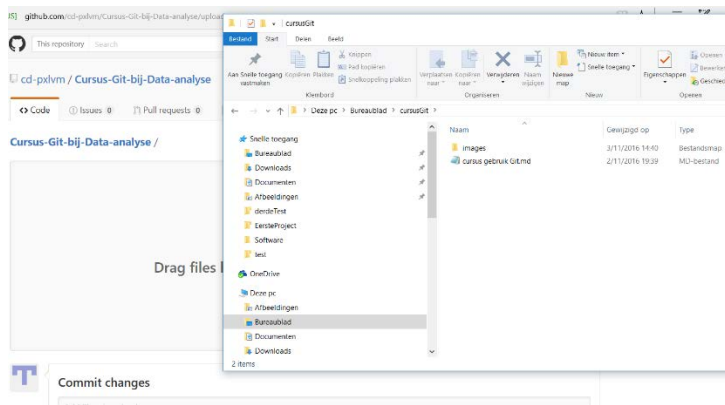
De cursus bestaat uit 'cursus gebruik Git.md' en de map images. Deze zitten in cursusGit.zip. Pak de zip uit en voeg ze toe aan het project, dat je net gemaakt hebt.

Dit kan vrij simpel:

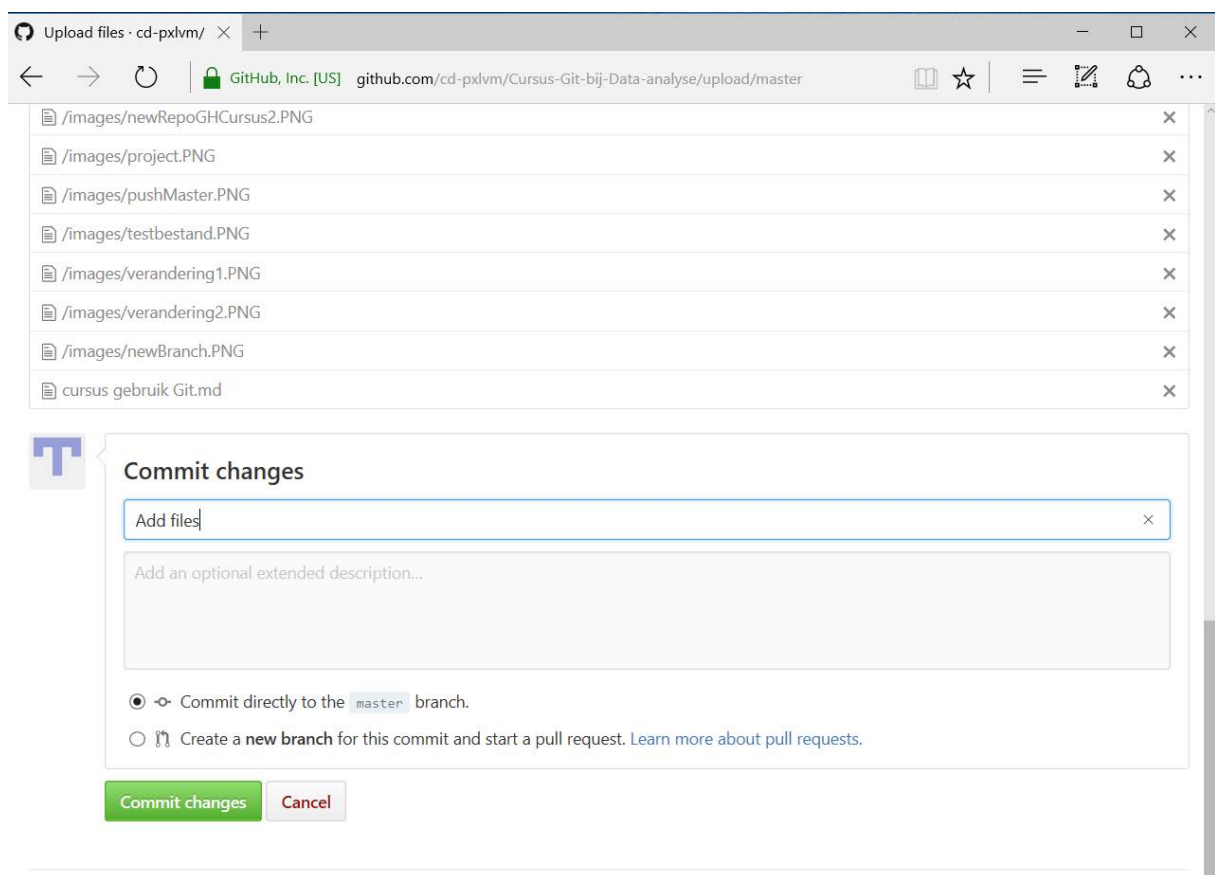
Je klikt op 'Upload Files'. Nu opent zich een subsite:



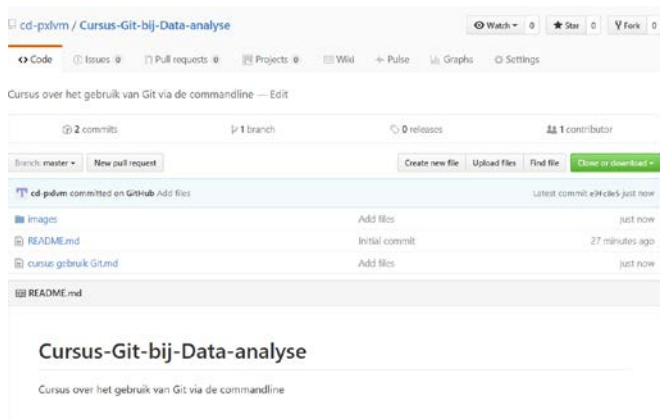
Open nu Windows verkenner in de map, die je net uitgepakt hebt. Je sleept nu het .md bestand en de images map naar de plaats waar staat; Drag files



Nu worden alle bestanden naar GitHub geüploaded. Hierna vul je een commit boodschap in en klikt op 'Commit changes'.



Nu staan alle bestanden in je repository op GitHub.



Als je al gekeken hebt naar het bestand 'cursus gebruik Git.md' in kladblok, dan zag je gewoon plain tekst met rare tekens in. Als je nu in GitHub 'cursus gebruik Git.md' opent, is dit een tekst met opmaak vergelijkbaar met een Word document.

Nu ga je verder in die cursus om je te verdiepen in het gebruik van Git en Github.

OPDRACHT:

Vervolledig de cursus 'cursus gebruik Git.md' door screenshots toe te voegen. Deze voeg je toe in de images map. In de .md file ga je op de juiste plaats in de cursus een verwijzing aanbrengen naar de screenshots. Dit doe je op volgende manier:

![voorbeeld commando](images/commando.PNG)

Tussen vierkante haken de tekst die bij de afbeelding hoort en tussen ronde haken, het relatieve pad naar de juiste afbeelding.

Bibliografie

- [1] B. S. Scott Chacon, „1.1 Aan de slag - Het wat en waarom van versiebeheer,” [Online]. Available: <https://git-scm.com/book/nl/v1/Aan-de-slag-Het-wat-en-waarom-van-versiebeheer>. [Geopend 12 oktober 2016].
- [2] W. d. Jong, 04 2012. [Online]. Available: <http://wouterj.nl/2012/04/git-introductie/>. [Geopend 12 10 2016].