

# Introduction to git

An initiative for 1BAinf

Stijn Rosaer - [stijn.rosaer@student.uantwerpen.be](mailto:stijn.rosaer@student.uantwerpen.be)  
Andrei Bondarenko - [andrei.bondarenko@student.uantwerpen.be](mailto:andrei.bondarenko@student.uantwerpen.be)  
Igor Schittekat - [igor.schittekat@student.uantwerpen.be](mailto:igor.schittekat@student.uantwerpen.be)  
Joey De Pauw - [joey.de\\_pauw@student.uantwerpen.be](mailto:joey.de_pauw@student.uantwerpen.be)  
Senne Rosaer - [senne.rosaer@student.uantwerpen.be](mailto:senne.rosaer@student.uantwerpen.be)  
Toon Meynen - [toon.meynen@student.uantwerpen.be](mailto:toon.meynen@student.uantwerpen.be)

November 13, 2018

# Contents

<b>1</b>	<b>Intro</b>	<b>1</b>
<b>2</b>	<b>How to use git</b>	<b>2</b>
2.1	Starting with Github . . . . .	2
2.2	Github in terminal . . . . .	2
2.3	Github in PyCharm/CLion . . . . .	3
2.4	Github rules . . . . .	3
2.5	.gitignore file . . . . .	3
<b>3</b>	<b>Exercise</b>	<b>4</b>
3.1	Maak een repository aan en start een project . . . . .	4
3.2	gitignore . . . . .	4
3.3	project . . . . .	4

## 1 Intro

Github is een versiebeheersysteem waar software geplaatst kan worden. Dit houdt in dat men code hier kan plaatsen, updaten, downloaden en terug gaan naar een voorgaande versie. Ook is dit gemakkelijk wanneer men met verschillende personen tezamen werkt aan dezelfde code. We willen zo een overzicht behouden en op een wel bepaald moment elks aan code kunnen werken, zonder dat de code van de ander op dat moment beïnvloed wordt.

## 2 How to use git

### 2.1 Starting with Github

1. Om github te gebruiken moeten we eerst een account aanmaken. Ga hiervoor naar [github.com](https://github.com) en creëer een account. Gebruik zeker je UA-mailadres (voornaam.achternaam@student.uantwerpen.be), je kan namelijk als student gratis private repositories aanmaken.<sup>1</sup>
2. Voor het aanvragen van je github student developer pack, moet je naar [education.github.com/pack](https://education.github.com/pack) gaan en deze hier aanvragen. Dit kan wat tijd vragen, maar dit hebben we niet nodig om verder te kunnen.

### 2.2 Github in terminal

- Een bestaande github repository downloaden:  
`git clone [repository url]`
- Maak van de huidige map een git repository:  
`git init [project name]`.
- Controleer de status van je git repository:  
`git status`.
- Voeg een of meerdere files toe die je op git wilt bewaren:  
`git add [filenames]`
- Voeg de items toe aan je locale versiebeheer:  
`git commit -m [bericht met een korte omschrijving van wat je gedaan hebt tussen ""]`
- Om dit online te zetten op github en te delen met andere is eerst een repository nodig op github

1. maak een repository aan:



stijnrosaer ▾

Repositories New

Find a repository...

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

Owner: stijnrosaer ▾ / Repository name:

Great repository names are short and memorable. Need inspiration? How about *crispy-invention*.

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

---

<sup>1</sup>Het is ook mogelijk om later je studenten account te linken en alsnog de voordelen te krijgen.

2. Voeg je online repository toe aan github:

*git remote add origin [url van je github repository].*

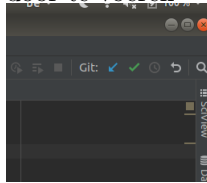
- Voeg hetgeen je op je pc hebt staan toe op github:  
*git push*
- Download de wijzigingen op github:  
*git pull*

Een samenvatting en extra features kan je vinden op:

<https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf>

## 2.3 Github in PyCharm/CLion

1. in terminal:  
*git clone [repository url]*
2. Maak nu een nieuw project met de map die je juist gecloned hebt aan en klik op JA, dat je de bestanden wilt vervangen
3. Gebruik de blauwe pijl om wijzigingen op te halen en de groene om er zelf door te voeren



4. Voor het toevoegen moet je nog een message geven en zeker niet vergeten om commit en push te kiezen ipv commit
5. gebruik de toetsencombinatie CTRL + SHIFT + TAB om te terminal te openen in PyCharm of CLion

## 2.4 Github rules

1. push pas wanneer je code compileert, comitten kan je al eerder doen aangezien dit enkel op je computer komt.
2. gebruik veel git status om te kijken wat er te gebeuren staat (voor git push, git pull, git commit, git add).
3. pull eerst steeds vooraleer je pusht.
4. breek alles op in kleine stukjes (telkens er iets werkt commit en push je dit).
5. doe NOOIT "git add ." vooraleer je git status gedaan hebt!

## 2.5 .gitignore file

De .gitignore file zorgt ervoor dat niet alle files meer toegevoegd worden. Hier kan je zelf nog extra extenties of files toevoegen. Git zal op deze manier bepaalde files niet mee comitten.



### 3 Exercise

<https://classroom.github.com/g/8kE8ZX7m>

#### 3.1 Maak een repository aan en start een project

1. (Persoon 1) ga naar de link en maak een groep aan
2. (Persoon 2) ga naar de link en join het team van je groepspartner
3. (Persoon 1&2) kopieer de url
4. (Persoon 1&2) open een terminal en typ "git clone" en doe CTRL + SHIFT + V. De repository wordt nu gedownload op je pc.

#### 3.2 gitignore

*(Persoon 1&2) Maak een python project met de gegeven bestanden.*

1. 1 iemand maakt een .gitignore file. Kijk via git status wat je niet wil toevoegen en zet dit in de file.
2. deze persoon doet een commit met een gepaste beschrijving en een push
3. de andere persoon doet een pull. Kijk wat er is veranderd.

#### 3.3 project

1. Open beide je project.
2. En persoon vervolledigd de fibonacci functie, de andere maakt factorial.
3. Zorg dat je beide de twee functies hebt door gebruik te maken van push, pull en commit
4. Pas beide in dezelfde functie hetzelfde aan, en push beide.
5. Je krijgt nu een merge conflict omdat je beide hetzelfde aangepast hebt. Los dit op!