

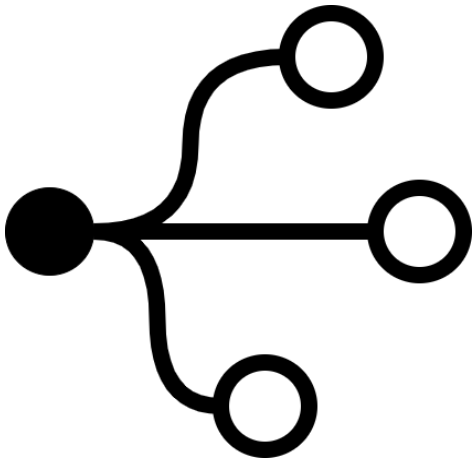
Git cracking

The basics is where you start

Contents

1.	GIT version control	3
2.	Concepts	4
3.	Commands	5
3.1.	Basic	5
3.2.	Bij problemen	5
3.3.	Remote	6
4.	Enkele bronnen:	7
5.	Contact	8

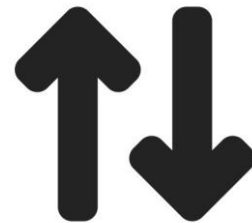
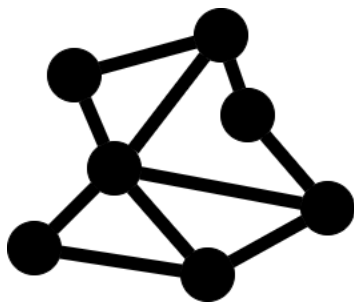
1. Version control



Is a class of **systems** responsible for managing **changes** to computer programs, documents, large web sites, or other collections of **information**

Van <https://en.wikipedia.org/wiki/Version_control>

Git werd in een paar weken ontwikkeld door Linux Torvalds vanuit een nood voor een degelijk version control system voor het open source project "Linux Kernel".
Het moest voldoen aan volgende 3 voorwaarden:



Distributed: Iedereen moet kunnen werken met de volledige geschiedenis van het totale project
Snelheid: Patches moeten toepasbaar zijn onder de 3 seconden
Veiligheid: Bestanden ingegeven in het systeem moeten ten allen tijde gevalideerd kunnen worden op correctheid

Git is in de afgelopen jaren zo populair geworden dat je als ontwikkelaar er zeker mee in contact zal komen. Volgens de Developer Survey van de bekende website [www.stackoverflow.com](https://survey.stackoverflow.co/2022#technology-version-control) gebruikt bijna 94% van alle bevraagde ontwikkelaars deze tool.

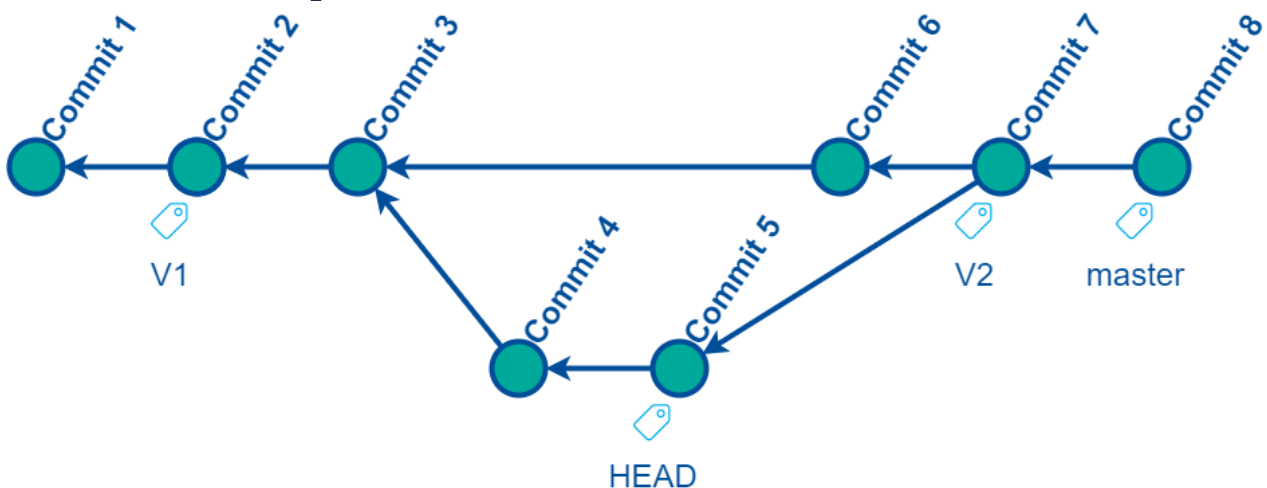
<https://survey.stackoverflow.co/2022#technology-version-control>

Maar toch is het geen geliefde tool voor iedereen.

Does anyone else hate git as much as I do?

You are doing it wrong!!!

2. Concepts



Repository: De verzameling van alle files die behoren tot het project + alle bestanden die nodig zijn voor het opbouwen van de geschiedenis (Magazijn)

Staging: De locatie waar aanpassingen verzameld worden voordat ze in de repository worden opgeslagen. In staging zijn nog steeds aanpassingen mogelijk (Open doos)

Commit: Een set van aanpassingen die opgeslagen worden in de repository en die vanaf dan enkel nog leesbaar zijn. Aanpassingen zijn niet meer mogelijk (Gesloten doos)

Tag: Een label dat aan een commit kan gehangen worden.

Er bestaan enkele tag types:

Tag: Een gewone tag is een leesbare aanduiding van een commit. (zoals een sticker met een naam op de doos)

HEAD: Deze tag zal altijd aan de commit hangen waarvan de status in de file directory word weergegeven (Post-it, GIT verhangt deze post-it)

Branch (master, main, ...): Deze tag's duiden de locatie van een branch aan. Indien gekoppeld met HEAD zal de branch tag mee verhuizen naar de volgende commit. (Attached HEAD)

Directed Acyclic Graph:

Een commit verwijst steeds naar zijn voorganger. Dit is geen aanduiding van tijd, maar een opvolg systeem zodat alle commit's steeds in de juiste volgorde blijven staan en kunnen uitgelezen worden. Denk hierbij aan de klanten in een winkel. Als je de winkel binnenstapt, heb je voldoende te weten wie de laatste was die de winkel binnen kwam. Als iedereen dit doet, zal de rij klanten steeds correct afgewerkt worden, zonder dat iemand de volledig rij in het oog moet houden.

Voor deze verwijzing en de identificatie van een commit word een nummering gebruik, SHA1, een berekening op basis van de inhoud van de commit die zal resulteren in een hexadecimale waarde van 40 karakters lang.

3. Commands

Download & install:

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

3.1. Basic

Naast het feit dat er veel programma's bestaan die GIT aanbieden via een klikbare user-interface, is GIT best goed te gebruiken via zijn CLI (Command Line Interface). Na installatie kan je gebruik maken van je favoriete command prompt om de nodige acties uit te voeren.

Windows: Klik op "start" -> type cmd -> klik op Opdrachtprompt

Navigeer naar de gewenste folder voor je project, daar kan je dan volgende commando's gebruiken:

Command	Uitleg
git init	Initialiseer een nieuwe repository (maak een nieuw magazijn)
git status	Beschrijf wat er aan de hand is in de repository
git add bestand git add .	Voeg een of meerdere bestanden toe aan staging
git commit -m "omschrijving"	Alle files in staging worden verpakt in een commit en in de repository gestoken met de omschrijving als documentatie bij de commit
git branch branchname	Maak een branch tag aan op de huidige commit met naam branchname
git checkout 32ba56c git checkout tagname git checkout branchname	Verplaats je HEAD naar de locatie aangegeven. (commit, getagde commit, commit met branchname tag) Het laatste commando zal er ook voor zorgen dat je HEAD attached is aan die branch, de eerste twee doen dit niet!
git checkout -b branchname	Maak een nieuwe branch aan met naam branchname en doe hierop direct een checkout: Attached je HEAD tag direct aan de nieuwe branch.
git merge branchname	Voeg alle verandering die in de "branchname" branch gemaakt zijn toe aan de branch waaraan je huidige HEAD gekoppeld is.
git log git log --graph --oneline	Toon alle commits and tags in de huidige repository. Teken hierbij het pad en toon elke commit als enkele lijn

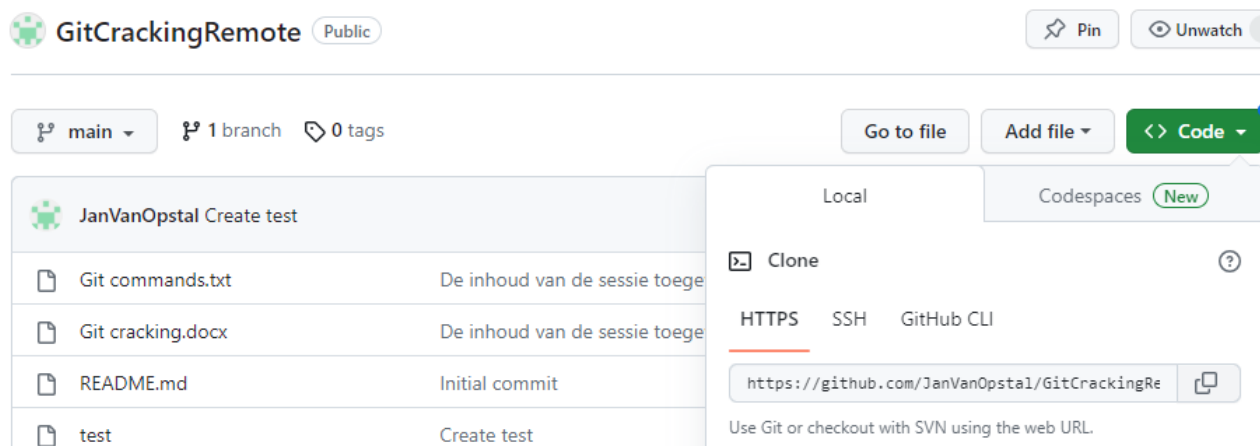
3.2. Bij problemen

Lees de boodschappen die Git geeft aandachtig en probeer dit op de theorie toe te passen.

Geraak je niet uit een weergave, probeer eens het volgend: type :q en druk op enter

3.3. Remote

Op github <https://github.com> kan je een gratis account maken en zelf een nieuwe repository maken. Laat Github een readme file toevoegen, zo is je repository ineens klaar en heeft het al z'n eerste commit. Onder de knop "Code" vind je een URL terug die je kan gebruiken om je repository lokaal te clonen.



Navigeer op je computer in de opdrachtprompt naar de gewenste folder waar je je repository wil hebben. Gebruik volgend commando om je remote repository te clonen:

```
git clone urlGecopieerdOpGithub
```

```
bv: git clone https://github.com/JanVanOpstal/GitCrackingRemote
```

Er zal een nieuwe folder aangemaakt worden met een copy. Navigeer in de opdrachtprompt naar die nieuwe folder.

Hierin kan je dan volgende commando's gebruiken om je repository up-to-date te houden met je online clone.

Command	Uitleg
git pull	Haal alle commit's die remote toegevoegd zijn aan de huidige branch op en pas ze lokaal toe
git push	Duw alle lokaal gemaakte commit's naar de branch op de remote clone
git fetch	Haal alle informatie op rond bestaande branches in de remote repository
Git checkout -b branchname --track origin/branchname	Maak een nieuwe lokale branch aan, attach je HEAD aan de nieuwe branch, en maak ineens ook een gekoppelde branch voor het remote opvolgen aan

4. Enkele bronnen:

GIT

<https://git-scm.com>

GIT Reference

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

GitHub

<https://github.com>

Branching models

<https://nvie.com/posts/a-successful-git-branching-model>

Stack overflow Developer Survey

<https://survey.stackoverflow.co/2022#technology-version-control>

GIT Internals

<https://git-scm.com/book/en/v2/Git-Internals-Git-Objects>

GIT repository

<https://github.com/git/git>

Linux repository

<https://github.com/torvalds/linux>

Branching models

<https://nvie.com/posts/a-successful-git-branching-model>

Matt Rickard History of VC

<https://matt-rickard.com/history-of-version-control-part-1>

Linus Torvalds at google

<https://www.youtube.com/watch?v=4XpnKHJAok8>



5. Contact

Jan Van Opstal

jan.vanopstal@inetum-realdolmen.world

<https://github.com/JanVanOpstal/GitCrackingRemote>

Any **expert** started out as a **beginner**.

Become an **intern**! Check out

[**jobs.realdolmen.com/internships**](https://jobs.realdolmen.com/internships)

Mail your CV to

[**internship@inetum-realdolmen.world**](mailto:internship@inetum-realdolmen.world)

Choose your future by **creating** it.

Check out

[**jobs.inetum-realdolmen.world**](https://jobs.inetum-realdolmen.world)

... or mail your CV to

[**recruitment@inetum-realdolmen.world**](mailto:recruitment@inetum-realdolmen.world)