# Crossmediaproductie 5

## Les 1 git commands

Tutorial: <https://try.github.io/levels/1/challenges/1>

* **Git init** = git starten
* **Git status** = de stand van zaken geven, waar is git op dit moment (jij werkt op branch master)
* **Git add** = dit bestand wil ik meenemen git
* **Git commit –m “beschrijving van versie/titel”** = Versie dat je hebt doorgeven aan git, de stand nu = commit, -m = naam geven, message
* **Git add ‘\*.txt’** = git add maar de \* (hij noemt dit wildcard) betekend dat je bedoeld de folder waarmee je bezig bent meenemen, op zoek gaan naar alles waar .txt in zit. Op die manier kan je makkelijk alleen bepaalde soort bestanden toevoegen.
* Altijd als je iets belangrijk hebt gedaan moet je save as doen, -> **git commit –m ‘beschrijvin/titel’**
* **Git log** = log open doen, hier kan je zien wat je allemaal al hebt gedaan
* **Git remote add origin htpps://github.com/try.git.git**
* Git = ik ga een commando in git uitvoeren, remote = een afstands-server, add = toevoegen, we geven hier het de naam origin, url = waar dit remote te vinden is
* Dit is de cloud, waar alles naar toegestuurd word als je in groep werkt
* Remote moet eindigen met .git, dit betekend dat het een git repository is
* **Git push –u origin master** = de bestanden van de pc naar github pushen, uploaden naar de git server. –u = (- is optie aanvinken) -> onthouden. Origin = de naam die je aan de server hebt gegeven, dit heet bijna altijd origin. Master = dit sturen we naar de server, alles van de master pushen we naar de origin. Dus: zet deze map op de server.
* Git pull origin master
* We zien dat er een bestand is bijgekomen (yellow\_octocat)
* Git diff HEAD
* We zien nu dat iemand octocat.txt heeft aangepast, yellow\_octocat weggenomen en een nieuwe toegevoegd. Git kan u vertellen welke tekst is (kan txt lezen, illustrator niet) aangepast bij vb. groepswerk
* Git add octofamily/octodog.txt -> We voegen octofamily toe en in die map zit octodog
* **Git diff- staged** = verschil bekijken. Dit heb je allemaal verandert sinds de vorige versie.
* **Git reset** betekend dat het gereset is, terug zoals ervoor, niemand ziet aanpassing ?
* **Git checkout – octacat.txt** = met checkout kan je bestand terug halen van server zoals het was, je eigen wijzigingen weg
* Je mag nooit programmeren op de master, dus we maken een nieuwe tak door
* **Git branch** clean up (geef nieuwe naam vb. clean up)
* Je hebt nu 2 branches, master en clean up
* We zitten nu op de clean up, deze moet nu op de server gezet worden zodat iedereen er aan kan. Git checkout clean up
* **Git rm ‘\*.txt’** = rm = remove, alle bestanden die eindigen om txt worden verwijdert.
* Nu moeten we opslaan -> git commit –m ……………
* **Git checkout** master = we gaan terug naar master
* **git merge clean\_up** = Met merge de takken laten samenkomen, merge en clean up samenvoegen
* **git branch –d** clean\_up -> is niet meer echt nuttig, is samengevoegd. –d = de optie delete, degene met de naam clean up verwijderen. We hebben dit niet meer nodig, we hebben de master.
* Als je klaar bent of pauze pakt, opslaan en op server zetten **=> git push**

Cheatsheet: <https://www.atlassian.com/git/tutorials/atlassian-git-cheatsheet>

Attlasian kan je info vinden over het vak. Hier staat ook vb. de tool sourcetree

Sourcetree = je kan alle mappen/projecten in 1 venster terug vinden

https://www.sourcetreeapp.com/