

Git

- Git
 - [Basiscommando's](#)
 - [Branches](#)
 - [Remote repositories](#)
 - [Wijzigingen terugdraaien](#)
 - [Inspectie & vergelijken](#)
 - [Tags & releases](#)
 - [Geavanceerd](#)
 - Branch
 - [toon, maak, switch, merge en delete branch](#)
 - [create a repo](#)
 - [create README file](#)
 - [git config](#)

◊ Basiscommando's

- `git init` → Maak een nieuwe lokale Git-repository in de huidige map.
- `git clone <url>` → Kopieer een bestaande repository (bijv. van GitHub) naar je computer.
- `git status` → Laat zien welke bestanden gewijzigd zijn, en of ze zijn toegevoegd voor commit.
- `git add <bestand>` → Voeg een bestand toe aan de staging area.
- `git add .` → Voeg alle gewijzigde bestanden toe aan de staging area.
- `git commit -m "bericht"` → Maak een commit met een beschrijving.
- `git log` → Toon de commitgeschiedenis.
- `git log --oneline` → Verkorte weergave van de commitgeschiedenis.

◊ Branches

- `git branch` → Toon alle lokale branches.
- `git branch <naam>` → Maak een nieuwe branch.
- `git checkout <branch>` → Wissel naar een branch.
- `git switch <branch>` → Nieuwe manier om naar een branch te wisselen.
- `git checkout -b <branch>` → Maak en wissel naar een nieuwe branch in één stap.
- `git merge <branch>` → Voeg een andere branch samen met de huidige.
- `git branch -d <branch>` → Verwijder een branch.

◊ Remote repositories

- `git remote -v` → Toon gekoppelde remote repositories (bv. origin).
- `git remote add origin <url>` → Koppel een remote repository.
- `git push origin <branch>` → Push lokale commits naar de remote.
- `git push -u origin <branch>` → Push en stel standaard remote branch in.
- `git pull` → Haal nieuwe commits op van de remote en merge ze.

- `git fetch` → Haal updates van de remote op, zonder te mergen.

◇ Wijzigingen terugdraaien

- `git reset <bestand>` → Haal bestand uit de staging area.
- `git reset --hard <commit>` → Zet repository volledig terug naar een bepaalde commit (**waarschuwing: werk gaat verloren!**).
- `git checkout -- <bestand>` → Zet een bestand terug naar de laatste commit.
- `git revert <commit>` → Maak een nieuwe commit die een bepaalde commit ongedaan maakt.

◇ Inspectie & vergelijken

- `git diff` → Toon ongecommitte wijzigingen.
- `git diff --staged` → Vergelijk staged changes met de laatste commit.
- `git show <commit>` → Toon details van een specifieke commit.
- `git blame <bestand>` → Toon wie welke regel in een bestand heeft gewijzigd.

◇ Tags & releases

- `git tag` → Toon tags.
- `git tag <naam>` → Maak een nieuwe tag.
- `git tag -a <naam> -m "bericht"` → Maak een *annotated* tag.
- `git push origin --tags` → Push alle tags naar remote.

◇ Geavanceerd

- `git stash` → Sla tijdelijke wijzigingen op zonder te committen.
- `git stash pop` → Haal de laatst opgeslagen wijzigingen terug.
- `git cherry-pick <commit>` → Kopieer één specifieke commit naar de huidige branch.
- `git rebase <branch>` → Pas commits opnieuw toe bovenop een andere branch.
- `git reflog` → Laat zien welke commits en HEAD-bewegingen je lokaal hebt gehad (handig om dingen terug te vinden).

Branch

toon, maak, switch, merge en delete branch

```
echo "Local branches:"
git branch
echo "Remote branches:"
git branch -r
git checkout -b "$branch_name"
git checkout master
git merge naam_van_branch
git branch -d naam_van_branch
```

create a repo

```
mkdir git_practice  
cd git_practice  
git init
```

create README file

```
touch README.md  
echo "This is readme file for git_practice project" > README.md
```

git config

```
git config user.name alex.schapelle  
git config user.mail alex@vaiolabs.com  
git remote add origin https://gitlab.com/url_to_your_project.git  
  
git config --global user.name Paul  
git config --global user.email "paul.cobbaut@gmail.com"  
git config --global core.editor vi
```