# Git Branches - Uitleg en Syntax

In Git zijn branches vertakkingen van je project. Ze laten je toe om nieuwe features, bugfixes of experimenten te ontwikkelen zonder de hoofdbranch (meestal 'main' of 'master') te beïnvloeden.

## Waarom branches gebruiken?

- Je kunt veilig nieuwe functies ontwikkelen zonder de stabiele versie te breken.  
- Makkelijker samenwerken in teams.  
- Je kunt verschillende versies of experimenten parallel beheren.

## Belangrijkste commando’s

1. \*\*Een nieuwe branch maken\*\*:

git branch nieuwe-branch

2. \*\*Overschakelen naar een branch\*\*:

git checkout nieuwe-branch  
# of, moderne syntax:  
git switch nieuwe-branch

3. \*\*Een nieuwe branch maken en er direct naar overschakelen\*\*:

git checkout -b nieuwe-branch  
# of:  
git switch -c nieuwe-branch

4. \*\*Alle branches bekijken\*\*:

git branch

5. \*\*Een branch mergen in de huidige branch\*\*:

git merge naam-van-branch

6. \*\*Een branch verwijderen\*\*:

git branch -d naam-van-branch  
# Forceer verwijderen:  
git branch -D naam-van-branch

7. \*\*Remote branches updaten en tonen\*\*:

git fetch --all  
git branch -r

## Voorbeeld workflow

1. Maak een nieuwe branch voor een feature:  
 git checkout -b feature-login  
  
2. Werk aan je code en commit wijzigingen:  
 git add .  
 git commit -m "Voeg login functionaliteit toe"  
  
3. Ga terug naar main en merge de feature:  
 git checkout main  
 git merge feature-login  
  
4. Verwijder de feature-branch:  
 git branch -d feature-login