

### Naamgeving

**HEAD** huidige commit of tip van huidige branch

**staging area** of **index** veranderingen die in de volgende commit gaan komen

**working directory** of **working tree** files zoals Ze op je harde schijf staan

**commit** een versie of snapshot van je repository

### Configuratie

**git init** initialiseer een lege repository

**git clone url** kopieer een remote repository

**git remote add origin url** voeg een remote met naam "origin" toe met als adres *url*

**.gitignore** bestand met files die git niet mag tracken

### Basiscommando's

**git add file** voeg *file* toe aan de staging area

**git add -u** voeg alle tracked files toe aan de staging area

**git commit** bundel veranderingen in een nieuwe versie (commit)

**git commit -a** voeg automatisch alle tracked bestanden toe

**git commit -m "message"** geef de commit message mee op de command line

**git restore --staged file** verwijder file uit Staging area

### Branches

**git branch naam** maak een nieuwe branch *naam*

**git branch** alle branches bekijken

**git switch branch** wissel naar branch

**git switch -c branch** maak een nieuwe branch en wissel ernaar

**git reset commit** reset branch en staging area naar commit

**git reset --hard commit** zoals git reset, maar reset ook working tree

**git merge branch** merge branch in huidige branch

**git push** stuur veranderingen naar remote

**git pull** haal veranderingen op van remote

### Bestandsacties

**git mv** verplaats een bestand in working tree en voeg verplaatsing toe aan staging area

**git rm** verwijder een bestand uit working tree en voeg verwijdering toe aan staging area

**git rm -r** verwijder een map uit working tree en voeg verwijdering toe aan staging area

### Stashes beheren

**git stash push** sla alle changes in de working tree en staging area op en reset dan naar HEAD

**git stash apply** pas de veranderingen van de laatste stash toe op de huidige working tree

**git stash drop** verwijder de laatste stash

**git stash pop** hetzelfde als *git stash apply* en dan *git stash drop*

**git stash show** toon de veranderingen in de laatste stash

### Commits bekijken

**git log** toon geschiedenis van huidige commit

**git log --oneline** een regel per commit

**git log --graph** teken branches

**git log --all** toon geschiedenis van alle branches ipv enkel de huidige branch

**git show id** toon de veranderingen in de commit *id*

**git diff** toon unstaged changes

**git diff --staged** toon staged changes

**git diff id1 id2** toon de veranderingen in *id2* ten opzichte van *id1*

<https://ulyssis.org/hosting/>

**Persoonlijk** €7/j

- ✓ 1 GB
- ✓ PHP, Python, Ruby, ...
- ✓ MySQL & PostgreSQL
- ✓ Git & SVN

Bestel nu!

**Persoonlijk** €12/j

- ✓ 2 GB
- ✓ PHP, Python, Ruby, ...
- ✓ MySQL & PostgreSQL
- ✓ Git & SVN

Bestel nu!

**Persoonlijk** €20/j

- ✓ 5 GB
- ✓ PHP, Python, Ruby, ...
- ✓ MySQL & PostgreSQL
- ✓ Git & SVN

Bestel nu!

**Organisatie** €7/j

- ✓ 5 GB
- ✓ PHP, Python, Ruby, ...
- ✓ MySQL & PostgreSQL
- ✓ Git & SVN

Bestel nu!

Mogelijk gemaakt dankzij

