

# PROGRAMMEREN 4

## Versiebeheer deel 1

# WAT IS VERSIEBEHEER?

---

- Systeem dat wijzigingen aan een bestand of collectie van gegevens bewaard.
- Hierdoor kunnen wijzigingen:
  - Door meerdere developers aangebracht worden
  - Vergeleken en samengevoegd worden
  - Hersteld worden

# WAAROM VERSIEBEHEER?

---

- Eigen systeem (via bijvoorbeeld dropbox) is mogelijk
- Nadelen:
  - Moeilijk wijzigingen in teamverband aan te brengen
  - Elke versie is een ander bestand en moeilijke naamgevingen
  - Geen mogelijkheid tot bijhouden wie welke wijziging aanbracht

# WAAROM VERSIEBEHEER?

---

- Specifiek ontworpen versiebeheersysteem (Bijvoorbeeld GIT)
- Voordelen:
  - Mogelijkheid om gelijktijdig wijzigingen te maken
  - Duidelijk overzicht van wijzigingen (wie, wat, wanneer)
  - Vorige versies continu beschikbaar
  - Waarschuwingen van conflicten
  - Geen onnodige storage

# MAAR HOE?

---

Gecentraliseerde systemen

OF

Gedistribueerde systemen

# GECENTRALISEERD SYSTEEM

---

- Bestanden (en historiek) op 1 centrale server
- Developers communiceren continu online bij het maken van wijzigingen
- Enkel online beschikbaar
- Bijvoorbeeld: CVS en TFS

# GEDISTRIBUEERD SYSTEEM

---

- Bestanden op server
- Bestanden worden gedownload
- Ook offline beschikbaar
- Bijvoorbeeld: Git en Mercurial

- Open source, gedistribueerd versiebeheersysteem
- Software
  - GUI Tool (Bijvoorbeeld: Sourcetree)
  - Command Line Interface



# GITHUB

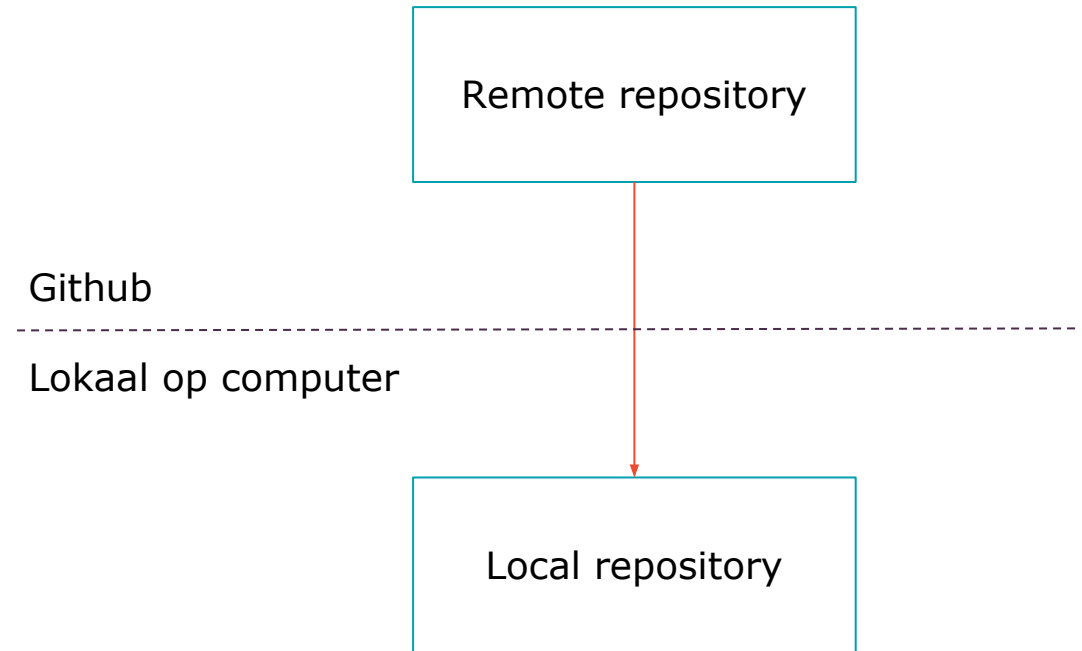
---

- Hosting service voor software
- Gebouwd rond het Git-versiebeheersysteem
- Extra features (Wiki, project- en teammanagement)

# WORKFLOW GIT - CLONE

## Clone

- Lokaal kopie maken
- Code van remote repository inladen in local repository
- Lokaal code aanpassen heeft geen rechtstreeks effect op de code van de remote repository



# WORKFLOW GIT - LOKALE WIJZIGINGEN

## Na wijzigingen

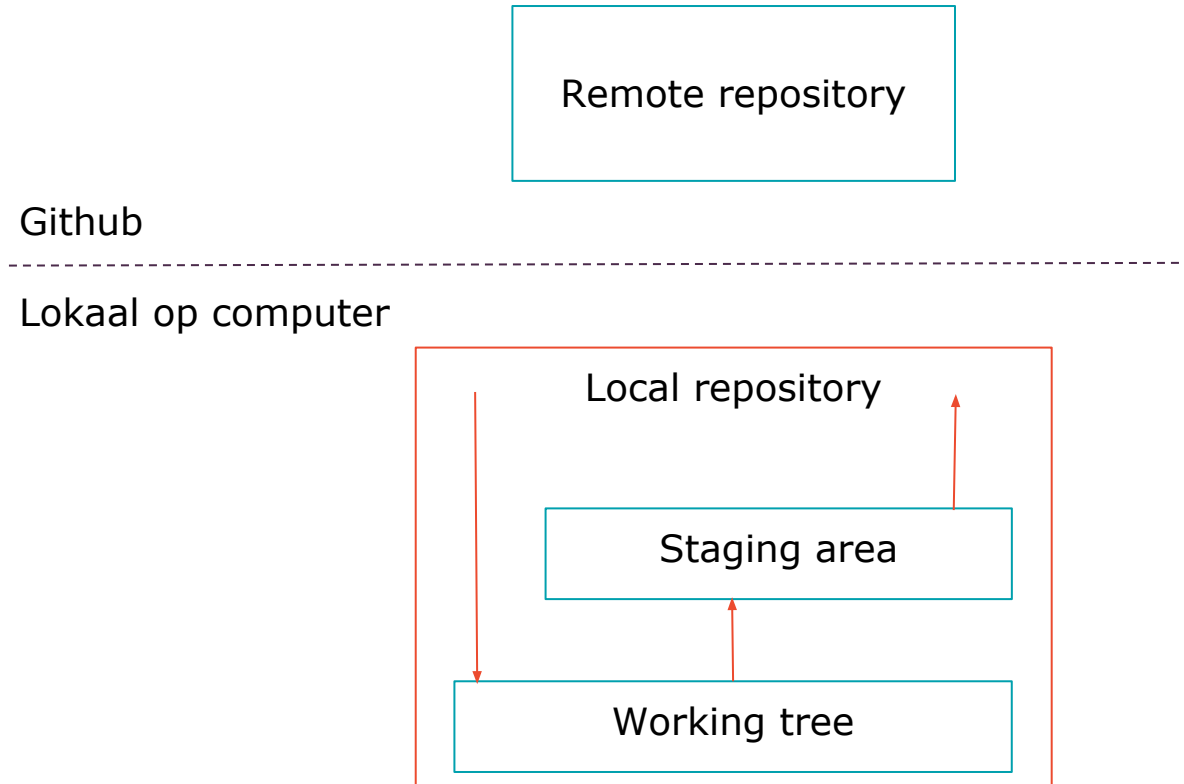
- Bestand naar working tree

## Add

- We zetten aangepaste code in de staging area

## Commit

- We zetten aangepaste code in de lokale repository



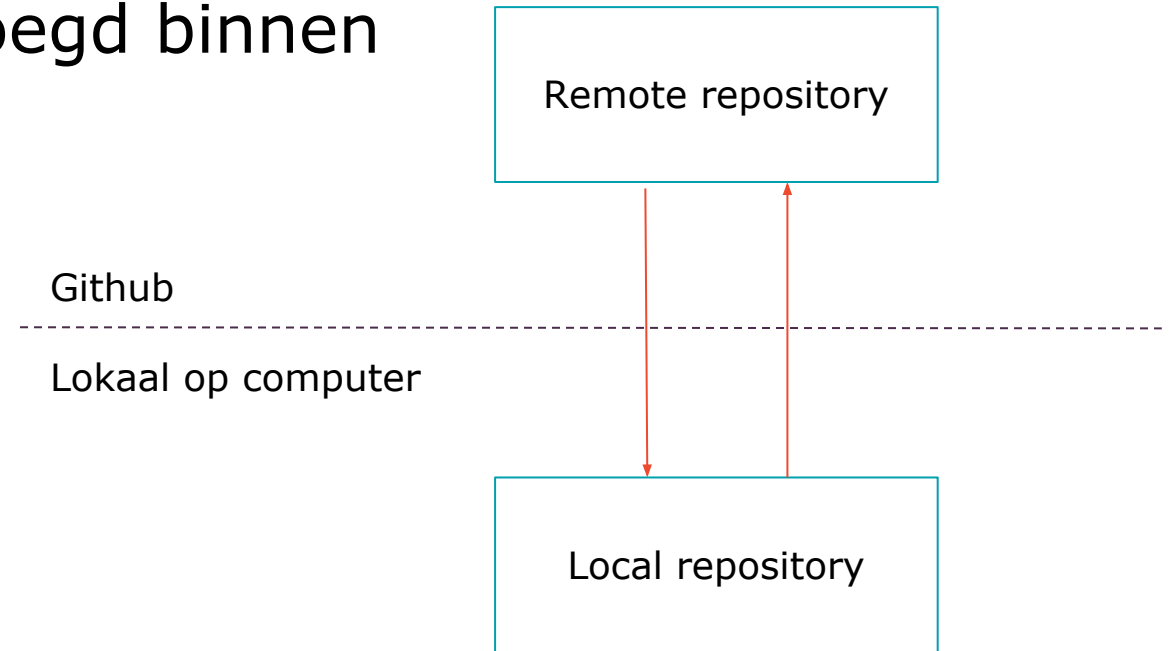
# WORKFLOW GIT - REMOTE WIJZIGINGEN

## Pull

- Aangepaste code van de remote repository wordt opgehaald en samengevoegd binnen de lokale repository (**Merging**)
- Mogelijks merge conflicten

## Push

- De aangepaste code van de lokale repository wordt



# GIT & GITHUB ACCOUNT

---

Installeer git

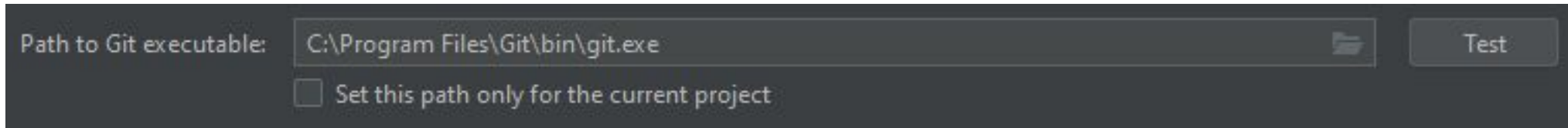
- <https://gitforwindows.org/>

Maak een github account aan

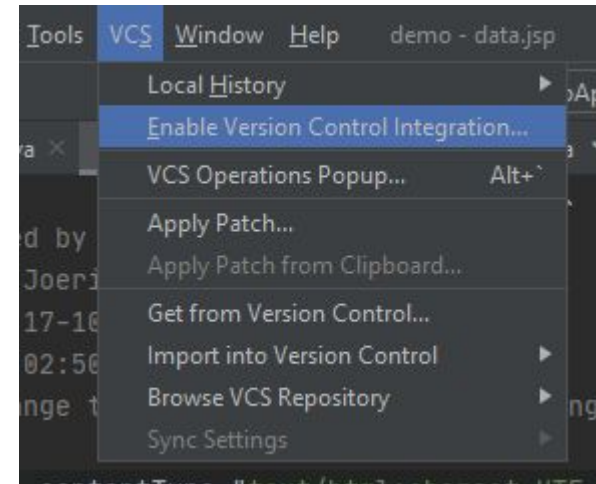
- <https://github.com/>

# GIT INTEGRATIE IN INTELLIJ

Settings > Version Control > Git



VCS > Enable version control integration



# GITHUB INTEGRATIE IN INTELLIJ

---

Settings > Version Control > Github

- Klik op het + icoontje
- Log in via Github..

Q

> Appearance & Behavior

Keymap

> Editor

Plugins

> Version Control

Background

Changelists

Commit

Confirmation

File Status Colors

Issue Navigation

Shelf

Git

GitHub

> Build, Execution, Deployment

> Languages & Frameworks

> Tools

Version Control > GitHub

For current project



Joeri Verlooy QuirkyDev  
github.com



+

-

✓

☐ Clone git repositories using ssh

Connection timeout: 5 seconds



OK

Cancel

Apply



# CONFIG INIT PROJECT

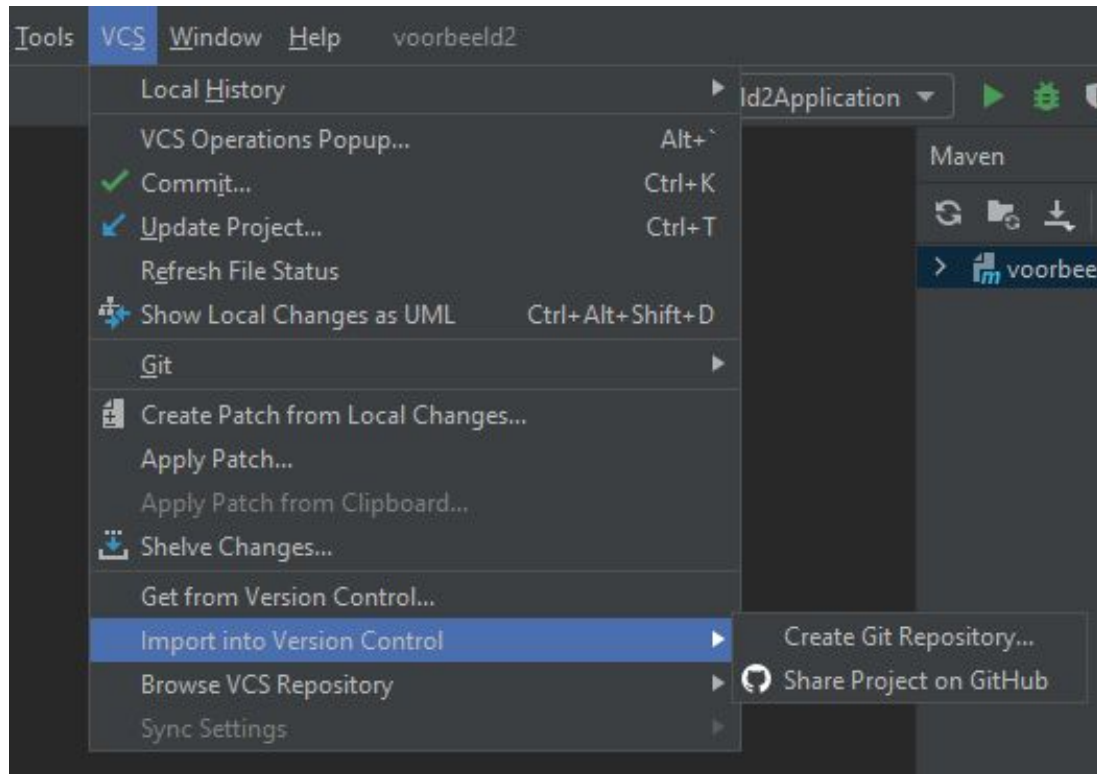
---

## In terminal

- `git init`
- `git config user.name "jouw naam"`
- `git config user.email "jouw@emailadres.com"`
- `git add *`
- `git commit -m "init project bericht"`

# PROJECT OP GITHUB SHAREN

VCS > Import into Version Control > Share project on Github



[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

master ▾

1 branch

0 tags

[Go to file](#)[Add file ▾](#)[Code ▾](#)

QuirkyDev init push

ed2afd8 23 minutes ago 1 commits



.mvn/wrapper

init push

23 minutes ago



src

init push

23 minutes ago



.gitignore

init push

23 minutes ago



mvnw

init push

23 minutes ago



mvnw.cmd

init push

23 minutes ago



pom.xml

init push

23 minutes ago

Help people interested in this repository understand your project by adding a README.

[Add a README](#)

Options

Manage access

Security & analysis

Branches

Webhooks

Notifications

Integrations

Deploy keys

Secrets

Actions

Moderation settings

Interaction limits

## Who has access

PUBLIC REPOSITORY



This repository is public and visible to anyone.

[Manage](#)

DIRECT ACCESS



0 collaborators have access to this repository. Only you can contribute to this repository.

## Manage access



You haven't invited any collaborators yet

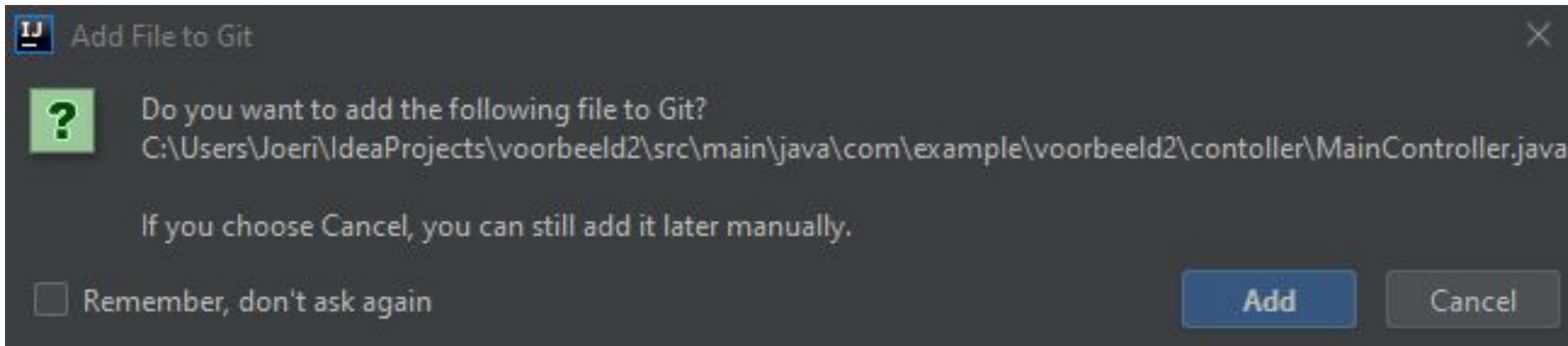
[Invite a collaborator](#)



# AANMAKEN NIEUWE FILES

Bij aanmaken nieuwe file

- Word gevraagd om aan git toe te voegen
- Selecteer '**Remember, don't ask again**' en klik op **Add**



# COMMIT

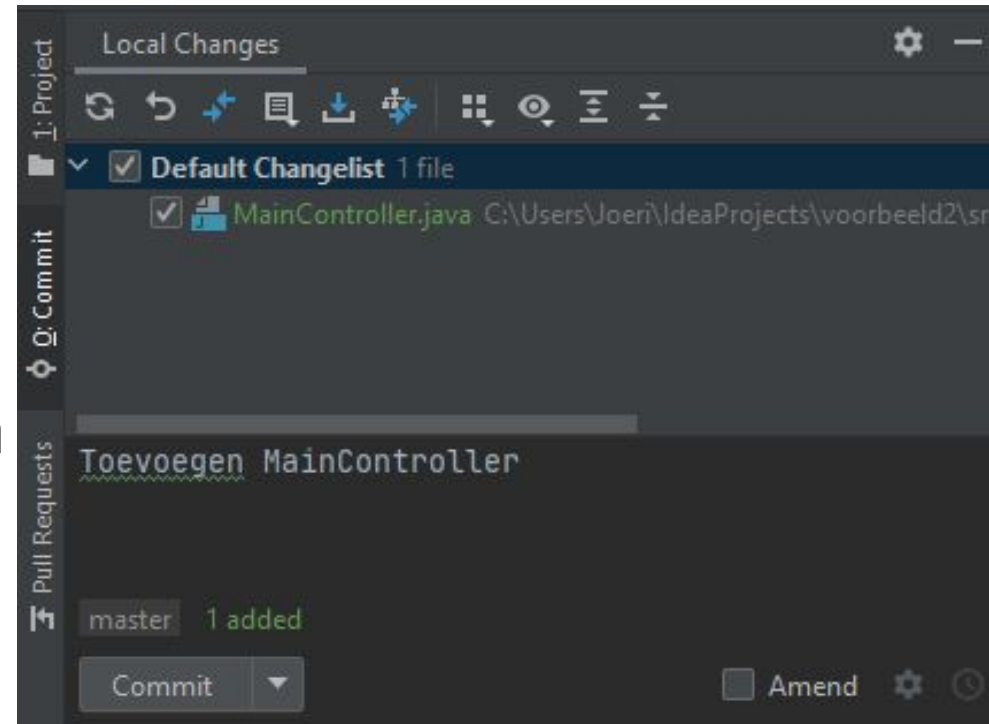
Rechtsboven > Groen vinkje



Lokale wijzigingen worden weergegeven in het tabje **commit** > **local changes**

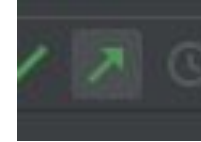
Commit nu door:

- De files aan te duiden die wil committen
- Een **duidelijke boodschap** mee te geven



# PUSH

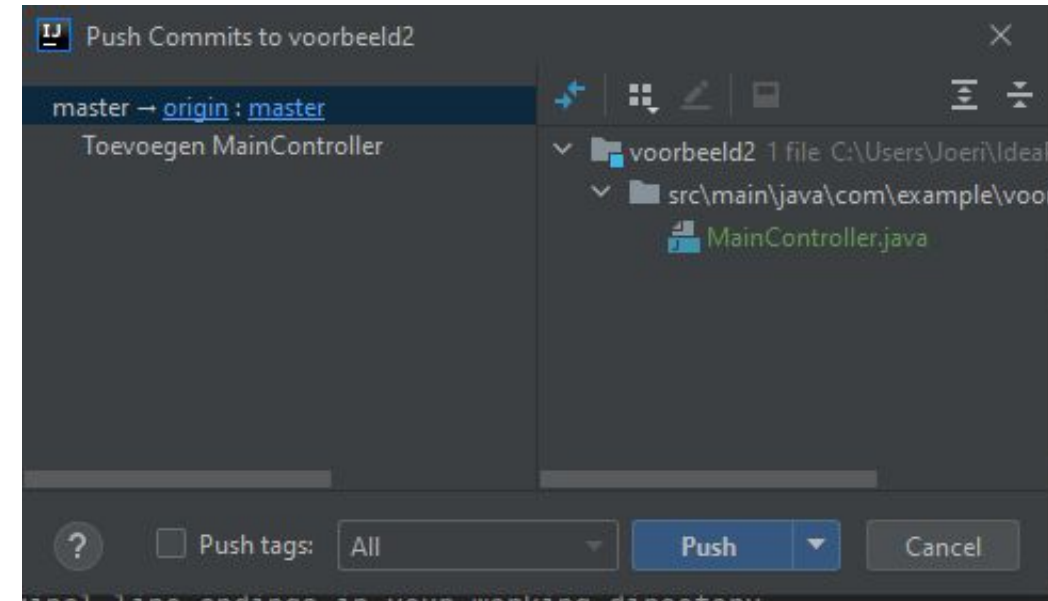
Rechtsboven > Groen pijltje



Overzicht van 1 of meerdere commits

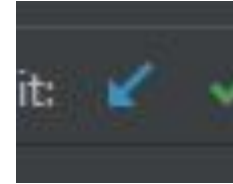
Push nu door:

- De **commits nogmaals te controleren**
- Op de push knop te klikken



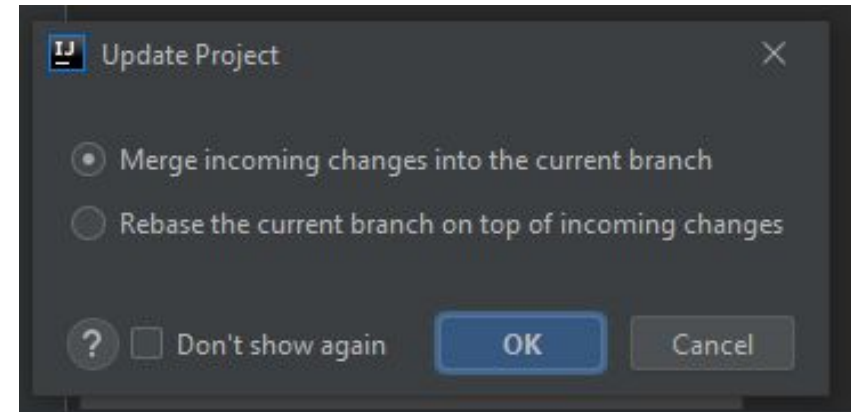
# PULL

Rechtsboven > blauwe pijltje



Merge incoming changes into current branch

Los eventuele merge conflicten op!





# VOOR PROJECT

---

- 1 Persoon plaatst het project op Github (**public access**)
- Deze persoon nodigt teamleden uit als contributors
- Andere teamleden pullen het project

# VERVOLG

---

- Branches
- Pull requests
- ...