

## Course 3a: Python II



**Geavanceerde concepten  
in Python en  
programmeren voor  
bio-informatica  
toepassingen**

# Curriculum overzicht

|         | Periode 1  | Periode 2   | Periode 3  | Periode 4   |
|---------|--|---|--|---|
| 1e jaar | Relationele databanken voor het beheer van gegevens uit biologische experimenten | Programmeren in Python (1)                                      | Programmeren in python (2)                         | Geautomatiseerde identificatie van eiwitten via sequentievergelijking               |
|         |  | Opsporen van genetische mutaties bij erfelijke ziektes          | Vergelijkende genoomanalyse: evolutie van virussen | Sequentie alignment (BLAST) en functionele eiwit analyse                            |
| 2e jaar | Programmeren in Java   | Datastructuren en algoritmen in Java                            | Analyse en ontwerptechnieken                       | Webtechnologie en textmining  |
|         | Proteomics: eiwitstructuren, eiwitfuncties, scheidingstechnieken en data analyse | Transcriptomics: data analyse mbt regulatie van metabole routes | Genomics: annoteren van genomisch DNA              | Moleculaire fylogenie: evolutie, multiple sequence alignment mbt signaaltransductie |
| 3e jaar | Stage of minor   |   | Data Mining en grid computing                      | RNA seq, biostatistiek, webservices en workflows                                    |
|         |  |   | Eiwitten: carcinogenese en eiwitstructuren         |   |
| 4e jaar | Stage, afstudeerstage of minor   |   | Minor of afstudeerstage                            |   |

## Terugblik

- Variabelen
- Conditionele controle
- Iteratieve controle
- Exceptions
- Commentaar

# Opzet OWE 3a

| es | Onderwerp   | Theorie  | Opgaven                 |
|----|---|--|-------------------------|
| 1  | <ul style="list-style-type: none"> <li>Review Python I</li> <li>Pseudocode</li> <li>Flowcharts</li> <li>Documenteren en Testen</li> </ul> | H1..8 SowP*  | 1<br>Afvinkopdracht     |
| 2  | <ul style="list-style-type: none"> <li>Graphs</li> <li>Strings</li> </ul>   | <a href="#">Matplotlib tutorial</a><br>H9 More about Strings   | 2<br>Afvinkopdracht     |
| 3  | <ul style="list-style-type: none"> <li>Datastructuren:</li> <li>Dictionaries</li> <li>Sets</li> </ul>                                     | H10 Dictionaries and Sets                                      | 3<br>Afvinkopdracht     |
| 4  | <ul style="list-style-type: none"> <li>Text and Language Processing</li> <li><a href="#">Regular Expressions</a></li> </ul>               | <a href="#">H7 DiP**</a>                                       | 4<br>Afvinkopdracht     |
| 5  | <ul style="list-style-type: none"> <li>Object-Oriented Programming</li> </ul>   | H11 Classes and Object-Oriented Programming<br>H12 Inheritance | 5<br>Afvinkopdracht     |
| 6  | <ul style="list-style-type: none"> <li>Recursion</li> </ul>   | H13 Recursion  | 6<br>Afvinkopdracht     |
| 7  | <ul style="list-style-type: none"> <li>GUI Programming</li> </ul>   | H14 GUI Programming  | Voorbeeld<br>thematoets |



\*SowP: Starting out with Python

\*\*DiP Dive into Python

# Agenda

- **Pseudocode**
- **Debugging en testing**
- **Gebruik van NetBeans**

## Wat is pseudocode?

- Met pseudocode wordt over het algemeen een onechte (niet-formele) programmeertaal bedoeld. 'Pseudo' betekent onecht, 'code' verwijst naar de broncode van een computerprogramma. Doordat pseudocode een informeel karakter heeft, is het niet geschikt om gecompileerd en uitgevoerd te worden. Het is hierdoor geen echte programmeertaal. (Bron: wikipedia)

## Pseudocode

- Benoem de voordelen van pseudocode ten op zichte van een flowchart
- Benoem de nadelen van pseudocode ten op zichte van een flowchart
- Wanneer gebruik je pseudocode?
- Wanneer gebruik je een flowchart?

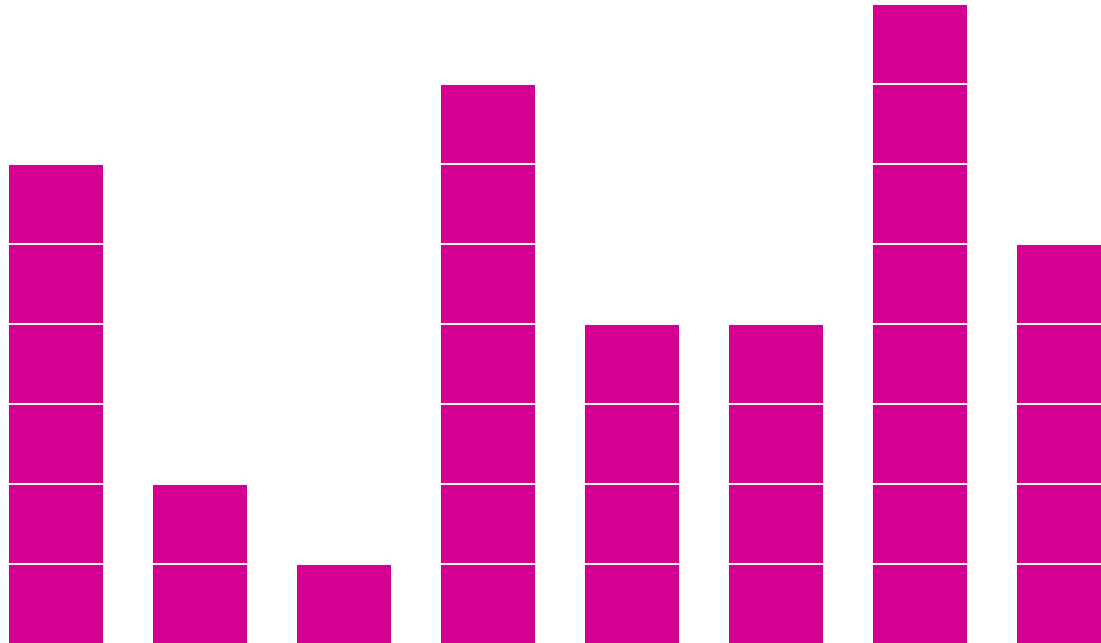
## Probleem

- We nemen het volgende probleem: een aantal elementen willen we sorteren op grootte
- Hierna bekijken we de oplossing in
  - Flowchart
  - Pseudocode
  - Python code

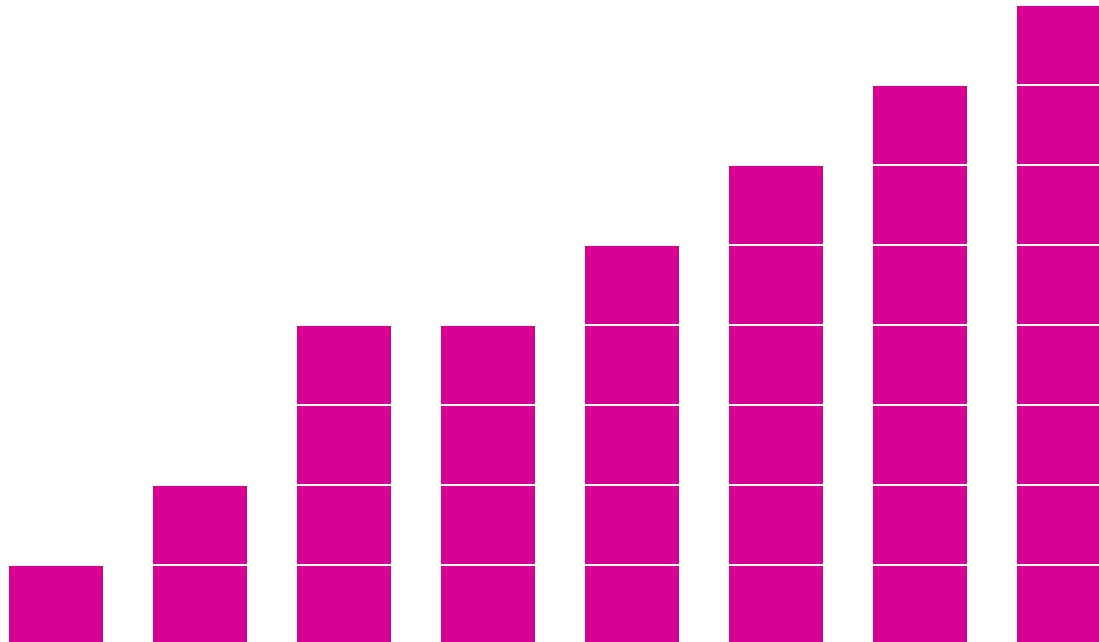


## Voorbeeld: sorteren

**Sorteer de onderdelen op grootte**



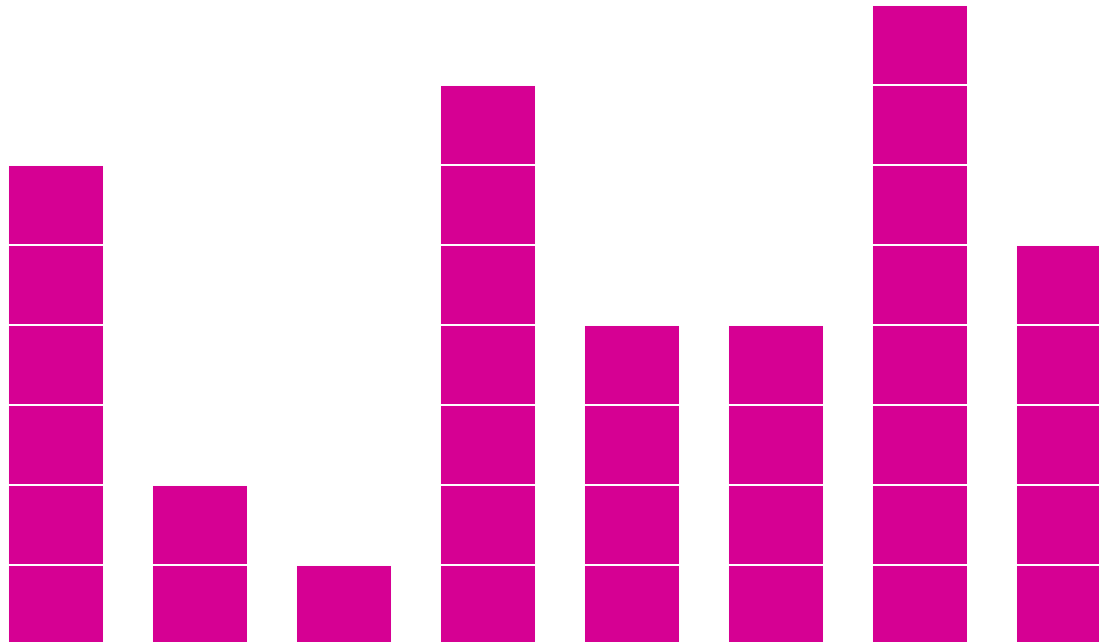
# Verwacht resultaat



## Vraag

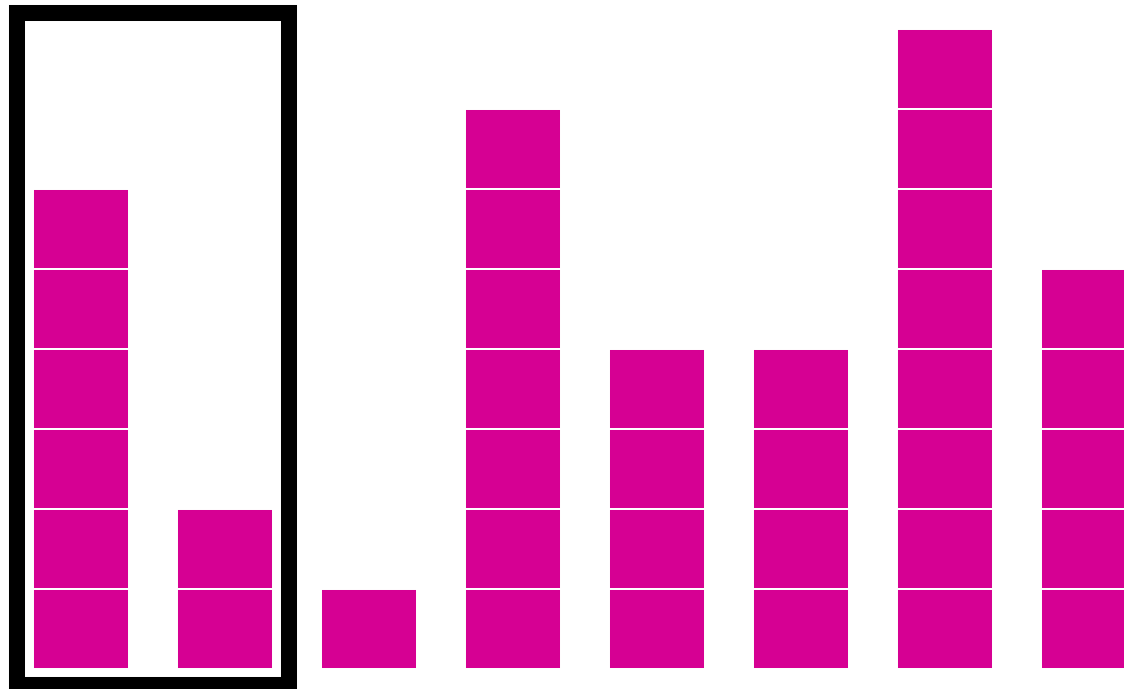
- Bepaal een oplossingsstrategie

# De objecten om te sorteren



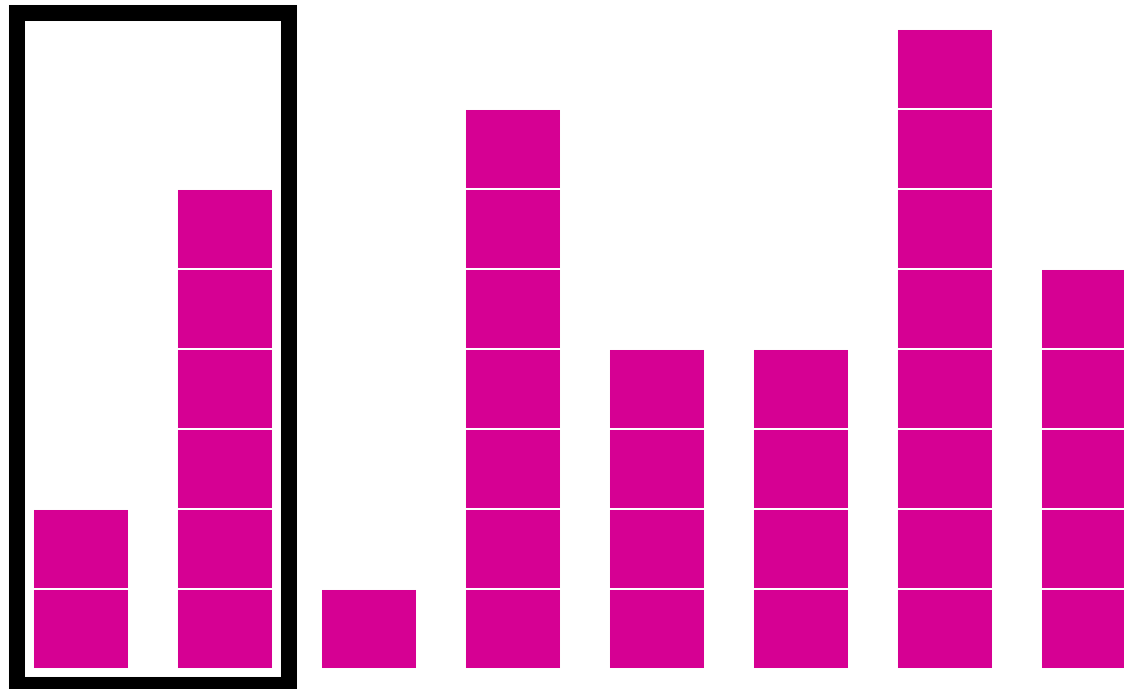
- **Is de de lijst op volgorde?**
  - Nee: ga door
  - Ja: stop

# Sorting: Step A1

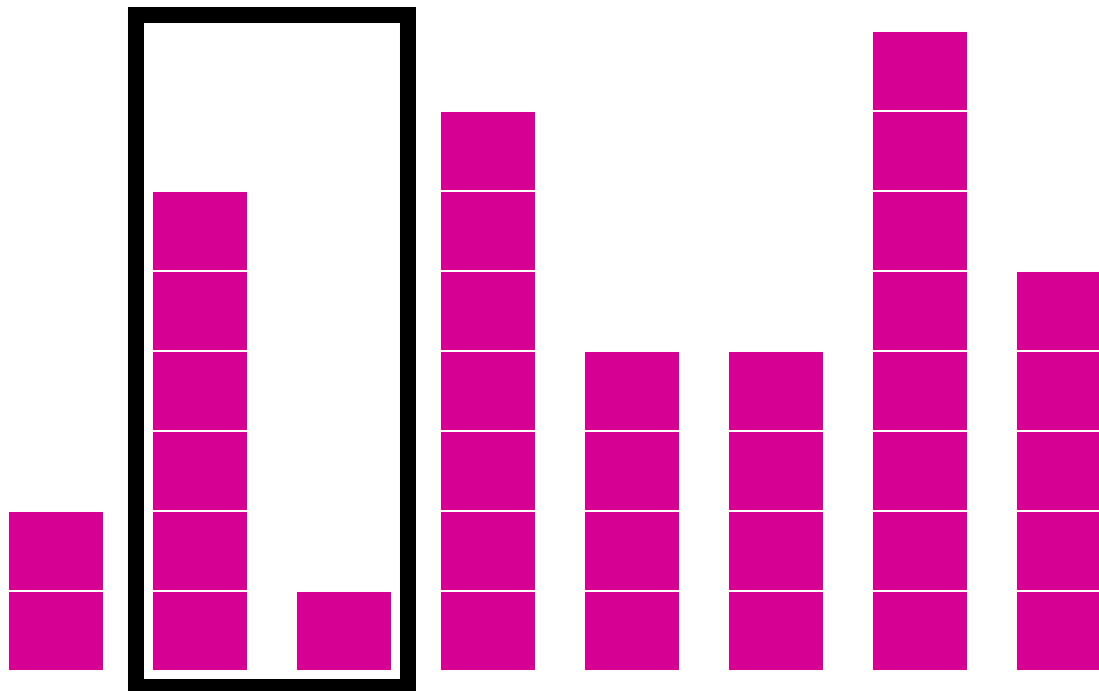


# Sorting: Step A1

Swap? Ja



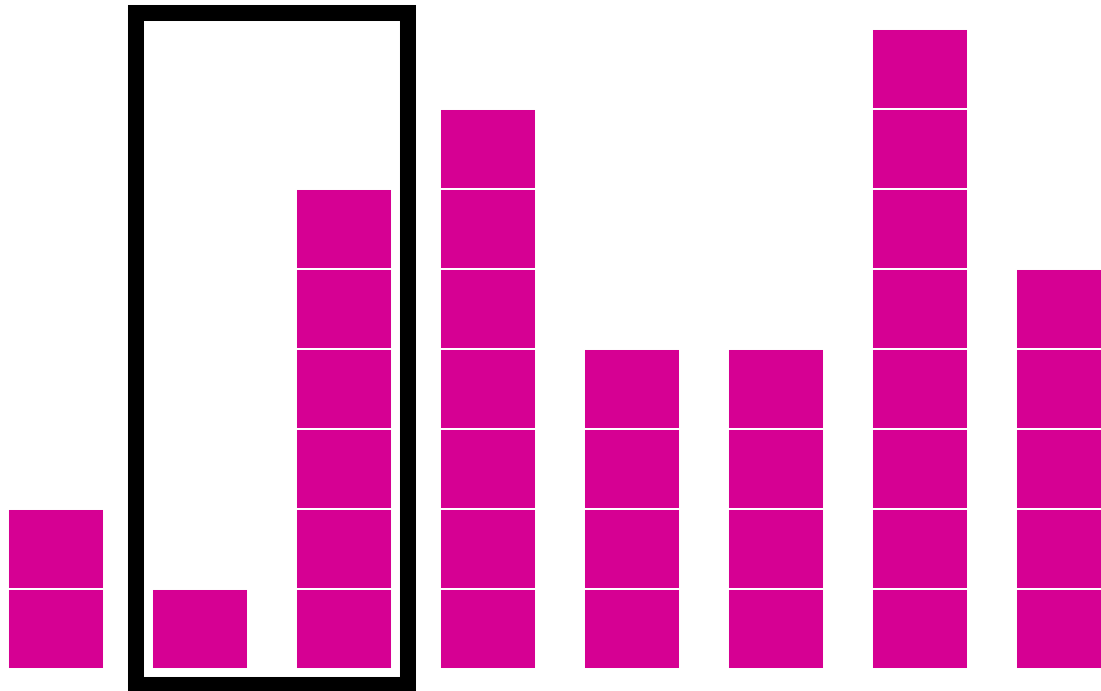
# Sorting: Step A2



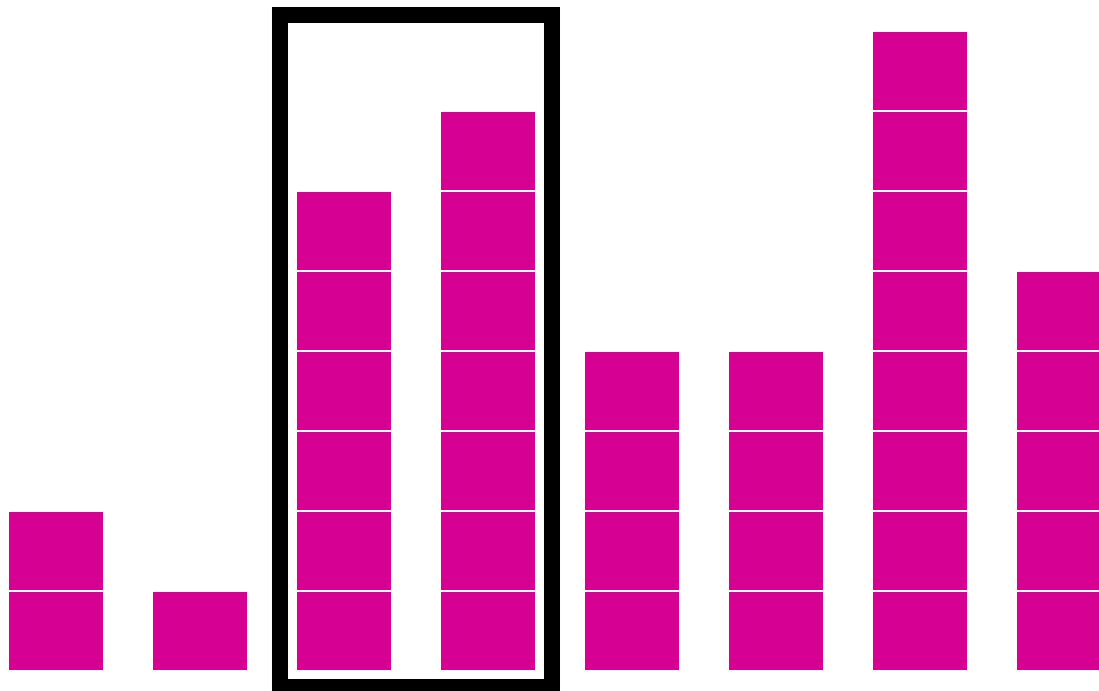


## Sorting: Step A2

Swap? Ja

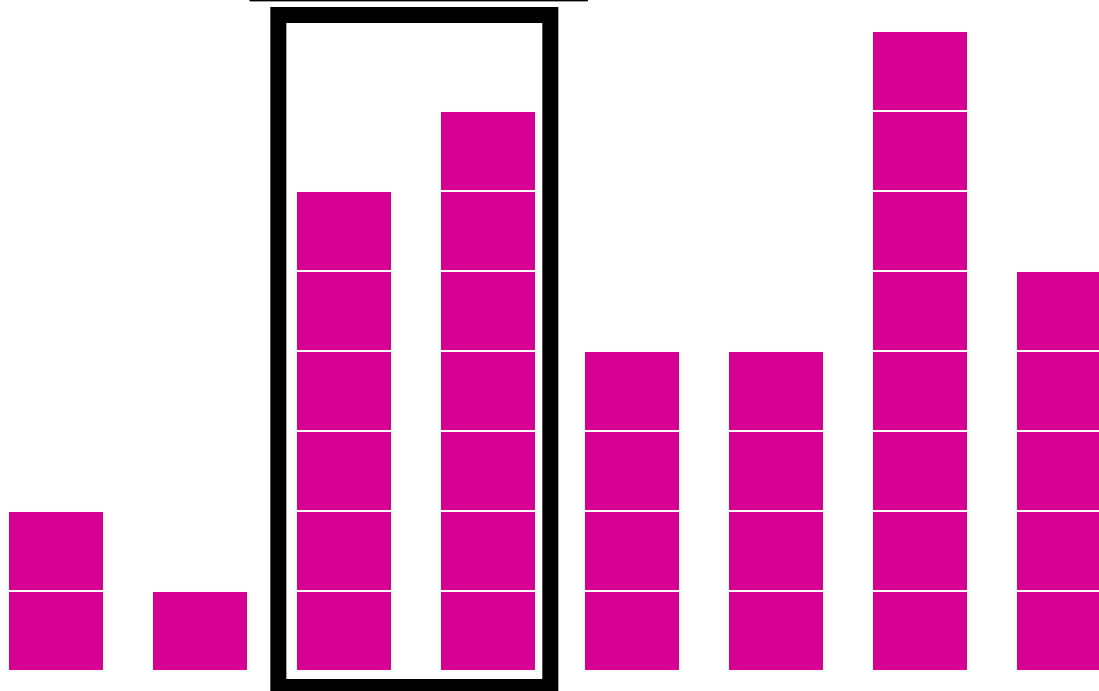


## Sorting: Step A3

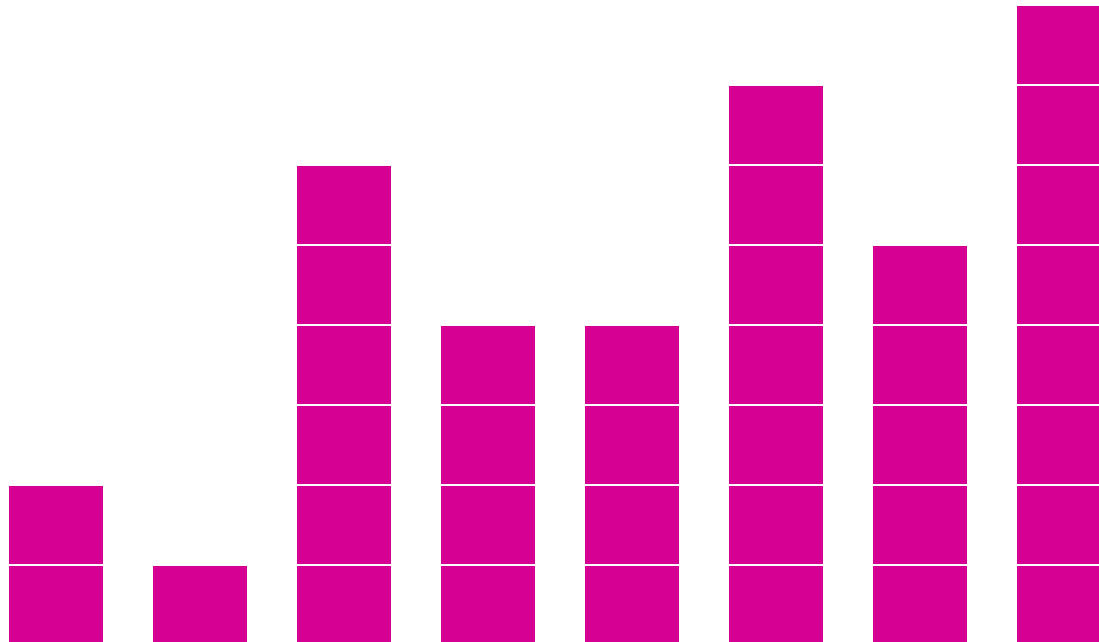


## Sorting: Step A3

Swap? No



# Volgorde na stap 7

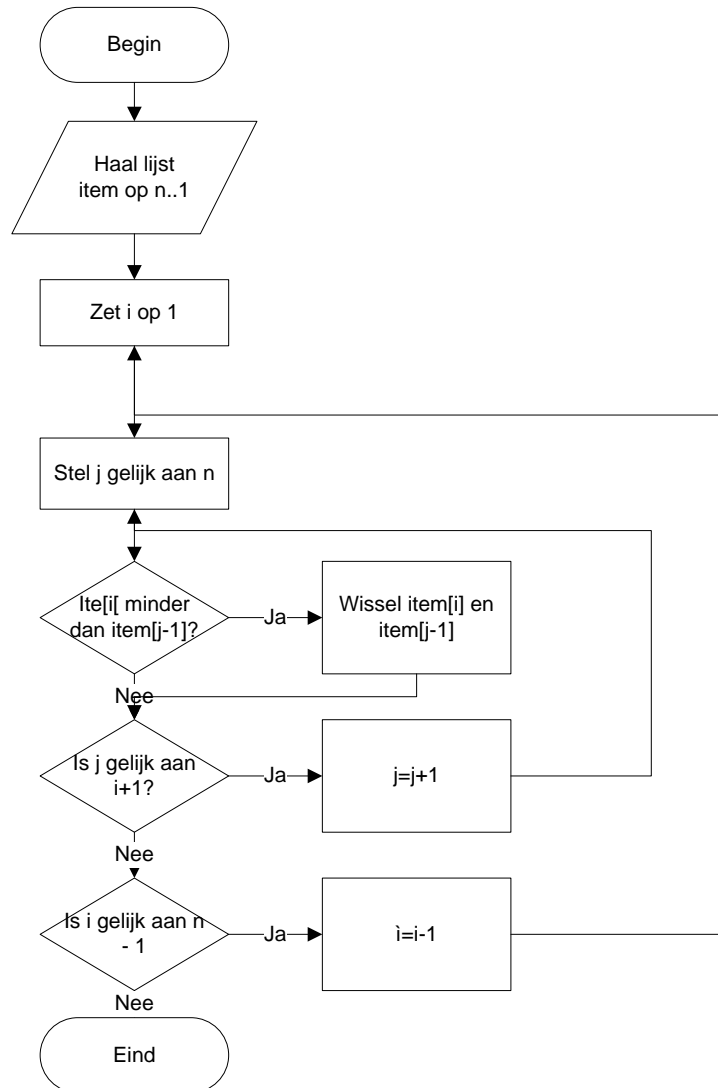


- **Herhaal dit proces tot alles op volgorde staat**

# Bubble Sort - Pseudocode

```
procedure bubbleSort( A : list of sortable
items ) defined as:
    do
        swapped := false
        for each i in 0 to length(A) - 2
            inclusive do:
                if A[ i ] > A[ i + 1 ] then
                    swap( A[ i ], A[ i + 1 ] )
                    swapped := true
                end if
            end for
        while swapped
    end procedure
```

# Bubble Sort - Flowchart



Flowchart eigen werk auteur

# Bubblesort in Python

```
# Bubble Sort
def bubbleSort(lijst, max):
    for n in range(0, max):
        temp = 0
        for i in range(1, max):
            temp = lijst[i]
            if lijst[i] < lijst[i-1]:
                lijst[i] = lijst[i-1]
                lijst[i-1] = temp
```



## Vragen

- Zal bubblesort het meest efficiënte algoritme zijn om te sorteren?
- Wat is het voordeel aan Pseudocode boven Python code?
- Wat is meer standaard, flowchart of pseudocode?
- Waar worden flowcharts toegepast?

## Samengevat

- **Pseudocode is een efficiënte manier om snel een kort stuk code uit te werken**
- **Het is niet de bedoeling hele programma's erin uit te werken**

# Agenda

- Pseudocode
- **Debugging en testing**
- Gebruik van NetBeans

## Debugging vs. Testing

- Debugging is het zoeken naar fouten in een programma dat wordt ontwikkeld. Deze fouten zijn **bekend**.
- Testen is het zoeken naar **onbekende** fouten. Verondersteld wordt dat het programma juist werkt.

## Debugging vs. Testing

- Met testen kunnen we de aanwezigheid van fouten aantonen maar niet de afwezigheid.
- Debuggen volgt dus na testen.

## Professional Testing

- Er zijn professionele testers
- Deze testen software op specificatie
- Bijvoorbeeld in de farmaceutische industrie is dit extreem belangrijk
- Testers maken ook gebruik van testtools

## Niveau van program *correctness*

- Geen syntax errors
- Geen semantische errors
- Er bestaat test data waar het programma met het juiste antwoord komt
- Het programma geeft een juist antwoord op willekeurige data
- Programma geeft correcte antwoorden met complexe test data

## Niveau van program *correctness*

- Alle legale input leidt tot een juist antwoord
- Alle legale en alle foute input leidt tot een juist antwoord



# Voorbeeld van een test script

PROJECT NAME: ENTER PROJECT NAME

| Test Script Name:      |             | T/S #:           |      |      |     |                 |
|------------------------|-------------|------------------|------|------|-----|-----------------|
| Scenario/Purpose:      |             |                  |      |      |     |                 |
| Test Cases Verified:   |             |                  |      |      |     |                 |
| Requirements Verified: |             |                  |      |      |     |                 |
| Prerequisites:         |             |                  |      |      |     |                 |
| Name of Tester:        |             | Date:            |      |      |     |                 |
| Build Number:          |             | Time:            |      |      |     |                 |
| Step                   | Description | Expected Results | Pass | Fail | N/A | Defect/Comments |
| 1                      |             |                  |      |      |     |                 |
| 2                      |             |                  |      |      |     |                 |
| 3                      |             |                  |      |      |     |                 |
| 4                      |             |                  |      |      |     |                 |
| 5                      |             |                  |      |      |     |                 |
| 6                      |             |                  |      |      |     |                 |
| 7                      |             |                  |      |      |     |                 |
| 8                      |             |                  |      |      |     |                 |
| 9                      |             |                  |      |      |     |                 |
| 10                     |             |                  |      |      |     |                 |
| 11                     |             |                  |      |      |     |                 |
| 12                     |             |                  |      |      |     |                 |
| 13                     |             |                  |      |      |     |                 |
| 14                     |             |                  |      |      |     |                 |
| 15                     |             |                  |      |      |     |                 |
| 16                     |             |                  |      |      |     |                 |
| 17                     |             |                  |      |      |     |                 |
| 18                     |             |                  |      |      |     |                 |
| 19                     |             |                  |      |      |     |                 |
| 20                     |             |                  |      |      |     |                 |
| End of Test Case       |             |                  |      |      |     |                 |

## Debugger Commands

- <http://pythonconquerstheuniverse.wordpress.com/category/python-debugger/>

```
# experiment with Python
debugger

import pdb

a = "aaa"

pdb.set_trace()

b = "bbb"

c = "ccc"

final = a + b + c

print (final)
```

## Samenvatting

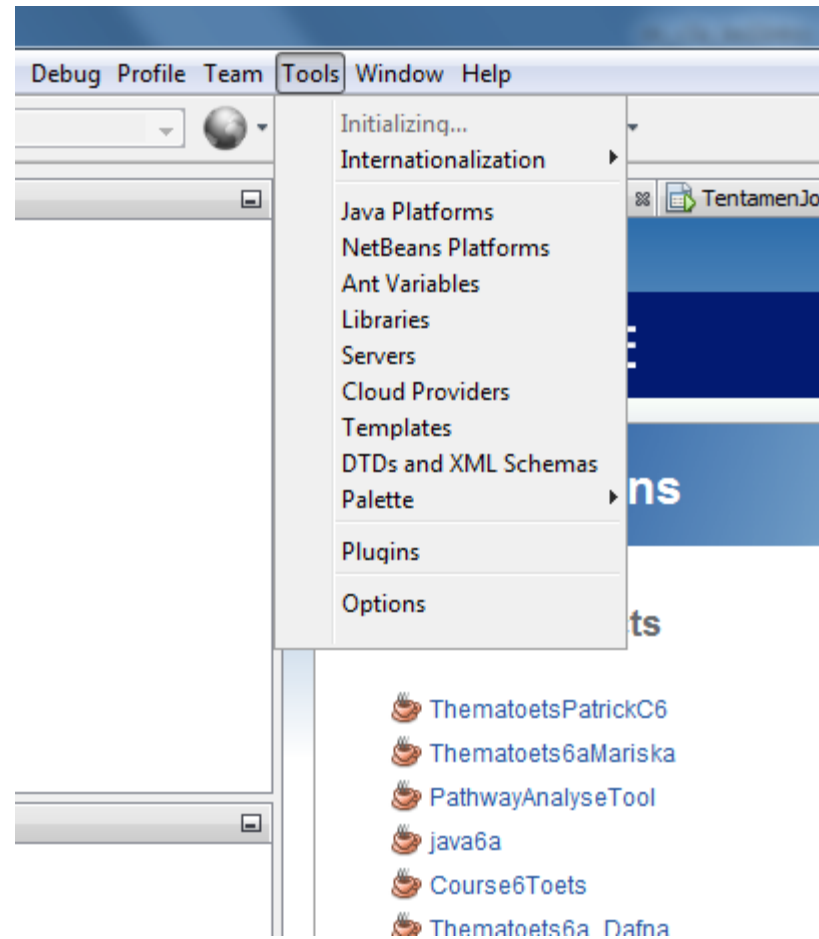
- Debugging en testen is essentieel voor software ontwikkeling
- <http://pythonconquerstheuniverse.wordpress.com/category/python-debugger/>

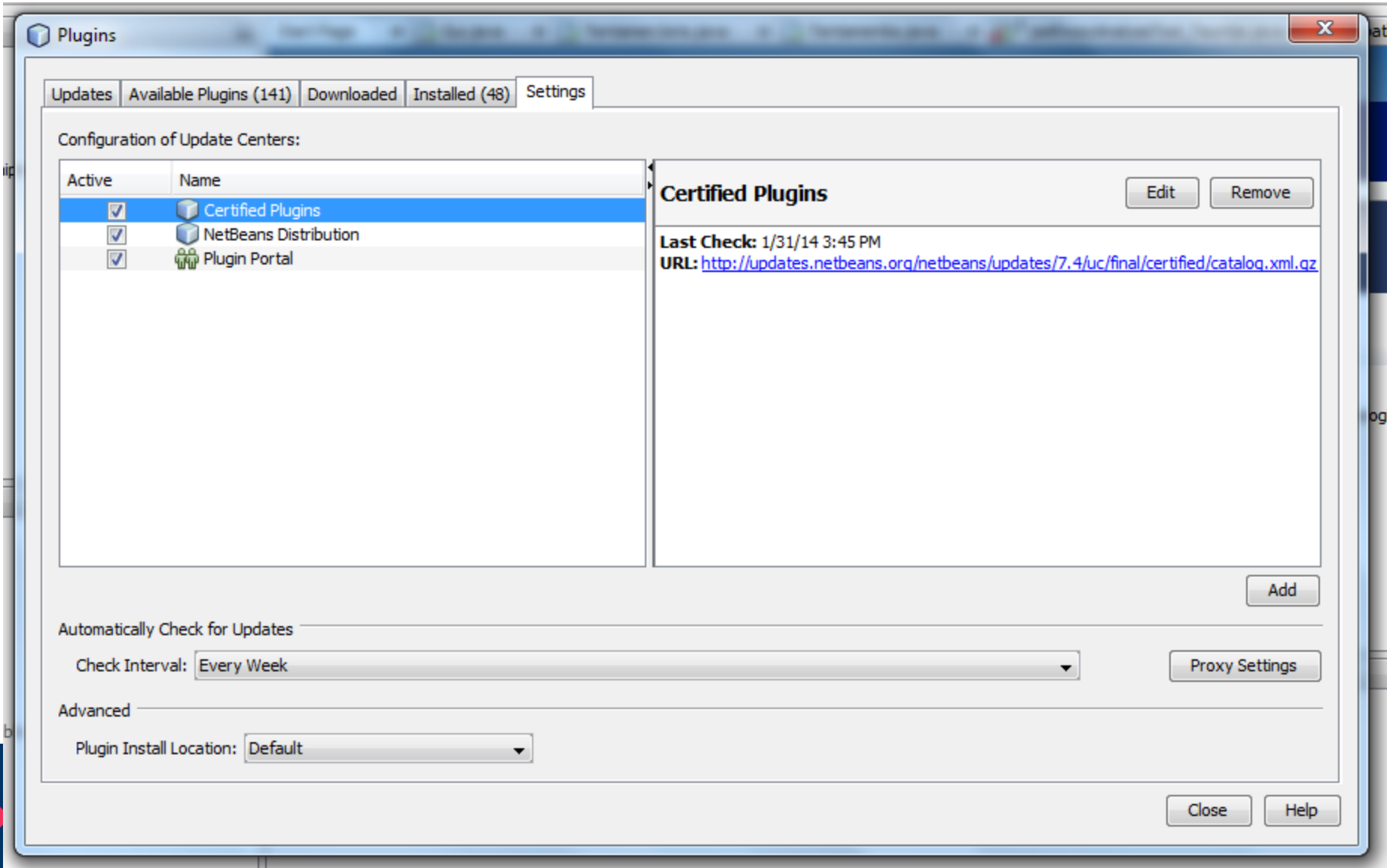
# Agenda

- Pseudocode
- Debugging en testing
- **Gebruik van NetBeans**

## NetBeans

- **NetBeans is een omgeving waarin professionele programmeurs software ontwikkelen**
- **Volgend jaar gebruiken we dit ook voor Java**

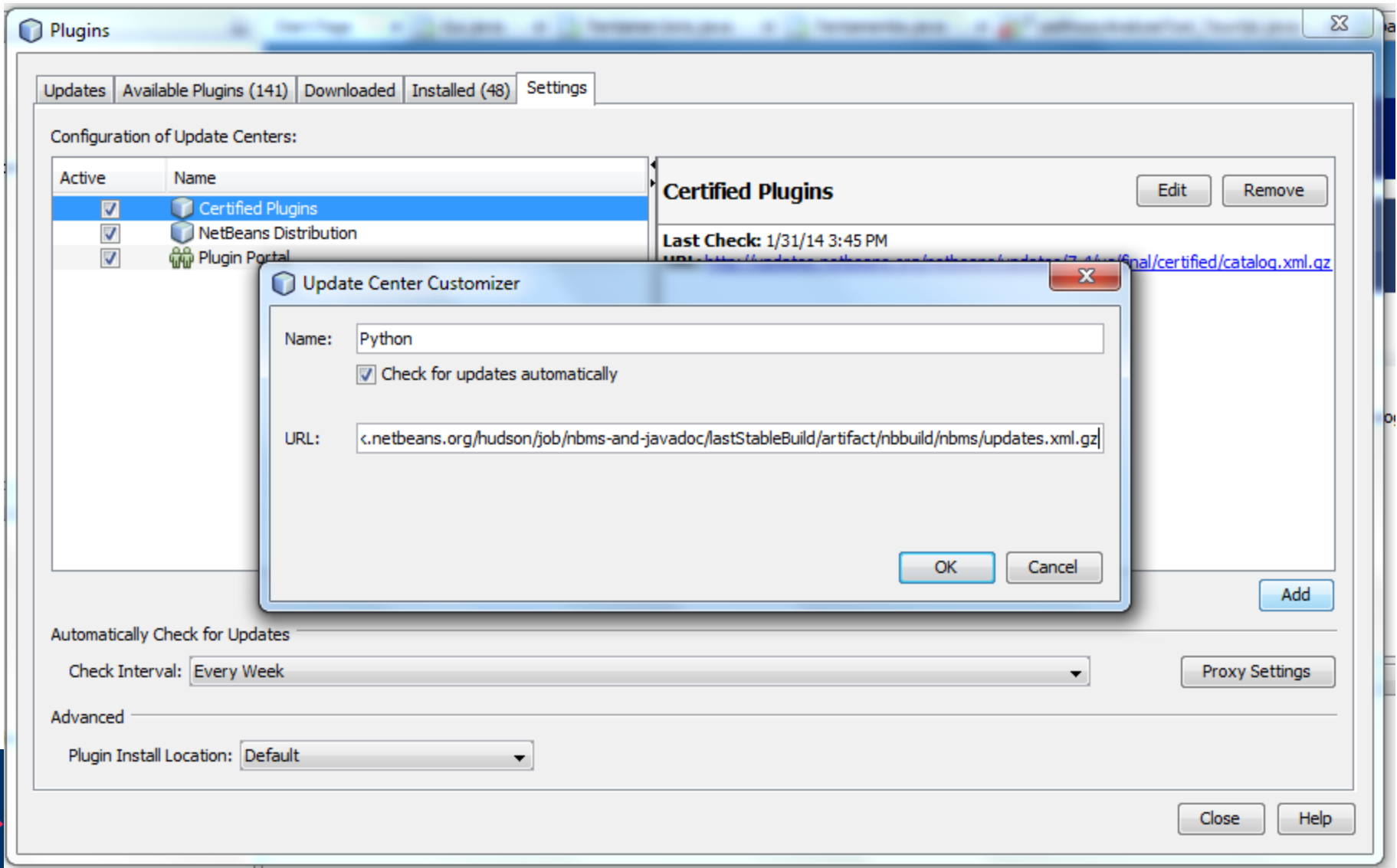




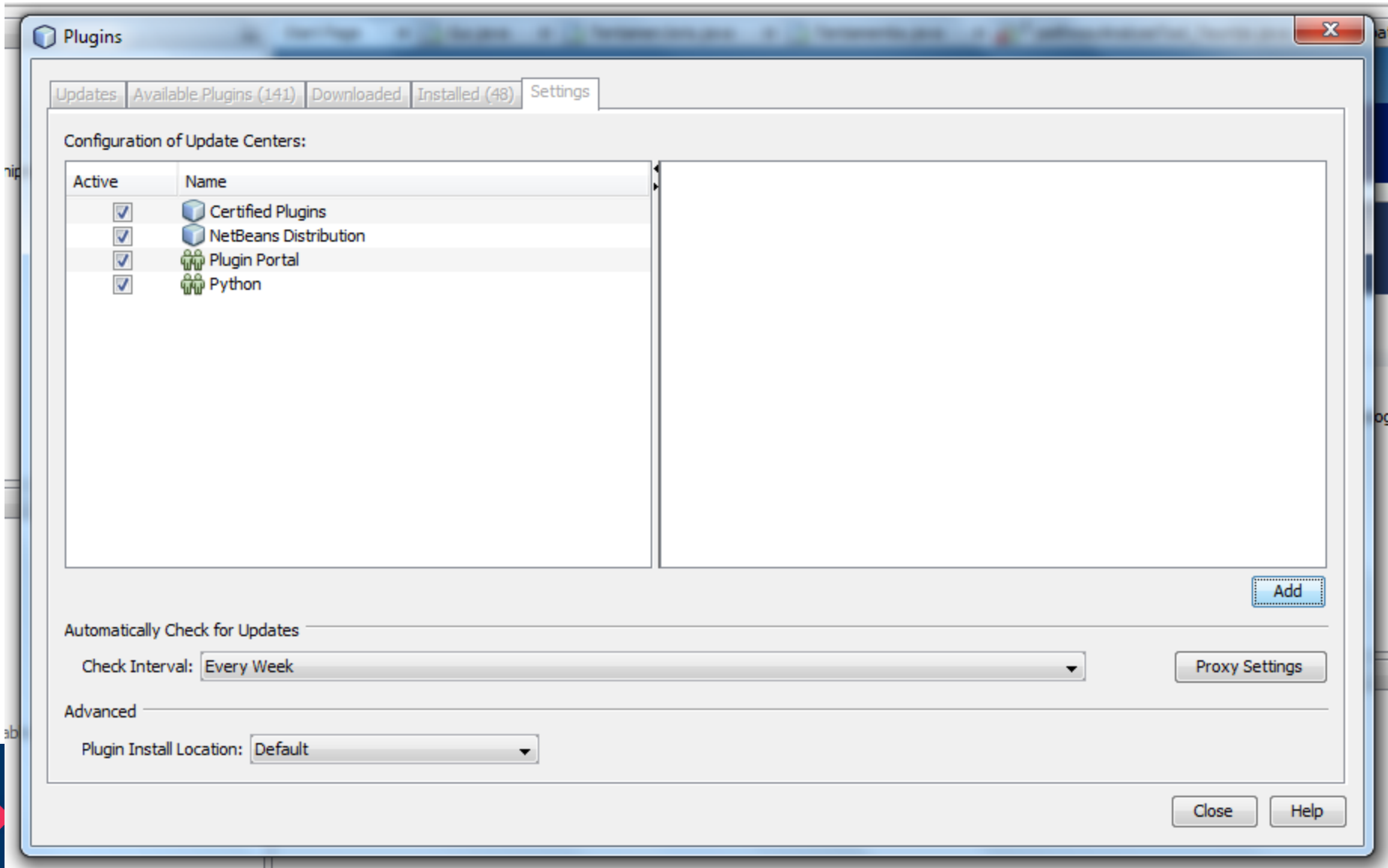
Screenshot NetBeans eigen werk auteur



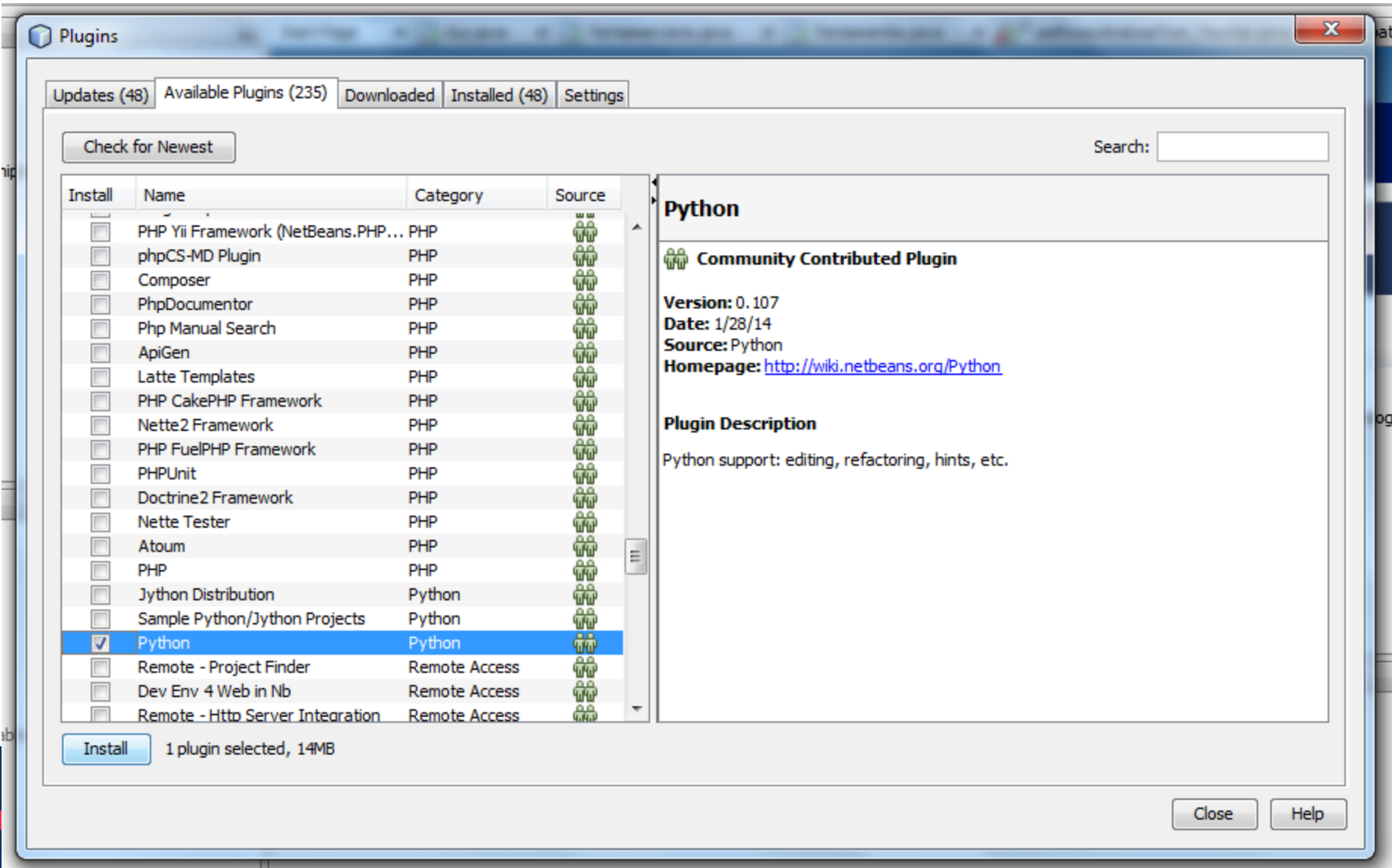
http://deadlock.netbeans.org/hudson/job/nbms-and-javadoc/lastStableBuild/artifact/nbbuild/nbms/updates.xml.gz



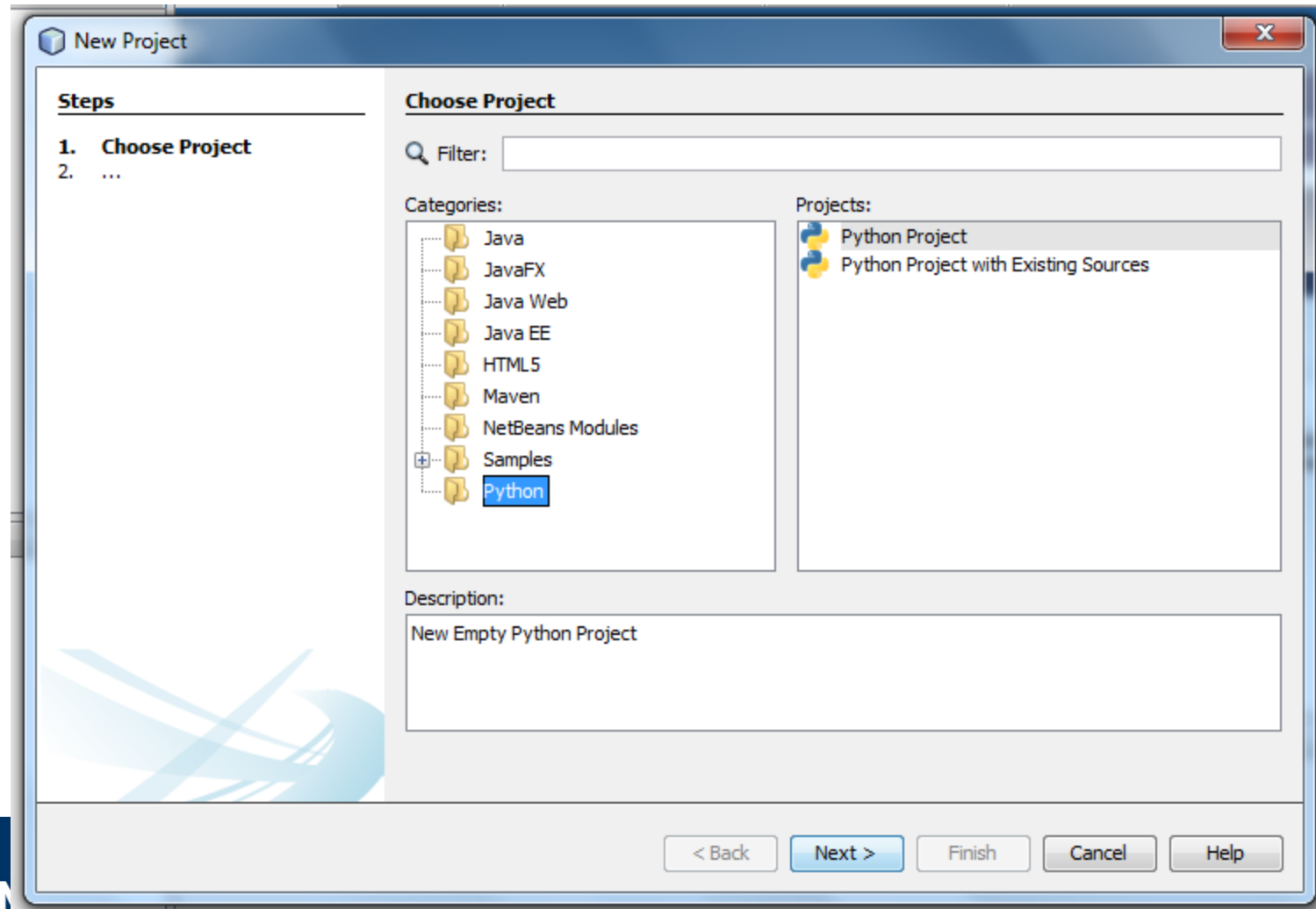
Screenshot NetBeans eigen werk auteur

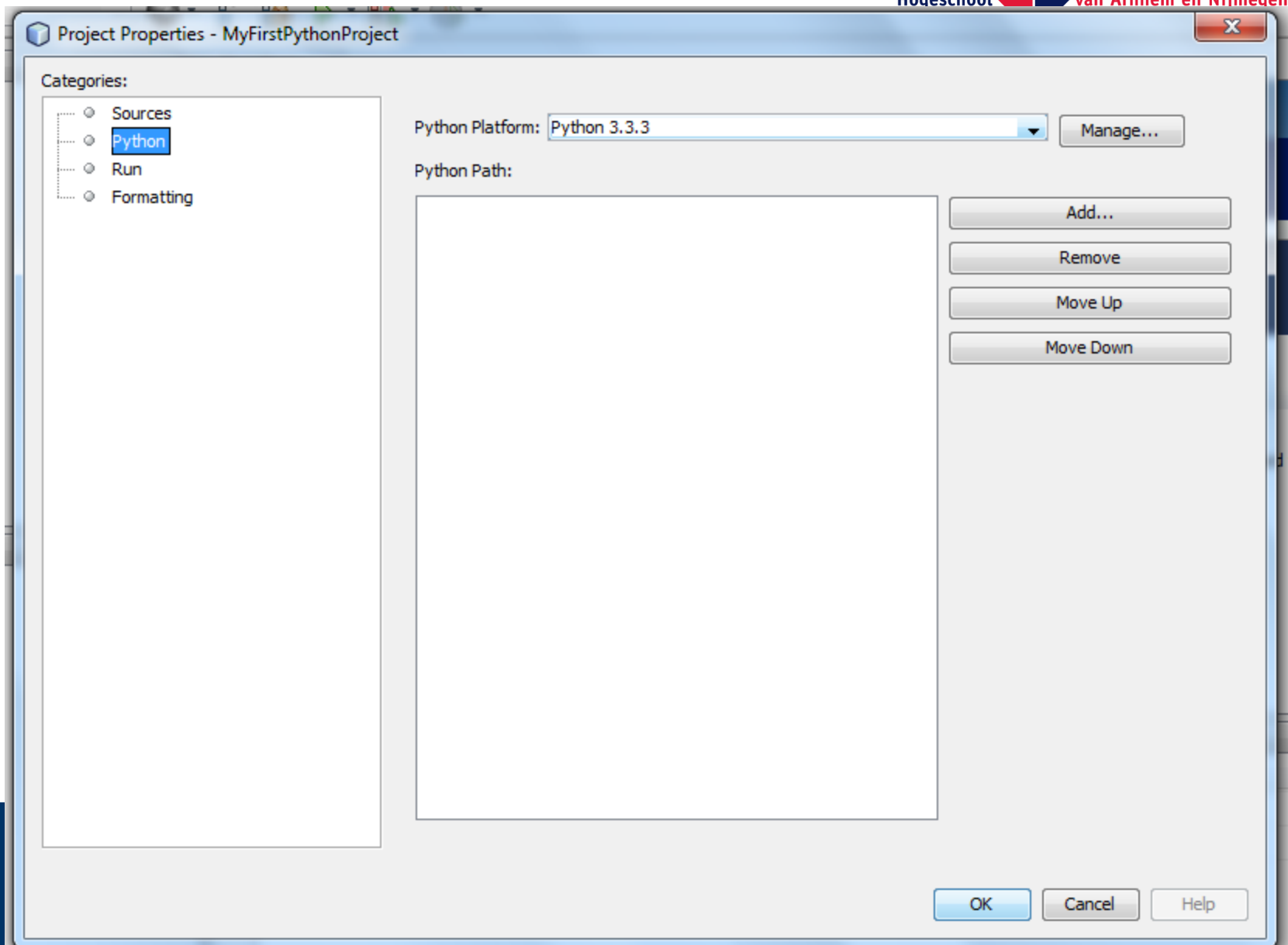


Screenshot NetBeans eigen werk auteur



Screenshot NetBeans eigen werk auteur





Screenshot NetBeans eigen werk auteur

## Verantwoording

- In deze uitgave is géén auteursrechtelijk beschermd werk opgenomen
- Alle teksten © Martijn van der Bruggen/HAN tenzij expliciet externe bronnen zijn aangegeven
- Screenshots op basis van eigen werk auteur en/of vernoemde sites
- Eventuele images zijn opgenomen met vermelding van bron