

# Titel



**BIN-OWE1**

# Studiewijzer

Les	Onderwerp	
	Algemeen	Python
1	Linux	
2	Git	Input/output
3	Pseudocode	If/elif/else Booleans
4		For loop
5		Lists and tuples Files (CSV bestanden)
6	Flowchart	Functies
7		Strings CSV bestanden

# Studiemateriaal

- Boek: “Starting Out with Python, third edition”
- Reader: “Linux voor Bio-informatici”
- Onderwijs Online
- Kom je er niet uit? → Google



# Doelstellingen

- Aan het eind van deze week kan je
  - Pseudocode gebruiken om een programma te ontwerpen
  - if/elif/else structuren toepassen
- Aan het eind van deze week begrijp je
  - De beginselen van flowcharts
  - Wat booleans zijn en hoe je ze kunt inzetten
  - Waarom je if/elif/else structuren gebruikt

# Inhoud

- Pseudocode
- Booleaanse variabelen
- Logische operatoren

# Pseudocode

- Het idee achter pseudocode is dat het code is, maar dan heel goed leesbaar voor mensen
- Programmeertaal onafhankelijk
- Handig als je met meerdere mensen samenwerkt

## Volgens wikipedia

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)

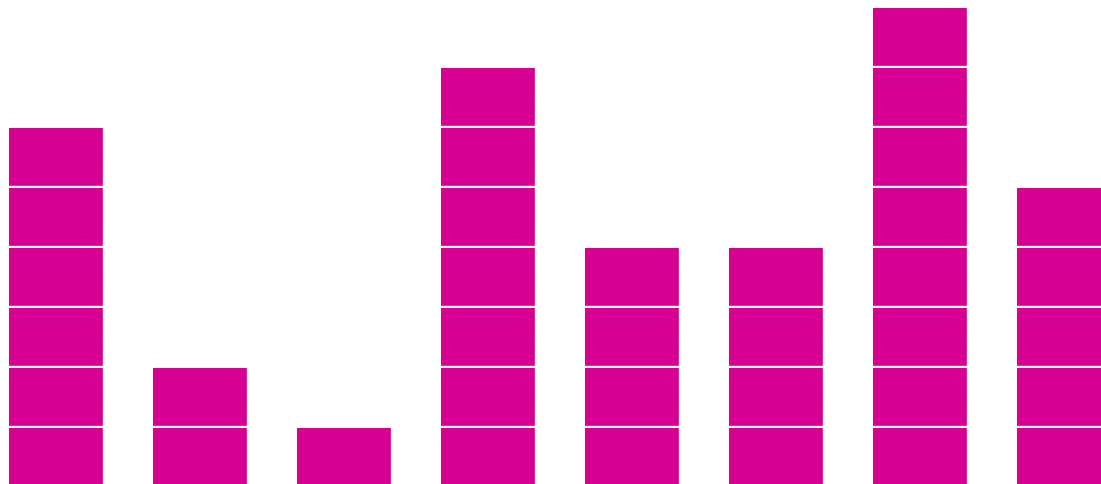
# Probleem

- We nemen het volgende probleem: een aantal elementen willen we sorteren op grootte

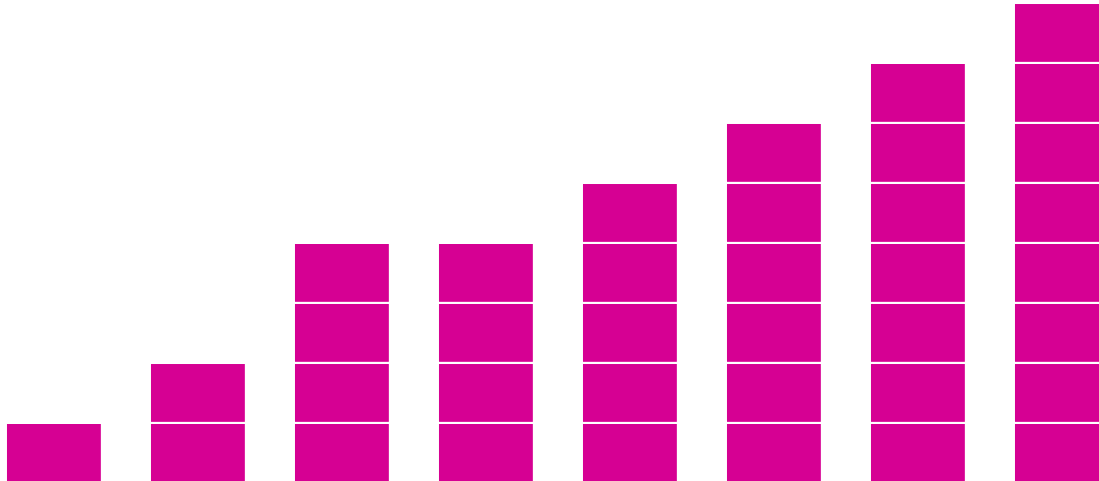


# Voorbeeld: sorteren

Sorteer de onderdelen op grootte

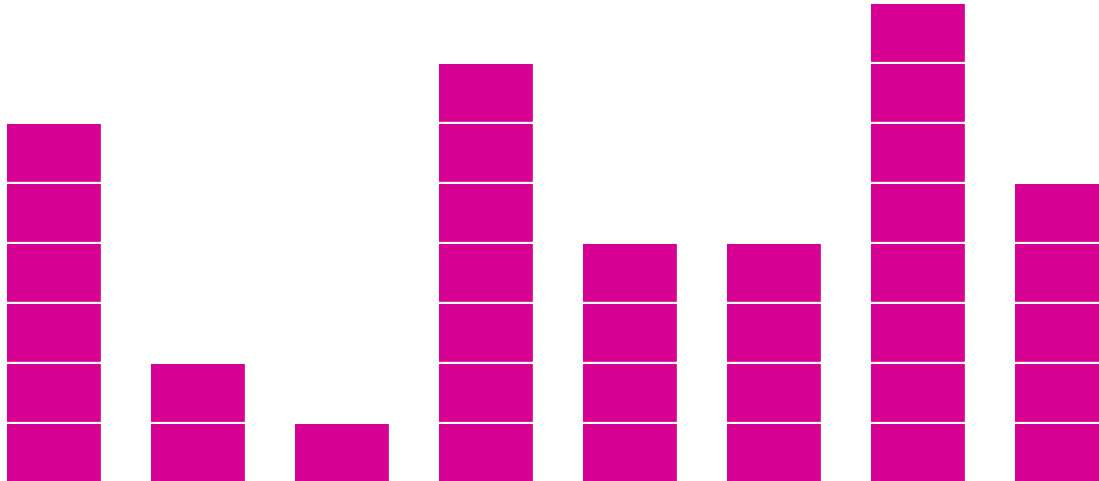


# Verwacht resultaat



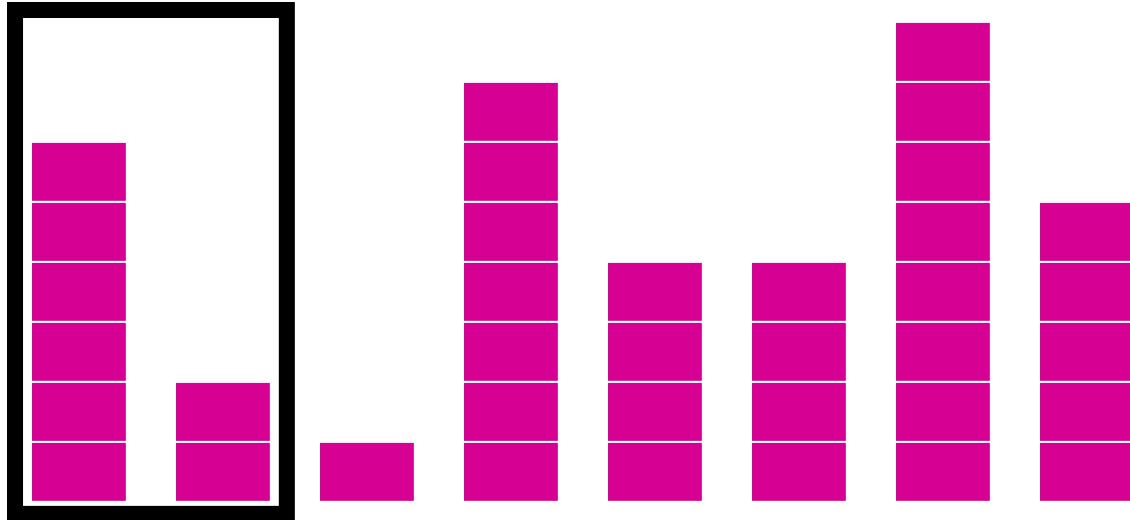
# De objecten om te sorteren

- Bepaal een oplossingsstrategie



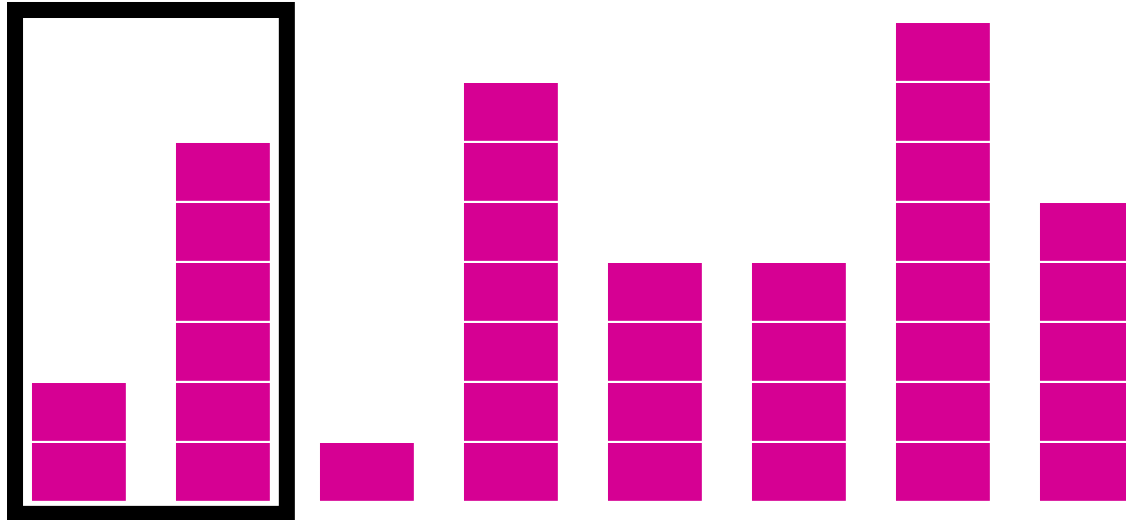
## Sorting: Step A1

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

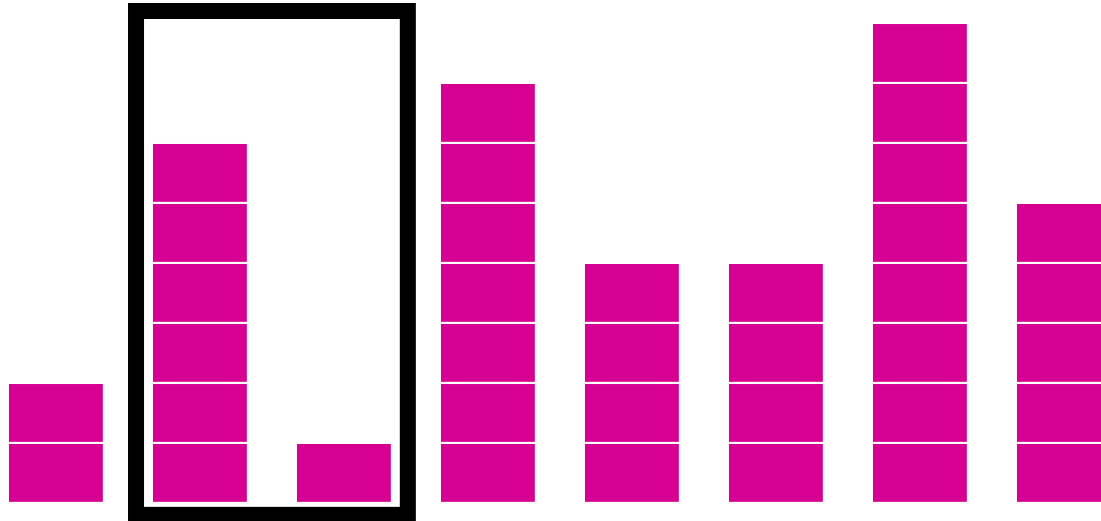


# Sorting: Step A1

Swap? Ja

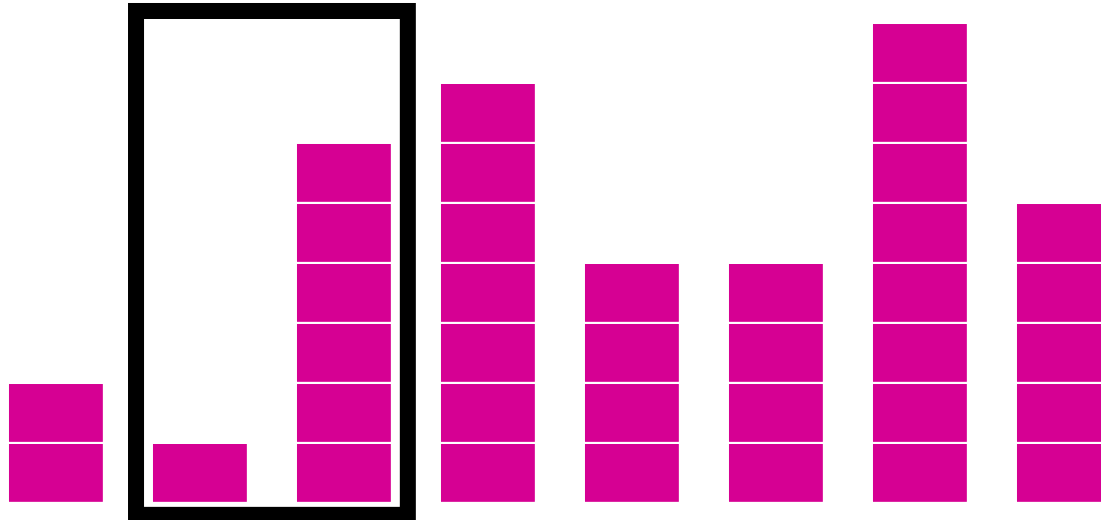


# Sorting: Step A2

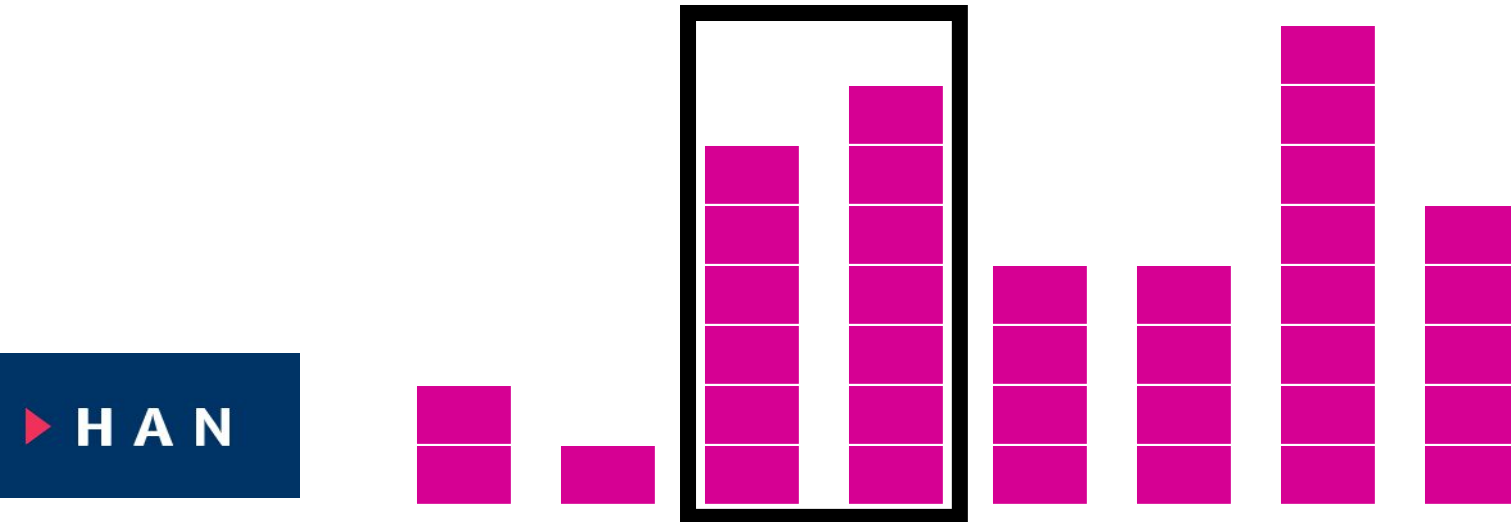


## Sorting: Step A2

Swap? Ja



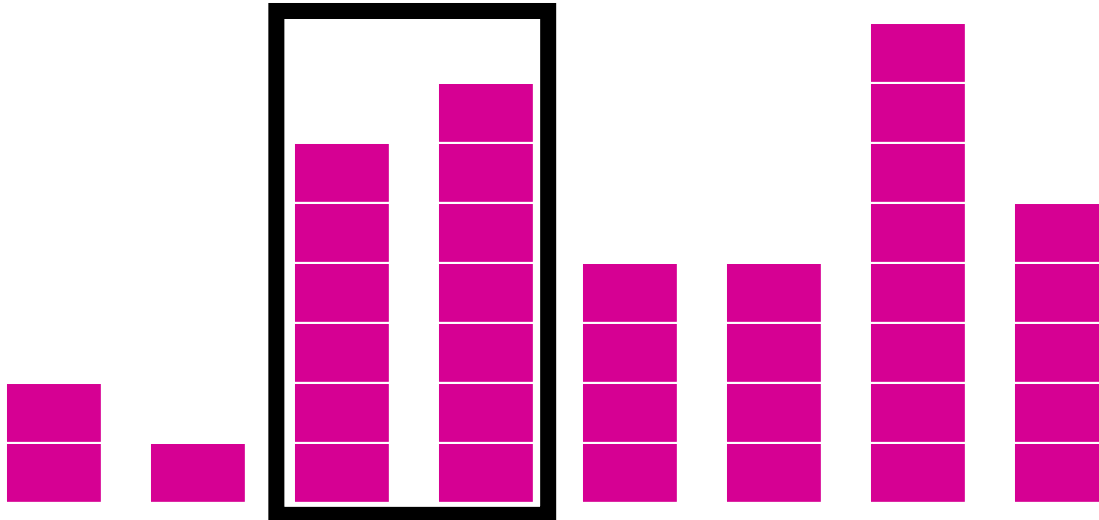
# Sorting: Step A3





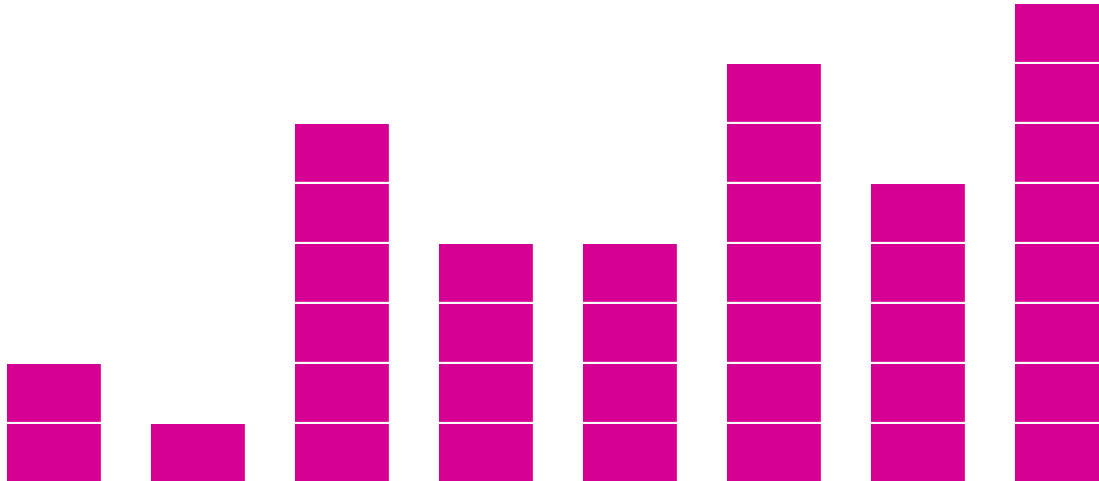
## Sorting: Step A3

Swap? Nee



# Volgorde na stap 7

- Herhaal dit proces tot alles op volgorde staat



# Bubble Sort - Officiële 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
```

# Voor nu

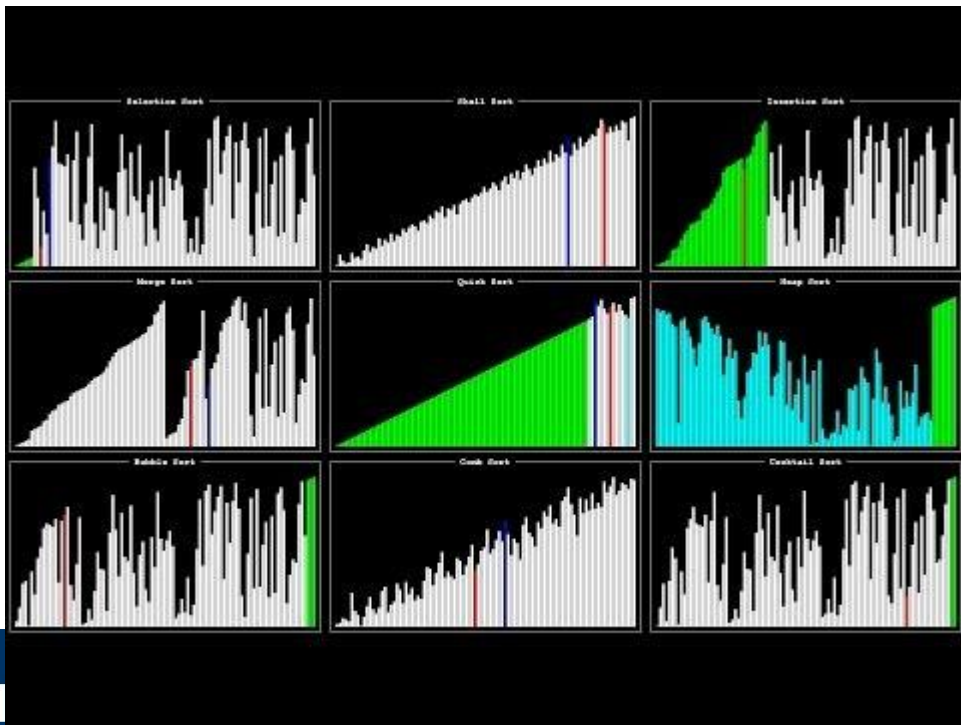
Bubble sort:

1. Neem een reeks met objecten
2. Begin bij objecten 1 en 2 uit de reeks
3. Staan ze op volgorde?
  - Zo ja; doe niets
  - Zo nee; wissel de objecten om
  - Verplaats naar de volgende twee objecten
  - Herhaal stap 3 tot je bij het einde van de reeks bent
4. Herhaal vanaf stap 2 tot er geen objecten meer te wisselen zijn

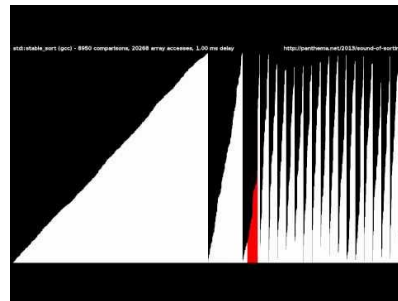
# Voor de nieuwsgierigen; in Python

```
# 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
```

## Even een kort intermezzo



En voor degene die nog meer willen zien over de afzonderlijke algoritmes:



Mijn favoriete algoritme staat helemaal op het laatst, raad eens waarom.

# Inhoud

- Pseudocode
- Booleaanse variabelen
- Logische operatoren

## Vorige keren

- Beslisstructuren zijn gebaseerd op booleaanse logica
  - Conditie is waar (true) of onwaar (false)

```
if <conditie>:
```

```
    statement
```

```
else:
```

```
    Statement
```

- We kennen verschillende types variabelen
  - String, integer, float



# Booleans

- We kennen naast strings etc. ook het type `bool`

```
bool1 = True
```

Zie ook hier geen `""` om

```
bool2 = False
```

True heen staan

```
print(type(bool1))
```

- Booleans zijn de kleinste variabelen die we kennen (1 of 0)
- We kunnen variabelen converteren naar booleans

## Even nog bij elkaar op een rijtje...

Variabele declaratie	Type(x)	Type variabele
X = "Hello World!"	<class 'str'>	String
X = 1	<class 'int'>	Integer (geheel getal)
X = 1.1	<class 'float'>	Float (gebroken getal)
X = True	<class 'bool'>	Boolean (True of False)

# Booleaanse vergelijking

- We kunnen vergelijkingen maken die een boolean opleveren
- Een vergelijking bestaat altijd uit twee waardes en een vergelijingsoperator
- Uit deze vergelijking volgt altijd een true of een false

<waarde> <operator> <waarde>

# Vergelijkingsoperatoren

Operator	Betekenis
==	Is gelijk aan
>	Groter dan
<	Kleiner dan
>=	Groter of gelijk aan
<=	Kleiner of gelijk aan
!=	Ongelijk aan

# Kleine test

Steek je hand omhoog als je denk dat de conversie werkt

1. `bool(0)`
2. `bool(1)`
3. `bool(true)`
4. `bool("True")`
5. `bool("true")`
6. `bool(1.0)`

# Inhoud

- Pseudocode
- Booleaanse variabelen
- Logische operatoren

# Logische operatoren

- Hiermee maak je een vergelijking die twee booleaanse vergelijkingen combineert.
- Syntax:

<waarde> <operator> <waarde>

<logische operator>

<waarde> <operator> <waarde>

# Logische operatoren

Eehm, heb je misschien ook een voorbeeld?

```
<waarde> <operator> <waarde>  
    <logische operator>  
<waarde> <operator> <waarde>
```

```
lunch == "aanwezig"  
and  
tijd >= 12.5
```



# En nu nog in code vorm

```
if lunch == "aanwezig" and tijd >= 12.5:  
    print("Lunchtijd!")  
elif lunch == "afwezig" and tijd >= 12.5:  
    print("Ga eens lunch kopen.")  
else:  
    print("Helaas nog geen lunchtijd.")
```

# Logische operatoren

Operator	Werking
and	Combinatie van twee booleaanse expressies waarbij beide juist moeten zijn
or	Combinatie van twee booleaanse expressies waarbij een van beide juist moet zijn
not	De negatie van de navolgende booleaanse expressie

## De truth tabel voor and

Expressie	Uitkomst
True and False	False
False and True	False
False and False	False
True and True	True

## De truth tabel voor or

Expressie	Uitkomst
True or False	True
False or True	True
False or False	False
True or True	True

## De truth tabel voor not

Expressie	Uitkomst
not True	False
not False	True

## Mutual exclusive or

- Een van beide moet waar zijn maar niet allebei
- De ene true sluit de ander uit

## De truth tabel voor xor

Expressie	Uitkomst
True ^ False	True
False ^ True	True
False ^ False	False
True ^ True	False

# Mutual eXclusive Or

- Voorbeelden van **xor** in het dagelijks leven
  - Wil je koffie of thee?
  - Is je auto rood of groen?
- Voorbeelden van **or** in het dagelijks leven
  - Wil je melk of suiker in je koffie?



# Samenvatting

- Pseudocode gebruik je om van te voren na te denken over hoe je een probleem wilt oplossen met je computer
- Booleans zijn kleine variabelen die handig zijn om snel iets te testen
- Booleans heb je nodig voor je conditionele controle
- Deze kan je combineren met logische operatoren

# Opdracht

- Ga naar OnderwijsOnline
- Maak Afvinkopdracht 3
- Bij problemen:
  - Google
  - Klasgenoten
  - Docent
- Lever deze in bij de praktijkdocent



INSERT COFFEE  
TO CONTINUE



# Verantwoording

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