

BIO-INFORMATICA

OWE 2A PYTHON II

PROGRAMMEREN IN PYTHON II



HAN_UNIVERSITY
OF APPLIED SCIENCES

CURRICULUM OVERZICHT

	Periode 1	Periode 2	Periode 3	Periode 4
1e jaar	Programmeren in Python (1)	Programmeren in Python (2)	Informatiesystemen met Python	Geautomatiseerde identificatie van eiwitten via sequentievergelijking
	Metabolisme	Opsporen van genetische mutaties bij erfelijke ziektes	Vergelijkende genomanalyse: 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,
			Eiwitten: carcinogenese en eiwitstructuren	Workflows
4e jaar	Stage, afstudeerstage of minor		Minor of afstudeerstage	

JAAR 1

Basis van programmeren in Python

Complexe structuren

Databases voor opslag van data

	Periode 1	Periode 2	Periode 3	Periode 4
1e jaar	Programmeren in Python (1)	Programmeren in Python (2)	Informatiesystemen met Python	Geautomatiseerde identificatie van eiwitten via sequentievergelijking
	Metabolisme	Opsporen van genetische mutaties bij erfelijke ziektes	Vergelijkende genoomanalyse: evolutie van virussen	Sequentie alignment (BLAST) en functionele eiwit analyse

Ontwikkelen van applicaties voor biologisch analyse en visualisatie op het web

OVERZICHT OWE 2A INFORMATICA

Week	Onderwerpen	
1	IDE: <u>PyCharm</u> Debuggen en testen PEP richtlijnen	<u>PEP tutorial</u>
2	Grafieken met Matplotlib	<u>Matplotlib.org</u>
3	Datastructuren	H9 SowP
4	Regular Expressions	H7 DiP
5	Object Oriëntatie	H10 & H11 SowP
6	Recursie	H12 SowP
7	Graphical User Interface	H13 SowP

AGENDA

- Terugblik Python I
- IDE: PyCharm
- Debuggen en testen
- PEP richtlijnen

TERUGBLIK

- Basis procedurele taal
 - Variabelen
 - Conditionele controle
 - Iteratieve controle
- Exceptions
- Bestanden lezen
- Lijsten

AGENDA

- Terugblik Python I
- IDE: PyCharm
- Debuggen en testen
- PEP richtlijnen

INSTALLER PROFESSIONAL EDITION



<https://www.jetbrains.com/pycharm/> 1 november 2018

PROFESSIONAL EDITION

- Installeer de professional edition
- Voorlopig zou de community edition volstaan maar later in het jaar hebben we de professional edition nodig
- Op <https://www.jetbrains.com/student/> kun je een gratis studentenlicentie aanvragen

AGENDA

- Terugblik Python I
- IDE: PyCharm
- Debuggen en testen
- PEP richtlijnen

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.
- <http://www.testbirds.com>

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
- Testers maken ook gebruik van testtools

HET VERIFICATIE PROCES

1. Blackbox of functioneel testen
2. Whitebox of structureel testen
 1. Door de code heenlopen met een debugger
3. Reviews of walkthroughs

VACATURES VOOR TESTEN

Wat
functie, trefwoorden of bedrijf

tester

Waar

Tip: Voer een plaats, provincie of postcode in bij "waar" om resultaten in omgeving weer te geven.

Sorteer op:
relevantie - [datum](#)

Werkgever/Recruiter
Werkgever (724)
Recruiter/Werving en Selectie (651)

Dienstverband
Fulltime (534)
Parttime (166)
Onbepaalde tijd (136)
Tijdelijk (45)
Bepaalde tijd (31)
[+](#) meer »

Plaats
Amsterdam (119)
Utrecht (98)
Eindhoven (69)
Apeldoorn (65)

Plaats je cv - In enkele seconden

vacatures voor tester

Pagina 1 van 1.375 vacatures

Tester / Test Engineer / Testcoördinator
CIMSOLUTIONS 2,4 ★
Deventer
€7.000 per maand
Wij spreken graag Testers / Test Engineers / Testcoördinatoren, o.a.
Voor de rollen Tester / testuitvoerder, test engineer, testcoördinator,
testadviseur,...
Gesponsord · [vacature opslaan](#)

Software tester / Test engineer
Ellips B.V.
Eindhoven
€3.000 - €6.000 per maand
[Deze werkgever werft via Indeed](#)
Ellips is een dynamische en vooruitstrevende ontwikkelaar van

TESTEN

- Waarom ga je testen? → <https://answer garden.ch/3870010>
- Bij welke software is testen heel belangrijk? → <https://answer garden.ch/3870008>
- Testen is:
- Vinden van fouten
- Kwaliteit meten van de applicatie
 - Het product vertrouwen geven

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 TESTSCRIPT

Test Case #: 2.2	Test Case Name: Change PIN	Page: 1 of 1
System: ATM	Subsystem: PIN	
Designed by: ABC	Design Date: 28/11/2004	
Executed by:	Execution Date:	
Short Description: Test the ATM Change PIN service		

Pre-conditions

The user has a valid ATM card - The user has accessed the ATM by placing his ATM card in the machine
The current PIN is 1234
The system displays the main menu

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Click the 'Change PIN' button	The system displays a message asking the user to enter the new PIN		
2	Enter '5555'	The system displays a message asking the user to confirm (re-enter) the new PIN		
3	Re-enter '5555'	The system displays a message of successful operation The system asks the user if he wants to perform other operations		
4	Click 'YES' button	The system displays the main menu		
5	Check post-condition 1			

Post-conditions

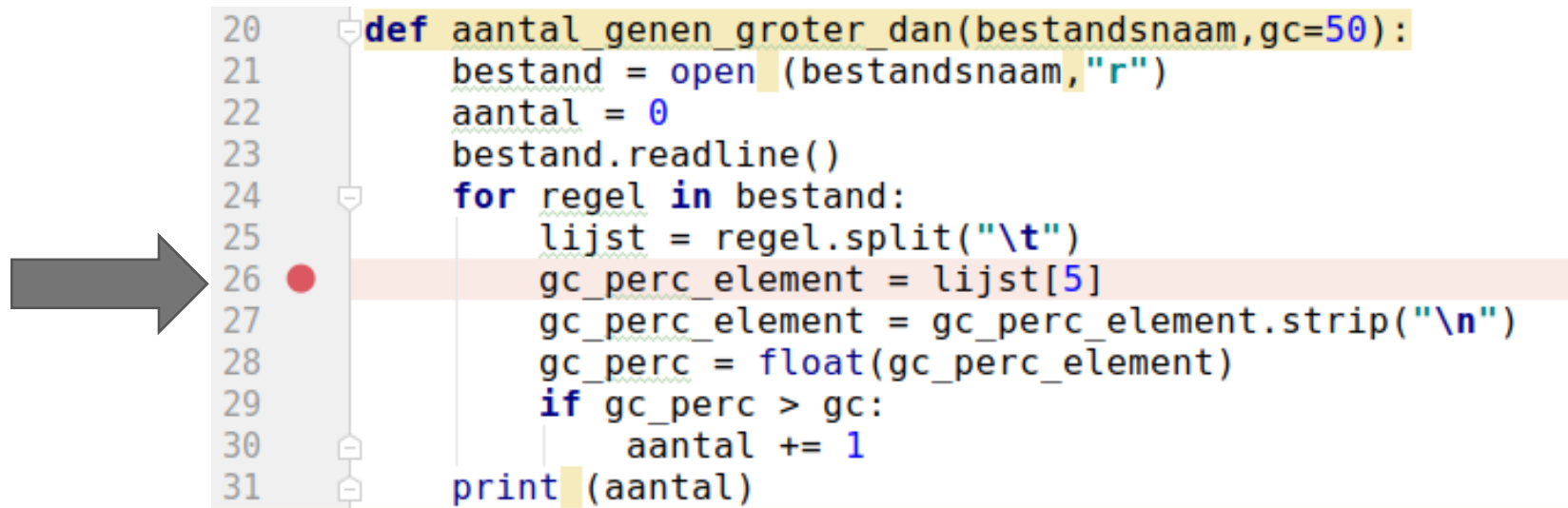
1. The new PIN '5555' is saved in the database

DEBUGGING CODE

- Wanneer je een onverwacht resultaat krijgt kun je je code doorlopen op zoek naar het probleem
- Met de debugger kun je stap voor stap door je code heen lopen

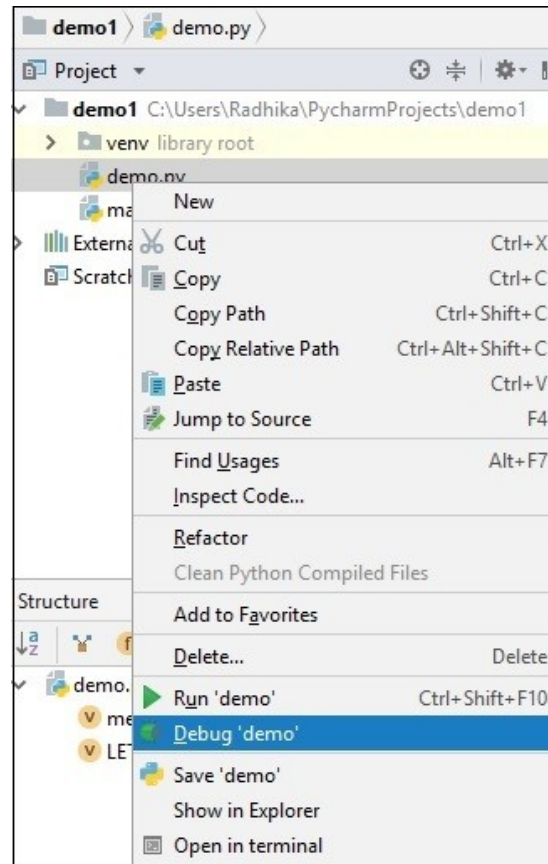
DEBUGGING CODE

- Plaats een breakpoint in je code door in de kantlijn te klikken



DEBUG

In plaats van te runnen start je het programma in debug mode



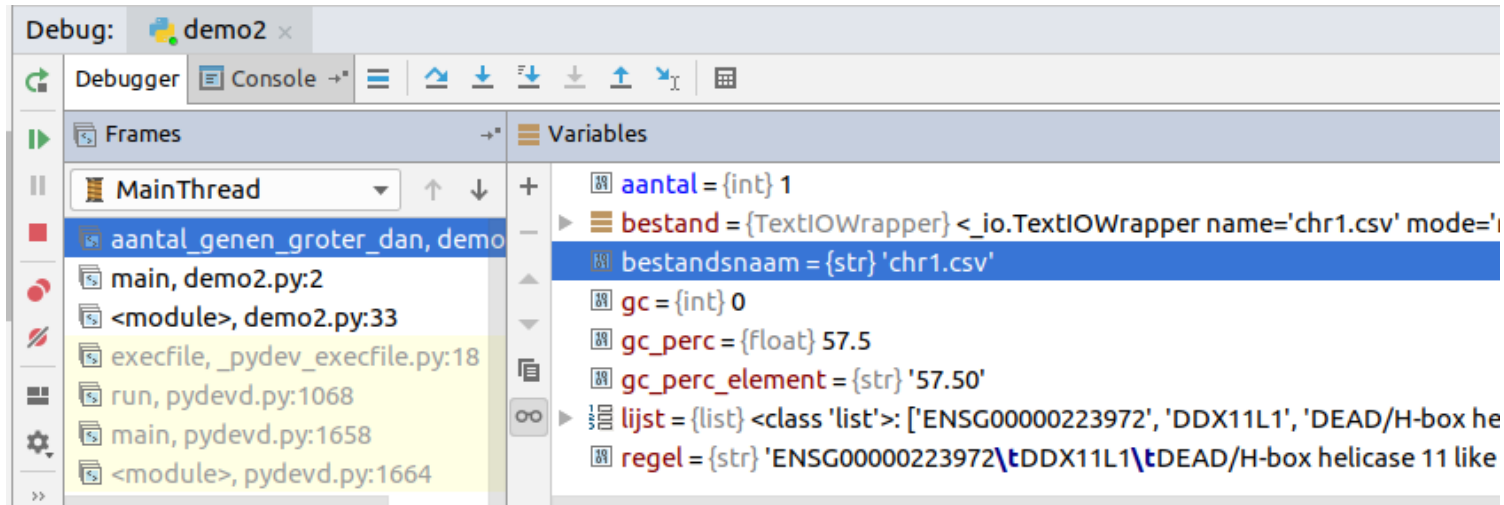
Screenshot eigen werk

DEBUGGEN: WEERGAVE TUSSENRESULTATEN

Breakpoint: een keer klikken in de kantlijn

```
23     aantal = 0    aantal: 0
24     bestand.readline()
25     for regel in bestand:    regel: 'ENSG00000223972\tDDX11L1\tDEAD/H-box 1
26         lijst = regel.split("\t")    lijst: <class 'list'>: ['ENSG000002239
27         gc_perc_element = lijst[5]
28         gc_perc_element = gc_perc_element.strip("\n")
29         gc_perc = float(gc_perc_element)
30         if gc_perc > gc:
31             aantal += 1
32     print(aantal)
33     main()
```

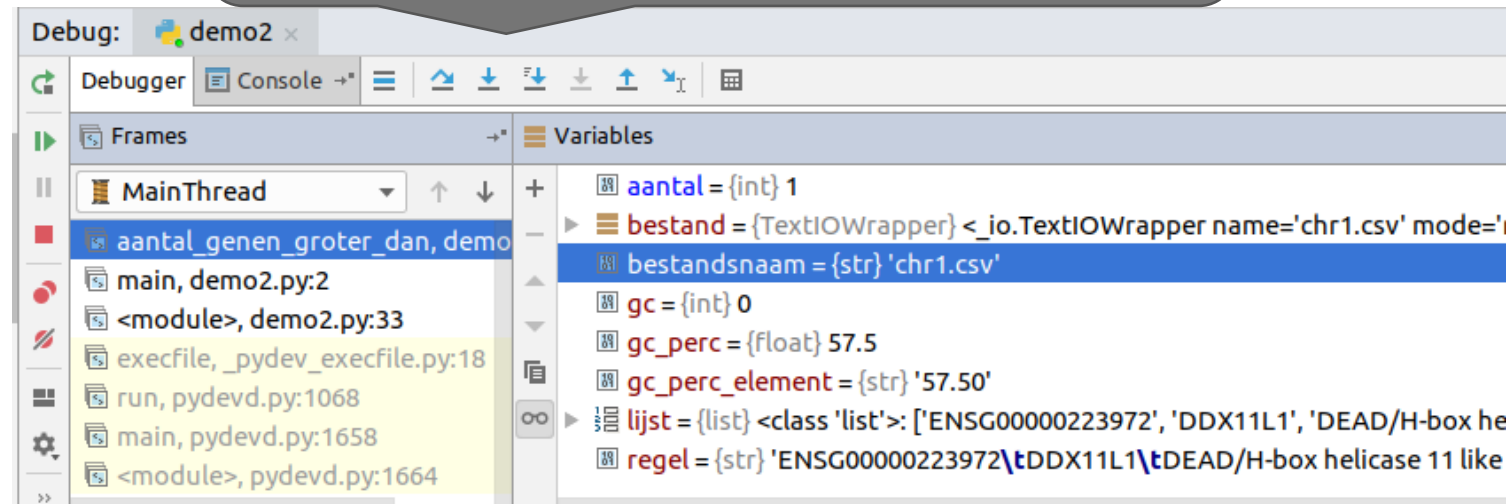
DEBUGGEN: WEERGAVE VARIABLEN



DEBUGGEN

De code kun je in slowmotion doorlopen

Buttons voor het stapsgewijs doorlopen van de code



SAMENVATTING

Debugging en testen is essentieel voor software ontwikkeling

AGENDA

- Terugblik Python I
- IDE: PyCharm
- Debuggen en testen
- PEP richtlijnen

CONVENTIES VOOR PYTHON

- PEP (Python Enhancement Proposals)
- PEP 8 is een richtlijn (conventie) voor de layout van Python
- "Programmeercode wordt vaker gelezen dan geschreven"

VOORBEELDEN VAN CONVENTIES

Een statement per regel

Fout

```
1 print ("Berlin"); print ("Cape Town")
2
3 if x == 1: print ("Amsterdam")
4
```

Goed

```
1 print ("Berlin")
2 print ("Cape Town")
3
4 if x == 1:
5     print ("Amsterdam")
6
```

ENKELE VOORSCHRIFTEN VAN PEP8

- Een regel per statement
- Na een functie twee witregels
- Regels maximaal 79 karakters
 - Docstrings en commentaar MAX 72 karakters
- Geen spaties bij toekenning waardes in parameters

DOCUMENTATION STRINGS

Een regel

- Alleen gebruiken als functie
geen parameters accepteert
- Gebruiken als er alleen een
return statement in functie staat
- Voorbeeld: tijd retourneren

Meerdere regels

Bestaat uit:

- Samenvatting/doel van die
functie
- Lege regel
- Input parameters
- Output (return) parameters

DOCUMENTATION STRINGS

ONE LINE

```
def get_current_working_directory():  
    """return working directory - string"""  
    return os.getcwd()
```

```
def get_current_date_time():  
    """return current date and time"""  
    return datetime.datetime.now()
```

DOCUMENTATION STRINGS

MULTIPLE LINES

```
def fasta_reader(bestandsnaam):  
    """Inlezen van fasta bestand  
  
    Input:  
    bestandsnaam - string, naam bestand  
  
    Output:  
    headers - lijst, headers van fasta bestand  
    sequenties - lijst, sequenties van fasta bestand  
    """  
    open_bestand = open(bestandsnaam, "r")  
    headers = []  
    sequenties = []  
    for line in open_bestand:  
        ...
```

DOCUMENTATION STRINGS VS COMMENTAAR

Documentation strings

Voor: mensen die je code gaan gebruiken zonder dat ze het hoeven te begrijpen

Docstrings kunnen direct gebruikt worden als documentatie over je code

Commentaar

Voor: mensen die je code moeten begrijpen

Commentaar wordt niet gebruikt als documentatie omdat:

- Het gaat over de code zelf
- Het gaat niet over hoe je het kunt gebruiken

PEPS

Meta-PEPs (PEPs about PEPs or Processes)

	PEP	PEP Title	PEP Author(s)
P	1	PEP Purpose and Guidelines	Warsaw, Hylton, Goodger, Coghlan
P	4	Deprecation of Standard Modules	Cannon, von Löwis
P	5	Guidelines for Language Evolution	Prescod
P	6	Bug Fix Releases	Aahz, Baxter
P	7	Style Guide for C Code	GvR, Warsaw
	8	Style Guide for Python Code	GvR, Warsaw, Coghlan
P	10	Voting Guidelines	Warsaw
P	11	Removing support for little used platforms	von Löwis, Cannon
P	12	Sample reStructuredText PEP Template	Goodger, Warsaw
P	573	Module State Access from C Extension Methods	Viktorin, Coghlan, Snow, Plch

PEP8

[Python](#) >>> [Python Developer's Guide](#) >>> [PEP Index](#) >>> [PEP 8 -- Style Guide for Python Code](#)

PEP 8 -- Style Guide for Python Code

PEP:	8
Title:	Style Guide for Python Code
Author:	Guido van Rossum <guido at python.org>, Barry Warsaw <barry at python.org>, Nick Coghlan <ncoghlan at gmail.com>
Status:	Active
Type:	Process
Created:	05-Jul-2001
Post-History:	05-Jul-2001, 01-Aug-2013

PEP8 DOCUMENTATIE

<https://buildmedia.readthedocs.org/media/pdf/pep8/release-1.7.x/pep8.pdf>

pep8 documentation

Release 1.7.1.dev0

Johann C. Rocholl, Florent Xicluna, Ian Lee

PEP8 CHECK TOOL - PYLINT

Met `sudo apt install pylint` installeer je een tool die je code op pep8 conventies kan controleren

PEP8 ONLINE

<http://pep8online.com/>

Check results

Save ▾

Share

Code	Line	Column	Text
E303	26	1	too many blank lines (3)
E128	67	13	continuation line under-indented for visual indent
W292	94	7	no newline at end of file

Your code

```
1 #!/usr/bin/python
2 # -*- coding: utf-8 -*-
3 """Module om te rekenen aan genen in bestanden met geninformatie
4
5 Auteur: Martijn van der Bruggen
6 Copyright: 2018 HAN University of Applied Science
```

PYCHARM EN PEP8

- In PyCharm kun je controle op het naleven van de PEP8 richtlijnen aanzetten onder:
- File>settings>Inspections
- Tip:
- Voor regellengte aan te geven met een verticale streep:
 - File → settings → editor → code style → markdown → hard wrap at: 72 of 79

SAMENVATTING

- PEP8 is een richtlijn om Python code te schrijven die leesbaar is
- Het gebruik is in de meeste projecten een vereiste en zorgt ervoor dat je code beter begrijpbaar is

OPDRACHT (OPTIONEEL)

Accepteer de opdracht in de github classroom: https://classroom.github.com/a/Q8gNvm_S

1. Maak de code volgens de PEP8 regels
2. Voeg correcte exception handling toe → probeer de code te 'slopen'



Time cap:
20 minuten

OPDRACHT

DNA bestaat uit intronen en exonen. Je krijgt een fasta bestand waarin de volledige DNA sequentie staat. Vervolgens krijg je een fasta bestand waar de intronen in staan. De opdracht is om de volledige sequentie DNA te converteren naar mRNA door de intronen eruit te knippen.

Gebruik de volgende fasta bestanden: complete.fasta & intronen.fasta die op Teams staan onder het vak informatica. **Bonus**

Transleer het mRNA naar een eiwit door deze op te delen in codons en de codons te vertalen naar het corresponderende aminozuur.



Mocht je eerder klaar zijn: ga je code eens doorlopen met een debugger

VERANTWOORDING

- In deze uitgave is géén auteursrechtelijk beschermd werk opgenomen
- Alle teksten © Teuntje Peeters/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